Subject: [RFC PATCH 0/31] An introduction and A path for merging network namespace work
Posted by ebiederm on Thu, 25 Jan 2007 18:55:04 GMT
View Forum Message <> Reply to Message

The idea of a network namespace is fundamentally quite simple. We create a mechanism that from the users perspective allows creation of separate instances of the network stack. When combined with mechanism like chroot this results in a much more complete isolation. When seen in the context of application migration this allows for taking your IP address and other global identifiers with you.

What does this mean in the context of the networking stack? The basic idea is to tag processes with a network namespace that is used when they create new sockets or otherwise initiate a new fresh communication with the networking stack. The idea is to tag all sockets with a network namespace they will always be in and all operations on them will be relative to. The idea is to tag all network devices with a network namespace they are a member of, but may be changed during the lifetime of a device.

Mostly a network namespace at it's most basic level is about names. It is about creating a view of the networking stack where you can name the network devices that are members anything you want. Likewise for iptables rules and all of the rest of the state. It is a lot like creating a new directory in a filesystem. The underlying data structures don't really change just the users view of those data structures, and we continue to have a single network stack.

My goal today is that even if we can't agree on a specific set of patches that we come to an agreement on roughly what those patches should accomplish, and what process we should go through to get them merged.

For implementing a network namespace the core problem is that there is a lot of networking code, and it is continually evolving. This means that the task of implementing a network namespace is not a small one, a lot of code must be read, touched and updated, while hoping someone doesn't change something important before you get your changes in. To do this sanely means we need an incremental path to our goal, that allows small pieces to be reviewed and merged as they are ready.

The path I am recommending today is to first lay down some basic infrastructure. Then one layer at a time modify the existing code to handle multiple simultaneous network namespaces but to modify each component of that layer to refuse to operate in the context of anything but the initial network namespace, thus preventing code that has not yet been updated with situations it does not know how to deal with.

Eventually this will get down to the real meat of the problem and practical things like ipv4 sockets will work.

This should allow for a network stack that compiles, builds and works at each step of the way. Not too far into the process support for multiple network namespaces that works should be available with the limitation that except for the initial network namespace all of the rest will look like a kernel with most parts of the networking stack compiled out, but within those parts that are present it should be fully useable.

To make my thinking clear I have provided a initial patchset, that makes quite a bit of progress especially in laying the ground work. My goal is to have the question does this basic path make sense?

To that end I have omitted posting some of the prerequisite cleanup and infrastructure patches (like my sysctl work), that are just noise in this context, and I have failed to rebase my patchset against Dave Miller's latest networking tree. Those are important details but they are not important to this conversation.

If my basic path and the basic patches look like they are heading in the right direction we can start moving towards what needs to happen to ensure a review of the patches, and what we need to do to start merging them. If the basic path does not appear reasonable well that would be good to know as well.

There are essentially two different approaches to modify networking code to handle multiple network namesspaces. Either all of the global variables can be replicated once for each network namespace and we

build up parallel namespace specific data structures. Or the data
elements in the data structure are tagged, with what namespace they
belong to and we filter them. It depends on the context which
is most appropriate and easier. As a general rule large hash tables
call for filtering and a small global variable set calls for simply
having multiple instances of the data structure.

The biggest intrusion I expect to see in the logic of the networking
stack is initialization and tear down. As we need to initialize
and clean up all of those per network namespace variables when
we create and destroy and network namespace.

A git tree with all of my patches against 2.6.20-rc5 is available at:
git://git.kernel.org/pub/scm/linux/kernel/git/ebiederm/linux-2.6-netns.git

In addition to what I have posted here and all of it's prerequisites
the tree includes further patches that get the basics of ipv4 and
iptables working. So people who are interested actually have
something more or less useful to play with.

At a big practical level what I don't yet see is how exactly the
inifiniband/rdma network subsystem fits into network namespaces yet.
Not at the ipoib layer but at the native layer. I think I want the
ability to say each pkey of each IB device can potentially be in
a different namespace or possibly each different queue pair. Suggestions
are welcome. I don't quite have my head wrapped around that the user
space API there yet.

I suppose on the infiniband/rdma side I should dig up all interactions
with user space and simply fail if that user is not in the initial
network namespace as a start. At the very least this is necessary
given how many calls the connection manager makes into the IP stack.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 1/31] net: Add net_namespace_type.h to allow for per network namespace variables.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:03 GMT
View Forum Message <> Reply to Message

The problem:
  To properly implement a ``level 2'' network namespace we need to
move many of the networking stack global variables into the network
namespace. We want to keep it explicit that the code is accessing a
variable in a network namespace. We want to be able to completely
compile out the network namespace support so we can do comparitive
performance testing, and so to not penalize users who don't need
network namespace support. Because the network stack is a moving
target we want something simple that allows for the bulk of the
changes to be merged before we enable network namespace support.

My biggest challenge when looking into this was to find an approach
that would allow the code to compile out, in a way that does not yield
any performance overhead and does not make the code ugly. While
playing with the different possibilities I discovered that gcc will
not pass 0 byte structures that are arguments to functions and instead
will simply optmize them away. This appears to be true on i386 all of
the way back to gcc-2.95 and I verified that it also works with gcc
4.1 on x86_64. Since this is part of the ABI I never expect it to
change. Hopefully gcc uses this nice optimization on all
architectures, I suspect so as C++ allows passing function arguments
of type void in certain circumstances.

Using this observation I was able to come up with an network namespace
implementation network namespace code that allows the changes to
completely compile out when we don't build the kernel with network
namespace support.

This patch implements my dummy network namespace support that should
completely compiles out. Further patches will add the real version.
Starting with the dummy gives a quick hint of where I am going and
allows for dependencies to be overcome.

When doing my proof of concept implementation one of the other
problems I had was that as the network stack comes in so many modular
pieces figuring out how to get their global variables into the network
namespace structure was a challenge. The basic technique used by our
per cpu variables for having the linker build and dynamically change
structures for us appears applicable here and a lot less nuisance then
what I did before so I am implementing a tailored version of that
technique as well, and again this makes it very simple to compile the
code out.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/net_namespace_type.h | 52 ++++++++++++++++++++++++++++++++++++
 1 files changed, 52 insertions(+), 0 deletions(-)

```
diff --git a/include/linux/net_namespace_type.h b/include/linux/net_namespace_type.h
new file mode 100644
index 0000000..8173f59
--- /dev/null
+++ b/include/linux/net_namespace_type.h
@@ -0,0 +1,52 @@
+/*
+ * Definition of the network namespace reference type
+ * And operations upon it.
+ */
+#ifndef __LINUX_NET_NAMESPACE_TYPE_H
+#define __LINUX_NET_NAMESPACE_TYPE_H
+
+#define __pernetname(name) per_net__##name
+
+typedef struct {} net_t;
+
+#define __data_pernet
+
+/* Look up a per network namespace variable */
+static inline unsigned long __per_net_offset(net_t net) { return 0; }
+
+/* Like per_net but returns a pseudo variable address that must be moved
+ * __per_net_offset() bytes before it will point to a real variable.
+ * Useful for static initializers.
+ */
+#define __per_net_base(name)   __pernetname(name)
+
+/* Get the network namespace reference from a per_net variable address */
+#define net_of(ptr, name) ({ net_t net; ptr; net; })
+
+/* Look up a per network namespace variable */
+#define per_net(name, net) \
+ (*(__per_net_offset(net), &__per_net_base(name)))
+
+/* Are the two network namespaces the same */
+static inline int net_eq(net_t a, net_t b) { return 1; }
+/* Get an unsigned value appropriate for hashing the network namespace */
+static inline unsigned int net_hval(net_t net) { return 0; }
+
+/* Convert to and from to and from void pointers */
+static inline void *net_to_voidp(net_t net) { return NULL; }
+static inline net_t net_from_voidp(void *ptr) { net_t net; return net; }
+
+static inline int null_net(net_t net) { return 0; }
+
+#define DEFINE_PER_NET(type, name)    \
+ __data_pernet __typeof__(type) __pernetname(name)
```

```
+
+#define DECLARE_PER_NET(type, name) \
+ extern __typeof__(type) __pernetname(name)
+
+#define EXPORT_PER_NET_SYMBOL(var) \
+ EXPORT_SYMBOL(__pernetname(var))
+#define EXPORT_PER_NET_SYMBOL_GPL(var) \
+ EXPORT_SYMBOL_GPL(__pernetname(var))
+
+#endif /* __LINUX_NET_NAMESPACE_TYPE_H */
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 2/31] net: Implement a place holder network namespace
Posted by ebiederm on Thu, 25 Jan 2007 19:00:04 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Many of the changes to the network stack will simply be adding a
network namespace parameter to function calls or moving variables
from globals to being per network namespace. When those variables
have initializers that cannot statically compute the proper value,
a function that runs at the creation and destruction of network
namespaces will need to be registered, and the logic will need to
be changed to accomidate that.

Adding unconditional support for these functions ensures that even when
everything else is compiled out the modified network stack logic will
continue to run correctly.

This patch adds struct pernet_operations that has an init (constructor)
and an exit (destructor) method. When registered the init method
is called for every existing namespace, and when unregistered the
exit method is called for every existing namespace. When a new
network namespace is created all of the init methods are called
in the order in which they were registered, and when a network namespace
is destroyed the exit methods are called in the reverse order in
which they were registered.

There are two distinct types of pernet_operations recognized: subsys and
device. At creation all subsys init functions are called before device

init functions, and at destruction all device exit functions are called
before subsys exit function. For other ordering the preservation
of the order of registration combined with the various kinds of
kernel initcalls should be sufficient.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 include/net/net_namespace.h |   62 +++++++++++++++++
 net/core/Makefile           |    2 +-
 net/core/net_namespace.c    |  149 ++++++++++++++++++++++++++++++++++++++++++++++
 3 files changed, 212 insertions(+), 1 deletions(-)

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
new file mode 100644
index 0000000..06a9ba1
--- /dev/null
+++ b/include/net/net_namespace.h
@@ -0,0 +1,62 @@
+/*
+ * Operations on the network namespace
+ */
+#ifndef __NET_NET_NAMESPACE_H
+#define __NET_NET_NAMESPACE_H
+
+#include <asm/atomic.h>
+#include <linux/workqueue.h>
+#include <linux/nsproxy.h>
+#include <linux/net_namespace_type.h>
+
+/* How many bytes in each network namespace should we allocate
+ * for use by modules when they are loaded.
+ */
+#ifdef CONFIG_MODULES
+# define PER_NET_MODULE_RESERVE 2048
+#else
+# define PER_NET_MODULE_RESERVE 0
+#endif
+
+struct net_namespace_head {
+ atomic_t count;  /* To decided when the network namespace
+     * should go
+     */
+ atomic_t use_count; /* For references we destroy on demand */
+ struct list_head list;
+ struct work_struct work;
+};
+
+static inline net_t get_net(net_t net) { return net; }
```

```
+static inline void put_net(net_t net) {}
+static inline net_t hold_net(net_t net) { return net; }
+static inline void release_net(net_t net) {}
+
+#define __per_net_start ((char *)0)
+#define __per_net_end ((char *)0)
+
+static inline int copy_net(int flags, struct task_struct *tsk) { return 0; }
+
+/* Don't let the list of network namespaces change */
+static inline void net_lock(void) {}
+static inline void net_unlock(void) {}
+
+#define for_each_net(VAR) if (1)
+
+extern net_t net_template;
+
+#define NET_CREATE 0x0001 /* A network namespace has been created */
+#define NET_DESTROY 0x0002 /* A network namespace is being destroyed */
+
+struct pernet_operations {
+ struct list_head list;
+ int (*init)(net_t net);
+ void (*exit)(net_t net);
+};
+
+extern int register_pernet_subsys(struct pernet_operations *);
+extern void unregister_pernet_subsys(struct pernet_operations *);
+extern int register_pernet_device(struct pernet_operations *);
+extern void unregister_pernet_device(struct pernet_operations *);
+
+#endif /* __NET_NET_NAMESPACE_H */
diff --git a/net/core/Makefile b/net/core/Makefile
index 73272d5..554dbdc 100644
--- a/net/core/Makefile
+++ b/net/core/Makefile
@@ -3,7 +3,7 @@
 #

 obj-y := sock.o request_sock.o skbuff.o iovec.o datagram.o stream.o scm.o \
-  gen_stats.o gen_estimator.o
+  gen_stats.o gen_estimator.o net_namespace.o

 obj-$(CONFIG_SYSCTL) += sysctl_net_core.o

diff --git a/net/core/net_namespace.c b/net/core/net_namespace.c
new file mode 100644
index 0000000..4ae266d
```

```
--- /dev/null
+++ b/net/core/net_namespace.c
@@ -0,0 +1,149 @@
+#include <linux/rtnetlink.h>
+#include <net/net_namespace.h>
+
+/*
+ * Our network namespace constructor/destructor lists
+ */
+
+static LIST_HEAD(pernet_list);
+static struct list_head *first_device = &pernet_list;
+static DEFINE_MUTEX(net_mutex);
+net_t net_template;
+
+static int register_pernet_operations(struct list_head *list,
+          struct pernet_operations *ops)
+{
+ net_t net, undo_net;
+ int error;
+
+ error = 0;
+ list_add_tail(&ops->list, list);
+ for_each_net(net) {
+  if (ops->init) {
+   error = ops->init(net);
+   if (error)
+    goto out_undo;
+  }
+ }
+out:
+ return error;
+
+out_undo:
+ /* If I have an error cleanup all namespaces I initialized */
+ list_del(&ops->list);
+ for_each_net(undo_net) {
+  if (net_eq(undo_net, net))
+   goto undone;
+  if (ops->exit)
+   ops->exit(undo_net);
+ }
+undone:
+ goto out;
+}
+
+static void unregister_pernet_operations(struct pernet_operations *ops)
+{
```

```
+ net_t net;
+
+ list_del(&ops->list);
+ for_each_net(net)
+  if (ops->exit)
+   ops->exit(net);
+}
+
+/**
+ *      register_pernet_subsys - register a network namespace subsystem
+ * @ops:  pernet operations structure for the subsystem
+ *
+ * Register a subsystem which has init and exit functions
+ * that are called when network namespaces are created and
+ * destroyed respectively.
+ *
+ * When registered all network namespace init functions are
+ * called for every existing network namespace.  Allowing kernel
+ * modules to have a race free view of the set of network namespaces.
+ *
+ * When a new network namespace is created all of the init
+ * methods are called in the order in which they were registered.
+ *
+ * When a network namespace is destroyed all of the exit methods
+ * are called in the reverse of the order with which they were
+ * registered.
+ */
+int register_pernet_subsys(struct pernet_operations *ops)
+{
+ int error;
+ mutex_lock(&net_mutex);
+ error =  register_pernet_operations(first_device, ops);
+ mutex_unlock(&net_mutex);
+ return error;
+}
+EXPORT_SYMBOL_GPL(register_pernet_subsys);
+
+/**
+ *      unregister_pernet_subsys - unregister a network namespace subsystem
+ * @ops: pernet operations structure to manipulate
+ *
+ * Remove the pernet operations structure from the list to be
+ * used when network namespaces are created or destoryed.  In
+ * addition run the exit method for all existing network
+ * namespaces.
+ */
+void unregister_pernet_subsys(struct pernet_operations *module)
+{
```

```
+ mutex_lock(&net_mutex);
+ unregister_pernet_operations(module);
+ mutex_unlock(&net_mutex);
+}
+EXPORT_SYMBOL_GPL(unregister_pernet_subsys);
+
+/**
+ *      register_pernet_device - register a network namespace device
+ * @ops:  pernet operations structure for the subsystem
+ *
+ * Register a device which has init and exit functions
+ * that are called when network namespaces are created and
+ * destroyed respectively.
+ *
+ * When registered all network namespace init functions are
+ * called for every existing network namespace.  Allowing kernel
+ * modules to have a race free view of the set of network namespaces.
+ *
+ * When a new network namespace is created all of the init
+ * methods are called in the order in which they were registered.
+ *
+ * When a network namespace is destroyed all of the exit methods
+ * are called in the reverse of the order with which they were
+ * registered.
+ */
+int register_pernet_device(struct pernet_operations *ops)
+{
+ int error;
+ mutex_lock(&net_mutex);
+ error = register_pernet_operations(&pernet_list, ops);
+ if (!error && (first_device == &pernet_list))
+  first_device = &ops->list;
+ mutex_unlock(&net_mutex);
+ return error;
+}
+EXPORT_SYMBOL_GPL(register_pernet_device);
+
+/**
+ *      unregister_pernet_device - unregister a network namespace netdevice
+ * @ops: pernet operations structure to manipulate
+ *
+ * Remove the pernet operations structure from the list to be
+ * used when network namespaces are created or destoryed.  In
+ * addition run the exit method for all existing network
+ * namespaces.
+ */
+void unregister_pernet_device(struct pernet_operations *ops)
+{
```

```
+ mutex_lock(&net_mutex);
+ if (&ops->list == first_device)
+  first_device = first_device->next;
+ unregister_pernet_operations(ops);
+ mutex_unlock(&net_mutex);
+}
+EXPORT_SYMBOL_GPL(unregister_pernet_device);
+
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 3/31] net: Add a network namespace parameter to tasks

Posted by ebiederm on Thu, 25 Jan 2007 19:00:05 GMT

View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This is the network namespace from which all which all sockets
and anything else under user control ultimately get their network
namespace parameters.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 include/linux/nsproxy.h |    2 ++
 1 files changed, 2 insertions(+), 0 deletions(-)

diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
index 0b9f0dc..cc76610 100644
--- a/include/linux/nsproxy.h
+++ b/include/linux/nsproxy.h
@@ -3,6 +3,7 @@

 #include <linux/spinlock.h>
 #include <linux/sched.h>
+#include <linux/net_namespace_type.h>

 struct mnt_namespace;
 struct uts_namespace;
@@ -28,6 +29,7 @@ struct nsproxy {
  struct ipc_namespace *ipc_ns;
  struct mnt_namespace *mnt_ns;
  struct pid_namespace *pid_ns;
```

```
+ net_t       net_ns;
 };
 extern struct nsproxy init_nsproxy;

--
1.4.4.1.g278f
```

Subject: [PATCH RFC 4/31] net: Add a network namespace tag to struct net_device
Posted by ebiederm on Thu, 25 Jan 2007 19:00:06 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Please note that network devices do not increase the count
count on the network namespace. The are inside the network
namespace and so the network namespace tag is in the nature
of a back pointer and so getting and putting the network namespace
is unnecessary.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/netdevice.h | 4 ++++
 1 files changed, 4 insertions(+), 0 deletions(-)

```
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 4cb8b39..6a1579d 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -38,6 +38,7 @@
 #include <linux/device.h>
 #include <linux/percpu.h>
 #include <linux/dmaengine.h>
+#include <linux/net_namespace_type.h>

 struct vlan_group;
 struct ethtool_ops;
@@ -525,6 +526,9 @@ struct net_device
  void                    (*poll_controller)(struct net_device *dev);
 #endif

+ /* Network namespace this network device is inside */
+ net_t   nd_net;
```

```
+
  /* bridge stuff */
  struct net_bridge_port *br_port;

--
1.4.4.1.g278f
```

Subject: [PATCH RFC 5/31] net: Add a network namespace parameter to struct sock

Posted by ebiederm on Thu, 25 Jan 2007 19:00:07 GMT

View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Sockets need to get a reference to their network namespace,
or possibly a simple hold if someone registers on the network
namespace notifier and will free the sockets when the namespace
is going to be destroyed.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/net/inet_timewait_sock.h | 1 +
 include/net/sock.h | 3 +++
 2 files changed, 4 insertions(+), 0 deletions(-)

```
diff --git a/include/net/inet_timewait_sock.h b/include/net/inet_timewait_sock.h
index f7be1ac..162c2b9 100644
--- a/include/net/inet_timewait_sock.h
+++ b/include/net/inet_timewait_sock.h
@@ -115,6 +115,7 @@ struct inet_timewait_sock {
 #define tw_refcnt  __tw_common.skc_refcnt
 #define tw_hash    __tw_common.skc_hash
 #define tw_prot    __tw_common.skc_prot
+#define tw_net     __tw_common.skc_net
   volatile unsigned char tw_substate;
   /* 3 bits hole, try to pack */
   unsigned char  tw_rcv_wscale;
diff --git a/include/net/sock.h b/include/net/sock.h
index 03684e7..5bf6bb5 100644
--- a/include/net/sock.h
+++ b/include/net/sock.h
@@ -105,6 +105,7 @@ struct proto;
```

```
  * @skc_refcnt: reference count
  * @skc_hash: hash value used with various protocol lookup tables
  * @skc_prot: protocol handlers inside a network family
+ * @skc_net: reference to the network namespace of this socket
  *
  * This is the minimal network layer representation of sockets, the header
  * for struct sock and struct inet_timewait_sock.
@@ -119,6 +120,7 @@ struct sock_common {
  atomic_t  skc_refcnt;
  unsigned int  skc_hash;
  struct proto  *skc_prot;
+ net_t    skc_net;
 };

 /**
@@ -195,6 +197,7 @@ struct sock {
 #define sk_refcnt  __sk_common.skc_refcnt
 #define sk_hash   __sk_common.skc_hash
 #define sk_prot   __sk_common.skc_prot
+#define sk_net   __sk_common.skc_net
  unsigned char  sk_shutdown : 2,
     sk_no_check : 2,
     sk_userlocks : 4;
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 6/31] net: Add a helper to get a reference to the initial network namespace.

Posted by ebiederm on Thu, 25 Jan 2007 19:00:08 GMT

View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

The initial network namespace is special and we need to use it for various
things. Probably the biggest initial use will be to ensure code that
can't cope with multiple namespaces only sees the initial network namespace.

For that reason and because getting at the initial network namespace is just
a little clumsy add a helper function.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 include/net/net_namespace.h |    6 ++++++
 1 files changed, 6 insertions(+), 0 deletions(-)

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index 06a9ba1..9208e2e 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -27,6 +27,12 @@ struct net_namespace_head {
  struct work_struct work;
 };

+/* Get the initial network namespace */
+static inline net_t init_net(void)
+{
+ return init_nsproxy.net_ns;
+}
+
 static inline net_t get_net(net_t net) { return net; }
 static inline void put_net(net_t net) {}
 static inline net_t hold_net(net_t net) { return net; }
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 7/31] net: Make /proc/net per network namespace
Posted by ebiederm on Thu, 25 Jan 2007 19:00:09 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch makes /proc/net per network namespace. It modifies the global
variables proc_net and proc_net_stat to be per network namespace.
The proc_net file helpers are modified to take a network namespace argument,
and all of their callers are fixed to pass init_net() for that argument.
This ensures that all of the /proc/net files are only visible and
usable in the initial network namespace until the code behind them
has bee updated to be handle multiple network namespaces.

Making /proc/net per namespace is necessary as at least some files
in /proc/net depend upon the set of network devices which is per
network namespace, and even more files in /proc/net have contents
that are relevant to a single network namespace.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/isdn/divert/divert_procfs.c | 7 +-
drivers/isdn/hardware/eicon/diva_didd.c | 5 +-
drivers/isdn/hysdn/hysdn_procconf.c | 4 +-
drivers/net/bonding/bond_main.c | 7 +-
drivers/net/hamradio/bpqether.c | 5 +-
drivers/net/hamradio/scc.c | 4 +-
drivers/net/hamradio/yam.c | 5 +-
drivers/net/ibmveth.c | 6 +-
drivers/net/pppoe.c | 5 +-
drivers/net/tc35815.c | 1 -
drivers/net/tokenring/lanstreamer.c | 4 +-
drivers/net/tokenring/olympic.c | 9 +-
drivers/net/wireless/hostap/hostap_main.c | 7 +-
drivers/net/wireless/strip.c | 5 +-
fs/proc/Makefile | 1 +
fs/proc/internal.h | 5 +
fs/proc/proc_net.c | 126 ++++++++++++++++++++++
fs/proc/root.c | 8 +-
include/linux/proc_fs.h | 28 +++--
include/net/net_namespace.h | 11 ++
net/802/tr.c | 3 +-
net/8021q/vlanproc.c | 5 +-
net/appletalk/atalk_proc.c | 7 +-
net/atm/proc.c | 5 +-
net/ax25/af_ax25.c | 13 +-
net/core/dev.c | 9 +-
net/core/dev_mcast.c | 3 +-
net/core/neighbour.c | 3 +-
net/core/pktgen.c | 9 +-
net/core/sock.c | 3 +-
net/core/wireless.c | 3 +-
net/dccp/probe.c | 7 +-
net/decnet/af_decnet.c | 5 +-
net/decnet/dn_dev.c | 5 +-
net/decnet/dn_neigh.c | 5 +-
net/decnet/dn_route.c | 5 +-
net/ieee80211/ieee80211_module.c | 6 +-
net/ipv4/arp.c | 3 +-
net/ipv4/fib_hash.c | 5 +-
net/ipv4/fib_trie.c | 17 ++--
net/ipv4/igmp.c | 5 +-
net/ipv4/ipconfig.c | 3 +-
net/ipv4/ipmr.c | 5 +-
net/ipv4/ipvs/ip_vs_app.c | 5 +-
net/ipv4/ipvs/ip_vs_conn.c | 5 +-
net/ipv4/ipvs/ip_vs_ctl.c | 9 +-

```
 net/ipv4/ipvs/ip_vs_lblcr.c                    |    4 +-
 net/ipv4/netfilter/ip_conntrack_standalone.c   |   16 ++--
 net/ipv4/netfilter/ip_queue.c                  |    7 +-
 net/ipv4/netfilter/ipt_CLUSTERIP.c             |    3 +-
 net/ipv4/netfilter/ipt_recent.c                |    5 +-
 .../netfilter/nf_conntrack_l3proto_ipv4_compat.c   |   17 ++--
 net/ipv4/proc.c                                |   11 +-
 net/ipv4/raw.c                                 |    5 +-
 net/ipv4/route.c                               |    7 +-
 net/ipv4/tcp_ipv4.c                            |    5 +-
 net/ipv4/tcp_probe.c                           |    6 +-
 net/ipv4/udp.c                                 |    5 +-
 net/ipv6/addrconf.c                            |    7 +-
 net/ipv6/anycast.c                             |    5 +-
 net/ipv6/ip6_flowlabel.c                       |    5 +-
 net/ipv6/mcast.c                               |    9 +-
 net/ipv6/netfilter/ip6_queue.c                 |    7 +-
 net/ipv6/proc.c                                |   17 ++--
 net/ipv6/raw.c                                 |    5 +-
 net/ipv6/route.c                               |    9 +-
 net/ipx/ipx_proc.c                             |    7 +-
 net/irda/irproc.c                              |    5 +-
 net/key/af_key.c                               |    5 +-
 net/llc/llc_proc.c                             |    7 +-
 net/netfilter/core.c                           |    3 +-
 net/netfilter/nf_conntrack_standalone.c        |   19 ++--
 net/netfilter/x_tables.c                       |   17 ++--
 net/netfilter/xt_hashlimit.c                   |   11 +-
 net/netlink/af_netlink.c                       |    3 +-
 net/netrom/af_netrom.c                         |   13 +-
 net/packet/af_packet.c                         |    5 +-
 net/rose/af_rose.c                             |   17 ++--
 net/rxrpc/proc.c                               |    7 +-
 net/sched/sch_api.c                            |    3 +-
 net/sctp/protocol.c                            |    5 +-
 net/sunrpc/stats.c                             |    5 +-
 net/unix/af_unix.c                             |    5 +-
 net/wanrouter/wanproc.c                        |    7 +-
 net/x25/x25_proc.c                             |    7 +-
 85 files changed, 462 insertions(+), 250 deletions(-)

diff --git a/drivers/isdn/divert/divert_procfs.c b/drivers/isdn/divert/divert_procfs.c
index 06967da..6517dd5 100644
--- a/drivers/isdn/divert/divert_procfs.c
+++ b/drivers/isdn/divert/divert_procfs.c
@@ -18,6 +18,7 @@
 #include <linux/fs.h>
 #endif
```

```
 #include <linux/isdnif.h>
+#include <net/net_namespace.h>
 #include "isdn_divert.h"


@@ -285,12 +286,12 @@ divert_dev_init(void)
  init_waitqueue_head(&rd_queue);

 #ifdef CONFIG_PROC_FS
- isdn_proc_entry = proc_mkdir("net/isdn", NULL);
+ isdn_proc_entry = proc_mkdir("isdn", per_net(proc_net, init_net()));
  if (!isdn_proc_entry)
   return (-1);
  isdn_divert_entry = create_proc_entry("divert", S_IFREG | S_IRUGO, isdn_proc_entry);
  if (!isdn_divert_entry) {
-  remove_proc_entry("net/isdn", NULL);
+  remove_proc_entry("isdn", per_net(proc_net, init_net()));
   return (-1);
  }
  isdn_divert_entry->proc_fops = &isdn_fops;
@@ -310,7 +311,7 @@ divert_dev_deinit(void)

 #ifdef CONFIG_PROC_FS
  remove_proc_entry("divert", isdn_proc_entry);
- remove_proc_entry("net/isdn", NULL);
+ remove_proc_entry("isdn", per_net(proc_net, init_net()));
 #endif /* CONFIG_PROC_FS */

  return (0);
diff --git a/drivers/isdn/hardware/eicon/diva_didd.c b/drivers/isdn/hardware/eicon/diva_didd.c
index 14298b8..1b7c0f9 100644
--- a/drivers/isdn/hardware/eicon/diva_didd.c
+++ b/drivers/isdn/hardware/eicon/diva_didd.c
@@ -15,6 +15,7 @@
 #include <linux/init.h>
 #include <linux/kernel.h>
 #include <linux/proc_fs.h>
+#include <net/net_namespace.h>

 #include "platform.h"
 #include "di_defs.h"
@@ -86,7 +87,7 @@ proc_read(char *page, char **start, off_t off, int count, int *eof,

 static int DIVA_INIT_FUNCTION create_proc(void)
 {
- proc_net_eicon = proc_mkdir("net/eicon", NULL);
+ proc_net_eicon = proc_mkdir("eicon", per_net(proc_net, init_net()));
```

```
  if (proc_net_eicon) {
   if ((proc_didd =
@@ -102,7 +103,7 @@ static int DIVA_INIT_FUNCTION create_proc(void)
 static void DIVA_EXIT_FUNCTION remove_proc(void)
 {
  remove_proc_entry(DRIVERLNAME, proc_net_eicon);
- remove_proc_entry("net/eicon", NULL);
+ remove_proc_entry("eicon", per_net(proc_net, init_net()));
 }

 static int DIVA_INIT_FUNCTION divadidd_init(void)
diff --git a/drivers/isdn/hysdn/hysdn_procconf.c b/drivers/isdn/hysdn/hysdn_procconf.c
index 94a9350..b634e67 100644
--- a/drivers/isdn/hysdn/hysdn_procconf.c
+++ b/drivers/isdn/hysdn/hysdn_procconf.c
@@ -392,7 +392,7 @@ hysdn_procconf_init(void)
  hysdn_card *card;
  unsigned char conf_name[20];

- hysdn_proc_entry = proc_mkdir(PROC_SUBDIR_NAME, proc_net);
+ hysdn_proc_entry = proc_mkdir(PROC_SUBDIR_NAME, per_net(proc_net, init_net()));
  if (!hysdn_proc_entry) {
   printk(KERN_ERR "HYSDN: unable to create hysdn subdir\n");
   return (-1);
@@ -437,5 +437,5 @@ hysdn_procconf_release(void)
   card = card->next; /* point to next card */
  }

- remove_proc_entry(PROC_SUBDIR_NAME, proc_net);
+ remove_proc_entry(PROC_SUBDIR_NAME, per_net(proc_net, init_net()));
 }
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 6482aed..9b3bf4e 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -75,6 +75,7 @@
 #include <linux/if_vlan.h>
 #include <linux/if_bonding.h>
 #include <net/route.h>
+#include <net/net_namespace.h>
 #include "bonding.h"
 #include "bond_3ad.h"
 #include "bond_alb.h"
@@ -3169,7 +3170,7 @@ static void bond_create_proc_dir(void)
 {
  int len = strlen(DRV_NAME);

- for (bond_proc_dir = proc_net->subdir; bond_proc_dir;
```

```
+ for (bond_proc_dir = per_net(proc_net, init_net())->subdir; bond_proc_dir;
       bond_proc_dir = bond_proc_dir->next) {
   if ((bond_proc_dir->namelen == len) &&
       !memcmp(bond_proc_dir->name, DRV_NAME, len)) {
@@ -3178,7 +3179,7 @@ static void bond_create_proc_dir(void)
  }

  if (!bond_proc_dir) {
-  bond_proc_dir = proc_mkdir(DRV_NAME, proc_net);
+  bond_proc_dir = proc_mkdir(DRV_NAME, per_net(proc_net, init_net()));
   if (bond_proc_dir) {
    bond_proc_dir->owner = THIS_MODULE;
   } else {
@@ -3213,7 +3214,7 @@ static void bond_destroy_proc_dir(void)
    bond_proc_dir->owner = NULL;
   }
  } else {
-  remove_proc_entry(DRV_NAME, proc_net);
+  remove_proc_entry(DRV_NAME, per_net(proc_net, init_net()));
   bond_proc_dir = NULL;
  }
 }
diff --git a/drivers/net/hamradio/bpqether.c b/drivers/net/hamradio/bpqether.c
index 5b788d8..9fc92ad 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -83,6 +83,7 @@

 #include <net/ip.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>

 #include <linux/bpqether.h>

@@ -594,7 +595,7 @@ static int bpq_device_event(struct notifier_block *this,unsigned long
event, voi
 static int __init bpq_init_driver(void)
 {
 #ifdef CONFIG_PROC_FS
- if (!proc_net_fops_create("bpqether", S_IRUGO, &bpq_info_fops)) {
+ if (!proc_net_fops_create(init_net(), "bpqether", S_IRUGO, &bpq_info_fops)) {
   printk(KERN_ERR
    "bpq: cannot create /proc/net/bpqether entry.\n");
   return -ENOENT;
@@ -618,7 +619,7 @@ static void __exit bpq_cleanup_driver(void)

  unregister_netdevice_notifier(&bpq_dev_notifier);
```

```
- proc_net_remove("bpqether");
+ proc_net_remove(init_net(), "bpqether");

  rtnl_lock();
  while (!list_empty(&bpq_devices)) {
diff --git a/drivers/net/hamradio/scc.c b/drivers/net/hamradio/scc.c
index 2ce047e..2000597 100644
--- a/drivers/net/hamradio/scc.c
+++ b/drivers/net/hamradio/scc.c
@@ -2114,7 +2114,7 @@ static int __init scc_init_driver (void)
  }
  rtnl_unlock();

- proc_net_fops_create("z8530drv", 0, &scc_net_seq_fops);
+ proc_net_fops_create(init_net(), "z8530drv", 0, &scc_net_seq_fops);

  return 0;
 }
@@ -2169,7 +2169,7 @@ static void __exit scc_cleanup_driver(void)
  if (Vector_Latch)
   release_region(Vector_Latch, 1);

- proc_net_remove("z8530drv");
+ proc_net_remove(init_net(), "z8530drv");
 }

 MODULE_AUTHOR("Joerg Reuter <jreuter@yaina.de>");
diff --git a/drivers/net/hamradio/yam.c b/drivers/net/hamradio/yam.c
index 6d74f08..3e92f3b 100644
--- a/drivers/net/hamradio/yam.c
+++ b/drivers/net/hamradio/yam.c
@@ -60,6 +60,7 @@
 #include <linux/etherdevice.h>
 #include <linux/skbuff.h>
 #include <net/ax25.h>
+#include <net/net_namespace.h>

 #include <linux/kernel.h>
 #include <linux/proc_fs.h>
@@ -1147,7 +1148,7 @@ static int __init yam_init_driver(void)
  yam_timer.expires = jiffies + HZ / 100;
  add_timer(&yam_timer);

- proc_net_fops_create("yam", S_IRUGO, &yam_info_fops);
+ proc_net_fops_create(init_net(), "yam", S_IRUGO, &yam_info_fops);
  return 0;
  error:
  while (--i >= 0) {
```

```
@@ -1179,7 +1180,7 @@ static void __exit yam_cleanup_driver(void)
   kfree(p);
  }

- proc_net_remove("yam");
+ proc_net_remove(init_net(), "yam");
 }

 /* --------------------------------------------------------------------- */
diff --git a/drivers/net/ibmveth.c b/drivers/net/ibmveth.c
index 99343b5..d8b0ba8 100644
--- a/drivers/net/ibmveth.c
+++ b/drivers/net/ibmveth.c
@@ -97,7 +97,7 @@ static inline void ibmveth_rxq_harvest_buffer(struct ibmveth_adapter
*adapter);
 static struct kobj_type ktype_veth_pool;

 #ifdef CONFIG_PROC_FS
-#define IBMVETH_PROC_DIR "net/ibmveth"
+#define IBMVETH_PROC_DIR "ibmveth"
 static struct proc_dir_entry *ibmveth_proc_dir;
 #endif

@@ -1073,7 +1073,7 @@ static int __devexit ibmveth_remove(struct vio_dev *dev)
 #ifdef CONFIG_PROC_FS
 static void ibmveth_proc_register_driver(void)
 {
- ibmveth_proc_dir = proc_mkdir(IBMVETH_PROC_DIR, NULL);
+ ibmveth_proc_dir = proc_mkdir(IBMVETH_PROC_DIR, per_net(proc_net, init_net()));
  if (ibmveth_proc_dir) {
   SET_MODULE_OWNER(ibmveth_proc_dir);
  }
@@ -1081,7 +1081,7 @@ static void ibmveth_proc_register_driver(void)

 static void ibmveth_proc_unregister_driver(void)
 {
- remove_proc_entry(IBMVETH_PROC_DIR, NULL);
+ remove_proc_entry(IBMVETH_PROC_DIR, per_net(proc_net, init_net()));
 }

 static void *ibmveth_seq_start(struct seq_file *seq, loff_t *pos)
diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index 315d5c3..d34fe16 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -72,6 +72,7 @@
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>
```

```
+#include <net/net_namespace.h>
 #include <net/sock.h>

 #include <asm/uaccess.h>
@@ -1055,7 +1056,7 @@ static int __init pppoe_proc_init(void)
 {
  struct proc_dir_entry *p;

- p = create_proc_entry("net/pppoe", S_IRUGO, NULL);
+ p = create_proc_entry("pppoe", S_IRUGO, per_net(proc_net, init_net()));
  if (!p)
   return -ENOMEM;

@@ -1126,7 +1127,7 @@ static void __exit pppoe_exit(void)
  dev_remove_pack(&pppoes_ptype);
  dev_remove_pack(&pppoed_ptype);
  unregister_netdevice_notifier(&pppoe_notifier);
- remove_proc_entry("net/pppoe", NULL);
+ remove_proc_entry("pppoe", per_net(proc_net, init_net()));
  proto_unregister(&pppoe_sk_proto);
 }

diff --git a/drivers/net/tc35815.c b/drivers/net/tc35815.c
index 81ed82f..1f26c29 100644
--- a/drivers/net/tc35815.c
+++ b/drivers/net/tc35815.c
@@ -61,7 +61,6 @@ static const char *version =
  * io regions, irqs and dma channels
  */
 static const char* cardname = "TC35815CF";
-#define TC35815_PROC_ENTRY "net/tc35815"

 #define TC35815_MODULE_NAME "TC35815CF"
 #define TX_TIMEOUT (4*HZ)
diff --git a/drivers/net/tokenring/lanstreamer.c b/drivers/net/tokenring/lanstreamer.c
index e999feb..b382ef3 100644
--- a/drivers/net/tokenring/lanstreamer.c
+++ b/drivers/net/tokenring/lanstreamer.c
@@ -250,7 +250,7 @@ static int __devinit streamer_init_one(struct pci_dev *pdev,
 #if STREAMER_NETWORK_MONITOR
 #ifdef CONFIG_PROC_FS
  if (!dev_streamer)
-  create_proc_read_entry("net/streamer_tr", 0, 0,
+  create_proc_read_entry("streamer_tr", 0, per_net(proc_net, init_net()),
       streamer_proc_info, NULL);
  streamer_priv->next = dev_streamer;
  dev_streamer = streamer_priv;
```

```
@@ -423,7 +423,7 @@ static void __devexit streamer_remove_one(struct pci_dev *pdev)
    }
   }
   if (!dev_streamer)
-   remove_proc_entry("net/streamer_tr", NULL);
+   remove_proc_entry("streamer_tr", per_net(proc_net, init_net()));
  }
 #endif
 #endif
diff --git a/drivers/net/tokenring/olympic.c b/drivers/net/tokenring/olympic.c
index 8f4ecc1..6b74c3b 100644
--- a/drivers/net/tokenring/olympic.c
+++ b/drivers/net/tokenring/olympic.c
@@ -101,6 +101,7 @@
 #include <linux/bitops.h>
 #include <linux/jiffies.h>

+#include <net/net_namespace.h>
 #include <net/checksum.h>

 #include <asm/io.h>
@@ -268,9 +269,9 @@ static int __devinit olympic_probe(struct pci_dev *pdev, const struct
pci_device
  printk("Olympic: %s registered as: %s\n",olympic_priv->olympic_card_name,dev->name);
  if (olympic_priv->olympic_network_monitor) { /* Must go after register_netdev as we need the
device name */
   char proc_name[20] ;
-  strcpy(proc_name,"net/olympic_") ;
+  strcpy(proc_name,"olympic_") ;
   strcat(proc_name,dev->name) ;
-  create_proc_read_entry(proc_name,0,NULL,olympic_proc_info,(void *)dev) ;
+  create_proc_read_entry(proc_name,0,per_net(proc_net, init_net()),olympic_proc_info,(void
*)dev) ;
   printk("Olympic: Network Monitor information: /proc/%s\n",proc_name);
  }
  return  0 ;
@@ -1750,9 +1751,9 @@ static void __devexit olympic_remove_one(struct pci_dev *pdev)

  if (olympic_priv->olympic_network_monitor) {
   char proc_name[20] ;
-  strcpy(proc_name,"net/olympic_") ;
+  strcpy(proc_name,"olympic_") ;
   strcat(proc_name,dev->name) ;
-  remove_proc_entry(proc_name,NULL);
+  remove_proc_entry(proc_name,per_net(proc_net, init_net()));
  }
  unregister_netdev(dev) ;
  iounmap(olympic_priv->olympic_mmio) ;
```

```
diff --git a/drivers/net/wireless/hostap/hostap_main.c b/drivers/net/wireless/hostap/hostap_main.c
index 04c19ce..69b56d6 100644
--- a/drivers/net/wireless/hostap/hostap_main.c
+++ b/drivers/net/wireless/hostap/hostap_main.c
@@ -24,6 +24,7 @@
 #include <linux/rtnetlink.h>
 #include <linux/wireless.h>
 #include <linux/etherdevice.h>
+#include <net/net_namespace.h>
 #include <net/iw_handler.h>
 #include <net/ieee80211.h>
 #include <net/ieee80211_crypt.h>
@@ -1093,8 +1094,8 @@ struct proc_dir_entry *hostap_proc;

 static int __init hostap_init(void)
 {
- if (proc_net != NULL) {
-  hostap_proc = proc_mkdir("hostap", proc_net);
+ if (per_net(proc_net, init_net()) != NULL) {
+  hostap_proc = proc_mkdir("hostap", per_net(proc_net, init_net()));
   if (!hostap_proc)
    printk(KERN_WARNING "Failed to mkdir "
           "/proc/net/hostap\n");
@@ -1109,7 +1110,7 @@ static void __exit hostap_exit(void)
 {
  if (hostap_proc != NULL) {
   hostap_proc = NULL;
-  remove_proc_entry("hostap", proc_net);
+  remove_proc_entry("hostap", per_net(proc_net, init_net()));
  }
 }

diff --git a/drivers/net/wireless/strip.c b/drivers/net/wireless/strip.c
index ce3a8ba..6c27ff2 100644
--- a/drivers/net/wireless/strip.c
+++ b/drivers/net/wireless/strip.c
@@ -107,6 +107,7 @@ static const char StripVersion[] = "1.3A-STUART.CHESHIRE";
 #include <linux/serialP.h>
 #include <linux/rcupdate.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>

 #include <linux/ip.h>
 #include <linux/tcp.h>
@@ -2789,7 +2790,7 @@ static int __init strip_init_driver(void)
  /*
   * Register the status file with /proc
   */
```

```
- proc_net_fops_create("strip", S_IFREG | S_IRUGO, &strip_seq_fops);
+ proc_net_fops_create(init_net(), "strip", S_IFREG | S_IRUGO, &strip_seq_fops);

  return status;
 }
@@ -2811,7 +2812,7 @@ static void __exit strip_exit_driver(void)
  }

  /* Unregister with the /proc/net file here. */
- proc_net_remove("strip");
+ proc_net_remove(init_net(), "strip");

  if ((i = tty_unregister_ldisc(N_STRIP)))
   printk(KERN_ERR "STRIP: can't unregister line discipline (err = %d)\n", i);
diff --git a/fs/proc/Makefile b/fs/proc/Makefile
index a6b3a8f..63cc3ce 100644
--- a/fs/proc/Makefile
+++ b/fs/proc/Makefile
@@ -10,6 +10,7 @@ proc-$(CONFIG_MMU) := mmu.o task_mmu.o
 proc-y       += inode.o root.o base.o generic.o array.o \
   proc_tty.o proc_misc.o proc_sysctl.o

+proc-$(CONFIG_NET)  += proc_net.o
 proc-$(CONFIG_PROC_KCORE) += kcore.o
 proc-$(CONFIG_PROC_VMCORE) += vmcore.o
 proc-$(CONFIG_PROC_DEVICETREE) += proc_devtree.o
diff --git a/fs/proc/internal.h b/fs/proc/internal.h
index 3c9a305..f916252 100644
--- a/fs/proc/internal.h
+++ b/fs/proc/internal.h
@@ -12,6 +12,11 @@
 #include <linux/proc_fs.h>

 extern int proc_sys_init(void);
+#ifdef CONFIG_NET
+extern int proc_net_init(void);
+#else
+static inline int proc_net_init(void) { return 0; }
+#endif

 struct vmalloc_info {
  unsigned long used;
diff --git a/fs/proc/proc_net.c b/fs/proc/proc_net.c
new file mode 100644
index 0000000..022dd9a
--- /dev/null
+++ b/fs/proc/proc_net.c
@@ -0,0 +1,126 @@
```

```
+/*
+ *  linux/fs/proc/net.c
+ *
+ *  Copyright (C) 2007
+ *
+ *  Author: Eric Biederman <ebiederm@xmission.com>
+ *
+ *  proc net directory handling functions
+ */
+
+#include <asm/uaccess.h>
+
+#include <linux/errno.h>
+#include <linux/time.h>
+#include <linux/proc_fs.h>
+#include <linux/stat.h>
+#include <linux/init.h>
+#include <linux/sched.h>
+#include <linux/module.h>
+#include <linux/bitops.h>
+#include <linux/smp_lock.h>
+#include <linux/mount.h>
+#include <linux/nsproxy.h>
+#include <net/net_namespace.h>
+
+#include "internal.h"
+
+static struct proc_dir_entry *proc_net_shadow;
+DEFINE_PER_NET(struct proc_dir_entry *, proc_net);
+DEFINE_PER_NET(struct proc_dir_entry *, proc_net_stat);
+EXPORT_PER_NET_SYMBOL(proc_net);
+EXPORT_PER_NET_SYMBOL(proc_net_stat);
+
+static DEFINE_PER_NET(struct proc_dir_entry, proc_net_root);
+
+static struct dentry *proc_net_shadow_dentry(struct dentry *parent,
+      struct proc_dir_entry *de)
+{
+ struct dentry *shadow = NULL;
+ struct inode *inode;
+ if (!de)
+  goto out;
+ inode = proc_get_inode(parent->d_inode->i_sb, de->low_ino, de);
+ if (!inode)
+  goto out;
+ shadow = d_alloc_name(parent, de->name);
+ if (!shadow)
+  goto out_iput;
```

```
+ shadow->d_op = parent->d_op; /* proc_dentry_operations */
+ d_instantiate(shadow, inode);
+out:
+ return shadow;
+out_iput:
+ iput(inode);
+ goto out;
+}
+
+static void *proc_net_follow_link(struct dentry *parent, struct nameidata *nd)
+{
+ net_t net = current->nsproxy->net_ns;
+ struct dentry *shadow;
+ shadow = proc_net_shadow_dentry(parent, per_net(proc_net, net));
+ if (!shadow)
+  return ERR_PTR(-ENOENT);
+
+ dput(nd->dentry);
+ /* My dentry count is 1 and that should be enough as the
+  * shadow dentry is thrown away immediately.
+  */
+ nd->dentry = shadow;
+ return NULL;
+}
+
+static const struct file_operations proc_net_dir_operations = {
+ .read   = generic_read_dir,
+};
+
+static struct inode_operations proc_net_dir_inode_operations = {
+ .follow_link = proc_net_follow_link,
+};
+
+
+static int proc_net_ns_init(net_t net)
+{
+ struct proc_dir_entry *netd, *net_statd;
+
+ netd = proc_mkdir("net", &per_net(proc_net_root, net));
+ if (!netd)
+  return -EEXIST;
+
+ net_statd = proc_mkdir("stat", netd);
+ if (!net_statd) {
+  remove_proc_entry("net", &per_net(proc_net_root, net));
+  return -EEXIST;
+ }
+
```

```
+ netd->data = net_to_voidp(net);
+ net_statd->data = net_to_voidp(net);
+ per_net(proc_net_root, net).data = net_to_voidp(net);
+
+ per_net(proc_net, net) = netd;
+ per_net(proc_net_stat, net) = net_statd;
+
+ return 0;
+}
+
+static void proc_net_ns_exit(net_t net)
+{
+ remove_proc_entry("stat", per_net(proc_net, net));
+ remove_proc_entry("net", &per_net(proc_net_root, net));
+
+}
+
+struct pernet_operations proc_net_ns_ops = {
+ .init = proc_net_ns_init,
+ .exit = proc_net_ns_exit,
+};
+
+int proc_net_init(void)
+{
+ proc_net_shadow = proc_mkdir("net", NULL);
+ proc_net_shadow->proc_iops = &proc_net_dir_inode_operations;
+ proc_net_shadow->proc_fops = &proc_net_dir_operations;
+
+ return register_pernet_subsys(&proc_net_ns_ops);
+}
diff --git a/fs/proc/root.c b/fs/proc/root.c
index 4d42406..7c3939c 100644
--- a/fs/proc/root.c
+++ b/fs/proc/root.c
@@ -21,7 +21,7 @@

 #include "internal.h"

-struct proc_dir_entry *proc_net, *proc_net_stat, *proc_bus, *proc_root_fs, *proc_root_driver;
+struct proc_dir_entry *proc_bus, *proc_root_fs, *proc_root_driver;

 static int proc_get_sb(struct file_system_type *fs_type,
  int flags, const char *dev_name, void *data, struct vfsmount *mnt)
@@ -61,8 +61,8 @@ void __init proc_root_init(void)
   return;
  }
  proc_misc_init();
- proc_net = proc_mkdir("net", NULL);
```

```
- proc_net_stat = proc_mkdir("net/stat", NULL);
+
+ proc_net_init();

 #ifdef CONFIG_SYSVIPC
  proc_mkdir("sysvipc", NULL);
@@ -161,7 +161,5 @@ EXPORT_SYMBOL(create_proc_entry);
 EXPORT_SYMBOL(remove_proc_entry);
 EXPORT_SYMBOL(proc_root);
 EXPORT_SYMBOL(proc_root_fs);
-EXPORT_SYMBOL(proc_net);
-EXPORT_SYMBOL(proc_net_stat);
 EXPORT_SYMBOL(proc_bus);
 EXPORT_SYMBOL(proc_root_driver);
diff --git a/include/linux/proc_fs.h b/include/linux/proc_fs.h
index 2969913..c1b958d 100644
--- a/include/linux/proc_fs.h
+++ b/include/linux/proc_fs.h
@@ -5,6 +5,7 @@
 #include <linux/fs.h>
 #include <linux/spinlock.h>
 #include <linux/magic.h>
+#include <linux/net_namespace_type.h>
 #include <asm/atomic.h>

 /*
@@ -85,8 +86,8 @@ struct vmcore {

 extern struct proc_dir_entry proc_root;
 extern struct proc_dir_entry *proc_root_fs;
-extern struct proc_dir_entry *proc_net;
-extern struct proc_dir_entry *proc_net_stat;
+DECLARE_PER_NET(struct proc_dir_entry *, proc_net);
+DECLARE_PER_NET(struct proc_dir_entry *, proc_net_stat);
 extern struct proc_dir_entry *proc_bus;
 extern struct proc_dir_entry *proc_root_driver;
 extern struct proc_dir_entry *proc_root_kcore;
@@ -183,24 +184,25 @@ static inline struct proc_dir_entry *create_proc_info_entry(const char
*name,
  return res;
 }

-static inline struct proc_dir_entry *proc_net_create(const char *name,
- mode_t mode, get_info_t *get_info)
+static inline struct proc_dir_entry *proc_net_create(net_t net,
+ const char *name, mode_t mode, get_info_t *get_info)
 {
- return create_proc_info_entry(name,mode,proc_net,get_info);
```

```
+ return create_proc_info_entry(name,mode,per_net(proc_net, net),get_info);
 }

-static inline struct proc_dir_entry *proc_net_fops_create(const char *name,
- mode_t mode, const struct file_operations *fops)
+static inline struct proc_dir_entry *proc_net_fops_create(net_t net,
+ const char *name, mode_t mode, const struct file_operations *fops)
 {
- struct proc_dir_entry *res = create_proc_entry(name, mode, proc_net);
+ struct proc_dir_entry *res =
+  create_proc_entry(name, mode, per_net(proc_net, net));
  if (res)
   res->proc_fops = fops;
  return res;
 }

-static inline void proc_net_remove(const char *name)
+static inline void proc_net_remove(net_t net, const char *name)
 {
- remove_proc_entry(name,proc_net);
+ remove_proc_entry(name, per_net(proc_net, net));
 }

 #else
@@ -209,9 +211,9 @@ static inline void proc_net_remove(const char *name)
 #define proc_net NULL
 #define proc_bus NULL

-#define proc_net_fops_create(name, mode, fops)  ({ (void)(mode), NULL; })
-#define proc_net_create(name, mode, info) ({ (void)(mode), NULL; })
-static inline void proc_net_remove(const char *name) {}
+#define proc_net_fops_create(net, name, mode, fops)  ({ (void)(mode), NULL; })
+#define proc_net_create(net, name, mode, info) ({ (void)(mode), NULL; })
+static inline void proc_net_remove(net_t net, const char *name) {}

 static inline void proc_flush_task(struct task_struct *task) { }

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index 9208e2e..b64568f 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -8,6 +8,7 @@
 #include <linux/workqueue.h>
 #include <linux/nsproxy.h>
 #include <linux/net_namespace_type.h>
+#include <linux/proc_fs.h>

 /* How many bytes in each network namespace should we allocate
```

```
  * for use by modules when they are loaded.
@@ -65,4 +66,14 @@ extern void unregister_pernet_subsys(struct pernet_operations *);
 extern int register_pernet_device(struct pernet_operations *);
 extern void unregister_pernet_device(struct pernet_operations *);

+static inline net_t PDE_NET(struct proc_dir_entry *pde)
+{
+ return net_from_voidp(pde->parent->data);
+}
+
+static inline net_t PROC_NET(const struct inode *inode)
+{
+ return PDE_NET(PDE(inode));
+}
+
 #endif /* __NET_NET_NAMESPACE_H */
diff --git a/net/802/tr.c b/net/802/tr.c
index 829deb4..3324fa6 100644
--- a/net/802/tr.c
+++ b/net/802/tr.c
@@ -36,6 +36,7 @@
 #include <linux/seq_file.h>
 #include <linux/init.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>

 static void tr_add_rif_info(struct trh_hdr *trh, struct net_device *dev);
 static void rif_check_expire(unsigned long dummy);
@@ -636,7 +637,7 @@ static int __init rif_init(void)
  rif_timer.function = rif_check_expire;
  add_timer(&rif_timer);

- proc_net_fops_create("tr_rif", S_IRUGO, &rif_seq_fops);
+ proc_net_fops_create(init_net(), "tr_rif", S_IRUGO, &rif_seq_fops);
  return 0;
 }

diff --git a/net/8021q/vlanproc.c b/net/8021q/vlanproc.c
index a8fc0de..abcf58c 100644
--- a/net/8021q/vlanproc.c
+++ b/net/8021q/vlanproc.c
@@ -33,6 +33,7 @@
 #include <linux/fs.h>
 #include <linux/netdevice.h>
 #include <linux/if_vlan.h>
+#include <net/net_namespace.h>
 #include "vlanproc.h"
 #include "vlan.h"
```

```
@@ -143,7 +144,7 @@ void vlan_proc_cleanup(void)
   remove_proc_entry(name_conf, proc_vlan_dir);

  if (proc_vlan_dir)
-  proc_net_remove(name_root);
+  proc_net_remove(init_net(), name_root);

  /* Dynamically added entries should be cleaned up as their vlan_device
   * is removed, so we should not have to take care of it here...
@@ -156,7 +157,7 @@ void vlan_proc_cleanup(void)

 int __init vlan_proc_init(void)
 {
- proc_vlan_dir = proc_mkdir(name_root, proc_net);
+ proc_vlan_dir = proc_mkdir(name_root, per_net(proc_net, init_net()));
  if (proc_vlan_dir) {
   proc_vlan_conf = create_proc_entry(name_conf,
         S_IFREG|S_IRUSR|S_IWUSR,
diff --git a/net/appletalk/atalk_proc.c b/net/appletalk/atalk_proc.c
index 7ae4916..0e77c68 100644
--- a/net/appletalk/atalk_proc.c
+++ b/net/appletalk/atalk_proc.c
@@ -13,6 +13,7 @@
 #include <linux/seq_file.h>
 #include <net/sock.h>
 #include <linux/atalk.h>
+#include <net/net_namespace.h>


 static __inline__ struct atalk_iface *atalk_get_interface_idx(loff_t pos)
@@ -271,7 +272,7 @@ int __init atalk_proc_init(void)
  struct proc_dir_entry *p;
  int rc = -ENOMEM;

- atalk_proc_dir = proc_mkdir("atalk", proc_net);
+ atalk_proc_dir = proc_mkdir("atalk", per_net(proc_net, init_net()));
  if (!atalk_proc_dir)
   goto out;
  atalk_proc_dir->owner = THIS_MODULE;
@@ -306,7 +307,7 @@ out_socket:
 out_route:
  remove_proc_entry("interface", atalk_proc_dir);
 out_interface:
- remove_proc_entry("atalk", proc_net);
+ remove_proc_entry("atalk", per_net(proc_net, init_net()));
  goto out;
 }
```

```
@@ -316,5 +317,5 @@ void __exit atalk_proc_exit(void)
  remove_proc_entry("route", atalk_proc_dir);
  remove_proc_entry("socket", atalk_proc_dir);
  remove_proc_entry("arp", atalk_proc_dir);
- remove_proc_entry("atalk", proc_net);
+ remove_proc_entry("atalk", per_net(proc_net, init_net()));
 }
diff --git a/net/atm/proc.c b/net/atm/proc.c
index 739866b..8b0299d 100644
--- a/net/atm/proc.c
+++ b/net/atm/proc.c
@@ -22,6 +22,7 @@
 #include <linux/netdevice.h>
 #include <linux/atmclip.h>
 #include <linux/init.h> /* for __init */
+#include <net/net_namespace.h>
 #include <net/atmclip.h>
 #include <asm/uaccess.h>
 #include <asm/atomic.h>
@@ -475,7 +476,7 @@ static void atm_proc_dirs_remove(void)
   if (e->dirent)
    remove_proc_entry(e->name, atm_proc_root);
  }
- remove_proc_entry("net/atm", NULL);
+ remove_proc_entry("atm", per_net(proc_net, init_net()));
 }

 int __init atm_proc_init(void)
@@ -483,7 +484,7 @@ int __init atm_proc_init(void)
  static struct atm_proc_entry *e;
  int ret;

- atm_proc_root = proc_mkdir("net/atm",NULL);
+ atm_proc_root = proc_mkdir("atm", per_net(proc_net, init_net()));
  if (!atm_proc_root)
   goto err_out;
  for (e = atm_proc_ents; e->name; e++) {
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index 42233df..e60af4e 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -48,6 +48,7 @@
 #include <net/tcp_states.h>
 #include <net/ip.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>
```

```
@@ -2000,9 +2001,9 @@ static int __init ax25_init(void)
  register_netdevice_notifier(&ax25_dev_notifier);
  ax25_register_sysctl();

- proc_net_fops_create("ax25_route", S_IRUGO, &ax25_route_fops);
- proc_net_fops_create("ax25", S_IRUGO, &ax25_info_fops);
- proc_net_fops_create("ax25_calls", S_IRUGO, &ax25_uid_fops);
+ proc_net_fops_create(init_net(), "ax25_route", S_IRUGO, &ax25_route_fops);
+ proc_net_fops_create(init_net(), "ax25", S_IRUGO, &ax25_info_fops);
+ proc_net_fops_create(init_net(), "ax25_calls", S_IRUGO, &ax25_uid_fops);
 out:
  return rc;
 }
@@ -2016,9 +2017,9 @@ MODULE_ALIAS_NETPROTO(PF_AX25);

 static void __exit ax25_exit(void)
 {
- proc_net_remove("ax25_route");
- proc_net_remove("ax25");
- proc_net_remove("ax25_calls");
+ proc_net_remove(init_net(), "ax25_route");
+ proc_net_remove(init_net(), "ax25");
+ proc_net_remove(init_net(), "ax25_calls");
  ax25_rt_free();
  ax25_uid_free();
  ax25_dev_free();
diff --git a/net/core/dev.c b/net/core/dev.c
index 17c07f3..90e4c0e 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -116,6 +116,7 @@
 #include <linux/dmaengine.h>
 #include <linux/err.h>
 #include <linux/ctype.h>
+#include <net/net_namespace.h>

 /*
  * The list of packet types we will receive (as opposed to discard)
@@ -2238,9 +2239,9 @@ static int __init dev_proc_init(void)
 {
  int rc = -ENOMEM;

- if (!proc_net_fops_create("dev", S_IRUGO, &dev_seq_fops))
+ if (!proc_net_fops_create(init_net(), "dev", S_IRUGO, &dev_seq_fops))
   goto out;
- if (!proc_net_fops_create("softnet_stat", S_IRUGO, &softnet_seq_fops))
```

```
+ if (!proc_net_fops_create(init_net(), "softnet_stat", S_IRUGO, &softnet_seq_fops))
    goto out_dev;
   if (wireless_proc_init())
    goto out_softnet;
@@ -2248,9 +2249,9 @@ static int __init dev_proc_init(void)
  out:
   return rc;
  out_softnet:
- proc_net_remove("softnet_stat");
+ proc_net_remove(init_net(), "softnet_stat");
  out_dev:
- proc_net_remove("dev");
+ proc_net_remove(init_net(), "dev");
   goto out;
  }
  #else
diff --git a/net/core/dev_mcast.c b/net/core/dev_mcast.c
index b22648d..623e606 100644
--- a/net/core/dev_mcast.c
+++ b/net/core/dev_mcast.c
@@ -47,6 +47,7 @@
  #include <linux/skbuff.h>
  #include <net/sock.h>
  #include <net/arp.h>
+#include <net/net_namespace.h>


  /*
@@ -289,7 +290,7 @@ static struct file_operations dev_mc_seq_fops = {

  void __init dev_mcast_init(void)
  {
- proc_net_fops_create("dev_mcast", 0, &dev_mc_seq_fops);
+ proc_net_fops_create(init_net(), "dev_mcast", 0, &dev_mc_seq_fops);
  }

  EXPORT_SYMBOL(dev_mc_add);
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 8437678..90e1d2e 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -34,6 +34,7 @@
  #include <linux/rtnetlink.h>
  #include <linux/random.h>
  #include <linux/string.h>
+#include <net/net_namespace.h>

  #define NEIGH_DEBUG 1
```

```
@@ -1348,7 +1349,7 @@ void neigh_table_init_no_netlink(struct neigh_table *tbl)
   panic("cannot create neighbour cache statistics");

 #ifdef CONFIG_PROC_FS
- tbl->pde = create_proc_entry(tbl->id, 0, proc_net_stat);
+ tbl->pde = create_proc_entry(tbl->id, 0, per_net(proc_net_stat, init_net()));
  if (!tbl->pde)
   panic("cannot create neighbour proc dir entry");
  tbl->pde->proc_fops = &neigh_stat_seq_fops;
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 04d4b93..ab48533 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -152,6 +152,7 @@
 #include <net/checksum.h>
 #include <net/ipv6.h>
 #include <net/addrconf.h>
+#include <net/net_namespace.h>
 #include <asm/byteorder.h>
 #include <linux/rcupdate.h>
 #include <asm/bitops.h>
@@ -3565,7 +3566,7 @@ static int __init pg_init(void)

  printk(version);

- pg_proc_dir = proc_mkdir(PG_PROC_DIR, proc_net);
+ pg_proc_dir = proc_mkdir(PG_PROC_DIR, per_net(proc_net, init_net()));
  if (!pg_proc_dir)
   return -ENODEV;
  pg_proc_dir->owner = THIS_MODULE;
@@ -3574,7 +3575,7 @@ static int __init pg_init(void)
  if (pe == NULL) {
   printk("pktgen: ERROR: cannot create %s procfs entry.\n",
          PGCTRL);
-  proc_net_remove(PG_PROC_DIR);
+  proc_net_remove(init_net(), PG_PROC_DIR);
   return -EINVAL;
  }

@@ -3597,7 +3598,7 @@ static int __init pg_init(void)
   printk("pktgen: ERROR: Initialization failed for all threads\n");
   unregister_netdevice_notifier(&pktgen_notifier_block);
   remove_proc_entry(PGCTRL, pg_proc_dir);
-  proc_net_remove(PG_PROC_DIR);
+  proc_net_remove(init_net(), PG_PROC_DIR);
   return -ENODEV;
  }
```

```
@@ -3624,7 +3625,7 @@ static void __exit pg_cleanup(void)

  /* Clean up proc file system */
  remove_proc_entry(PGCTRL, pg_proc_dir);
- proc_net_remove(PG_PROC_DIR);
+ proc_net_remove(init_net(), PG_PROC_DIR);
 }

 module_init(pg_init);
diff --git a/net/core/sock.c b/net/core/sock.c
index 0ed5b4f..5555364 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -123,6 +123,7 @@
 #include <net/sock.h>
 #include <net/xfrm.h>
 #include <linux/ipsec.h>
+#include <net/net_namespace.h>

 #include <linux/filter.h>

@@ -1922,7 +1923,7 @@ static struct file_operations proto_seq_fops = {
 static int __init proto_init(void)
 {
  /* register /proc/net/protocols */
- return proc_net_fops_create("protocols", S_IRUGO, &proto_seq_fops) == NULL ? -ENOBUFS :
0;
+ return proc_net_fops_create(init_net(), "protocols", S_IRUGO, &proto_seq_fops) == NULL ?
-ENOBUFS : 0;
 }

 subsys_initcall(proto_init);
diff --git a/net/core/wireless.c b/net/core/wireless.c
index f69ab7b..faa242f 100644
--- a/net/core/wireless.c
+++ b/net/core/wireless.c
@@ -94,6 +94,7 @@
 #include <linux/wireless.h>  /* Pretty obvious */
 #include <net/iw_handler.h>  /* New driver API */
 #include <net/netlink.h>
+#include <net/net_namespace.h>

 #include <asm/uaccess.h>  /* copy_to_user() */

@@ -685,7 +686,7 @@ static struct file_operations wireless_seq_fops = {
 int __init wireless_proc_init(void)
 {
```

```
  /* Create /proc/net/wireless entry */
- if (!proc_net_fops_create("wireless", S_IRUGO, &wireless_seq_fops))
+ if (!proc_net_fops_create(init_net(), "wireless", S_IRUGO, &wireless_seq_fops))
    return -ENOMEM;

  return 0;
diff --git a/net/dccp/probe.c b/net/dccp/probe.c
index f81e37d..7c1c1ef 100644
--- a/net/dccp/probe.c
+++ b/net/dccp/probe.c
@@ -30,6 +30,7 @@
 #include <linux/module.h>
 #include <linux/kfifo.h>
 #include <linux/vmalloc.h>
+#include <net/net_namespace.h>

 #include "dccp.h"
 #include "ccid.h"
@@ -165,7 +166,7 @@ static __init int dccpprobe_init(void)
  if (IS_ERR(dccpw.fifo))
    return PTR_ERR(dccpw.fifo);

- if (!proc_net_fops_create(procname, S_IRUSR, &dccpprobe_fops))
+ if (!proc_net_fops_create(init_net(), procname, S_IRUSR, &dccpprobe_fops))
    goto err0;

  ret = register_jprobe(&dccp_send_probe);
@@ -175,7 +176,7 @@ static __init int dccpprobe_init(void)
  pr_info("DCCP watch registered (port=%d)\n", port);
  return 0;
 err1:
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
 err0:
  kfifo_free(dccpw.fifo);
  return ret;
@@ -185,7 +186,7 @@ module_init(dccpprobe_init);
 static __exit void dccpprobe_exit(void)
 {
  kfifo_free(dccpw.fifo);
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
  unregister_jprobe(&dccp_send_probe);

 }
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index 21f20f2..77cd802 100644
--- a/net/decnet/af_decnet.c
```

```
+++ b/net/decnet/af_decnet.c
@@ -131,6 +131,7 @@ Version 0.0.6    2.1.110   07-aug-98   Eduardo Marcelo Serrat
 #include <net/neighbour.h>
 #include <net/dst.h>
 #include <net/fib_rules.h>
+#include <net/net_namespace.h>
 #include <net/dn.h>
 #include <net/dn_nsp.h>
 #include <net/dn_dev.h>
@@ -2396,7 +2397,7 @@ static int __init decnet_init(void)
  dev_add_pack(&dn_dix_packet_type);
  register_netdevice_notifier(&dn_dev_notifier);

- proc_net_fops_create("decnet", S_IRUGO, &dn_socket_seq_fops);
+ proc_net_fops_create(init_net(), "decnet", S_IRUGO, &dn_socket_seq_fops);
  dn_register_sysctl();
 out:
  return rc;
@@ -2424,7 +2425,7 @@ static void __exit decnet_exit(void)
  dn_neigh_cleanup();
  dn_fib_cleanup();

- proc_net_remove("decnet");
+ proc_net_remove(init_net(), "decnet");

  proto_unregister(&dn_proto);
 }
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index 913e25a..19b1469 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -47,6 +47,7 @@
 #include <net/flow.h>
 #include <net/fib_rules.h>
 #include <net/netlink.h>
+#include <net/net_namespace.h>
 #include <net/dn.h>
 #include <net/dn_dev.h>
 #include <net/dn_route.h>
@@ -1483,7 +1484,7 @@ void __init dn_dev_init(void)

  rtnetlink_links[PF_DECnet] = dnet_rtnetlink_table;

- proc_net_fops_create("decnet_dev", S_IRUGO, &dn_dev_seq_fops);
+ proc_net_fops_create(init_net(), "decnet_dev", S_IRUGO, &dn_dev_seq_fops);

 #ifdef CONFIG_SYSCTL
  {
```

```
@@ -1506,7 +1507,7 @@ void __exit dn_dev_cleanup(void)
  }
 #endif /* CONFIG_SYSCTL */

- proc_net_remove("decnet_dev");
+ proc_net_remove(init_net(), "decnet_dev");

  dn_dev_devices_off();
 }
diff --git a/net/decnet/dn_neigh.c b/net/decnet/dn_neigh.c
index 7322bb3..fd99aca 100644
--- a/net/decnet/dn_neigh.c
+++ b/net/decnet/dn_neigh.c
@@ -38,6 +38,7 @@
 #include <linux/rcupdate.h>
 #include <linux/jhash.h>
 #include <asm/atomic.h>
+#include <net/net_namespace.h>
 #include <net/neighbour.h>
 #include <net/dst.h>
 #include <net/flow.h>
@@ -611,11 +612,11 @@ static struct file_operations dn_neigh_seq_fops = {
 void __init dn_neigh_init(void)
 {
  neigh_table_init(&dn_neigh_table);
- proc_net_fops_create("decnet_neigh", S_IRUGO, &dn_neigh_seq_fops);
+ proc_net_fops_create(init_net(), "decnet_neigh", S_IRUGO, &dn_neigh_seq_fops);
 }

 void __exit dn_neigh_cleanup(void)
 {
- proc_net_remove("decnet_neigh");
+ proc_net_remove(init_net(), "decnet_neigh");
  neigh_table_clear(&dn_neigh_table);
 }
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 9881933..0d657eb 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -81,6 +81,7 @@
 #include <net/dst.h>
 #include <net/flow.h>
 #include <net/fib_rules.h>
+#include <net/net_namespace.h>
 #include <net/dn.h>
 #include <net/dn_dev.h>
 #include <net/dn_nsp.h>
@@ -1811,7 +1812,7 @@ void __init dn_route_init(void)
```

```
        dn_dst_ops.gc_thresh = (dn_rt_hash_mask + 1);

- proc_net_fops_create("decnet_cache", S_IRUGO, &dn_rt_cache_seq_fops);
+ proc_net_fops_create(init_net(), "decnet_cache", S_IRUGO, &dn_rt_cache_seq_fops);
 }

 void __exit dn_route_cleanup(void)
@@ -1819,6 +1820,6 @@ void __exit dn_route_cleanup(void)
  del_timer(&dn_route_timer);
  dn_run_flush(0);

- proc_net_remove("decnet_cache");
+ proc_net_remove(init_net(), "decnet_cache");
 }

diff --git a/net/ieee80211/ieee80211_module.c b/net/ieee80211/ieee80211_module.c
index b1c6d1f..23539f6 100644
--- a/net/ieee80211/ieee80211_module.c
+++ b/net/ieee80211/ieee80211_module.c
@@ -263,7 +263,7 @@ static int __init ieee80211_init(void)
  struct proc_dir_entry *e;

  ieee80211_debug_level = debug;
- ieee80211_proc = proc_mkdir(DRV_NAME, proc_net);
+ ieee80211_proc = proc_mkdir(DRV_NAME, per_net(proc_net, init_net()));
  if (ieee80211_proc == NULL) {
   IEEE80211_ERROR("Unable to create " DRV_NAME
     " proc directory\n");
@@ -272,7 +272,7 @@ static int __init ieee80211_init(void)
  e = create_proc_entry("debug_level", S_IFREG | S_IRUGO | S_IWUSR,
         ieee80211_proc);
  if (!e) {
-  remove_proc_entry(DRV_NAME, proc_net);
+  remove_proc_entry(DRV_NAME, per_net(proc_net, init_net()));
   ieee80211_proc = NULL;
   return -EIO;
  }
@@ -292,7 +292,7 @@ static void __exit ieee80211_exit(void)
 #ifdef CONFIG_IEEE80211_DEBUG
  if (ieee80211_proc) {
   remove_proc_entry("debug_level", ieee80211_proc);
-  remove_proc_entry(DRV_NAME, proc_net);
+  remove_proc_entry(DRV_NAME, per_net(proc_net, init_net()));
   ieee80211_proc = NULL;
  }
 #endif    /* CONFIG_IEEE80211_DEBUG */
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
```

```
index 3981e8b..e3b89a7 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -110,6 +110,7 @@
 #include <net/protocol.h>
 #include <net/tcp.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>
 #include <net/arp.h>
 #if defined(CONFIG_AX25) || defined(CONFIG_AX25_MODULE)
 #include <net/ax25.h>
@@ -1400,7 +1401,7 @@ static struct file_operations arp_seq_fops = {

 static int __init arp_proc_init(void)
 {
- if (!proc_net_fops_create("arp", S_IRUGO, &arp_seq_fops))
+ if (!proc_net_fops_create(init_net(), "arp", S_IRUGO, &arp_seq_fops))
   return -ENOMEM;
  return 0;
 }
diff --git a/net/ipv4/fib_hash.c b/net/ipv4/fib_hash.c
index 648f47c..42ea992 100644
--- a/net/ipv4/fib_hash.c
+++ b/net/ipv4/fib_hash.c
@@ -41,6 +41,7 @@
 #include <net/route.h>
 #include <net/tcp.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>
 #include <net/ip_fib.h>

 #include "fib_lookup.h"
@@ -1067,13 +1068,13 @@ static struct file_operations fib_seq_fops = {

 int __init fib_proc_init(void)
 {
- if (!proc_net_fops_create("route", S_IRUGO, &fib_seq_fops))
+ if (!proc_net_fops_create(init_net(), "route", S_IRUGO, &fib_seq_fops))
   return -ENOMEM;
  return 0;
 }

 void __init fib_proc_exit(void)
 {
- proc_net_remove("route");
+ proc_net_remove(init_net(), "route");
 }
 #endif /* CONFIG_PROC_FS */
```

```
diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index 13307c0..94598b3 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -79,6 +79,7 @@
 #include <net/route.h>
 #include <net/tcp.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>
 #include <net/ip_fib.h>
 #include "fib_lookup.h"

@@ -2494,30 +2495,30 @@ static struct file_operations fib_route_fops = {

 int __init fib_proc_init(void)
 {
- if (!proc_net_fops_create("fib_trie", S_IRUGO, &fib_trie_fops))
+ if (!proc_net_fops_create(init_net(), "fib_trie", S_IRUGO, &fib_trie_fops))
   goto out1;

- if (!proc_net_fops_create("fib_triestat", S_IRUGO, &fib_triestat_fops))
+ if (!proc_net_fops_create(init_net(), "fib_triestat", S_IRUGO, &fib_triestat_fops))
   goto out2;

- if (!proc_net_fops_create("route", S_IRUGO, &fib_route_fops))
+ if (!proc_net_fops_create(init_net(), "route", S_IRUGO, &fib_route_fops))
   goto out3;

  return 0;

 out3:
- proc_net_remove("fib_triestat");
+ proc_net_remove(init_net(), "fib_triestat");
 out2:
- proc_net_remove("fib_trie");
+ proc_net_remove(init_net(), "fib_trie");
 out1:
  return -ENOMEM;
 }

 void __init fib_proc_exit(void)
 {
- proc_net_remove("fib_trie");
- proc_net_remove("fib_triestat");
- proc_net_remove("route");
+ proc_net_remove(init_net(), "fib_trie");
+ proc_net_remove(init_net(), "fib_triestat");
+ proc_net_remove(init_net(), "route");
```

```
 }

 #endif /* CONFIG_PROC_FS */
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 0017ccb..92624cc 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -97,6 +97,7 @@
 #include <net/route.h>
 #include <net/sock.h>
 #include <net/checksum.h>
+#include <net/net_namespace.h>
 #include <linux/netfilter_ipv4.h>
 #ifdef CONFIG_IP_MROUTE
 #include <linux/mroute.h>
@@ -2585,8 +2586,8 @@ static struct file_operations igmp_mcf_seq_fops = {

 int __init igmp_mc_proc_init(void)
 {
- proc_net_fops_create("igmp", S_IRUGO, &igmp_mc_seq_fops);
- proc_net_fops_create("mcfilter", S_IRUGO, &igmp_mcf_seq_fops);
+ proc_net_fops_create(init_net(), "igmp", S_IRUGO, &igmp_mc_seq_fops);
+ proc_net_fops_create(init_net(), "mcfilter", S_IRUGO, &igmp_mcf_seq_fops);
  return 0;
 }
 #endif
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index afa60b9..8b649c5 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -59,6 +59,7 @@
 #include <net/ip.h>
 #include <net/ipconfig.h>
 #include <net/route.h>
+#include <net/net_namespace.h>

 #include <asm/uaccess.h>
 #include <net/checksum.h>
@@ -1252,7 +1253,7 @@ static int __init ip_auto_config(void)
  __be32 addr;

 #ifdef CONFIG_PROC_FS
- proc_net_fops_create("pnp", S_IRUGO, &pnp_seq_fops);
+ proc_net_fops_create(init_net(), "pnp", S_IRUGO, &pnp_seq_fops);
 #endif /* CONFIG_PROC_FS */

  if (!ic_enable)
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
```

```
index ecb5422..af50394 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -63,6 +63,7 @@
 #include <linux/netfilter_ipv4.h>
 #include <net/ipip.h>
 #include <net/checksum.h>
+#include <net/net_namespace.h>

 #if defined(CONFIG_IP_PIMSM_V1) || defined(CONFIG_IP_PIMSM_V2)
 #define CONFIG_IP_PIMSM 1
@@ -1906,7 +1907,7 @@ void __init ip_mr_init(void)
  ipmr_expire_timer.function=ipmr_expire_process;
  register_netdevice_notifier(&ip_mr_notifier);
 #ifdef CONFIG_PROC_FS
- proc_net_fops_create("ip_mr_vif", 0, &ipmr_vif_fops);
- proc_net_fops_create("ip_mr_cache", 0, &ipmr_mfc_fops);
+ proc_net_fops_create(init_net(),"ip_mr_vif", 0, &ipmr_vif_fops);
+ proc_net_fops_create(init_net(),"ip_mr_cache", 0, &ipmr_mfc_fops);
 #endif
 }
diff --git a/net/ipv4/ipvs/ip_vs_app.c b/net/ipv4/ipvs/ip_vs_app.c
index 6c40899..4f44452 100644
--- a/net/ipv4/ipvs/ip_vs_app.c
+++ b/net/ipv4/ipvs/ip_vs_app.c
@@ -32,6 +32,7 @@
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>
 #include <linux/mutex.h>
+#include <net/net_namespace.h>

 #include <net/ip_vs.h>

@@ -618,12 +619,12 @@ int ip_vs_skb_replace(struct sk_buff *skb, gfp_t pri,
 int ip_vs_app_init(void)
 {
  /* we will replace it with proc_net_ipvs_create() soon */
- proc_net_fops_create("ip_vs_app", 0, &ip_vs_app_fops);
+ proc_net_fops_create(init_net(), "ip_vs_app", 0, &ip_vs_app_fops);
  return 0;
 }


 void ip_vs_app_cleanup(void)
 {
- proc_net_remove("ip_vs_app");
+ proc_net_remove(init_net(), "ip_vs_app");
 }
```

```
diff --git a/net/ipv4/ipvs/ip_vs_conn.c b/net/ipv4/ipvs/ip_vs_conn.c
index 8086787..0764e0f 100644
--- a/net/ipv4/ipvs/ip_vs_conn.c
+++ b/net/ipv4/ipvs/ip_vs_conn.c
@@ -34,6 +34,7 @@
 #include <linux/seq_file.h>
 #include <linux/jhash.h>
 #include <linux/random.h>
+#include <net/net_namespace.h>

 #include <net/ip_vs.h>

@@ -923,7 +924,7 @@ int ip_vs_conn_init(void)
   rwlock_init(&__ip_vs_conntbl_lock_array[idx].l);
  }

- proc_net_fops_create("ip_vs_conn", 0, &ip_vs_conn_fops);
+ proc_net_fops_create(init_net(), "ip_vs_conn", 0, &ip_vs_conn_fops);

  /* calculate the random value for connection hash */
  get_random_bytes(&ip_vs_conn_rnd, sizeof(ip_vs_conn_rnd));
@@ -939,6 +940,6 @@ void ip_vs_conn_cleanup(void)

  /* Release the empty cache */
  kmem_cache_destroy(ip_vs_conn_cachep);
- proc_net_remove("ip_vs_conn");
+ proc_net_remove(init_net(), "ip_vs_conn");
  vfree(ip_vs_conn_tab);
 }
diff --git a/net/ipv4/ipvs/ip_vs_ctl.c b/net/ipv4/ipvs/ip_vs_ctl.c
index c4e4237..d4bf160 100644
--- a/net/ipv4/ipvs/ip_vs_ctl.c
+++ b/net/ipv4/ipvs/ip_vs_ctl.c
@@ -39,6 +39,7 @@
 #include <net/ip.h>
 #include <net/route.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>

 #include <asm/uaccess.h>

@@ -2356,8 +2357,8 @@ int ip_vs_control_init(void)
   return ret;
  }

- proc_net_fops_create("ip_vs", 0, &ip_vs_info_fops);
- proc_net_fops_create("ip_vs_stats",0, &ip_vs_stats_fops);
+ proc_net_fops_create(init_net(), "ip_vs", 0, &ip_vs_info_fops);
```

```
+ proc_net_fops_create(init_net(), "ip_vs_stats",0, &ip_vs_stats_fops);

  sysctl_header = register_sysctl_table(vs_root_table);

@@ -2389,8 +2390,8 @@ void ip_vs_control_cleanup(void)
  cancel_rearming_delayed_work(&defense_work);
  ip_vs_kill_estimator(&ip_vs_stats);
  unregister_sysctl_table(sysctl_header);
- proc_net_remove("ip_vs_stats");
- proc_net_remove("ip_vs");
+ proc_net_remove(init_net(), "ip_vs_stats");
+ proc_net_remove(init_net(), "ip_vs");
  nf_unregister_sockopt(&ip_vs_sockopts);
  LeaveFunction(2);
 }
diff --git a/net/ipv4/ipvs/ip_vs_lblcr.c b/net/ipv4/ipvs/ip_vs_lblcr.c
index 22004f8..f8491e7 100644
--- a/net/ipv4/ipvs/ip_vs_lblcr.c
+++ b/net/ipv4/ipvs/ip_vs_lblcr.c
@@ -843,7 +843,7 @@ static int __init ip_vs_lblcr_init(void)
  INIT_LIST_HEAD(&ip_vs_lblcr_scheduler.n_list);
  sysctl_header = register_sysctl_table(lblcr_root_table);
 #ifdef CONFIG_IP_VS_LBLCR_DEBUG
- proc_net_create("ip_vs_lblcr", 0, ip_vs_lblcr_getinfo);
+ proc_net_create(init_net(), "ip_vs_lblcr", 0, ip_vs_lblcr_getinfo);
 #endif
  return register_ip_vs_scheduler(&ip_vs_lblcr_scheduler);
 }
@@ -852,7 +852,7 @@ static int __init ip_vs_lblcr_init(void)
 static void __exit ip_vs_lblcr_cleanup(void)
 {
 #ifdef CONFIG_IP_VS_LBLCR_DEBUG
- proc_net_remove("ip_vs_lblcr");
+ proc_net_remove(init_net(), "ip_vs_lblcr");
 #endif
  unregister_sysctl_table(sysctl_header);
  unregister_ip_vs_scheduler(&ip_vs_lblcr_scheduler);
diff --git a/net/ipv4/netfilter/ip_conntrack_standalone.c
b/net/ipv4/netfilter/ip_conntrack_standalone.c
index 9d89469..d04cbb0 100644
--- a/net/ipv4/netfilter/ip_conntrack_standalone.c
+++ b/net/ipv4/netfilter/ip_conntrack_standalone.c
@@ -828,14 +828,14 @@ static int __init ip_conntrack_standalone_init(void)

 #ifdef CONFIG_PROC_FS
  ret = -ENOMEM;
- proc = proc_net_fops_create("ip_conntrack", 0440, &ct_file_ops);
+ proc = proc_net_fops_create(init_net(), "ip_conntrack", 0440, &ct_file_ops);
```

```
  if (!proc) goto cleanup_init;

- proc_exp = proc_net_fops_create("ip_conntrack_expect", 0440,
+ proc_exp = proc_net_fops_create(init_net(), "ip_conntrack_expect", 0440,
      &exp_file_ops);
  if (!proc_exp) goto cleanup_proc;

- proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, proc_net_stat);
+ proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, per_net(proc_net_stat, init_net()));
  if (!proc_stat)
   goto cleanup_proc_exp;

@@ -864,11 +864,11 @@ static int __init ip_conntrack_standalone_init(void)
 #endif
  cleanup_proc_stat:
 #ifdef CONFIG_PROC_FS
- remove_proc_entry("ip_conntrack", proc_net_stat);
+ remove_proc_entry("ip_conntrack", per_net(proc_net_stat, init_net()));
  cleanup_proc_exp:
- proc_net_remove("ip_conntrack_expect");
+ proc_net_remove(init_net(), "ip_conntrack_expect");
  cleanup_proc:
- proc_net_remove("ip_conntrack");
+ proc_net_remove(init_net(), "ip_conntrack");
  cleanup_init:
 #endif /* CONFIG_PROC_FS */
  ip_conntrack_cleanup();
@@ -884,8 +884,8 @@ static void __exit ip_conntrack_standalone_fini(void)
  nf_unregister_hooks(ip_conntrack_ops, ARRAY_SIZE(ip_conntrack_ops));
 #ifdef CONFIG_PROC_FS
  remove_proc_entry("ip_conntrack", proc_net_stat);
- proc_net_remove("ip_conntrack_expect");
- proc_net_remove("ip_conntrack");
+ proc_net_remove(init_net(), "ip_conntrack_expect");
+ proc_net_remove(init_net(), "ip_conntrack");
 #endif /* CONFIG_PROC_FS */
  ip_conntrack_cleanup();
 }
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index 3446d4a..aae660c 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -38,6 +38,7 @@
 #include <linux/mutex.h>
 #include <net/sock.h>
 #include <net/route.h>
+#include <net/net_namespace.h>
```

```
 #define IPQ_QMAX_DEFAULT 1024
 #define IPQ_PROC_FS_NAME "ip_queue"
@@ -684,7 +685,7 @@ static int __init ip_queue_init(void)
   goto cleanup_netlink_notifier;
  }

- proc = proc_net_create(IPQ_PROC_FS_NAME, 0, ipq_get_info);
+ proc = proc_net_create(init_net(), IPQ_PROC_FS_NAME, 0, ipq_get_info);
  if (proc)
   proc->owner = THIS_MODULE;
  else {
@@ -705,7 +706,7 @@ static int __init ip_queue_init(void)
 cleanup_sysctl:
  unregister_sysctl_table(ipq_sysctl_header);
  unregister_netdevice_notifier(&ipq_dev_notifier);
- proc_net_remove(IPQ_PROC_FS_NAME);
+ proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

 cleanup_ipqnl:
  sock_release(ipqnl->sk_socket);
@@ -725,7 +726,7 @@ static void __exit ip_queue_fini(void)

  unregister_sysctl_table(ipq_sysctl_header);
  unregister_netdevice_notifier(&ipq_dev_notifier);
- proc_net_remove(IPQ_PROC_FS_NAME);
+ proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

  sock_release(ipqnl->sk_socket);
  mutex_lock(&ipqnl_mutex);
diff --git a/net/ipv4/netfilter/ipt_CLUSTERIP.c b/net/ipv4/netfilter/ipt_CLUSTERIP.c
index b1c1116..779e2c6 100644
--- a/net/ipv4/netfilter/ipt_CLUSTERIP.c
+++ b/net/ipv4/netfilter/ipt_CLUSTERIP.c
@@ -22,6 +22,7 @@
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>

+#include <net/net_namespace.h>
 #include <net/checksum.h>

 #include <linux/netfilter_arp.h>
@@ -736,7 +737,7 @@ static int __init ipt_clusterip_init(void)
   goto cleanup_target;

 #ifdef CONFIG_PROC_FS
- clusterip_procdir = proc_mkdir("ipt_CLUSTERIP", proc_net);
+ clusterip_procdir = proc_mkdir("ipt_CLUSTERIP", per_net(proc_net, init_net()));
  if (!clusterip_procdir) {
```

```
   printk(KERN_ERR "CLUSTERIP: Unable to proc dir entry\n");
   ret = -ENOMEM;
diff --git a/net/ipv4/netfilter/ipt_recent.c b/net/ipv4/netfilter/ipt_recent.c
index 4db0e73..4bfa2f9 100644
--- a/net/ipv4/netfilter/ipt_recent.c
+++ b/net/ipv4/netfilter/ipt_recent.c
@@ -23,6 +23,7 @@
 #include <linux/bitops.h>
 #include <linux/skbuff.h>
 #include <linux/inet.h>
+#include <net/net_namespace.h>

 #include <linux/netfilter_ipv4/ip_tables.h>
 #include <linux/netfilter_ipv4/ipt_recent.h>
@@ -483,7 +484,7 @@ static int __init ipt_recent_init(void)
 #ifdef CONFIG_PROC_FS
  if (err)
   return err;
- proc_dir = proc_mkdir("ipt_recent", proc_net);
+ proc_dir = proc_mkdir("ipt_recent", per_net(proc_net, init_net()));
  if (proc_dir == NULL) {
   ipt_unregister_match(&recent_match);
   err = -ENOMEM;
@@ -497,7 +498,7 @@ static void __exit ipt_recent_exit(void)
  BUG_ON(!list_empty(&tables));
  ipt_unregister_match(&recent_match);
 #ifdef CONFIG_PROC_FS
- remove_proc_entry("ipt_recent", proc_net);
+ remove_proc_entry("ipt_recent", per_net(proc_net, init_net()));
 #endif
 }

diff --git a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
index 3b31bc6..ebdb56e 100644
--- a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
+++ b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
@@ -11,6 +11,7 @@
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>
 #include <linux/percpu.h>
+#include <net/net_namespace.h>

 #include <linux/netfilter.h>
 #include <net/netfilter/nf_conntrack_core.h>
@@ -378,16 +379,16 @@ int __init nf_conntrack_ipv4_compat_init(void)
 {
  struct proc_dir_entry *proc, *proc_exp, *proc_stat;
```

```
- proc = proc_net_fops_create("ip_conntrack", 0440, &ct_file_ops);
+ proc = proc_net_fops_create(init_net(), "ip_conntrack", 0440, &ct_file_ops);
  if (!proc)
   goto err1;

- proc_exp = proc_net_fops_create("ip_conntrack_expect", 0440,
+ proc_exp = proc_net_fops_create(init_net(), "ip_conntrack_expect", 0440,
      &ip_exp_file_ops);
  if (!proc_exp)
   goto err2;

- proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, proc_net_stat);
+ proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, per_net(proc_net_stat, init_net()));
  if (!proc_stat)
   goto err3;

@@ -397,16 +398,16 @@ int __init nf_conntrack_ipv4_compat_init(void)
  return 0;

 err3:
- proc_net_remove("ip_conntrack_expect");
+ proc_net_remove(init_net(), "ip_conntrack_expect");
 err2:
- proc_net_remove("ip_conntrack");
+ proc_net_remove(init_net(), "ip_conntrack");
 err1:
  return -ENOMEM;
 }

 void __exit nf_conntrack_ipv4_compat_fini(void)
 {
- remove_proc_entry("ip_conntrack", proc_net_stat);
- proc_net_remove("ip_conntrack_expect");
- proc_net_remove("ip_conntrack");
+ remove_proc_entry("ip_conntrack", per_net(proc_net_stat, init_net()));
+ proc_net_remove(init_net(), "ip_conntrack_expect");
+ proc_net_remove(init_net(), "ip_conntrack");
 }
diff --git a/net/ipv4/proc.c b/net/ipv4/proc.c
index cd873da..c9c5601 100644
--- a/net/ipv4/proc.c
+++ b/net/ipv4/proc.c
@@ -44,6 +44,7 @@
 #include <linux/seq_file.h>
 #include <net/sock.h>
 #include <net/raw.h>
+#include <net/net_namespace.h>
```

```
 static int fold_prot_inuse(struct proto *proto)
 {
@@ -372,20 +373,20 @@ int __init ip_misc_proc_init(void)
 {
  int rc = 0;

- if (!proc_net_fops_create("netstat", S_IRUGO, &netstat_seq_fops))
+ if (!proc_net_fops_create(init_net(), "netstat", S_IRUGO, &netstat_seq_fops))
    goto out_netstat;

- if (!proc_net_fops_create("snmp", S_IRUGO, &snmp_seq_fops))
+ if (!proc_net_fops_create(init_net(), "snmp", S_IRUGO, &snmp_seq_fops))
    goto out_snmp;

- if (!proc_net_fops_create("sockstat", S_IRUGO, &sockstat_seq_fops))
+ if (!proc_net_fops_create(init_net(), "sockstat", S_IRUGO, &sockstat_seq_fops))
    goto out_sockstat;
 out:
  return rc;
 out_sockstat:
- proc_net_remove("snmp");
+ proc_net_remove(init_net(), "snmp");
 out_snmp:
- proc_net_remove("netstat");
+ proc_net_remove(init_net(), "netstat");
 out_netstat:
  rc = -ENOMEM;
  goto out;
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
index a6c63bb..38fe668 100644
--- a/net/ipv4/raw.c
+++ b/net/ipv4/raw.c
@@ -73,6 +73,7 @@
 #include <net/inet_common.h>
 #include <net/checksum.h>
 #include <net/xfrm.h>
+#include <net/net_namespace.h>
 #include <linux/rtnetlink.h>
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>
@@ -926,13 +927,13 @@ static struct file_operations raw_seq_fops = {

 int __init raw_proc_init(void)
 {
- if (!proc_net_fops_create("raw", S_IRUGO, &raw_seq_fops))
+ if (!proc_net_fops_create(init_net(), "raw", S_IRUGO, &raw_seq_fops))
    return -ENOMEM;
```

```
  return 0;
 }

 void __init raw_proc_exit(void)
 {
- proc_net_remove("raw");
+ proc_net_remove(init_net(), "raw");
 }
 #endif /* CONFIG_PROC_FS */
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 2daa0dc..8be7506 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -105,6 +105,7 @@
 #include <net/xfrm.h>
 #include <net/ip_mp_alg.h>
 #include <net/netevent.h>
+#include <net/net_namespace.h>
 #ifdef CONFIG_SYSCTL
 #include <linux/sysctl.h>
 #endif
@@ -3178,15 +3179,15 @@ int __init ip_rt_init(void)
 #ifdef CONFIG_PROC_FS
  {
  struct proc_dir_entry *rtstat_pde = NULL; /* keep gcc happy */
- if (!proc_net_fops_create("rt_cache", S_IRUGO, &rt_cache_seq_fops) ||
+ if (!proc_net_fops_create(init_net(), "rt_cache", S_IRUGO, &rt_cache_seq_fops) ||
      !(rtstat_pde = create_proc_entry("rt_cache", S_IRUGO,
-               proc_net_stat))) {
+               per_net(proc_net_stat, init_net())))) {
   return -ENOMEM;
  }
  rtstat_pde->proc_fops = &rt_cpu_seq_fops;
  }
 #ifdef CONFIG_NET_CLS_ROUTE
- create_proc_read_entry("rt_acct", 0, proc_net, ip_rt_acct_read, NULL);
+ create_proc_read_entry("rt_acct", 0, per_net(proc_net, init_net()), ip_rt_acct_read, NULL);
 #endif
 #endif
 #ifdef CONFIG_XFRM
diff --git a/net/ipv4/tcp_ipv4.c b/net/ipv4/tcp_ipv4.c
index 12de90a..ee4306f 100644
--- a/net/ipv4/tcp_ipv4.c
+++ b/net/ipv4/tcp_ipv4.c
@@ -71,6 +71,7 @@
 #include <net/timewait_sock.h>
 #include <net/xfrm.h>
 #include <net/netdma.h>
```

```
+#include <net/net_namespace.h>

 #include <linux/inet.h>
 #include <linux/ipv6.h>
@@ -2252,7 +2253,7 @@ int tcp_proc_register(struct tcp_seq_afinfo *afinfo)
  afinfo->seq_fops->llseek = seq_lseek;
  afinfo->seq_fops->release = seq_release_private;

- p = proc_net_fops_create(afinfo->name, S_IRUGO, afinfo->seq_fops);
+ p = proc_net_fops_create(init_net(), afinfo->name, S_IRUGO, afinfo->seq_fops);
  if (p)
   p->data = afinfo;
  else
@@ -2264,7 +2265,7 @@ void tcp_proc_unregister(struct tcp_seq_afinfo *afinfo)
 {
  if (!afinfo)
   return;
- proc_net_remove(afinfo->name);
+ proc_net_remove(init_net(), afinfo->name);
  memset(afinfo->seq_fops, 0, sizeof(*afinfo->seq_fops));
 }

diff --git a/net/ipv4/tcp_probe.c b/net/ipv4/tcp_probe.c
index f230eee..e8a3d96 100644
--- a/net/ipv4/tcp_probe.c
+++ b/net/ipv4/tcp_probe.c
@@ -159,7 +159,7 @@ static __init int tcpprobe_init(void)
  if (IS_ERR(tcpw.fifo))
   return PTR_ERR(tcpw.fifo);

- if (!proc_net_fops_create(procname, S_IRUSR, &tcpprobe_fops))
+ if (!proc_net_fops_create(init_net(), procname, S_IRUSR, &tcpprobe_fops))
   goto err0;

  ret = register_jprobe(&tcp_send_probe);
@@ -169,7 +169,7 @@ static __init int tcpprobe_init(void)
  pr_info("TCP watch registered (port=%d)\n", port);
  return 0;
  err1:
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
  err0:
  kfifo_free(tcpw.fifo);
  return ret;
@@ -179,7 +179,7 @@ module_init(tcpprobe_init);
 static __exit void tcpprobe_exit(void)
 {
  kfifo_free(tcpw.fifo);
```

```
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
  unregister_jprobe(&tcp_send_probe);

 }
diff --git a/net/ipv4/udp.c b/net/ipv4/udp.c
index cfff930..7527183 100644
--- a/net/ipv4/udp.c
+++ b/net/ipv4/udp.c
@@ -101,6 +101,7 @@
 #include <net/route.h>
 #include <net/checksum.h>
 #include <net/xfrm.h>
+#include <net/net_namespace.h>
 #include "udp_impl.h"

 /*
@@ -1643,7 +1644,7 @@ int udp_proc_register(struct udp_seq_afinfo *afinfo)
  afinfo->seq_fops->llseek = seq_lseek;
  afinfo->seq_fops->release = seq_release_private;

- p = proc_net_fops_create(afinfo->name, S_IRUGO, afinfo->seq_fops);
+ p = proc_net_fops_create(init_net(), afinfo->name, S_IRUGO, afinfo->seq_fops);
  if (p)
   p->data = afinfo;
  else
@@ -1655,7 +1656,7 @@ void udp_proc_unregister(struct udp_seq_afinfo *afinfo)
 {
  if (!afinfo)
   return;
- proc_net_remove(afinfo->name);
+ proc_net_remove(init_net(), afinfo->name);
  memset(afinfo->seq_fops, 0, sizeof(*afinfo->seq_fops));
 }

diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 6aded83..52bd4dd 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -74,6 +74,7 @@
 #include <net/tcp.h>
 #include <net/ip.h>
 #include <net/netlink.h>
+#include <net/net_namespace.h>
 #include <linux/if_tunnel.h>
 #include <linux/rtnetlink.h>

@@ -2780,14 +2781,14 @@ static struct file_operations if6_fops = {
```

```
 int __init if6_proc_init(void)
 {
- if (!proc_net_fops_create("if_inet6", S_IRUGO, &if6_fops))
+ if (!proc_net_fops_create(init_net(), "if_inet6", S_IRUGO, &if6_fops))
   return -ENOMEM;
  return 0;
 }

 void if6_proc_exit(void)
 {
- proc_net_remove("if_inet6");
+ proc_net_remove(init_net(), "if_inet6");
 }
 #endif /* CONFIG_PROC_FS */

@@ -4143,6 +4144,6 @@ void __exit addrconf_cleanup(void)
  rtnl_unlock();

 #ifdef CONFIG_PROC_FS
- proc_net_remove("if_inet6");
+ proc_net_remove(init_net(), "if_inet6");
 #endif
 }
diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index a960476..c42bad9 100644
--- a/net/ipv6/anycast.c
+++ b/net/ipv6/anycast.c
@@ -33,6 +33,7 @@

 #include <net/sock.h>
 #include <net/snmp.h>
+#include <net/net_namespace.h>

 #include <net/ipv6.h>
 #include <net/protocol.h>
@@ -575,7 +576,7 @@ static struct file_operations ac6_seq_fops = {

 int __init ac6_proc_init(void)
 {
- if (!proc_net_fops_create("anycast6", S_IRUGO, &ac6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "anycast6", S_IRUGO, &ac6_seq_fops))
   return -ENOMEM;

  return 0;
@@ -583,7 +584,7 @@ int __init ac6_proc_init(void)

 void ac6_proc_exit(void)
```

```
 {
- proc_net_remove("anycast6");
+ proc_net_remove(init_net(), "anycast6");
 }
 #endif

diff --git a/net/ipv6/ip6_flowlabel.c b/net/ipv6/ip6_flowlabel.c
index 624fae2..350aedb 100644
--- a/net/ipv6/ip6_flowlabel.c
+++ b/net/ipv6/ip6_flowlabel.c
@@ -22,6 +22,7 @@
 #include <linux/seq_file.h>

 #include <net/sock.h>
+#include <net/net_namespace.h>

 #include <net/ipv6.h>
 #include <net/ndisc.h>
@@ -690,7 +691,7 @@ static struct file_operations ip6fl_seq_fops = {
 void ip6_flowlabel_init(void)
 {
 #ifdef CONFIG_PROC_FS
- proc_net_fops_create("ip6_flowlabel", S_IRUGO, &ip6fl_seq_fops);
+ proc_net_fops_create(init_net(), "ip6_flowlabel", S_IRUGO, &ip6fl_seq_fops);
 #endif
 }

@@ -698,6 +699,6 @@ void ip6_flowlabel_cleanup(void)
 {
  del_timer(&ip6_fl_gc_timer);
 #ifdef CONFIG_PROC_FS
- proc_net_remove("ip6_flowlabel");
+ proc_net_remove(init_net(), "ip6_flowlabel");
 #endif
 }
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index a1c231a..2759571 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -51,6 +51,7 @@

 #include <net/sock.h>
 #include <net/snmp.h>
+#include <net/net_namespace.h>

 #include <net/ipv6.h>
 #include <net/protocol.h>
@@ -2661,8 +2662,8 @@ int __init igmp6_init(struct net_proto_family *ops)
```

```
  np->hop_limit = 1;

 #ifdef CONFIG_PROC_FS
- proc_net_fops_create("igmp6", S_IRUGO, &igmp6_mc_seq_fops);
- proc_net_fops_create("mcfilter6", S_IRUGO, &igmp6_mcf_seq_fops);
+ proc_net_fops_create(init_net(), "igmp6", S_IRUGO, &igmp6_mc_seq_fops);
+ proc_net_fops_create(init_net(), "mcfilter6", S_IRUGO, &igmp6_mcf_seq_fops);
 #endif

  return 0;
@@ -2674,7 +2675,7 @@ void igmp6_cleanup(void)
  igmp6_socket = NULL; /* for safety */

 #ifdef CONFIG_PROC_FS
- proc_net_remove("mcfilter6");
- proc_net_remove("igmp6");
+ proc_net_remove(init_net(), "mcfilter6");
+ proc_net_remove(init_net(), "igmp6");
 #endif
 }
diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index e774be7..45b64a5 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -36,6 +36,7 @@
 #include <linux/sysctl.h>
 #include <linux/proc_fs.h>
 #include <linux/mutex.h>
+#include <net/net_namespace.h>
 #include <net/sock.h>
 #include <net/ipv6.h>
 #include <net/ip6_route.h>
@@ -674,7 +675,7 @@ static int __init ip6_queue_init(void)
   goto cleanup_netlink_notifier;
  }

- proc = proc_net_create(IPQ_PROC_FS_NAME, 0, ipq_get_info);
+ proc = proc_net_create(init_net(), IPQ_PROC_FS_NAME, 0, ipq_get_info);
  if (proc)
   proc->owner = THIS_MODULE;
  else {
@@ -695,7 +696,7 @@ static int __init ip6_queue_init(void)
 cleanup_sysctl:
  unregister_sysctl_table(ipq_sysctl_header);
  unregister_netdevice_notifier(&ipq_dev_notifier);
- proc_net_remove(IPQ_PROC_FS_NAME);
+ proc_net_remove(init_net(), IPQ_PROC_FS_NAME);
```

```
 cleanup_ipqnl:
  sock_release(ipqnl->sk_socket);
@@ -715,7 +716,7 @@ static void __exit ip6_queue_fini(void)

  unregister_sysctl_table(ipq_sysctl_header);
  unregister_netdevice_notifier(&ipq_dev_notifier);
- proc_net_remove(IPQ_PROC_FS_NAME);
+ proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

  sock_release(ipqnl->sk_socket);
  mutex_lock(&ipqnl_mutex);
diff --git a/net/ipv6/proc.c b/net/ipv6/proc.c
index 35249d8..1827885 100644
--- a/net/ipv6/proc.c
+++ b/net/ipv6/proc.c
@@ -28,6 +28,7 @@
 #include <net/tcp.h>
 #include <net/transp_v6.h>
 #include <net/ipv6.h>
+#include <net/net_namespace.h>

 #ifdef CONFIG_PROC_FS
 static struct proc_dir_entry *proc_net_devsnmp6;
@@ -244,22 +245,22 @@ int __init ipv6_misc_proc_init(void)
 {
  int rc = 0;

- if (!proc_net_fops_create("snmp6", S_IRUGO, &snmp6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "snmp6", S_IRUGO, &snmp6_seq_fops))
   goto proc_snmp6_fail;

- proc_net_devsnmp6 = proc_mkdir("dev_snmp6", proc_net);
+ proc_net_devsnmp6 = proc_mkdir("dev_snmp6", per_net(proc_net, init_net()));
  if (!proc_net_devsnmp6)
   goto proc_dev_snmp6_fail;

- if (!proc_net_fops_create("sockstat6", S_IRUGO, &sockstat6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "sockstat6", S_IRUGO, &sockstat6_seq_fops))
   goto proc_sockstat6_fail;
 out:
  return rc;

 proc_sockstat6_fail:
- proc_net_remove("dev_snmp6");
+ proc_net_remove(init_net(), "dev_snmp6");
 proc_dev_snmp6_fail:
- proc_net_remove("snmp6");
+ proc_net_remove(init_net(), "snmp6");
```

```
 proc_snmp6_fail:
  rc = -ENOMEM;
  goto out;
@@ -267,9 +268,9 @@ proc_snmp6_fail:

 void ipv6_misc_proc_exit(void)
 {
- proc_net_remove("sockstat6");
- proc_net_remove("dev_snmp6");
- proc_net_remove("snmp6");
+ proc_net_remove(init_net(), "sockstat6");
+ proc_net_remove(init_net(), "dev_snmp6");
+ proc_net_remove(init_net(), "snmp6");
 }

 #else /* CONFIG_PROC_FS */
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index 4ae1b19..2e1825c 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -50,6 +50,7 @@
 #include <net/udp.h>
 #include <net/inet_common.h>
 #include <net/tcp_states.h>
+#include <net/net_namespace.h>
 #ifdef CONFIG_IPV6_MIP6
 #include <net/mip6.h>
 #endif
@@ -1274,13 +1275,13 @@ static struct file_operations raw6_seq_fops = {

 int __init raw6_proc_init(void)
 {
- if (!proc_net_fops_create("raw6", S_IRUGO, &raw6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "raw6", S_IRUGO, &raw6_seq_fops))
   return -ENOMEM;
  return 0;
 }

 void raw6_proc_exit(void)
 {
- proc_net_remove("raw6");
+ proc_net_remove(init_net(), "raw6");
 }
 #endif /* CONFIG_PROC_FS */
diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 8c3d568..8c9fef9 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
```

```
@@ -56,6 +56,7 @@
 #include <net/xfrm.h>
 #include <net/netevent.h>
 #include <net/netlink.h>
+#include <net/net_namespace.h>

 #include <asm/uaccess.h>

@@ -2458,11 +2459,11 @@ void __init ip6_route_init(void)
       SLAB_HWCACHE_ALIGN|SLAB_PANIC, NULL, NULL);
  fib6_init();
 #ifdef  CONFIG_PROC_FS
- p = proc_net_create("ipv6_route", 0, rt6_proc_info);
+ p = proc_net_create(init_net(), "ipv6_route", 0, rt6_proc_info);
  if (p)
   p->owner = THIS_MODULE;

- proc_net_fops_create("rt6_stats", S_IRUGO, &rt6_stats_seq_fops);
+ proc_net_fops_create(init_net(), "rt6_stats", S_IRUGO, &rt6_stats_seq_fops);
 #endif
 #ifdef CONFIG_XFRM
  xfrm6_init();
@@ -2478,8 +2479,8 @@ void ip6_route_cleanup(void)
  fib6_rules_cleanup();
 #endif
 #ifdef CONFIG_PROC_FS
- proc_net_remove("ipv6_route");
- proc_net_remove("rt6_stats");
+ proc_net_remove(init_net(), "ipv6_route");
+ proc_net_remove(init_net(), "rt6_stats");
 #endif
 #ifdef CONFIG_XFRM
  xfrm6_fini();
diff --git a/net/ipx/ipx_proc.c b/net/ipx/ipx_proc.c
index b7463df..bda8775 100644
--- a/net/ipx/ipx_proc.c
+++ b/net/ipx/ipx_proc.c
@@ -9,6 +9,7 @@
 #include <linux/proc_fs.h>
 #include <linux/spinlock.h>
 #include <linux/seq_file.h>
+#include <net/net_namespace.h>
 #include <net/tcp_states.h>
 #include <net/ipx.h>

@@ -353,7 +354,7 @@ int __init ipx_proc_init(void)
  struct proc_dir_entry *p;
  int rc = -ENOMEM;
```

```
- ipx_proc_dir = proc_mkdir("ipx", proc_net);
+ ipx_proc_dir = proc_mkdir("ipx", per_net(proc_net, init_net()));

  if (!ipx_proc_dir)
   goto out;
@@ -381,7 +382,7 @@ out_socket:
 out_route:
  remove_proc_entry("interface", ipx_proc_dir);
 out_interface:
- remove_proc_entry("ipx", proc_net);
+ remove_proc_entry("ipx", per_net(proc_net, init_net()));
  goto out;
 }

@@ -390,7 +391,7 @@ void __exit ipx_proc_exit(void)
  remove_proc_entry("interface", ipx_proc_dir);
  remove_proc_entry("route", ipx_proc_dir);
  remove_proc_entry("socket", ipx_proc_dir);
- remove_proc_entry("ipx", proc_net);
+ remove_proc_entry("ipx", per_net(proc_net, init_net()));
 }

 #else /* CONFIG_PROC_FS */
diff --git a/net/irda/irproc.c b/net/irda/irproc.c
index 88b9c43..0af0f55 100644
--- a/net/irda/irproc.c
+++ b/net/irda/irproc.c
@@ -28,6 +28,7 @@
 #include <linux/seq_file.h>
 #include <linux/module.h>
 #include <linux/init.h>
+#include <net/net_namespace.h>

 #include <net/irda/irda.h>
 #include <net/irda/irlap.h>
@@ -66,7 +67,7 @@ void __init irda_proc_register(void)
  int i;
  struct proc_dir_entry *d;

- proc_irda = proc_mkdir("irda", proc_net);
+ proc_irda = proc_mkdir("irda", per_net(proc_net, init_net()));
  if (proc_irda == NULL)
   return;
  proc_irda->owner = THIS_MODULE;
@@ -92,7 +93,7 @@ void __exit irda_proc_unregister(void)
                for (i=0; i<ARRAY_SIZE(irda_dirs); i++)
                        remove_proc_entry(irda_dirs[i].name, proc_irda);
```

```
-               remove_proc_entry("irda", proc_net);
+               remove_proc_entry("irda", per_net(proc_net, init_net()));
                proc_irda = NULL;
        }
 }
diff --git a/net/key/af_key.c b/net/key/af_key.c
index 5dd5094..c79f9c4 100644
--- a/net/key/af_key.c
+++ b/net/key/af_key.c
@@ -28,6 +28,7 @@
 #include <linux/init.h>
 #include <net/xfrm.h>
 #include <linux/audit.h>
+#include <net/net_namespace.h>

 #include <net/sock.h>

@@ -3292,7 +3293,7 @@ static struct xfrm_mgr pfkeyv2_mgr =
 static void __exit ipsec_pfkey_exit(void)
 {
  xfrm_unregister_km(&pfkeyv2_mgr);
- remove_proc_entry("net/pfkey", NULL);
+ remove_proc_entry("pfkey", per_net(proc_net, init_net()));
  sock_unregister(PF_KEY);
  proto_unregister(&key_proto);
 }
@@ -3309,7 +3310,7 @@ static int __init ipsec_pfkey_init(void)
   goto out_unregister_key_proto;
 #ifdef CONFIG_PROC_FS
  err = -ENOMEM;
- if (create_proc_read_entry("net/pfkey", 0, NULL, pfkey_read_proc, NULL) == NULL)
+ if (create_proc_read_entry("pfkey", 0, per_net(proc_net, init_net()), pfkey_read_proc, NULL) ==
NULL)
   goto out_sock_unregister;
 #endif
  err = xfrm_register_km(&pfkeyv2_mgr);
diff --git a/net/llc/llc_proc.c b/net/llc/llc_proc.c
index 19308fe..4d0a804 100644
--- a/net/llc/llc_proc.c
+++ b/net/llc/llc_proc.c
@@ -18,6 +18,7 @@
 #include <linux/errno.h>
 #include <linux/seq_file.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>
 #include <net/llc.h>
 #include <net/llc_c_ac.h>
```

```
 #include <net/llc_c_ev.h>
@@ -231,7 +232,7 @@ int __init llc_proc_init(void)
  int rc = -ENOMEM;
  struct proc_dir_entry *p;

- llc_proc_dir = proc_mkdir("llc", proc_net);
+ llc_proc_dir = proc_mkdir("llc", per_net(proc_net, init_net()));
  if (!llc_proc_dir)
   goto out;
  llc_proc_dir->owner = THIS_MODULE;
@@ -254,7 +255,7 @@ out:
 out_core:
  remove_proc_entry("socket", llc_proc_dir);
 out_socket:
- remove_proc_entry("llc", proc_net);
+ remove_proc_entry("llc", per_net(proc_net, init_net()));
  goto out;
 }

@@ -262,5 +263,5 @@ void llc_proc_exit(void)
 {
  remove_proc_entry("socket", llc_proc_dir);
  remove_proc_entry("core", llc_proc_dir);
- remove_proc_entry("llc", proc_net);
+ remove_proc_entry("llc", per_net(proc_net, init_net()));
 }
diff --git a/net/netfilter/core.c b/net/netfilter/core.c
index 291b8c6..cafa00c 100644
--- a/net/netfilter/core.c
+++ b/net/netfilter/core.c
@@ -23,6 +23,7 @@
 #include <linux/inetdevice.h>
 #include <linux/proc_fs.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>

 #include "nf_internals.h"

@@ -269,7 +270,7 @@ void __init netfilter_init(void)
  }

 #ifdef CONFIG_PROC_FS
- proc_net_netfilter = proc_mkdir("netfilter", proc_net);
+ proc_net_netfilter = proc_mkdir("netfilter", per_net(proc_net, init_net()));
  if (!proc_net_netfilter)
   panic("cannot create netfilter proc entry");
 #endif
diff --git a/net/netfilter/nf_conntrack_standalone.c b/net/netfilter/nf_conntrack_standalone.c
```

```
index 2587b49..314dc2c 100644
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -25,6 +25,7 @@
 #include <linux/seq_file.h>
 #include <linux/percpu.h>
 #include <linux/netdevice.h>
+#include <net/net_namespace.h>
 #ifdef CONFIG_SYSCTL
 #include <linux/sysctl.h>
 #endif
@@ -430,14 +431,14 @@ static int __init nf_conntrack_standalone_init(void)
   return ret;

 #ifdef CONFIG_PROC_FS
- proc = proc_net_fops_create("nf_conntrack", 0440, &ct_file_ops);
+ proc = proc_net_fops_create(init_net(), "nf_conntrack", 0440, &ct_file_ops);
  if (!proc) goto cleanup_init;

- proc_exp = proc_net_fops_create("nf_conntrack_expect", 0440,
+ proc_exp = proc_net_fops_create(init_net(), "nf_conntrack_expect", 0440,
      &exp_file_ops);
  if (!proc_exp) goto cleanup_proc;

- proc_stat = create_proc_entry("nf_conntrack", S_IRUGO, proc_net_stat);
+ proc_stat = create_proc_entry("nf_conntrack", S_IRUGO, per_net(proc_net_stat, init_net()));
  if (!proc_stat)
   goto cleanup_proc_exp;

@@ -458,11 +459,11 @@ static int __init nf_conntrack_standalone_init(void)
  cleanup_proc_stat:
 #endif
 #ifdef CONFIG_PROC_FS
- remove_proc_entry("nf_conntrack", proc_net_stat);
+ remove_proc_entry("nf_conntrack", per_net(proc_net_stat, init_net()));
  cleanup_proc_exp:
- proc_net_remove("nf_conntrack_expect");
+ proc_net_remove(init_net(), "nf_conntrack_expect");
  cleanup_proc:
- proc_net_remove("nf_conntrack");
+ proc_net_remove(init_net(), "nf_conntrack");
  cleanup_init:
 #endif /* CNFIG_PROC_FS */
  nf_conntrack_cleanup();
@@ -475,9 +476,9 @@ static void __exit nf_conntrack_standalone_fini(void)
   unregister_sysctl_table(nf_ct_sysctl_header);
 #endif
 #ifdef CONFIG_PROC_FS
```

```
- remove_proc_entry("nf_conntrack", proc_net_stat);
- proc_net_remove("nf_conntrack_expect");
- proc_net_remove("nf_conntrack");
+ remove_proc_entry("nf_conntrack", per_net(proc_net_stat, init_net()));
+ proc_net_remove(init_net(), "nf_conntrack_expect");
+ proc_net_remove(init_net(), "nf_conntrack");
 #endif /* CNFIG_PROC_FS */
  nf_conntrack_cleanup();
 }
diff --git a/net/netfilter/x_tables.c b/net/netfilter/x_tables.c
index 8996584..9fb3491 100644
--- a/net/netfilter/x_tables.c
+++ b/net/netfilter/x_tables.c
@@ -22,6 +22,7 @@
 #include <linux/vmalloc.h>
 #include <linux/mutex.h>
 #include <linux/mm.h>
+#include <net/net_namespace.h>

 #include <linux/netfilter/x_tables.h>
 #include <linux/netfilter_arp.h>
@@ -800,7 +801,7 @@ int xt_proto_init(int af)
 #ifdef CONFIG_PROC_FS
  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_TABLES, sizeof(buf));
- proc = proc_net_fops_create(buf, 0440, &xt_file_ops);
+ proc = proc_net_fops_create(init_net(), buf, 0440, &xt_file_ops);
  if (!proc)
   goto out;
  proc->data = (void *) ((unsigned long) af | (TABLE << 16));
@@ -808,14 +809,14 @@ int xt_proto_init(int af)

  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_MATCHES, sizeof(buf));
- proc = proc_net_fops_create(buf, 0440, &xt_file_ops);
+ proc = proc_net_fops_create(init_net(), buf, 0440, &xt_file_ops);
  if (!proc)
   goto out_remove_tables;
  proc->data = (void *) ((unsigned long) af | (MATCH << 16));

  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_TARGETS, sizeof(buf));
- proc = proc_net_fops_create(buf, 0440, &xt_file_ops);
+ proc = proc_net_fops_create(init_net(), buf, 0440, &xt_file_ops);
  if (!proc)
   goto out_remove_matches;
  proc->data = (void *) ((unsigned long) af | (TARGET << 16));
@@ -827,12 +828,12 @@ int xt_proto_init(int af)
```

```
 out_remove_matches:
  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_MATCHES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);

 out_remove_tables:
  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_TABLES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);
 out:
  return -1;
 #endif
@@ -846,15 +847,15 @@ void xt_proto_fini(int af)

  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_TABLES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);

  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_TARGETS, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);

  strlcpy(buf, xt_proto_prefix[af], sizeof(buf));
  strlcat(buf, FORMAT_MATCHES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);
 #endif /*CONFIG_PROC_FS*/
 }
 EXPORT_SYMBOL_GPL(xt_proto_fini);
diff --git a/net/netfilter/xt_hashlimit.c b/net/netfilter/xt_hashlimit.c
index f28bf69..21c07df 100644
--- a/net/netfilter/xt_hashlimit.c
+++ b/net/netfilter/xt_hashlimit.c
@@ -21,6 +21,7 @@
 #include <linux/in.h>
 #include <linux/ip.h>
 #include <linux/ipv6.h>
+#include <net/net_namespace.h>

 #include <linux/netfilter/x_tables.h>
 #include <linux/netfilter_ipv4/ip_tables.h>
@@ -737,13 +738,13 @@ static int __init xt_hashlimit_init(void)
   printk(KERN_ERR "xt_hashlimit: unable to create slab cache\n");
   goto err2;
```

```
  }
- hashlimit_procdir4 = proc_mkdir("ipt_hashlimit", proc_net);
+ hashlimit_procdir4 = proc_mkdir("ipt_hashlimit", per_net(proc_net, init_net()));
  if (!hashlimit_procdir4) {
   printk(KERN_ERR "xt_hashlimit: unable to create proc dir "
     "entry\n");
   goto err3;
  }
- hashlimit_procdir6 = proc_mkdir("ip6t_hashlimit", proc_net);
+ hashlimit_procdir6 = proc_mkdir("ip6t_hashlimit", per_net(proc_net, init_net()));
  if (!hashlimit_procdir6) {
   printk(KERN_ERR "xt_hashlimit: unable to create proc dir "
     "entry\n");
@@ -751,7 +752,7 @@ static int __init xt_hashlimit_init(void)
  }
  return 0;
 err4:
- remove_proc_entry("ipt_hashlimit", proc_net);
+ remove_proc_entry("ipt_hashlimit", per_net(proc_net, init_net()));
 err3:
  kmem_cache_destroy(hashlimit_cachep);
 err2:
@@ -763,8 +764,8 @@ err1:

 static void __exit xt_hashlimit_fini(void)
 {
- remove_proc_entry("ipt_hashlimit", proc_net);
- remove_proc_entry("ip6t_hashlimit", proc_net);
+ remove_proc_entry("ipt_hashlimit", per_net(proc_net, init_net()));
+ remove_proc_entry("ip6t_hashlimit", per_net(proc_net, init_net()));
  kmem_cache_destroy(hashlimit_cachep);
  xt_unregister_matches(xt_hashlimit, ARRAY_SIZE(xt_hashlimit));
 }
diff --git a/net/netlink/af_netlink.c b/net/netlink/af_netlink.c
index 383dd4e..3c00f48 100644
--- a/net/netlink/af_netlink.c
+++ b/net/netlink/af_netlink.c
@@ -60,6 +60,7 @@
 #include <net/sock.h>
 #include <net/scm.h>
 #include <net/netlink.h>
+#include <net/net_namespace.h>

 #define NLGRPSZ(x) (ALIGN(x, sizeof(unsigned long) * 8) / 8)

@@ -1806,7 +1807,7 @@ static int __init netlink_proto_init(void)

  sock_register(&netlink_family_ops);
```

```
 #ifdef CONFIG_PROC_FS
- proc_net_fops_create("netlink", 0, &netlink_seq_fops);
+ proc_net_fops_create(init_net(), "netlink", 0, &netlink_seq_fops);
 #endif
  /* The netlink device handler may be needed early. */
  rtnetlink_init();
diff --git a/net/netrom/af_netrom.c b/net/netrom/af_netrom.c
index 43bbe2c..601d58c 100644
--- a/net/netrom/af_netrom.c
+++ b/net/netrom/af_netrom.c
@@ -41,6 +41,7 @@
 #include <net/ip.h>
 #include <net/tcp_states.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>
 #include <linux/init.h>

 static int nr_ndevs = 4;
@@ -1442,9 +1443,9 @@ static int __init nr_proto_init(void)

  nr_loopback_init();

- proc_net_fops_create("nr", S_IRUGO, &nr_info_fops);
- proc_net_fops_create("nr_neigh", S_IRUGO, &nr_neigh_fops);
- proc_net_fops_create("nr_nodes", S_IRUGO, &nr_nodes_fops);
+ proc_net_fops_create(init_net(), "nr", S_IRUGO, &nr_info_fops);
+ proc_net_fops_create(init_net(), "nr_neigh", S_IRUGO, &nr_neigh_fops);
+ proc_net_fops_create(init_net(), "nr_nodes", S_IRUGO, &nr_nodes_fops);
 out:
  return rc;
 fail:
@@ -1472,9 +1473,9 @@ static void __exit nr_exit(void)
 {
  int i;

- proc_net_remove("nr");
- proc_net_remove("nr_neigh");
- proc_net_remove("nr_nodes");
+ proc_net_remove(init_net(), "nr");
+ proc_net_remove(init_net(), "nr_neigh");
+ proc_net_remove(init_net(), "nr_nodes");
  nr_loopback_clear();

  nr_rt_free();
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index da73e8a..04e295a 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
```

```
@@ -65,6 +65,7 @@
 #include <net/protocol.h>
 #include <linux/skbuff.h>
 #include <net/sock.h>
+#include <net/net_namespace.h>
 #include <linux/errno.h>
 #include <linux/timer.h>
 #include <asm/system.h>
@@ -1911,7 +1912,7 @@ static struct file_operations packet_seq_fops = {

 static void __exit packet_exit(void)
 {
- proc_net_remove("packet");
+ proc_net_remove(init_net(), "packet");
  unregister_netdevice_notifier(&packet_netdev_notifier);
  sock_unregister(PF_PACKET);
  proto_unregister(&packet_proto);
@@ -1926,7 +1927,7 @@ static int __init packet_init(void)

  sock_register(&packet_family_ops);
  register_netdevice_notifier(&packet_netdev_notifier);
- proc_net_fops_create("packet", 0, &packet_seq_fops);
+ proc_net_fops_create(init_net(), "packet", 0, &packet_seq_fops);
 out:
  return rc;
 }
diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index 9e27946..5532340 100644
--- a/net/rose/af_rose.c
+++ b/net/rose/af_rose.c
@@ -45,6 +45,7 @@
 #include <net/tcp_states.h>
 #include <net/ip.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>

 static int rose_ndevs = 10;

@@ -1550,10 +1551,10 @@ static int __init rose_proto_init(void)

  rose_add_loopback_neigh();

- proc_net_fops_create("rose", S_IRUGO, &rose_info_fops);
- proc_net_fops_create("rose_neigh", S_IRUGO, &rose_neigh_fops);
- proc_net_fops_create("rose_nodes", S_IRUGO, &rose_nodes_fops);
- proc_net_fops_create("rose_routes", S_IRUGO, &rose_routes_fops);
+ proc_net_fops_create(init_net(), "rose", S_IRUGO, &rose_info_fops);
+ proc_net_fops_create(init_net(), "rose_neigh", S_IRUGO, &rose_neigh_fops);
```

```
+ proc_net_fops_create(init_net(), "rose_nodes", S_IRUGO, &rose_nodes_fops);
+ proc_net_fops_create(init_net(), "rose_routes", S_IRUGO, &rose_routes_fops);
 out:
  return rc;
 fail:
@@ -1580,10 +1581,10 @@ static void __exit rose_exit(void)
 {
  int i;

- proc_net_remove("rose");
- proc_net_remove("rose_neigh");
- proc_net_remove("rose_nodes");
- proc_net_remove("rose_routes");
+ proc_net_remove(init_net(), "rose");
+ proc_net_remove(init_net(), "rose_neigh");
+ proc_net_remove(init_net(), "rose_nodes");
+ proc_net_remove(init_net(), "rose_routes");
  rose_loopback_clear();

  rose_rt_free();
diff --git a/net/rxrpc/proc.c b/net/rxrpc/proc.c
index 29975d9..e7bd87b 100644
--- a/net/rxrpc/proc.c
+++ b/net/rxrpc/proc.c
@@ -14,6 +14,7 @@
 #include <linux/module.h>
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>
+#include <net/net_namespace.h>
 #include <rxrpc/rxrpc.h>
 #include <rxrpc/transport.h>
 #include <rxrpc/peer.h>
@@ -133,7 +134,7 @@ int rxrpc_proc_init(void)
 {
  struct proc_dir_entry *p;

- proc_rxrpc = proc_mkdir("rxrpc", proc_net);
+ proc_rxrpc = proc_mkdir("rxrpc", per_net(proc_net, init_net()));
  if (!proc_rxrpc)
   goto error;
  proc_rxrpc->owner = THIS_MODULE;
@@ -169,7 +170,7 @@ int rxrpc_proc_init(void)
  error_calls:
  remove_proc_entry("calls", proc_rxrpc);
  error_proc:
- remove_proc_entry("rxrpc", proc_net);
+ remove_proc_entry("rxrpc", per_net(proc_net, init_net()));
  error:
```

```
  return -ENOMEM;
 } /* end rxrpc_proc_init() */
@@ -185,7 +186,7 @@ void rxrpc_proc_cleanup(void)
  remove_proc_entry("connections", proc_rxrpc);
  remove_proc_entry("calls", proc_rxrpc);

- remove_proc_entry("rxrpc", proc_net);
+ remove_proc_entry("rxrpc", per_net(proc_net, init_net()));

 } /* end rxrpc_proc_cleanup() */

diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index 65825f4..da7e1eb 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -36,6 +36,7 @@
 #include <linux/list.h>
 #include <linux/bitops.h>

+#include <net/net_namespace.h>
 #include <net/sock.h>
 #include <net/pkt_sched.h>

@@ -1296,7 +1297,7 @@ static int __init pktsched_init(void)

  register_qdisc(&pfifo_qdisc_ops);
  register_qdisc(&bfifo_qdisc_ops);
- proc_net_fops_create("psched", 0, &psched_fops);
+ proc_net_fops_create(init_net(), "psched", 0, &psched_fops);

  return 0;
 }
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 225f39b..ea94951 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -59,6 +59,7 @@
 #include <net/addrconf.h>
 #include <net/inet_common.h>
 #include <net/inet_ecn.h>
+#include <net/net_namespace.h>

 /* Global data structures. */
 struct sctp_globals sctp_globals __read_mostly;
@@ -93,7 +94,7 @@ static __init int sctp_proc_init(void)
 {
  if (!proc_net_sctp) {
   struct proc_dir_entry *ent;
```

```
-  ent = proc_mkdir("net/sctp", NULL);
+  ent = proc_mkdir("sctp", per_net(proc_net, init_net()));
   if (ent) {
    ent->owner = THIS_MODULE;
    proc_net_sctp = ent;
@@ -126,7 +127,7 @@ static void sctp_proc_exit(void)

  if (proc_net_sctp) {
   proc_net_sctp = NULL;
-  remove_proc_entry("net/sctp", NULL);
+  remove_proc_entry("sctp", per_net(proc_net, init_net()));
  }
 }

diff --git a/net/sunrpc/stats.c b/net/sunrpc/stats.c
index bd98124..996b71c 100644
--- a/net/sunrpc/stats.c
+++ b/net/sunrpc/stats.c
@@ -22,6 +22,7 @@
 #include <linux/sunrpc/clnt.h>
 #include <linux/sunrpc/svcsock.h>
 #include <linux/sunrpc/metrics.h>
+#include <net/net_namespace.h>

 #define RPCDBG_FACILITY RPCDBG_MISC

@@ -266,7 +267,7 @@ rpc_proc_init(void)
  dprintk("RPC: registering /proc/net/rpc\n");
  if (!proc_net_rpc) {
   struct proc_dir_entry *ent;
-  ent = proc_mkdir("rpc", proc_net);
+  ent = proc_mkdir("rpc", per_net(proc_net, init_net()));
   if (ent) {
    ent->owner = THIS_MODULE;
    proc_net_rpc = ent;
@@ -280,7 +281,7 @@ rpc_proc_exit(void)
  dprintk("RPC: unregistering /proc/net/rpc\n");
  if (proc_net_rpc) {
   proc_net_rpc = NULL;
-  remove_proc_entry("net/rpc", NULL);
+  remove_proc_entry("rpc", per_net(proc_net, init_net()));
  }
 }

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 2f208c7..30855e1 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
```

```
@@ -116,6 +116,7 @@
 #include <linux/mount.h>
 #include <net/checksum.h>
 #include <linux/security.h>
+#include <net/net_namespace.h>

 int sysctl_unix_max_dgram_qlen __read_mostly = 10;

@@ -2072,7 +2073,7 @@ static int __init af_unix_init(void)

  sock_register(&unix_family_ops);
 #ifdef CONFIG_PROC_FS
- proc_net_fops_create("unix", 0, &unix_seq_fops);
+ proc_net_fops_create(init_net(), "unix", 0, &unix_seq_fops);
 #endif
  unix_sysctl_register();
 out:
@@ -2083,7 +2084,7 @@ static void __exit af_unix_exit(void)
 {
  sock_unregister(PF_UNIX);
  unix_sysctl_unregister();
- proc_net_remove("unix");
+ proc_net_remove(init_net(), "unix");
  proto_unregister(&unix_proto);
 }

diff --git a/net/wanrouter/wanproc.c b/net/wanrouter/wanproc.c
index 930ea59..1fcb0b8 100644
--- a/net/wanrouter/wanproc.c
+++ b/net/wanrouter/wanproc.c
@@ -28,6 +28,7 @@
 #include <linux/wanrouter.h> /* WAN router API definitions */
 #include <linux/seq_file.h>
 #include <linux/smp_lock.h>
+#include <net/net_namespace.h>

 #include <asm/io.h>

@@ -287,7 +288,7 @@ static struct file_operations wandev_fops = {
 int __init wanrouter_proc_init(void)
 {
  struct proc_dir_entry *p;
- proc_router = proc_mkdir(ROUTER_NAME, proc_net);
+ proc_router = proc_mkdir(ROUTER_NAME, per_net(proc_net, init_net()));
  if (!proc_router)
   goto fail;

@@ -303,7 +304,7 @@ int __init wanrouter_proc_init(void)
```

```
 fail_stat:
  remove_proc_entry("config", proc_router);
 fail_config:
- remove_proc_entry(ROUTER_NAME, proc_net);
+ remove_proc_entry(ROUTER_NAME, per_net(proc_net, init_net()));
 fail:
  return -ENOMEM;
 }
@@ -316,7 +317,7 @@ void wanrouter_proc_cleanup(void)
 {
  remove_proc_entry("config", proc_router);
  remove_proc_entry("status", proc_router);
- remove_proc_entry(ROUTER_NAME, proc_net);
+ remove_proc_entry(ROUTER_NAME, per_net(proc_net, init_net()));
 }

 /*
diff --git a/net/x25/x25_proc.c b/net/x25/x25_proc.c
index a11837d..7bcf98d 100644
--- a/net/x25/x25_proc.c
+++ b/net/x25/x25_proc.c
@@ -20,6 +20,7 @@
 #include <linux/init.h>
 #include <linux/proc_fs.h>
 #include <linux/seq_file.h>
+#include <net/net_namespace.h>
 #include <net/sock.h>
 #include <net/x25.h>

@@ -212,7 +213,7 @@ int __init x25_proc_init(void)
  struct proc_dir_entry *p;
  int rc = -ENOMEM;

- x25_proc_dir = proc_mkdir("x25", proc_net);
+ x25_proc_dir = proc_mkdir("x25", per_net(proc_net, init_net()));
  if (!x25_proc_dir)
   goto out;

@@ -231,7 +232,7 @@ out:
 out_socket:
  remove_proc_entry("route", x25_proc_dir);
 out_route:
- remove_proc_entry("x25", proc_net);
+ remove_proc_entry("x25", per_net(proc_net, init_net()));
  goto out;
 }

@@ -239,7 +240,7 @@ void __exit x25_proc_exit(void)
```

```
 {
  remove_proc_entry("route", x25_proc_dir);
  remove_proc_entry("socket", x25_proc_dir);
- remove_proc_entry("x25", proc_net);
+ remove_proc_entry("x25", per_net(proc_net, init_net()));
 }

 #else /* CONFIG_PROC_FS */
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 8/31] net: Make /sys/class/net handle multiple network namespaces

Posted by ebiederm on Thu, 25 Jan 2007 19:00:10 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

In combination with the sysfs support I am in the process of merging
with gregkh, creates a separate instance of the /sys/class/net directory
for each network namespace so two devices with the same name do not conflict.
Then a network namespace sensitive follow link method on the /sys/class/net
directory ensures that you see the directory instance for your current network
namespace.

Ensuring all existing applications continue to see what we is currently
present in sysfs.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/core/net-sysfs.c | 53 ++-
 1 files changed, 52 insertions(+), 1 deletions(-)

diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
index 5d08cc9..b08c1be 100644
--- a/net/core/net-sysfs.c
+++ b/net/core/net-sysfs.c
@@ -11,12 +11,14 @@

 #include <linux/capability.h>
 #include <linux/kernel.h>
+#include <linux/sysfs.h>

```
 #include <linux/netdevice.h>
 #include <linux/if_arp.h>
 #include <net/sock.h>
 #include <linux/rtnetlink.h>
 #include <linux/wireless.h>
 #include <net/iw_handler.h>
+#include <net/net_namespace.h>

 #define to_class_dev(obj) container_of(obj,struct class_device,kobj)
 #define to_net_dev(class) container_of(class, struct net_device, class_dev)
@@ -431,6 +433,24 @@ static void netdev_release(struct class_device *cd)
  kfree((char *)dev - dev->padded);
 }

+static DEFINE_PER_NET(struct dentry *, net_shadow) = NULL;
+
+static struct dentry *net_class_device_dparent(struct class_device *cd)
+{
+ struct net_device *dev
+  = container_of(cd, struct net_device, class_dev);
+ net_t net = dev->nd_net;
+
+ return per_net(net_shadow, net);
+}
+
+static void *class_net_follow_link(struct dentry *dentry, struct nameidata *nd)
+{
+ dput(nd->dentry);
+ nd->dentry = dget(per_net(net_shadow, current->nsproxy->net_ns));
+ return NULL;
+}
+
 static struct class net_class = {
  .name = "net",
  .release = netdev_release,
@@ -438,6 +458,8 @@ static struct class net_class = {
 #ifdef CONFIG_HOTPLUG
  .uevent = netdev_uevent,
 #endif
+ .class_device_dparent = net_class_device_dparent,
+ .class_follow_link = class_net_follow_link,
 };

 void netdev_unregister_sysfs(struct net_device * dev)
@@ -470,7 +492,36 @@ int netdev_register_sysfs(struct net_device *dev)
  return class_device_add(class_dev);
 }
```

```
+static int netdev_sysfs_net_init(net_t net)
+{
+ struct dentry *shadow;
+ int error = 0;
+ shadow = sysfs_create_shadow_dir(&net_class.subsys.kset.kobj);
+ if (IS_ERR(shadow))
+  error = PTR_ERR(shadow);
+ else
+  per_net(net_shadow, net) = shadow;
+ return error;
+}
+
+static void netdev_sysfs_net_exit(net_t net)
+{
+ sysfs_remove_shadow_dir(per_net(net_shadow, net));
+ per_net(net_shadow, net) = NULL;
+}
+
+static struct pernet_operations netdev_sysfs_ops = {
+ .init = netdev_sysfs_net_init,
+ .exit = netdev_sysfs_net_exit,
+};
+
 int netdev_sysfs_init(void)
 {
- return class_register(&net_class);
+ int rc;
+ if ((rc = class_register(&net_class)))
+  goto out;
+ if ((rc = register_pernet_subsys(&netdev_sysfs_ops)))
+  goto out;
+out:
+ return rc;
 }
--
1.4.4.1.g278f
```


Subject: [PATCH RFC 9/31] net: Implement the per network namespace sysctl infrastructure
Posted by ebiederm on Thu, 25 Jan 2007 19:00:11 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

The user interface is: register_net_sysctl_table and
unregister_net_sysctl_table. Very much like the current
interface except there is an network namespace parameter.

This this any sysctl in the net_root_table and it's
subdirectories are registered with register_net_sysctl
shows up only to tasks in the same network namespace.

All other sysctls continue to be globally visible.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/sysctl.h | 7 ++++
 include/net/sock.h | 1 +
 kernel/sysctl.c | 71 ++-
 net/core/sysctl_net_core.c | 5 +++
 net/sysctl_net.c | 20 ++++++++++++
 5 files changed, 102 insertions(+), 2 deletions(-)

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 8eba2d2..286e723 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -1044,6 +1044,13 @@ struct ctl_table_header * register_sysctl_table(ctl_table * table);

 void unregister_sysctl_table(struct ctl_table_header * table);

+#ifdef CONFIG_NET
+#include <linux/net_namespace_type.h>
+extern struct ctl_table_header *register_net_sysctl_table(net_t net, struct ctl_table *table);
+extern void unregister_net_sysctl_table(struct ctl_table_header *header);
+DECLARE_PER_NET(struct ctl_table, net_root_table[]);
+#endif
+
 #else /* __KERNEL__ */

 #endif /* __KERNEL__ */
diff --git a/include/net/sock.h b/include/net/sock.h
index 5bf6bb5..01a2781 100644
--- a/include/net/sock.h
+++ b/include/net/sock.h
@@ -1414,6 +1414,7 @@ extern void sk_init(void);

 #ifdef CONFIG_SYSCTL
 extern struct ctl_table core_table[];
+DECLARE_PER_NET(struct ctl_table, multi_core_table[]);

```
 #endif

 extern int sysctl_optmem_max;
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 7da313e..ae6a424 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -45,6 +45,7 @@
 #include <linux/syscalls.h>
 #include <linux/nfs_fs.h>
 #include <linux/acpi.h>
+#include <net/net_namespace.h>

 #include <asm/uaccess.h>
 #include <asm/processor.h>
@@ -135,6 +136,10 @@ static int proc_do_cad_pid(ctl_table *table, int write, struct file *filp,
     void __user *buffer, size_t *lenp, loff_t *ppos);
 #endif

+#ifdef CONFIG_NET
+static DEFINE_PER_NET(struct ctl_table_header, net_table_header);
+#endif
+
 static ctl_table root_table[];
 static struct ctl_table_header root_table_header =
  { root_table, LIST_HEAD_INIT(root_table_header.ctl_entry) };
@@ -1059,6 +1064,7 @@ struct ctl_table_header *sysctl_head_next(struct ctl_table_header
*prev)
 {
  struct ctl_table_header *head;
  struct list_head *tmp;
+ net_t net = current->nsproxy->net_ns;
  spin_lock(&sysctl_lock);
  if (prev) {
   tmp = &prev->ctl_entry;
@@ -1076,6 +1082,10 @@ struct ctl_table_header *sysctl_head_next(struct ctl_table_header
*prev)
  next:
   tmp = tmp->next;
   if (tmp == &root_table_header.ctl_entry)
+#ifdef CONFIG_NET
+   tmp = &per_net(net_table_header, net).ctl_entry;
+  else if (tmp == &per_net(net_table_header, net).ctl_entry)
+#endif
    break;
  }
  spin_unlock(&sysctl_lock);
@@ -1290,7 +1300,8 @@ int do_sysctl_strategy (ctl_table *table,
```

```
  * This routine returns %NULL on a failure to register, and a pointer
  * to the table header on success.
  */
-struct ctl_table_header *register_sysctl_table(ctl_table * table)
+static struct ctl_table_header *__register_sysctl_table(
+ struct ctl_table_header *root, ctl_table * table)
 {
  struct ctl_table_header *tmp;
  tmp = kmalloc(sizeof(struct ctl_table_header), GFP_KERNEL);
@@ -1301,11 +1312,16 @@ struct ctl_table_header *register_sysctl_table(ctl_table * table)
  tmp->used = 0;
  tmp->unregistering = NULL;
  spin_lock(&sysctl_lock);
- list_add_tail(&tmp->ctl_entry, &root_table_header.ctl_entry);
+ list_add_tail(&tmp->ctl_entry, &root->ctl_entry);
  spin_unlock(&sysctl_lock);
  return tmp;
 }

+struct ctl_table_header *register_sysctl_table(ctl_table *table)
+{
+ return __register_sysctl_table(&root_table_header, table);
+}
+
 /**
  * unregister_sysctl_table - unregister a sysctl table hierarchy
  * @header: the header returned from register_sysctl_table
@@ -1322,6 +1338,57 @@ void unregister_sysctl_table(struct ctl_table_header * header)
  kfree(header);
 }

+#ifdef CONFIG_NET
+
+static void *fixup_per_net_addr(net_t net, void *addr)
+{
+ char *ptr = addr;
+ if ((ptr >= __per_net_start) && (ptr < __per_net_end))
+  ptr += __per_net_offset(net);
+ return ptr;
+}
+
+static void sysctl_net_table_fixup(net_t net, struct ctl_table *table)
+{
+ for (; table->ctl_name || table->procname; table++) {
+  table->child  = fixup_per_net_addr(net, table->child);
+  table->data   = fixup_per_net_addr(net, table->data);
+  table->extra1 = fixup_per_net_addr(net, table->extra1);
+  table->extra2 = fixup_per_net_addr(net, table->extra2);
```

```
+
+  /* Whee recursive functions on the kernel stack */
+  if (table->child)
+   sysctl_net_table_fixup(net, table->child);
+ }
+}
+
+static void sysctl_net_init(net_t net)
+{
+ struct ctl_table *table = per_net(net_root_table, net);
+
+ sysctl_net_table_fixup(net, table);
+ per_net(net_table_header, net).ctl_table = table;
+
+ INIT_LIST_HEAD(&per_net(net_table_header, net).ctl_entry);
+}
+
+struct ctl_table_header *register_net_sysctl_table(net_t net, ctl_table *table)
+{
+ if (!per_net(net_table_header, net).ctl_table)
+  sysctl_net_init(net);
+ sysctl_net_table_fixup(net, table);
+ return __register_sysctl_table(&per_net(net_table_header, net), table);
+}
+EXPORT_SYMBOL_GPL(register_net_sysctl_table);
+
+void unregister_net_sysctl_table(struct ctl_table_header *header)
+{
+ return unregister_sysctl_table(header);
+}
+EXPORT_SYMBOL_GPL(unregister_net_sysctl_table);
+#endif
+
+
 #else /* !CONFIG_SYSCTL */
 struct ctl_table_header * register_sysctl_table(ctl_table * table,
       int insert_at_head)
diff --git a/net/core/sysctl_net_core.c b/net/core/sysctl_net_core.c
index 176ad08..76f7a29 100644
--- a/net/core/sysctl_net_core.c
+++ b/net/core/sysctl_net_core.c
@@ -125,3 +125,8 @@ ctl_table core_table[] = {
  },
  { .ctl_name = 0 }
 };
+
+DEFINE_PER_NET(struct ctl_table, multi_core_table[]) = {
+ /* Stub for holding per network namespace sysctls */
```

```
+ {}
+};
diff --git a/net/sysctl_net.c b/net/sysctl_net.c
index cd4eafb..359c163 100644
--- a/net/sysctl_net.c
+++ b/net/sysctl_net.c
@@ -54,3 +54,23 @@ struct ctl_table net_table[] = {
 #endif
  { 0 },
 };
+
+DEFINE_PER_NET(struct ctl_table, multi_net_table[]) = {
+ {
+  .ctl_name = NET_CORE,
+  .procname = "core",
+  .mode  = 0555,
+  .child  = __per_net_base(multi_core_table),
+ },
+ {},
+};
+
+DEFINE_PER_NET(struct ctl_table, net_root_table[]) = {
+ {
+  .ctl_name = CTL_NET,
+  .procname = "net",
+  .mode  = 0555,
+  .child  = __per_net_base(multi_net_table),
+ },
+ {},
+};
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 10/31] net: Make socket creation namespace safe.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:12 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch passes in the namespace a new socket should be created in and has the socket code do the appropriate reference counting. By virtue of this all socket create methods are touched. In addition

the socket create methods are modified so that they will fail if
you attempt to create a socket in a non-default network namespace.

Failing if we attempt to create a socket outside of the default
socket namespace ensures that as we incrementally make the network stack
network namespace aware we will not export functionality that someone
has not audited and made certain is network namespace safe.
Allowing us to partially enable network namespaces before all of the
exotic protocols are supported.

Any protocol layers I have missed will fail to compile because I now
pass an extra parameter into the socket creation code.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 drivers/net/pppoe.c          |    4 ++--
 drivers/net/pppox.c          |    7 +++++--
 include/linux/if_pppox.h     |    2 +-
 include/linux/net.h          |    3 ++-
 include/net/llc_conn.h       |    2 +-
 include/net/sock.h           |    4 +++-
 net/appletalk/ddp.c          |    7 +++++--
 net/atm/common.c             |    4 ++--
 net/atm/common.h             |    2 +-
 net/atm/pvc.c                |    7 +++++--
 net/atm/svc.c                |   11 +++++++----
 net/ax25/af_ax25.c           |    9 ++++++---
 net/bluetooth/af_bluetooth.c |    7 +++++--
 net/bluetooth/bnep/sock.c    |    4 ++--
 net/bluetooth/cmtp/sock.c    |    4 ++--
 net/bluetooth/hci_sock.c     |    4 ++--
 net/bluetooth/hidp/sock.c    |    4 ++--
 net/bluetooth/l2cap.c        |   10 +++++-----
 net/bluetooth/rfcomm/sock.c  |   10 +++++-----
 net/bluetooth/sco.c          |   10 +++++-----
 net/core/sock.c              |    6 ++++--
 net/decnet/af_decnet.c       |   13 +++++++++-----
 net/econet/af_econet.c       |    7 +++++--
 net/ipv4/af_inet.c           |    7 +++++--
 net/ipv6/af_inet6.c          |    7 +++++--
 net/ipx/af_ipx.c             |    7 +++++--
 net/irda/af_irda.c           |   11 +++++++----
 net/key/af_key.c             |    7 +++++--
 net/llc/af_llc.c             |    7 +++++--
 net/llc/llc_conn.c           |    6 ++++--
 net/netlink/af_netlink.c     |   13 +++++++++-----
 net/netrom/af_netrom.c       |    9 ++++++---
 net/packet/af_packet.c       |    7 +++++--
```

```
 net/rose/af_rose.c          |    9 ++++++---
 net/sctp/ipv6.c             |    2 +-
 net/sctp/protocol.c         |    2 +-
 net/socket.c                |    8 ++++----
 net/tipc/socket.c           |    9 ++++++---
 net/unix/af_unix.c          |   13 ++++++++-----
 net/wanrouter/af_wanpipe.c  |   15 +++++++++++------
 net/x25/af_x25.c            |   13 ++++++++-----
 41 files changed, 182 insertions(+), 111 deletions(-)

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index d34fe16..d09334d 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -475,12 +475,12 @@ static struct proto pppoe_sk_proto = {
  * Initialize a new struct sock.
  *
  **********************************************************************/
-static int pppoe_create(struct socket *sock)
+static int pppoe_create(net_t net, struct socket *sock)
 {
  int error = -ENOMEM;
  struct sock *sk;

- sk = sk_alloc(PF_PPPOX, GFP_KERNEL, &pppoe_sk_proto, 1);
+ sk = sk_alloc(net, PF_PPPOX, GFP_KERNEL, &pppoe_sk_proto, 1);
  if (!sk)
   goto out;

diff --git a/drivers/net/pppox.c b/drivers/net/pppox.c
index 9315046..0d5c7bc 100644
--- a/drivers/net/pppox.c
+++ b/drivers/net/pppox.c
@@ -106,10 +106,13 @@ int pppox_ioctl(struct socket *sock, unsigned int cmd, unsigned long
arg)

 EXPORT_SYMBOL(pppox_ioctl);

-static int pppox_create(struct socket *sock, int protocol)
+static int pppox_create(net_t net, struct socket *sock, int protocol)
 {
  int rc = -EPROTOTYPE;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (protocol < 0 || protocol > PX_MAX_PROTO)
   goto out;
```

```
@@ -118,7 +121,7 @@ static int pppox_create(struct socket *sock, int protocol)
      !try_module_get(pppox_protos[protocol]->owner))
   goto out;

- rc = pppox_protos[protocol]->create(sock);
+ rc = pppox_protos[protocol]->create(net, sock);

  module_put(pppox_protos[protocol]->owner);
 out:
diff --git a/include/linux/if_pppox.h b/include/linux/if_pppox.h
index 4fab3d0..f6ffd83 100644
--- a/include/linux/if_pppox.h
+++ b/include/linux/if_pppox.h
@@ -148,7 +148,7 @@ static inline struct sock *sk_pppox(struct pppox_sock *po)
 struct module;

 struct pppox_proto {
- int  (*create)(struct socket *sock);
+ int  (*create)(net_t net, struct socket *sock);
  int  (*ioctl)(struct socket *sock, unsigned int cmd,
      unsigned long arg);
  struct module *owner;
diff --git a/include/linux/net.h b/include/linux/net.h
index f28d8a2..4136768 100644
--- a/include/linux/net.h
+++ b/include/linux/net.h
@@ -19,6 +19,7 @@
 #define _LINUX_NET_H

 #include <linux/wait.h>
+#include <linux/net_namespace_type.h>
 #include <asm/socket.h>

 struct poll_table_struct;
@@ -169,7 +170,7 @@ struct proto_ops {

 struct net_proto_family {
  int  family;
- int  (*create)(struct socket *sock, int protocol);
+ int  (*create)(net_t net, struct socket *sock, int protocol);
  struct module *owner;
 };

diff --git a/include/net/llc_conn.h b/include/net/llc_conn.h
index 00730d2..e4f7104 100644
--- a/include/net/llc_conn.h
+++ b/include/net/llc_conn.h
```

```
@@ -93,7 +93,7 @@ static __inline__ char llc_backlog_type(struct sk_buff *skb)
  return skb->cb[sizeof(skb->cb) - 1];
 }

-extern struct sock *llc_sk_alloc(int family, gfp_t priority,
+extern struct sock *llc_sk_alloc(net_t net, int family, gfp_t priority,
       struct proto *prot);
 extern void llc_sk_free(struct sock *sk);

diff --git a/include/net/sock.h b/include/net/sock.h
index 01a2781..ebcaa7f 100644
--- a/include/net/sock.h
+++ b/include/net/sock.h
@@ -55,6 +55,7 @@
 #include <asm/atomic.h>
 #include <net/dst.h>
 #include <net/checksum.h>
+#include <net/net_namespace.h>

 /*
  * This structure really needs to be cleaned up.
@@ -784,7 +785,7 @@ extern void FASTCALL(release_sock(struct sock *sk));
     SINGLE_DEPTH_NESTING)
 #define bh_unlock_sock(__sk) spin_unlock(&((__sk)->sk_lock.slock))

-extern struct sock  *sk_alloc(int family,
+extern struct sock  *sk_alloc(net_t net, int family,
        gfp_t priority,
        struct proto *prot, int zero_it);
 extern void   sk_free(struct sock *sk);
@@ -1013,6 +1014,7 @@ static inline void sock_copy(struct sock *nsk, const struct sock *osk)
 #endif

  memcpy(nsk, osk, osk->sk_prot->obj_size);
+ get_net(nsk->sk_net);
 #ifdef CONFIG_SECURITY_NETWORK
  nsk->sk_security = sptr;
  security_sk_clone(osk, nsk);
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index 5b8a8ce..e08367b 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -1026,11 +1026,14 @@ static struct proto ddp_proto = {
  * Create a socket. Initialise the socket, blank the addresses
  * set the state.
  */
-static int atalk_create(struct socket *sock, int protocol)
+static int atalk_create(net_t net, struct socket *sock, int protocol)
```

```
 {
  struct sock *sk;
  int rc = -ESOCKTNOSUPPORT;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  /*
   * We permit SOCK_DGRAM and RAW is an extension. It is trivial to do
   * and gives you the full ELAP frame. Should be handy for CAP 8)
@@ -1038,7 +1041,7 @@ static int atalk_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW && sock->type != SOCK_DGRAM)
   goto out;
  rc = -ENOMEM;
- sk = sk_alloc(PF_APPLETALK, GFP_KERNEL, &ddp_proto, 1);
+ sk = sk_alloc(net, PF_APPLETALK, GFP_KERNEL, &ddp_proto, 1);
  if (!sk)
   goto out;
  rc = 0;
diff --git a/net/atm/common.c b/net/atm/common.c
index fbabff4..c4329f0 100644
--- a/net/atm/common.c
+++ b/net/atm/common.c
@@ -132,7 +132,7 @@ static struct proto vcc_proto = {
  .obj_size = sizeof(struct atm_vcc),
 };

-int vcc_create(struct socket *sock, int protocol, int family)
+int vcc_create(net_t net, struct socket *sock, int protocol, int family)
 {
  struct sock *sk;
  struct atm_vcc *vcc;
@@ -140,7 +140,7 @@ int vcc_create(struct socket *sock, int protocol, int family)
  sock->sk = NULL;
  if (sock->type == SOCK_STREAM)
   return -EINVAL;
- sk = sk_alloc(family, GFP_KERNEL, &vcc_proto, 1);
+ sk = sk_alloc(net, family, GFP_KERNEL, &vcc_proto, 1);
  if (!sk)
   return -ENOMEM;
  sock_init_data(sock, sk);
diff --git a/net/atm/common.h b/net/atm/common.h
index a422da7..c7101c7 100644
--- a/net/atm/common.h
+++ b/net/atm/common.h
@@ -10,7 +10,7 @@
 #include <linux/poll.h> /* for poll_table */
```

```
-int vcc_create(struct socket *sock, int protocol, int family);
+int vcc_create(net_t net, struct socket *sock, int protocol, int family);
 int vcc_release(struct socket *sock);
 int vcc_connect(struct socket *sock, int itf, short vpi, int vci);
 int vcc_recvmsg(struct kiocb *iocb, struct socket *sock, struct msghdr *msg,
diff --git a/net/atm/pvc.c b/net/atm/pvc.c
index b2148b4..13bf58e 100644
--- a/net/atm/pvc.c
+++ b/net/atm/pvc.c
@@ -124,10 +124,13 @@ static const struct proto_ops pvc_proto_ops = {
 };


-static int pvc_create(struct socket *sock,int protocol)
+static int pvc_create(net_t net, struct socket *sock,int protocol)
 {
+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  sock->ops = &pvc_proto_ops;
- return vcc_create(sock, protocol, PF_ATMPVC);
+ return vcc_create(net, sock, protocol, PF_ATMPVC);
 }


diff --git a/net/atm/svc.c b/net/atm/svc.c
index 3a180cf..e78d9f7 100644
--- a/net/atm/svc.c
+++ b/net/atm/svc.c
@@ -33,7 +33,7 @@
 #endif


-static int svc_create(struct socket *sock,int protocol);
+static int svc_create(net_t net, struct socket *sock,int protocol);


 /*
@@ -335,7 +335,7 @@ static int svc_accept(struct socket *sock,struct socket *newsock,int flags)

  lock_sock(sk);

- error = svc_create(newsock,0);
+ error = svc_create(sk->sk_net, newsock,0);
  if (error)
   goto out;
```

```
@@ -636,12 +636,15 @@ static const struct proto_ops svc_proto_ops = {
 };


-static int svc_create(struct socket *sock,int protocol)
+static int svc_create(net_t net, struct socket *sock,int protocol)
 {
  int error;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  sock->ops = &svc_proto_ops;
- error = vcc_create(sock, protocol, AF_ATMSVC);
+ error = vcc_create(net, sock, protocol, AF_ATMSVC);
  if (error) return error;
  ATM_SD(sock)->local.sas_family = AF_ATMSVC;
  ATM_SD(sock)->remote.sas_family = AF_ATMSVC;
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index e60af4e..cdbf3f6 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -781,11 +781,14 @@ static struct proto ax25_proto = {
  .obj_size = sizeof(struct sock),
 };

-static int ax25_create(struct socket *sock, int protocol)
+static int ax25_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  ax25_cb *ax25;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  switch (sock->type) {
  case SOCK_DGRAM:
   if (protocol == 0 || protocol == PF_AX25)
@@ -831,7 +834,7 @@ static int ax25_create(struct socket *sock, int protocol)
   return -ESOCKTNOSUPPORT;
  }

- if ((sk = sk_alloc(PF_AX25, GFP_ATOMIC, &ax25_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_AX25, GFP_ATOMIC, &ax25_proto, 1)) == NULL)
   return -ENOMEM;

  ax25 = sk->sk_protinfo = ax25_create_cb();
@@ -856,7 +859,7 @@ struct sock *ax25_make_new(struct sock *osk, struct ax25_dev
```

```
*ax25_dev)
  struct sock *sk;
  ax25_cb *ax25, *oax25;

- if ((sk = sk_alloc(PF_AX25, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
+ if ((sk = sk_alloc(osk->sk_net, PF_AX25, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
   return NULL;

  if ((ax25 = ax25_create_cb()) == NULL) {
diff --git a/net/bluetooth/af_bluetooth.c b/net/bluetooth/af_bluetooth.c
index 67df99e..7110360 100644
--- a/net/bluetooth/af_bluetooth.c
+++ b/net/bluetooth/af_bluetooth.c
@@ -95,10 +95,13 @@ int bt_sock_unregister(int proto)
 }
 EXPORT_SYMBOL(bt_sock_unregister);

-static int bt_sock_create(struct socket *sock, int proto)
+static int bt_sock_create(net_t net, struct socket *sock, int proto)
 {
  int err;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (proto < 0 || proto >= BT_MAX_PROTO)
   return -EINVAL;

@@ -113,7 +116,7 @@ static int bt_sock_create(struct socket *sock, int proto)
  read_lock(&bt_proto_lock);

  if (bt_proto[proto] && try_module_get(bt_proto[proto]->owner)) {
-  err = bt_proto[proto]->create(sock, proto);
+  err = bt_proto[proto]->create(net, sock, proto);
   module_put(bt_proto[proto]->owner);
  }

diff --git a/net/bluetooth/bnep/sock.c b/net/bluetooth/bnep/sock.c
index 5563db1..dc9b1ef 100644
--- a/net/bluetooth/bnep/sock.c
+++ b/net/bluetooth/bnep/sock.c
@@ -205,7 +205,7 @@ static struct proto bnep_proto = {
  .obj_size = sizeof(struct bt_sock)
 };

-static int bnep_sock_create(struct socket *sock, int protocol)
+static int bnep_sock_create(net_t net, struct socket *sock, int protocol)
 {
```

```
  struct sock *sk;

@@ -214,7 +214,7 @@ static int bnep_sock_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW)
   return -ESOCKTNOSUPPORT;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &bnep_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &bnep_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/cmtp/sock.c b/net/bluetooth/cmtp/sock.c
index 53295d3..107dbfe 100644
--- a/net/bluetooth/cmtp/sock.c
+++ b/net/bluetooth/cmtp/sock.c
@@ -196,7 +196,7 @@ static struct proto cmtp_proto = {
  .obj_size = sizeof(struct bt_sock)
 };

-static int cmtp_sock_create(struct socket *sock, int protocol)
+static int cmtp_sock_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -205,7 +205,7 @@ static int cmtp_sock_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW)
   return -ESOCKTNOSUPPORT;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &cmtp_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &cmtp_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/hci_sock.c b/net/bluetooth/hci_sock.c
index dbf98c4..3a15a31 100644
--- a/net/bluetooth/hci_sock.c
+++ b/net/bluetooth/hci_sock.c
@@ -610,7 +610,7 @@ static struct proto hci_sk_proto = {
  .obj_size = sizeof(struct hci_pinfo)
 };

-static int hci_sock_create(struct socket *sock, int protocol)
+static int hci_sock_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -621,7 +621,7 @@ static int hci_sock_create(struct socket *sock, int protocol)
```

```
  sock->ops = &hci_sock_ops;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &hci_sk_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &hci_sk_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/hidp/sock.c b/net/bluetooth/hidp/sock.c
index 407fba4..647f85e 100644
--- a/net/bluetooth/hidp/sock.c
+++ b/net/bluetooth/hidp/sock.c
@@ -247,7 +247,7 @@ static struct proto hidp_proto = {
  .obj_size = sizeof(struct bt_sock)
 };

-static int hidp_sock_create(struct socket *sock, int protocol)
+static int hidp_sock_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -256,7 +256,7 @@ static int hidp_sock_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW)
   return -ESOCKTNOSUPPORT;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &hidp_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &hidp_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/l2cap.c b/net/bluetooth/l2cap.c
index 29a8fa4..13e9b5b 100644
--- a/net/bluetooth/l2cap.c
+++ b/net/bluetooth/l2cap.c
@@ -517,11 +517,11 @@ static struct proto l2cap_proto = {
  .obj_size = sizeof(struct l2cap_pinfo)
 };

-static struct sock *l2cap_sock_alloc(struct socket *sock, int proto, gfp_t prio)
+static struct sock *l2cap_sock_alloc(net_t net, struct socket *sock, int proto, gfp_t prio)
 {
  struct sock *sk;

- sk = sk_alloc(PF_BLUETOOTH, prio, &l2cap_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, prio, &l2cap_proto, 1);
  if (!sk)
   return NULL;

@@ -542,7 +542,7 @@ static struct sock *l2cap_sock_alloc(struct socket *sock, int proto, gfp_t
```

```
prio)
  return sk;
 }

-static int l2cap_sock_create(struct socket *sock, int protocol)
+static int l2cap_sock_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -559,7 +559,7 @@ static int l2cap_sock_create(struct socket *sock, int protocol)

  sock->ops = &l2cap_sock_ops;

- sk = l2cap_sock_alloc(sock, protocol, GFP_ATOMIC);
+ sk = l2cap_sock_alloc(net, sock, protocol, GFP_ATOMIC);
  if (!sk)
   return -ENOMEM;

@@ -1412,7 +1412,7 @@ static inline int l2cap_connect_req(struct l2cap_conn *conn, struct
l2cap_cmd_hd
   goto response;
  }

- sk = l2cap_sock_alloc(NULL, BTPROTO_L2CAP, GFP_ATOMIC);
+ sk = l2cap_sock_alloc(parent->sk_net, NULL, BTPROTO_L2CAP, GFP_ATOMIC);
  if (!sk)
   goto response;

diff --git a/net/bluetooth/rfcomm/sock.c b/net/bluetooth/rfcomm/sock.c
index cb7e855..12ff829 100644
--- a/net/bluetooth/rfcomm/sock.c
+++ b/net/bluetooth/rfcomm/sock.c
@@ -282,12 +282,12 @@ static struct proto rfcomm_proto = {
  .obj_size = sizeof(struct rfcomm_pinfo)
 };

-static struct sock *rfcomm_sock_alloc(struct socket *sock, int proto, gfp_t prio)
+static struct sock *rfcomm_sock_alloc(net_t net, struct socket *sock, int proto, gfp_t prio)
 {
  struct rfcomm_dlc *d;
  struct sock *sk;

- sk = sk_alloc(PF_BLUETOOTH, prio, &rfcomm_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, prio, &rfcomm_proto, 1);
  if (!sk)
   return NULL;

@@ -323,7 +323,7 @@ static struct sock *rfcomm_sock_alloc(struct socket *sock, int proto, gfp_t
```

```
prio
  return sk;
 }

-static int rfcomm_sock_create(struct socket *sock, int protocol)
+static int rfcomm_sock_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -336,7 +336,7 @@ static int rfcomm_sock_create(struct socket *sock, int protocol)

  sock->ops = &rfcomm_sock_ops;

- sk = rfcomm_sock_alloc(sock, protocol, GFP_ATOMIC);
+ sk = rfcomm_sock_alloc(net, sock, protocol, GFP_ATOMIC);
  if (!sk)
   return -ENOMEM;

@@ -868,7 +868,7 @@ int rfcomm_connect_ind(struct rfcomm_session *s, u8 channel, struct
rfcomm_dlc *
   goto done;
  }

- sk = rfcomm_sock_alloc(NULL, BTPROTO_RFCOMM, GFP_ATOMIC);
+ sk = rfcomm_sock_alloc(parent->sk_net, NULL, BTPROTO_RFCOMM, GFP_ATOMIC);
  if (!sk)
   goto done;

diff --git a/net/bluetooth/sco.c b/net/bluetooth/sco.c
index 5d13d4f..6d424ea 100644
--- a/net/bluetooth/sco.c
+++ b/net/bluetooth/sco.c
@@ -414,11 +414,11 @@ static struct proto sco_proto = {
  .obj_size = sizeof(struct sco_pinfo)
 };

-static struct sock *sco_sock_alloc(struct socket *sock, int proto, gfp_t prio)
+static struct sock *sco_sock_alloc(net_t net, struct socket *sock, int proto, gfp_t prio)
 {
  struct sock *sk;

- sk = sk_alloc(PF_BLUETOOTH, prio, &sco_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, prio, &sco_proto, 1);
  if (!sk)
   return NULL;

@@ -439,7 +439,7 @@ static struct sock *sco_sock_alloc(struct socket *sock, int proto, gfp_t
prio)
```

```
 return sk;
}

-static int sco_sock_create(struct socket *sock, int protocol)
+static int sco_sock_create(net_t net, struct socket *sock, int protocol)
{
 struct sock *sk;

@@ -452,7 +452,7 @@ static int sco_sock_create(struct socket *sock, int protocol)

 sock->ops = &sco_sock_ops;

- sk = sco_sock_alloc(sock, protocol, GFP_ATOMIC);
+ sk = sco_sock_alloc(net, sock, protocol, GFP_ATOMIC);
 if (!sk)
  return -ENOMEM;

@@ -807,7 +807,7 @@ static void sco_conn_ready(struct sco_conn *conn)

  bh_lock_sock(parent);

-  sk = sco_sock_alloc(NULL, BTPROTO_SCO, GFP_ATOMIC);
+  sk = sco_sock_alloc(parent->sk_net, NULL, BTPROTO_SCO, GFP_ATOMIC);
  if (!sk) {
   bh_unlock_sock(parent);
   goto done;
diff --git a/net/core/sock.c b/net/core/sock.c
index 5555364..e42f7df 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -825,7 +825,7 @@ static void inline sock_lock_init(struct sock *sk)
 * @prot: struct proto associated with this new sock instance
 * @zero_it: if we should zero the newly allocated sock
 */
-struct sock *sk_alloc(int family, gfp_t priority,
+struct sock *sk_alloc(net_t net, int family, gfp_t priority,
        struct proto *prot, int zero_it)
{
 struct sock *sk = NULL;
@@ -846,6 +846,7 @@ struct sock *sk_alloc(int family, gfp_t priority,
    */
   sk->sk_prot = sk->sk_prot_creator = prot;
   sock_lock_init(sk);
+   sk->sk_net = get_net(net);
  }

  if (security_sk_alloc(sk, family, priority))
@@ -885,6 +886,7 @@ void sk_free(struct sock *sk)
```

```
        __FUNCTION__, atomic_read(&sk->sk_omem_alloc));

  security_sk_free(sk);
+ put_net(sk->sk_net);
  if (sk->sk_prot_creator->slab != NULL)
   kmem_cache_free(sk->sk_prot_creator->slab, sk);
  else
@@ -894,7 +896,7 @@ void sk_free(struct sock *sk)

 struct sock *sk_clone(const struct sock *sk, const gfp_t priority)
 {
- struct sock *newsk = sk_alloc(sk->sk_family, priority, sk->sk_prot, 0);
+ struct sock *newsk = sk_alloc(sk->sk_net, sk->sk_family, priority, sk->sk_prot, 0);

  if (newsk != NULL) {
   struct sk_filter *filter;
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index 77cd802..f1553fa 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -471,10 +471,10 @@ static struct proto dn_proto = {
  .obj_size  = sizeof(struct dn_sock),
 };

-static struct sock *dn_alloc_sock(struct socket *sock, gfp_t gfp)
+static struct sock *dn_alloc_sock(net_t net, struct socket *sock, gfp_t gfp)
 {
  struct dn_scp *scp;
- struct sock *sk = sk_alloc(PF_DECnet, gfp, &dn_proto, 1);
+ struct sock *sk = sk_alloc(net, PF_DECnet, gfp, &dn_proto, 1);

  if  (!sk)
   goto out;
@@ -675,10 +675,13 @@ char *dn_addr2asc(__u16 addr, char *buf)



-static int dn_create(struct socket *sock, int protocol)
+static int dn_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  switch(sock->type) {
   case SOCK_SEQPACKET:
    if (protocol != DNPROTO_NSP)
```

```
@@ -691,7 +694,7 @@ static int dn_create(struct socket *sock, int protocol)
  }


- if ((sk = dn_alloc_sock(sock, GFP_KERNEL)) == NULL)
+ if ((sk = dn_alloc_sock(net, sock, GFP_KERNEL)) == NULL)
   return -ENOBUFS;

  sk->sk_protocol = protocol;
@@ -1088,7 +1091,7 @@ static int dn_accept(struct socket *sock, struct socket *newsock, int
flags)

  cb = DN_SKB_CB(skb);
  sk->sk_ack_backlog--;
- newsk = dn_alloc_sock(newsock, sk->sk_allocation);
+ newsk = dn_alloc_sock(sk->sk_net, newsock, sk->sk_allocation);
  if (newsk == NULL) {
   release_sock(sk);
   kfree_skb(skb);
diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index 4d66aac..a0b3fc5 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -609,12 +609,15 @@ static struct proto econet_proto = {
  * Create an Econet socket
  */

-static int econet_create(struct socket *sock, int protocol)
+static int econet_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct econet_sock *eo;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  /* Econet only provides datagram services. */
  if (sock->type != SOCK_DGRAM)
   return -ESOCKTNOSUPPORT;
@@ -622,7 +625,7 @@ static int econet_create(struct socket *sock, int protocol)
  sock->state = SS_UNCONNECTED;

  err = -ENOBUFS;
- sk = sk_alloc(PF_ECONET, GFP_KERNEL, &econet_proto, 1);
+ sk = sk_alloc(net, PF_ECONET, GFP_KERNEL, &econet_proto, 1);
  if (sk == NULL)
   goto out;
```

```
diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c
index 8640096..cb07cb6 100644
--- a/net/ipv4/af_inet.c
+++ b/net/ipv4/af_inet.c
@@ -221,7 +221,7 @@ out:
  * Create an inet socket.
  */

-static int inet_create(struct socket *sock, int protocol)
+static int inet_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct list_head *p;
@@ -233,6 +233,9 @@ static int inet_create(struct socket *sock, int protocol)
  int try_loading_module = 0;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  sock->state = SS_UNCONNECTED;

  /* Look for the requested type/protocol pair. */
@@ -295,7 +298,7 @@ lookup_protocol:
  BUG_TRAP(answer_prot->slab != NULL);

  err = -ENOBUFS;
- sk = sk_alloc(PF_INET, GFP_KERNEL, answer_prot, 1);
+ sk = sk_alloc(net, PF_INET, GFP_KERNEL, answer_prot, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/ipv6/af_inet6.c b/net/ipv6/af_inet6.c
index 0e0e426..00bd55a 100644
--- a/net/ipv6/af_inet6.c
+++ b/net/ipv6/af_inet6.c
@@ -86,7 +86,7 @@ static __inline__ struct ipv6_pinfo *inet6_sk_generic(struct sock *sk)
  return (struct ipv6_pinfo *)(((u8 *)sk) + offset);
 }

-static int inet6_create(struct socket *sock, int protocol)
+static int inet6_create(net_t net, struct socket *sock, int protocol)
 {
  struct inet_sock *inet;
  struct ipv6_pinfo *np;
@@ -99,6 +99,9 @@ static int inet6_create(struct socket *sock, int protocol)
  int try_loading_module = 0;
```

```
  int err;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  /* Look for the requested type/protocol pair. */
  answer = NULL;
 lookup_protocol:
@@ -159,7 +162,7 @@ lookup_protocol:
  BUG_TRAP(answer_prot->slab != NULL);

  err = -ENOBUFS;
- sk = sk_alloc(PF_INET6, GFP_KERNEL, answer_prot, 1);
+ sk = sk_alloc(net, PF_INET6, GFP_KERNEL, answer_prot, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index 76c6615..2ec4a3c 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -1358,11 +1358,14 @@ static struct proto ipx_proto = {
  .obj_size = sizeof(struct ipx_sock),
 };

-static int ipx_create(struct socket *sock, int protocol)
+static int ipx_create(net_t net, struct socket *sock, int protocol)
 {
  int rc = -ESOCKTNOSUPPORT;
  struct sock *sk;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  /*
   * SPX support is not anymore in the kernel sources. If you want to
   * ressurrect it, completing it and making it understand shared skbs,
@@ -1373,7 +1376,7 @@ static int ipx_create(struct socket *sock, int protocol)
   goto out;

         rc = -ENOMEM;
- sk = sk_alloc(PF_IPX, GFP_KERNEL, &ipx_proto, 1);
+ sk = sk_alloc(net, PF_IPX, GFP_KERNEL, &ipx_proto, 1);
  if (!sk)
   goto out;
 #ifdef IPX_REFCNT_DEBUG
diff --git a/net/irda/af_irda.c b/net/irda/af_irda.c
index 7e1aea8..e3344c3 100644
```

```
--- a/net/irda/af_irda.c
+++ b/net/irda/af_irda.c
@@ -60,7 +60,7 @@

 #include <net/irda/af_irda.h>

-static int irda_create(struct socket *sock, int protocol);
+static int irda_create(net_t net, struct socket *sock, int protocol);

 static const struct proto_ops irda_stream_ops;
 static const struct proto_ops irda_seqpacket_ops;
@@ -844,7 +844,7 @@ static int irda_accept(struct socket *sock, struct socket *newsock, int
flags)

  IRDA_ASSERT(self != NULL, return -1;);

- err = irda_create(newsock, sk->sk_protocol);
+ err = irda_create(sk->sk_net, newsock, sk->sk_protocol);
  if (err)
   return err;

@@ -1085,13 +1085,16 @@ static struct proto irda_proto = {
  *    Create IrDA socket
  *
  */
-static int irda_create(struct socket *sock, int protocol)
+static int irda_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct irda_sock *self;

  IRDA_DEBUG(2, "%s()\n", __FUNCTION__);

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  /* Check for valid socket type */
  switch (sock->type) {
  case SOCK_STREAM:     /* For TTP connections with SAR disabled */
@@ -1103,7 +1106,7 @@ static int irda_create(struct socket *sock, int protocol)
  }

  /* Allocate networking socket */
- sk = sk_alloc(PF_IRDA, GFP_ATOMIC, &irda_proto, 1);
+ sk = sk_alloc(net, PF_IRDA, GFP_ATOMIC, &irda_proto, 1);
  if (sk == NULL)
   return -ENOMEM;
```

```
diff --git a/net/key/af_key.c b/net/key/af_key.c
index c79f9c4..244ab5b 100644
--- a/net/key/af_key.c
+++ b/net/key/af_key.c
@@ -137,11 +137,14 @@ static struct proto key_proto = {
  .obj_size = sizeof(struct pfkey_sock),
 };

-static int pfkey_create(struct socket *sock, int protocol)
+static int pfkey_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (!capable(CAP_NET_ADMIN))
   return -EPERM;
  if (sock->type != SOCK_RAW)
@@ -150,7 +153,7 @@ static int pfkey_create(struct socket *sock, int protocol)
   return -EPROTONOSUPPORT;

  err = -ENOMEM;
- sk = sk_alloc(PF_KEY, GFP_KERNEL, &key_proto, 1);
+ sk = sk_alloc(net, PF_KEY, GFP_KERNEL, &key_proto, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/llc/af_llc.c b/net/llc/af_llc.c
index 190bb3e..6bc0fff 100644
--- a/net/llc/af_llc.c
+++ b/net/llc/af_llc.c
@@ -150,14 +150,17 @@ static struct proto llc_proto = {
  * socket type we have available.
  * Returns 0 upon success, negative upon failure.
  */
-static int llc_ui_create(struct socket *sock, int protocol)
+static int llc_ui_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  int rc = -ESOCKTNOSUPPORT;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (likely(sock->type == SOCK_DGRAM || sock->type == SOCK_STREAM)) {
   rc = -ENOMEM;
```

```
-  sk = llc_sk_alloc(PF_LLC, GFP_KERNEL, &llc_proto);
+  sk = llc_sk_alloc(net, PF_LLC, GFP_KERNEL, &llc_proto);
   if (sk) {
    rc = 0;
    llc_ui_sk_init(sock, sk);
diff --git a/net/llc/llc_conn.c b/net/llc/llc_conn.c
index c761c15..49f8703 100644
--- a/net/llc/llc_conn.c
+++ b/net/llc/llc_conn.c
@@ -700,7 +700,7 @@ static struct sock *llc_create_incoming_sock(struct sock *sk,
          struct llc_addr *saddr,
          struct llc_addr *daddr)
 {
- struct sock *newsk = llc_sk_alloc(sk->sk_family, GFP_ATOMIC,
+ struct sock *newsk = llc_sk_alloc(sk->sk_net, sk->sk_family, GFP_ATOMIC,
        sk->sk_prot);
  struct llc_sock *newllc, *llc = llc_sk(sk);

@@ -867,9 +867,9 @@ static void llc_sk_init(struct sock* sk)
  * Allocates a LLC sock and initializes it. Returns the new LLC sock
  * or %NULL if there's no memory available for one
  */
-struct sock *llc_sk_alloc(int family, gfp_t priority, struct proto *prot)
+struct sock *llc_sk_alloc(net_t net, int family, gfp_t priority, struct proto *prot)
 {
- struct sock *sk = sk_alloc(family, priority, prot, 1);
+ struct sock *sk = sk_alloc(net, family, priority, prot, 1);

  if (!sk)
   goto out;
diff --git a/net/netlink/af_netlink.c b/net/netlink/af_netlink.c
index 3c00f48..7433e71 100644
--- a/net/netlink/af_netlink.c
+++ b/net/netlink/af_netlink.c
@@ -371,14 +371,14 @@ static struct proto netlink_proto = {
  .obj_size = sizeof(struct netlink_sock),
 };

-static int __netlink_create(struct socket *sock, int protocol)
+static int __netlink_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct netlink_sock *nlk;

  sock->ops = &netlink_ops;

- sk = sk_alloc(PF_NETLINK, GFP_KERNEL, &netlink_proto, 1);
+ sk = sk_alloc(net, PF_NETLINK, GFP_KERNEL, &netlink_proto, 1);
```

```
  if (!sk)
   return -ENOMEM;

@@ -393,13 +393,16 @@ static int __netlink_create(struct socket *sock, int protocol)
  return 0;
 }

-static int netlink_create(struct socket *sock, int protocol)
+static int netlink_create(net_t net, struct socket *sock, int protocol)
 {
  struct module *module = NULL;
  struct netlink_sock *nlk;
  unsigned int groups;
  int err = 0;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  sock->state = SS_UNCONNECTED;

  if (sock->type != SOCK_RAW && sock->type != SOCK_DGRAM)
@@ -422,7 +425,7 @@ static int netlink_create(struct socket *sock, int protocol)
  groups = nl_table[protocol].groups;
  netlink_unlock_table();

- if ((err = __netlink_create(sock, protocol)) < 0)
+ if ((err = __netlink_create(net, sock, protocol)) < 0)
   goto out_module;

  nlk = nlk_sk(sock->sk);
@@ -1281,7 +1284,7 @@ netlink_kernel_create(int unit, unsigned int groups,
  if (sock_create_lite(PF_NETLINK, SOCK_DGRAM, unit, &sock))
   return NULL;

- if (__netlink_create(sock, unit) < 0)
+ if (__netlink_create(init_net(), sock, unit) < 0)
   goto out_sock_release;

  if (groups < 32)
diff --git a/net/netrom/af_netrom.c b/net/netrom/af_netrom.c
index 601d58c..3fa3f1a 100644
--- a/net/netrom/af_netrom.c
+++ b/net/netrom/af_netrom.c
@@ -409,15 +409,18 @@ static struct proto nr_proto = {
  .obj_size = sizeof(struct nr_sock),
 };

-static int nr_create(struct socket *sock, int protocol)
```

```
+static int nr_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct nr_sock *nr;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_SEQPACKET || protocol != 0)
   return -ESOCKTNOSUPPORT;

- if ((sk = sk_alloc(PF_NETROM, GFP_ATOMIC, &nr_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_NETROM, GFP_ATOMIC, &nr_proto, 1)) == NULL)
   return -ENOMEM;

  nr = nr_sk(sk);
@@ -459,7 +462,7 @@ static struct sock *nr_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_SEQPACKET)
   return NULL;

- if ((sk = sk_alloc(PF_NETROM, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
+ if ((sk = sk_alloc(osk->sk_net, PF_NETROM, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
   return NULL;

  nr = nr_sk(sk);
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index 04e295a..ca371ea 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -981,13 +981,16 @@ static struct proto packet_proto = {
  * Create a packet of type SOCK_PACKET.
  */

-static int packet_create(struct socket *sock, int protocol)
+static int packet_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct packet_sock *po;
  __be16 proto = (__force __be16)protocol; /* weird, but documented */
  int err;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (!capable(CAP_NET_RAW))
   return -EPERM;
  if (sock->type != SOCK_DGRAM && sock->type != SOCK_RAW
@@ -1000,7 +1003,7 @@ static int packet_create(struct socket *sock, int protocol)
```

```
  sock->state = SS_UNCONNECTED;

  err = -ENOBUFS;
- sk = sk_alloc(PF_PACKET, GFP_KERNEL, &packet_proto, 1);
+ sk = sk_alloc(net, PF_PACKET, GFP_KERNEL, &packet_proto, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index 5532340..7d5e593 100644
--- a/net/rose/af_rose.c
+++ b/net/rose/af_rose.c
@@ -499,15 +499,18 @@ static struct proto rose_proto = {
  .obj_size = sizeof(struct rose_sock),
 };

-static int rose_create(struct socket *sock, int protocol)
+static int rose_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct rose_sock *rose;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_SEQPACKET || protocol != 0)
   return -ESOCKTNOSUPPORT;

- if ((sk = sk_alloc(PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
   return -ENOMEM;

  rose = rose_sk(sk);
@@ -545,7 +548,7 @@ static struct sock *rose_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_SEQPACKET)
   return NULL;

- if ((sk = sk_alloc(PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
+ if ((sk = sk_alloc(osk->sk_net, PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
   return NULL;

  rose = rose_sk(sk);
diff --git a/net/sctp/ipv6.c b/net/sctp/ipv6.c
index ef36be0..0217546 100644
--- a/net/sctp/ipv6.c
+++ b/net/sctp/ipv6.c
@@ -622,7 +622,7 @@ static struct sock *sctp_v6_create_accept_sk(struct sock *sk,
  struct ipv6_pinfo *newnp, *np = inet6_sk(sk);
```

```
  struct sctp6_sock *newsctp6sk;

- newsk = sk_alloc(PF_INET6, GFP_KERNEL, sk->sk_prot, 1);
+ newsk = sk_alloc(sk->sk_net, PF_INET6, GFP_KERNEL, sk->sk_prot, 1);
  if (!newsk)
   goto out;

diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index ea94951..9461a10 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -540,7 +540,7 @@ static struct sock *sctp_v4_create_accept_sk(struct sock *sk,
 {
  struct inet_sock *inet = inet_sk(sk);
  struct inet_sock *newinet;
- struct sock *newsk = sk_alloc(PF_INET, GFP_KERNEL, sk->sk_prot, 1);
+ struct sock *newsk = sk_alloc(sk->sk_net, PF_INET, GFP_KERNEL, sk->sk_prot, 1);

  if (!newsk)
   goto out;
diff --git a/net/socket.c b/net/socket.c
index 4e39631..0d0c92b 100644
--- a/net/socket.c
+++ b/net/socket.c
@@ -1053,7 +1053,7 @@ call_kill:
  return 0;
 }

-static int __sock_create(int family, int type, int protocol,
+static int __sock_create(net_t net, int family, int type, int protocol,
     struct socket **res, int kern)
 {
  int err;
@@ -1129,7 +1129,7 @@ static int __sock_create(int family, int type, int protocol,
  /* Now protected by module ref count */
  rcu_read_unlock();

- err = pf->create(sock, protocol);
+ err = pf->create(net, sock, protocol);
  if (err < 0)
   goto out_module_put;

@@ -1168,12 +1168,12 @@ out_release:

 int sock_create(int family, int type, int protocol, struct socket **res)
 {
- return __sock_create(family, type, protocol, res, 0);
+ return __sock_create(current->nsproxy->net_ns, family, type, protocol, res, 0);
```

```
 }

 int sock_create_kern(int family, int type, int protocol, struct socket **res)
 {
- return __sock_create(family, type, protocol, res, 1);
+ return __sock_create(init_net(), family, type, protocol, res, 1);
 }

 asmlinkage long sys_socket(int family, int type, int protocol)
diff --git a/net/tipc/socket.c b/net/tipc/socket.c
index 2a6a5a6..cf02a0c 100644
--- a/net/tipc/socket.c
+++ b/net/tipc/socket.c
@@ -162,13 +162,16 @@ static void advance_queue(struct tipc_sock *tsock)
  *
  * Returns 0 on success, errno otherwise
  */
-static int tipc_create(struct socket *sock, int protocol)
+static int tipc_create(net_t net, struct socket *sock, int protocol)
 {
  struct tipc_sock *tsock;
  struct tipc_port *port;
  struct sock *sk;
        u32 ref;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (unlikely(protocol != 0))
   return -EPROTONOSUPPORT;

@@ -198,7 +201,7 @@ static int tipc_create(struct socket *sock, int protocol)
   return -EPROTOTYPE;
  }

- sk = sk_alloc(AF_TIPC, GFP_KERNEL, &tipc_proto, 1);
+ sk = sk_alloc(net, AF_TIPC, GFP_KERNEL, &tipc_proto, 1);
  if (!sk) {
   tipc_deleteport(ref);
   return -ENOMEM;
@@ -1363,7 +1366,7 @@ static int accept(struct socket *sock, struct socket *newsock, int flags)
  }
  buf = skb_peek(&sock->sk->sk_receive_queue);

- res = tipc_create(newsock, 0);
+ res = tipc_create(sock->sk->sk_net, newsock, 0);
  if (!res) {
   struct tipc_sock *new_tsock = tipc_sk(newsock->sk);
```

```
   struct tipc_portid id;
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 30855e1..8015a03 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -569,7 +569,7 @@ static struct proto unix_proto = {
  */
 static struct lock_class_key af_unix_sk_receive_queue_lock_key;

-static struct sock * unix_create1(struct socket *sock)
+static struct sock * unix_create1(net_t net, struct socket *sock)
 {
  struct sock *sk = NULL;
  struct unix_sock *u;
@@ -577,7 +577,7 @@ static struct sock * unix_create1(struct socket *sock)
  if (atomic_read(&unix_nr_socks) >= 2*get_max_files())
   goto out;

- sk = sk_alloc(PF_UNIX, GFP_KERNEL, &unix_proto, 1);
+ sk = sk_alloc(net, PF_UNIX, GFP_KERNEL, &unix_proto, 1);
  if (!sk)
   goto out;

@@ -602,8 +602,11 @@ out:
  return sk;
 }

-static int unix_create(struct socket *sock, int protocol)
+static int unix_create(net_t net, struct socket *sock, int protocol)
 {
+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (protocol && protocol != PF_UNIX)
   return -EPROTONOSUPPORT;

@@ -629,7 +632,7 @@ static int unix_create(struct socket *sock, int protocol)
   return -ESOCKTNOSUPPORT;
  }

- return unix_create1(sock) ? 0 : -ENOMEM;
+ return unix_create1(net, sock) ? 0 : -ENOMEM;
 }

 static int unix_release(struct socket *sock)
@@ -980,7 +983,7 @@ static int unix_stream_connect(struct socket *sock, struct sockaddr
*uaddr,
  err = -ENOMEM;
```

```
  /* create new sock for complete connection */
- newsk = unix_create1(NULL);
+ newsk = unix_create1(sk->sk_net, NULL);
  if (newsk == NULL)
   goto out;

diff --git a/net/wanrouter/af_wanpipe.c b/net/wanrouter/af_wanpipe.c
index c205973..542c737 100644
--- a/net/wanrouter/af_wanpipe.c
+++ b/net/wanrouter/af_wanpipe.c
@@ -191,7 +191,7 @@ struct net_device *wanpipe_find_free_dev(sdla_t *card);
 static void wanpipe_unlink_card (struct sock *);
 static int wanpipe_link_card (struct sock *);
 static struct sock *wanpipe_make_new(struct sock *);
-static struct sock *wanpipe_alloc_socket(void);
+static struct sock *wanpipe_alloc_socket(net_t net);
 static inline int get_atomic_device(struct net_device *dev);
 static int wanpipe_exec_cmd(struct sock *, int, unsigned int);
 static int get_ioctl_cmd (struct sock *, void *);
@@ -455,7 +455,7 @@ static struct sock *wanpipe_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_RAW)
   return NULL;

- if ((sk = wanpipe_alloc_socket()) == NULL)
+ if ((sk = wanpipe_alloc_socket(osk->sk_net)) == NULL)
   return NULL;

  sk->sk_type = osk->sk_type;
@@ -498,12 +498,12 @@ static struct proto wanpipe_proto = {
  *
  *============================================================*/

-static struct sock *wanpipe_alloc_socket(void)
+static struct sock *wanpipe_alloc_socket(net_t net)
 {
  struct sock *sk;
  struct wanpipe_opt *wan_opt;

- if ((sk = sk_alloc(PF_WANPIPE, GFP_ATOMIC, &wanpipe_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_WANPIPE, GFP_ATOMIC, &wanpipe_proto, 1)) == NULL)
   return NULL;

  if ((wan_opt = kzalloc(sizeof(struct wanpipe_opt), GFP_ATOMIC)) == NULL) {
@@ -1498,10 +1498,13 @@ struct net_device *wanpipe_find_free_dev(sdla_t *card)
  *      Crates AF_WANPIPE socket.
  *============================================================*/
```

```
-static int wanpipe_create(struct socket *sock, int protocol)
+static int wanpipe_create(net_t net, socket *sock, int protocol)
 {
  struct sock *sk;

+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  //FIXME: This checks for root user, SECURITY ?
  //if (!capable(CAP_NET_RAW))
  // return -EPERM;
@@ -1511,7 +1514,7 @@ static int wanpipe_create(struct socket *sock, int protocol)

  sock->state = SS_UNCONNECTED;

- if ((sk = wanpipe_alloc_socket()) == NULL)
+ if ((sk = wanpipe_alloc_socket(net)) == NULL)
   return -ENOBUFS;

  sk->sk_reuse = 1;
diff --git a/net/x25/af_x25.c b/net/x25/af_x25.c
index b5c80b1..6602a34 100644
--- a/net/x25/af_x25.c
+++ b/net/x25/af_x25.c
@@ -465,10 +465,10 @@ static struct proto x25_proto = {
  .obj_size = sizeof(struct x25_sock),
 };

-static struct sock *x25_alloc_socket(void)
+static struct sock *x25_alloc_socket(net_t net)
 {
  struct x25_sock *x25;
- struct sock *sk = sk_alloc(AF_X25, GFP_ATOMIC, &x25_proto, 1);
+ struct sock *sk = sk_alloc(net, AF_X25, GFP_ATOMIC, &x25_proto, 1);

  if (!sk)
   goto out;
@@ -484,17 +484,20 @@ out:
  return sk;
 }

-static int x25_create(struct socket *sock, int protocol)
+static int x25_create(net_t net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct x25_sock *x25;
  int rc = -ESOCKTNOSUPPORT;
```

```
+ if (!net_eq(net, init_net()))
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_SEQPACKET || protocol)
   goto out;

  rc = -ENOMEM;
- if ((sk = x25_alloc_socket()) == NULL)
+ if ((sk = x25_alloc_socket(net)) == NULL)
   goto out;

  x25 = x25_sk(sk);
@@ -542,7 +545,7 @@ static struct sock *x25_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_SEQPACKET)
   goto out;

- if ((sk = x25_alloc_socket()) == NULL)
+ if ((sk = x25_alloc_socket(osk->sk_net)) == NULL)
   goto out;

  x25 = x25_sk(sk);
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 11/31] net: Initialize the network namespace of network devices.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:13 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Except for carefully selected pseudo devices all network
interfaces should start out in the initial network namespace.
Ultimately it will be register_netdev that examines what
dev->nd_net is set to and places a device in a network namespace.

This patch modifies alloc_netdev to initialize the network
namespace a device is in with the initial network namespace.
This gets it right for the vast majority of devices so their
drivers need not be modified and for those few pseudo devices
that need something different they can change this parameter
before calling register_netdevice.

The network namespace parameter on a network device is not
reference counted as the devices are inside of a network namespace
and cannot remain in that namespace past the lifetime of the
network namespace.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/net/loopback.c | 1 +
 net/core/dev.c | 1 +
 2 files changed, 2 insertions(+), 0 deletions(-)

diff --git a/drivers/net/loopback.c b/drivers/net/loopback.c
index 2b739fd..22b672d 100644
--- a/drivers/net/loopback.c
+++ b/drivers/net/loopback.c
@@ -231,6 +231,7 @@ struct net_device loopback_dev = {
 /* Setup and register the loopback device. */
 static int __init loopback_init(void)
 {
+ loopback_dev.nd_net = init_net();
  return register_netdev(&loopback_dev);
 };

diff --git a/net/core/dev.c b/net/core/dev.c
index 90e4c0e..a3ee150 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -3192,6 +3192,7 @@ struct net_device *alloc_netdev(int sizeof_priv, const char *name,
  dev = (struct net_device *)
   (((long)p + NETDEV_ALIGN_CONST) & ~NETDEV_ALIGN_CONST);
  dev->padded = (char *)dev - (char *)p;
+ dev->nd_net = init_net();

  if (sizeof_priv)
   dev->priv = netdev_priv(dev);
--
1.4.4.1.g278f

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 12/31] net: Make packet reception network namespace safe
Posted by ebiederm on Thu, 25 Jan 2007 19:00:14 GMT

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch modifies every packet receive function
registered with dev_add_pack() to drop packets if they
are not from the initial network namespace, in addition
to ensure consistency of argument passing the unnecessary
device parameter is removed.

This should ensure that the various network stacks do
not receive packets in a anything but the initial network
namespace until the code has been converted and is ready
for them.

Anything I may have missed will generate a compiler error,
as the function protype has changed, preventing us from
overlooking something by accident.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/block/aoe/aoenet.c | 7 ++++++-
drivers/net/bonding/bond_3ad.c | 7 ++++++-
drivers/net/bonding/bond_3ad.h | 2 +-
drivers/net/bonding/bond_alb.c | 6 +++++-
drivers/net/bonding/bond_main.c | 6 +++++-
drivers/net/hamradio/bpqether.c | 8 ++++++--
drivers/net/pppoe.c | 8 ++++++--
drivers/net/wan/hdlc.c | 10 +++++++++-
drivers/net/wan/lapbether.c | 6 +++++-
drivers/net/wan/syncppp.c | 14 ++++++++++----
include/linux/netdevice.h | 1 -
include/net/ax25.h | 2 +-
include/net/datalink.h | 2 +-
include/net/ip.h | 2 +-
include/net/ipv6.h | 1 -
include/net/llc.h | 4 +---
include/net/p8022.h | 1 -
include/net/psnap.h | 2 +-
include/net/x25.h | 2 +-
net/802/p8022.c | 1 -
net/802/psnap.c | 5 ++++-
net/8021q/vlan.h | 2 +-
net/8021q/vlan_dev.c | 8 +++++++-
net/appletalk/aarp.c | 6 +++++-
net/appletalk/ddp.c | 15 +++++++++++++---
net/ax25/ax25_in.c | 8 +++++++-
net/bridge/br_private.h | 2 +-
net/bridge/br_stp_bpdu.c | 8 ++++++--

```
 net/core/dev.c                |    6 +++---
 net/decnet/af_decnet.c        |    2 +-
 net/decnet/dn_route.c         |    6 +++++-
 net/econet/af_econet.c        |    6 +++++-
 net/ipv4/arp.c                |    6 +++++-
 net/ipv4/ip_input.c           |    7 +++++--
 net/ipv4/ipconfig.c           |   16 ++++++++++++----
 net/ipv6/ip6_input.c          |    8 +++++++-
 net/ipx/af_ipx.c              |    6 +++++-
 net/irda/irlap_frame.c        |    7 +++++--
 net/irda/irmod.c              |    2 +-
 net/llc/llc_core.c            |    1 -
 net/llc/llc_input.c           |   10 +++++++---
 net/packet/af_packet.c        |   18 ++++++++++++++---
 net/tipc/eth_media.c          |    9 ++++++++-
 net/x25/x25_dev.c             |    6 +++++-
 44 files changed, 195 insertions(+), 67 deletions(-)

diff --git a/drivers/block/aoe/aoenet.c b/drivers/block/aoe/aoenet.c
index 9626e0f..9b72a58 100644
--- a/drivers/block/aoe/aoenet.c
+++ b/drivers/block/aoe/aoenet.c
@@ -8,6 +8,7 @@
 #include <linux/blkdev.h>
 #include <linux/netdevice.h>
 #include <linux/moduleparam.h>
+#include <net/net_namespace.h>
 #include "aoe.h"

 #define NECODES 5
@@ -108,11 +109,15 @@ aoenet_xmit(struct sk_buff *sl)
  * (1) len doesn't include the header by default.  I want this.
  */
 static int
-aoenet_rcv(struct sk_buff *skb, struct net_device *ifp, struct packet_type *pt, struct net_device
*orig_dev)
+aoenet_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *ifp = skb->dev;
  struct aoe_hdr *h;
  u32 n;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto exit;
+
  skb = skb_share_check(skb, GFP_ATOMIC);
  if (skb == NULL)
   return 0;
```

```
diff --git a/drivers/net/bonding/bond_3ad.c b/drivers/net/bonding/bond_3ad.c
index 3fb354d..eea4f11 100644
--- a/drivers/net/bonding/bond_3ad.c
+++ b/drivers/net/bonding/bond_3ad.c
@@ -29,6 +29,7 @@
 #include <linux/ethtool.h>
 #include <linux/if_bonding.h>
 #include <linux/pkt_sched.h>
+#include <net/net_namespace.h>
 #include "bonding.h"
 #include "bond_3ad.h"

@@ -2443,12 +2444,16 @@ out:
  return 0;
 }

-int bond_3ad_lacpdu_recv(struct sk_buff *skb, struct net_device *dev, struct packet_type* ptype,
struct net_device *orig_dev)
+int bond_3ad_lacpdu_recv(struct sk_buff *skb, struct packet_type* ptype, struct net_device
*orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct bonding *bond = dev->priv;
  struct slave *slave = NULL;
  int ret = NET_RX_DROP;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto out;
+
  if (!(dev->flags & IFF_MASTER))
   goto out;

diff --git a/drivers/net/bonding/bond_3ad.h b/drivers/net/bonding/bond_3ad.h
index 6ad5ad6..1f2d7d2 100644
--- a/drivers/net/bonding/bond_3ad.h
+++ b/drivers/net/bonding/bond_3ad.h
@@ -282,7 +282,7 @@ void bond_3ad_adapter_duplex_changed(struct slave *slave);
 void bond_3ad_handle_link_change(struct slave *slave, char link);
 int  bond_3ad_get_active_agg_info(struct bonding *bond, struct ad_info *ad_info);
 int bond_3ad_xmit_xor(struct sk_buff *skb, struct net_device *dev);
-int bond_3ad_lacpdu_recv(struct sk_buff *skb, struct net_device *dev, struct packet_type* ptype,
struct net_device *orig_dev);
+int bond_3ad_lacpdu_recv(struct sk_buff *skb, struct packet_type* ptype, struct net_device
*orig_dev);
 int bond_3ad_set_carrier(struct bonding *bond);
 #endif //__BOND_3AD_H__

diff --git a/drivers/net/bonding/bond_alb.c b/drivers/net/bonding/bond_alb.c
```

```
index 3292316..be780a8 100644
--- a/drivers/net/bonding/bond_alb.c
+++ b/drivers/net/bonding/bond_alb.c
@@ -336,12 +336,16 @@ static void rlb_update_entry_from_arp(struct bonding *bond, struct
arp_pkt *arp)
  _unlock_rx_hashtbl(bond);
 }

-static int rlb_arp_recv(struct sk_buff *skb, struct net_device *bond_dev, struct packet_type
*ptype, struct net_device *orig_dev)
+static int rlb_arp_recv(struct sk_buff *skb, struct packet_type *ptype, struct net_device *orig_dev)
 {
+ struct net_device *bond_dev = skb->dev;
  struct bonding *bond = bond_dev->priv;
  struct arp_pkt *arp = (struct arp_pkt *)skb->data;
  int res = NET_RX_DROP;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto out;
+
  if (!(bond_dev->flags & IFF_MASTER))
   goto out;

diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 9b3bf4e..9c70568 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -2475,14 +2475,18 @@ static void bond_validate_arp(struct bonding *bond, struct slave
*slave, u32 sip
  }
 }

-static int bond_arp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int bond_arp_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct arphdr *arp;
  struct slave *slave;
  struct bonding *bond;
  unsigned char *arp_ptr;
  u32 sip, tip;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto out;
+
  if (!(dev->priv_flags & IFF_BONDING) || !(dev->flags & IFF_MASTER))
   goto out;
```

```
diff --git a/drivers/net/hamradio/bpqether.c b/drivers/net/hamradio/bpqether.c
index 9fc92ad..c513e90 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -93,7 +93,7 @@ static char bcast_addr[6]={0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

 static char bpq_eth_addr[6];

-static int bpq_rcv(struct sk_buff *, struct net_device *, struct packet_type *, struct net_device *);
+static int bpq_rcv(struct sk_buff *, struct packet_type *, struct net_device *);
 static int bpq_device_event(struct notifier_block *, unsigned long, void *);
 static const char *bpq_print_ethaddr(const unsigned char *);

@@ -166,13 +166,17 @@ static inline int dev_is_ethdev(struct net_device *dev)
 /*
  * Receive an AX.25 frame via an ethernet interface.
  */
-static int bpq_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *ptype, struct
net_device *orig_dev)
+static int bpq_rcv(struct sk_buff *skb, struct packet_type *ptype, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  int len;
  char * ptr;
  struct ethhdr *eth;
  struct bpqdev *bpq;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   return NET_RX_DROP;

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index d09334d..caf8ca3 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -376,7 +376,6 @@ abort_kfree:
  *
  ************************************************************************/
 static int pppoe_rcv(struct sk_buff *skb,
-       struct net_device *dev,
        struct packet_type *pt,
        struct net_device *orig_dev)

@@ -384,6 +383,9 @@ static int pppoe_rcv(struct sk_buff *skb,
  struct pppoe_hdr *ph;
```

```
  struct pppox_sock *po;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  if (!pskb_may_pull(skb, sizeof(struct pppoe_hdr)))
   goto drop;

@@ -408,7 +410,6 @@ out:
  *
  ***********************************************************************/
 static int pppoe_disc_rcv(struct sk_buff *skb,
-      struct net_device *dev,
       struct packet_type *pt,
       struct net_device *orig_dev)

@@ -416,6 +417,9 @@ static int pppoe_disc_rcv(struct sk_buff *skb,
  struct pppoe_hdr *ph;
  struct pppox_sock *po;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto abort;
+
  if (!pskb_may_pull(skb, sizeof(struct pppoe_hdr)))
   goto abort;

diff --git a/drivers/net/wan/hdlc.c b/drivers/net/wan/hdlc.c
index db354e0..f3bf160 100644
--- a/drivers/net/wan/hdlc.c
+++ b/drivers/net/wan/hdlc.c
@@ -36,6 +36,7 @@
 #include <linux/rtnetlink.h>
 #include <linux/notifier.h>
 #include <linux/hdlc.h>
+#include <net/net_namespace.h>


 static const char* version = "HDLC support module revision 1.20";
@@ -62,10 +63,17 @@ static struct net_device_stats *hdlc_get_stats(struct net_device *dev)



-static int hdlc_rcv(struct sk_buff *skb, struct net_device *dev,
+static int hdlc_rcv(struct sk_buff *skb,
       struct packet_type *p, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct hdlc_device_desc *desc = dev_to_desc(dev);
```

```
+
+ if (!net_eq(skb->dev->nd_net, init_net())) {
+  kfree_skb(skb);
+  return 0;
+ }
+
  if (desc->netif_rx)
   return desc->netif_rx(skb);

diff --git a/drivers/net/wan/lapbether.c b/drivers/net/wan/lapbether.c
index 6c302e9..c1de21e 100644
--- a/drivers/net/wan/lapbether.c
+++ b/drivers/net/wan/lapbether.c
@@ -86,11 +86,15 @@ static __inline__ int dev_is_ethdev(struct net_device *dev)
 /*
  * Receive a LAPB frame via an ethernet interface.
  */
-static int lapbeth_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *ptype, struct
net_device *orig_dev)
+static int lapbeth_rcv(struct sk_buff *skb, struct packet_type *ptype, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  int len, err;
  struct lapbethdev *lapbeth;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   return NET_RX_DROP;

diff --git a/drivers/net/wan/syncppp.c b/drivers/net/wan/syncppp.c
index 218f7b5..25137fc 100644
--- a/drivers/net/wan/syncppp.c
+++ b/drivers/net/wan/syncppp.c
@@ -51,6 +51,7 @@
 #include <linux/spinlock.h>
 #include <linux/rcupdate.h>

+#include <net/net_namespace.h>
 #include <net/syncppp.h>

 #include <asm/byteorder.h>
@@ -220,13 +221,13 @@ static void sppp_clear_timeout(struct sppp *p)
  * here.
  */

-static void sppp_input (struct net_device *dev, struct sk_buff *skb)
```

```
+static void sppp_input (struct sk_buff *skb)
 {
+ struct net_device *dev = skb->dev;
  struct ppp_header *h;
  struct sppp *sp = (struct sppp *)sppp_of(dev);
  unsigned long flags;

- skb->dev=dev;
  skb->mac.raw=skb->data;

  if (dev->flags & IFF_RUNNING)
@@ -1443,11 +1444,16 @@ static void sppp_print_bytes (u_char *p, u16 len)
  * after interrupt servicing to process frames queued via netif_rx.
  */

-static int sppp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *p, struct
net_device *orig_dev)
+static int sppp_rcv(struct sk_buff *skb, struct packet_type *p, struct net_device *orig_dev)
 {
+ if (!net_eq(skb->dev->nd_net, init_net())) {
+  kfree_skb(skb);
+  return 0;
+ }
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   return NET_RX_DROP;
- sppp_input(dev,skb);
+ sppp_input(skb);
  return 0;
 }

diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 6a1579d..9e28671 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -558,7 +558,6 @@ struct packet_type {
  __be16   type; /* This is really htons(ether_type). */
  struct net_device *dev; /* NULL is wildcarded here      */
  int   (*func) (struct sk_buff *,
-      struct net_device *,
       struct packet_type *,
       struct net_device *);
  struct sk_buff  *(*gso_segment)(struct sk_buff *skb,
diff --git a/include/net/ax25.h b/include/net/ax25.h
index 5ae10dd..a2ad59a 100644
--- a/include/net/ax25.h
+++ b/include/net/ax25.h
@@ -360,7 +360,7 @@ extern int  ax25_protocol_is_registered(unsigned int);
```

```
 /* ax25_in.c */
 extern int  ax25_rx_iframe(ax25_cb *, struct sk_buff *);
-extern int  ax25_kiss_rcv(struct sk_buff *, struct net_device *, struct packet_type *, struct
net_device *);
+extern int  ax25_kiss_rcv(struct sk_buff *, struct packet_type *, struct net_device *);

 /* ax25_ip.c */
 extern int  ax25_hard_header(struct sk_buff *, struct net_device *, unsigned short, void *, void *,
unsigned int);
diff --git a/include/net/datalink.h b/include/net/datalink.h
index deb7ca7..133d55e 100644
--- a/include/net/datalink.h
+++ b/include/net/datalink.h
@@ -8,7 +8,7 @@ struct datalink_proto {

         unsigned short  header_length;

-        int     (*rcvfunc)(struct sk_buff *, struct net_device *,
+        int     (*rcvfunc)(struct sk_buff *,
                            struct packet_type *, struct net_device *);
  int     (*request)(struct datalink_proto *, struct sk_buff *,
                                 unsigned char *);
diff --git a/include/net/ip.h b/include/net/ip.h
index 053f02b..c0c0dfd 100644
--- a/include/net/ip.h
+++ b/include/net/ip.h
@@ -88,7 +88,7 @@ extern int  igmp_mc_proc_init(void);
 extern int  ip_build_and_send_pkt(struct sk_buff *skb, struct sock *sk,
           __be32 saddr, __be32 daddr,
           struct ip_options *opt);
-extern int  ip_rcv(struct sk_buff *skb, struct net_device *dev,
+extern int  ip_rcv(struct sk_buff *skb,
          struct packet_type *pt, struct net_device *orig_dev);
 extern int  ip_local_deliver(struct sk_buff *skb);
 extern int  ip_mr_input(struct sk_buff *skb);
diff --git a/include/net/ipv6.h b/include/net/ipv6.h
index 00328b7..0b1d1a9 100644
--- a/include/net/ipv6.h
+++ b/include/net/ipv6.h
@@ -438,7 +438,6 @@ static inline int ipv6_addr_diff(const struct in6_addr *a1, const struct
in6_add
  */

 extern int   ipv6_rcv(struct sk_buff *skb,
-      struct net_device *dev,
       struct packet_type *pt,
       struct net_device *orig_dev);
```

```
diff --git a/include/net/llc.h b/include/net/llc.h
index f502458..dae09b9 100644
--- a/include/net/llc.h
+++ b/include/net/llc.h
@@ -48,7 +48,6 @@ struct llc_sap {
  unsigned char  f_bit;
  atomic_t         refcnt;
  int   (*rcv_func)(struct sk_buff *skb,
-         struct net_device *dev,
          struct packet_type *pt,
          struct net_device *orig_dev);
  struct llc_addr  laddr;
@@ -67,7 +66,7 @@ extern struct list_head llc_sap_list;
 extern rwlock_t llc_sap_list_lock;
 extern unsigned char llc_station_mac_sa[ETH_ALEN];

-extern int llc_rcv(struct sk_buff *skb, struct net_device *dev,
+extern int llc_rcv(struct sk_buff *skb,
       struct packet_type *pt, struct net_device *orig_dev);

 extern int llc_mac_hdr_init(struct sk_buff *skb,
@@ -81,7 +80,6 @@ extern void llc_set_station_handler(void (*handler)(struct sk_buff *skb));

 extern struct llc_sap *llc_sap_open(unsigned char lsap,
         int (*rcv)(struct sk_buff *skb,
-            struct net_device *dev,
             struct packet_type *pt,
             struct net_device *orig_dev));
 static inline void llc_sap_hold(struct llc_sap *sap)
diff --git a/include/net/p8022.h b/include/net/p8022.h
index 42e9fac..545c15e 100644
--- a/include/net/p8022.h
+++ b/include/net/p8022.h
@@ -3,7 +3,6 @@
 extern struct datalink_proto *
  register_8022_client(unsigned char type,
         int (*func)(struct sk_buff *skb,
-       struct net_device *dev,
        struct packet_type *pt,
        struct net_device *orig_dev));
 extern void unregister_8022_client(struct datalink_proto *proto);
diff --git a/include/net/psnap.h b/include/net/psnap.h
index b2e01cc..e935d50 100644
--- a/include/net/psnap.h
+++ b/include/net/psnap.h
@@ -1,7 +1,7 @@
 #ifndef _NET_PSNAP_H
```

```
 #define _NET_PSNAP_H

-extern struct datalink_proto *register_snap_client(unsigned char *desc, int (*rcvfunc)(struct
sk_buff *, struct net_device *, struct packet_type *, struct net_device *orig_dev));
+extern struct datalink_proto *register_snap_client(unsigned char *desc, int (*rcvfunc)(struct
sk_buff *, struct packet_type *, struct net_device *orig_dev));
 extern void unregister_snap_client(struct datalink_proto *proto);

 #endif
diff --git a/include/net/x25.h b/include/net/x25.h
index e47fe44..e3d4cfb 100644
--- a/include/net/x25.h
+++ b/include/net/x25.h
@@ -184,7 +184,7 @@ extern void x25_kill_by_neigh(struct x25_neigh *);

 /* x25_dev.c */
 extern void x25_send_frame(struct sk_buff *, struct x25_neigh *);
-extern int  x25_lapb_receive_frame(struct sk_buff *, struct net_device *, struct packet_type *,
struct net_device *);
+extern int  x25_lapb_receive_frame(struct sk_buff *, struct packet_type *, struct net_device *);
 extern void x25_establish_link(struct x25_neigh *);
 extern void x25_terminate_link(struct x25_neigh *);

diff --git a/net/802/p8022.c b/net/802/p8022.c
index 2530f35..1c7022d 100644
--- a/net/802/p8022.c
+++ b/net/802/p8022.c
@@ -34,7 +34,6 @@ static int p8022_request(struct datalink_proto *dl, struct sk_buff *skb,

 struct datalink_proto *register_8022_client(unsigned char type,
          int (*func)(struct sk_buff *skb,
-        struct net_device *dev,
         struct packet_type *pt,
         struct net_device *orig_dev))
 {
diff --git a/net/802/psnap.c b/net/802/psnap.c
index 270b9d2..59ac0c5 100644
--- a/net/802/psnap.c
+++ b/net/802/psnap.c
@@ -46,7 +46,7 @@ static struct datalink_proto *find_snap_client(unsigned char *desc)
 /*
  * A SNAP packet has arrived
  */
-static int snap_rcv(struct sk_buff *skb, struct net_device *dev,
+static int snap_rcv(struct sk_buff *skb,
       struct packet_type *pt, struct net_device *orig_dev)
 {
  int rc = 1;
```

```
@@ -61,7 +61,7 @@ static int snap_rcv(struct sk_buff *skb, struct net_device *dev,
   /* Pass the frame on. */
   skb->h.raw  += 5;
   skb_pull_rcsum(skb, 5);
-  rc = proto->rcvfunc(skb, dev, &snap_packet_type, orig_dev);
+  rc = proto->rcvfunc(skb, &snap_packet_type, orig_dev);
  } else {
   skb->sk = NULL;
   kfree_skb(skb);
@@ -117,7 +117,6 @@ module_exit(snap_exit);
  */
 struct datalink_proto *register_snap_client(unsigned char *desc,
         int (*rcvfunc)(struct sk_buff *,
-              struct net_device *,
           struct packet_type *,
           struct net_device *))
 {
diff --git a/net/8021q/vlan.h b/net/8021q/vlan.h
index 9ae3a14..9207999 100644
--- a/net/8021q/vlan.h
+++ b/net/8021q/vlan.h
@@ -50,7 +50,7 @@ struct net_device *__find_vlan_dev(struct net_device* real_dev,

 /* found in vlan_dev.c */
 int vlan_dev_rebuild_header(struct sk_buff *skb);
-int vlan_skb_recv(struct sk_buff *skb, struct net_device *dev,
+int vlan_skb_recv(struct sk_buff *skb,
                   struct packet_type *ptype, struct net_device *orig_dev);
 int vlan_dev_hard_header(struct sk_buff *skb, struct net_device *dev,
                          unsigned short type, void *daddr, void *saddr,
diff --git a/net/8021q/vlan_dev.c b/net/8021q/vlan_dev.c
index 60a508e..9fce3a8 100644
--- a/net/8021q/vlan_dev.c
+++ b/net/8021q/vlan_dev.c
@@ -112,9 +112,10 @@ static inline struct sk_buff *vlan_check_reorder_header(struct sk_buff
*skb)
  *                been commented out now...  --Ben
  *
  */
-int vlan_skb_recv(struct sk_buff *skb, struct net_device *dev,
+int vlan_skb_recv(struct sk_buff *skb,
                   struct packet_type* ptype, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  unsigned char *rawp = NULL;
  struct vlan_hdr *vhdr = (struct vlan_hdr *)(skb->data);
  unsigned short vid;
@@ -122,6 +123,11 @@ int vlan_skb_recv(struct sk_buff *skb, struct net_device *dev,
```

```
  unsigned short vlan_TCI;
  __be16 proto;

+ if (!net_eq(skb->dev->nd_net, init_net())) {
+  kfree_skb(skb);
+  return 0;
+ }
+
  /* vlan_TCI = ntohs(get_unaligned(&vhdr->h_vlan_TCI)); */
  vlan_TCI = ntohs(vhdr->h_vlan_TCI);

diff --git a/net/appletalk/aarp.c b/net/appletalk/aarp.c
index b51a010..85c4dbc 100644
--- a/net/appletalk/aarp.c
+++ b/net/appletalk/aarp.c
@@ -697,9 +697,10 @@ static void __aarp_resolved(struct aarp_entry **list, struct aarp_entry
*a,
  * This is called by the SNAP driver whenever we see an AARP SNAP
  * frame. We currently only support Ethernet.
  */
-static int aarp_rcv(struct sk_buff *skb, struct net_device *dev,
+static int aarp_rcv(struct sk_buff *skb,
       struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct elapaarp *ea = aarp_hdr(skb);
  int hash, ret = 0;
  __u16 function;
@@ -707,6 +708,9 @@ static int aarp_rcv(struct sk_buff *skb, struct net_device *dev,
  struct atalk_addr sa, *ma, da;
  struct atalk_iface *ifa;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto out0;
+
  /* We only do Ethernet SNAP AARP. */
  if (dev->type != ARPHRD_ETHER)
   goto out0;
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index e08367b..f4ff8aa 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -1393,9 +1393,10 @@ free_it:
  * extracted. PPP should probably pass frames marked as for this layer.
  * [ie ARPHRD_ETHERTALK]
  */
-static int atalk_rcv(struct sk_buff *skb, struct net_device *dev,
+static int atalk_rcv(struct sk_buff *skb,
```

```
        struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct ddpehdr *ddp;
  struct sock *sock;
  struct atalk_iface *atif;
@@ -1403,6 +1404,9 @@ static int atalk_rcv(struct sk_buff *skb, struct net_device *dev,
         int origlen;
  __u16 len_hops;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto freeit;
+
  /* Don't mangle buffer if shared */
  if (!(skb = skb_share_check(skb, GFP_ATOMIC)))
   goto out;
@@ -1482,9 +1486,14 @@ freeit:
  * Caller must provide enough headroom on the packet to pull the short
  * header and append a long one.
  */
-static int ltalk_rcv(struct sk_buff *skb, struct net_device *dev,
+static int ltalk_rcv(struct sk_buff *skb,
         struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
+
+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto freeit;
+
  /* Expand any short form frames */
  if (skb->mac.raw[2] == 1) {
   struct ddpehdr *ddp;
@@ -1526,7 +1535,7 @@ static int ltalk_rcv(struct sk_buff *skb, struct net_device *dev,
  }
  skb->h.raw = skb->data;

- return atalk_rcv(skb, dev, pt, orig_dev);
+ return atalk_rcv(skb, pt, orig_dev);
 freeit:
  kfree_skb(skb);
  return 0;
diff --git a/net/ax25/ax25_in.c b/net/ax25/ax25_in.c
index e9d9429..8c9b0dd 100644
--- a/net/ax25/ax25_in.c
+++ b/net/ax25/ax25_in.c
@@ -444,12 +444,18 @@ static int ax25_rcv(struct sk_buff *skb, struct net_device *dev,
 /*
  * Receive an AX.25 frame via a SLIP interface.
```

```
  */
-int ax25_kiss_rcv(struct sk_buff *skb, struct net_device *dev,
+int ax25_kiss_rcv(struct sk_buff *skb,
      struct packet_type *ptype, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  skb->sk = NULL;  /* Initially we don't know who it's for */
  skb->destructor = NULL; /* Who initializes this, dammit?! */

+ if (!net_eq(skb->dev->nd_net, init_net())) {
+  kfree_skb(skb);
+  return 0;
+ }
+
  if ((*skb->data & 0x0F) != 0) {
   kfree_skb(skb); /* Not a KISS data frame */
   return 0;
diff --git a/net/bridge/br_private.h b/net/bridge/br_private.h
index 3a534e9..f1712b9 100644
--- a/net/bridge/br_private.h
+++ b/net/bridge/br_private.h
@@ -223,7 +223,7 @@ extern void br_stp_set_path_cost(struct net_bridge_port *p,
 extern ssize_t br_show_bridge_id(char *buf, const struct bridge_id *id);

 /* br_stp_bpdu.c */
-extern int br_stp_rcv(struct sk_buff *skb, struct net_device *dev,
+extern int br_stp_rcv(struct sk_buff *skb,
         struct packet_type *pt, struct net_device *orig_dev);

 /* br_stp_timer.c */
diff --git a/net/bridge/br_stp_bpdu.c b/net/bridge/br_stp_bpdu.c
index 068d8af..7f9f8b4 100644
--- a/net/bridge/br_stp_bpdu.c
+++ b/net/bridge/br_stp_bpdu.c
@@ -17,6 +17,7 @@
 #include <linux/netfilter_bridge.h>
 #include <linux/etherdevice.h>
 #include <linux/llc.h>
+#include <net/net_namespace.h>
 #include <net/llc.h>
 #include <net/llc_pdu.h>
 #include <asm/unaligned.h>
@@ -129,15 +130,18 @@ void br_send_tcn_bpdu(struct net_bridge_port *p)
  *
  * NO locks, but rcu_read_lock (preempt_disabled)
  */
-int br_stp_rcv(struct sk_buff *skb, struct net_device *dev,
+int br_stp_rcv(struct sk_buff *skb,
```

```
        struct packet_type *pt, struct net_device *orig_dev)
 {
  const struct llc_pdu_un *pdu = llc_pdu_un_hdr(skb);
  const unsigned char *dest = eth_hdr(skb)->h_dest;
- struct net_bridge_port *p = rcu_dereference(dev->br_port);
+ struct net_bridge_port *p = rcu_dereference(skb->dev->br_port);
  struct net_bridge *br;
  const unsigned char *buf;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto err;
+
  if (!p)
   goto err;

diff --git a/net/core/dev.c b/net/core/dev.c
index a3ee150..d8aa534 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -1094,7 +1094,7 @@ static void dev_queue_xmit_nit(struct sk_buff *skb, struct net_device
*dev)

    skb2->h.raw = skb2->nh.raw;
    skb2->pkt_type = PACKET_OUTGOING;
-   ptype->func(skb2, skb->dev, ptype, skb->dev);
+   ptype->func(skb2, ptype, skb->dev);
   }
  }
  rcu_read_unlock();
@@ -1693,7 +1693,7 @@ static __inline__ int deliver_skb(struct sk_buff *skb,
       struct net_device *orig_dev)
 {
  atomic_inc(&skb->users);
- return pt_prev->func(skb, skb->dev, pt_prev, orig_dev);
+ return pt_prev->func(skb, pt_prev, orig_dev);
 }

 #if defined(CONFIG_BRIDGE) || defined (CONFIG_BRIDGE_MODULE)
@@ -1841,7 +1841,7 @@ ncls:
  }

  if (pt_prev) {
-  ret = pt_prev->func(skb, skb->dev, pt_prev, orig_dev);
+  ret = pt_prev->func(skb, pt_prev, orig_dev);
  } else {
   kfree_skb(skb);
   /* Jamal, now you will not able to escape explaining
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
```

```
index f1553fa..5e8042f 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -2104,7 +2104,7 @@ static struct notifier_block dn_dev_notifier = {
  .notifier_call = dn_device_event,
 };

-extern int dn_route_rcv(struct sk_buff *, struct net_device *, struct packet_type *, struct
net_device *);
+extern int dn_route_rcv(struct sk_buff *, struct packet_type *, struct net_device *);

 static struct packet_type dn_dix_packet_type = {
  .type =  __constant_htons(ETH_P_DNA_RT),
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 0d657eb..4263cd9 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -575,14 +575,18 @@ static int dn_route_ptp_hello(struct sk_buff *skb)
  return NET_RX_SUCCESS;
 }

-int dn_route_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+int dn_route_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct dn_skb_cb *cb;
  unsigned char flags = 0;
  __u16 len = dn_ntohs(*(__le16 *)skb->data);
  struct dn_dev *dn = (struct dn_dev *)dev->dn_ptr;
  unsigned char padlen = 0;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto dump_it;
+
  if (dn == NULL)
   goto dump_it;

diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index a0b3fc5..0baffda 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1057,12 +1057,16 @@ release:
  * Receive an Econet frame from a device.
  */

-static int econet_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
```

```
+static int econet_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct ec_framehdr *hdr;
  struct sock *sk;
  struct ec_device *edev = dev->ec_ptr;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  if (skb->pkt_type == PACKET_OTHERHOST)
   goto drop;

diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index e3b89a7..95a34c7 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -928,11 +928,15 @@ static void parp_redo(struct sk_buff *skb)
  * Receive an arp request from the device layer.
  */

-static int arp_rcv(struct sk_buff *skb, struct net_device *dev,
+static int arp_rcv(struct sk_buff *skb,
      struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct arphdr *arp;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto freeskb;
+
  /* ARP header, plus 2 device addresses, plus 2 IP addresses.  */
  if (!pskb_may_pull(skb, (sizeof(struct arphdr) +
      (2 * dev->addr_len) +
diff --git a/net/ipv4/ip_input.c b/net/ipv4/ip_input.c
index 212734c..77dddce 100644
--- a/net/ipv4/ip_input.c
+++ b/net/ipv4/ip_input.c
@@ -370,11 +370,14 @@ drop:
 /*
  *  Main IP Receive routine.
  */
-int ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device
*orig_dev)
+int ip_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
  struct iphdr *iph;
  u32 len;
```

```
+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  /* When the interface is in promisc. mode, drop all the crap
   * that it receives, do not try to analyse it.
   */
@@ -431,7 +434,7 @@ int ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type
*pt,
  /* Remove any debris in the socket control block */
  memset(IPCB(skb), 0, sizeof(struct inet_skb_parm));

- return NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, dev, NULL,
+ return NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, skb->dev, NULL,
         ip_rcv_finish);

 inhdr_error:
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index 8b649c5..91b5729 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -397,7 +397,7 @@ static int __init ic_defaults(void)

 #ifdef IPCONFIG_RARP

-static int ic_rarp_recv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev);
+static int ic_rarp_recv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev);

 static struct packet_type rarp_packet_type __initdata = {
  .type = __constant_htons(ETH_P_RARP),
@@ -418,14 +418,18 @@ static inline void ic_rarp_cleanup(void)
  *  Process received RARP packet.
  */
 static int __init
-ic_rarp_recv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device
*orig_dev)
+ic_rarp_recv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct arphdr *rarp;
  unsigned char *rarp_ptr;
  __be32 sip, tip;
  unsigned char *sha, *tha;  /* s for "source", t for "target" */
  struct ic_device *d;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
```

```
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   return NET_RX_DROP;

@@ -559,7 +563,7 @@ struct bootp_pkt {  /* BOOTP packet format */
 #define DHCPRELEASE 7
 #define DHCPINFORM 8

-static int ic_bootp_recv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev);
+static int ic_bootp_recv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev);

 static struct packet_type bootp_packet_type __initdata = {
  .type = __constant_htons(ETH_P_IP),
@@ -827,13 +831,17 @@ static void __init ic_do_bootp_ext(u8 *ext)
 /*
  *  Receive BOOTP reply.
  */
-static int __init ic_bootp_recv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt,
struct net_device *orig_dev)
+static int __init ic_bootp_recv(struct sk_buff *skb, struct packet_type *pt, struct net_device
*orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct bootp_pkt *b;
  struct iphdr *h;
  struct ic_device *d;
  int len, ext_len;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  /* Perform verifications before taking the lock.  */
  if (skb->pkt_type == PACKET_OTHERHOST)
   goto drop;
diff --git a/net/ipv6/ip6_input.c b/net/ipv6/ip6_input.c
index ad0b8ab..ac366b9 100644
--- a/net/ipv6/ip6_input.c
+++ b/net/ipv6/ip6_input.c
@@ -56,12 +56,18 @@ inline int ip6_rcv_finish( struct sk_buff *skb)
  return dst_input(skb);
 }

-int ipv6_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device
*orig_dev)
+int ipv6_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
```

```
  struct ipv6hdr *hdr;
  u32   pkt_len;
  struct inet6_dev *idev;

+ if (!net_eq(skb->dev->nd_net, init_net())) {
+  kfree_skb(skb);
+  return 0;
+ }
+
  if (skb->pkt_type == PACKET_OTHERHOST) {
   kfree_skb(skb);
   return 0;
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index 2ec4a3c..5c5f2cd 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -1637,14 +1637,18 @@ out:
  return rc;
 }

-static int ipx_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int ipx_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  /* NULL here for pt means the packet was looped back */
  struct ipx_interface *intrfc;
  struct ipxhdr *ipx;
  u16 ipx_pktsize;
  int rc = 0;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  /* Not ours */
        if (skb->pkt_type == PACKET_OTHERHOST)
         goto drop;
diff --git a/net/irda/irlap_frame.c b/net/irda/irlap_frame.c
index dba349c..3252be7 100644
--- a/net/irda/irlap_frame.c
+++ b/net/irda/irlap_frame.c
@@ -1306,7 +1306,7 @@ static void irlap_recv_test_frame(struct irlap_cb *self, struct sk_buff
*skb,
  * LMP level in irlmp.c.
  * Jean II
  */
-int irlap_driver_rcv(struct sk_buff *skb, struct net_device *dev,
+int irlap_driver_rcv(struct sk_buff *skb,
```

```
        struct packet_type *ptype, struct net_device *orig_dev)
 {
  struct irlap_info info;
@@ -1314,8 +1314,11 @@ int irlap_driver_rcv(struct sk_buff *skb, struct net_device *dev,
  int command;
  __u8 control;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto out;
+
  /* FIXME: should we get our own field? */
- self = (struct irlap_cb *) dev->atalk_ptr;
+ self = (struct irlap_cb *) skb->dev->atalk_ptr;

  /* If the net device is down, then IrLAP is gone! */
  if (!self || self->magic != LAP_MAGIC) {
diff --git a/net/irda/irmod.c b/net/irda/irmod.c
index 2869b16..6b1989c 100644
--- a/net/irda/irmod.c
+++ b/net/irda/irmod.c
@@ -52,7 +52,7 @@ extern void irda_sysctl_unregister(void);
 extern int  irsock_init(void);
 extern void irsock_cleanup(void);
 /* irlap_frame.c */
-extern int  irlap_driver_rcv(struct sk_buff *, struct net_device *,
+extern int  irlap_driver_rcv(struct sk_buff *,
         struct packet_type *, struct net_device *);

 /*
diff --git a/net/llc/llc_core.c b/net/llc/llc_core.c
index d12413c..f438c38 100644
--- a/net/llc/llc_core.c
+++ b/net/llc/llc_core.c
@@ -112,7 +112,6 @@ struct llc_sap *llc_sap_find(unsigned char sap_value)
  */
 struct llc_sap *llc_sap_open(unsigned char lsap,
         int (*func)(struct sk_buff *skb,
-       struct net_device *dev,
        struct packet_type *pt,
        struct net_device *orig_dev))
 {
diff --git a/net/llc/llc_input.c b/net/llc/llc_input.c
index db82aff..cecb4a9 100644
--- a/net/llc/llc_input.c
+++ b/net/llc/llc_input.c
@@ -12,6 +12,7 @@
  * See the GNU General Public License for more details.
  */
```

```
 #include <linux/netdevice.h>
+#include <net/net_namespace.h>
 #include <net/llc.h>
 #include <net/llc_pdu.h>
 #include <net/llc_sap.h>
@@ -136,15 +137,18 @@ static inline int llc_fixup_skb(struct sk_buff *skb)
  * the frame is related to a busy connection (a connection is sending
  * data now), it queues this frame in the connection's backlog.
  */
-int llc_rcv(struct sk_buff *skb, struct net_device *dev,
+int llc_rcv(struct sk_buff *skb,
      struct packet_type *pt, struct net_device *orig_dev)
 {
  struct llc_sap *sap;
  struct llc_pdu_sn *pdu;
  int dest;
- int (*rcv)(struct sk_buff *, struct net_device *,
+ int (*rcv)(struct sk_buff *,
      struct packet_type *, struct net_device *);

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  /*
   * When the interface is in promisc. mode, drop all the crap that it
   * receives, do not try to analyse it.
@@ -175,7 +179,7 @@ int llc_rcv(struct sk_buff *skb, struct net_device *dev,
  if (rcv) {
    struct sk_buff *cskb = skb_clone(skb, GFP_ATOMIC);
    if (cskb)
-    rcv(cskb, dev, pt, orig_dev);
+    rcv(cskb, pt, orig_dev);
  }
  dest = llc_pdu_type(skb);
  if (unlikely(!dest || !llc_type_handlers[dest - 1]))
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index ca371ea..aa298c3 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -258,11 +258,15 @@ static const struct proto_ops packet_ops;
 #ifdef CONFIG_SOCK_PACKET
 static const struct proto_ops packet_ops_spkt;

-static int packet_rcv_spkt(struct sk_buff *skb, struct net_device *dev,  struct packet_type *pt,
struct net_device *orig_dev)
+static int packet_rcv_spkt(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
```

```
  struct sock *sk;
  struct sockaddr_pkt *spkt;

+ if (!net_eq(dev->nd_net, init_net()))
+  goto out;
+
  /*
   * When we registered the protocol we saved the socket in the data
   * field for just this event.
@@ -461,8 +465,9 @@ static inline int run_filter(struct sk_buff *skb, struct sock *sk,
    we will not harm anyone.
  */

-static int packet_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int packet_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct sock *sk;
  struct sockaddr_ll *sll;
  struct packet_sock *po;
@@ -470,6 +475,9 @@ static int packet_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet
  int skb_len = skb->len;
  unsigned snaplen;

+ if (!net_eq(dev->nd_net, init_net()))
+  goto drop;
+
  if (skb->pkt_type == PACKET_LOOPBACK)
   goto drop;

@@ -561,8 +569,9 @@ drop:
 }

 #ifdef CONFIG_PACKET_MMAP
-static int tpacket_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int tpacket_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct sock *sk;
  struct packet_sock *po;
  struct sockaddr_ll *sll;
@@ -574,6 +583,9 @@ static int tpacket_rcv(struct sk_buff *skb, struct net_device *dev, struct
packe
  unsigned short macoff, netoff;
  struct sk_buff *copy_skb = NULL;
```

```
+ if (!net_eq(dev->nd_net, init_net()))
+  goto drop;
+
  if (skb->pkt_type == PACKET_LOOPBACK)
   goto drop;

diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index 682da4a..b181cf9 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -38,6 +38,7 @@
 #include <net/tipc/tipc_bearer.h>
 #include <net/tipc/tipc_msg.h>
 #include <linux/netdevice.h>
+#include <net/net_namespace.h>

 #define MAX_ETH_BEARERS  2
 #define ETH_LINK_PRIORITY TIPC_DEF_LINK_PRI
@@ -91,12 +92,18 @@ static int send_msg(struct sk_buff *buf, struct tipc_bearer *tb_ptr,
  * and ensures message size matches actual length
  */

-static int recv_msg(struct sk_buff *buf, struct net_device *dev,
+static int recv_msg(struct sk_buff *buf,
       struct packet_type *pt, struct net_device *orig_dev)
 {
+ struct net_device *dev = buf->dev;
  struct eth_bearer *eb_ptr = (struct eth_bearer *)pt->af_packet_priv;
  u32 size;

+ if (!net_eq(buf->dev->nd_net, init_net())) {
+  kfree_skb(buf);
+  return 0;
+ }
+
  if (likely(eb_ptr->bearer)) {
        if (likely(!dev->promiscuity) ||
            !memcmp(buf->mac.raw,dev->dev_addr,ETH_ALEN) ||
diff --git a/net/x25/x25_dev.c b/net/x25/x25_dev.c
index 47b68a3..0f63415 100644
--- a/net/x25/x25_dev.c
+++ b/net/x25/x25_dev.c
@@ -79,12 +79,16 @@ static int x25_receive_data(struct sk_buff *skb, struct x25_neigh *nb)
  return 0;
 }

-int x25_lapb_receive_frame(struct sk_buff *skb, struct net_device *dev,
```

```
+int x25_lapb_receive_frame(struct sk_buff *skb,
       struct packet_type *ptype, struct net_device *orig_dev)
 {
+ struct net_device *dev = skb->dev;
  struct sk_buff *nskb;
  struct x25_neigh *nb;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+  goto drop;
+
  nskb = skb_copy(skb, GFP_ATOMIC);
  if (!nskb)
   goto drop;
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 13/31] net: Make device event notification network namespace safe
Posted by ebiederm on Thu, 25 Jan 2007 19:00:15 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Every user of the network device notifiers is either a protocol stack or a pseudo device. If a protocol stack that does not have support for multiple network namespaces receives an event for a device that is not in the initial network namespace it quite possibly can get confused and do the wrong thing.

To avoid problems until all of the protocol stacks are converted this patch modifies all netdev event handlers to ignore events on devices that are not in the initial network namespace.

As the rest of the code is made network namespace aware these checks can be removed.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 arch/ia64/hp/sim/simeth.c          |    3 +++
 drivers/net/bonding/bond_main.c    |    3 +++
 drivers/net/hamradio/bpqether.c    |    3 +++
 drivers/net/pppoe.c                |    3 +++
```

```
 drivers/net/wan/dlci.c             |    3 +++
 drivers/net/wan/hdlc.c             |    3 +++
 drivers/net/wan/lapbether.c        |    3 +++
 net/8021q/vlan.c                   |    4 ++++
 net/appletalk/aarp.c               |    3 +++
 net/appletalk/ddp.c                |    3 +++
 net/atm/clip.c                     |    3 +++
 net/atm/mpc.c                      |    4 ++++
 net/ax25/af_ax25.c                 |    3 +++
 net/bridge/br_notify.c             |    4 ++++
 net/core/dst.c                     |    4 ++++
 net/core/fib_rules.c               |    4 ++++
 net/core/pktgen.c                  |    3 +++
 net/core/rtnetlink.c               |    4 ++++
 net/decnet/af_decnet.c             |    3 +++
 net/econet/af_econet.c             |    3 +++
 net/ipv4/arp.c                     |    3 +++
 net/ipv4/devinet.c                 |    3 +++
 net/ipv4/fib_frontend.c            |    3 +++
 net/ipv4/ipmr.c                    |    7 ++++++-
 net/ipv4/multipath_drr.c           |    3 +++
 net/ipv4/netfilter/ip_queue.c      |    3 +++
 net/ipv4/netfilter/ipt_MASQUERADE.c |    3 +++
 net/ipv6/addrconf.c                |    3 +++
 net/ipv6/ndisc.c                   |    3 +++
 net/ipv6/netfilter/ip6_queue.c     |    3 +++
 net/ipx/af_ipx.c                   |    3 +++
 net/netfilter/nfnetlink_queue.c    |    3 +++
 net/netrom/af_netrom.c             |    3 +++
 net/packet/af_packet.c             |    3 +++
 net/rose/af_rose.c                 |    3 +++
 net/tipc/eth_media.c               |    3 +++
 net/wanrouter/af_wanpipe.c         |    3 +++
 net/x25/af_x25.c                   |    3 +++
 net/xfrm/xfrm_policy.c             |    5 +++++
 security/selinux/netif.c           |    3 +++
 40 files changed, 131 insertions(+), 1 deletions(-)

diff --git a/arch/ia64/hp/sim/simeth.c b/arch/ia64/hp/sim/simeth.c
index 424e925..1cbaa9e 100644
--- a/arch/ia64/hp/sim/simeth.c
+++ b/arch/ia64/hp/sim/simeth.c
@@ -300,6 +300,9 @@ simeth_device_event(struct notifier_block *this,unsigned long event, void
*ptr)
   return NOTIFY_DONE;
  }

+ if (!net_eq(dev->nd_net, init_net()))
```

```
+  return NOTIFY_DONE;
+
  if ( event != NETDEV_UP && event != NETDEV_DOWN ) return NOTIFY_DONE;

  /*
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 9c70568..3e04f58 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -3325,6 +3325,9 @@ static int bond_netdev_event(struct notifier_block *this, unsigned long
event, v
 {
  struct net_device *event_dev = (struct net_device *)ptr;

+ if (!net_eq(event_dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  dprintk("event_dev: %s, event: %lx\n",
   (event_dev ? event_dev->name : "None"),
   event);
diff --git a/drivers/net/hamradio/bpqether.c b/drivers/net/hamradio/bpqether.c
index c513e90..8826a96 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -564,6 +564,9 @@ static int bpq_device_event(struct notifier_block *this,unsigned long
event, voi
 {
  struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (!dev_is_ethdev(dev))
   return NOTIFY_DONE;

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index caf8ca3..3618862 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -299,6 +299,9 @@ static int pppoe_device_event(struct notifier_block *this,
 {
  struct net_device *dev = (struct net_device *) ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* Only look at sockets that are using this specific device. */
  switch (event) {
```

```
  case NETDEV_CHANGEMTU:
diff --git a/drivers/net/wan/dlci.c b/drivers/net/wan/dlci.c
index 7369875..f826494 100644
--- a/drivers/net/wan/dlci.c
+++ b/drivers/net/wan/dlci.c
@@ -513,6 +513,9 @@ static int dlci_dev_event(struct notifier_block *unused,
 {
  struct net_device *dev = (struct net_device *) ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_UNREGISTER) {
   struct dlci_local *dlp;

diff --git a/drivers/net/wan/hdlc.c b/drivers/net/wan/hdlc.c
index f3bf160..e56e0a1 100644
--- a/drivers/net/wan/hdlc.c
+++ b/drivers/net/wan/hdlc.c
@@ -110,6 +110,9 @@ static int hdlc_device_event(struct notifier_block *this, unsigned long
event,
  unsigned long flags;
  int on;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (dev->get_stats != hdlc_get_stats)
   return NOTIFY_DONE; /* not an HDLC device */

diff --git a/drivers/net/wan/lapbether.c b/drivers/net/wan/lapbether.c
index c1de21e..a3560a9 100644
--- a/drivers/net/wan/lapbether.c
+++ b/drivers/net/wan/lapbether.c
@@ -395,6 +395,9 @@ static int lapbeth_device_event(struct notifier_block *this,
  struct lapbethdev *lapbeth;
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (!dev_is_ethdev(dev))
   return NOTIFY_DONE;

diff --git a/net/8021q/vlan.c b/net/8021q/vlan.c
index 18fcb9f..f80cfdd 100644
--- a/net/8021q/vlan.c
+++ b/net/8021q/vlan.c
```

```
@@ -31,6 +31,7 @@
 #include <net/arp.h>
 #include <linux/rtnetlink.h>
 #include <linux/notifier.h>
+#include <net/net_namespace.h>

 #include <linux/if_vlan.h>
 #include "vlan.h"
@@ -595,6 +596,9 @@ static int vlan_device_event(struct notifier_block *unused, unsigned long
event,
  int i, flgs;
  struct net_device *vlandev;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (!grp)
   goto out;

diff --git a/net/appletalk/aarp.c b/net/appletalk/aarp.c
index 85c4dbc..6fd58a6 100644
--- a/net/appletalk/aarp.c
+++ b/net/appletalk/aarp.c
@@ -327,6 +327,9 @@ static int aarp_device_event(struct notifier_block *this, unsigned long
event,
  struct net_device *dev = ptr;
  int ct;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_DOWN) {
   write_lock_bh(&aarp_lock);

diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index f4ff8aa..61f36b1 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -649,6 +649,9 @@ static int ddp_device_event(struct notifier_block *this, unsigned long
event,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_DOWN)
   /* Discard any use of this */
```

```
          atalk_dev_down(dev);
diff --git a/net/atm/clip.c b/net/atm/clip.c
index 5f8a1d2..7d150c2 100644
--- a/net/atm/clip.c
+++ b/net/atm/clip.c
@@ -629,6 +629,9 @@ static int clip_device_event(struct notifier_block *this, unsigned long
event,
 {
  struct net_device *dev = arg;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_UNREGISTER) {
   neigh_ifdown(&clip_tbl, dev);
   return NOTIFY_DONE;
diff --git a/net/atm/mpc.c b/net/atm/mpc.c
index c18f737..4fdb1af 100644
--- a/net/atm/mpc.c
+++ b/net/atm/mpc.c
@@ -953,6 +953,10 @@ static int mpoa_event_listener(struct notifier_block *mpoa_notifier,
unsigned lo
  struct lec_priv *priv;

  dev = (struct net_device *)dev_ptr;
+
+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (dev->name == NULL || strncmp(dev->name, "lec", 3))
   return NOTIFY_DONE; /* we are only interested in lec:s */

diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index cdbf3f6..8c187a6 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -105,6 +105,9 @@ static int ax25_device_event(struct notifier_block *this, unsigned long
event,
 {
  struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* Reject non AX.25 devices */
  if (dev->type != ARPHRD_AX25)
   return NOTIFY_DONE;
diff --git a/net/bridge/br_notify.c b/net/bridge/br_notify.c
```

```
index 2027849..0d56bc2 100644
--- a/net/bridge/br_notify.c
+++ b/net/bridge/br_notify.c
@@ -15,6 +15,7 @@

 #include <linux/kernel.h>
 #include <linux/rtnetlink.h>
+#include <net/net_namespace.h>

 #include "br_private.h"

@@ -36,6 +37,9 @@ static int br_device_event(struct notifier_block *unused, unsigned long
event, v
  struct net_bridge_port *p = dev->br_port;
  struct net_bridge *br;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* not a port of a bridge */
  if (p == NULL)
   return NOTIFY_DONE;
diff --git a/net/core/dst.c b/net/core/dst.c
index 836ec66..8c4a272 100644
--- a/net/core/dst.c
+++ b/net/core/dst.c
@@ -16,6 +16,7 @@
 #include <linux/skbuff.h>
 #include <linux/string.h>
 #include <linux/types.h>
+#include <net/net_namespace.h>

 #include <net/dst.h>

@@ -256,6 +257,9 @@ static int dst_dev_event(struct notifier_block *this, unsigned long event,
void
  struct net_device *dev = ptr;
  struct dst_entry *dst;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (event) {
  case NETDEV_UNREGISTER:
  case NETDEV_DOWN:
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
index 1df6cd4..ffc31c1 100644
--- a/net/core/fib_rules.c
```

```
+++ b/net/core/fib_rules.c
@@ -11,6 +11,7 @@
 #include <linux/types.h>
 #include <linux/kernel.h>
 #include <linux/list.h>
+#include <net/net_namespace.h>
 #include <net/fib_rules.h>

 static LIST_HEAD(rules_ops);
@@ -441,6 +442,9 @@ static int fib_rules_event(struct notifier_block *this, unsigned long event,
  struct net_device *dev = ptr;
  struct fib_rules_ops *ops;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  ASSERT_RTNL();
  rcu_read_lock();

diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index ab48533..7796b39 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -1892,6 +1892,9 @@ static int pktgen_device_event(struct notifier_block *unused,
 {
  struct net_device *dev = (struct net_device *)(ptr);

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* It is OK that we do not hold the group lock right now,
   * as we run under the RTNL lock.
   */
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index e76539a..7841e89 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -829,6 +829,10 @@ static struct rtnetlink_link link_rtnetlink_table[RTM_NR_MSGTYPES] =
 static int rtnetlink_event(struct notifier_block *this, unsigned long event, void *ptr)
 {
  struct net_device *dev = ptr;
+
+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (event) {
  case NETDEV_UNREGISTER:
   rtmsg_ifinfo(RTM_DELLINK, dev, ~0U);
```

```
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index 5e8042f..b27b2ac 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -2086,6 +2086,9 @@ static int dn_device_event(struct notifier_block *this, unsigned long
event,
 {
  struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch(event) {
   case NETDEV_UP:
    dn_dev_up(dev);
diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index 0baffda..cbf87f4 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1121,6 +1121,9 @@ static int econet_notifier(struct notifier_block *this, unsigned long msg,
void
  struct net_device *dev = (struct net_device *)data;
  struct ec_device *edev;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (msg) {
  case NETDEV_UNREGISTER:
   /* A device has gone down - kill any data we hold for it. */
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index 95a34c7..0d23fb2 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -1206,6 +1206,9 @@ static int arp_netdev_event(struct notifier_block *this, unsigned long
event, vo
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (event) {
  case NETDEV_CHANGEADDR:
   neigh_changeaddr(&arp_tbl, dev);
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 216cf2b..a7d991d 100644
--- a/net/ipv4/devinet.c
```

```
+++ b/net/ipv4/devinet.c
@@ -1050,6 +1050,9 @@ static int inetdev_event(struct notifier_block *this, unsigned long event,
  struct net_device *dev = ptr;
  struct in_device *in_dev = __in_dev_get_rtnl(dev);

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  ASSERT_RTNL();

  if (!in_dev) {
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d47b72a..049c370 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -860,6 +860,9 @@ static int fib_netdev_event(struct notifier_block *this, unsigned long
event, vo
  struct net_device *dev = ptr;
  struct in_device *in_dev = __in_dev_get_rtnl(dev);

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_UNREGISTER) {
   fib_disable_ip(dev, 2);
   return NOTIFY_DONE;
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index af50394..9afaa13 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -1075,13 +1075,18 @@ int ipmr_ioctl(struct sock *sk, int cmd, void __user *arg)

 static int ipmr_device_event(struct notifier_block *this, unsigned long event, void *ptr)
 {
+ struct net_device *dev = ptr;
  struct vif_device *v;
  int ct;
+
+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event != NETDEV_UNREGISTER)
   return NOTIFY_DONE;
  v=&vif_table[0];
  for(ct=0;ct<maxvif;ct++,v++) {
-  if (v->dev==ptr)
+  if (v->dev==dev)
    vif_delete(ct);
```

```
  }
  return NOTIFY_DONE;
diff --git a/net/ipv4/multipath_drr.c b/net/ipv4/multipath_drr.c
index 252e837..b14d6ae 100644
--- a/net/ipv4/multipath_drr.c
+++ b/net/ipv4/multipath_drr.c
@@ -87,6 +87,9 @@ static int drr_dev_event(struct notifier_block *this,
  struct net_device *dev = ptr;
  int devidx;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (event) {
  case NETDEV_UNREGISTER:
  case NETDEV_DOWN:
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index aae660c..8650a57 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -567,6 +567,9 @@ ipq_rcv_dev_event(struct notifier_block *this,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* Drop any packets associated with the downed device */
  if (event == NETDEV_DOWN)
   ipq_dev_drop(dev->ifindex);
diff --git a/net/ipv4/netfilter/ipt_MASQUERADE.c b/net/ipv4/netfilter/ipt_MASQUERADE.c
index d669685..41fe6b5 100644
--- a/net/ipv4/netfilter/ipt_MASQUERADE.c
+++ b/net/ipv4/netfilter/ipt_MASQUERADE.c
@@ -152,6 +152,9 @@ static int masq_device_event(struct notifier_block *this,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_DOWN) {
   /* Device was downed.  Search entire table for
      conntracks which were associated with that device,
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 52bd4dd..7be542f 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
```

```
@@ -2235,6 +2235,9 @@ static int addrconf_notify(struct notifier_block *this, unsigned long
event,
  struct inet6_dev *idev = __in6_dev_get(dev);
  int run_pending = 0;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch(event) {
  case NETDEV_UP:
  case NETDEV_CHANGE:
diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index 6a9f616..9b3495f 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -1586,6 +1586,9 @@ static int ndisc_netdev_event(struct notifier_block *this, unsigned long
event,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (event) {
  case NETDEV_CHANGEADDR:
   neigh_changeaddr(&nd_tbl, dev);
diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index 45b64a5..f6e108c 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -557,6 +557,9 @@ ipq_rcv_dev_event(struct notifier_block *this,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* Drop any packets associated with the downed device */
  if (event == NETDEV_DOWN)
   ipq_dev_drop(dev->ifindex);
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index 5c5f2cd..f2674fe 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -347,6 +347,9 @@ static int ipxitf_device_event(struct notifier_block *notifier,
  struct net_device *dev = ptr;
  struct ipx_interface *i, *tmp;
```

```
+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event != NETDEV_DOWN && event != NETDEV_UP)
   goto out;

diff --git a/net/netfilter/nfnetlink_queue.c b/net/netfilter/nfnetlink_queue.c
index a88a017..59bf595 100644
--- a/net/netfilter/nfnetlink_queue.c
+++ b/net/netfilter/nfnetlink_queue.c
@@ -734,6 +734,9 @@ nfqnl_rcv_dev_event(struct notifier_block *this,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  /* Drop any packets associated with the downed device */
  if (event == NETDEV_DOWN)
   nfqnl_dev_drop(dev->ifindex);
diff --git a/net/netrom/af_netrom.c b/net/netrom/af_netrom.c
index 3fa3f1a..6965a1a 100644
--- a/net/netrom/af_netrom.c
+++ b/net/netrom/af_netrom.c
@@ -106,6 +106,9 @@ static int nr_device_event(struct notifier_block *this, unsigned long event,
voi
 {
  struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event != NETDEV_DOWN)
   return NOTIFY_DONE;

diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index aa298c3..6e3b947 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -1439,6 +1439,9 @@ static int packet_notifier(struct notifier_block *this, unsigned long msg,
void
  struct hlist_node *node;
  struct net_device *dev = (struct net_device*)data;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  read_lock(&packet_sklist_lock);
```

```
  sk_for_each(sk, node, &packet_sklist) {
   struct packet_sock *po = pkt_sk(sk);
diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index 7d5e593..dad50d3 100644
--- a/net/rose/af_rose.c
+++ b/net/rose/af_rose.c
@@ -197,6 +197,9 @@ static int rose_device_event(struct notifier_block *this, unsigned long
event,
 {
  struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event != NETDEV_DOWN)
   return NOTIFY_DONE;

diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index b181cf9..c6f64de 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -194,6 +194,9 @@ static int recv_notification(struct notifier_block *nb, unsigned long evt,
  struct eth_bearer *eb_ptr = &eth_bearers[0];
  struct eth_bearer *stop = &eth_bearers[MAX_ETH_BEARERS];

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  while ((eb_ptr->dev != dev)) {
   if (++eb_ptr == stop)
    return NOTIFY_DONE; /* couldn't find device */
diff --git a/net/wanrouter/af_wanpipe.c b/net/wanrouter/af_wanpipe.c
index 542c737..f9b896c 100644
--- a/net/wanrouter/af_wanpipe.c
+++ b/net/wanrouter/af_wanpipe.c
@@ -1714,6 +1714,9 @@ static int wanpipe_notifier(struct notifier_block *this, unsigned long
msg, void
  hlist_node *node;
  struct net_device *dev = (struct net_device *)data;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  sk_for_each(sk, node, &wanpipe_sklist) {
   struct wanpipe_opt *po = wp_sk(sk);

diff --git a/net/x25/af_x25.c b/net/x25/af_x25.c
index 6602a34..f4fad10 100644
```

```
--- a/net/x25/af_x25.c
+++ b/net/x25/af_x25.c
@@ -190,6 +190,9 @@ static int x25_device_event(struct notifier_block *this, unsigned long
event,
  struct net_device *dev = ptr;
  struct x25_neigh *nb;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (dev->type == ARPHRD_X25
 #if defined(CONFIG_LLC) || defined(CONFIG_LLC_MODULE)
   || dev->type == ARPHRD_ETHER
diff --git a/net/xfrm/xfrm_policy.c b/net/xfrm/xfrm_policy.c
index bebd40e..0248343 100644
--- a/net/xfrm/xfrm_policy.c
+++ b/net/xfrm/xfrm_policy.c
@@ -2188,6 +2188,11 @@ static void xfrm_policy_unlock_afinfo(struct xfrm_policy_afinfo
*afinfo)

 static int xfrm_dev_event(struct notifier_block *this, unsigned long event, void *ptr)
 {
+ struct net_device *dev = ptr;
+
+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  switch (event) {
  case NETDEV_DOWN:
   xfrm_flush_bundles();
diff --git a/security/selinux/netif.c b/security/selinux/netif.c
index b10c34e..45c422f 100644
--- a/security/selinux/netif.c
+++ b/security/selinux/netif.c
@@ -234,6 +234,9 @@ static int sel_netif_netdev_notifier_handler(struct notifier_block *this,
 {
  struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+  return NOTIFY_DONE;
+
  if (event == NETDEV_DOWN)
   sel_netif_kill(dev);

--
1.4.4.1.g278f
```

Subject: [PATCH RFC 14/31] net: Support multiple network namespaces with netlink
Posted by ebiederm on Thu, 25 Jan 2007 19:00:16 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Each netlink socket will live in exactly one network namespace,
this includes the controlling kernel sockets.

This patch updates all of the existing netlink protocols
to only support the initial network namespace. Request
by clients in other namespaces will get -ECONREFUSED.
As they would if the kernel did not have the support for
that netlink protocol compiled in.

As each netlink protocol is updated to be multiple network
namespace safe it can register multiple kernel sockets
to acquire a presence in the rest of the network namespaces.

The implementation in af_netlink is a simple filter implemenation
at hash table insertion and hash table look up time.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/scsi/scsi_netlink.c | 2 +-
drivers/scsi/scsi_transport_iscsi.c | 2 +-
include/linux/netlink.h | 3 +-
kernel/audit.c | 4 +-
lib/kobject_uevent.c | 4 +-
net/bridge/netfilter/ebt_ulog.c | 5 +-
net/core/rtnetlink.c | 4 +-
net/decnet/netfilter/dn_rtmsg.c | 3 +-
net/ipv4/fib_frontend.c | 3 +-
net/ipv4/inet_diag.c | 4 +-
net/ipv4/netfilter/ip_queue.c | 6 +-
net/ipv4/netfilter/ipt_ULOG.c | 4 +-
net/ipv6/netfilter/ip6_queue.c | 4 +-
net/netfilter/nfnetlink.c | 2 +-
net/netfilter/nfnetlink_log.c | 3 +-
net/netfilter/nfnetlink_queue.c | 3 +-
net/netlink/af_netlink.c | 104 ++++++++++++++++++++++++++++++++---------
net/netlink/genetlink.c | 4 +-

```
 net/xfrm/xfrm_user.c               |    2 +-
 19 files changed, 112 insertions(+), 54 deletions(-)

diff --git a/drivers/scsi/scsi_netlink.c b/drivers/scsi/scsi_netlink.c
index 1b59b27..02c2c1e 100644
--- a/drivers/scsi/scsi_netlink.c
+++ b/drivers/scsi/scsi_netlink.c
@@ -167,7 +167,7 @@ scsi_netlink_init(void)
   return;
  }

- scsi_nl_sock = netlink_kernel_create(NETLINK_SCSITRANSPORT,
+ scsi_nl_sock = netlink_kernel_create(init_net(), NETLINK_SCSITRANSPORT,
     SCSI_NL_GRP_CNT, scsi_nl_rcv, THIS_MODULE);
  if (!scsi_nl_sock) {
   printk(KERN_ERR "%s: register of recieve handler failed\n",
diff --git a/drivers/scsi/scsi_transport_iscsi.c b/drivers/scsi/scsi_transport_iscsi.c
index 9c22f13..1ad22c2 100644
--- a/drivers/scsi/scsi_transport_iscsi.c
+++ b/drivers/scsi/scsi_transport_iscsi.c
@@ -1435,7 +1435,7 @@ static __init int iscsi_transport_init(void)
  if (err)
   goto unregister_conn_class;

- nls = netlink_kernel_create(NETLINK_ISCSI, 1, iscsi_if_rx,
+ nls = netlink_kernel_create(init_net(), NETLINK_ISCSI, 1, iscsi_if_rx,
    THIS_MODULE);
  if (!nls) {
   err = -ENOBUFS;
diff --git a/include/linux/netlink.h b/include/linux/netlink.h
index b3b9b60..9dacd00 100644
--- a/include/linux/netlink.h
+++ b/include/linux/netlink.h
@@ -151,7 +151,7 @@ struct netlink_skb_parms
 #define NETLINK_CREDS(skb) (&NETLINK_CB((skb)).creds)


-extern struct sock *netlink_kernel_create(int unit, unsigned int groups, void (*input)(struct sock
*sk, int len), struct module *module);
+extern struct sock *netlink_kernel_create(net_t net, int unit, unsigned int groups, void
(*input)(struct sock *sk, int len), struct module *module);
 extern void netlink_ack(struct sk_buff *in_skb, struct nlmsghdr *nlh, int err);
 extern int netlink_has_listeners(struct sock *sk, unsigned int group);
 extern int netlink_unicast(struct sock *ssk, struct sk_buff *skb, __u32 pid, int nonblock);
@@ -188,6 +188,7 @@ struct netlink_callback

 struct netlink_notify
 {
```

```
+ net_t net;
  int pid;
  int protocol;
 };
diff --git a/kernel/audit.c b/kernel/audit.c
index d9b690a..b0c5c61 100644
--- a/kernel/audit.c
+++ b/kernel/audit.c
@@ -696,8 +696,8 @@ static int __init audit_init(void)

  printk(KERN_INFO "audit: initializing netlink socket (%s)\n",
         audit_default ? "enabled" : "disabled");
- audit_sock = netlink_kernel_create(NETLINK_AUDIT, 0, audit_receive,
-         THIS_MODULE);
+ audit_sock = netlink_kernel_create(init_net(), NETLINK_AUDIT, 0,
+         audit_receive, THIS_MODULE);
  if (!audit_sock)
   audit_panic("cannot initialize netlink socket");
  else
diff --git a/lib/kobject_uevent.c b/lib/kobject_uevent.c
index 84272ed..9a5d4ca 100644
--- a/lib/kobject_uevent.c
+++ b/lib/kobject_uevent.c
@@ -292,8 +292,8 @@ EXPORT_SYMBOL_GPL(add_uevent_var);
 #if defined(CONFIG_NET)
 static int __init kobject_uevent_init(void)
 {
- uevent_sock = netlink_kernel_create(NETLINK_KOBJECT_UEVENT, 1, NULL,
-         THIS_MODULE);
+ uevent_sock = netlink_kernel_create(init_net(), NETLINK_KOBJECT_UEVENT, 1,
+         NULL, THIS_MODULE);

  if (!uevent_sock) {
   printk(KERN_ERR
diff --git a/net/bridge/netfilter/ebt_ulog.c b/net/bridge/netfilter/ebt_ulog.c
index c1af68b..abf2be7 100644
--- a/net/bridge/netfilter/ebt_ulog.c
+++ b/net/bridge/netfilter/ebt_ulog.c
@@ -301,8 +301,9 @@ static int __init ebt_ulog_init(void)
   spin_lock_init(&ulog_buffers[i].lock);
  }

- ebtulognl = netlink_kernel_create(NETLINK_NFLOG, EBT_ULOG_MAXNLGROUPS,
-                                   NULL, THIS_MODULE);
+ ebtulognl = netlink_kernel_create(init_net(), NETLINK_NFLOG,
+       EBT_ULOG_MAXNLGROUPS, NULL,
+       THIS_MODULE);
  if (!ebtulognl)
```

```
   ret = -ENOMEM;
  else if ((ret = ebt_register_watcher(&ulog)))
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index 7841e89..8f3dda8 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -870,8 +870,8 @@ void __init rtnetlink_init(void)
  if (!rta_buf)
   panic("rtnetlink_init: cannot allocate rta_buf\n");

- rtnl = netlink_kernel_create(NETLINK_ROUTE, RTNLGRP_MAX, rtnetlink_rcv,
-                              THIS_MODULE);
+ rtnl = netlink_kernel_create(init_net(), NETLINK_ROUTE, RTNLGRP_MAX,
+          rtnetlink_rcv, THIS_MODULE);
  if (rtnl == NULL)
   panic("rtnetlink_init: cannot initialize rtnetlink\n");
  netlink_set_nonroot(NETLINK_ROUTE, NL_NONROOT_RECV);
diff --git a/net/decnet/netfilter/dn_rtmsg.c b/net/decnet/netfilter/dn_rtmsg.c
index 8b99bd3..14089ed 100644
--- a/net/decnet/netfilter/dn_rtmsg.c
+++ b/net/decnet/netfilter/dn_rtmsg.c
@@ -137,7 +137,8 @@ static int __init dn_rtmsg_init(void)
 {
  int rv = 0;

- dnrmg = netlink_kernel_create(NETLINK_DNRTMSG, DNRNG_NLGRP_MAX,
+ dnrmg = netlink_kernel_create(init_net(),
+          NETLINK_DNRTMSG, DNRNG_NLGRP_MAX,
                                dnrmg_receive_user_sk, THIS_MODULE);
  if (dnrmg == NULL) {
   printk(KERN_ERR "dn_rtmsg: Cannot create netlink socket");
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 049c370..d1859ff 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -817,7 +817,8 @@ static void nl_fib_input(struct sock *sk, int len)

 static void nl_fib_lookup_init(void)
 {
-      netlink_kernel_create(NETLINK_FIB_LOOKUP, 0, nl_fib_input, THIS_MODULE);
+ netlink_kernel_create(init_net(), NETLINK_FIB_LOOKUP, 0, nl_fib_input,
+         THIS_MODULE);
 }

 static void fib_disable_ip(struct net_device *dev, int force)
diff --git a/net/ipv4/inet_diag.c b/net/ipv4/inet_diag.c
index 77761ac..bdf3064 100644
--- a/net/ipv4/inet_diag.c
```

```
+++ b/net/ipv4/inet_diag.c
@@ -913,8 +913,8 @@ static int __init inet_diag_init(void)
  if (!inet_diag_table)
   goto out;

- idiagnl = netlink_kernel_create(NETLINK_INET_DIAG, 0, inet_diag_rcv,
-     THIS_MODULE);
+ idiagnl = netlink_kernel_create(init_net(), NETLINK_INET_DIAG, 0,
+     inet_diag_rcv, THIS_MODULE);
  if (idiagnl == NULL)
   goto out_free_table;
  err = 0;
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index 8650a57..d1c42b5 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -589,7 +589,7 @@ ipq_rcv_nl_event(struct notifier_block *this,
  if (event == NETLINK_URELEASE &&
      n->protocol == NETLINK_FIREWALL && n->pid) {
   write_lock_bh(&queue_lock);
-  if (n->pid == peer_pid)
+  if (net_eq(n->net, init_net()) && (n->pid == peer_pid))
    __ipq_reset();
   write_unlock_bh(&queue_lock);
  }
@@ -681,8 +681,8 @@ static int __init ip_queue_init(void)
  struct proc_dir_entry *proc;

  netlink_register_notifier(&ipq_nl_notifier);
- ipqnl = netlink_kernel_create(NETLINK_FIREWALL, 0, ipq_rcv_sk,
-          THIS_MODULE);
+ ipqnl = netlink_kernel_create(init_net(), NETLINK_FIREWALL, 0,
+          ipq_rcv_sk, THIS_MODULE);
  if (ipqnl == NULL) {
   printk(KERN_ERR "ip_queue: failed to create netlink socket\n");
   goto cleanup_netlink_notifier;
diff --git a/net/ipv4/netfilter/ipt_ULOG.c b/net/ipv4/netfilter/ipt_ULOG.c
index dbd3478..8071d15 100644
--- a/net/ipv4/netfilter/ipt_ULOG.c
+++ b/net/ipv4/netfilter/ipt_ULOG.c
@@ -395,8 +395,8 @@ static int __init ipt_ulog_init(void)
   ulog_buffers[i].timer.data = i;
  }

- nflognl = netlink_kernel_create(NETLINK_NFLOG, ULOG_MAXNLGROUPS, NULL,
-                                 THIS_MODULE);
+ nflognl = netlink_kernel_create(init_net(), NETLINK_NFLOG,
+     ULOG_MAXNLGROUPS, NULL, THIS_MODULE);
```

```
  if (!nflognl)
   return -ENOMEM;

diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index f6e108c..02589b2 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -579,7 +579,7 @@ ipq_rcv_nl_event(struct notifier_block *this,
  if (event == NETLINK_URELEASE &&
      n->protocol == NETLINK_IP6_FW && n->pid) {
   write_lock_bh(&queue_lock);
-  if (n->pid == peer_pid)
+  if (net_eq(n->net, init_net()) && (n->pid == peer_pid))
    __ipq_reset();
   write_unlock_bh(&queue_lock);
  }
@@ -671,7 +671,7 @@ static int __init ip6_queue_init(void)
  struct proc_dir_entry *proc;

  netlink_register_notifier(&ipq_nl_notifier);
- ipqnl = netlink_kernel_create(NETLINK_IP6_FW, 0, ipq_rcv_sk,
+ ipqnl = netlink_kernel_create(init_net(), NETLINK_IP6_FW, 0, ipq_rcv_sk,
                               THIS_MODULE);
  if (ipqnl == NULL) {
   printk(KERN_ERR "ip6_queue: failed to create netlink socket\n");
diff --git a/net/netfilter/nfnetlink.c b/net/netfilter/nfnetlink.c
index 52fdfa2..180353f 100644
--- a/net/netfilter/nfnetlink.c
+++ b/net/netfilter/nfnetlink.c
@@ -356,7 +356,7 @@ static int __init nfnetlink_init(void)
 {
  printk("Netfilter messages via NETLINK v%s.\n", nfversion);

- nfnl = netlink_kernel_create(NETLINK_NETFILTER, NFNLGRP_MAX,
+ nfnl = netlink_kernel_create(init_net(), NETLINK_NETFILTER, NFNLGRP_MAX,
                              nfnetlink_rcv, THIS_MODULE);
  if (!nfnl) {
   printk(KERN_ERR "cannot initialize nfnetlink!\n");
diff --git a/net/netfilter/nfnetlink_log.c b/net/netfilter/nfnetlink_log.c
index d1505dd..1bdf9af 100644
--- a/net/netfilter/nfnetlink_log.c
+++ b/net/netfilter/nfnetlink_log.c
@@ -741,7 +741,8 @@ nfulnl_rcv_nl_event(struct notifier_block *this,

    hlist_for_each_entry_safe(inst, tmp, t2, head, hlist) {
     UDEBUG("node = %p\n", inst);
-    if (n->pid == inst->peer_pid)
+    if (net_eq(n->net, init_net()) &&
```

```
+       (n->pid == inst->peer_pid))
      __instance_destroy(inst);
    }
   }
diff --git a/net/netfilter/nfnetlink_queue.c b/net/netfilter/nfnetlink_queue.c
index 59bf595..8e0c289 100644
--- a/net/netfilter/nfnetlink_queue.c
+++ b/net/netfilter/nfnetlink_queue.c
@@ -765,7 +765,8 @@ nfqnl_rcv_nl_event(struct notifier_block *this,
    struct hlist_head *head = &instance_table[i];

    hlist_for_each_entry_safe(inst, tmp, t2, head, hlist) {
-    if (n->pid == inst->peer_pid)
+    if (net_eq(n->net, init_net()) &&
+        (n->pid == inst->peer_pid))
      __instance_destroy(inst);
    }
   }
diff --git a/net/netlink/af_netlink.c b/net/netlink/af_netlink.c
index 7433e71..37b6c28 100644
--- a/net/netlink/af_netlink.c
+++ b/net/netlink/af_netlink.c
@@ -200,7 +200,7 @@ netlink_unlock_table(void)
   wake_up(&nl_table_wait);
 }

-static __inline__ struct sock *netlink_lookup(int protocol, u32 pid)
+static __inline__ struct sock *netlink_lookup(net_t net, int protocol, u32 pid)
 {
  struct nl_pid_hash *hash = &nl_table[protocol].hash;
  struct hlist_head *head;
@@ -210,7 +210,7 @@ static __inline__ struct sock *netlink_lookup(int protocol, u32 pid)
  read_lock(&nl_table_lock);
  head = nl_pid_hashfn(hash, pid);
  sk_for_each(sk, node, head) {
-  if (nlk_sk(sk)->pid == pid) {
+  if (net_eq(sk->sk_net, net) && (nlk_sk(sk)->pid == pid)) {
    sock_hold(sk);
    goto found;
   }
@@ -315,7 +315,7 @@ netlink_update_listeners(struct sock *sk)
   * makes sure updates are visible before bind or setsockopt return. */
 }

-static int netlink_insert(struct sock *sk, u32 pid)
+static int netlink_insert(struct sock *sk, net_t net, u32 pid)
 {
  struct nl_pid_hash *hash = &nl_table[sk->sk_protocol].hash;
```

```
  struct hlist_head *head;
@@ -328,7 +328,7 @@ static int netlink_insert(struct sock *sk, u32 pid)
  head = nl_pid_hashfn(hash, pid);
  len = 0;
  sk_for_each(osk, node, head) {
-  if (nlk_sk(osk)->pid == pid)
+  if (net_eq(osk->sk_net, net) && (nlk_sk(osk)->pid == pid))
     break;
   len++;
  }
@@ -400,9 +400,6 @@ static int netlink_create(net_t net, struct socket *sock, int protocol)
  unsigned int groups;
  int err = 0;

- if (!net_eq(net, init_net()))
-  return -EAFNOSUPPORT;
-
  sock->state = SS_UNCONNECTED;

  if (sock->type != SOCK_RAW && sock->type != SOCK_DGRAM)
@@ -469,6 +466,7 @@ static int netlink_release(struct socket *sock)

  if (nlk->pid && !nlk->subscriptions) {
   struct netlink_notify n = {
+       .net = sk->sk_net,
        .protocol = sk->sk_protocol,
        .pid = nlk->pid,
        };
@@ -497,6 +495,7 @@ static int netlink_release(struct socket *sock)
 static int netlink_autobind(struct socket *sock)
 {
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct nl_pid_hash *hash = &nl_table[sk->sk_protocol].hash;
  struct hlist_head *head;
  struct sock *osk;
@@ -510,6 +509,8 @@ retry:
  netlink_table_grab();
  head = nl_pid_hashfn(hash, pid);
  sk_for_each(osk, node, head) {
+  if (!net_eq(osk->sk_net, net))
+   continue;
   if (nlk_sk(osk)->pid == pid) {
    /* Bind collision, search negative pid values. */
    pid = rover--;
@@ -521,7 +522,7 @@ retry:
  }
  netlink_table_ungrab();
```

```
- err = netlink_insert(sk, pid);
+ err = netlink_insert(sk, net, pid);
  if (err == -EADDRINUSE)
   goto retry;

@@ -575,6 +576,7 @@ static int netlink_alloc_groups(struct sock *sk)
 static int netlink_bind(struct socket *sock, struct sockaddr *addr, int addr_len)
 {
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct netlink_sock *nlk = nlk_sk(sk);
  struct sockaddr_nl *nladdr = (struct sockaddr_nl *)addr;
  int err;
@@ -598,7 +600,7 @@ static int netlink_bind(struct socket *sock, struct sockaddr *addr, int
addr_len
    return -EINVAL;
  } else {
   err = nladdr->nl_pid ?
-   netlink_insert(sk, nladdr->nl_pid) :
+   netlink_insert(sk, net, nladdr->nl_pid) :
    netlink_autobind(sock);
   if (err)
    return err;
@@ -682,10 +684,12 @@ static void netlink_overrun(struct sock *sk)
 static struct sock *netlink_getsockbypid(struct sock *ssk, u32 pid)
 {
  int protocol = ssk->sk_protocol;
+ net_t net;
  struct sock *sock;
  struct netlink_sock *nlk;

- sock = netlink_lookup(protocol, pid);
+ net = ssk->sk_net;
+ sock = netlink_lookup(net, protocol, pid);
  if (!sock)
   return ERR_PTR(-ECONNREFUSED);

@@ -858,6 +862,7 @@ static __inline__ int netlink_broadcast_deliver(struct sock *sk, struct
sk_buff

 struct netlink_broadcast_data {
  struct sock *exclude_sk;
+ net_t net;
  u32 pid;
  u32 group;
  int failure;
@@ -880,6 +885,9 @@ static inline int do_one_broadcast(struct sock *sk,
```

```
      !test_bit(p->group - 1, nlk->groups))
   goto out;

+ if (!net_eq(sk->sk_net, p->net))
+  goto out;
+
  if (p->failure) {
   netlink_overrun(sk);
   goto out;
@@ -918,6 +926,7 @@ out:
 int netlink_broadcast(struct sock *ssk, struct sk_buff *skb, u32 pid,
         u32 group, gfp_t allocation)
 {
+ net_t net = ssk->sk_net;
  struct netlink_broadcast_data info;
  struct hlist_node *node;
  struct sock *sk;
@@ -925,6 +934,7 @@ int netlink_broadcast(struct sock *ssk, struct sk_buff *skb, u32 pid,
  skb = netlink_trim(skb, allocation);

  info.exclude_sk = ssk;
+ info.net = net;
  info.pid = pid;
  info.group = group;
  info.failure = 0;
@@ -973,6 +983,9 @@ static inline int do_one_set_err(struct sock *sk,
  if (sk == p->exclude_sk)
   goto out;

+ if (!net_eq(sk->sk_net, p->exclude_sk->sk_net))
+  goto out;
+
  if (nlk->pid == p->pid || p->group - 1 >= nlk->ngroups ||
      !test_bit(p->group - 1, nlk->groups))
   goto out;
@@ -1267,7 +1280,7 @@ static void netlink_data_ready(struct sock *sk, int len)
  */

 struct sock *
-netlink_kernel_create(int unit, unsigned int groups,
+netlink_kernel_create(net_t net, int unit, unsigned int groups,
                       void (*input)(struct sock *sk, int len),
                       struct module *module)
 {
@@ -1284,7 +1297,7 @@ netlink_kernel_create(int unit, unsigned int groups,
  if (sock_create_lite(PF_NETLINK, SOCK_DGRAM, unit, &sock))
   return NULL;
```

```
- if (__netlink_create(init_net(), sock, unit) < 0)
+ if (__netlink_create(net, sock, unit) < 0)
   goto out_sock_release;

  if (groups < 32)
@@ -1299,17 +1312,19 @@ netlink_kernel_create(int unit, unsigned int groups,
  if (input)
   nlk_sk(sk)->data_ready = input;

- if (netlink_insert(sk, 0))
+ if (netlink_insert(sk, net, 0))
   goto out_sock_release;

  nlk = nlk_sk(sk);
  nlk->flags |= NETLINK_KERNEL_SOCKET;

  netlink_table_grab();
- nl_table[unit].groups = groups;
- nl_table[unit].listeners = listeners;
- nl_table[unit].module = module;
- nl_table[unit].registered = 1;
+ if (!nl_table[unit].registered) {
+  nl_table[unit].groups = groups;
+  nl_table[unit].listeners = listeners;
+  nl_table[unit].module = module;
+  nl_table[unit].registered = 1;
+ }
  netlink_table_ungrab();

  return sk;
@@ -1410,7 +1425,7 @@ int netlink_dump_start(struct sock *ssk, struct sk_buff *skb,
  atomic_inc(&skb->users);
  cb->skb = skb;

- sk = netlink_lookup(ssk->sk_protocol, NETLINK_CB(skb).pid);
+ sk = netlink_lookup(ssk->sk_net, ssk->sk_protocol, NETLINK_CB(skb).pid);
  if (sk == NULL) {
   netlink_destroy_callback(cb);
   return -ECONNREFUSED;
@@ -1447,7 +1462,8 @@ void netlink_ack(struct sk_buff *in_skb, struct nlmsghdr *nlh, int err)
  if (!skb) {
   struct sock *sk;

-  sk = netlink_lookup(in_skb->sk->sk_protocol,
+  sk = netlink_lookup(in_skb->sk->sk_net,
+        in_skb->sk->sk_protocol,
         NETLINK_CB(in_skb).pid);
   if (sk) {
```

```
    sk->sk_err = ENOBUFS;
@@ -1585,6 +1601,7 @@ int nlmsg_notify(struct sock *sk, struct sk_buff *skb, u32 pid,

 #ifdef CONFIG_PROC_FS
 struct nl_seq_iter {
+ net_t net;
  int link;
  int hash_idx;
 };
@@ -1602,6 +1619,8 @@ static struct sock *netlink_seq_socket_idx(struct seq_file *seq, loff_t
pos)

   for (j = 0; j <= hash->mask; j++) {
    sk_for_each(s, node, &hash->table[j]) {
+    if (!net_eq(iter->net, s->sk_net))
+     continue;
     if (off == pos) {
      iter->link = i;
      iter->hash_idx = j;
@@ -1630,12 +1649,15 @@ static void *netlink_seq_next(struct seq_file *seq, void *v, loff_t
*pos)

  if (v == SEQ_START_TOKEN)
   return netlink_seq_socket_idx(seq, 0);
-
- s = sk_next(v);
+
+ iter = seq->private;
+ s = v;
+ do {
+  s = sk_next(s);
+ } while (s && !net_eq(iter->net, s->sk_net));
  if (s)
   return s;

- iter = seq->private;
  i = iter->link;
  j = iter->hash_idx + 1;

@@ -1644,6 +1666,8 @@ static void *netlink_seq_next(struct seq_file *seq, void *v, loff_t *pos)

   for (; j <= hash->mask; j++) {
    s = sk_head(&hash->table[j]);
+   while (s && !net_eq(iter->net, s->sk_net))
+    s = sk_next(s);
    if (s) {
     iter->link = i;
     iter->hash_idx = j;
```

```
@@ -1714,15 +1738,24 @@ static int netlink_seq_open(struct inode *inode, struct file *file)

  seq = file->private_data;
  seq->private = iter;
+ iter->net = get_net(PROC_NET(inode));
  return 0;
 }

+static int netlink_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ struct nl_seq_iter *iter = seq->private;
+ put_net(iter->net);
+ return seq_release_private(inode, file);
+}
+
 static struct file_operations netlink_seq_fops = {
  .owner  = THIS_MODULE,
  .open  = netlink_seq_open,
  .read  = seq_read,
  .llseek  = seq_lseek,
- .release = seq_release_private,
+ .release = netlink_seq_release,
 };

 #endif
@@ -1764,6 +1797,27 @@ static struct net_proto_family netlink_family_ops = {
  .owner = THIS_MODULE, /* for consistency 8) */
 };

+static int netlink_net_init(net_t net)
+{
+#ifdef CONFIG_PROC_FS
+ if (!proc_net_fops_create(net, "netlink", 0, &netlink_seq_fops))
+  return -ENOMEM;
+#endif
+ return 0;
+}
+
+static void netlink_net_exit(net_t net)
+{
+#ifdef CONFIG_PROC_FS
+ proc_net_remove(net, "netlink");
+#endif
+}
+
+static struct pernet_operations netlink_net_ops = {
+ .init = netlink_net_init,
```

```
+ .exit = netlink_net_exit,
+};
+
 static int __init netlink_proto_init(void)
 {
  struct sk_buff *dummy_skb;
@@ -1809,9 +1863,7 @@ static int __init netlink_proto_init(void)
  }

  sock_register(&netlink_family_ops);
-#ifdef CONFIG_PROC_FS
- proc_net_fops_create(init_net(), "netlink", 0, &netlink_seq_fops);
-#endif
+ register_pernet_subsys(&netlink_net_ops);
  /* The netlink device handler may be needed early. */
  rtnetlink_init();
 out:
diff --git a/net/netlink/genetlink.c b/net/netlink/genetlink.c
index 548e4e6..d6717e8 100644
--- a/net/netlink/genetlink.c
+++ b/net/netlink/genetlink.c
@@ -585,8 +585,8 @@ static int __init genl_init(void)
   goto errout_register;

  netlink_set_nonroot(NETLINK_GENERIC, NL_NONROOT_RECV);
- genl_sock = netlink_kernel_create(NETLINK_GENERIC, GENL_MAX_ID,
-       genl_rcv, THIS_MODULE);
+ genl_sock = netlink_kernel_create(init_net(), NETLINK_GENERIC,
+       GENL_MAX_ID, genl_rcv, THIS_MODULE);
  if (genl_sock == NULL)
   panic("GENL: Cannot initialize generic netlink\n");

diff --git a/net/xfrm/xfrm_user.c b/net/xfrm/xfrm_user.c
index 82f36d3..55affa7 100644
--- a/net/xfrm/xfrm_user.c
+++ b/net/xfrm/xfrm_user.c
@@ -2293,7 +2293,7 @@ static int __init xfrm_user_init(void)

  printk(KERN_INFO "Initializing XFRM netlink socket\n");

- nlsk = netlink_kernel_create(NETLINK_XFRM, XFRMNLGRP_MAX,
+ nlsk = netlink_kernel_create(init_net(), NETLINK_XFRM, XFRMNLGRP_MAX,
                              xfrm_netlink_rcv, THIS_MODULE);
  if (nlsk == NULL)
   return -ENOMEM;
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 15/31] net: Make the loopback device per network namespace
Posted by ebiederm on Thu, 25 Jan 2007 19:00:17 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch makes the loopback_dev per network namespace.
The loopback device registers itself as a pernet_device so
we can register the new loopback_dev instance when we add
a new network namespace and so we can unregister the
loopback device when we destory the network namespace.

Currently the loopback device statitics are kept accross
all loopback devices, a minor glitch that will not affect
correct operation but something we may want to fix.

This patch modifies all users the loopback_dev so they
access it as per_net(loopback_dev, init_net()), keeping all of the
code compiling and working. A later pass will be needed to
update the users to use something other than the initial network
namespace.

The only non-trivial modification was the ipv6 code in route.c as the
loopback_dev can no longer be used in static initializers, and
even that change was very simple.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/net/loopback.c | 24 ++++++++++++++++++++----
include/linux/netdevice.h | 2 +-
net/core/dst.c | 8 ++++----
net/decnet/dn_dev.c | 4 ++--
net/decnet/dn_route.c | 14 +++++++-------
net/ipv4/devinet.c | 4 ++--
net/ipv4/ipconfig.c | 8 +++++---
net/ipv4/ipvs/ip_vs_core.c | 2 +-
net/ipv4/route.c | 18 +++++++++---------
net/ipv4/xfrm4_policy.c | 2 +-
net/ipv6/addrconf.c | 8 ++++----
net/ipv6/netfilter/ip6t_REJECT.c | 2 +-
net/ipv6/route.c | 24 ++++++++++++++---------

```
 net/ipv6/xfrm6_policy.c         |    2 +-
 net/xfrm/xfrm_policy.c          |    4 ++--
 15 files changed, 75 insertions(+), 51 deletions(-)

diff --git a/drivers/net/loopback.c b/drivers/net/loopback.c
index 22b672d..e9abf3f 100644
--- a/drivers/net/loopback.c
+++ b/drivers/net/loopback.c
@@ -57,6 +57,7 @@
 #include <linux/ip.h>
 #include <linux/tcp.h>
 #include <linux/percpu.h>
+#include <net/net_namespace.h>

 struct pcpu_lstats {
  unsigned long packets;
@@ -204,7 +205,7 @@ static const struct ethtool_ops loopback_ethtool_ops = {
  * The loopback device is special. There is only one instance and
  * it is statically allocated. Don't do this for other devices.
  */
-struct net_device loopback_dev = {
+DEFINE_PER_NET(struct net_device, loopback_dev) = {
  .name    = "lo",
  .get_stats  = &get_stats,
  .priv   = &loopback_stats,
@@ -228,13 +229,28 @@ struct net_device loopback_dev = {
  .ethtool_ops  = &loopback_ethtool_ops,
 };

+static int loopback_net_init(net_t net)
+{
+ per_net(loopback_dev, net).nd_net = net;
+ return register_netdev(&per_net(loopback_dev, net));
+}
+
+static void loopback_net_exit(net_t net)
+{
+ unregister_netdev(&per_net(loopback_dev, net));
+}
+
+static struct pernet_operations loopback_net_ops = {
+ .init = loopback_net_init,
+ .exit = loopback_net_exit,
+};
+
 /* Setup and register the loopback device. */
 static int __init loopback_init(void)
 {
```

```
- loopback_dev.nd_net = init_net();
- return register_netdev(&loopback_dev);
+ return register_pernet_device(&loopback_net_ops);
 };

 module_init(loopback_init);

-EXPORT_SYMBOL(loopback_dev);
+EXPORT_PER_NET_SYMBOL(loopback_dev);
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 9e28671..73931a0 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -570,7 +570,7 @@ struct packet_type {
 #include <linux/interrupt.h>
 #include <linux/notifier.h>

-extern struct net_device  loopback_dev;  /* The loopback */
+DECLARE_PER_NET(struct net_device, loopback_dev);  /* The loopback */
 extern struct net_device  *dev_base;  /* All devices */
 extern rwlock_t    dev_base_lock;  /* Device list lock */

diff --git a/net/core/dst.c b/net/core/dst.c
index 8c4a272..3435771 100644
--- a/net/core/dst.c
+++ b/net/core/dst.c
@@ -241,13 +241,13 @@ static inline void dst_ifdown(struct dst_entry *dst, struct net_device
*dev,
   dst->input = dst_discard_in;
   dst->output = dst_discard_out;
  } else {
-  dst->dev = &loopback_dev;
-  dev_hold(&loopback_dev);
+  dst->dev = &per_net(loopback_dev, init_net());
+  dev_hold(dst->dev);
   dev_put(dev);
   if (dst->neighbour && dst->neighbour->dev == dev) {
-   dst->neighbour->dev = &loopback_dev;
+   dst->neighbour->dev = &per_net(loopback_dev, init_net());
    dev_put(dev);
-   dev_hold(&loopback_dev);
+   dev_hold(dst->neighbour->dev);
   }
  }
 }
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index 19b1469..dbaf001 100644
--- a/net/decnet/dn_dev.c
```

```
+++ b/net/decnet/dn_dev.c
@@ -866,10 +866,10 @@ last_chance:
   rv = dn_dev_get_first(dev, addr);
   read_unlock(&dev_base_lock);
   dev_put(dev);
-  if (rv == 0 || dev == &loopback_dev)
+  if (rv == 0 || dev == &per_net(loopback_dev, init_net()))
    return rv;
  }
- dev = &loopback_dev;
+ dev = &per_net(loopback_dev, init_net());
  dev_hold(dev);
  goto last_chance;
 }
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 4263cd9..b553cd4 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -887,7 +887,7 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old
      .scope = RT_SCOPE_UNIVERSE,
         } },
        .mark = oldflp->mark,
-       .iif = loopback_dev.ifindex,
+       .iif = per_net(loopback_dev, init_net()).ifindex,
        .oif = oldflp->oif };
  struct dn_route *rt = NULL;
  struct net_device *dev_out = NULL;
@@ -904,7 +904,7 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old
          "dn_route_output_slow: dst=%04x src=%04x mark=%d"
          " iif=%d oif=%d\n", dn_ntohs(oldflp->fld_dst),
          dn_ntohs(oldflp->fld_src),
-                  oldflp->mark, loopback_dev.ifindex, oldflp->oif);
+                  oldflp->mark, per_net(loopback_dev, init_net()).ifindex, oldflp->oif);

  /* If we have an output interface, verify its a DECnet device */
  if (oldflp->oif) {
@@ -955,7 +955,7 @@ source_ok:
   err = -EADDRNOTAVAIL;
   if (dev_out)
    dev_put(dev_out);
-  dev_out = &loopback_dev;
+  dev_out = &per_net(loopback_dev, init_net());
   dev_hold(dev_out);
   if (!fl.fld_dst) {
    fl.fld_dst =
@@ -964,7 +964,7 @@ source_ok:
```

```
    if (!fl.fld_dst)
     goto out;
   }
-  fl.oif = loopback_dev.ifindex;
+  fl.oif = per_net(loopback_dev, init_net()).ifindex;
   res.type = RTN_LOCAL;
   goto make_route;
  }
@@ -1010,7 +1010,7 @@ source_ok:
      if (dev_out)
       dev_put(dev_out);
      if (dn_dev_islocal(neigh->dev, fl.fld_dst)) {
-      dev_out = &loopback_dev;
+      dev_out = &per_net(loopback_dev, init_net());
       res.type = RTN_LOCAL;
      } else {
       dev_out = neigh->dev;
@@ -1031,7 +1031,7 @@ source_ok:
   /* Possible improvement - check all devices for local addr */
   if (dn_dev_islocal(dev_out, fl.fld_dst)) {
    dev_put(dev_out);
-   dev_out = &loopback_dev;
+   dev_out = &per_net(loopback_dev, init_net());
    dev_hold(dev_out);
    res.type = RTN_LOCAL;
    goto select_source;
@@ -1067,7 +1067,7 @@ select_source:
    fl.fld_src = fl.fld_dst;
   if (dev_out)
    dev_put(dev_out);
-  dev_out = &loopback_dev;
+  dev_out = &per_net(loopback_dev, init_net());
   dev_hold(dev_out);
   fl.oif = dev_out->ifindex;
   if (res.fi)
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index a7d991d..201442c 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1056,7 +1056,7 @@ static int inetdev_event(struct notifier_block *this, unsigned long event,
  ASSERT_RTNL();

  if (!in_dev) {
-  if (event == NETDEV_REGISTER && dev == &loopback_dev) {
+  if (event == NETDEV_REGISTER && dev == &per_net(loopback_dev, init_net())) {
    in_dev = inetdev_init(dev);
    if (!in_dev)
     panic("devinet: Failed to create loopback\n");
```

```
@@ -1074,7 +1074,7 @@ static int inetdev_event(struct notifier_block *this, unsigned long event,
  case NETDEV_UP:
   if (dev->mtu < 68)
    break;
-  if (dev == &loopback_dev) {
+  if (dev == &per_net(loopback_dev, init_net())) {
    struct in_ifaddr *ifa;
    if ((ifa = inet_alloc_ifa()) != NULL) {
     ifa->ifa_local =
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index 91b5729..ee77938 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -185,16 +185,18 @@ static int __init ic_open_devs(void)
  struct ic_device *d, **last;
  struct net_device *dev;
  unsigned short oflags;
+ struct net_device *lo;

  last = &ic_first_dev;
  rtnl_lock();

  /* bring loopback device up first */
- if (dev_change_flags(&loopback_dev, loopback_dev.flags | IFF_UP) < 0)
-  printk(KERN_ERR "IP-Config: Failed to open %s\n", loopback_dev.name);
+ lo = &per_net(loopback_dev, init_net());
+ if (dev_change_flags(lo, lo->flags | IFF_UP) < 0)
+  printk(KERN_ERR "IP-Config: Failed to open %s\n", lo->name);

  for (dev = dev_base; dev; dev = dev->next) {
-  if (dev == &loopback_dev)
+  if (dev == lo)
    continue;
   if (user_dev_name[0] ? !strcmp(dev->name, user_dev_name) :
       (!(dev->flags & IFF_LOOPBACK) &&
diff --git a/net/ipv4/ipvs/ip_vs_core.c b/net/ipv4/ipvs/ip_vs_core.c
index 3425752..2e1e41f 100644
--- a/net/ipv4/ipvs/ip_vs_core.c
+++ b/net/ipv4/ipvs/ip_vs_core.c
@@ -963,7 +963,7 @@ ip_vs_in(unsigned int hooknum, struct sk_buff **pskb,
   * ... don't know why 1st test DOES NOT include 2nd (?)
   */
  if (unlikely(skb->pkt_type != PACKET_HOST
-       || skb->dev == &loopback_dev || skb->sk)) {
+       || skb->dev == &per_net(loopback_dev, init_net()) || skb->sk)) {
   IP_VS_DBG(12, "packet type=%d proto=%d daddr=%d.%d.%d.%d ignored\n",
     skb->pkt_type,
     skb->nh.iph->protocol,
```

```
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 8be7506..d23a0d7 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -1498,8 +1498,8 @@ static void ipv4_dst_ifdown(struct dst_entry *dst, struct net_device
*dev,
 {
  struct rtable *rt = (struct rtable *) dst;
  struct in_device *idev = rt->idev;
- if (dev != &loopback_dev && idev && idev->dev == dev) {
-  struct in_device *loopback_idev = in_dev_get(&loopback_dev);
+ if (dev != &per_net(loopback_dev, init_net()) && idev && idev->dev == dev) {
+  struct in_device *loopback_idev = in_dev_get(&per_net(loopback_dev, init_net()));
   if (loopback_idev) {
    rt->idev = loopback_idev;
    in_dev_put(idev);
@@ -1651,7 +1651,7 @@ static int ip_route_input_mc(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
 #endif
  rth->rt_iif =
  rth->fl.iif = dev->ifindex;
- rth->u.dst.dev = &loopback_dev;
+ rth->u.dst.dev = &per_net(loopback_dev, init_net());
  dev_hold(rth->u.dst.dev);
  rth->idev = in_dev_get(rth->u.dst.dev);
  rth->fl.oif = 0;
@@ -1969,7 +1969,7 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
  if (res.type == RTN_LOCAL) {
   int result;
   result = fib_validate_source(saddr, daddr, tos,
-          loopback_dev.ifindex,
+          per_net(loopback_dev, init_net()).ifindex,
           dev, &spec_dst, &itag);
   if (result < 0)
    goto martian_source;
@@ -2036,7 +2036,7 @@ local_input:
 #endif
  rth->rt_iif =
  rth->fl.iif = dev->ifindex;
- rth->u.dst.dev = &loopback_dev;
+ rth->u.dst.dev = &per_net(loopback_dev, init_net());
  dev_hold(rth->u.dst.dev);
  rth->idev = in_dev_get(rth->u.dst.dev);
  rth->rt_gateway = daddr;
@@ -2375,7 +2375,7 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)
        RT_SCOPE_UNIVERSE),
```

```
          } },
        .mark = oldflp->mark,
-       .iif = loopback_dev.ifindex,
+       .iif = per_net(loopback_dev, init_net()).ifindex,
        .oif = oldflp->oif };
  struct fib_result res;
  unsigned flags = 0;
@@ -2469,9 +2469,9 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)
    fl.fl4_dst = fl.fl4_src = htonl(INADDR_LOOPBACK);
   if (dev_out)
    dev_put(dev_out);
-  dev_out = &loopback_dev;
+  dev_out = &per_net(loopback_dev, init_net());
   dev_hold(dev_out);
-  fl.oif = loopback_dev.ifindex;
+  fl.oif = per_net(loopback_dev, init_net()).ifindex;
   res.type = RTN_LOCAL;
   flags |= RTCF_LOCAL;
   goto make_route;
@@ -2516,7 +2516,7 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)
    fl.fl4_src = fl.fl4_dst;
   if (dev_out)
    dev_put(dev_out);
-  dev_out = &loopback_dev;
+  dev_out = &per_net(loopback_dev, init_net());
   dev_hold(dev_out);
   fl.oif = dev_out->ifindex;
   if (res.fi)
diff --git a/net/ipv4/xfrm4_policy.c b/net/ipv4/xfrm4_policy.c
index fb9f69c..39a0ba2 100644
--- a/net/ipv4/xfrm4_policy.c
+++ b/net/ipv4/xfrm4_policy.c
@@ -289,7 +289,7 @@ static void xfrm4_dst_ifdown(struct dst_entry *dst, struct net_device
*dev,

  xdst = (struct xfrm_dst *)dst;
  if (xdst->u.rt.idev->dev == dev) {
-  struct in_device *loopback_idev = in_dev_get(&loopback_dev);
+  struct in_device *loopback_idev = in_dev_get(&per_net(loopback_dev, init_net()));
   BUG_ON(!loopback_idev);

   do {
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 7be542f..c9fa27a 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
```

```
@@ -2365,7 +2365,7 @@ static int addrconf_ifdown(struct net_device *dev, int how)

  ASSERT_RTNL();

- if (dev == &loopback_dev && how == 1)
+ if (dev == &per_net(loopback_dev, init_net()) && how == 1)
   how = 0;

  rt6_ifdown(dev);
@@ -4074,13 +4074,13 @@ int __init addrconf_init(void)
   * device and it being up should be removed.
   */
  rtnl_lock();
- if (!ipv6_add_dev(&loopback_dev))
+ if (!ipv6_add_dev(&per_net(loopback_dev, init_net())))
   err = -ENOMEM;
  rtnl_unlock();
  if (err)
   return err;

- ip6_null_entry.rt6i_idev = in6_dev_get(&loopback_dev);
+ ip6_null_entry.rt6i_idev = in6_dev_get(&per_net(loopback_dev, init_net()));

  register_netdevice_notifier(&ipv6_dev_notf);

@@ -4121,7 +4121,7 @@ void __exit addrconf_cleanup(void)
    continue;
   addrconf_ifdown(dev, 1);
  }
- addrconf_ifdown(&loopback_dev, 2);
+ addrconf_ifdown(&per_net(loopback_dev, init_net()), 2);

  /*
   * Check hash table.
diff --git a/net/ipv6/netfilter/ip6t_REJECT.c b/net/ipv6/netfilter/ip6t_REJECT.c
index 311eae8..a80bbee 100644
--- a/net/ipv6/netfilter/ip6t_REJECT.c
+++ b/net/ipv6/netfilter/ip6t_REJECT.c
@@ -170,7 +170,7 @@ static inline void
 send_unreach(struct sk_buff *skb_in, unsigned char code, unsigned int hooknum)
 {
  if (hooknum == NF_IP6_LOCAL_OUT && skb_in->dev == NULL)
-  skb_in->dev = &loopback_dev;
+  skb_in->dev = &per_net(loopback_dev, init_net());

  icmpv6_send(skb_in, ICMPV6_DEST_UNREACH, code, 0, NULL);
 }
diff --git a/net/ipv6/route.c b/net/ipv6/route.c
```

```
index 8c9fef9..6805c39 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -125,7 +125,7 @@ struct rt6_info ip6_null_entry = {
   .dst = {
    .__refcnt = ATOMIC_INIT(1),
    .__use  = 1,
-   .dev  = &loopback_dev,
+   .dev  = NULL,
    .obsolete = -1,
    .error  = -ENETUNREACH,
    .metrics = { [RTAX_HOPLIMIT - 1] = 255, },
@@ -151,7 +151,7 @@ struct rt6_info ip6_prohibit_entry = {
   .dst = {
    .__refcnt = ATOMIC_INIT(1),
    .__use  = 1,
-   .dev  = &loopback_dev,
+   .dev  = NULL,
    .obsolete = -1,
    .error  = -EACCES,
    .metrics = { [RTAX_HOPLIMIT - 1] = 255, },
@@ -171,7 +171,7 @@ struct rt6_info ip6_blk_hole_entry = {
   .dst = {
    .__refcnt = ATOMIC_INIT(1),
    .__use  = 1,
-   .dev  = &loopback_dev,
+   .dev  = NULL,
    .obsolete = -1,
    .error  = -EINVAL,
    .metrics = { [RTAX_HOPLIMIT - 1] = 255, },
@@ -211,8 +211,8 @@ static void ip6_dst_ifdown(struct dst_entry *dst, struct net_device *dev,
  struct rt6_info *rt = (struct rt6_info *)dst;
  struct inet6_dev *idev = rt->rt6i_idev;

- if (dev != &loopback_dev && idev != NULL && idev->dev == dev) {
-  struct inet6_dev *loopback_idev = in6_dev_get(&loopback_dev);
+ if (dev != &per_net(loopback_dev, init_net()) && idev != NULL && idev->dev == dev) {
+  struct inet6_dev *loopback_idev = in6_dev_get(&per_net(loopback_dev, init_net()));
   if (loopback_idev != NULL) {
    rt->rt6i_idev = loopback_idev;
    in6_dev_put(idev);
@@ -1103,12 +1103,12 @@ int ip6_route_add(struct fib6_config *cfg)
  if ((cfg->fc_flags & RTF_REJECT) ||
      (dev && (dev->flags&IFF_LOOPBACK) && !(addr_type&IPV6_ADDR_LOOPBACK))) {
   /* hold loopback dev/idev if we haven't done so. */
-  if (dev != &loopback_dev) {
+  if (dev != &per_net(loopback_dev, init_net())) {
    if (dev) {
```

```
     dev_put(dev);
     in6_dev_put(idev);
    }
-   dev = &loopback_dev;
+   dev = &per_net(loopback_dev, init_net());
    dev_hold(dev);
    idev = in6_dev_get(dev);
    if (!idev) {
@@ -1803,13 +1803,13 @@ struct rt6_info *addrconf_dst_alloc(struct inet6_dev *idev,
  if (rt == NULL)
   return ERR_PTR(-ENOMEM);

- dev_hold(&loopback_dev);
+ dev_hold(&per_net(loopback_dev, init_net()));
  in6_dev_hold(idev);

  rt->u.dst.flags = DST_HOST;
  rt->u.dst.input = ip6_input;
  rt->u.dst.output = ip6_output;
- rt->rt6i_dev = &loopback_dev;
+ rt->rt6i_dev = &per_net(loopback_dev, init_net());
  rt->rt6i_idev = idev;
  rt->u.dst.metrics[RTAX_MTU-1] = ipv6_get_mtu(rt->rt6i_dev);
  rt->u.dst.metrics[RTAX_ADVMSS-1] = ipv6_advmss(dst_mtu(&rt->u.dst));
@@ -2457,6 +2457,12 @@ void __init ip6_route_init(void)
  ip6_dst_ops.kmem_cachep =
   kmem_cache_create("ip6_dst_cache", sizeof(struct rt6_info), 0,
       SLAB_HWCACHE_ALIGN|SLAB_PANIC, NULL, NULL);
+ /* Perform the initialization we can't perform at compile time */
+ ip6_null_entry.u.dst.dev = &per_net(loopback_dev, init_net());
+#ifdef CONFIG_IPV6_MULTIPLE_TABLES
+ ip6_prohibit_entry.u.dst.dev = &per_net(loopback_dev, init_net());
+ ip6_blk_hole_entry.u.dst.dev = &per_net(loopback_dev, init_net());
+#endif
  fib6_init();
 #ifdef  CONFIG_PROC_FS
  p = proc_net_create(init_net(), "ipv6_route", 0, rt6_proc_info);
diff --git a/net/ipv6/xfrm6_policy.c b/net/ipv6/xfrm6_policy.c
index 8dffd4d..2608c75 100644
--- a/net/ipv6/xfrm6_policy.c
+++ b/net/ipv6/xfrm6_policy.c
@@ -354,7 +354,7 @@ static void xfrm6_dst_ifdown(struct dst_entry *dst, struct net_device
*dev,

  xdst = (struct xfrm_dst *)dst;
  if (xdst->u.rt6.rt6i_idev->dev == dev) {
-  struct inet6_dev *loopback_idev = in6_dev_get(&loopback_dev);
+  struct inet6_dev *loopback_idev = in6_dev_get(&per_net(loopback_dev, init_net()));
```

```
   BUG_ON(!loopback_idev);

   do {
diff --git a/net/xfrm/xfrm_policy.c b/net/xfrm/xfrm_policy.c
index 0248343..51ab8ac 100644
--- a/net/xfrm/xfrm_policy.c
+++ b/net/xfrm/xfrm_policy.c
@@ -1799,8 +1799,8 @@ static int stale_bundle(struct dst_entry *dst)
 void xfrm_dst_ifdown(struct dst_entry *dst, struct net_device *dev)
 {
  while ((dst = dst->child) && dst->xfrm && dst->dev == dev) {
-  dst->dev = &loopback_dev;
-  dev_hold(&loopback_dev);
+  dst->dev = &per_net(loopback_dev, init_net());
+  dev_hold(dst->dev);
   dev_put(dev);
  }
 }
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 16/31] net: Make the device list and device lookups per namespace.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:18 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch makes most of the generic device layer network
namespace safe. This patch makes dev_base, dev_base_lock
per network namespace variables, and then it picks up
a few associated variables. The funnctions:
dev_getbyhwaddr
dev_getfirsthwbytype
dev_get_by_flags
dev_get_by_name
__dev_get_by_name
dev_get_by_index
__dev_get_by_index
dev_ioctl
dev_ethtool
dev_load

wireless_process_ioctl

were modified to take a network namespace argument, and
deal with it.

vlan_ioctl_set and brioctl_set were modified so their
hooks will receive a network namespace argument.

So basically anthing in the core of the network stack that was
affected to by the change of dev_base and dev_base_lock was
modified to handle multiple network namespaces. The rest of the
network stack was simply modified to explicitly use init_net()
the initial network namespace. This can be fixed when those
components of the network stack are modified to handle multiple
network namespaces.

For now the ifindex generator is left global.

Fundametally ifindex numbers are per namespace, or else
we will have corner case problems with migration when
we get that far.

At the same time there are assumptions in the network stack
that the ifindex of a network device won't change. Making
the ifindex number global seems a good compromise until
the network stack can cope with ifindex changes when
you change namespaces, and the like.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/s390/appldata/appldata_net_sum.c | 6 +-
 arch/sparc64/solaris/ioctl.c | 6 +-
 drivers/atm/idt77252.c | 2 +-
 drivers/block/aoe/aoecmd.c | 7 +-
 drivers/net/bonding/bond_main.c | 6 +-
 drivers/net/bonding/bond_sysfs.c | 3 +-
 drivers/net/eql.c | 9 +-
 drivers/net/pppoe.c | 2 +-
 drivers/net/shaper.c | 3 +-
 drivers/net/tun.c | 3 +-
 drivers/net/wan/dlci.c | 4 +-
 drivers/net/wan/sbni.c | 2 +-
 drivers/net/wireless/strip.c | 8 +-
 drivers/parisc/led.c | 6 +-
 include/linux/if_bridge.h | 2 +-
 include/linux/if_vlan.h | 2 +-
 include/linux/netdevice.h | 24 ++++--
 include/net/iw_handler.h | 2 +-

```
net/802/tr.c                    |    2 +-
net/8021q/vlan.c                |   10 +-
net/8021q/vlan_dev.c            |   12 +-
net/8021q/vlanproc.c            |    8 +-
net/appletalk/ddp.c             |    6 +-
net/atm/mpc.c                   |    2 +-
net/ax25/af_ax25.c              |    2 +-
net/bridge/br_if.c              |    6 +-
net/bridge/br_ioctl.c           |    7 +-
net/bridge/br_netlink.c         |    9 +-
net/bridge/br_private.h         |    2 +-
net/core/dev.c                  |  282 ++++++++++++++++++++++++++-----------
net/core/dev_mcast.c            |   46 +++++-
net/core/ethtool.c              |    4 +-
net/core/fib_rules.c            |    4 +-
net/core/link_watch.c           |    5 +-
net/core/neighbour.c            |    6 +-
net/core/net-sysfs.c            |   27 ++--
net/core/netpoll.c              |    2 +-
net/core/pktgen.c               |    2 +-
net/core/rtnetlink.c            |   24 ++--
net/core/sock.c                 |    3 +-
net/core/wireless.c             |   43 +++++-
net/decnet/af_decnet.c          |    6 +-
net/decnet/dn_dev.c             |   32 ++--
net/decnet/dn_fib.c             |   12 +-
net/decnet/dn_route.c           |   14 +-
net/decnet/sysctl_net_decnet.c  |    4 +-
net/econet/af_econet.c          |    2 +-
net/ipv4/arp.c                  |    4 +-
net/ipv4/devinet.c              |   36 ++--
net/ipv4/fib_frontend.c         |    2 +-
net/ipv4/fib_semantics.c        |    4 +-
net/ipv4/igmp.c                 |   12 +-
net/ipv4/ip_fragment.c          |    2 +-
net/ipv4/ip_gre.c               |    4 +-
net/ipv4/ip_sockglue.c          |    2 +-
net/ipv4/ipconfig.c             |    2 +-
net/ipv4/ipip.c                 |    4 +-
net/ipv4/ipmr.c                 |    4 +-
net/ipv4/ipvs/ip_vs_sync.c      |   10 +-
net/ipv4/netfilter/ipt_CLUSTERIP.c |    2 +-
net/ipv4/route.c                |    4 +-
net/ipv6/addrconf.c             |   44 +++---
net/ipv6/af_inet6.c             |    2 +-
net/ipv6/anycast.c              |   20 ++--
net/ipv6/datagram.c             |    2 +-
net/ipv6/ip6_tunnel.c           |    6 +-
```

```
 net/ipv6/ipv6_sockglue.c            |    2 +-
 net/ipv6/mcast.c                    |   20 ++--
 net/ipv6/raw.c                      |    2 +-
 net/ipv6/reassembly.c               |    2 +-
 net/ipv6/route.c                    |    4 +-
 net/ipv6/sit.c                      |    4 +-
 net/ipx/af_ipx.c                    |    6 +-
 net/llc/af_llc.c                    |    4 +-
 net/llc/llc_core.c                  |    5 +-
 net/netrom/nr_route.c               |   14 +-
 net/packet/af_packet.c              |   18 +-
 net/rose/rose_route.c               |   20 ++--
 net/sched/act_mirred.c              |    2 +-
 net/sched/cls_api.c                 |    4 +-
 net/sched/em_meta.c                 |    2 +-
 net/sched/sch_api.c                 |   14 +-
 net/sctp/ipv6.c                     |    4 +-
 net/sctp/protocol.c                 |    6 +-
 net/socket.c                        |   22 ++-
 net/tipc/eth_media.c                |    2 +-
 net/wanrouter/af_wanpipe.c          |   24 ++--
 net/x25/x25_route.c                 |    2 +-
 88 files changed, 597 insertions(+), 433 deletions(-)

diff --git a/arch/s390/appldata/appldata_net_sum.c b/arch/s390/appldata/appldata_net_sum.c
index 075e619..4a32370 100644
--- a/arch/s390/appldata/appldata_net_sum.c
+++ b/arch/s390/appldata/appldata_net_sum.c
@@ -106,8 +106,8 @@ static void appldata_get_net_sum_data(void *data)
  rx_dropped = 0;
  tx_dropped = 0;
  collisions = 0;
- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
   if (dev->get_stats == NULL) {
    continue;
   }
@@ -123,7 +123,7 @@ static void appldata_get_net_sum_data(void *data)
   collisions += stats->collisions;
   i++;
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  net_data->nr_interfaces = i;
  net_data->rx_packets = rx_packets;
  net_data->tx_packets = tx_packets;
```

```
diff --git a/arch/sparc64/solaris/ioctl.c b/arch/sparc64/solaris/ioctl.c
index 330743c..1ecf4ab 100644
--- a/arch/sparc64/solaris/ioctl.c
+++ b/arch/sparc64/solaris/ioctl.c
@@ -685,9 +685,9 @@ static inline int solaris_i(unsigned int fd, unsigned int cmd, u32 arg)
    struct net_device *d;
    int i = 0;

-   read_lock_bh(&dev_base_lock);
-   for (d = dev_base; d; d = d->next) i++;
-   read_unlock_bh(&dev_base_lock);
+   read_lock_bh(&per_net(dev_base_lock, init_net()));
+   for (d = per_net(dev_base, init_net()); d; d = d->next) i++;
+   read_unlock_bh(&per_net(dev_base_lock, init_net()));

    if (put_user (i, (int __user *)A(arg)))
     return -EFAULT;
diff --git a/drivers/atm/idt77252.c b/drivers/atm/idt77252.c
index f407861..3e75e0e 100644
--- a/drivers/atm/idt77252.c
+++ b/drivers/atm/idt77252.c
@@ -3569,7 +3569,7 @@ init_card(struct atm_dev *dev)
   * XXX: <hack>
   */
  sprintf(tname, "eth%d", card->index);
- tmp = dev_get_by_name(tname); /* jhs: was "tmp = dev_get(tname);" */
+ tmp = dev_get_by_name(init_net(), tname); /* jhs: was "tmp = dev_get(tname);" */
  if (tmp) {
   memcpy(card->atmdev->esi, tmp->dev_addr, 6);

diff --git a/drivers/block/aoe/aoecmd.c b/drivers/block/aoe/aoecmd.c
index bb022ed..9678169 100644
--- a/drivers/block/aoe/aoecmd.c
+++ b/drivers/block/aoe/aoecmd.c
@@ -9,6 +9,7 @@
 #include <linux/skbuff.h>
 #include <linux/netdevice.h>
 #include <linux/genhd.h>
+#include <net/net_namespace.h>
 #include <asm/unaligned.h>
 #include "aoe.h"

@@ -192,8 +193,8 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigned char aoeminor, struct
sk_buff **tail)

  sl = sl_tail = NULL;

- read_lock(&dev_base_lock);
```

```
- for (ifp = dev_base; ifp; dev_put(ifp), ifp = ifp->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (ifp = per_net(dev_base, init_net()); ifp; dev_put(ifp), ifp = ifp->next) {
   dev_hold(ifp);
   if (!is_aoe_netif(ifp))
    continue;
@@ -221,7 +222,7 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigned char aoeminor, struct
sk_buff **tail)
   skb->next = sl;
   sl = skb;
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  if (tail != NULL)
   *tail = sl_tail;
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 3e04f58..2963004 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -2932,7 +2932,7 @@ static void *bond_info_seq_start(struct seq_file *seq, loff_t *pos)
  int i;

  /* make sure the bond won't be taken away */
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  read_lock_bh(&bond->lock);

  if (*pos == 0) {
@@ -2968,7 +2968,7 @@ static void bond_info_seq_stop(struct seq_file *seq, void *v)
  struct bonding *bond = seq->private;

  read_unlock_bh(&bond->lock);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static void bond_info_show_master(struct seq_file *seq)
@@ -3742,7 +3742,7 @@ static int bond_do_ioctl(struct net_device *bond_dev, struct ifreq *ifr,
int cmd
  }

  down_write(&(bonding_rwsem));
- slave_dev = dev_get_by_name(ifr->ifr_slave);
+ slave_dev = dev_get_by_name(init_net(), ifr->ifr_slave);

  dprintk("slave_dev=%p: \n", slave_dev);
```

```
diff --git a/drivers/net/bonding/bond_sysfs.c b/drivers/net/bonding/bond_sysfs.c
index ced9ed8..561707c 100644
--- a/drivers/net/bonding/bond_sysfs.c
+++ b/drivers/net/bonding/bond_sysfs.c
@@ -36,6 +36,7 @@
 #include <linux/ctype.h>
 #include <linux/inet.h>
 #include <linux/rtnetlink.h>
+#include <net/net_namespace.h>

 /* #define BONDING_DEBUG 1 */
 #include "bonding.h"
@@ -298,7 +299,7 @@ static ssize_t bonding_store_slaves(struct class_device *cd, const char
*buffer,
   read_unlock_bh(&bond->lock);
   printk(KERN_INFO DRV_NAME ": %s: Adding slave %s.\n",
          bond->dev->name, ifname);
-  dev = dev_get_by_name(ifname);
+  dev = dev_get_by_name(init_net(), ifname);
   if (!dev) {
    printk(KERN_INFO DRV_NAME
           ": %s: Interface %s does not exist!\n",
diff --git a/drivers/net/eql.c b/drivers/net/eql.c
index a93700e..ceae8a0 100644
--- a/drivers/net/eql.c
+++ b/drivers/net/eql.c
@@ -116,6 +116,7 @@
 #include <linux/init.h>
 #include <linux/timer.h>
 #include <linux/netdevice.h>
+#include <net/net_namespace.h>

 #include <linux/if.h>
 #include <linux/if_arp.h>
@@ -412,7 +413,7 @@ static int eql_enslave(struct net_device *master_dev, slaving_request_t
__user *
  if (copy_from_user(&srq, srqp, sizeof (slaving_request_t)))
   return -EFAULT;

- slave_dev  = dev_get_by_name(srq.slave_name);
+ slave_dev  = dev_get_by_name(init_net(), srq.slave_name);
  if (slave_dev) {
   if ((master_dev->flags & IFF_UP) == IFF_UP) {
    /* slave is not a master & not already a slave: */
@@ -460,7 +461,7 @@ static int eql_emancipate(struct net_device *master_dev,
slaving_request_t __use
  if (copy_from_user(&srq, srqp, sizeof (slaving_request_t)))
   return -EFAULT;
```

```
- slave_dev = dev_get_by_name(srq.slave_name);
+ slave_dev = dev_get_by_name(init_net(), srq.slave_name);
  ret = -EINVAL;
  if (slave_dev) {
   spin_lock_bh(&eql->queue.lock);
@@ -493,7 +494,7 @@ static int eql_g_slave_cfg(struct net_device *dev, slave_config_t __user
*scp)
  if (copy_from_user(&sc, scp, sizeof (slave_config_t)))
   return -EFAULT;

- slave_dev = dev_get_by_name(sc.slave_name);
+ slave_dev = dev_get_by_name(init_net(), sc.slave_name);
  if (!slave_dev)
   return -ENODEV;

@@ -528,7 +529,7 @@ static int eql_s_slave_cfg(struct net_device *dev, slave_config_t __user
*scp)
  if (copy_from_user(&sc, scp, sizeof (slave_config_t)))
   return -EFAULT;

- slave_dev = dev_get_by_name(sc.slave_name);
+ slave_dev = dev_get_by_name(init_net(), sc.slave_name);
  if (!slave_dev)
   return -ENODEV;

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index 3618862..3c8b0a7 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -586,7 +586,7 @@ static int pppoe_connect(struct socket *sock, struct sockaddr *uservaddr,

  /* Don't re-bind if sid==0 */
  if (sp->sa_addr.pppoe.sid != 0) {
-  dev = dev_get_by_name(sp->sa_addr.pppoe.dev);
+  dev = dev_get_by_name(init_net(), sp->sa_addr.pppoe.dev);

   error = -ENODEV;
   if (!dev)
diff --git a/drivers/net/shaper.c b/drivers/net/shaper.c
index e886e8d..b852055 100644
--- a/drivers/net/shaper.c
+++ b/drivers/net/shaper.c
@@ -86,6 +86,7 @@

 #include <net/dst.h>
 #include <net/arp.h>
+#include <net/net_namespace.h>
```

```
 struct shaper_cb {
  unsigned long shapeclock;  /* Time it should go out */
@@ -488,7 +489,7 @@ static int shaper_ioctl(struct net_device *dev,  struct ifreq *ifr, int cmd)
  {
   case SHAPER_SET_DEV:
   {
-   struct net_device *them=__dev_get_by_name(ss->ss_name);
+   struct net_device *them=__dev_get_by_name(init_net(), ss->ss_name);
    if(them==NULL)
     return -ENODEV;
    if(sh->dev)
diff --git a/drivers/net/tun.c b/drivers/net/tun.c
index 151a2e1..efa1db8 100644
--- a/drivers/net/tun.c
+++ b/drivers/net/tun.c
@@ -58,6 +58,7 @@
 #include <linux/if_ether.h>
 #include <linux/if_tun.h>
 #include <linux/crc32.h>
+#include <net/net_namespace.h>

 #include <asm/system.h>
 #include <asm/uaccess.h>
@@ -464,7 +465,7 @@ static int tun_set_iff(struct file *file, struct ifreq *ifr)
       current->euid != tun->owner && !capable(CAP_NET_ADMIN))
    return -EPERM;
  }
- else if (__dev_get_by_name(ifr->ifr_name))
+ else if (__dev_get_by_name(init_net(), ifr->ifr_name))
   return -EINVAL;
  else {
   char *name;
diff --git a/drivers/net/wan/dlci.c b/drivers/net/wan/dlci.c
index f826494..1e3b73b 100644
--- a/drivers/net/wan/dlci.c
+++ b/drivers/net/wan/dlci.c
@@ -361,7 +361,7 @@ static int dlci_add(struct dlci_add *dlci)


  /* validate slave device */
- slave = dev_get_by_name(dlci->devname);
+ slave = dev_get_by_name(init_net(), dlci->devname);
  if (!slave)
   return -ENODEV;

@@ -427,7 +427,7 @@ static int dlci_del(struct dlci_add *dlci)
  int   err;
```

```
  /* validate slave device */
- master = __dev_get_by_name(dlci->devname);
+ master = __dev_get_by_name(init_net(), dlci->devname);
  if (!master)
   return(-ENODEV);

diff --git a/drivers/net/wan/sbni.c b/drivers/net/wan/sbni.c
index fc5c0c6..ae8f941 100644
--- a/drivers/net/wan/sbni.c
+++ b/drivers/net/wan/sbni.c
@@ -1367,7 +1367,7 @@ sbni_ioctl( struct net_device  *dev,  struct ifreq  *ifr,  int  cmd )

   if (copy_from_user( slave_name, ifr->ifr_data, sizeof slave_name ))
    return -EFAULT;
-  slave_dev = dev_get_by_name( slave_name );
+  slave_dev = dev_get_by_name(init_net(), slave_name );
   if( !slave_dev  ||  !(slave_dev->flags & IFF_UP) ) {
    printk( KERN_ERR "%s: trying to enslave non-active "
     "device %s\n", dev->name, slave_name );
diff --git a/drivers/net/wireless/strip.c b/drivers/net/wireless/strip.c
index 6c27ff2..6c2ada3 100644
--- a/drivers/net/wireless/strip.c
+++ b/drivers/net/wireless/strip.c
@@ -1971,8 +1971,8 @@ static struct net_device *get_strip_dev(struct strip *strip_info)
      && memcmp(&strip_info->true_dev_addr, zero_address.c,
        sizeof(zero_address))) {
   struct net_device *dev;
-  read_lock_bh(&dev_base_lock);
-  dev = dev_base;
+  read_lock_bh(&per_net(dev_base_lock, init_net()));
+  dev = per_net(dev_base, init_net());
   while (dev) {
    if (dev->type == strip_info->dev->type &&
       !memcmp(dev->dev_addr,
@@ -1981,12 +1981,12 @@ static struct net_device *get_strip_dev(struct strip *strip_info)
     printk(KERN_INFO
           "%s: Transferred packet ownership to %s.\n",
           strip_info->dev->name, dev->name);
-    read_unlock_bh(&dev_base_lock);
+    read_unlock_bh(&per_net(dev_base_lock, init_net()));
     return (dev);
    }
    dev = dev->next;
   }
-  read_unlock_bh(&dev_base_lock);
+  read_unlock_bh(&per_net(dev_base_lock, init_net()));
  }
```

```
  return (strip_info->dev);
 }
diff --git a/drivers/parisc/led.c b/drivers/parisc/led.c
index 8dac2ba..62662f3 100644
--- a/drivers/parisc/led.c
+++ b/drivers/parisc/led.c
@@ -365,9 +365,9 @@ static __inline__ int led_get_net_activity(void)

  /* we are running as a workqueue task, so locking dev_base
   * for reading should be OK */
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
      struct net_device_stats *stats;
      struct in_device *in_dev = __in_dev_get_rcu(dev);
      if (!in_dev || !in_dev->ifa_list)
@@ -381,7 +381,7 @@ static __inline__ int led_get_net_activity(void)
      tx_total += stats->tx_packets;
  }
  rcu_read_unlock();
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  retval = 0;

diff --git a/include/linux/if_bridge.h b/include/linux/if_bridge.h
index fd1b6eb..3b74be1 100644
--- a/include/linux/if_bridge.h
+++ b/include/linux/if_bridge.h
@@ -104,7 +104,7 @@ struct __fdb_entry

 #include <linux/netdevice.h>

-extern void brioctl_set(int (*ioctl_hook)(unsigned int, void __user *));
+extern void brioctl_set(int (*ioctl_hook)(net_t, unsigned int, void __user *));
 extern int (*br_handle_frame_hook)(struct net_bridge_port *p, struct sk_buff **pskb);
 extern int (*br_should_route_hook)(struct sk_buff **pskb);

diff --git a/include/linux/if_vlan.h b/include/linux/if_vlan.h
index 35cb385..8ee195a 100644
--- a/include/linux/if_vlan.h
+++ b/include/linux/if_vlan.h
@@ -62,7 +62,7 @@ struct vlan_hdr {
 #define VLAN_VID_MASK 0xfff

 /* found in socket.c */
```

```
-extern void vlan_ioctl_set(int (*hook)(void __user *));
+extern void vlan_ioctl_set(int (*hook)(net_t, void __user *));

 #define VLAN_NAME "vlan"

diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 73931a0..0b4a4dc 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -571,21 +571,21 @@ struct packet_type {
 #include <linux/notifier.h>

 DECLARE_PER_NET(struct net_device, loopback_dev);  /* The loopback */
-extern struct net_device  *dev_base;  /* All devices */
-extern rwlock_t    dev_base_lock;  /* Device list lock */
+DECLARE_PER_NET(struct net_device *, dev_base);  /* All devices */
+DECLARE_PER_NET(rwlock_t,  dev_base_lock);  /* Device list lock */

 extern int    netdev_boot_setup_check(struct net_device *dev);
 extern unsigned long  netdev_boot_base(const char *prefix, int unit);
-extern struct net_device    *dev_getbyhwaddr(unsigned short type, char *hwaddr);
-extern struct net_device *dev_getfirstbyhwtype(unsigned short type);
+extern struct net_device    *dev_getbyhwaddr(net_t net, unsigned short type, char *hwaddr);
+extern struct net_device *dev_getfirstbyhwtype(net_t net, unsigned short type);
 extern void  dev_add_pack(struct packet_type *pt);
 extern void  dev_remove_pack(struct packet_type *pt);
 extern void  __dev_remove_pack(struct packet_type *pt);

-extern struct net_device *dev_get_by_flags(unsigned short flags,
+extern struct net_device *dev_get_by_flags(net_t net, unsigned short flags,
        unsigned short mask);
-extern struct net_device *dev_get_by_name(const char *name);
-extern struct net_device *__dev_get_by_name(const char *name);
+extern struct net_device *dev_get_by_name(net_t net, const char *name);
+extern struct net_device *__dev_get_by_name(net_t net, const char *name);
 extern int  dev_alloc_name(struct net_device *dev, const char *name);
 extern int  dev_open(struct net_device *dev);
 extern int  dev_close(struct net_device *dev);
@@ -597,8 +597,8 @@ extern void  synchronize_net(void);
 extern int   register_netdevice_notifier(struct notifier_block *nb);
 extern int  unregister_netdevice_notifier(struct notifier_block *nb);
 extern int call_netdevice_notifiers(unsigned long val, struct net_device *dev);
-extern struct net_device *dev_get_by_index(int ifindex);
-extern struct net_device *__dev_get_by_index(int ifindex);
+extern struct net_device *dev_get_by_index(net_t net, int ifindex);
+extern struct net_device *__dev_get_by_index(net_t net, int ifindex);
 extern int  dev_restart(struct net_device *dev);
 #ifdef CONFIG_NETPOLL_TRAP
```

```
 extern int  netpoll_trap(void);
@@ -705,8 +705,8 @@ extern int  netif_rx_ni(struct sk_buff *skb);
 #define HAVE_NETIF_RECEIVE_SKB 1
 extern int  netif_receive_skb(struct sk_buff *skb);
 extern int  dev_valid_name(const char *name);
-extern int  dev_ioctl(unsigned int cmd, void __user *);
-extern int  dev_ethtool(struct ifreq *);
+extern int  dev_ioctl(net_t net, unsigned int cmd, void __user *);
+extern int  dev_ethtool(net_t net, struct ifreq *);
 extern unsigned  dev_get_flags(const struct net_device *);
 extern int  dev_change_flags(struct net_device *, unsigned);
 extern int  dev_change_name(struct net_device *, char *);
@@ -982,7 +982,7 @@ extern void  dev_set_allmulti(struct net_device *dev, int inc);
 extern void  netdev_state_change(struct net_device *dev);
 extern void  netdev_features_change(struct net_device *dev);
 /* Load a device via the kmod */
-extern void  dev_load(const char *name);
+extern void  dev_load(net_t net, const char *name);
 extern void  dev_mcast_init(void);
 extern int  netdev_max_backlog;
 extern int  weight_p;
diff --git a/include/net/iw_handler.h b/include/net/iw_handler.h
index 10559e9..f274eca 100644
--- a/include/net/iw_handler.h
+++ b/include/net/iw_handler.h
@@ -434,7 +434,7 @@ extern int dev_get_wireless_info(char * buffer, char **start, off_t offset,
      int length);

 /* Handle IOCTLs, called in net/core/dev.c */
-extern int wireless_process_ioctl(struct ifreq *ifr, unsigned int cmd);
+extern int wireless_process_ioctl(net_t net, struct ifreq *ifr, unsigned int cmd);

 /* Handle RtNetlink requests, called in net/core/rtnetlink.c */
 extern int wireless_rtnetlink_set(struct net_device * dev,
diff --git a/net/802/tr.c b/net/802/tr.c
index 3324fa6..7a8cfbe 100644
--- a/net/802/tr.c
+++ b/net/802/tr.c
@@ -531,7 +531,7 @@ static int rif_seq_show(struct seq_file *seq, void *v)
   seq_puts(seq,
        "if     TR address       TTL   rcf   routing segments\n");
  else {
-  struct net_device *dev = dev_get_by_index(entry->iface);
+  struct net_device *dev = dev_get_by_index(init_net(), entry->iface);
   long ttl = (long) (entry->last_used + sysctl_tr_rif_timeout)
     - (long) jiffies;

diff --git a/net/8021q/vlan.c b/net/8021q/vlan.c
```

```
index f80cfdd..e03d7de 100644
--- a/net/8021q/vlan.c
+++ b/net/8021q/vlan.c
@@ -51,7 +51,7 @@ static char vlan_copyright[] = "Ben Greear <greearb@candelatech.com>";
 static char vlan_buggyright[] = "David S. Miller <davem@redhat.com>";

 static int vlan_device_event(struct notifier_block *, unsigned long, void *);
-static int vlan_ioctl_handler(void __user *);
+static int vlan_ioctl_handler(net_t net, void __user *);
 static int unregister_vlan_dev(struct net_device *, unsigned short );

 static struct notifier_block vlan_notifier_block = {
@@ -118,7 +118,7 @@ static void __exit vlan_cleanup_devices(void)
  struct net_device *dev, *nxt;

  rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = nxt) {
   nxt = dev->next;
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
    unregister_vlan_dev(VLAN_DEV_INFO(dev)->real_dev,
@@ -279,7 +279,7 @@ static int unregister_vlan_device(const char *vlan_IF_name)
  int ret;


- dev = dev_get_by_name(vlan_IF_name);
+ dev = dev_get_by_name(init_net(), vlan_IF_name);
  ret = -EINVAL;
  if (dev) {
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -390,7 +390,7 @@ static struct net_device *register_vlan_device(const char *eth_IF_name,
   goto out_ret_null;

  /* find the device relating to eth_IF_name. */
- real_dev = dev_get_by_name(eth_IF_name);
+ real_dev = dev_get_by_name(init_net(), eth_IF_name);
  if (!real_dev)
   goto out_ret_null;

@@ -678,7 +678,7 @@ out:
  * o execute requested action or pass command to the device driver
  *   arg is really a struct vlan_ioctl_args __user *.
  */
-static int vlan_ioctl_handler(void __user *arg)
+static int vlan_ioctl_handler(net_t net, void __user *arg)
 {
  int err = 0;
  unsigned short vid = 0;
```

```
diff --git a/net/8021q/vlan_dev.c b/net/8021q/vlan_dev.c
index 9fce3a8..fa2186d 100644
--- a/net/8021q/vlan_dev.c
+++ b/net/8021q/vlan_dev.c
@@ -539,7 +539,7 @@ int vlan_dev_change_mtu(struct net_device *dev, int new_mtu)

 int vlan_dev_set_ingress_priority(char *dev_name, __u32 skb_prio, short vlan_prio)
 {
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);

  if (dev) {
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -556,7 +556,7 @@ int vlan_dev_set_ingress_priority(char *dev_name, __u32 skb_prio, short
vlan_pri

 int vlan_dev_set_egress_priority(char *dev_name, __u32 skb_prio, short vlan_prio)
 {
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);
  struct vlan_priority_tci_mapping *mp = NULL;
  struct vlan_priority_tci_mapping *np;

@@ -596,7 +596,7 @@ int vlan_dev_set_egress_priority(char *dev_name, __u32 skb_prio, short
vlan_prio
 /* Flags are defined in the vlan_dev_info class in include/linux/if_vlan.h file. */
 int vlan_dev_set_vlan_flag(char *dev_name, __u32 flag, short flag_val)
 {
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);

  if (dev) {
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -632,7 +632,7 @@ int vlan_dev_set_vlan_flag(char *dev_name, __u32 flag, short flag_val)

 int vlan_dev_get_realdev_name(const char *dev_name, char* result)
 {
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);
  int rv = 0;
  if (dev) {
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -650,7 +650,7 @@ int vlan_dev_get_realdev_name(const char *dev_name, char* result)

 int vlan_dev_get_vid(const char *dev_name, unsigned short* result)
 {
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);
```

```
  int rv = 0;
  if (dev) {
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -821,7 +821,7 @@ int vlan_dev_ioctl(struct net_device *dev, struct ifreq *ifr, int cmd)
    break;

  case SIOCETHTOOL:
-  err = dev_ethtool(&ifrr);
+  err = dev_ethtool(real_dev->nd_net, &ifrr);
  }

  if (!err)
diff --git a/net/8021q/vlanproc.c b/net/8021q/vlanproc.c
index abcf58c..0e93991 100644
--- a/net/8021q/vlanproc.c
+++ b/net/8021q/vlanproc.c
@@ -253,12 +253,12 @@ static void *vlan_seq_start(struct seq_file *seq, loff_t *pos)
  struct net_device *dev;
  loff_t i = 1;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));

  if (*pos == 0)
   return SEQ_START_TOKEN;

- for (dev = vlan_skip(dev_base); dev && i < *pos;
+ for (dev = vlan_skip(per_net(dev_base, init_net())); dev && i < *pos;
       dev = vlan_skip(dev->next), ++i);

  return  (i == *pos) ? dev : NULL;
@@ -269,13 +269,13 @@ static void *vlan_seq_next(struct seq_file *seq, void *v, loff_t *pos)
  ++*pos;

  return vlan_skip((v == SEQ_START_TOKEN)
-       ? dev_base
+       ? per_net(dev_base, init_net())
        : ((struct net_device *)v)->next);
 }

 static void vlan_seq_stop(struct seq_file *seq, void *v)
 {
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static int vlan_seq_show(struct seq_file *seq, void *v)
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
```

```
index 61f36b1..4cdcae3 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -677,7 +677,7 @@ static int atif_ioctl(int cmd, void __user *arg)
  if (copy_from_user(&atreq, arg, sizeof(atreq)))
   return -EFAULT;

- dev = __dev_get_by_name(atreq.ifr_name);
+ dev = __dev_get_by_name(init_net(), atreq.ifr_name);
  if (!dev)
   return -ENODEV;

@@ -901,7 +901,7 @@ static int atrtr_ioctl(unsigned int cmd, void __user *arg)
     if (copy_from_user(name, rt.rt_dev, IFNAMSIZ-1))
      return -EFAULT;
     name[IFNAMSIZ-1] = '\0';
-    dev = __dev_get_by_name(name);
+    dev = __dev_get_by_name(init_net(), name);
     if (!dev)
      return -ENODEV;
    }
@@ -1273,7 +1273,7 @@ static __inline__ int is_ip_over_ddp(struct sk_buff *skb)

 static int handle_ip_over_ddp(struct sk_buff *skb)
 {
-        struct net_device *dev = __dev_get_by_name("ipddp0");
+        struct net_device *dev = __dev_get_by_name(init_net(), "ipddp0");
  struct net_device_stats *stats;

  /* This needs to be able to handle ipddp"N" devices */
diff --git a/net/atm/mpc.c b/net/atm/mpc.c
index 4fdb1af..e17c10b 100644
--- a/net/atm/mpc.c
+++ b/net/atm/mpc.c
@@ -244,7 +244,7 @@ static struct net_device *find_lec_by_itfnum(int itf)
  char name[IFNAMSIZ];

  sprintf(name, "lec%d", itf);
- dev = dev_get_by_name(name);
+ dev = dev_get_by_name(init_net(), name);

  return dev;
 }
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index 8c187a6..e2f6fed 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -632,7 +632,7 @@ static int ax25_setsockopt(struct socket *sock, int level, int optname,
```

```
    break;
   }

-  dev = dev_get_by_name(devname);
+  dev = dev_get_by_name(init_net(), devname);
   if (dev == NULL) {
    res = -ENODEV;
    break;
diff --git a/net/bridge/br_if.c b/net/bridge/br_if.c
index 55bb263..22509f1 100644
--- a/net/bridge/br_if.c
+++ b/net/bridge/br_if.c
@@ -45,7 +45,7 @@ static int port_cost(struct net_device *dev)

  old_fs = get_fs();
  set_fs(KERNEL_DS);
- err = dev_ethtool(&ifr);
+ err = dev_ethtool(dev->nd_net, &ifr);
  set_fs(old_fs);

  if (!err) {
@@ -328,7 +328,7 @@ int br_del_bridge(const char *name)
  int ret = 0;

  rtnl_lock();
- dev = __dev_get_by_name(name);
+ dev = __dev_get_by_name(init_net(), name);
  if (dev == NULL)
   ret =  -ENXIO;  /* Could not find device */

@@ -483,7 +483,7 @@ void __exit br_cleanup_bridges(void)
  struct net_device *dev, *nxt;

  rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = nxt) {
   nxt = dev->next;
   if (dev->priv_flags & IFF_EBRIDGE)
    del_br(dev->priv);
diff --git a/net/bridge/br_ioctl.c b/net/bridge/br_ioctl.c
index 4c61a7e..2be1c2d 100644
--- a/net/bridge/br_ioctl.c
+++ b/net/bridge/br_ioctl.c
@@ -18,6 +18,7 @@
 #include <linux/if_bridge.h>
 #include <linux/netdevice.h>
 #include <linux/times.h>
+#include <net/net_namespace.h>
```

```
 #include <asm/uaccess.h>
 #include "br_private.h"

@@ -27,7 +28,7 @@ static int get_bridge_ifindices(int *indices, int num)
  struct net_device *dev;
  int i = 0;

- for (dev = dev_base; dev && i < num; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev && i < num; dev = dev->next) {
   if (dev->priv_flags & IFF_EBRIDGE)
    indices[i++] = dev->ifindex;
  }
@@ -88,7 +89,7 @@ static int add_del_if(struct net_bridge *br, int ifindex, int isadd)
  if (!capable(CAP_NET_ADMIN))
   return -EPERM;

- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
  if (dev == NULL)
   return -EINVAL;

@@ -362,7 +363,7 @@ static int old_deviceless(void __user *uarg)
  return -EOPNOTSUPP;
 }

-int br_ioctl_deviceless_stub(unsigned int cmd, void __user *uarg)
+int br_ioctl_deviceless_stub(net_t net, unsigned int cmd, void __user *uarg)
 {
  switch (cmd) {
  case SIOCGIFBR:
diff --git a/net/bridge/br_netlink.c b/net/bridge/br_netlink.c
index a913968..119b97d 100644
--- a/net/bridge/br_netlink.c
+++ b/net/bridge/br_netlink.c
@@ -13,6 +13,7 @@
 #include <linux/kernel.h>
 #include <linux/rtnetlink.h>
 #include <net/netlink.h>
+#include <net/net_namespace.h>
 #include "br_private.h"

 static inline size_t br_nlmsg_size(void)
@@ -106,8 +107,8 @@ static int br_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
  struct net_device *dev;
  int idx;

- read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next) {
```

```
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next) {
   /* not a bridge port */
   if (dev->br_port == NULL || idx < cb->args[0])
    goto skip;
@@ -119,7 +120,7 @@ static int br_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
 skip:
   ++idx;
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  cb->args[0] = idx;

@@ -153,7 +154,7 @@ static int br_rtm_setlink(struct sk_buff *skb,  struct nlmsghdr *nlh, void
*arg)
  if (new_state > BR_STATE_BLOCKING)
   return -EINVAL;

- dev = __dev_get_by_index(ifm->ifi_index);
+ dev = __dev_get_by_index(init_net(), ifm->ifi_index);
  if (!dev)
   return -ENODEV;

diff --git a/net/bridge/br_private.h b/net/bridge/br_private.h
index f1712b9..1d60ee3 100644
--- a/net/bridge/br_private.h
+++ b/net/bridge/br_private.h
@@ -189,7 +189,7 @@ extern int br_handle_frame(struct net_bridge_port *p, struct sk_buff
**pskb);

 /* br_ioctl.c */
 extern int br_dev_ioctl(struct net_device *dev, struct ifreq *rq, int cmd);
-extern int br_ioctl_deviceless_stub(unsigned int cmd, void __user *arg);
+extern int br_ioctl_deviceless_stub(net_t net, unsigned int cmd, void __user *arg);

 /* br_netfilter.c */
 #ifdef CONFIG_BRIDGE_NETFILTER
diff --git a/net/core/dev.c b/net/core/dev.c
index d8aa534..32fe905 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -175,26 +175,27 @@ static spinlock_t net_dma_event_lock;
  * unregister_netdevice(), which must be called with the rtnl
  * semaphore held.
  */
-struct net_device *dev_base;
-static struct net_device **dev_tail = &dev_base;
```

```
-DEFINE_RWLOCK(dev_base_lock);
+DEFINE_PER_NET(struct net_device *, dev_base);
+static DEFINE_PER_NET(struct net_device **, dev_tail);
+DEFINE_PER_NET(rwlock_t, dev_base_lock);

-EXPORT_SYMBOL(dev_base);
-EXPORT_SYMBOL(dev_base_lock);
+EXPORT_PER_NET_SYMBOL(dev_base);
+EXPORT_PER_NET_SYMBOL(dev_base_lock);

 #define NETDEV_HASHBITS 8
-static struct hlist_head dev_name_head[1<<NETDEV_HASHBITS];
-static struct hlist_head dev_index_head[1<<NETDEV_HASHBITS];
+typedef struct hlist_head dev_hlists_t[1<<NETDEV_HASHBITS];
+static DEFINE_PER_NET(dev_hlists_t, dev_name_head);
+static DEFINE_PER_NET(dev_hlists_t, dev_index_head);

-static inline struct hlist_head *dev_name_hash(const char *name)
+static inline struct hlist_head *dev_name_hash(net_t net, const char *name)
 {
  unsigned hash = full_name_hash(name, strnlen(name, IFNAMSIZ));
- return &dev_name_head[hash & ((1<<NETDEV_HASHBITS)-1)];
+ return &per_net(dev_name_head, net)[hash & ((1<<NETDEV_HASHBITS)-1)];
 }

-static inline struct hlist_head *dev_index_hash(int ifindex)
+static inline struct hlist_head *dev_index_hash(net_t net, int ifindex)
 {
- return &dev_index_head[ifindex & ((1<<NETDEV_HASHBITS)-1)];
+ return &per_net(dev_index_head, net)[ifindex & ((1<<NETDEV_HASHBITS)-1)];
 }

 /*
@@ -418,7 +419,7 @@ unsigned long netdev_boot_base(const char *prefix, int unit)
   * If device already registered then return base of 1
   * to indicate not to probe for this interface
   */
- if (__dev_get_by_name(name))
+ if (__dev_get_by_name(init_net(), name))
   return 1;

  for (i = 0; i < NETDEV_BOOT_SETUP_MAX; i++)
@@ -473,11 +474,11 @@ __setup("netdev=", netdev_boot_setup);
  * careful with locks.
  */

-struct net_device *__dev_get_by_name(const char *name)
+struct net_device *__dev_get_by_name(net_t net, const char *name)
```

```
 {
  struct hlist_node *p;

- hlist_for_each(p, dev_name_hash(name)) {
+ hlist_for_each(p, dev_name_hash(net, name)) {
    struct net_device *dev
     = hlist_entry(p, struct net_device, name_hlist);
    if (!strncmp(dev->name, name, IFNAMSIZ))
@@ -497,15 +498,15 @@ struct net_device *__dev_get_by_name(const char *name)
  * matching device is found.
  */

-struct net_device *dev_get_by_name(const char *name)
+struct net_device *dev_get_by_name(net_t net, const char *name)
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_name(name);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_name(net, name);
  if (dev)
   dev_hold(dev);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  return dev;
 }

@@ -520,11 +521,11 @@ struct net_device *dev_get_by_name(const char *name)
  * or @dev_base_lock.
  */

-struct net_device *__dev_get_by_index(int ifindex)
+struct net_device *__dev_get_by_index(net_t net, int ifindex)
 {
  struct hlist_node *p;

- hlist_for_each(p, dev_index_hash(ifindex)) {
+ hlist_for_each(p, dev_index_hash(net, ifindex)) {
    struct net_device *dev
     = hlist_entry(p, struct net_device, index_hlist);
    if (dev->ifindex == ifindex)
@@ -544,15 +545,15 @@ struct net_device *__dev_get_by_index(int ifindex)
  * dev_put to indicate they have finished with it.
  */

-struct net_device *dev_get_by_index(int ifindex)
+struct net_device *dev_get_by_index(net_t net, int ifindex)
```

```
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_index(ifindex);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_index(net, ifindex);
  if (dev)
   dev_hold(dev);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  return dev;
 }

@@ -570,13 +571,13 @@ struct net_device *dev_get_by_index(int ifindex)
  * If the API was consistent this would be __dev_get_by_hwaddr
  */

-struct net_device *dev_getbyhwaddr(unsigned short type, char *ha)
+struct net_device *dev_getbyhwaddr(net_t net, unsigned short type, char *ha)
 {
  struct net_device *dev;

  ASSERT_RTNL();

- for (dev = dev_base; dev; dev = dev->next)
+ for (dev = per_net(dev_base, net); dev; dev = dev->next)
   if (dev->type == type &&
       !memcmp(dev->dev_addr, ha, dev->addr_len))
    break;
@@ -585,12 +586,12 @@ struct net_device *dev_getbyhwaddr(unsigned short type, char *ha)

 EXPORT_SYMBOL(dev_getbyhwaddr);

-struct net_device *dev_getfirstbyhwtype(unsigned short type)
+struct net_device *dev_getfirstbyhwtype(net_t net, unsigned short type)
 {
  struct net_device *dev;

  rtnl_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, net); dev; dev = dev->next) {
   if (dev->type == type) {
    dev_hold(dev);
    break;
@@ -613,18 +614,18 @@ EXPORT_SYMBOL(dev_getfirstbyhwtype);
  * dev_put to indicate they have finished with it.
  */
```

```
-struct net_device * dev_get_by_flags(unsigned short if_flags, unsigned short mask)
+struct net_device * dev_get_by_flags(net_t net, unsigned short if_flags, unsigned short mask)
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, net));
+ for (dev = per_net(dev_base, net); dev != NULL; dev = dev->next) {
   if (((dev->flags ^ if_flags) & mask) == 0) {
    dev_hold(dev);
    break;
   }
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  return dev;
 }

@@ -675,6 +676,10 @@ int dev_alloc_name(struct net_device *dev, const char *name)
  const int max_netdevices = 8*PAGE_SIZE;
  long *inuse;
  struct net_device *d;
+ net_t net;
+
+ BUG_ON(null_net(dev->nd_net));
+ net = dev->nd_net;

  p = strnchr(name, IFNAMSIZ-1, '%');
  if (p) {
@@ -691,7 +696,7 @@ int dev_alloc_name(struct net_device *dev, const char *name)
   if (!inuse)
    return -ENOMEM;

-  for (d = dev_base; d; d = d->next) {
+  for (d = per_net(dev_base, net); d; d = d->next) {
    if (!sscanf(d->name, name, &i))
     continue;
    if (i < 0 || i >= max_netdevices)
@@ -708,7 +713,7 @@ int dev_alloc_name(struct net_device *dev, const char *name)
  }

  snprintf(buf, sizeof(buf), name, i);
- if (!__dev_get_by_name(buf)) {
+ if (!__dev_get_by_name(net, buf)) {
   strlcpy(dev->name, buf, IFNAMSIZ);
   return i;
```

```
  }
@@ -732,9 +737,12 @@ int dev_alloc_name(struct net_device *dev, const char *name)
 int dev_change_name(struct net_device *dev, char *newname)
 {
  int err = 0;
+ net_t net;

  ASSERT_RTNL();
+ BUG_ON(null_net(dev->nd_net));

+ net = dev->nd_net;
  if (dev->flags & IFF_UP)
   return -EBUSY;

@@ -747,7 +755,7 @@ int dev_change_name(struct net_device *dev, char *newname)
    return err;
   strcpy(newname, dev->name);
  }
- else if (__dev_get_by_name(newname))
+ else if (__dev_get_by_name(net, newname))
   return -EEXIST;
  else
   strlcpy(dev->name, newname, IFNAMSIZ);
@@ -755,7 +763,7 @@ int dev_change_name(struct net_device *dev, char *newname)
  err = class_device_rename(&dev->class_dev, dev->name);
  if (!err) {
   hlist_del(&dev->name_hlist);
-  hlist_add_head(&dev->name_hlist, dev_name_hash(dev->name));
+  hlist_add_head(&dev->name_hlist, dev_name_hash(net, dev->name));
   raw_notifier_call_chain(&netdev_chain,
     NETDEV_CHANGENAME, dev);
  }
@@ -801,13 +809,13 @@ void netdev_state_change(struct net_device *dev)
  * available in this kernel then it becomes a nop.
  */

-void dev_load(const char *name)
+void dev_load(net_t net, const char *name)
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_name(name);
- read_unlock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_name(net, name);
+ read_unlock(&per_net(dev_base_lock, net));
```

```
  if (!dev && capable(CAP_SYS_MODULE))
   request_module("%s", name);
@@ -977,11 +985,14 @@ int register_netdevice_notifier(struct notifier_block *nb)
  rtnl_lock();
  err = raw_notifier_chain_register(&netdev_chain, nb);
  if (!err) {
-  for (dev = dev_base; dev; dev = dev->next) {
-   nb->notifier_call(nb, NETDEV_REGISTER, dev);
+  net_t net;
+  for_each_net(net) {
+   for (dev = per_net(dev_base, net); dev; dev = dev->next) {
+    nb->notifier_call(nb, NETDEV_REGISTER, dev);

-   if (dev->flags & IFF_UP)
-    nb->notifier_call(nb, NETDEV_UP, dev);
+    if (dev->flags & IFF_UP)
+     nb->notifier_call(nb, NETDEV_UP, dev);
+   }
   }
  }
  rtnl_unlock();
@@ -1991,7 +2002,7 @@ int register_gifconf(unsigned int family, gifconf_func_t * gifconf)
  * match.  --pb
  */

-static int dev_ifname(struct ifreq __user *arg)
+static int dev_ifname(net_t net, struct ifreq __user *arg)
 {
  struct net_device *dev;
  struct ifreq ifr;
@@ -2003,15 +2014,15 @@ static int dev_ifname(struct ifreq __user *arg)
  if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
   return -EFAULT;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_index(ifr.ifr_ifindex);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_index(net, ifr.ifr_ifindex);
  if (!dev) {
-  read_unlock(&dev_base_lock);
+  read_unlock(&per_net(dev_base_lock, net));
   return -ENODEV;
  }

  strcpy(ifr.ifr_name, dev->name);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
```

```
  if (copy_to_user(arg, &ifr, sizeof(struct ifreq)))
   return -EFAULT;
@@ -2024,7 +2035,7 @@ static int dev_ifname(struct ifreq __user *arg)
  * Thus we will need a 'compatibility mode'.
  */

-static int dev_ifconf(char __user *arg)
+static int dev_ifconf(net_t net, char __user *arg)
 {
  struct ifconf ifc;
  struct net_device *dev;
@@ -2048,7 +2059,7 @@ static int dev_ifconf(char __user *arg)
   */

  total = 0;
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, net); dev; dev = dev->next) {
   for (i = 0; i < NPROTO; i++) {
    if (gifconf_list[i]) {
     int done;
@@ -2080,31 +2091,35 @@ static int dev_ifconf(char __user *arg)
  * This is invoked by the /proc filesystem handler to display a device
  * in detail.
  */
-static __inline__ struct net_device *dev_get_idx(loff_t pos)
+static __inline__ struct net_device *dev_get_idx(net_t net, loff_t pos)
 {
  struct net_device *dev;
  loff_t i;

- for (i = 0, dev = dev_base; dev && i < pos; ++i, dev = dev->next);
+ for (i = 0, dev = per_net(dev_base, net); dev && i < pos; ++i, dev = dev->next);

  return i == pos ? dev : NULL;
 }

 void *dev_seq_start(struct seq_file *seq, loff_t *pos)
 {
- read_lock(&dev_base_lock);
- return *pos ? dev_get_idx(*pos - 1) : SEQ_START_TOKEN;
+ net_t net = net_from_voidp(seq->private);
+
+ read_lock(&per_net(dev_base_lock, net));
+ return *pos ? dev_get_idx(net, *pos - 1) : SEQ_START_TOKEN;
 }

 void *dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
 {
```

```
+ net_t net = net_from_voidp(seq->private);
  ++*pos;
- return v == SEQ_START_TOKEN ? dev_base : ((struct net_device *)v)->next;
+ return v == SEQ_START_TOKEN ? per_net(dev_base, net) : ((struct net_device *)v)->next;
 }

 void dev_seq_stop(struct seq_file *seq, void *v)
 {
- read_unlock(&dev_base_lock);
+ net_t net = net_from_voidp(seq->private);
+ read_unlock(&per_net(dev_base_lock, net));
 }

 static void dev_seq_printf_stats(struct seq_file *seq, struct net_device *dev)
@@ -2198,7 +2213,22 @@ static struct seq_operations dev_seq_ops = {

 static int dev_seq_open(struct inode *inode, struct file *file)
 {
- return seq_open(file, &dev_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res =  seq_open(file, &dev_seq_ops);
+ if (!res) {
+  seq = file->private_data;
+  seq->private = net_to_voidp(get_net(PROC_NET(inode)));
+ }
+ return res;
+}
+
+static int dev_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
 }

 static struct file_operations dev_seq_fops = {
@@ -2206,7 +2236,7 @@ static struct file_operations dev_seq_fops = {
  .open    = dev_seq_open,
  .read    = seq_read,
  .llseek  = seq_lseek,
- .release = seq_release,
+ .release = dev_seq_release,
 };

 static struct seq_operations softnet_seq_ops = {
@@ -2235,23 +2265,44 @@ extern int wireless_proc_init(void);
```

```
 #define wireless_proc_init() 0
 #endif

-static int __init dev_proc_init(void)
+static int dev_proc_net_init(net_t net)
 {
  int rc = -ENOMEM;

- if (!proc_net_fops_create(init_net(), "dev", S_IRUGO, &dev_seq_fops))
+ if (!proc_net_fops_create(net, "dev", S_IRUGO, &dev_seq_fops))
    goto out;
- if (!proc_net_fops_create(init_net(), "softnet_stat", S_IRUGO, &softnet_seq_fops))
+ if (!proc_net_fops_create(net, "softnet_stat", S_IRUGO, &softnet_seq_fops))
    goto out_dev;
- if (wireless_proc_init())
-  goto out_softnet;
  rc = 0;
 out:
  return rc;
-out_softnet:
- proc_net_remove(init_net(), "softnet_stat");
 out_dev:
- proc_net_remove(init_net(), "dev");
+ proc_net_remove(net, "dev");
+ goto out;
+}
+
+static void dev_proc_net_exit(net_t net)
+{
+ proc_net_remove(net, "softnet_stat");
+ proc_net_remove(net, "dev");
+}
+
+static struct pernet_operations dev_proc_ops = {
+ .init = dev_proc_net_init,
+ .exit = dev_proc_net_exit,
+};
+
+static int __init dev_proc_init(void)
+{
+ int rc;
+ if ((rc = register_pernet_subsys(&dev_proc_ops)))
+  goto out;
+ if ((rc = wireless_proc_init()))
+  goto out_softnet;
+out:
+ return rc;
+out_softnet:
```

```
+ unregister_pernet_subsys(&dev_proc_ops);
  goto out;
 }
 #else
@@ -2485,10 +2536,10 @@ int dev_set_mac_address(struct net_device *dev, struct sockaddr
*sa)
 /*
  * Perform the SIOCxIFxxx calls.
  */
-static int dev_ifsioc(struct ifreq *ifr, unsigned int cmd)
+static int dev_ifsioc(net_t net, struct ifreq *ifr, unsigned int cmd)
 {
  int err;
- struct net_device *dev = __dev_get_by_name(ifr->ifr_name);
+ struct net_device *dev = __dev_get_by_name(net, ifr->ifr_name);

  if (!dev)
   return -ENODEV;
@@ -2641,7 +2692,7 @@ static int dev_ifsioc(struct ifreq *ifr, unsigned int cmd)
  * positive or a negative errno code on error.
  */

-int dev_ioctl(unsigned int cmd, void __user *arg)
+int dev_ioctl(net_t net, unsigned int cmd, void __user *arg)
 {
  struct ifreq ifr;
  int ret;
@@ -2654,12 +2705,12 @@ int dev_ioctl(unsigned int cmd, void __user *arg)

  if (cmd == SIOCGIFCONF) {
   rtnl_lock();
-  ret = dev_ifconf((char __user *) arg);
+  ret = dev_ifconf(net, (char __user *) arg);
   rtnl_unlock();
   return ret;
  }
  if (cmd == SIOCGIFNAME)
-  return dev_ifname((struct ifreq __user *)arg);
+  return dev_ifname(net, (struct ifreq __user *)arg);

  if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
   return -EFAULT;
@@ -2689,10 +2740,10 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
   case SIOCGIFMAP:
   case SIOCGIFINDEX:
   case SIOCGIFTXQLEN:
-   dev_load(ifr.ifr_name);
-   read_lock(&dev_base_lock);
```

```
-   ret = dev_ifsioc(&ifr, cmd);
-   read_unlock(&dev_base_lock);
+   dev_load(net, ifr.ifr_name);
+   read_lock(&per_net(dev_base_lock, net));
+   ret = dev_ifsioc(net, &ifr, cmd);
+   read_unlock(&per_net(dev_base_lock, net));
    if (!ret) {
     if (colon)
      *colon = ':';
@@ -2703,9 +2754,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    return ret;

   case SIOCETHTOOL:
-   dev_load(ifr.ifr_name);
+   dev_load(net, ifr.ifr_name);
    rtnl_lock();
-   ret = dev_ethtool(&ifr);
+   ret = dev_ethtool(net, &ifr);
    rtnl_unlock();
    if (!ret) {
     if (colon)
@@ -2727,9 +2778,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
   case SIOCSIFNAME:
    if (!capable(CAP_NET_ADMIN))
     return -EPERM;
-   dev_load(ifr.ifr_name);
+   dev_load(net, ifr.ifr_name);
    rtnl_lock();
-   ret = dev_ifsioc(&ifr, cmd);
+   ret = dev_ifsioc(net, &ifr, cmd);
    rtnl_unlock();
    if (!ret) {
     if (colon)
@@ -2768,9 +2819,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    /* fall through */
   case SIOCBONDSLAVEINFOQUERY:
   case SIOCBONDINFOQUERY:
-   dev_load(ifr.ifr_name);
+   dev_load(net, ifr.ifr_name);
    rtnl_lock();
-   ret = dev_ifsioc(&ifr, cmd);
+   ret = dev_ifsioc(net, &ifr, cmd);
    rtnl_unlock();
    return ret;

@@ -2790,9 +2841,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    if (cmd == SIOCWANDEV ||
        (cmd >= SIOCDEVPRIVATE &&
```

```
        cmd <= SIOCDEVPRIVATE + 15)) {
-    dev_load(ifr.ifr_name);
+    dev_load(net, ifr.ifr_name);
     rtnl_lock();
-    ret = dev_ifsioc(&ifr, cmd);
+    ret = dev_ifsioc(net, &ifr, cmd);
     rtnl_unlock();
     if (!ret && copy_to_user(arg, &ifr,
         sizeof(struct ifreq)))
@@ -2810,10 +2861,10 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
      if (!capable(CAP_NET_ADMIN))
       return -EPERM;
     }
-    dev_load(ifr.ifr_name);
+    dev_load(net, ifr.ifr_name);
     rtnl_lock();
     /* Follow me in net/core/wireless.c */
-    ret = wireless_process_ioctl(&ifr, cmd);
+    ret = wireless_process_ioctl(net, &ifr, cmd);
     rtnl_unlock();
     if (IW_IS_GET(cmd) &&
         copy_to_user(arg, &ifr,
@@ -2834,13 +2885,13 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
  * number.  The caller must hold the rtnl semaphore or the
  * dev_base_lock to be sure it remains unique.
  */
-static int dev_new_index(void)
+static int dev_new_index(net_t net)
 {
  static int ifindex;
  for (;;) {
   if (++ifindex <= 0)
    ifindex = 1;
-  if (!__dev_get_by_index(ifindex))
+  if (!__dev_get_by_index(net, ifindex))
    return ifindex;
  }
 }
@@ -2880,6 +2931,7 @@ int register_netdevice(struct net_device *dev)
  struct hlist_head *head;
  struct hlist_node *p;
  int ret;
+ net_t net;

  BUG_ON(dev_boot_phase);
  ASSERT_RTNL();
@@ -2888,6 +2940,8 @@ int register_netdevice(struct net_device *dev)
```

```
  /* When net_device's are persistent, this will be fatal. */
  BUG_ON(dev->reg_state != NETREG_UNINITIALIZED);
+ BUG_ON(null_net(dev->nd_net));
+ net = dev->nd_net;

  spin_lock_init(&dev->queue_lock);
  spin_lock_init(&dev->_xmit_lock);
@@ -2913,12 +2967,12 @@ int register_netdevice(struct net_device *dev)
   goto out;
  }

- dev->ifindex = dev_new_index();
+ dev->ifindex = dev_new_index(net);
  if (dev->iflink == -1)
   dev->iflink = dev->ifindex;

  /* Check for existence of name */
- head = dev_name_hash(dev->name);
+ head = dev_name_hash(net, dev->name);
  hlist_for_each(p, head) {
   struct net_device *d
    = hlist_entry(p, struct net_device, name_hlist);
@@ -2980,13 +3034,13 @@ int register_netdevice(struct net_device *dev)

  dev->next = NULL;
  dev_init_scheduler(dev);
- write_lock_bh(&dev_base_lock);
- *dev_tail = dev;
- dev_tail = &dev->next;
+ write_lock_bh(&per_net(dev_base_lock, net));
+ *per_net(dev_tail, net) = dev;
+ per_net(dev_tail, net) = &dev->next;
  hlist_add_head(&dev->name_hlist, head);
- hlist_add_head(&dev->index_hlist, dev_index_hash(dev->ifindex));
+ hlist_add_head(&dev->index_hlist, dev_index_hash(net, dev->ifindex));
  dev_hold(dev);
- write_unlock_bh(&dev_base_lock);
+ write_unlock_bh(&per_net(dev_base_lock, net));

  /* Notify protocols, that a new device appeared. */
  raw_notifier_call_chain(&netdev_chain, NETDEV_REGISTER, dev);
@@ -3252,6 +3306,7 @@ void synchronize_net(void)
 int unregister_netdevice(struct net_device *dev)
 {
  struct net_device *d, **dp;
+ net_t net = dev->nd_net;

  BUG_ON(dev_boot_phase);
```

```
  ASSERT_RTNL();
@@ -3270,15 +3325,15 @@ int unregister_netdevice(struct net_device *dev)
    dev_close(dev);

   /* And unlink it from device chain. */
- for (dp = &dev_base; (d = *dp) != NULL; dp = &d->next) {
+ for (dp = &per_net(dev_base, net); (d = *dp) != NULL; dp = &d->next) {
    if (d == dev) {
-    write_lock_bh(&dev_base_lock);
+    write_lock_bh(&per_net(dev_base_lock, net));
     hlist_del(&dev->name_hlist);
     hlist_del(&dev->index_hlist);
-    if (dev_tail == &dev->next)
-     dev_tail = dp;
+    if (per_net(dev_tail, net) == &dev->next)
+     per_net(dev_tail, net) = dp;
     *dp = d->next;
-    write_unlock_bh(&dev_base_lock);
+    write_unlock_bh(&per_net(dev_base_lock, net));
     break;
    }
   }
@@ -3464,6 +3519,26 @@ static int __init netdev_dma_register(void)
 static int __init netdev_dma_register(void) { return -ENODEV; }
 #endif /* CONFIG_NET_DMA */

+/* Initialize per network namespace state */
+static int netdev_init(net_t net)
+{
+ int i;
+ per_net(dev_tail, net) = &per_net(dev_base, net);
+ rwlock_init(&per_net(dev_base_lock, net));
+
+ for (i = 0; i < ARRAY_SIZE(per_net(dev_name_head, net)); i++)
+  INIT_HLIST_HEAD(&per_net(dev_name_head, net)[i]);
+
+ for (i = 0; i < ARRAY_SIZE(per_net(dev_index_head, net)); i++)
+  INIT_HLIST_HEAD(&per_net(dev_index_head, net)[i]);
+
+ return 0;
+}
+
+static struct pernet_operations netdev_net_ops = {
+ .init = netdev_init,
+};
+
 /*
  * Initialize the DEV module. At boot time this walks the device list and
```

```
  * unhooks any devices that fail to initialise (normally hardware not
@@ -3491,11 +3566,8 @@ static int __init net_dev_init(void)
  for (i = 0; i < 16; i++)
   INIT_LIST_HEAD(&ptype_base[i]);

- for (i = 0; i < ARRAY_SIZE(dev_name_head); i++)
-  INIT_HLIST_HEAD(&dev_name_head[i]);
-
- for (i = 0; i < ARRAY_SIZE(dev_index_head); i++)
-  INIT_HLIST_HEAD(&dev_index_head[i]);
+ if (register_pernet_subsys(&netdev_net_ops))
+  goto out;

  /*
   * Initialise the packet receive queues.
diff --git a/net/core/dev_mcast.c b/net/core/dev_mcast.c
index 623e606..131746b 100644
--- a/net/core/dev_mcast.c
+++ b/net/core/dev_mcast.c
@@ -221,11 +221,12 @@ void dev_mc_discard(struct net_device *dev)
 #ifdef CONFIG_PROC_FS
 static void *dev_mc_seq_start(struct seq_file *seq, loff_t *pos)
 {
+ net_t net = net_from_voidp(seq->private);
  struct net_device *dev;
  loff_t off = 0;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, net));
+ for (dev = per_net(dev_base, net); dev; dev = dev->next) {
   if (off++ == *pos)
    return dev;
  }
@@ -241,7 +242,8 @@ static void *dev_mc_seq_next(struct seq_file *seq, void *v, loff_t *pos)

 static void dev_mc_seq_stop(struct seq_file *seq, void *v)
 {
- read_unlock(&dev_base_lock);
+ net_t net = net_from_voidp(seq->private);
+ read_unlock(&per_net(dev_base_lock, net));
 }


@@ -275,7 +277,22 @@ static struct seq_operations dev_mc_seq_ops = {

 static int dev_mc_seq_open(struct inode *inode, struct file *file)
 {
```

```
- return seq_open(file, &dev_mc_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res = seq_open(file, &dev_mc_seq_ops);
+ if (!res) {
+  seq = file->private_data;
+  seq->private = net_to_voidp(get_net(PROC_NET(inode)));
+ }
+ return res;
+}
+
+static int dev_mc_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
 }

 static struct file_operations dev_mc_seq_fops = {
@@ -283,14 +300,31 @@ static struct file_operations dev_mc_seq_fops = {
  .open    = dev_mc_seq_open,
  .read    = seq_read,
  .llseek  = seq_lseek,
- .release = seq_release,
+ .release = dev_mc_seq_release,
 };

 #endif

+static int dev_mc_net_init(net_t net)
+{
+ if (!proc_net_fops_create(net, "dev_mcast", 0, &dev_mc_seq_fops))
+  return -ENOMEM;
+ return 0;
+}
+
+static void dev_mc_net_exit(net_t net)
+{
+ proc_net_remove(net, "dev_mcast");
+}
+
+static struct pernet_operations dev_mc_net_ops = {
+ .init = dev_mc_net_init,
+ .exit = dev_mc_net_exit,
+};
+
 void __init dev_mcast_init(void)
```

```
 {
- proc_net_fops_create(init_net(), "dev_mcast", 0, &dev_mc_seq_fops);
+ register_pernet_subsys(&dev_mc_net_ops);
 }

 EXPORT_SYMBOL(dev_mc_add);
diff --git a/net/core/ethtool.c b/net/core/ethtool.c
index 87dc556..d142377 100644
--- a/net/core/ethtool.c
+++ b/net/core/ethtool.c
@@ -798,9 +798,9 @@ static int ethtool_get_perm_addr(struct net_device *dev, void __user
*useraddr)

 /* The main entry point in this file.  Called from net/core/dev.c */

-int dev_ethtool(struct ifreq *ifr)
+int dev_ethtool(net_t net, struct ifreq *ifr)
 {
- struct net_device *dev = __dev_get_by_name(ifr->ifr_name);
+ struct net_device *dev = __dev_get_by_name(net, ifr->ifr_name);
  void __user *useraddr = ifr->ifr_data;
  u32 ethcmd;
  int rc;
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
index ffc31c1..2fa2708 100644
--- a/net/core/fib_rules.c
+++ b/net/core/fib_rules.c
@@ -12,6 +12,7 @@
 #include <linux/kernel.h>
 #include <linux/list.h>
 #include <net/net_namespace.h>
+#include <net/sock.h>
 #include <net/fib_rules.h>

 static LIST_HEAD(rules_ops);
@@ -155,6 +156,7 @@ EXPORT_SYMBOL_GPL(fib_rules_lookup);

 int fib_nl_newrule(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct fib_rule_hdr *frh = nlmsg_data(nlh);
  struct fib_rules_ops *ops = NULL;
  struct fib_rule *rule, *r, *last = NULL;
@@ -188,7 +190,7 @@ int fib_nl_newrule(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)

   rule->ifindex = -1;
   nla_strlcpy(rule->ifname, tb[FRA_IFNAME], IFNAMSIZ);
-  dev = __dev_get_by_name(rule->ifname);
```

```
+  dev = __dev_get_by_name(net, rule->ifname);
   if (dev)
    rule->ifindex = dev->ifindex;
  }
diff --git a/net/core/link_watch.c b/net/core/link_watch.c
index 549a2ce..8e317cf 100644
--- a/net/core/link_watch.c
+++ b/net/core/link_watch.c
@@ -63,12 +63,13 @@ static unsigned char default_operstate(const struct net_device *dev)

 static void rfc2863_policy(struct net_device *dev)
 {
+ net_t net = dev->nd_net;
  unsigned char operstate = default_operstate(dev);

  if (operstate == dev->operstate)
   return;

- write_lock_bh(&dev_base_lock);
+ write_lock_bh(&per_net(dev_base_lock, net));

  switch(dev->link_mode) {
  case IF_LINK_MODE_DORMANT:
@@ -83,7 +84,7 @@ static void rfc2863_policy(struct net_device *dev)

  dev->operstate = operstate;

- write_unlock_bh(&dev_base_lock);
+ write_unlock_bh(&per_net(dev_base_lock, net));
 }


diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 90e1d2e..f5d4f92 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -1438,6 +1438,7 @@ int neigh_table_clear(struct neigh_table *tbl)

 int neigh_delete(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ndmsg *ndm;
  struct nlattr *dst_attr;
  struct neigh_table *tbl;
@@ -1453,7 +1454,7 @@ int neigh_delete(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)

  ndm = nlmsg_data(nlh);
  if (ndm->ndm_ifindex) {
```

```
-  dev = dev_get_by_index(ndm->ndm_ifindex);
+  dev = dev_get_by_index(net, ndm->ndm_ifindex);
   if (dev == NULL) {
    err = -ENODEV;
    goto out;
@@ -1503,6 +1504,7 @@ out:

 int neigh_add(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ndmsg *ndm;
  struct nlattr *tb[NDA_MAX+1];
  struct neigh_table *tbl;
@@ -1519,7 +1521,7 @@ int neigh_add(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)

  ndm = nlmsg_data(nlh);
  if (ndm->ndm_ifindex) {
-  dev = dev_get_by_index(ndm->ndm_ifindex);
+  dev = dev_get_by_index(net, ndm->ndm_ifindex);
   if (dev == NULL) {
    err = -ENODEV;
    goto out;
diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
index b08c1be..1be6f94 100644
--- a/net/core/net-sysfs.c
+++ b/net/core/net-sysfs.c
@@ -38,12 +38,13 @@ static ssize_t netdev_show(const struct class_device *cd, char *buf,
       ssize_t (*format)(const struct net_device *, char *))
 {
  struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
  ssize_t ret = -EINVAL;
-
- read_lock(&dev_base_lock);
+
+ read_lock(&per_net(dev_base_lock, net));
  if (dev_isalive(dev))
   ret = (*format)(dev, buf);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

  return ret;
 }
@@ -109,12 +110,13 @@ static ssize_t format_addr(char *buf, const unsigned char *addr, int
len)
 static ssize_t show_address(struct class_device *cd, char *buf)
 {
  struct net_device *dev = to_net_dev(cd);
```

```
+ net_t net = dev->nd_net;
  ssize_t ret = -EINVAL;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
  if (dev_isalive(dev))
      ret = format_addr(buf, dev->dev_addr, dev->addr_len);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  return ret;
 }

@@ -158,13 +160,14 @@ static const char *operstates[] = {
 static ssize_t show_operstate(struct class_device *cd, char *buf)
 {
  const struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
  unsigned char operstate;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
  operstate = dev->operstate;
  if (!netif_running(dev))
   operstate = IF_OPER_DOWN;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

  if (operstate >= ARRAY_SIZE(operstates))
   return -EINVAL; /* should not happen */
@@ -248,6 +251,7 @@ static ssize_t netstat_show(const struct class_device *cd, char *buf,
       unsigned long offset)
 {
  struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
  struct net_device_stats *stats;
  ssize_t ret = -EINVAL;

@@ -255,13 +259,13 @@ static ssize_t netstat_show(const struct class_device *cd, char *buf,
      offset % sizeof(unsigned long) != 0)
   WARN_ON(1);

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
  if (dev_isalive(dev) && dev->get_stats &&
      (stats = (*dev->get_stats)(dev)))
   ret = sprintf(buf, fmt_ulong,
         *(unsigned long *)(((u8 *) stats) + offset));
```

```
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  return ret;
 }

@@ -338,10 +342,11 @@ static ssize_t wireless_show(struct class_device *cd, char *buf,
            char *))
 {
  struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
  const struct iw_statistics *iw = NULL;
  ssize_t ret = -EINVAL;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
  if (dev_isalive(dev)) {
   if(dev->wireless_handlers &&
      dev->wireless_handlers->get_wireless_stats)
@@ -349,7 +354,7 @@ static ssize_t wireless_show(struct class_device *cd, char *buf,
   if (iw != NULL)
    ret = (*format)(iw, buf);
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

  return ret;
 }
diff --git a/net/core/netpoll.c b/net/core/netpoll.c
index 823215d..f2c7c07 100644
--- a/net/core/netpoll.c
+++ b/net/core/netpoll.c
@@ -621,7 +621,7 @@ int netpoll_setup(struct netpoll *np)
  int err;

  if (np->dev_name)
-  ndev = dev_get_by_name(np->dev_name);
+  ndev = dev_get_by_name(init_net(), np->dev_name);
  if (!ndev) {
   printk(KERN_ERR "%s: %s doesn't exist, aborting.\n",
          np->name, np->dev_name);
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 7796b39..a415efb 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -1928,7 +1928,7 @@ static struct net_device *pktgen_setup_dev(struct pktgen_dev
*pkt_dev)
   pkt_dev->odev = NULL;
  }
```

```
- odev = dev_get_by_name(pkt_dev->ifname);
+ odev = dev_get_by_name(init_net(), pkt_dev->ifname);

  if (!odev) {
   printk("pktgen: no such netdevice: \"%s\"\n", pkt_dev->ifname);
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index 8f3dda8..5ac07a0 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -235,6 +235,7 @@ EXPORT_SYMBOL_GPL(rtnl_put_cacheinfo);

 static void set_operstate(struct net_device *dev, unsigned char transition)
 {
+ net_t net = dev->nd_net;
  unsigned char operstate = dev->operstate;

  switch(transition) {
@@ -253,9 +254,9 @@ static void set_operstate(struct net_device *dev, unsigned char
transition)
  };

  if (dev->operstate != operstate) {
-  write_lock_bh(&dev_base_lock);
+  write_lock_bh(&per_net(dev_base_lock, net));
   dev->operstate = operstate;
-  write_unlock_bh(&dev_base_lock);
+  write_unlock_bh(&per_net(dev_base_lock, net));
   netdev_state_change(dev);
  }
 }
@@ -389,12 +390,13 @@ nla_put_failure:

 static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int idx;
  int s_idx = cb->args[0];
  struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, net));
+ for (dev=per_net(dev_base, net), idx=0; dev; dev = dev->next, idx++) {
   if (idx < s_idx)
    continue;
   if (rtnl_fill_ifinfo(skb, dev, NULL, 0, RTM_NEWLINK,
@@ -402,7 +404,7 @@ static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
```

```
         cb->nlh->nlmsg_seq, 0, NLM_F_MULTI) <= 0)
    break;
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  cb->args[0] = idx;

  return skb->len;
@@ -420,6 +422,7 @@ static struct nla_policy ifla_policy[IFLA_MAX+1] __read_mostly = {

 static int rtnl_setlink(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ifinfomsg *ifm;
  struct net_device *dev;
  int err, send_addr_notify = 0, modified = 0;
@@ -438,9 +441,9 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
  err = -EINVAL;
  ifm = nlmsg_data(nlh);
  if (ifm->ifi_index >= 0)
-  dev = dev_get_by_index(ifm->ifi_index);
+  dev = dev_get_by_index(net, ifm->ifi_index);
  else if (tb[IFLA_IFNAME])
-  dev = dev_get_by_name(ifname);
+  dev = dev_get_by_name(net, ifname);
  else
   goto errout;

@@ -566,9 +569,9 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
   set_operstate(dev, nla_get_u8(tb[IFLA_OPERSTATE]));

  if (tb[IFLA_LINKMODE]) {
-  write_lock_bh(&dev_base_lock);
+  write_lock_bh(&per_net(dev_base_lock, net));
   dev->link_mode = nla_get_u8(tb[IFLA_LINKMODE]);
-  write_unlock_bh(&dev_base_lock);
+  write_unlock_bh(&per_net(dev_base_lock, net));
  }

  err = 0;
@@ -590,6 +593,7 @@ errout:

 static int rtnl_getlink(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ifinfomsg *ifm;
  struct nlattr *tb[IFLA_MAX+1];
  struct net_device *dev = NULL;
```

```
@@ -604,7 +608,7 @@ static int rtnl_getlink(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)

  ifm = nlmsg_data(nlh);
  if (ifm->ifi_index >= 0) {
-  dev = dev_get_by_index(ifm->ifi_index);
+  dev = dev_get_by_index(net, ifm->ifi_index);
   if (dev == NULL)
    return -ENODEV;
  } else
diff --git a/net/core/sock.c b/net/core/sock.c
index e42f7df..737838c 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -343,6 +343,7 @@ int sock_setsockopt(struct socket *sock, int level, int optname,
       char __user *optval, int optlen)
 {
  struct sock *sk=sock->sk;
+ net_t net = sk->sk_net;
  struct sk_filter *filter;
  int val;
  int valbool;
@@ -579,7 +580,7 @@ set_rcvbuf:
     if (devname[0] == '\0') {
      sk->sk_bound_dev_if = 0;
     } else {
-     struct net_device *dev = dev_get_by_name(devname);
+     struct net_device *dev = dev_get_by_name(net, devname);
      if (!dev) {
       ret = -ENODEV;
       break;
diff --git a/net/core/wireless.c b/net/core/wireless.c
index faa242f..d1418bf 100644
--- a/net/core/wireless.c
+++ b/net/core/wireless.c
@@ -672,7 +672,22 @@ static struct seq_operations wireless_seq_ops = {

 static int wireless_seq_open(struct inode *inode, struct file *file)
 {
- return seq_open(file, &wireless_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res = seq_open(file, &wireless_seq_ops);
+ if (!res) {
+  seq = file->private_data;
+  seq->private = net_to_voidp(get_net(PROC_NET(inode)));
+ }
+ return res;
+}
```

```
+
+static int wireless_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
 }

 static struct file_operations wireless_seq_fops = {
@@ -680,17 +695,33 @@ static struct file_operations wireless_seq_fops = {
  .open    = wireless_seq_open,
  .read    = seq_read,
  .llseek  = seq_lseek,
- .release = seq_release,
+ .release = wireless_seq_release,
 };

-int __init wireless_proc_init(void)
+static int wireless_proc_net_init(net_t net)
 {
  /* Create /proc/net/wireless entry */
- if (!proc_net_fops_create(init_net(), "wireless", S_IRUGO, &wireless_seq_fops))
+ if (!proc_net_fops_create(net, "wireless", S_IRUGO, &wireless_seq_fops))
   return -ENOMEM;

  return 0;
 }
+
+static void wireless_proc_net_exit(net_t net)
+{
+ proc_net_remove(net, "wireless");
+}
+
+static struct pernet_operations wireless_proc_ops = {
+ .init = wireless_proc_net_init,
+ .exit = wireless_proc_net_exit,
+};
+
+int __init wireless_proc_init(void)
+{
+ return register_pernet_subsys(&wireless_proc_ops);
+}
+
 #endif /* CONFIG_PROC_FS */

 /************************* IOCTL SUPPORT *************************/
@@ -1066,7 +1097,7 @@ static inline int ioctl_private_call(struct net_device * dev,
```

```
  * (dev_ioctl() in net/core/dev.c).
  * Check the type of IOCTL and call the appropriate wrapper...
  */
-int wireless_process_ioctl(struct ifreq *ifr, unsigned int cmd)
+int wireless_process_ioctl(net_t net, struct ifreq *ifr, unsigned int cmd)
 {
  struct net_device *dev;
  iw_handler handler;
@@ -1075,7 +1106,7 @@ int wireless_process_ioctl(struct ifreq *ifr, unsigned int cmd)
   * The copy_to/from_user() of ifr is also dealt with in there */

  /* Make sure the device exist */
- if ((dev = __dev_get_by_name(ifr->ifr_name)) == NULL)
+ if ((dev = __dev_get_by_name(net, ifr->ifr_name)) == NULL)
   return -ENODEV;

  /* A bunch of special cases, then the generic case...
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index b27b2ac..1cc502a 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -749,14 +749,14 @@ static int dn_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len)

  if (!(saddr->sdn_flags & SDF_WILD)) {
   if (dn_ntohs(saddr->sdn_nodeaddrl)) {
-   read_lock(&dev_base_lock);
-   for(dev = dev_base; dev; dev = dev->next) {
+   read_lock(&per_net(dev_base_lock, init_net()));
+   for(dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
     if (!dev->dn_ptr)
      continue;
     if (dn_dev_islocal(dev, dn_saddr2dn(saddr)))
      break;
    }
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));
    if (dev == NULL)
     return -EADDRNOTAVAIL;
   }
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index dbaf001..c83c8d1 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -514,7 +514,7 @@ int dn_dev_ioctl(unsigned int cmd, void __user *arg)
  ifr->ifr_name[IFNAMSIZ-1] = 0;

 #ifdef CONFIG_KMOD
```

```
- dev_load(ifr->ifr_name);
+ dev_load(init_net(), ifr->ifr_name);
 #endif

  switch(cmd) {
@@ -532,7 +532,7 @@ int dn_dev_ioctl(unsigned int cmd, void __user *arg)

  rtnl_lock();

- if ((dev = __dev_get_by_name(ifr->ifr_name)) == NULL) {
+ if ((dev = __dev_get_by_name(init_net(), ifr->ifr_name)) == NULL) {
   ret = -ENODEV;
   goto done;
  }
@@ -630,7 +630,7 @@ static struct dn_dev *dn_dev_by_index(int ifindex)
 {
  struct net_device *dev;
  struct dn_dev *dn_dev = NULL;
- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
  if (dev) {
   dn_dev = dev->dn_ptr;
   dev_put(dev);
@@ -695,7 +695,7 @@ static int dn_nl_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg)
   return -EINVAL;

  ifm = nlmsg_data(nlh);
- if ((dev = __dev_get_by_index(ifm->ifa_index)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), ifm->ifa_index)) == NULL)
   return -ENODEV;

  if ((dn_db = dev->dn_ptr) == NULL) {
@@ -796,8 +796,8 @@ static int dn_nl_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
  skip_ndevs = cb->args[0];
  skip_naddr = cb->args[1];

- read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
   if (idx < skip_ndevs)
    continue;
   else if (idx > skip_ndevs) {
@@ -821,7 +821,7 @@ static int dn_nl_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
   }
```

```
  }
 done:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  cb->args[0] = idx;
  cb->args[1] = dn_idx;
@@ -862,9 +862,9 @@ int dn_dev_bind_default(__le16 *addr)
  dev = dn_dev_get_default();
 last_chance:
  if (dev) {
-  read_lock(&dev_base_lock);
+  read_lock(&per_net(dev_base_lock, init_net()));
   rv = dn_dev_get_first(dev, addr);
-  read_unlock(&dev_base_lock);
+  read_unlock(&per_net(dev_base_lock, init_net()));
   dev_put(dev);
   if (rv == 0 || dev == &per_net(loopback_dev, init_net()))
    return rv;
@@ -1296,7 +1296,7 @@ void dn_dev_devices_off(void)
  struct net_device *dev;

  rtnl_lock();
- for(dev = dev_base; dev; dev = dev->next)
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next)
   dn_dev_down(dev);
  rtnl_unlock();

@@ -1307,7 +1307,7 @@ void dn_dev_devices_on(void)
  struct net_device *dev;

  rtnl_lock();
- for(dev = dev_base; dev; dev = dev->next) {
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   if (dev->flags & IFF_UP)
    dn_dev_up(dev);
  }
@@ -1338,7 +1338,7 @@ static struct net_device *dn_dev_get_idx(struct seq_file *seq, loff_t
pos)
 {
  struct net_device *dev;

- dev = dev_base;
+ dev = per_net(dev_base, init_net());
  if (dev && !dev->dn_ptr)
   dev = dn_dev_get_next(seq, dev);
  if (pos) {
@@ -1352,10 +1352,10 @@ static void *dn_dev_seq_start(struct seq_file *seq, loff_t *pos)
```

```
 {
  if (*pos) {
   struct net_device *dev;
-  read_lock(&dev_base_lock);
+  read_lock(&per_net(dev_base_lock, init_net()));
   dev = dn_dev_get_idx(seq, *pos - 1);
   if (dev == NULL)
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));
   return dev;
  }
  return SEQ_START_TOKEN;
@@ -1371,7 +1371,7 @@ static void *dn_dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
  } else {
   dev = dn_dev_get_next(seq, dev);
   if (dev == NULL)
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));
  }
  ++*pos;
  return dev;
@@ -1380,7 +1380,7 @@ static void *dn_dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
 static void dn_dev_seq_stop(struct seq_file *seq, void *v)
 {
  if (v && v != SEQ_START_TOKEN)
-  read_unlock(&dev_base_lock);
+  read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static char *dn_type2asc(char type)
diff --git a/net/decnet/dn_fib.c b/net/decnet/dn_fib.c
index 1cf0101..cc2ab1f 100644
--- a/net/decnet/dn_fib.c
+++ b/net/decnet/dn_fib.c
@@ -212,7 +212,7 @@ static int dn_fib_check_nh(const struct rtmsg *r, struct dn_fib_info *fi,
struct
     return -EINVAL;
    if (dnet_addr_type(nh->nh_gw) != RTN_UNICAST)
     return -EINVAL;
-   if ((dev = __dev_get_by_index(nh->nh_oif)) == NULL)
+   if ((dev = __dev_get_by_index(init_net(), nh->nh_oif)) == NULL)
     return -ENODEV;
    if (!(dev->flags&IFF_UP))
     return -ENETDOWN;
@@ -255,7 +255,7 @@ out:
   if (nh->nh_flags&(RTNH_F_PERVASIVE|RTNH_F_ONLINK))
    return -EINVAL;
```

```
-  dev = __dev_get_by_index(nh->nh_oif);
+  dev = __dev_get_by_index(init_net(), nh->nh_oif);
   if (dev == NULL || dev->dn_ptr == NULL)
    return -ENODEV;
   if (!(dev->flags&IFF_UP))
@@ -352,7 +352,7 @@ struct dn_fib_info *dn_fib_create_info(const struct rtmsg *r, struct
dn_kern_rta
   if (nhs != 1 || nh->nh_gw)
    goto err_inval;
   nh->nh_scope = RT_SCOPE_NOWHERE;
-  nh->nh_dev = dev_get_by_index(fi->fib_nh->nh_oif);
+  nh->nh_dev = dev_get_by_index(init_net(), fi->fib_nh->nh_oif);
   err = -ENODEV;
   if (nh->nh_dev == NULL)
    goto failure;
@@ -598,8 +598,8 @@ static void dn_fib_del_ifaddr(struct dn_ifaddr *ifa)
  ASSERT_RTNL();

  /* Scan device list */
- read_lock(&dev_base_lock);
- for(dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   dn_db = dev->dn_ptr;
   if (dn_db == NULL)
    continue;
@@ -610,7 +610,7 @@ static void dn_fib_del_ifaddr(struct dn_ifaddr *ifa)
    }
   }
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  if (found_it == 0) {
   fib_magic(RTM_DELROUTE, RTN_LOCAL, ifa->ifa_local, 16, ifa);
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index b553cd4..9669e50 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -849,7 +849,7 @@ static __le16 dnet_select_source(const struct net_device *dev, __le16
daddr, int
  int best_match = 0;
  int ret;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  for(ifa = dn_db->ifa_list; ifa; ifa = ifa->ifa_next) {
   if (ifa->ifa_scope > scope)
```

```
    continue;
@@ -863,7 +863,7 @@ static __le16 dnet_select_source(const struct net_device *dev, __le16
daddr, int
    if (best_match == 0)
     saddr = ifa->ifa_local;
   }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

   return saddr;
 }
@@ -908,7 +908,7 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old

   /* If we have an output interface, verify its a DECnet device */
   if (oldflp->oif) {
-   dev_out = dev_get_by_index(oldflp->oif);
+   dev_out = dev_get_by_index(init_net(), oldflp->oif);
    err = -ENODEV;
    if (dev_out && dev_out->dn_ptr == NULL) {
     dev_put(dev_out);
@@ -928,8 +928,8 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old
     dev_put(dev_out);
     goto out;
    }
-   read_lock(&dev_base_lock);
-   for(dev_out = dev_base; dev_out; dev_out = dev_out->next) {
+   read_lock(&per_net(dev_base_lock, init_net()));
+   for(dev_out = per_net(dev_base, init_net()); dev_out; dev_out = dev_out->next) {
     if (!dev_out->dn_ptr)
      continue;
     if (!dn_dev_islocal(dev_out, oldflp->fld_src))
@@ -940,7 +940,7 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old
      continue;
     break;
    }
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));
    if (dev_out == NULL)
     goto out;
    dev_hold(dev_out);
@@ -1554,7 +1554,7 @@ int dn_cache_getroute(struct sk_buff *in_skb, struct nlmsghdr *nlh,
void *arg)

   if (fl.iif) {
    struct net_device *dev;
```

```
-  if ((dev = dev_get_by_index(fl.iif)) == NULL) {
+  if ((dev = dev_get_by_index(init_net(), fl.iif)) == NULL) {
     kfree_skb(skb);
     return -ENODEV;
   }
diff --git a/net/decnet/sysctl_net_decnet.c b/net/decnet/sysctl_net_decnet.c
index 37fff9a..778b8a5 100644
--- a/net/decnet/sysctl_net_decnet.c
+++ b/net/decnet/sysctl_net_decnet.c
@@ -259,7 +259,7 @@ static int dn_def_dev_strategy(ctl_table *table, int __user *name, int
nlen,

    devname[newlen] = 0;

-   dev = dev_get_by_name(devname);
+   dev = dev_get_by_name(init_net(), devname);
    if (dev == NULL)
     return -ENODEV;

@@ -299,7 +299,7 @@ static int dn_def_dev_handler(ctl_table *table, int write,
    devname[*lenp] = 0;
    strip_it(devname);

-   dev = dev_get_by_name(devname);
+   dev = dev_get_by_name(init_net(), devname);
    if (dev == NULL)
     return -ENODEV;

diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index cbf87f4..cd5336b 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -663,7 +663,7 @@ static int ec_dev_ioctl(struct socket *sock, unsigned int cmd, void __user
*arg)
   if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
    return -EFAULT;

- if ((dev = dev_get_by_name(ifr.ifr_name)) == NULL)
+ if ((dev = dev_get_by_name(init_net(), ifr.ifr_name)) == NULL)
    return -ENODEV;

   sec = (struct sockaddr_ec *)&ifr.ifr_addr;
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index 0d23fb2..39d2ac4 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -983,7 +983,7 @@ static int arp_req_set(struct arpreq *r, struct net_device * dev)
    if (mask && mask != htonl(0xFFFFFFFF))
```

```
    return -EINVAL;
   if (!dev && (r->arp_flags & ATF_COM)) {
-   dev = dev_getbyhwaddr(r->arp_ha.sa_family, r->arp_ha.sa_data);
+   dev = dev_getbyhwaddr(init_net(), r->arp_ha.sa_family, r->arp_ha.sa_data);
    if (!dev)
     return -ENODEV;
   }
@@ -1170,7 +1170,7 @@ int arp_ioctl(unsigned int cmd, void __user *arg)
  rtnl_lock();
  if (r.arp_dev[0]) {
   err = -ENODEV;
-  if ((dev = __dev_get_by_name(r.arp_dev)) == NULL)
+  if ((dev = __dev_get_by_name(init_net(), r.arp_dev)) == NULL)
    goto out;

   /* Mmmm... It is wrong... ARPHRD_NETROM==0 */
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 201442c..b0d12ec 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -419,11 +419,11 @@ struct in_device *inetdev_by_index(int ifindex)
 {
  struct net_device *dev;
  struct in_device *in_dev = NULL;
- read_lock(&dev_base_lock);
- dev = __dev_get_by_index(ifindex);
+ read_lock(&per_net(dev_base_lock, init_net()));
+ dev = __dev_get_by_index(init_net(), ifindex);
  if (dev)
   in_dev = in_dev_get(dev);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  return in_dev;
 }

@@ -504,7 +504,7 @@ static struct in_ifaddr *rtm_to_ifaddr(struct nlmsghdr *nlh)
  if (ifm->ifa_prefixlen > 32 || tb[IFA_LOCAL] == NULL)
   goto errout;

- dev = __dev_get_by_index(ifm->ifa_index);
+ dev = __dev_get_by_index(init_net(), ifm->ifa_index);
  if (dev == NULL) {
   err = -ENODEV;
   goto errout;
@@ -627,7 +627,7 @@ int devinet_ioctl(unsigned int cmd, void __user *arg)
   *colon = 0;

 #ifdef CONFIG_KMOD
```

```
- dev_load(ifr.ifr_name);
+ dev_load(init_net(), ifr.ifr_name);
 #endif

  switch(cmd) {
@@ -668,7 +668,7 @@ int devinet_ioctl(unsigned int cmd, void __user *arg)
  rtnl_lock();

  ret = -ENODEV;
- if ((dev = __dev_get_by_name(ifr.ifr_name)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifr.ifr_name)) == NULL)
   goto done;

  if (colon)
@@ -906,9 +906,9 @@ no_in_dev:
     in this case. It is importnat that lo is the first interface
     in dev_base list.
   */
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
    continue;

@@ -921,7 +921,7 @@ no_in_dev:
   } endfor_ifa(in_dev);
  }
 out_unlock_both:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  rcu_read_unlock();
 out:
  return addr;
@@ -985,9 +985,9 @@ __be32 inet_confirm_addr(const struct net_device *dev, __be32 dst,
__be32 local,
   return addr;
  }

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   if ((in_dev = __in_dev_get_rcu(dev))) {
    addr = confirm_addr_indev(in_dev, dst, local, scope);
    if (addr)
```

```
@@ -995,7 +995,7 @@ __be32 inet_confirm_addr(const struct net_device *dev, __be32 dst,
__be32 local,
   }
  }
  rcu_read_unlock();
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  return addr;
 }
@@ -1180,8 +1180,8 @@ static int inet_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
  int s_ip_idx, s_idx = cb->args[0];

  s_ip_idx = ip_idx = cb->args[1];
- read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
   if (idx < s_idx)
    continue;
   if (idx > s_idx)
@@ -1207,7 +1207,7 @@ static int inet_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
  }

 done:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  cb->args[0] = idx;
  cb->args[1] = ip_idx;

@@ -1258,8 +1258,8 @@ void inet_forward_change(void)
  ipv4_devconf.accept_redirects = !on;
  ipv4_devconf_dflt.forwarding = on;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   struct in_device *in_dev;
   rcu_read_lock();
   in_dev = __in_dev_get_rcu(dev);
@@ -1267,7 +1267,7 @@ void inet_forward_change(void)
    in_dev->cnf.forwarding = on;
   rcu_read_unlock();
  }
- read_unlock(&dev_base_lock);
```

```
+ read_unlock(&per_net(dev_base_lock, init_net()));

  rt_cache_flush(0);
 }
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d1859ff..449f42d 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -337,7 +337,7 @@ static int rtentry_to_fib_config(int cmd, struct rtentry *rt,
    colon = strchr(devname, ':');
    if (colon)
     *colon = 0;
-   dev = __dev_get_by_name(devname);
+   dev = __dev_get_by_name(init_net(), devname);
    if (!dev)
     return -ENODEV;
    cfg->fc_oif = dev->ifindex;
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index e63b8a9..76218e5 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -530,7 +530,7 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
      return -EINVAL;
     if (inet_addr_type(nh->nh_gw) != RTN_UNICAST)
      return -EINVAL;
-    if ((dev = __dev_get_by_index(nh->nh_oif)) == NULL)
+    if ((dev = __dev_get_by_index(init_net(), nh->nh_oif)) == NULL)
      return -ENODEV;
     if (!(dev->flags&IFF_UP))
      return -ENETDOWN;
@@ -807,7 +807,7 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
    if (nhs != 1 || nh->nh_gw)
     goto err_inval;
    nh->nh_scope = RT_SCOPE_NOWHERE;
-   nh->nh_dev = dev_get_by_index(fi->fib_nh->nh_oif);
+   nh->nh_dev = dev_get_by_index(init_net(), fi->fib_nh->nh_oif);
    err = -ENODEV;
    if (nh->nh_dev == NULL)
     goto failure;
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 92624cc..0455935 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -2262,7 +2262,7 @@ static inline struct ip_mc_list *igmp_mc_get_first(struct seq_file *seq)
  struct ip_mc_list *im = NULL;
  struct igmp_mc_iter_state *state = igmp_mc_seq_private(seq);

- for (state->dev = dev_base, state->in_dev = NULL;
```

```
+ for (state->dev = per_net(dev_base, init_net()), state->in_dev = NULL;
       state->dev;
       state->dev = state->dev->next) {
   struct in_device *in_dev;
@@ -2315,7 +2315,7 @@ static struct ip_mc_list *igmp_mc_get_idx(struct seq_file *seq, loff_t
pos)

 static void *igmp_mc_seq_start(struct seq_file *seq, loff_t *pos)
 {
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  return *pos ? igmp_mc_get_idx(seq, *pos - 1) : SEQ_START_TOKEN;
 }

@@ -2339,7 +2339,7 @@ static void igmp_mc_seq_stop(struct seq_file *seq, void *v)
    state->in_dev = NULL;
  }
  state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static int igmp_mc_seq_show(struct seq_file *seq, void *v)
@@ -2424,7 +2424,7 @@ static inline struct ip_sf_list *igmp_mcf_get_first(struct seq_file *seq)
  struct ip_mc_list *im = NULL;
  struct igmp_mcf_iter_state *state = igmp_mcf_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL, state->im = NULL;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL, state->im = NULL;
       state->dev;
       state->dev = state->dev->next) {
   struct in_device *idev;
@@ -2493,7 +2493,7 @@ static struct ip_sf_list *igmp_mcf_get_idx(struct seq_file *seq, loff_t
pos)

 static void *igmp_mcf_seq_start(struct seq_file *seq, loff_t *pos)
 {
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  return *pos ? igmp_mcf_get_idx(seq, *pos - 1) : SEQ_START_TOKEN;
 }

@@ -2521,7 +2521,7 @@ static void igmp_mcf_seq_stop(struct seq_file *seq, void *v)
    state->idev = NULL;
  }
  state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
```

```
 }

 static int igmp_mcf_seq_show(struct seq_file *seq, void *v)
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 8ce00d3..078eed3 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -292,7 +292,7 @@ static void ip_expire(unsigned long arg)
  if ((qp->last_in&FIRST_IN) && qp->fragments != NULL) {
   struct sk_buff *head = qp->fragments;
   /* Send an ICMP "Fragment Reassembly Timeout" message. */
-  if ((head->dev = dev_get_by_index(qp->iif)) != NULL) {
+  if ((head->dev = dev_get_by_index(init_net(), qp->iif)) != NULL) {
    icmp_send(head, ICMP_TIME_EXCEEDED, ICMP_EXC_FRAGTIME, 0);
    dev_put(head->dev);
   }
diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index 476cb60..a21688c 100644
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -266,7 +266,7 @@ static struct ip_tunnel * ipgre_tunnel_locate(struct ip_tunnel_parm
*parms, int
   int i;
   for (i=1; i<100; i++) {
    sprintf(name, "gre%d", i);
-   if (__dev_get_by_name(name) == NULL)
+   if (__dev_get_by_name(init_net(), name) == NULL)
     break;
   }
   if (i==100)
@@ -1196,7 +1196,7 @@ static int ipgre_tunnel_init(struct net_device *dev)
  }

  if (!tdev && tunnel->parms.link)
-  tdev = __dev_get_by_index(tunnel->parms.link);
+  tdev = __dev_get_by_index(init_net(), tunnel->parms.link);

  if (tdev) {
   hlen = tdev->hard_header_len;
diff --git a/net/ipv4/ip_sockglue.c b/net/ipv4/ip_sockglue.c
index 57d4bae..95094c5 100644
--- a/net/ipv4/ip_sockglue.c
+++ b/net/ipv4/ip_sockglue.c
@@ -597,7 +597,7 @@ static int do_ip_setsockopt(struct sock *sk, int level,
      dev_put(dev);
     }
    } else
-    dev = __dev_get_by_index(mreq.imr_ifindex);
```

```
+    dev = __dev_get_by_index(init_net(), mreq.imr_ifindex);


    err = -EADDRNOTAVAIL;
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index ee77938..2606b8c 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -195,7 +195,7 @@ static int __init ic_open_devs(void)
  if (dev_change_flags(lo, lo->flags | IFF_UP) < 0)
   printk(KERN_ERR "IP-Config: Failed to open %s\n", lo->name);

- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   if (dev == lo)
    continue;
   if (user_dev_name[0] ? !strcmp(dev->name, user_dev_name) :
diff --git a/net/ipv4/ipip.c b/net/ipv4/ipip.c
index 9d719d6..4e75691 100644
--- a/net/ipv4/ipip.c
+++ b/net/ipv4/ipip.c
@@ -232,7 +232,7 @@ static struct ip_tunnel * ipip_tunnel_locate(struct ip_tunnel_parm *parms,
int c
   int i;
   for (i=1; i<100; i++) {
    sprintf(name, "tunl%d", i);
-   if (__dev_get_by_name(name) == NULL)
+   if (__dev_get_by_name(init_net(), name) == NULL)
     break;
   }
   if (i==100)
@@ -827,7 +827,7 @@ static int ipip_tunnel_init(struct net_device *dev)
  }

  if (!tdev && tunnel->parms.link)
-  tdev = __dev_get_by_index(tunnel->parms.link);
+  tdev = __dev_get_by_index(init_net(), tunnel->parms.link);

  if (tdev) {
   dev->hard_header_len = tdev->hard_header_len + sizeof(struct iphdr);
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index 9afaa13..d2e7e55 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -125,7 +125,7 @@ struct net_device *ipmr_new_tunnel(struct vifctl *v)
 {
  struct net_device  *dev;
```

```
- dev = __dev_get_by_name("tunl0");
+ dev = __dev_get_by_name(init_net(), "tunl0");

  if (dev) {
   int err;
@@ -149,7 +149,7 @@ struct net_device *ipmr_new_tunnel(struct vifctl *v)

   dev = NULL;

-  if (err == 0 && (dev = __dev_get_by_name(p.name)) != NULL) {
+  if (err == 0 && (dev = __dev_get_by_name(init_net(), p.name)) != NULL) {
    dev->flags |= IFF_MULTICAST;

    in_dev = __in_dev_get_rtnl(dev);
diff --git a/net/ipv4/ipvs/ip_vs_sync.c b/net/ipv4/ipvs/ip_vs_sync.c
index 7ea2d98..fd6d1ca 100644
--- a/net/ipv4/ipvs/ip_vs_sync.c
+++ b/net/ipv4/ipvs/ip_vs_sync.c
@@ -382,7 +382,7 @@ static int set_mcast_if(struct sock *sk, char *ifname)
  struct net_device *dev;
  struct inet_sock *inet = inet_sk(sk);

- if ((dev = __dev_get_by_name(ifname)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifname)) == NULL)
   return -ENODEV;

  if (sk->sk_bound_dev_if && dev->ifindex != sk->sk_bound_dev_if)
@@ -407,7 +407,7 @@ static int set_sync_mesg_maxlen(int sync_state)
  int num;

  if (sync_state == IP_VS_STATE_MASTER) {
-  if ((dev = __dev_get_by_name(ip_vs_master_mcast_ifn)) == NULL)
+  if ((dev = __dev_get_by_name(init_net(), ip_vs_master_mcast_ifn)) == NULL)
    return -ENODEV;

   num = (dev->mtu - sizeof(struct iphdr) -
@@ -418,7 +418,7 @@ static int set_sync_mesg_maxlen(int sync_state)
   IP_VS_DBG(7, "setting the maximum length of sync sending "
     "message %d.\n", sync_send_mesg_maxlen);
  } else if (sync_state == IP_VS_STATE_BACKUP) {
-  if ((dev = __dev_get_by_name(ip_vs_backup_mcast_ifn)) == NULL)
+  if ((dev = __dev_get_by_name(init_net(), ip_vs_backup_mcast_ifn)) == NULL)
    return -ENODEV;

   sync_recv_mesg_maxlen = dev->mtu -
@@ -446,7 +446,7 @@ join_mcast_group(struct sock *sk, struct in_addr *addr, char *ifname)
  memset(&mreq, 0, sizeof(mreq));
  memcpy(&mreq.imr_multiaddr, addr, sizeof(struct in_addr));
```

```
- if ((dev = __dev_get_by_name(ifname)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifname)) == NULL)
   return -ENODEV;
  if (sk->sk_bound_dev_if && dev->ifindex != sk->sk_bound_dev_if)
   return -EINVAL;
@@ -467,7 +467,7 @@ static int bind_mcastif_addr(struct socket *sock, char *ifname)
  __be32 addr;
  struct sockaddr_in sin;

- if ((dev = __dev_get_by_name(ifname)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifname)) == NULL)
   return -ENODEV;

  addr = inet_select_addr(dev, 0, RT_SCOPE_UNIVERSE);
diff --git a/net/ipv4/netfilter/ipt_CLUSTERIP.c b/net/ipv4/netfilter/ipt_CLUSTERIP.c
index 779e2c6..02003ff 100644
--- a/net/ipv4/netfilter/ipt_CLUSTERIP.c
+++ b/net/ipv4/netfilter/ipt_CLUSTERIP.c
@@ -430,7 +430,7 @@ checkentry(const char *tablename,
     return 0;
    }

-   dev = dev_get_by_name(e->ip.iniface);
+   dev = dev_get_by_name(init_net(), e->ip.iniface);
    if (!dev) {
     printk(KERN_WARNING "CLUSTERIP: no such interface %s\n", e->ip.iniface);
     return 0;
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index d23a0d7..509bfb1 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2436,7 +2436,7 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)


  if (oldflp->oif) {
-  dev_out = dev_get_by_index(oldflp->oif);
+  dev_out = dev_get_by_index(init_net(), oldflp->oif);
   err = -ENODEV;
   if (dev_out == NULL)
    goto out;
@@ -2761,7 +2761,7 @@ int inet_rtm_getroute(struct sk_buff *in_skb, struct nlmsghdr* nlh, void
*arg)
  if (iif) {
   struct net_device *dev;

-  dev = __dev_get_by_index(iif);
```

```
+  dev = __dev_get_by_index(init_net(), iif);
   if (dev == NULL) {
    err = -ENODEV;
    goto errout_free;
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index c9fa27a..7afe698 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -477,8 +477,8 @@ static void addrconf_forward_change(void)
  struct net_device *dev;
  struct inet6_dev *idev;

- read_lock(&dev_base_lock);
- for (dev=dev_base; dev; dev=dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev=per_net(dev_base, init_net()); dev; dev=dev->next) {
   rcu_read_lock();
   idev = __in6_dev_get(dev);
   if (idev) {
@@ -489,7 +489,7 @@ static void addrconf_forward_change(void)
   }
   rcu_read_unlock();
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }
 #endif

@@ -919,10 +919,10 @@ int ipv6_dev_get_saddr(struct net_device *daddr_dev,

  memset(&hiscore, 0, sizeof(hiscore));

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  rcu_read_lock();

- for (dev = dev_base; dev; dev=dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev=dev->next) {
   struct inet6_dev *idev;
   struct inet6_ifaddr *ifa;

@@ -1151,7 +1151,7 @@ record_it:
   read_unlock_bh(&idev->lock);
  }
  rcu_read_unlock();
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
```

```
  if (!ifa_result)
   return -EADDRNOTAVAIL;
@@ -1853,7 +1853,7 @@ int addrconf_set_dstaddr(void __user *arg)
  if (copy_from_user(&ireq, arg, sizeof(struct in6_ifreq)))
   goto err_exit;

- dev = __dev_get_by_index(ireq.ifr6_ifindex);
+ dev = __dev_get_by_index(init_net(), ireq.ifr6_ifindex);

  err = -ENODEV;
  if (dev == NULL)
@@ -1884,7 +1884,7 @@ int addrconf_set_dstaddr(void __user *arg)

   if (err == 0) {
    err = -ENOBUFS;
-   if ((dev = __dev_get_by_name(p.name)) == NULL)
+   if ((dev = __dev_get_by_name(init_net(), p.name)) == NULL)
     goto err_exit;
    err = dev_open(dev);
   }
@@ -1913,7 +1913,7 @@ static int inet6_addr_add(int ifindex, struct in6_addr *pfx, int plen,
  if (!valid_lft || prefered_lft > valid_lft)
   return -EINVAL;

- if ((dev = __dev_get_by_index(ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), ifindex)) == NULL)
   return -ENODEV;

  if ((idev = addrconf_add_dev(dev)) == NULL)
@@ -1956,7 +1956,7 @@ static int inet6_addr_del(int ifindex, struct in6_addr *pfx, int plen)
  struct inet6_dev *idev;
  struct net_device *dev;

- if ((dev = __dev_get_by_index(ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), ifindex)) == NULL)
   return -ENODEV;

  if ((idev = __in6_dev_get(dev)) == NULL)
@@ -2051,7 +2051,7 @@ static void sit_add_v4_addrs(struct inet6_dev *idev)
   return;
  }

-       for (dev = dev_base; dev != NULL; dev = dev->next) {
+       for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
   struct in_device * in_dev = __in_dev_get_rtnl(dev);
   if (in_dev && (dev->flags & IFF_UP)) {
    struct in_ifaddr * ifa;
@@ -2198,12 +2198,12 @@ static void ip6_tnl_add_linklocal(struct inet6_dev *idev)
```

```
  /* first try to inherit the link-local address from the link device */
  if (idev->dev->iflink &&
-     (link_dev = __dev_get_by_index(idev->dev->iflink))) {
+     (link_dev = __dev_get_by_index(init_net(), idev->dev->iflink))) {
    if (!ipv6_inherit_linklocal(idev, link_dev))
     return;
  }
  /* then try to inherit it from any device */
- for (link_dev = dev_base; link_dev; link_dev = link_dev->next) {
+ for (link_dev = per_net(dev_base, init_net()); link_dev; link_dev = link_dev->next) {
    if (!ipv6_inherit_linklocal(idev, link_dev))
     return;
  }
@@ -3032,7 +3032,7 @@ inet6_rtm_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg)
   valid_lft = INFINITY_LIFE_TIME;
  }

- dev =  __dev_get_by_index(ifm->ifa_index);
+ dev =  __dev_get_by_index(init_net(), ifm->ifa_index);
  if (dev == NULL)
   return -ENODEV;

@@ -3208,9 +3208,9 @@ static int inet6_dump_addr(struct sk_buff *skb, struct netlink_callback
*cb,

  s_idx = cb->args[0];
  s_ip_idx = ip_idx = cb->args[1];
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));

- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
    if (idx < s_idx)
     continue;
    if (idx > s_idx)
@@ -3270,7 +3270,7 @@ done:
   read_unlock_bh(&idev->lock);
   in6_dev_put(idev);
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  cb->args[0] = idx;
  cb->args[1] = ip_idx;
  return skb->len;
@@ -3318,7 +3318,7 @@ static int inet6_rtm_getaddr(struct sk_buff *in_skb, struct nlmsghdr*
nlh,
```

```
  ifm = nlmsg_data(nlh);
  if (ifm->ifa_index)
-  dev = __dev_get_by_index(ifm->ifa_index);
+  dev = __dev_get_by_index(init_net(), ifm->ifa_index);

  if ((ifa = ipv6_get_ifaddr(addr, dev, 1)) == NULL) {
   err = -EADDRNOTAVAIL;
@@ -3477,8 +3477,8 @@ static int inet6_dump_ifinfo(struct sk_buff *skb, struct netlink_callback
*cb)
  struct net_device *dev;
  struct inet6_dev *idev;

- read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev=per_net(dev_base, init_net()), idx=0; dev; dev = dev->next, idx++) {
   if (idx < s_idx)
    continue;
   if ((idev = in6_dev_get(dev)) == NULL)
@@ -3489,7 +3489,7 @@ static int inet6_dump_ifinfo(struct sk_buff *skb, struct netlink_callback
*cb)
   if (err <= 0)
    break;
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  cb->args[0] = idx;

  return skb->len;
@@ -4116,7 +4116,7 @@ void __exit addrconf_cleanup(void)
   * clean dev list.
   */

- for (dev=dev_base; dev; dev=dev->next) {
+ for (dev=per_net(dev_base, init_net()); dev; dev=dev->next) {
   if ((idev = __in6_dev_get(dev)) == NULL)
    continue;
   addrconf_ifdown(dev, 1);
diff --git a/net/ipv6/af_inet6.c b/net/ipv6/af_inet6.c
index 00bd55a..84f0623 100644
--- a/net/ipv6/af_inet6.c
+++ b/net/ipv6/af_inet6.c
@@ -302,7 +302,7 @@ int inet6_bind(struct socket *sock, struct sockaddr *uaddr, int addr_len)
      err = -EINVAL;
      goto out;
     }
-    dev = dev_get_by_index(sk->sk_bound_dev_if);
```

```
+    dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
     if (!dev) {
      err = -ENODEV;
      goto out;
diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index c42bad9..906ef0f 100644
--- a/net/ipv6/anycast.c
+++ b/net/ipv6/anycast.c
@@ -113,10 +113,10 @@ int ipv6_sock_ac_join(struct sock *sk, int ifindex, struct in6_addr *addr)
   } else {
    /* router, no matching interface: just pick one */

-   dev = dev_get_by_flags(IFF_UP, IFF_UP|IFF_LOOPBACK);
+   dev = dev_get_by_flags(init_net(), IFF_UP, IFF_UP|IFF_LOOPBACK);
   }
  } else
-  dev = dev_get_by_index(ifindex);
+  dev = dev_get_by_index(init_net(), ifindex);

  if (dev == NULL) {
   err = -ENODEV;
@@ -197,7 +197,7 @@ int ipv6_sock_ac_drop(struct sock *sk, int ifindex, struct in6_addr *addr)

  write_unlock_bh(&ipv6_sk_ac_lock);

- dev = dev_get_by_index(pac->acl_ifindex);
+ dev = dev_get_by_index(init_net(), pac->acl_ifindex);
  if (dev) {
   ipv6_dev_ac_dec(dev, &pac->acl_addr);
   dev_put(dev);
@@ -225,7 +225,7 @@ void ipv6_sock_ac_close(struct sock *sk)
   if (pac->acl_ifindex != prev_index) {
    if (dev)
     dev_put(dev);
-   dev = dev_get_by_index(pac->acl_ifindex);
+   dev = dev_get_by_index(init_net(), pac->acl_ifindex);
    prev_index = pac->acl_ifindex;
   }
   if (dev)
@@ -427,11 +427,11 @@ int ipv6_chk_acast_addr(struct net_device *dev, struct in6_addr *addr)
 {
  if (dev)
   return ipv6_chk_acast_dev(dev, addr);
- read_lock(&dev_base_lock);
- for (dev=dev_base; dev; dev=dev->next)
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev=per_net(dev_base, init_net()); dev; dev=dev->next)
   if (ipv6_chk_acast_dev(dev, addr))
```

```
    break;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  return dev != 0;
 }

@@ -449,7 +449,7 @@ static inline struct ifacaddr6 *ac6_get_first(struct seq_file *seq)
  struct ifacaddr6 *im = NULL;
  struct ac6_iter_state *state = ac6_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL;
       state->dev;
       state->dev = state->dev->next) {
   struct inet6_dev *idev;
@@ -502,7 +502,7 @@ static struct ifacaddr6 *ac6_get_idx(struct seq_file *seq, loff_t pos)

 static void *ac6_seq_start(struct seq_file *seq, loff_t *pos)
 {
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  return ac6_get_idx(seq, *pos);
 }

@@ -521,7 +521,7 @@ static void ac6_seq_stop(struct seq_file *seq, void *v)
   read_unlock_bh(&state->idev->lock);
   in6_dev_put(state->idev);
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static int ac6_seq_show(struct seq_file *seq, void *v)
diff --git a/net/ipv6/datagram.c b/net/ipv6/datagram.c
index 5c94fea..c5dfb4e 100644
--- a/net/ipv6/datagram.c
+++ b/net/ipv6/datagram.c
@@ -536,7 +536,7 @@ int datagram_send_ctl(struct msghdr *msg, struct flowi *fl,
     if (!src_info->ipi6_ifindex)
      return -EINVAL;
     else {
-     dev = dev_get_by_index(src_info->ipi6_ifindex);
+     dev = dev_get_by_index(init_net(), src_info->ipi6_ifindex);
      if (!dev)
       return -ENODEV;
     }
diff --git a/net/ipv6/ip6_tunnel.c b/net/ipv6/ip6_tunnel.c
index 8d91834..9006cbf 100644
```

```
--- a/net/ipv6/ip6_tunnel.c
+++ b/net/ipv6/ip6_tunnel.c
@@ -231,7 +231,7 @@ static struct ip6_tnl *ip6_tnl_create(struct ip6_tnl_parm *p)
   int i;
   for (i = 1; i < IP6_TNL_MAX; i++) {
    sprintf(name, "ip6tnl%d", i);
-   if (__dev_get_by_name(name) == NULL)
+   if (__dev_get_by_name(init_net(), name) == NULL)
     break;
   }
   if (i == IP6_TNL_MAX)
@@ -505,7 +505,7 @@ static inline int ip6_tnl_rcv_ctl(struct ip6_tnl *t)
       struct net_device *ldev = NULL;

   if (p->link)
-   ldev = dev_get_by_index(p->link);
+   ldev = dev_get_by_index(init_net(), p->link);

   if ((ipv6_addr_is_multicast(&p->laddr) ||
        likely(ipv6_chk_addr(&p->laddr, ldev, 0))) &&
@@ -621,7 +621,7 @@ static inline int ip6_tnl_xmit_ctl(struct ip6_tnl *t)
   struct net_device *ldev = NULL;

   if (p->link)
-   ldev = dev_get_by_index(p->link);
+   ldev = dev_get_by_index(init_net(), p->link);

   if (unlikely(!ipv6_chk_addr(&p->laddr, ldev, 0)))
    printk(KERN_WARNING
diff --git a/net/ipv6/ipv6_sockglue.c b/net/ipv6/ipv6_sockglue.c
index 352690e..65d9b82 100644
--- a/net/ipv6/ipv6_sockglue.c
+++ b/net/ipv6/ipv6_sockglue.c
@@ -548,7 +548,7 @@ done:
   if (sk->sk_bound_dev_if && sk->sk_bound_dev_if != val)
    goto e_inval;

-  if (__dev_get_by_index(val) == NULL) {
+  if (__dev_get_by_index(init_net(), val) == NULL) {
    retv = -ENODEV;
    break;
   }
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index 2759571..da45f46 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -215,7 +215,7 @@ int ipv6_sock_mc_join(struct sock *sk, int ifindex, struct in6_addr *addr)
    dst_release(&rt->u.dst);
```

```
   }
  } else
-  dev = dev_get_by_index(ifindex);
+  dev = dev_get_by_index(init_net(), ifindex);

  if (dev == NULL) {
   sock_kfree_s(sk, mc_lst, sizeof(*mc_lst));
@@ -266,7 +266,7 @@ int ipv6_sock_mc_drop(struct sock *sk, int ifindex, struct in6_addr *addr)
    *lnk = mc_lst->next;
    write_unlock_bh(&ipv6_sk_mc_lock);

-   if ((dev = dev_get_by_index(mc_lst->ifindex)) != NULL) {
+   if ((dev = dev_get_by_index(init_net(), mc_lst->ifindex)) != NULL) {
     struct inet6_dev *idev = in6_dev_get(dev);

     (void) ip6_mc_leave_src(sk, mc_lst, idev);
@@ -301,7 +301,7 @@ static struct inet6_dev *ip6_mc_find_dev(struct in6_addr *group, int
ifindex)
    dst_release(&rt->u.dst);
   }
  } else
-  dev = dev_get_by_index(ifindex);
+  dev = dev_get_by_index(init_net(), ifindex);

  if (!dev)
   return NULL;
@@ -332,7 +332,7 @@ void ipv6_sock_mc_close(struct sock *sk)
   np->ipv6_mc_list = mc_lst->next;
   write_unlock_bh(&ipv6_sk_mc_lock);

-  dev = dev_get_by_index(mc_lst->ifindex);
+  dev = dev_get_by_index(init_net(), mc_lst->ifindex);
   if (dev) {
    struct inet6_dev *idev = in6_dev_get(dev);

@@ -2334,7 +2334,7 @@ static inline struct ifmcaddr6 *igmp6_mc_get_first(struct seq_file *seq)
  struct ifmcaddr6 *im = NULL;
  struct igmp6_mc_iter_state *state = igmp6_mc_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL;
       state->dev;
       state->dev = state->dev->next) {
   struct inet6_dev *idev;
@@ -2388,7 +2388,7 @@ static struct ifmcaddr6 *igmp6_mc_get_idx(struct seq_file *seq, loff_t
pos)

 static void *igmp6_mc_seq_start(struct seq_file *seq, loff_t *pos)
```

```
 {
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  return igmp6_mc_get_idx(seq, *pos);
 }

@@ -2409,7 +2409,7 @@ static void igmp6_mc_seq_stop(struct seq_file *seq, void *v)
   state->idev = NULL;
  }
  state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static int igmp6_mc_seq_show(struct seq_file *seq, void *v)
@@ -2478,7 +2478,7 @@ static inline struct ip6_sf_list *igmp6_mcf_get_first(struct seq_file *seq)
  struct ifmcaddr6 *im = NULL;
  struct igmp6_mcf_iter_state *state = igmp6_mcf_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL, state->im = NULL;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL, state->im = NULL;
       state->dev;
       state->dev = state->dev->next) {
   struct inet6_dev *idev;
@@ -2547,7 +2547,7 @@ static struct ip6_sf_list *igmp6_mcf_get_idx(struct seq_file *seq, loff_t
pos)

 static void *igmp6_mcf_seq_start(struct seq_file *seq, loff_t *pos)
 {
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  return *pos ? igmp6_mcf_get_idx(seq, *pos - 1) : SEQ_START_TOKEN;
 }

@@ -2575,7 +2575,7 @@ static void igmp6_mcf_seq_stop(struct seq_file *seq, void *v)
   state->idev = NULL;
  }
  state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 static int igmp6_mcf_seq_show(struct seq_file *seq, void *v)
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index 2e1825c..5a68e2d 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -256,7 +256,7 @@ static int rawv6_bind(struct sock *sk, struct sockaddr *uaddr, int
```

```
addr_len)
    if (!sk->sk_bound_dev_if)
     goto out;

-   dev = dev_get_by_index(sk->sk_bound_dev_if);
+   dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    if (!dev) {
     err = -ENODEV;
     goto out;
diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 6f9a904..0441380 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -301,7 +301,7 @@ static void ip6_frag_expire(unsigned long data)

  fq_kill(fq);

- dev = dev_get_by_index(fq->iif);
+ dev = dev_get_by_index(init_net(), fq->iif);
  if (!dev)
   goto out;

diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 6805c39..4519006 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -1045,7 +1045,7 @@ int ip6_route_add(struct fib6_config *cfg)
 #endif
  if (cfg->fc_ifindex) {
   err = -ENODEV;
-  dev = dev_get_by_index(cfg->fc_ifindex);
+  dev = dev_get_by_index(init_net(), cfg->fc_ifindex);
   if (!dev)
    goto out;
   idev = in6_dev_get(dev);
@@ -2168,7 +2168,7 @@ int inet6_rtm_getroute(struct sk_buff *in_skb, struct nlmsghdr* nlh,
void *arg)

  if (iif) {
   struct net_device *dev;
-  dev = __dev_get_by_index(iif);
+  dev = __dev_get_by_index(init_net(), iif);
   if (!dev) {
    err = -ENODEV;
    goto errout;
diff --git a/net/ipv6/sit.c b/net/ipv6/sit.c
index 77b7b09..8f97692 100644
--- a/net/ipv6/sit.c
```

```
+++ b/net/ipv6/sit.c
@@ -173,7 +173,7 @@ static struct ip_tunnel * ipip6_tunnel_locate(struct ip_tunnel_parm
*parms, int
   int i;
   for (i=1; i<100; i++) {
    sprintf(name, "sit%d", i);
-   if (__dev_get_by_name(name) == NULL)
+   if (__dev_get_by_name(init_net(), name) == NULL)
     break;
   }
   if (i==100)
@@ -759,7 +759,7 @@ static int ipip6_tunnel_init(struct net_device *dev)
  }

  if (!tdev && tunnel->parms.link)
-  tdev = __dev_get_by_index(tunnel->parms.link);
+  tdev = __dev_get_by_index(init_net(), tunnel->parms.link);

  if (tdev) {
   dev->hard_header_len = tdev->hard_header_len + sizeof(struct iphdr);
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index f2674fe..0e63fd2 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -987,7 +987,7 @@ static int ipxitf_create(struct ipx_interface_definition *idef)
  if (intrfc)
   ipxitf_put(intrfc);

- dev = dev_get_by_name(idef->ipx_device);
+ dev = dev_get_by_name(init_net(), idef->ipx_device);
  rc = -ENODEV;
  if (!dev)
   goto out;
@@ -1095,7 +1095,7 @@ static int ipxitf_delete(struct ipx_interface_definition *idef)
  if (!dlink_type)
   goto out;

- dev = __dev_get_by_name(idef->ipx_device);
+ dev = __dev_get_by_name(init_net(), idef->ipx_device);
  rc = -ENODEV;
  if (!dev)
   goto out;
@@ -1190,7 +1190,7 @@ static int ipxitf_ioctl(unsigned int cmd, void __user *arg)
   if (copy_from_user(&ifr, arg, sizeof(ifr)))
    break;
   sipx = (struct sockaddr_ipx *)&ifr.ifr_addr;
-  dev  = __dev_get_by_name(ifr.ifr_name);
+  dev  = __dev_get_by_name(init_net(), ifr.ifr_name);
```

```
   rc   = -ENODEV;
   if (!dev)
    break;
diff --git a/net/llc/af_llc.c b/net/llc/af_llc.c
index 6bc0fff..ac380ac 100644
--- a/net/llc/af_llc.c
+++ b/net/llc/af_llc.c
@@ -252,7 +252,7 @@ static int llc_ui_autobind(struct socket *sock, struct sockaddr_llc *addr)
  if (!sock_flag(sk, SOCK_ZAPPED))
   goto out;
  rc = -ENODEV;
- llc->dev = dev_getfirstbyhwtype(addr->sllc_arphrd);
+ llc->dev = dev_getfirstbyhwtype(init_net(), addr->sllc_arphrd);
  if (!llc->dev)
   goto out;
  rc = -EUSERS;
@@ -303,7 +303,7 @@ static int llc_ui_bind(struct socket *sock, struct sockaddr *uaddr, int
addrlen)
   goto out;
  rc = -ENODEV;
  rtnl_lock();
- llc->dev = dev_getbyhwaddr(addr->sllc_arphrd, addr->sllc_mac);
+ llc->dev = dev_getbyhwaddr(init_net(), addr->sllc_arphrd, addr->sllc_mac);
  rtnl_unlock();
  if (!llc->dev)
   goto out;
diff --git a/net/llc/llc_core.c b/net/llc/llc_core.c
index f438c38..24a5739 100644
--- a/net/llc/llc_core.c
+++ b/net/llc/llc_core.c
@@ -19,6 +19,7 @@
 #include <linux/slab.h>
 #include <linux/string.h>
 #include <linux/init.h>
+#include <net/net_namespace.h>
 #include <net/llc.h>

 LIST_HEAD(llc_sap_list);
@@ -159,8 +160,8 @@ static struct packet_type llc_tr_packet_type = {

 static int __init llc_init(void)
 {
- if (dev_base->next)
-  memcpy(llc_station_mac_sa, dev_base->next->dev_addr, ETH_ALEN);
+ if (per_net(dev_base, init_net())->next)
+  memcpy(llc_station_mac_sa, per_net(dev_base, init_net())->next->dev_addr, ETH_ALEN);
  else
   memset(llc_station_mac_sa, 0, ETH_ALEN);
```

```
  dev_add_pack(&llc_packet_type);
diff --git a/net/netrom/nr_route.c b/net/netrom/nr_route.c
index 8f88964..5bfd12e 100644
--- a/net/netrom/nr_route.c
+++ b/net/netrom/nr_route.c
@@ -581,7 +581,7 @@ static struct net_device *nr_ax25_dev_get(char *devname)
 {
  struct net_device *dev;

- if ((dev = dev_get_by_name(devname)) == NULL)
+ if ((dev = dev_get_by_name(init_net(), devname)) == NULL)
    return NULL;

  if ((dev->flags & IFF_UP) && dev->type == ARPHRD_AX25)
@@ -598,15 +598,15 @@ struct net_device *nr_dev_first(void)
 {
  struct net_device *dev, *first = NULL;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
    if ((dev->flags & IFF_UP) && dev->type == ARPHRD_NETROM)
     if (first == NULL || strncmp(dev->name, first->name, 3) < 0)
      first = dev;
  }
  if (first)
   dev_hold(first);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  return first;
 }
@@ -618,15 +618,15 @@ struct net_device *nr_dev_get(ax25_address *addr)
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
    if ((dev->flags & IFF_UP) && dev->type == ARPHRD_NETROM && ax25cmp(addr,
(ax25_address *)dev->dev_addr) == 0) {
     dev_hold(dev);
     goto out;
    }
  }
 out:
```

```
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  return dev;
 }

diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index 6e3b947..4ac9f9f 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -359,7 +359,7 @@ static int packet_sendmsg_spkt(struct kiocb *iocb, struct socket *sock,
   */

  saddr->spkt_device[13] = 0;
- dev = dev_get_by_name(saddr->spkt_device);
+ dev = dev_get_by_name(init_net(), saddr->spkt_device);
  err = -ENODEV;
  if (dev == NULL)
   goto out_unlock;
@@ -744,7 +744,7 @@ static int packet_sendmsg(struct kiocb *iocb, struct socket *sock,
  }


- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
  err = -ENXIO;
  if (dev == NULL)
   goto out_unlock;
@@ -943,7 +943,7 @@ static int packet_bind_spkt(struct socket *sock, struct sockaddr *uaddr,
int add
   return -EINVAL;
  strlcpy(name,uaddr->sa_data,sizeof(name));

- dev = dev_get_by_name(name);
+ dev = dev_get_by_name(init_net(), name);
  if (dev) {
   err = packet_do_bind(sk, dev, pkt_sk(sk)->num);
   dev_put(dev);
@@ -971,7 +971,7 @@ static int packet_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len

  if (sll->sll_ifindex) {
   err = -ENODEV;
-  dev = dev_get_by_index(sll->sll_ifindex);
+  dev = dev_get_by_index(init_net(), sll->sll_ifindex);
   if (dev == NULL)
    goto out;
  }
@@ -1158,7 +1158,7 @@ static int packet_getname_spkt(struct socket *sock, struct sockaddr
```

```
*uaddr,
   return -EOPNOTSUPP;

  uaddr->sa_family = AF_PACKET;
- dev = dev_get_by_index(pkt_sk(sk)->ifindex);
+ dev = dev_get_by_index(init_net(), pkt_sk(sk)->ifindex);
  if (dev) {
   strlcpy(uaddr->sa_data, dev->name, 15);
   dev_put(dev);
@@ -1184,7 +1184,7 @@ static int packet_getname(struct socket *sock, struct sockaddr *uaddr,
  sll->sll_family = AF_PACKET;
  sll->sll_ifindex = po->ifindex;
  sll->sll_protocol = po->num;
- dev = dev_get_by_index(po->ifindex);
+ dev = dev_get_by_index(init_net(), po->ifindex);
  if (dev) {
   sll->sll_hatype = dev->type;
   sll->sll_halen = dev->addr_len;
@@ -1237,7 +1237,7 @@ static int packet_mc_add(struct sock *sk, struct packet_mreq_max
*mreq)
  rtnl_lock();

  err = -ENODEV;
- dev = __dev_get_by_index(mreq->mr_ifindex);
+ dev = __dev_get_by_index(init_net(), mreq->mr_ifindex);
  if (!dev)
   goto done;

@@ -1291,7 +1291,7 @@ static int packet_mc_drop(struct sock *sk, struct packet_mreq_max
*mreq)
    if (--ml->count == 0) {
     struct net_device *dev;
     *mlp = ml->next;
-    dev = dev_get_by_index(ml->ifindex);
+    dev = dev_get_by_index(init_net(), ml->ifindex);
     if (dev) {
      packet_dev_mc(dev, ml, -1);
      dev_put(dev);
@@ -1319,7 +1319,7 @@ static void packet_flush_mclist(struct sock *sk)
   struct net_device *dev;

   po->mclist = ml->next;
-  if ((dev = dev_get_by_index(ml->ifindex)) != NULL) {
+  if ((dev = dev_get_by_index(init_net(), ml->ifindex)) != NULL) {
    packet_dev_mc(dev, ml, -1);
    dev_put(dev);
   }
diff --git a/net/rose/rose_route.c b/net/rose/rose_route.c
```

```
index 8028c0d..92343be 100644
--- a/net/rose/rose_route.c
+++ b/net/rose/rose_route.c
@@ -579,7 +579,7 @@ static struct net_device *rose_ax25_dev_get(char *devname)
 {
  struct net_device *dev;

- if ((dev = dev_get_by_name(devname)) == NULL)
+ if ((dev = dev_get_by_name(init_net(), devname)) == NULL)
   return NULL;

  if ((dev->flags & IFF_UP) && dev->type == ARPHRD_AX25)
@@ -596,13 +596,13 @@ struct net_device *rose_dev_first(void)
 {
  struct net_device *dev, *first = NULL;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE)
    if (first == NULL || strncmp(dev->name, first->name, 3) < 0)
     first = dev;
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  return first;
 }
@@ -614,15 +614,15 @@ struct net_device *rose_dev_get(rose_address *addr)
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE && rosecmp(addr, (rose_address
*)dev->dev_addr) == 0) {
    dev_hold(dev);
    goto out;
   }
  }
 out:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  return dev;
 }
```

```
@@ -630,13 +630,13 @@ static int rose_dev_exists(rose_address *addr)
 {
  struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE && rosecmp(addr, (rose_address
*)dev->dev_addr) == 0)
    goto out;
  }
 out:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  return dev != NULL;
 }

diff --git a/net/sched/act_mirred.c b/net/sched/act_mirred.c
index 4838972..2c3e4af 100644
--- a/net/sched/act_mirred.c
+++ b/net/sched/act_mirred.c
@@ -85,7 +85,7 @@ static int tcf_mirred_init(struct rtattr *rta, struct rtattr *est,
  parm = RTA_DATA(tb[TCA_MIRRED_PARMS-1]);

  if (parm->ifindex) {
-  dev = __dev_get_by_index(parm->ifindex);
+  dev = __dev_get_by_index(init_net(), parm->ifindex);
   if (dev == NULL)
    return -ENODEV;
   switch (dev->type) {
diff --git a/net/sched/cls_api.c b/net/sched/cls_api.c
index edb8fc9..19935f9 100644
--- a/net/sched/cls_api.c
+++ b/net/sched/cls_api.c
@@ -164,7 +164,7 @@ replay:
  /* Find head of filter chain. */

  /* Find link */
- if ((dev = __dev_get_by_index(t->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), t->tcm_ifindex)) == NULL)
   return -ENODEV;

  /* Find qdisc */
@@ -397,7 +397,7 @@ static int tc_dump_tfilter(struct sk_buff *skb, struct netlink_callback *cb)

  if (cb->nlh->nlmsg_len < NLMSG_LENGTH(sizeof(*tcm)))
```

```
   return skb->len;
- if ((dev = dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return skb->len;

  read_lock(&qdisc_tree_lock);
diff --git a/net/sched/em_meta.c b/net/sched/em_meta.c
index 45d47d3..5df7cdf 100644
--- a/net/sched/em_meta.c
+++ b/net/sched/em_meta.c
@@ -291,7 +291,7 @@ META_COLLECTOR(var_sk_bound_if)
   } else  {
   struct net_device *dev;

-  dev = dev_get_by_index(skb->sk->sk_bound_dev_if);
+  dev = dev_get_by_index(init_net(), skb->sk->sk_bound_dev_if);
   *err = var_dev(dev, dst);
   if (dev)
    dev_put(dev);
diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index da7e1eb..912e8e1 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -586,7 +586,7 @@ static int tc_get_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
  struct Qdisc *p = NULL;
  int err;

- if ((dev = __dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return -ENODEV;

  if (clid) {
@@ -653,7 +653,7 @@ replay:
  clid = tcm->tcm_parent;
  q = p = NULL;

- if ((dev = __dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return -ENODEV;

  if (clid) {
@@ -858,8 +858,8 @@ static int tc_dump_qdisc(struct sk_buff *skb, struct netlink_callback *cb)

  s_idx = cb->args[0];
  s_q_idx = q_idx = cb->args[1];
- read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
```

```
+ for (dev=per_net(dev_base, init_net()), idx=0; dev; dev = dev->next, idx++) {
   if (idx < s_idx)
    continue;
   if (idx > s_idx)
@@ -882,7 +882,7 @@ static int tc_dump_qdisc(struct sk_buff *skb, struct netlink_callback *cb)
  }

 done:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  cb->args[0] = idx;
  cb->args[1] = q_idx;
@@ -912,7 +912,7 @@ static int tc_ctl_tclass(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
  u32 qid = TC_H_MAJ(clid);
  int err;

- if ((dev = __dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return -ENODEV;

  /*
@@ -1095,7 +1095,7 @@ static int tc_dump_tclass(struct sk_buff *skb, struct netlink_callback
*cb)

  if (cb->nlh->nlmsg_len < NLMSG_LENGTH(sizeof(*tcm)))
   return 0;
- if ((dev = dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return 0;

  s_t = cb->args[0];
diff --git a/net/sctp/ipv6.c b/net/sctp/ipv6.c
index 0217546..10b748c 100644
--- a/net/sctp/ipv6.c
+++ b/net/sctp/ipv6.c
@@ -844,7 +844,7 @@ static int sctp_inet6_bind_verify(struct sctp_sock *opt, union sctp_addr
*addr)
   if (type & IPV6_ADDR_LINKLOCAL) {
    if (!addr->v6.sin6_scope_id)
     return 0;
-   dev = dev_get_by_index(addr->v6.sin6_scope_id);
+   dev = dev_get_by_index(init_net(), addr->v6.sin6_scope_id);
    if (!dev)
     return 0;
    dev_put(dev);
@@ -871,7 +871,7 @@ static int sctp_inet6_send_verify(struct sctp_sock *opt, union sctp_addr
*addr)
```

```
   if (type & IPV6_ADDR_LINKLOCAL) {
    if (!addr->v6.sin6_scope_id)
     return 0;
-   dev = dev_get_by_index(addr->v6.sin6_scope_id);
+   dev = dev_get_by_index(init_net(), addr->v6.sin6_scope_id);
    if (!dev)
     return 0;
    dev_put(dev);
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 9461a10..05e2bb5 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -170,14 +170,14 @@ static void sctp_get_local_addr_list(void)
  struct list_head *pos;
  struct sctp_af *af;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
   __list_for_each(pos, &sctp_address_families) {
    af = list_entry(pos, struct sctp_af, list);
    af->copy_addrlist(&sctp_local_addr_list, dev);
   }
  }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
 }

 /* Free the existing local addresses.  */
diff --git a/net/socket.c b/net/socket.c
index 0d0c92b..7371654 100644
--- a/net/socket.c
+++ b/net/socket.c
@@ -772,9 +772,9 @@ static ssize_t sock_aio_write(struct kiocb *iocb, const struct iovec *iov,
  */

 static DEFINE_MUTEX(br_ioctl_mutex);
-static int (*br_ioctl_hook) (unsigned int cmd, void __user *arg) = NULL;
+static int (*br_ioctl_hook) (net_t, unsigned int cmd, void __user *arg) = NULL;

-void brioctl_set(int (*hook) (unsigned int, void __user *))
+void brioctl_set(int (*hook) (net_t, unsigned int, void __user *))
 {
  mutex_lock(&br_ioctl_mutex);
  br_ioctl_hook = hook;
@@ -784,9 +784,9 @@ void brioctl_set(int (*hook) (unsigned int, void __user *))
 EXPORT_SYMBOL(brioctl_set);
```

```
 static DEFINE_MUTEX(vlan_ioctl_mutex);
-static int (*vlan_ioctl_hook) (void __user *arg);
+static int (*vlan_ioctl_hook) (net_t, void __user *arg);

-void vlan_ioctl_set(int (*hook) (void __user *))
+void vlan_ioctl_set(int (*hook) (net_t, void __user *))
 {
  mutex_lock(&vlan_ioctl_mutex);
  vlan_ioctl_hook = hook;
@@ -815,16 +815,20 @@ EXPORT_SYMBOL(dlci_ioctl_set);
 static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)
 {
  struct socket *sock;
+ struct sock *sk;
  void __user *argp = (void __user *)arg;
  int pid, err;
+ net_t net;

  sock = file->private_data;
+ sk = sock->sk;
+ net = sk->sk_net;
  if (cmd >= SIOCDEVPRIVATE && cmd <= (SIOCDEVPRIVATE + 15)) {
-  err = dev_ioctl(cmd, argp);
+  err = dev_ioctl(net, cmd, argp);
  } else
 #ifdef CONFIG_WIRELESS_EXT
  if (cmd >= SIOCIWFIRST && cmd <= SIOCIWLAST) {
-  err = dev_ioctl(cmd, argp);
+  err = dev_ioctl(net, cmd, argp);
  } else
 #endif    /* CONFIG_WIRELESS_EXT */
   switch (cmd) {
@@ -850,7 +854,7 @@ static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)

    mutex_lock(&br_ioctl_mutex);
    if (br_ioctl_hook)
-    err = br_ioctl_hook(cmd, argp);
+    err = br_ioctl_hook(net, cmd, argp);
    mutex_unlock(&br_ioctl_mutex);
    break;
   case SIOCGIFVLAN:
@@ -861,7 +865,7 @@ static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)

    mutex_lock(&vlan_ioctl_mutex);
    if (vlan_ioctl_hook)
-    err = vlan_ioctl_hook(argp);
+    err = vlan_ioctl_hook(net, argp);
```

```
    mutex_unlock(&vlan_ioctl_mutex);
    break;
   case SIOCADDDLCI:
@@ -884,7 +888,7 @@ static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)
     * to the NIC driver.
     */
    if (err == -ENOIOCTLCMD)
-    err = dev_ioctl(cmd, argp);
+    err = dev_ioctl(net, cmd, argp);
    break;
   }
  return err;
diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index c6f64de..ba207ba 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -127,7 +127,7 @@ static int recv_msg(struct sk_buff *buf,

 static int enable_bearer(struct tipc_bearer *tb_ptr)
 {
- struct net_device *dev = dev_base;
+ struct net_device *dev = per_net(dev_base, init_net());
  struct eth_bearer *eb_ptr = &eth_bearers[0];
  struct eth_bearer *stop = &eth_bearers[MAX_ETH_BEARERS];
  char *driver_name = strchr((const char *)tb_ptr->name, ':') + 1;
diff --git a/net/wanrouter/af_wanpipe.c b/net/wanrouter/af_wanpipe.c
index f9b896c..397e876 100644
--- a/net/wanrouter/af_wanpipe.c
+++ b/net/wanrouter/af_wanpipe.c
@@ -586,7 +586,7 @@ static int wanpipe_sendmsg(struct kiocb *iocb, struct socket *sock,
   addr = saddr->sll_addr;
  }

- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
  if (dev == NULL){
   printk(KERN_INFO "wansock: Send failed, dev index: %i\n",ifindex);
   return -ENXIO;
@@ -769,7 +769,7 @@ static int execute_command(struct sock *sk,  unsigned char cmd,
unsigned int fla
  int err=0;
  DECLARE_WAITQUEUE(wait, current);

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
  if (dev == NULL){
   printk(KERN_INFO "wansock: Exec failed no dev %i\n",
    sk->sk_bound_dev_if);
```

```
@@ -878,7 +878,7 @@ static void wanpipe_unlink_driver (struct sock *sk)
  sk->sk_state = WANSOCK_DISCONNECTED;
  wp_sk(sk)->dev = NULL;

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
  if (!dev){
   printk(KERN_INFO "wansock: No dev on release\n");
   return;
@@ -974,7 +974,7 @@ static int wanpipe_release(struct socket *sock)

  if (wp->num == htons(X25_PROT) &&
      sk->sk_state != WANSOCK_DISCONNECTED && sock_flag(sk, SOCK_ZAPPED)) {
-  struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+  struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
   wanpipe_common_t *chan;
   if (dev){
    chan=dev->priv;
@@ -1153,7 +1153,7 @@ static void wanpipe_kill_sock_timer (unsigned long data)

  if (wp_sk(sk)->num == htons(X25_PROT) &&
      sk->sk_state != WANSOCK_DISCONNECTED) {
-  struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+  struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
   wanpipe_common_t *chan;
   if (dev){
    chan=dev->priv;
@@ -1396,7 +1396,7 @@ static int wanpipe_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_le
                * This is used by PVC mostly
                */
   strlcpy(name,sll->sll_device,sizeof(name));
-  dev = dev_get_by_name(name);
+  dev = dev_get_by_name(init_net(), name);
   if (dev == NULL){
    printk(KERN_INFO "wansock: Failed to get Dev from name: %s,\n",
      name);
@@ -1641,7 +1641,7 @@ static void wanpipe_wakeup_driver(struct sock *sk)
  struct net_device *dev = NULL;
  wanpipe_common_t *chan=NULL;

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
  if (!dev)
   return;

@@ -1680,7 +1680,7 @@ static int wanpipe_getname(struct socket *sock, struct sockaddr
*uaddr,
```

```
  sll->sll_family = AF_WANPIPE;
  sll->sll_ifindex = sk->sk_bound_dev_if;
  sll->sll_protocol = wp_sk(sk)->num;
- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
  if (dev) {
   sll->sll_hatype = dev->type;
   sll->sll_halen = dev->addr_len;
@@ -1898,7 +1898,7 @@ static int wanpipe_debug (struct sock *origsk, void *arg)
    return err;

   if (sk->sk_bound_dev_if) {
-   dev = dev_get_by_index(sk->sk_bound_dev_if);
+   dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    if (!dev)
     continue;

@@ -2009,7 +2009,7 @@ static int set_ioctl_cmd (struct sock *sk, void *arg)

  if (!wp_sk(sk)->mbox) {
   void *mbox_ptr;
-  struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+  struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
   if (!dev)
    return -ENODEV;

@@ -2352,7 +2352,7 @@ static int wanpipe_exec_cmd(struct sock *sk, int cmd, unsigned int
flags)

 static int check_driver_busy (struct sock *sk)
 {
- struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+ struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
  wanpipe_common_t *chan;

  if (!dev)
@@ -2507,7 +2507,7 @@ static int wanpipe_connect(struct socket *sock, struct sockaddr *uaddr,
int addr
  if (addr->sll_family != AF_WANPIPE)
   return -EINVAL;

- if ((dev = dev_get_by_index(sk->sk_bound_dev_if)) == NULL)
+ if ((dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if)) == NULL)
   return -ENETUNREACH;

  dev_put(dev);
diff --git a/net/x25/x25_route.c b/net/x25/x25_route.c
index 2a3fe98..091b96e 100644
```

```
--- a/net/x25/x25_route.c
+++ b/net/x25/x25_route.c
@@ -126,7 +126,7 @@ void x25_route_device_down(struct net_device *dev)
  */
 struct net_device *x25_dev_get(char *devname)
 {
- struct net_device *dev = dev_get_by_name(devname);
+ struct net_device *dev = dev_get_by_name(init_net(), devname);

  if (dev &&
      (!(dev->flags & IFF_UP) || (dev->type != ARPHRD_X25
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 17/31] net: Factor out __dev_alloc_name from dev_alloc_name

Posted by ebiederm on Thu, 25 Jan 2007 19:00:19 GMT

View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

When forcibly changing the network namespace of a device
I need something that can generate a name for the device
in the new namespace without overwriting the old name.

__dev_alloc_name provides me that functionality.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 net/core/dev.c |   44 ++++++++++++++++++++++++++++++++++----------
 1 files changed, 33 insertions(+), 11 deletions(-)

diff --git a/net/core/dev.c b/net/core/dev.c
index 32fe905..fc0d2af 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -655,9 +655,10 @@ int dev_valid_name(const char *name)
 }

 /**
- * dev_alloc_name - allocate a name for a device
- * @dev: device
```

```
+ * __dev_alloc_name - allocate a name for a device
+ * @net: network namespace to allocate the device name in
  * @name: name format string
+ * @buf:  scratch buffer and result name string
  *
  * Passed a format string - eg "lt%d" it will try and find a suitable
  * id. It scans list of devices to build up a free map, then chooses
@@ -668,18 +669,13 @@ int dev_valid_name(const char *name)
  * Returns the number of the unit assigned or a negative errno code.
  */

-int dev_alloc_name(struct net_device *dev, const char *name)
+static int __dev_alloc_name(net_t net, const char *name, char buf[IFNAMSIZ])
 {
  int i = 0;
- char buf[IFNAMSIZ];
  const char *p;
  const int max_netdevices = 8*PAGE_SIZE;
  long *inuse;
  struct net_device *d;
- net_t net;
-
- BUG_ON(null_net(dev->nd_net));
- net = dev->nd_net;

  p = strnchr(name, IFNAMSIZ-1, '%');
  if (p) {
@@ -713,10 +709,8 @@ int dev_alloc_name(struct net_device *dev, const char *name)
  }

  snprintf(buf, sizeof(buf), name, i);
- if (!__dev_get_by_name(net, buf)) {
-  strlcpy(dev->name, buf, IFNAMSIZ);
+ if (!__dev_get_by_name(net, buf))
   return i;
- }

  /* It is possible to run out of possible slots
   * when the name is long and there isn't enough space left
@@ -725,6 +719,34 @@ int dev_alloc_name(struct net_device *dev, const char *name)
  return -ENFILE;
 }

+/**
+ * dev_alloc_name - allocate a name for a device
+ * @dev: device
+ * @name: name format string
+ *
```

```
+ * Passed a format string - eg "lt%d" it will try and find a suitable
+ * id. It scans list of devices to build up a free map, then chooses
+ * the first empty slot. The caller must hold the dev_base or rtnl lock
+ * while allocating the name and adding the device in order to avoid
+ * duplicates.
+ * Limited to bits_per_byte * page size devices (ie 32K on most platforms).
+ * Returns the number of the unit assigned or a negative errno code.
+ */
+
+int dev_alloc_name(struct net_device *dev, const char *name)
+{
+ char buf[IFNAMSIZ];
+ net_t net;
+ int ret;
+
+ BUG_ON(null_net(dev->nd_net));
+ net = dev->nd_net;
+ ret = __dev_alloc_name(net, name, buf);
+ if (ret >= 0)
+  strlcpy(dev->name, buf, IFNAMSIZ);
+ return ret;
+}
+

 /**
  * dev_change_name - change name of a device
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 18/31] net: Implment network device movement between namespaces
Posted by ebiederm on Thu, 25 Jan 2007 19:00:20 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch introduces NETIF_F_NETNS_LOCAL a flag to indicate
a network device is local to a single network namespace and
should never be moved. Useful for pseudo devices that we
need an instance in each network namespace (like the loopback
device) and for any device we find that cannot handle multiple
network namespaces so we may trap them in the initial network

namespace.

This patch introduces the function dev_change_net_namespace
a function used to move a network device from one network
namespace to another. To the network device nothing
special appears to happen, to the components of the network
stack it appears as if the network device was unregistered
in the network namespace it is in, and a new device
was registered in the network namespace the device
was moved to.

This patch sets up a namespace device destructor that
upon the exit of a network namespace moves all of the
movable network devices to the initial network namespace
so they are not lost.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/net/loopback.c | 3 +-
 include/linux/netdevice.h | 3 +
 net/core/dev.c | 222 ++++++++++++++++++++++++++++++++++++++-----
 3 files changed, 201 insertions(+), 27 deletions(-)

diff --git a/drivers/net/loopback.c b/drivers/net/loopback.c
index e9abf3f..7d15de0 100644
--- a/drivers/net/loopback.c
+++ b/drivers/net/loopback.c
@@ -225,7 +225,8 @@ DEFINE_PER_NET(struct net_device, loopback_dev) = {
       | NETIF_F_TSO
 #endif
       | NETIF_F_NO_CSUM | NETIF_F_HIGHDMA
-      | NETIF_F_LLTX,
+ | NETIF_F_LLTX
+ | NETIF_F_NETNS_LOCAL,
   .ethtool_ops = &loopback_ethtool_ops,
 };

diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 0b4a4dc..3fcaf60 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -324,6 +324,7 @@ struct net_device
 #define NETIF_F_VLAN_CHALLENGED 1024 /* Device cannot handle VLAN packets */
 #define NETIF_F_GSO 2048 /* Enable software GSO. */
 #define NETIF_F_LLTX 4096 /* LockLess TX */
+#define NETIF_F_NETNS_LOCAL 8192 /* Does not change network namespaces */

   /* Segmentation offload features */

```
 #define NETIF_F_GSO_SHIFT 16
@@ -710,6 +711,8 @@ extern int  dev_ethtool(net_t net, struct ifreq *);
 extern unsigned  dev_get_flags(const struct net_device *);
 extern int  dev_change_flags(struct net_device *, unsigned);
 extern int  dev_change_name(struct net_device *, char *);
+extern int  dev_change_net_namespace(struct net_device *, net_t,
+       const char *);
 extern int  dev_set_mtu(struct net_device *, int);
 extern int  dev_set_mac_address(struct net_device *,
         struct sockaddr *);
diff --git a/net/core/dev.c b/net/core/dev.c
index fc0d2af..52994e4 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -198,6 +198,52 @@ static inline struct hlist_head *dev_index_hash(net_t net, int ifindex)
  return &per_net(dev_index_head, net)[ifindex & ((1<<NETDEV_HASHBITS)-1)];
 }

+/* Device list insertion */
+static int list_netdevice(struct net_device *dev)
+{
+ net_t net = dev->nd_net;
+
+ ASSERT_RTNL();
+
+ dev->next = NULL;
+ write_lock_bh(&per_net(dev_base_lock, net));
+ *per_net(dev_tail, net) = dev;
+ per_net(dev_tail, net) = &dev->next;
+ hlist_add_head(&dev->name_hlist, dev_name_hash(net, dev->name));
+ hlist_add_head(&dev->index_hlist, dev_index_hash(net, dev->ifindex));
+ write_unlock_bh(&per_net(dev_base_lock, net));
+ return 0;
+}
+
+/* Device list removal */
+static int unlist_netdevice(struct net_device *dev)
+{
+ struct net_device *d, **dp;
+ net_t net = dev->nd_net;
+
+ ASSERT_RTNL();
+
+ /* Unlink dev from the device chain */
+ for (dp = &per_net(dev_base, net); (d = *dp) != NULL; dp = &d->next) {
+  if (d == dev) {
+   write_lock_bh(&per_net(dev_base_lock, net));
+   hlist_del(&dev->name_hlist);
```

```
+   hlist_del(&dev->index_hlist);
+   if (per_net(dev_tail, net) == &dev->next)
+    per_net(dev_tail, net) = dp;
+   *dp = d->next;
+   write_unlock_bh(&per_net(dev_base_lock, net));
+   break;
+  }
+ }
+ if (!d) {
+  printk(KERN_ERR "unlist net_device: '%s' not found\n",
+         dev->name);
+  return -ENODEV;
+ }
+ return 0;
+}
+
 /*
  * Our notifier list
  */
@@ -3054,15 +3100,9 @@ int register_netdevice(struct net_device *dev)

  set_bit(__LINK_STATE_PRESENT, &dev->state);

- dev->next = NULL;
  dev_init_scheduler(dev);
- write_lock_bh(&per_net(dev_base_lock, net));
- *per_net(dev_tail, net) = dev;
- per_net(dev_tail, net) = &dev->next;
- hlist_add_head(&dev->name_hlist, head);
- hlist_add_head(&dev->index_hlist, dev_index_hash(net, dev->ifindex));
  dev_hold(dev);
- write_unlock_bh(&per_net(dev_base_lock, net));
+ list_netdevice(dev);

  /* Notify protocols, that a new device appeared. */
  raw_notifier_call_chain(&netdev_chain, NETDEV_REGISTER, dev);
@@ -3327,9 +3367,6 @@ void synchronize_net(void)

 int unregister_netdevice(struct net_device *dev)
 {
- struct net_device *d, **dp;
- net_t net = dev->nd_net;
-
  BUG_ON(dev_boot_phase);
  ASSERT_RTNL();

@@ -3347,23 +3384,8 @@ int unregister_netdevice(struct net_device *dev)
   dev_close(dev);
```

```
  /* And unlink it from device chain. */
- for (dp = &per_net(dev_base, net); (d = *dp) != NULL; dp = &d->next) {
-  if (d == dev) {
-   write_lock_bh(&per_net(dev_base_lock, net));
-   hlist_del(&dev->name_hlist);
-   hlist_del(&dev->index_hlist);
-   if (per_net(dev_tail, net) == &dev->next)
-    per_net(dev_tail, net) = dp;
-   *dp = d->next;
-   write_unlock_bh(&per_net(dev_base_lock, net));
-   break;
-  }
- }
- if (!d) {
-  printk(KERN_ERR "unregister net_device: '%s' not found\n",
-         dev->name);
+ if (unlist_netdevice(dev))
   return -ENODEV;
- }

  dev->reg_state = NETREG_UNREGISTERING;

@@ -3419,6 +3441,120 @@ void unregister_netdev(struct net_device *dev)

 EXPORT_SYMBOL(unregister_netdev);

+/**
+ * dev_change_net_namespace - move device to different nethost namespace
+ * @dev: device
+ * @net: network namespace
+ * @pat: If not NULL name pattern to try if the current device name
+ *       is already taken in the destination network namespace.
+ *
+ * This function shuts down a device interface and moves it
+ * to a new network namespace. On success 0 is returned, on
+ * a failure a netagive errno code is returned.
+ *
+ * Callers must hold the rtnl semaphore.
+ */
+
+int dev_change_net_namespace(struct net_device *dev, net_t net, const char *pat)
+{
+ char buf[IFNAMSIZ];
+ const char *destname;
+ int err;
+
+ ASSERT_RTNL();
```

```
+
+ /* Don't allow namespace local devices to be moved. */
+ err = -EINVAL;
+ if (dev->features & NETIF_F_NETNS_LOCAL)
+  goto out;
+
+ /* Ensure the device has been registrered */
+ err = -EINVAL;
+ if (dev->reg_state != NETREG_REGISTERED)
+  goto out;
+
+ /* Get out if there is nothing todo */
+ err = 0;
+ if (net_eq(dev->nd_net, net))
+  goto out;
+
+ /* Pick the destination device name, and ensure
+  * we can use it in the destination network namespace.
+  */
+ err = -EEXIST;
+ destname = dev->name;
+ if (__dev_get_by_name(net, destname) && pat) {
+  /* We get here if we can't use the current device name */
+  if (!dev_valid_name(pat))
+   goto out;
+  if (strchr(pat, '%')) {
+   if (__dev_alloc_name(net, pat, buf) < 0)
+    goto out;
+   destname = buf;
+  } else
+   destname = pat;
+  if (__dev_get_by_name(net, destname))
+   goto out;
+ }
+
+ /*
+  * And now a mini version of register_netdevice unregister_netdevice.
+  */
+
+ /* If device is running close it first. */
+ if (dev->flags & IFF_UP)
+  dev_close(dev);
+
+ /* And unlink it from device chain */
+ err = -ENODEV;
+ if (unlist_netdevice(dev))
+  goto out;
+
```

```
+ synchronize_net();
+
+ /* Shutdown queueing discipline. */
+ dev_shutdown(dev);
+
+ /* Notify protocols, that we are about to destroy
+    this device. They should clean all the things.
+ */
+ call_netdevice_notifiers(NETDEV_UNREGISTER, dev);
+
+ /*
+  * Flush the multicast chain
+  */
+ dev_mc_discard(dev);
+
+ /* Actually switch the network namespace */
+ dev->nd_net = net;
+
+ /* Assign the new device name */
+ if (destname != dev->name)
+  strcpy(dev->name, destname);
+
+ /* If there is an ifindex conflict assign a new one */
+ if (__dev_get_by_index(net, dev->ifindex)) {
+  int iflink = (dev->iflink == dev->ifindex);
+  dev->ifindex = dev_new_index(net);
+  if (iflink)
+   dev->iflink = dev->ifindex;
+ }
+
+ /* Fixup sysfs */
+ class_device_rename(&dev->class_dev, dev->name);
+
+ /* Add the device back in the hashes */
+ list_netdevice(dev);
+
+ /* Notify protocols, that a new device appeared. */
+ call_netdevice_notifiers(NETDEV_REGISTER, dev);
+
+ synchronize_net();
+ err = 0;
+out:
+ return err;
+}
+
 static int dev_cpu_callback(struct notifier_block *nfb,
        unsigned long action,
        void *ocpu)
```

```
@@ -3561,6 +3697,37 @@ static struct pernet_operations netdev_net_ops = {
  .init = netdev_init,
 };

+static void default_device_exit(net_t net)
+{
+ struct net_device *dev, *next;
+ /*
+  * Push all migratable of the network devices back to the
+  * initial network namespace
+  */
+ rtnl_lock();
+ for (dev = per_net(dev_base, net); dev; dev = next) {
+  int err;
+  next = dev->next;
+
+  /* Ignore unmoveable devices (i.e. loopback) */
+  if (dev->features & NETIF_F_NETNS_LOCAL)
+   continue;
+
+  /* Push remaing network devices to init_net */
+  err = dev_change_net_namespace(dev, init_net(), "dev%d");
+  if (err) {
+   printk(KERN_WARNING "%s: failed to move %s to init_net: %d\n",
+    __func__, dev->name, err);
+   unregister_netdevice(dev);
+  }
+ }
+ rtnl_unlock();
+}
+
+static struct pernet_operations default_device_ops = {
+ .exit = default_device_exit,
+};
+
 /*
  * Initialize the DEV module. At boot time this walks the device list and
  * unhooks any devices that fail to initialise (normally hardware not
@@ -3591,6 +3758,9 @@ static int __init net_dev_init(void)
  if (register_pernet_subsys(&netdev_net_ops))
   goto out;

+ if (register_pernet_device(&default_device_ops))
+  goto out;
+
  /*
   * Initialise the packet receive queues.
   */
```

--
1.4.4.1.g278f

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 19/31] net: sysfs interface support for moving devices between network namespaces.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:21 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

I haven't a clue if this interface will meet with widespread approval but
at this point it is simple, and very useful.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/core/net-sysfs.c | 35 +++++++++++++++++++++++++++++++++++
 1 files changed, 35 insertions(+), 0 deletions(-)

diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
index 1be6f94..f8a5c6b 100644
--- a/net/core/net-sysfs.c
+++ b/net/core/net-sysfs.c
@@ -188,6 +188,40 @@ static ssize_t store_mtu(struct class_device *cd, const char *buf, size_t len)
  return netdev_store(cd, buf, len, change_mtu);
 }

+static ssize_t show_new_ns_pid(struct class_device *cd, char *buf)
+{
+ return -EPERM;
+}
+static int change_new_ns_pid(struct net_device *dev, unsigned long new_ns_pid)
+{
+ struct task_struct *tsk;
+ int err;
+ net_t net;
+ /* Look up the network namespace */
+ err = -ESRCH;
+ rcu_read_lock();
+ tsk = find_task_by_pid(new_ns_pid);
+ if (tsk) {
+ task_lock(tsk);

```
+  if (tsk->nsproxy) {
+   err = 0;
+   net = get_net(tsk->nsproxy->net_ns);
+  }
+  task_unlock(tsk);
+ }
+ rcu_read_unlock();
+ /* If I found a network namespace move the device */
+ if (!err) {
+  err = dev_change_net_namespace(dev, net, NULL);
+  put_net(net);
+ }
+ return err;
+}
+static ssize_t store_new_ns_pid(struct class_device *cd, const char *buf, size_t len)
+{
+ return netdev_store(cd, buf, len, change_new_ns_pid);
+}
+
 NETDEVICE_SHOW(flags, fmt_hex);

 static int change_flags(struct net_device *dev, unsigned long new_flags)
@@ -243,6 +277,7 @@ static struct class_device_attribute net_class_attributes[] = {
  __ATTR(tx_queue_len, S_IRUGO | S_IWUSR, show_tx_queue_len,
         store_tx_queue_len),
  __ATTR(weight, S_IRUGO | S_IWUSR, show_weight, store_weight),
+ __ATTR(new_ns_pid, S_IWUSR, show_new_ns_pid, store_new_ns_pid),
  {}
 };

--
1.4.4.1.g278f
```

Subject: [PATCH RFC 20/31] net: Implement CONFIG_NET_NS
Posted by ebiederm on Thu, 25 Jan 2007 19:00:22 GMT

View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Add the config option to enable multiple network namespaces.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 net/Kconfig |    7 +++++++
 1 files changed, 7 insertions(+), 0 deletions(-)

diff --git a/net/Kconfig b/net/Kconfig
index 7dfc949..4671398 100644
--- a/net/Kconfig
+++ b/net/Kconfig
@@ -27,6 +27,13 @@ if NET

 menu "Networking options"

+config NET_NS
+ bool "Network namespace support"
+ depends on EXPERIMENTAL
+ help
+   Support what appear to user space as multiple instances of the
+    network stack.
+
 config NETDEBUG
  bool "Network packet debugging"
  help
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 21/31] net: Implement the guts of the network namespace infrastructure

Posted by ebiederm on Thu, 25 Jan 2007 19:00:23 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Support is added for the .data.pernet section where all of
the variables who have a single instance in each network
namespace will live. Every architectures linker script
is modified so is should work.

Summarizing the functions:
net_ns_init creates a slab and allocates the template and
the initial network namespace.

pernet_modcopy keeps the network namespaces in sync with

the loaded modules. Initializing new data variables as
they are added.

The network namespace destruction because the last reference
can come from interrupt context queues itself for later with
schedule_work. Then we alert everyone the network namespace
is disappearing. If a buggy user is still holding a reference
to the network namespace we print a nasty message and leak
the network namespace.

The wrest are just light-weight wrapper functions to make things
more convinient.

A little should probably be said about net_head the variable
at the start of my network namespace structure. It is the only
variable with a location decided by the C code instead of the linker
and I string them together in a linked list so I can iterate.

Probably more interesting is that it looks like it is saner not
to directly use a pointer to my network namespace but instead to
use an offset. All of the references to data in my network namespace
are coming from per_net(...) which takes the address of the variable
in the .data.pernet section and then adds my magic offset. If I
used a pointer I would have to subract an additional value and
export an extra symbol. Not good for performance or maintenance :)

The expected usage of network namespace variables is to replace
sequences like: &loopback_dev with &per_net(loopback_dev, net)
where net is some network namespace reference. In my preliminary
tests the only a single additional addition is inserted so it
appears to be an efficient idiom. Hopefully it is also easy to
comprehend and use.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/alpha/kernel/vmlinux.lds.S | 2 +
 arch/arm/kernel/vmlinux.lds.S | 3 +
 arch/arm26/kernel/vmlinux-arm26-xip.lds.in | 3 +
 arch/arm26/kernel/vmlinux-arm26.lds.in | 3 +
 arch/avr32/kernel/vmlinux.lds.c | 3 +
 arch/cris/arch-v10/vmlinux.lds.S | 2 +
 arch/cris/arch-v32/vmlinux.lds.S | 2 +
 arch/frv/kernel/vmlinux.lds.S | 2 +
 arch/h8300/kernel/vmlinux.lds.S | 3 +
 arch/i386/kernel/vmlinux.lds.S | 3 +
 arch/ia64/kernel/vmlinux.lds.S | 2 +
 arch/m32r/kernel/vmlinux.lds.S | 3 +
 arch/m68k/kernel/vmlinux-std.lds | 3 +

```
 arch/m68k/kernel/vmlinux-sun3.lds         |    3 +
 arch/m68knommu/kernel/vmlinux.lds.S       |    3 +
 arch/mips/kernel/vmlinux.lds.S            |    3 +
 arch/parisc/kernel/vmlinux.lds.S          |    3 +
 arch/powerpc/kernel/vmlinux.lds.S         |    2 +
 arch/ppc/kernel/vmlinux.lds.S             |    2 +
 arch/s390/kernel/vmlinux.lds.S            |    3 +
 arch/sh/kernel/vmlinux.lds.S              |    3 +
 arch/sh64/kernel/vmlinux.lds.S            |    3 +
 arch/sparc/kernel/vmlinux.lds.S           |    3 +
 arch/sparc64/kernel/vmlinux.lds.S         |    3 +
 arch/v850/kernel/vmlinux.lds.S            |    6 +-
 arch/x86_64/kernel/vmlinux.lds.S          |    3 +
 arch/xtensa/kernel/vmlinux.lds.S          |    2 +
 include/asm-generic/vmlinux.lds.h         |    8 +
 include/asm-um/common.lds.S               |    4 +-
 include/linux/module.h                    |    3 +
 include/linux/net_namespace_type.h        |   63 ++++++++-
 include/net/net_namespace.h               |   49 ++++++-
 kernel/module.c                           |  211 ++++++++++++++++++++++++++-
 net/core/net_namespace.c                  |  232 ++++++++++++++++++++++++++++++
 34 files changed, 631 insertions(+), 15 deletions(-)

diff --git a/arch/alpha/kernel/vmlinux.lds.S b/arch/alpha/kernel/vmlinux.lds.S
index 76bf071..ad20077 100644
--- a/arch/alpha/kernel/vmlinux.lds.S
+++ b/arch/alpha/kernel/vmlinux.lds.S
@@ -72,6 +72,8 @@ SECTIONS
   .data.percpu : { *(.data.percpu) }
   __per_cpu_end = .;

+  DATA_PER_NET
+
   . = ALIGN(2*8192);
   __init_end = .;
   /* Freed after init ends here */
diff --git a/arch/arm/kernel/vmlinux.lds.S b/arch/arm/kernel/vmlinux.lds.S
index a8fa75e..5b003f9 100644
--- a/arch/arm/kernel/vmlinux.lds.S
+++ b/arch/arm/kernel/vmlinux.lds.S
@@ -61,6 +61,9 @@ SECTIONS
   __per_cpu_start = .;
    *(.data.percpu)
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
 #ifndef CONFIG_XIP_KERNEL
```

```
   __init_begin = _stext;
   *(.init.data)
diff --git a/arch/arm26/kernel/vmlinux-arm26-xip.lds.in
b/arch/arm26/kernel/vmlinux-arm26-xip.lds.in
index ca61ec8..69d5772 100644
--- a/arch/arm26/kernel/vmlinux-arm26-xip.lds.in
+++ b/arch/arm26/kernel/vmlinux-arm26-xip.lds.in
@@ -50,6 +50,9 @@ SECTIONS
   __initramfs_start = .;
    usr/built-in.o(.init.ramfs)
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(32768);
   __init_end = .;
  }
diff --git a/arch/arm26/kernel/vmlinux-arm26.lds.in b/arch/arm26/kernel/vmlinux-arm26.lds.in
index d1d3418..473a5b4 100644
--- a/arch/arm26/kernel/vmlinux-arm26.lds.in
+++ b/arch/arm26/kernel/vmlinux-arm26.lds.in
@@ -51,6 +51,9 @@ SECTIONS
   __initramfs_start = .;
    usr/built-in.o(.init.ramfs)
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(32768);
   __init_end = .;
  }
diff --git a/arch/avr32/kernel/vmlinux.lds.c b/arch/avr32/kernel/vmlinux.lds.c
index 5c4424e..dee3715 100644
--- a/arch/avr32/kernel/vmlinux.lds.c
+++ b/arch/avr32/kernel/vmlinux.lds.c
@@ -50,6 +50,9 @@ SECTIONS
   __initramfs_start = .;
    *(.init.ramfs)
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(4096);
   __init_end = .;
  }
diff --git a/arch/cris/arch-v10/vmlinux.lds.S b/arch/cris/arch-v10/vmlinux.lds.S
index 689729a..f1c890c 100644
--- a/arch/cris/arch-v10/vmlinux.lds.S
```

```
+++ b/arch/cris/arch-v10/vmlinux.lds.S
@@ -83,6 +83,8 @@ SECTIONS
  }
  SECURITY_INIT

+ DATA_PER_NET
+
  .init.ramfs : {
   __initramfs_start = .;
   *(.init.ramfs)
diff --git a/arch/cris/arch-v32/vmlinux.lds.S b/arch/cris/arch-v32/vmlinux.lds.S
index 472d4b3..eb08771 100644
--- a/arch/cris/arch-v32/vmlinux.lds.S
+++ b/arch/cris/arch-v32/vmlinux.lds.S
@@ -95,6 +95,8 @@ SECTIONS
  .data.percpu  : { *(.data.percpu) }
  __per_cpu_end = .;

+ DATA_PER_NET
+
  .init.ramfs : {
   __initramfs_start = .;
   *(.init.ramfs)
diff --git a/arch/frv/kernel/vmlinux.lds.S b/arch/frv/kernel/vmlinux.lds.S
index 9c1fb12..f383c83 100644
--- a/arch/frv/kernel/vmlinux.lds.S
+++ b/arch/frv/kernel/vmlinux.lds.S
@@ -61,6 +61,8 @@ SECTIONS
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;

+  DATA_PER_NET
+
   . = ALIGN(4096);
   __initramfs_start = .;
   .init.ramfs : { *(.init.ramfs) }
diff --git a/arch/h8300/kernel/vmlinux.lds.S b/arch/h8300/kernel/vmlinux.lds.S
index f05288b..5d5fda5 100644
--- a/arch/h8300/kernel/vmlinux.lds.S
+++ b/arch/h8300/kernel/vmlinux.lds.S
@@ -130,6 +130,9 @@ SECTIONS
  ___initramfs_start = .;
     *(.init.ramfs)
    ___initramfs_end = .;
+
+ DATA_PER_NET
+
  . = ALIGN(0x4) ;
```

```
  ___init_end = .;
  __edata = . ;
diff --git a/arch/i386/kernel/vmlinux.lds.S b/arch/i386/kernel/vmlinux.lds.S
index a53c8b1..1aae8b4 100644
--- a/arch/i386/kernel/vmlinux.lds.S
+++ b/arch/i386/kernel/vmlinux.lds.S
@@ -193,6 +193,9 @@ SECTIONS
  *(.data.percpu)
  __per_cpu_end = .;
   }
+
+  DATA_PER_NET
+
   . = ALIGN(4096);
   /* freed after init ends here */

diff --git a/arch/ia64/kernel/vmlinux.lds.S b/arch/ia64/kernel/vmlinux.lds.S
index d6083a0..28dd9eb 100644
--- a/arch/ia64/kernel/vmlinux.lds.S
+++ b/arch/ia64/kernel/vmlinux.lds.S
@@ -118,6 +118,8 @@ SECTIONS
    __initramfs_end = .;
  }

+  DATA_PER_NET
+
    . = ALIGN(16);
   .init.setup : AT(ADDR(.init.setup) - LOAD_OFFSET)
         {
diff --git a/arch/m32r/kernel/vmlinux.lds.S b/arch/m32r/kernel/vmlinux.lds.S
index 358b9ce..3e8c624 100644
--- a/arch/m32r/kernel/vmlinux.lds.S
+++ b/arch/m32r/kernel/vmlinux.lds.S
@@ -107,6 +107,9 @@ SECTIONS
   __per_cpu_start = .;
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(4096);
   __init_end = .;
   /* freed after init ends here */
diff --git a/arch/m68k/kernel/vmlinux-std.lds b/arch/m68k/kernel/vmlinux-std.lds
index d279445..d60cb7e 100644
--- a/arch/m68k/kernel/vmlinux-std.lds
+++ b/arch/m68k/kernel/vmlinux-std.lds
@@ -65,6 +65,9 @@ SECTIONS
```

```
   __initramfs_start = .;
   .init.ramfs : { *(.init.ramfs) }
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(8192);
   __init_end = .;

diff --git a/arch/m68k/kernel/vmlinux-sun3.lds b/arch/m68k/kernel/vmlinux-sun3.lds
index 8c7eccb..101ec12 100644
--- a/arch/m68k/kernel/vmlinux-sun3.lds
+++ b/arch/m68k/kernel/vmlinux-sun3.lds
@@ -59,6 +59,9 @@ __init_begin = .;
  __initramfs_start = .;
  .init.ramfs : { *(.init.ramfs) }
  __initramfs_end = .;
+
+ DATA_PER_NET
+
  . = ALIGN(8192);
  __init_end = .;
  .data.init.task : { *(.data.init_task) }
diff --git a/arch/m68knommu/kernel/vmlinux.lds.S b/arch/m68knommu/kernel/vmlinux.lds.S
index 2b2a10d..e713614 100644
--- a/arch/m68knommu/kernel/vmlinux.lds.S
+++ b/arch/m68knommu/kernel/vmlinux.lds.S
@@ -153,6 +153,9 @@ SECTIONS {
   __initramfs_start = .;
   *(.init.ramfs)
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(4096);
   __init_end = .;
  } > INIT
diff --git a/arch/mips/kernel/vmlinux.lds.S b/arch/mips/kernel/vmlinux.lds.S
index cecff24..a5cfeef 100644
--- a/arch/mips/kernel/vmlinux.lds.S
+++ b/arch/mips/kernel/vmlinux.lds.S
@@ -121,6 +121,9 @@ SECTIONS
   __per_cpu_start = .;
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
```

```
   . = ALIGN(_PAGE_SIZE);
   __init_end = .;
   /* freed after init ends here */
diff --git a/arch/parisc/kernel/vmlinux.lds.S b/arch/parisc/kernel/vmlinux.lds.S
index 7b943b4..2cf241b 100644
--- a/arch/parisc/kernel/vmlinux.lds.S
+++ b/arch/parisc/kernel/vmlinux.lds.S
@@ -181,6 +181,9 @@ SECTIONS
   __per_cpu_start = .;
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(ASM_PAGE_SIZE);
   __init_end = .;
   /* freed after init ends here */
diff --git a/arch/powerpc/kernel/vmlinux.lds.S b/arch/powerpc/kernel/vmlinux.lds.S
index 04b8e71..bdd4f05 100644
--- a/arch/powerpc/kernel/vmlinux.lds.S
+++ b/arch/powerpc/kernel/vmlinux.lds.S
@@ -150,6 +150,8 @@ SECTIONS
   __per_cpu_end = .;
  }

+ DATA_PER_NET
+
  . = ALIGN(8);
  .machine.desc : {
   __machine_desc_start = . ;
diff --git a/arch/ppc/kernel/vmlinux.lds.S b/arch/ppc/kernel/vmlinux.lds.S
index 6192126..59c5e6c 100644
--- a/arch/ppc/kernel/vmlinux.lds.S
+++ b/arch/ppc/kernel/vmlinux.lds.S
@@ -135,6 +135,8 @@ SECTIONS
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;

+  DATA_PER_NET
+
   . = ALIGN(4096);
   __initramfs_start = .;
   .init.ramfs : { *(.init.ramfs) }
diff --git a/arch/s390/kernel/vmlinux.lds.S b/arch/s390/kernel/vmlinux.lds.S
index fe0f2e9..bcdd353 100644
--- a/arch/s390/kernel/vmlinux.lds.S
+++ b/arch/s390/kernel/vmlinux.lds.S
@@ -99,6 +99,9 @@ SECTIONS
```

```
   __per_cpu_start = .;
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(4096);
   __init_end = .;
   /* freed after init ends here */
diff --git a/arch/sh/kernel/vmlinux.lds.S b/arch/sh/kernel/vmlinux.lds.S
index f34bdcc..0a4249d 100644
--- a/arch/sh/kernel/vmlinux.lds.S
+++ b/arch/sh/kernel/vmlinux.lds.S
@@ -86,6 +86,9 @@ SECTIONS
   __initramfs_start = .;
   .init.ramfs : { *(.init.ramfs) }
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   __machvec_start = .;
   .init.machvec : { *(.init.machvec) }
   __machvec_end = .;
diff --git a/arch/sh64/kernel/vmlinux.lds.S b/arch/sh64/kernel/vmlinux.lds.S
index 95c4d75..0c1a30e 100644
--- a/arch/sh64/kernel/vmlinux.lds.S
+++ b/arch/sh64/kernel/vmlinux.lds.S
@@ -118,6 +118,9 @@ SECTIONS
   __initramfs_start = .;
   .init.ramfs : C_PHYS(.init.ramfs) { *(.init.ramfs) }
   __initramfs_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(PAGE_SIZE);
   __init_end = .;

diff --git a/arch/sparc/kernel/vmlinux.lds.S b/arch/sparc/kernel/vmlinux.lds.S
index b73e6b9..c1ff7de 100644
--- a/arch/sparc/kernel/vmlinux.lds.S
+++ b/arch/sparc/kernel/vmlinux.lds.S
@@ -65,6 +65,9 @@ SECTIONS
   __per_cpu_start = .;
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
```

```
   . = ALIGN(4096);
   __init_end = .;
   . = ALIGN(32);
diff --git a/arch/sparc64/kernel/vmlinux.lds.S b/arch/sparc64/kernel/vmlinux.lds.S
index 4a6063f..24e6b7f 100644
--- a/arch/sparc64/kernel/vmlinux.lds.S
+++ b/arch/sparc64/kernel/vmlinux.lds.S
@@ -89,6 +89,9 @@ SECTIONS
   __per_cpu_start = .;
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(8192);
   __init_end = .;
   __bss_start = .;
diff --git a/arch/v850/kernel/vmlinux.lds.S b/arch/v850/kernel/vmlinux.lds.S
index 3a5fd07..b87a4cb 100644
--- a/arch/v850/kernel/vmlinux.lds.S
+++ b/arch/v850/kernel/vmlinux.lds.S
@@ -163,7 +163,8 @@
    *(.text.init) /* 2.4 convention */       \
    *(.data.init)          \
   INITCALL_CONTENTS          \
-  INITRAMFS_CONTENTS
+  INITRAMFS_CONTENTS          \
+  DATA_PER_NET

  /* The contents of `init' section for a ROM-resident kernel which
     should go into RAM.  */
@@ -183,7 +184,8 @@
    *(.text.init) /* 2.4 convention */       \
   INITCALL_CONTENTS          \
-  INITRAMFS_CONTENTS
+  INITRAMFS_CONTENTS          \
+  DATA_PER_NET

  /* A root filesystem image, for kernels with an embedded root filesystem.  */
  #define ROOT_FS_CONTENTS           \
diff --git a/arch/x86_64/kernel/vmlinux.lds.S b/arch/x86_64/kernel/vmlinux.lds.S
index 1e54ddf..38061b2 100644
--- a/arch/x86_64/kernel/vmlinux.lds.S
+++ b/arch/x86_64/kernel/vmlinux.lds.S
@@ -200,6 +200,9 @@ SECTIONS
   __per_cpu_start = .;
   .data.percpu  : AT(ADDR(.data.percpu) - LOAD_OFFSET) { *(.data.percpu) }
```

```
   __per_cpu_end = .;
+
+  DATA_PER_NET
+
   . = ALIGN(4096);
   __init_end = .;

diff --git a/arch/xtensa/kernel/vmlinux.lds.S b/arch/xtensa/kernel/vmlinux.lds.S
index a36c104..e77ed43 100644
--- a/arch/xtensa/kernel/vmlinux.lds.S
+++ b/arch/xtensa/kernel/vmlinux.lds.S
@@ -203,6 +203,8 @@ SECTIONS
   .data.percpu  : { *(.data.percpu) }
   __per_cpu_end = .;

+  DATA_PER_NET
+
   . = ALIGN(4096);
   __initramfs_start =.;
   .init.ramfs : { *(.init.ramfs) }
diff --git a/include/asm-generic/vmlinux.lds.h b/include/asm-generic/vmlinux.lds.h
index 9fcc8d9..298ed43 100644
--- a/include/asm-generic/vmlinux.lds.h
+++ b/include/asm-generic/vmlinux.lds.h
@@ -229,3 +229,11 @@
    *(.initcall7.init)      \
    *(.initcall7s.init)

+#define DATA_PER_NET       \
+ .data.pernet : AT(ADDR(.data.pernet) - LOAD_OFFSET) {  \
+  VMLINUX_SYMBOL(__per_net_start) = .;   \
+  *(.data.pernet.head)     \
+  *(.data.pernet)      \
+  VMLINUX_SYMBOL(__per_net_end) = .;   \
+ }
+
diff --git a/include/asm-um/common.lds.S b/include/asm-um/common.lds.S
index f045451..1208960 100644
--- a/include/asm-um/common.lds.S
+++ b/include/asm-um/common.lds.S
@@ -39,7 +39,9 @@
   __per_cpu_start = . ;
   .data.percpu : { *(.data.percpu) }
   __per_cpu_end = . ;
-
+
+  DATA_PER_NET
+
```

```
   __initcall_start = .;
   .initcall.init : {
  INITCALLS
diff --git a/include/linux/module.h b/include/linux/module.h
index 10f771a..755f1b5 100644
--- a/include/linux/module.h
+++ b/include/linux/module.h
@@ -353,6 +353,9 @@ struct module
  /* Per-cpu data. */
  void *percpu;

+ /* Per-net data. */
+ void *pernet;
+
  /* The command line arguments (may be mangled).  People like
     keeping pointers to this stuff */
  char *args;
diff --git a/include/linux/net_namespace_type.h b/include/linux/net_namespace_type.h
index 8173f59..5075199 100644
--- a/include/linux/net_namespace_type.h
+++ b/include/linux/net_namespace_type.h
@@ -7,14 +7,70 @@

 #define __pernetname(name) per_net__##name

+#ifdef CONFIG_NET_NS
+
+typedef struct {
+ unsigned long offset;
+} net_t;
+
+#define __data_pernet __attribute__((__section__(".data.pernet")))
+
+static inline unsigned long __per_net_offset(net_t net) { return net.offset; }
+
+/* Like per_net but returns a pseudo variable address that must be offset
+ * __per_net_offset() bytes before it will point to a real variable.
+ * Useful for static initializers.
+ */
+#define __per_net_base(name)   __pernetname(name)
+
+/* Get the network namespace reference from a per_net variable address */
+#define net_of(ptr, name)      \
+({         \
+ net_t net = { .offset = 0 };     \
+ char *__ptr = (void *)(ptr);     \
+ if (__ptr)       \
+  net.offset = __ptr - ((char *)&__per_net_base(name)); \
```

```
+ net;       \
+})
+
+/* Look up a per network namespace variable */
+#define per_net(var, net) (*(     \
+ RELOC_HIDE(&__per_net_base(var), __per_net_offset(net))))
+
+/* A more efficient form if gcc doesn't overoptimize it */
+#ifndef per_net
+#define per_net(var, net) (*(     \
+ (typeof(__pernetname(var)) *)    \
+ (((char *)&__per_net_base(var)) + __per_net_offset(net))))
+#endif
+
+
+/* Are the two network namespaces the same */
+static inline int net_eq(net_t a, net_t b) { return a.offset == b.offset; }
+
+/* Get an unsigned value appropriate for hashing the network namespace */
+static inline unsigned int net_hval(net_t net) { return net.offset; }
+
+/* Convert to and from to and from void pointers */
+static inline void *net_to_voidp(net_t net) { return (void *)net.offset; }
+static inline net_t net_from_voidp(void *ptr)
+{
+ net_t r;
+ r.offset = (unsigned long)ptr;
+ return r;
+}
+
+static inline int null_net(net_t net) { return net.offset == 0; }
+
+#else /* CONFIG_NET_NS */
+
 typedef struct {} net_t;

 #define __data_pernet

-/* Look up a per network namespace variable */
 static inline unsigned long __per_net_offset(net_t net) { return 0; }

-/* Like per_net but returns a pseudo variable address that must be moved
+/* Like per_net but returns a pseudo variable address that must be offset
  * __per_net_offset() bytes before it will point to a real variable.
  * Useful for static initializers.
  */
@@ -38,6 +94,9 @@ static inline net_t net_from_voidp(void *ptr) { net_t net; return net; }
```

```
 static inline int null_net(net_t net) { return 0; }

+#endif /* CONFIG_NET_NS */
+
+
 #define DEFINE_PER_NET(type, name)    \
  __data_pernet __typeof__(type) __pernetname(name)

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index b64568f..a2042ac 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -24,7 +24,8 @@ struct net_namespace_head {
      * should go
      */
  atomic_t use_count; /* For references we destroy on demand */
- struct list_head list;
+ net_t next;
+ net_t prev;
  struct work_struct work;
 };

@@ -34,6 +35,50 @@ static inline net_t init_net(void)
  return init_nsproxy.net_ns;
 }

+#ifdef CONFIG_NET_NS
+
+DECLARE_PER_NET(struct net_namespace_head, net_head);
+
+extern void pernet_modcopy(void *, const void *, unsigned long);
+extern int copy_net(int, struct task_struct *);
+extern void __put_net(net_t net);
+
+static inline net_t get_net(net_t net)
+{
+ atomic_inc(&per_net(net_head, net).count);
+ return net;
+}
+
+static inline void put_net(net_t net)
+{
+ if (atomic_dec_and_test(&per_net(net_head, net).count))
+  __put_net(net);
+}
+
+static inline net_t hold_net(net_t net)
+{
```

```
+ atomic_inc(&per_net(net_head, net).use_count);
+ return net;
+}
+
+static inline void release_net(net_t net)
+{
+ atomic_dec(&per_net(net_head, net).use_count);
+}
+
+/* Created by linker magic */
+extern char __per_net_start[], __per_net_end[];
+
+extern void net_lock(void);
+extern void net_unlock(void);
+
+#define for_each_net(VAR)    \
+ for ( (VAR) = init_net(); !null_net((VAR)); \
+  (VAR) = per_net(net_head, (VAR)).next)
+
+
+#else /* CONFIG_NET_NS */
+
 static inline net_t get_net(net_t net) { return net; }
 static inline void put_net(net_t net) {}
 static inline net_t hold_net(net_t net) { return net; }
@@ -50,6 +95,8 @@ static inline void net_unlock(void) {}

 #define for_each_net(VAR) if (1)

+#endif /* CONFIG_NET_NS */
+
 extern net_t net_template;

 #define NET_CREATE 0x0001 /* A network namespace has been created */
diff --git a/kernel/module.c b/kernel/module.c
index d0f2260..6f45090 100644
--- a/kernel/module.c
+++ b/kernel/module.c
@@ -44,6 +44,7 @@
 #include <asm/semaphore.h>
 #include <asm/cacheflush.h>
 #include <linux/license.h>
+#include <net/net_namespace.h>

 #if 0
 #define DEBUGP printk
@@ -304,7 +305,7 @@ static unsigned int pcpu_num_used, pcpu_num_allocated;
 /* Size of each block.  -ve means used. */
```

```
 static int *pcpu_size;

-static int split_block(unsigned int i, unsigned short size)
+static int pcpu_split_block(unsigned int i, unsigned short size)
 {
  /* Reallocation required? */
  if (pcpu_num_used + 1 > pcpu_num_allocated) {
@@ -329,7 +330,7 @@ static int split_block(unsigned int i, unsigned short size)
  return 1;
 }

-static inline unsigned int block_size(int val)
+static inline unsigned int pcpu_block_size(int val)
 {
  if (val < 0)
   return -val;
@@ -353,7 +354,7 @@ static void *percpu_modalloc(unsigned long size, unsigned long align,
  }

  ptr = __per_cpu_start;
- for (i = 0; i < pcpu_num_used; ptr += block_size(pcpu_size[i]), i++) {
+ for (i = 0; i < pcpu_num_used; ptr += pcpu_block_size(pcpu_size[i]), i++) {
   /* Extra for alignment requirement. */
   extra = ALIGN((unsigned long)ptr, align) - (unsigned long)ptr;
   BUG_ON(i == 0 && extra != 0);
@@ -371,7 +372,7 @@ static void *percpu_modalloc(unsigned long size, unsigned long align,

   /* Split block if warranted */
   if (pcpu_size[i] - size > sizeof(unsigned long))
-   if (!split_block(i, size))
+   if (!pcpu_split_block(i, size))
     return NULL;

   /* Mark allocated */
@@ -387,10 +388,10 @@ static void *percpu_modalloc(unsigned long size, unsigned long align,
 static void percpu_modfree(void *freeme)
 {
  unsigned int i;
- void *ptr = __per_cpu_start + block_size(pcpu_size[0]);
+ void *ptr = __per_cpu_start + pcpu_block_size(pcpu_size[0]);

  /* First entry is core kernel percpu data. */
- for (i = 1; i < pcpu_num_used; ptr += block_size(pcpu_size[i]), i++) {
+ for (i = 1; i < pcpu_num_used; ptr += pcpu_block_size(pcpu_size[i]), i++) {
   if (ptr == freeme) {
    pcpu_size[i] = -pcpu_size[i];
    goto free;
@@ -465,6 +466,169 @@ static inline void percpu_modcopy(void *pcpudst, const void *src,
```

```
 }
 #endif /* CONFIG_SMP */

+#ifdef CONFIG_NET_NS
+/* Number of blocks used and allocated. */
+static unsigned int pnet_num_used, pnet_num_allocated;
+/* Size of each block.  -ve means used. */
+static int *pnet_size;
+
+static int pnet_split_block(unsigned int i, unsigned short size)
+{
+ /* Reallocation required? */
+ if (pnet_num_used + 1 > pnet_num_allocated) {
+  int *new = kmalloc(sizeof(new[0]) * pnet_num_allocated*2,
+       GFP_KERNEL);
+  if (!new)
+   return 0;
+
+  memcpy(new, pnet_size, sizeof(new[0])*pnet_num_allocated);
+  pnet_num_allocated *= 2;
+  kfree(pnet_size);
+  pnet_size = new;
+ }
+
+ /* Insert a new subblock */
+ memmove(&pnet_size[i+1], &pnet_size[i],
+  sizeof(pnet_size[0]) * (pnet_num_used - i));
+ pnet_num_used++;
+
+ pnet_size[i+1] -= size;
+ pnet_size[i] = size;
+ return 1;
+}
+
+static inline unsigned int pnet_block_size(int val)
+{
+ if (val < 0)
+  return -val;
+ return val;
+}
+
+static void *pernet_modalloc(unsigned long size, unsigned long align,
+        const char *name)
+{
+ unsigned long extra;
+ unsigned int i;
+ void *ptr;
+
```

```
+ if (align > SMP_CACHE_BYTES) {
+  printk(KERN_WARNING "%s: per-net alignment %li > %i\n",
+         name, align, SMP_CACHE_BYTES);
+  align = SMP_CACHE_BYTES;
+ }
+
+ ptr = __per_net_start;
+ for (i = 0; i < pnet_num_used; ptr += pnet_block_size(pnet_size[i]), i++) {
+  /* Extra for alignment requirement. */
+  extra = ALIGN((unsigned long)ptr, align) - (unsigned long)ptr;
+  BUG_ON(i == 0 && extra != 0);
+
+  if (pnet_size[i] < 0 || pnet_size[i] < extra + size)
+   continue;
+
+  /* Transfer extra to previous block. */
+  if (pnet_size[i-1] < 0)
+   pnet_size[i-1] -= extra;
+  else
+   pnet_size[i-1] += extra;
+  pnet_size[i] -= extra;
+  ptr += extra;
+
+  /* Split block if warranted */
+  if (pnet_size[i] - size > sizeof(unsigned long))
+   if (!pnet_split_block(i, size))
+    return NULL;
+
+  /* Mark allocated */
+  pnet_size[i] = -pnet_size[i];
+  return ptr;
+ }
+
+ printk(KERN_WARNING "Could not allocate %lu bytes pernet data\n",
+        size);
+ return NULL;
+}
+
+static void pernet_modfree(void *freeme)
+{
+ unsigned int i;
+ void *ptr = __per_net_start + pnet_block_size(pnet_size[0]);
+
+ /* First entry is core kernel pernet data. */
+ for (i = 1; i < pnet_num_used; ptr += pnet_block_size(pnet_size[i]), i++) {
+  if (ptr == freeme) {
+   pnet_size[i] = -pnet_size[i];
+   goto free;
```

```
+  }
+ }
+ BUG();
+
+ free:
+ /* Merge with previous? */
+ if (pnet_size[i-1] >= 0) {
+  pnet_size[i-1] += pnet_size[i];
+  pnet_num_used--;
+  memmove(&pnet_size[i], &pnet_size[i+1],
+   (pnet_num_used - i) * sizeof(pnet_size[0]));
+  i--;
+ }
+ /* Merge with next? */
+ if (i+1 < pnet_num_used && pnet_size[i+1] >= 0) {
+  pnet_size[i] += pnet_size[i+1];
+  pnet_num_used--;
+  memmove(&pnet_size[i+1], &pnet_size[i+2],
+   (pnet_num_used - (i+1)) * sizeof(pnet_size[0]));
+ }
+}
+
+static unsigned int find_pnetsec(Elf_Ehdr *hdr,
+     Elf_Shdr *sechdrs,
+     const char *secstrings)
+{
+ return find_sec(hdr, sechdrs, secstrings, ".data.pernet");
+}
+
+static int pernet_modinit(void)
+{
+ pnet_num_used = 2;
+ pnet_num_allocated = 2;
+ pnet_size = kmalloc(sizeof(pnet_size[0]) *pnet_num_allocated,
+       GFP_KERNEL);
+ /* Static in-kernel pernet data (used). */
+ pnet_size[0] = -ALIGN(__per_net_end-__per_net_start, SMP_CACHE_BYTES);
+ /* Free room. */
+ pnet_size[1] = PER_NET_MODULE_RESERVE;
+ if (pnet_size[1] <= 0) {
+  printk(KERN_ERR "No per-net room for modules.\n");
+  pnet_num_used = 1 ;
+ }
+ return 0;
+}
+__initcall(pernet_modinit);
+#else /* ... !CONFIG_NET_NS */
+static inline void *pernet_modalloc(unsigned long size, unsigned long align,
```

```
+        const char *name)
+{
+ return NULL;
+}
+static inline void pernet_modfree(void *pnetptr)
+{
+ BUG();
+}
+static inline unsigned int find_pnetsec(Elf_Ehdr *hdr,
+      Elf_Shdr *sechdrs,
+      const char *secstrings)
+{
+ return 0;
+}
+static inline void pernet_modcopy(void *pnetdst, const void *src,
+       unsigned long size)
+{
+ /* pnetsec should be 0, and size of that section should be 0. */
+ BUG_ON(size != 0);
+}
+#endif /* CONFIG_NET_NS */
+
 #define MODINFO_ATTR(field) \
 static void setup_modinfo_##field(struct module *mod, const char *s)  \
 {                                                                     \
@@ -1198,6 +1362,8 @@ static void free_module(struct module *mod)
  /* This may be NULL, but that's OK */
  module_free(mod, mod->module_init);
  kfree(mod->args);
+ if (mod->pernet)
+  pernet_modfree(mod->pernet);
  if (mod->percpu)
   percpu_modfree(mod->percpu);

@@ -1263,6 +1429,7 @@ static int simplify_symbols(Elf_Shdr *sechdrs,
        const char *strtab,
        unsigned int versindex,
        unsigned int pcpuindex,
+       unsigned int pnetindex,
        struct module *mod)
 {
  Elf_Sym *sym = (void *)sechdrs[symindex].sh_addr;
@@ -1308,6 +1475,9 @@ static int simplify_symbols(Elf_Shdr *sechdrs,
    /* Divert to percpu allocation if a percpu var. */
    if (sym[i].st_shndx == pcpuindex)
     secbase = (unsigned long)mod->percpu;
+   /* Divert to pernet allocation if a pernet var. */
+   else if (sym[i].st_shndx == pnetindex)
```

```
+    secbase = (unsigned long)mod->pernet;
     else
      secbase = sechdrs[sym[i].st_shndx].sh_addr;
     sym[i].st_value += secbase;
@@ -1554,6 +1724,7 @@ static struct module *load_module(void __user *umod,
  unsigned int gplcrcindex;
  unsigned int versindex;
  unsigned int pcpuindex;
+ unsigned int pnetindex;
  unsigned int gplfutureindex;
  unsigned int gplfuturecrcindex;
  unsigned int unwindex = 0;
@@ -1563,7 +1734,7 @@ static struct module *load_module(void __user *umod,
  unsigned int unusedgplcrcindex;
  struct module *mod;
  long err = 0;
- void *percpu = NULL, *ptr = NULL; /* Stops spurious gcc warning */
+ void *percpu = NULL, *pernet = NULL, *ptr = NULL; /* Stops spurious gcc warning */
  struct exception_table_entry *extable;
  mm_segment_t old_fs;

@@ -1654,6 +1825,7 @@ static struct module *load_module(void __user *umod,
  versindex = find_sec(hdr, sechdrs, secstrings, "__versions");
  infoindex = find_sec(hdr, sechdrs, secstrings, ".modinfo");
  pcpuindex = find_pcpusec(hdr, sechdrs, secstrings);
+ pnetindex = find_pnetsec(hdr, sechdrs, secstrings);
 #ifdef ARCH_UNWIND_SECTION_NAME
  unwindex = find_sec(hdr, sechdrs, secstrings, ARCH_UNWIND_SECTION_NAME);
 #endif
@@ -1719,6 +1891,20 @@ static struct module *load_module(void __user *umod,
   mod->percpu = percpu;
  }

+ if (pnetindex) {
+  /* We have a special allocation for this section */
+  pernet = pernet_modalloc(sechdrs[pnetindex].sh_size,
+       sechdrs[pnetindex].sh_addralign,
+       mod->name);
+
+  if (!pernet) {
+   err = -ENOMEM;
+   goto free_percpu;
+  }
+  sechdrs[pnetindex].sh_flags &= ~(unsigned long)SHF_ALLOC;
+  mod->pernet = pernet;
+ }
+
  /* Determine total sizes, and put offsets in sh_entsize.  For now
```

```
     this is done generically; there doesn't appear to be any
     special cases for the architectures. */
@@ -1728,7 +1914,7 @@ static struct module *load_module(void __user *umod,
  ptr = module_alloc(mod->core_size);
  if (!ptr) {
   err = -ENOMEM;
-  goto free_percpu;
+  goto free_pernet;
  }
  memset(ptr, 0, mod->core_size);
  mod->module_core = ptr;
@@ -1781,7 +1967,7 @@ static struct module *load_module(void __user *umod,

  /* Fix up syms, so that st_value is a pointer to location. */
  err = simplify_symbols(sechdrs, symindex, strtab, versindex, pcpuindex,
-         mod);
+         pnetindex, mod);
  if (err < 0)
   goto cleanup;

@@ -1860,6 +2046,10 @@ static struct module *load_module(void __user *umod,
  percpu_modcopy(mod->percpu, (void *)sechdrs[pcpuindex].sh_addr,
         sechdrs[pcpuindex].sh_size);

+ /* Copy pernet area over. */
+ pernet_modcopy(mod->pernet, (void *)sechdrs[pnetindex].sh_addr,
+         sechdrs[pnetindex].sh_size);
+
  add_kallsyms(mod, sechdrs, symindex, strindex, secstrings);

  err = module_finalize(hdr, sechdrs, mod);
@@ -1924,6 +2114,9 @@ static struct module *load_module(void __user *umod,
  cleanup:
  module_unload_free(mod);
  module_free(mod, mod->module_init);
+ free_pernet:
+ if (pernet)
+  pernet_modfree(pernet);
  free_core:
  module_free(mod, mod->module_core);
  free_percpu:
diff --git a/net/core/net_namespace.c b/net/core/net_namespace.c
index 4ae266d..93e3879 100644
--- a/net/core/net_namespace.c
+++ b/net/core/net_namespace.c
@@ -1,4 +1,9 @@
+#include <linux/workqueue.h>
 #include <linux/rtnetlink.h>
```

```
+#include <linux/cache.h>
+#include <linux/slab.h>
+#include <linux/list.h>
+#include <linux/delay.h>
 #include <net/net_namespace.h>

 /*
@@ -10,6 +15,233 @@ static struct list_head *first_device = &pernet_list;
 static DEFINE_MUTEX(net_mutex);
 net_t net_template;

+#ifdef CONFIG_NET_NS
+
+static DEFINE_MUTEX(net_list_mutex);
+
+static net_t net_tail;
+static struct kmem_cache *net_cachep;
+static size_t net_size;
+
+/* By using a special section for the first variable in the
+ * per net sectionI get several advantages.
+ * - I can align the entire network namespace structure easily
+ *   to any desired alignment without needing an alignment directive
+ *   in the linker script.  In the worst case the section will start
+ *   with some padding I will never see.
+ * - The code is C so I don't need linker script or header file tricks
+ *   to make the alignment SMP_CACHE_BYTES
+ * - I am guaranteed what the first structure in the network namespace is.
+ *   This allows things like container_of to work and be useful.
+ */
+__attribute__((section(".data.pernet.head"), aligned(SMP_CACHE_BYTES)))
+struct net_namespace_head __pernetname(net_head) = {
+ .count = ATOMIC_INIT(1),
+ .use_count = ATOMIC_INIT(0),
+};
+EXPORT_PER_NET_SYMBOL_GPL(net_head);
+
+void net_lock(void)
+{
+ mutex_lock(&net_list_mutex);
+}
+
+void net_unlock(void)
+{
+ mutex_unlock(&net_list_mutex);
+}
+
+static void net_list_remove(net_t net)
```

```
+{
+ net_t next, prev;
+ BUG_ON(net_eq(net, init_net()));
+
+ next = per_net(net_head, net).next;
+ prev = per_net(net_head, net).prev;
+
+ per_net(net_head, prev).next = next;
+ if (null_net(next)) {
+  net_tail = prev;
+ } else {
+  per_net(net_head, next).prev = prev;
+ }
+}
+
+static void net_list_append(net_t net)
+{
+
+ per_net(net_head, net_tail).next = net;
+ per_net(net_head, net).prev = net_tail;
+ net_tail = net;
+}
+
+static net_t net_alloc(void)
+{
+ return net_of(kmem_cache_alloc(net_cachep, GFP_KERNEL), net_head);
+}
+
+static void net_free(net_t net)
+{
+ struct net_namespace_head *head;
+ if (null_net(net))
+  return;
+
+ head = &per_net(net_head, net);
+
+ if (unlikely(atomic_read(&head->use_count) != 0)) {
+  printk(KERN_EMERG "network namespace not free! Usage: %d\n",
+   atomic_read(&head->use_count));
+  return;
+ }
+
+ kmem_cache_free(net_cachep, head);
+}
+
+static void cleanup_net(struct work_struct *work)
+{
+ struct pernet_operations *ops;
```

```
+ struct list_head *ptr;
+ net_t net;
+
+ net = net_of(work, net_head.work);
+
+ mutex_lock(&net_mutex);
+
+ /* Don't let anyone else find us. */
+ net_lock();
+ net_list_remove(net);
+ net_unlock();
+
+ /* Run all of the network namespace exit methods */
+ list_for_each_prev(ptr, &pernet_list) {
+  ops = list_entry(ptr, struct pernet_operations, list);
+  if (ops->exit)
+   ops->exit(net);
+ }
+
+ mutex_unlock(&net_mutex);
+
+ /* Ensure there are no outstanding rcu callbacks using this
+  * network namespace.
+  */
+ rcu_barrier();
+
+ /* Finally it is safe to free my network namespace structure */
+ net_free(net);
+}
+
+
+void __put_net(net_t net)
+{
+ /* Cleanup the network namespace in process context */
+ INIT_WORK(&per_net(net_head, net).work, cleanup_net);
+ schedule_work(&per_net(net_head, net).work);
+}
+EXPORT_SYMBOL_GPL(__put_net);
+
+/*
+ * setup_net runs the initializers for the network namespace object.
+ */
+static int setup_net(net_t net)
+{
+ /* Must be called with net_mutex held */
+ struct pernet_operations *ops;
+ struct list_head *ptr;
+ int error;
```

```
+
+ /* First initialize the data from the template */
+ memcpy(&per_net(net_head, net), &per_net(net_head, net_template), net_size);
+
+ error = 0;
+ list_for_each(ptr, &pernet_list) {
+  ops = list_entry(ptr, struct pernet_operations, list);
+  if (ops->init) {
+   error = ops->init(net);
+   if (error < 0)
+    goto out_undo;
+  }
+ }
+out:
+ return error;
+out_undo:
+ /* Walk through the list backwards calling the exit functions
+  * for the pernet modules whose init functions did not fail.
+  */
+ for (ptr = ptr->prev; ptr != &pernet_list; ptr = ptr->prev) {
+  ops = list_entry(ptr, struct pernet_operations, list);
+  if (ops->exit)
+   ops->exit(net);
+ }
+ goto out;
+}
+
+void pernet_modcopy(void *pnetdst, const void *src, unsigned long size)
+{
+ net_t net;
+
+ mutex_lock(&net_mutex);
+ memcpy(pnetdst + __per_net_offset(net_template), src, size);
+ for_each_net(net)
+  memcpy(pnetdst + __per_net_offset(net), src, size);
+ mutex_unlock(&net_mutex);
+}
+
+static int __init net_ns_init(void)
+{
+ size_t init_size;
+ net_t init_net;
+ int err;
+
+ /* Compute the size of the init section */
+ init_size = __per_net_end - __per_net_start;
+
+ /* Compute how large my net namespace structure will be */
```

```
+ net_size = ALIGN(init_size, SMP_CACHE_BYTES);
+ net_size += PER_NET_MODULE_RESERVE;
+ net_size = ALIGN(net_size, SMP_CACHE_BYTES);
+
+ printk(KERN_INFO "net_namespace: %zd bytes\n", net_size);
+ net_cachep = kmem_cache_create("net_namespace", net_size,
+     SMP_CACHE_BYTES,
+     SLAB_PANIC, NULL, NULL);
+
+ /* Allocate my template */
+ net_template = net_alloc();
+ if (null_net(net_template))
+  panic("Could not allocate network namespace template");
+
+ /* Initialize my template */
+ memset(&per_net(net_head, net_template), '\0', net_size);
+ memcpy(&per_net(net_head, net_template),
+  &__pernetname(net_head),
+  init_size);
+
+ /* Setup the initial network namespace */
+ init_net = net_alloc();
+ if (null_net(init_net))
+  panic("Could not allocate initial network namespace");
+
+ mutex_lock(&net_mutex);
+ err = setup_net(init_net);
+
+ net_lock();
+ net_tail = init_net;
+ net_unlock();
+
+ mutex_unlock(&net_mutex);
+ if (err)
+  panic("Could not setup the initial network namespace");
+
+ /* Initialze the init_nsproxy */
+ init_nsproxy.net_ns = init_net;
+
+ return 0;
+}
+
+pure_initcall(net_ns_init);
+
+#endif /* CONFIG_NET_NS */
+
 static int register_pernet_operations(struct list_head *list,
         struct pernet_operations *ops)
```

{
--
1.4.4.1.g278f

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 22/31] net: Add network namespace clone support.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:24 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch allows you to create a new network namespace
using sys_clone(...).

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
include/linux/sched.h | 1 +
kernel/nsproxy.c | 11 ++++++++++
net/core/net_namespace.c | 38 ++++++++++++++++++++++++++++++++++++++
3 files changed, 50 insertions(+), 0 deletions(-)

diff --git a/include/linux/sched.h b/include/linux/sched.h
index 4463735..9e0f91a 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -26,6 +26,7 @@
#define CLONE_STOPPED 0x02000000 /* Start in stopped state */
#define CLONE_NEWUTS 0x04000000 /* New utsname group? */
#define CLONE_NEWIPC 0x08000000 /* New ipcs */
+#define CLONE_NEWNET 0x20000000 /* New network namespace */

/*
* Scheduling policies
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index 4f3c95a..7861c4c 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -20,6 +20,7 @@
#include <linux/mnt_namespace.h>
#include <linux/utsname.h>
#include <linux/pid_namespace.h>
+#include <net/net_namespace.h>

```
 struct nsproxy init_nsproxy = INIT_NSPROXY(init_nsproxy);
 EXPORT_SYMBOL_GPL(init_nsproxy);
@@ -70,6 +71,7 @@ struct nsproxy *dup_namespaces(struct nsproxy *orig)
    get_ipc_ns(ns->ipc_ns);
   if (ns->pid_ns)
    get_pid_ns(ns->pid_ns);
+  get_net(ns->net_ns);
  }

  return ns;
@@ -117,10 +119,18 @@ int copy_namespaces(int flags, struct task_struct *tsk)
  if (err)
   goto out_pid;

+ err = copy_net(flags, tsk);
+ if (err)
+  goto out_net;
+
 out:
  put_nsproxy(old_ns);
  return err;

+out_net:
+ if (new_ns->pid_ns)
+  put_pid_ns(new_ns->pid_ns);
+
 out_pid:
  if (new_ns->ipc_ns)
   put_ipc_ns(new_ns->ipc_ns);
@@ -146,5 +156,6 @@ void free_nsproxy(struct nsproxy *ns)
   put_ipc_ns(ns->ipc_ns);
  if (ns->pid_ns)
   put_pid_ns(ns->pid_ns);
+ put_net(ns->net_ns);
  kfree(ns);
 }
diff --git a/net/core/net_namespace.c b/net/core/net_namespace.c
index 93e3879..cc56105 100644
--- a/net/core/net_namespace.c
+++ b/net/core/net_namespace.c
@@ -175,6 +175,44 @@ out_undo:
  goto out;
 }

+int copy_net(int flags, struct task_struct *tsk)
+{
+ net_t old_net = tsk->nsproxy->net_ns;
+ net_t new_net;
```

```
+ int err;
+
+ get_net(old_net);
+
+ if (!(flags & CLONE_NEWNET))
+  return 0;
+
+ err = -EPERM;
+ if (!capable(CAP_SYS_ADMIN))
+  goto out;
+
+ err = -ENOMEM;
+ new_net = net_alloc();
+ if (null_net(new_net))
+  goto out;
+
+ mutex_lock(&net_mutex);
+ err = setup_net(new_net);
+ if (err)
+  goto out_unlock;
+
+ net_lock();
+ net_list_append(new_net);
+ net_unlock();
+
+ tsk->nsproxy->net_ns = new_net;
+
+out_unlock:
+ mutex_unlock(&net_mutex);
+out:
+ put_net(old_net);
+ return err;
+}
+
 void pernet_modcopy(void *pnetdst, const void *src, unsigned long size)
 {
  net_t net;
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 23/31] net: Modify all rtnetlink methods to only work in the

initial namespace
Posted by ebiederm on Thu, 25 Jan 2007 19:00:25 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Before I can enable rtnetlink to work in all network namespaces
I need to be certain that something won't break. So this
patch deliberately disables all of the methods and when they
are audited this extra check can be disabled.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
net/bridge/br_netlink.c | 9 +++++++++
net/core/fib_rules.c | 7 +++++++
net/core/neighbour.c | 18 ++++++++++++++++++
net/core/rtnetlink.c | 13 +++++++++++++
net/decnet/dn_dev.c | 12 ++++++++++++
net/decnet/dn_fib.c | 8 ++++++++
net/decnet/dn_route.c | 8 ++++++++
net/decnet/dn_rules.c | 5 +++++
net/decnet/dn_table.c | 4 ++++
net/ipv4/devinet.c | 12 ++++++++++++
net/ipv4/fib_frontend.c | 12 ++++++++++++
net/ipv4/fib_rules.c | 5 +++++
net/ipv6/addrconf.c | 31 +++++++++++++++++++++++++++++++
net/ipv6/fib6_rules.c | 5 +++++
net/ipv6/ip6_fib.c | 4 ++++
net/ipv6/route.c | 12 ++++++++++++
net/sched/act_api.c | 8 ++++++++
net/sched/cls_api.c | 8 ++++++++
net/sched/sch_api.c | 20 ++++++++++++++++++++
19 files changed, 201 insertions(+), 0 deletions(-)

diff --git a/net/bridge/br_netlink.c b/net/bridge/br_netlink.c
index 119b97d..85165a1 100644
--- a/net/bridge/br_netlink.c
+++ b/net/bridge/br_netlink.c
@@ -14,6 +14,7 @@
#include <linux/rtnetlink.h>
#include <net/netlink.h>
#include <net/net_namespace.h>
+#include <net/sock.h>
#include "br_private.h"

static inline size_t br_nlmsg_size(void)
@@ -104,9 +105,13 @@ errout:
*/
static int br_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)

```
 {
+ net_t net = skb->sk->sk_net;
  struct net_device *dev;
  int idx;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  read_lock(&per_net(dev_base_lock, init_net()));
  for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next) {
   /* not a bridge port */
@@ -133,12 +138,16 @@ skip:
  */
 static int br_rtm_setlink(struct sk_buff *skb,  struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ifinfomsg *ifm;
  struct nlattr *protinfo;
  struct net_device *dev;
  struct net_bridge_port *p;
  u8 new_state;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (nlmsg_len(nlh) < sizeof(*ifm))
   return -EINVAL;

diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
index 2fa2708..00b4148 100644
--- a/net/core/fib_rules.c
+++ b/net/core/fib_rules.c
@@ -163,6 +163,9 @@ int fib_nl_newrule(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
  struct nlattr *tb[FRA_MAX+1];
  int err = -EINVAL;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (nlh->nlmsg_len < nlmsg_msg_size(sizeof(*frh)))
   goto errout;

@@ -244,12 +247,16 @@ errout:

 int fib_nl_delrule(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct fib_rule_hdr *frh = nlmsg_data(nlh);
```

```
  struct fib_rules_ops *ops = NULL;
  struct fib_rule *rule;
  struct nlattr *tb[FRA_MAX+1];
  int err = -EINVAL;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (nlh->nlmsg_len < nlmsg_msg_size(sizeof(*frh)))
   goto errout;

diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index f5d4f92..d89c6fe 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -1445,6 +1445,9 @@ int neigh_delete(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
  struct net_device *dev = NULL;
  int err = -EINVAL;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (nlmsg_len(nlh) < sizeof(*ndm))
   goto out;

@@ -1511,6 +1514,9 @@ int neigh_add(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
  struct net_device *dev = NULL;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ndm), tb, NDA_MAX, NULL);
  if (err < 0)
   goto out;
@@ -1783,11 +1789,15 @@ static struct nla_policy nl_ntbl_parm_policy[NDTPA_MAX+1]
__read_mostly = {

 int neightbl_set(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct neigh_table *tbl;
  struct ndtmsg *ndtmsg;
  struct nlattr *tb[NDTA_MAX+1];
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
```

```
+
  err = nlmsg_parse(nlh, sizeof(*ndtmsg), tb, NDTA_MAX,
      nl_neightbl_policy);
  if (err < 0)
@@ -1907,11 +1917,15 @@ errout:

 int neightbl_dump_info(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int family, tidx, nidx = 0;
  int tbl_skip = cb->args[0];
  int neigh_skip = cb->args[1];
  struct neigh_table *tbl;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  family = ((struct rtgenmsg *) nlmsg_data(cb->nlh))->rtgen_family;

  read_lock(&neigh_tbl_lock);
@@ -2030,9 +2044,13 @@ out:

 int neigh_dump_info(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  struct neigh_table *tbl;
  int t, family, s_t;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  read_lock(&neigh_tbl_lock);
  family = ((struct rtgenmsg *) nlmsg_data(cb->nlh))->rtgen_family;
  s_t = cb->args[0];
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index 5ac07a0..9be586c 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -395,6 +395,9 @@ static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
  int s_idx = cb->args[0];
  struct net_device *dev;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  read_lock(&per_net(dev_base_lock, net));
  for (dev=per_net(dev_base, net), idx=0; dev; dev = dev->next, idx++) {
   if (idx < s_idx)
```

```
@@ -429,6 +432,9 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
  struct nlattr *tb[IFLA_MAX+1];
  char ifname[IFNAMSIZ];

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFLA_MAX, ifla_policy);
  if (err < 0)
   goto errout;
@@ -602,6 +608,9 @@ static int rtnl_getlink(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
  int iw_buf_len = 0;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFLA_MAX, ifla_policy);
  if (err < 0)
   return err;
@@ -650,9 +659,13 @@ errout:

 static int rtnl_dump_all(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int idx;
  int s_idx = cb->family;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (s_idx == 0)
   s_idx = 1;
  for (idx=1; idx<NPROTO; idx++) {
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index c83c8d1..a09275b 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -648,12 +648,16 @@ static struct nla_policy dn_ifa_policy[IFA_MAX+1] __read_mostly = {

 static int dn_nl_deladdr(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct nlattr *tb[IFA_MAX+1];
  struct dn_dev *dn_db;
  struct ifaddrmsg *ifm;
  struct dn_ifaddr *ifa, **ifap;
  int err = -EADDRNOTAVAIL;
```

```
+ if (!net_eq(net, init_net()))
+  goto errout;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, dn_ifa_policy);
  if (err < 0)
   goto errout;
@@ -680,6 +684,7 @@ errout:

 static int dn_nl_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct nlattr *tb[IFA_MAX+1];
  struct net_device *dev;
  struct dn_dev *dn_db;
@@ -687,6 +692,9 @@ static int dn_nl_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg)
  struct dn_ifaddr *ifa;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, dn_ifa_policy);
  if (err < 0)
   return err;
@@ -788,11 +796,15 @@ errout:

 static int dn_nl_dump_ifaddr(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int idx, dn_idx = 0, skip_ndevs, skip_naddr;
  struct net_device *dev;
  struct dn_dev *dn_db;
  struct dn_ifaddr *ifa;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  skip_ndevs = cb->args[0];
  skip_naddr = cb->args[1];

diff --git a/net/decnet/dn_fib.c b/net/decnet/dn_fib.c
index cc2ab1f..832e1b4 100644
--- a/net/decnet/dn_fib.c
+++ b/net/decnet/dn_fib.c
@@ -503,10 +503,14 @@ static int dn_fib_check_attr(struct rtmsg *r, struct rtattr **rta)
```

```
 int dn_fib_rtm_delroute(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct dn_fib_table *tb;
  struct rtattr **rta = arg;
  struct rtmsg *r = NLMSG_DATA(nlh);

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (dn_fib_check_attr(r, rta))
   return -EINVAL;

@@ -519,10 +523,14 @@ int dn_fib_rtm_delroute(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg)

 int dn_fib_rtm_newroute(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct dn_fib_table *tb;
  struct rtattr **rta = arg;
  struct rtmsg *r = NLMSG_DATA(nlh);

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (dn_fib_check_attr(r, rta))
   return -EINVAL;

diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 9669e50..d942ea0 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -1528,6 +1528,7 @@ rtattr_failure:
  */
 int dn_cache_getroute(struct sk_buff *in_skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = in_skb->sk->sk_net;
  struct rtattr **rta = arg;
  struct rtmsg *rtm = NLMSG_DATA(nlh);
  struct dn_route *rt = NULL;
@@ -1536,6 +1537,9 @@ int dn_cache_getroute(struct sk_buff *in_skb, struct nlmsghdr *nlh,
void *arg)
  struct sk_buff *skb;
  struct flowi fl;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
```

```
+
  memset(&fl, 0, sizeof(fl));
  fl.proto = DNPROTO_NSP;

@@ -1613,10 +1617,14 @@ out_free:
  */
 int dn_cache_dump(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  struct dn_route *rt;
  int h, s_h;
  int idx, s_idx;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (NLMSG_PAYLOAD(cb->nlh, 0) < sizeof(struct rtmsg))
   return -EINVAL;
  if (!((struct rtmsg *)NLMSG_DATA(cb->nlh))->rtm_flags&RTM_F_CLONED))
diff --git a/net/decnet/dn_rules.c b/net/decnet/dn_rules.c
index e32d0c3..84eec40 100644
--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -243,6 +243,11 @@ static u32 dn_fib_rule_default_pref(void)

 int dn_fib_dump_rules(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
+
+ if (!net_eq(net, init_net()))
+  return 0;
+
  return fib_rules_dump(skb, cb, AF_DECnet);
 }

diff --git a/net/decnet/dn_table.c b/net/decnet/dn_table.c
index 13b2421..3ff151c 100644
--- a/net/decnet/dn_table.c
+++ b/net/decnet/dn_table.c
@@ -459,12 +459,16 @@ static int dn_fib_table_dump(struct dn_fib_table *tb, struct sk_buff
*skb,

 int dn_fib_dump(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  unsigned int h, s_h;
  unsigned int e = 0, s_e;
  struct dn_fib_table *tb;
```

```
  struct hlist_node *node;
  int dumped = 0;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (NLMSG_PAYLOAD(cb->nlh, 0) >= sizeof(struct rtmsg) &&
   ((struct rtmsg *)NLMSG_DATA(cb->nlh))->rtm_flags&RTM_F_CLONED)
    return dn_cache_dump(skb, cb);
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index b0d12ec..7769b1c 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -443,6 +443,7 @@ struct in_ifaddr *inet_ifa_byprefix(struct in_device *in_dev, __be32 prefix,

 static int inet_rtm_deladdr(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct nlattr *tb[IFA_MAX+1];
  struct in_device *in_dev;
  struct ifaddrmsg *ifm;
@@ -451,6 +452,9 @@ static int inet_rtm_deladdr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg

  ASSERT_RTNL();

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, ifa_ipv4_policy);
  if (err < 0)
   goto errout;
@@ -562,10 +566,14 @@ errout:

 static int inet_rtm_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct in_ifaddr *ifa;

  ASSERT_RTNL();

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  ifa = rtm_to_ifaddr(nlh);
  if (IS_ERR(ifa))
   return PTR_ERR(ifa);
@@ -1173,12 +1181,16 @@ nla_put_failure:
```

```
 static int inet_dump_ifaddr(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int idx, ip_idx;
  struct net_device *dev;
  struct in_device *in_dev;
  struct in_ifaddr *ifa;
  int s_ip_idx, s_idx = cb->args[0];

+ if (!net_eq(net, init_net()))
+  return 0;
+
  s_ip_idx = ip_idx = cb->args[1];
  read_lock(&per_net(dev_base_lock, init_net()));
  for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 449f42d..0e48fb8 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -538,10 +538,14 @@ errout:

 int inet_rtm_delroute(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct fib_config cfg;
  struct fib_table *tb;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = rtm_to_fib_config(skb, nlh, &cfg);
  if (err < 0)
   goto errout;
@@ -559,10 +563,14 @@ errout:

 int inet_rtm_newroute(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct fib_config cfg;
  struct fib_table *tb;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = rtm_to_fib_config(skb, nlh, &cfg);
```

```
  if (err < 0)
   goto errout;
@@ -580,12 +588,16 @@ errout:

 int inet_dump_fib(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  unsigned int h, s_h;
  unsigned int e = 0, s_e;
  struct fib_table *tb;
  struct hlist_node *node;
  int dumped = 0;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (nlmsg_len(cb->nlh) >= sizeof(struct rtmsg) &&
      ((struct rtmsg *) nlmsg_data(cb->nlh))->rtm_flags & RTM_F_CLONED)
   return ip_rt_dump(skb, cb);
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index b837c33..f2c50e0 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -279,6 +279,11 @@ nla_put_failure:

 int fib4_rules_dump(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
+
+ if (!net_eq(net, init_net()))
+  return 0;
+
  return fib_rules_dump(skb, cb, AF_INET);
 }

diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 7afe698..83b7312 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -2951,11 +2951,15 @@ static struct nla_policy ifa_ipv6_policy[IFA_MAX+1] __read_mostly
= {
 static int
 inet6_rtm_deladdr(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ifaddrmsg *ifm;
  struct nlattr *tb[IFA_MAX+1];
  struct in6_addr *pfx;
```

```
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, ifa_ipv6_policy);
  if (err < 0)
   return err;
@@ -3003,6 +3007,7 @@ static int inet6_addr_modify(struct inet6_ifaddr *ifp, u8 ifa_flags,
 static int
 inet6_rtm_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct ifaddrmsg *ifm;
  struct nlattr *tb[IFA_MAX+1];
  struct in6_addr *pfx;
@@ -3012,6 +3017,9 @@ inet6_rtm_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg)
  u8 ifa_flags;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, ifa_ipv6_policy);
  if (err < 0)
   return err;
@@ -3278,26 +3286,42 @@ done:

 static int inet6_dump_ifaddr(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  enum addr_type_t type = UNICAST_ADDR;
+
+ if (!net_eq(net, init_net()))
+  return 0;
+
  return inet6_dump_addr(skb, cb, type);
 }

 static int inet6_dump_ifmcaddr(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  enum addr_type_t type = MULTICAST_ADDR;
+
+ if (!net_eq(net, init_net()))
+  return 0;
+
```

```
  return inet6_dump_addr(skb, cb, type);
 }


 static int inet6_dump_ifacaddr(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  enum addr_type_t type = ANYCAST_ADDR;
+
+ if (!net_eq(net, init_net()))
+  return 0;
+
  return inet6_dump_addr(skb, cb, type);
 }

 static int inet6_rtm_getaddr(struct sk_buff *in_skb, struct nlmsghdr* nlh,
        void *arg)
 {
+ net_t net = in_skb->sk->sk_net;
  struct ifaddrmsg *ifm;
  struct nlattr *tb[IFA_MAX+1];
  struct in6_addr *addr = NULL;
@@ -3306,6 +3330,9 @@ static int inet6_rtm_getaddr(struct sk_buff *in_skb, struct nlmsghdr*
nlh,
  struct sk_buff *skb;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, ifa_ipv6_policy);
  if (err < 0)
   goto errout;
@@ -3472,11 +3499,15 @@ nla_put_failure:

 static int inet6_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int idx, err;
  int s_idx = cb->args[0];
  struct net_device *dev;
  struct inet6_dev *idev;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  read_lock(&per_net(dev_base_lock, init_net()));
  for (dev=per_net(dev_base, init_net()), idx=0; dev; dev = dev->next, idx++) {
```

```
   if (idx < s_idx)
diff --git a/net/ipv6/fib6_rules.c b/net/ipv6/fib6_rules.c
index 0862809..80d6de6 100644
--- a/net/ipv6/fib6_rules.c
+++ b/net/ipv6/fib6_rules.c
@@ -223,6 +223,11 @@ nla_put_failure:

 int fib6_rules_dump(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
+
+ if (!net_eq(net, init_net()))
+  return 0;
+
  return fib_rules_dump(skb, cb, AF_INET6);
 }

diff --git a/net/ipv6/ip6_fib.c b/net/ipv6/ip6_fib.c
index 96d8310..97814ed 100644
--- a/net/ipv6/ip6_fib.c
+++ b/net/ipv6/ip6_fib.c
@@ -362,6 +362,7 @@ end:

 int inet6_dump_fib(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  unsigned int h, s_h;
  unsigned int e = 0, s_e;
  struct rt6_rtnl_dump_arg arg;
@@ -370,6 +371,9 @@ int inet6_dump_fib(struct sk_buff *skb, struct netlink_callback *cb)
  struct hlist_node *node;
  int res = 0;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  s_h = cb->args[0];
  s_e = cb->args[1];

diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 4519006..02fd8ae 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -1985,9 +1985,13 @@ errout:

 int inet6_rtm_delroute(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
```

```
  struct fib6_config cfg;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = rtm_to_fib6_config(skb, nlh, &cfg);
  if (err < 0)
   return err;
@@ -1997,9 +2001,13 @@ int inet6_rtm_delroute(struct sk_buff *skb, struct nlmsghdr* nlh, void
*arg)

 int inet6_rtm_newroute(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct fib6_config cfg;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = rtm_to_fib6_config(skb, nlh, &cfg);
  if (err < 0)
   return err;
@@ -2132,6 +2140,7 @@ int rt6_dump_route(struct rt6_info *rt, void *p_arg)

 int inet6_rtm_getroute(struct sk_buff *in_skb, struct nlmsghdr* nlh, void *arg)
 {
+ net_t net = in_skb->sk->sk_net;
  struct nlattr *tb[RTA_MAX+1];
  struct rt6_info *rt;
  struct sk_buff *skb;
@@ -2139,6 +2148,9 @@ int inet6_rtm_getroute(struct sk_buff *in_skb, struct nlmsghdr* nlh,
void *arg)
  struct flowi fl;
  int err, iif = 0;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  err = nlmsg_parse(nlh, sizeof(*rtm), tb, RTA_MAX, rtm_ipv6_policy);
  if (err < 0)
   goto errout;
diff --git a/net/sched/act_api.c b/net/sched/act_api.c
index 835070e..18d8f68 100644
--- a/net/sched/act_api.c
+++ b/net/sched/act_api.c
@@ -942,10 +942,14 @@ done:
```

```
 static int tc_ctl_action(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct rtattr **tca = arg;
  u32 pid = skb ? NETLINK_CB(skb).pid : 0;
  int ret = 0, ovr = 0;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if (tca[TCA_ACT_TAB-1] == NULL) {
   printk("tc_ctl_action: received NO action attribs\n");
   return -EINVAL;
@@ -1015,6 +1019,7 @@ find_dump_kind(struct nlmsghdr *n)
 static int
 tc_dump_action(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  struct nlmsghdr *nlh;
  unsigned char *b = skb->tail;
  struct rtattr *x;
@@ -1024,6 +1029,9 @@ tc_dump_action(struct sk_buff *skb, struct netlink_callback *cb)
  struct tcamsg *t = (struct tcamsg *) NLMSG_DATA(cb->nlh);
  struct rtattr *kind = find_dump_kind(cb->nlh);

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (kind == NULL) {
   printk("tc_dump_action: action bad kind\n");
   return 0;
diff --git a/net/sched/cls_api.c b/net/sched/cls_api.c
index 19935f9..09a3ec8 100644
--- a/net/sched/cls_api.c
+++ b/net/sched/cls_api.c
@@ -129,6 +129,7 @@ static __inline__ u32 tcf_auto_prio(struct tcf_proto *tp)

 static int tc_ctl_tfilter(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct rtattr **tca;
  struct tcmsg *t;
  u32 protocol;
@@ -145,6 +146,9 @@ static int tc_ctl_tfilter(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
  unsigned long fh;
  int err;
```

```
+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
 replay:
  tca = arg;
  t = NLMSG_DATA(n);
@@ -385,6 +389,7 @@ static int tcf_node_dump(struct tcf_proto *tp, unsigned long n, struct
tcf_walke

 static int tc_dump_tfilter(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int t;
  int s_t;
  struct net_device *dev;
@@ -395,6 +400,9 @@ static int tc_dump_tfilter(struct sk_buff *skb, struct netlink_callback *cb)
  struct Qdisc_class_ops *cops;
  struct tcf_dump_args arg;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (cb->nlh->nlmsg_len < NLMSG_LENGTH(sizeof(*tcm)))
   return skb->len;
  if ((dev = dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index 912e8e1..7e33f73 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -578,6 +578,7 @@ check_loop_fn(struct Qdisc *q, unsigned long cl, struct qdisc_walker *w)

 static int tc_get_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct tcmsg *tcm = NLMSG_DATA(n);
  struct rtattr **tca = arg;
  struct net_device *dev;
@@ -586,6 +587,9 @@ static int tc_get_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
  struct Qdisc *p = NULL;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
  if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return -ENODEV;

@@ -639,6 +643,7 @@ static int tc_get_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
```

```
 static int tc_modify_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct tcmsg *tcm;
  struct rtattr **tca;
  struct net_device *dev;
@@ -646,6 +651,9 @@ static int tc_modify_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void
*arg)
  struct Qdisc *q, *p;
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
+
 replay:
  /* Reinit, just in case something touches this. */
  tcm = NLMSG_DATA(n);
@@ -851,11 +859,15 @@ err_out:

 static int tc_dump_qdisc(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int idx, q_idx;
  int s_idx, s_q_idx;
  struct net_device *dev;
  struct Qdisc *q;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  s_idx = cb->args[0];
  s_q_idx = q_idx = cb->args[1];
  read_lock(&per_net(dev_base_lock, init_net()));
@@ -900,6 +912,7 @@ done:

 static int tc_ctl_tclass(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
 {
+ net_t net = skb->sk->sk_net;
  struct tcmsg *tcm = NLMSG_DATA(n);
  struct rtattr **tca = arg;
  struct net_device *dev;
@@ -912,6 +925,9 @@ static int tc_ctl_tclass(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
  u32 qid = TC_H_MAJ(clid);
  int err;

+ if (!net_eq(net, init_net()))
+  return -EINVAL;
```

```
+
  if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
   return -ENODEV;

@@ -1086,6 +1102,7 @@ static int qdisc_class_dump(struct Qdisc *q, unsigned long cl, struct
qdisc_walk

 static int tc_dump_tclass(struct sk_buff *skb, struct netlink_callback *cb)
 {
+ net_t net = skb->sk->sk_net;
  int t;
  int s_t;
  struct net_device *dev;
@@ -1093,6 +1110,9 @@ static int tc_dump_tclass(struct sk_buff *skb, struct netlink_callback
*cb)
  struct tcmsg *tcm = (struct tcmsg*)NLMSG_DATA(cb->nlh);
  struct qdisc_dump_args arg;

+ if (!net_eq(net, init_net()))
+  return 0;
+
  if (cb->nlh->nlmsg_len < NLMSG_LENGTH(sizeof(*tcm)))
   return 0;
  if ((dev = dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
--
1.4.4.1.g278f
```


Subject: [PATCH RFC 24/31] net: Make rtnetlink network namespace aware
Posted by ebiederm on Thu, 25 Jan 2007 19:00:26 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

After this patch none of the netlink callback support anything
except the initial network namespace but the rtnetlink infrastructure
now handles multiple network namespaces.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 include/linux/rtnetlink.h |    8 ++--
 net/bridge/br_netlink.c   |    4 +-
 net/core/fib_rules.c      |    4 +-
```

```
 net/core/neighbour.c      |    4 +-
 net/core/rtnetlink.c      |   74 ++++++++++++++++++++++++++++++++---------
 net/core/wireless.c       |    5 ++-
 net/decnet/dn_dev.c       |    4 +-
 net/decnet/dn_route.c     |    2 +-
 net/decnet/dn_table.c     |    4 +-
 net/ipv4/devinet.c        |    4 +-
 net/ipv4/fib_semantics.c  |    4 +-
 net/ipv4/ipmr.c           |    4 +-
 net/ipv4/route.c          |    2 +-
 net/ipv6/addrconf.c       |   14 ++++++----
 net/ipv6/route.c          |    6 ++--
 net/sched/cls_api.c       |    2 +-
 net/sched/sch_api.c       |    4 +-
 17 files changed, 98 insertions(+), 51 deletions(-)

diff --git a/include/linux/rtnetlink.h b/include/linux/rtnetlink.h
index 4a629ea..6c8281d 100644
--- a/include/linux/rtnetlink.h
+++ b/include/linux/rtnetlink.h
@@ -581,11 +581,11 @@ struct rtnetlink_link
 };

 extern struct rtnetlink_link * rtnetlink_links[NPROTO];
-extern int rtnetlink_send(struct sk_buff *skb, u32 pid, u32 group, int echo);
-extern int rtnl_unicast(struct sk_buff *skb, u32 pid);
-extern int rtnl_notify(struct sk_buff *skb, u32 pid, u32 group,
+extern int rtnetlink_send(struct sk_buff *skb, net_t net, u32 pid, u32 group, int echo);
+extern int rtnl_unicast(struct sk_buff *skb, net_t net, u32 pid);
+extern int rtnl_notify(struct sk_buff *skb, net_t net, u32 pid, u32 group,
                        struct nlmsghdr *nlh, gfp_t flags);
-extern void rtnl_set_sk_err(u32 group, int error);
+extern void rtnl_set_sk_err(net_t net, u32 group, int error);
 extern int rtnetlink_put_metrics(struct sk_buff *skb, u32 *metrics);
 extern int rtnl_put_cacheinfo(struct sk_buff *skb, struct dst_entry *dst,
                               u32 id, u32 ts, u32 tsage, long expires,
diff --git a/net/bridge/br_netlink.c b/net/bridge/br_netlink.c
index 85165a1..372fb18 100644
--- a/net/bridge/br_netlink.c
+++ b/net/bridge/br_netlink.c
@@ -94,10 +94,10 @@ void br_ifinfo_notify(int event, struct net_bridge_port *port)
   /* failure implies BUG in br_nlmsg_size() */
   BUG_ON(err < 0);

-  err = rtnl_notify(skb, 0, RTNLGRP_LINK, NULL, GFP_ATOMIC);
+  err = rtnl_notify(skb, init_net(), 0, RTNLGRP_LINK, NULL, GFP_ATOMIC);
 errout:
   if (err < 0)
```

```
-  rtnl_set_sk_err(RTNLGRP_LINK, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_LINK, err);
 }

 /*
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
index 00b4148..5f65973 100644
--- a/net/core/fib_rules.c
+++ b/net/core/fib_rules.c
@@ -418,10 +418,10 @@ static void notify_rule_change(int event, struct fib_rule *rule,
  /* failure implies BUG in fib_rule_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, pid, ops->nlgroup, nlh, GFP_KERNEL);
+ err = rtnl_notify(skb, init_net(), pid, ops->nlgroup, nlh, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(ops->nlgroup, err);
+  rtnl_set_sk_err(init_net(), ops->nlgroup, err);
 }

 static void attach_rules(struct list_head *rules, struct net_device *dev)
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index d89c6fe..6f61207 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -2453,10 +2453,10 @@ static void __neigh_notify(struct neighbour *n, int type, int flags)
  /* failure implies BUG in neigh_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, 0, RTNLGRP_NEIGH, NULL, GFP_ATOMIC);
+ err = rtnl_notify(skb, init_net(), 0, RTNLGRP_NEIGH, NULL, GFP_ATOMIC);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_NEIGH, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_NEIGH, err);
 }

 void neigh_app_ns(struct neighbour *n)
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index 9be586c..29a81bf 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -58,7 +58,7 @@
 #endif /* CONFIG_NET_WIRELESS_RTNETLINK */

 static DEFINE_MUTEX(rtnl_mutex);
-static struct sock *rtnl;
```

```
+static DEFINE_PER_NET(struct sock *, rtnl);

 void rtnl_lock(void)
 {
@@ -72,9 +72,17 @@ void __rtnl_unlock(void)

 void rtnl_unlock(void)
 {
+ net_t net;
  mutex_unlock(&rtnl_mutex);
- if (rtnl && rtnl->sk_receive_queue.qlen)
-  rtnl->sk_data_ready(rtnl, 0);
+
+ net_lock();
+ for_each_net(net) {
+  struct sock *rtnl = per_net(rtnl, net);
+  if (rtnl && rtnl->sk_receive_queue.qlen)
+   rtnl->sk_data_ready(rtnl, 0);
+ }
+ net_unlock();
+
  netdev_run_todo();
 }

@@ -151,8 +159,9 @@ size_t rtattr_strlcpy(char *dest, const struct rtattr *rta, size_t size)
  return ret;
 }

-int rtnetlink_send(struct sk_buff *skb, u32 pid, unsigned group, int echo)
+int rtnetlink_send(struct sk_buff *skb, net_t net, u32 pid, unsigned group, int echo)
 {
+ struct sock *rtnl = per_net(rtnl, net);
  int err = 0;

  NETLINK_CB(skb).dst_group = group;
@@ -164,14 +173,17 @@ int rtnetlink_send(struct sk_buff *skb, u32 pid, unsigned group, int
echo)
  return err;
 }

-int rtnl_unicast(struct sk_buff *skb, u32 pid)
+int rtnl_unicast(struct sk_buff *skb, net_t net, u32 pid)
 {
+ struct sock *rtnl = per_net(rtnl, net);
+
  return nlmsg_unicast(rtnl, skb, pid);
 }
```

```
-int rtnl_notify(struct sk_buff *skb, u32 pid, u32 group,
+int rtnl_notify(struct sk_buff *skb, net_t net, u32 pid, u32 group,
    struct nlmsghdr *nlh, gfp_t flags)
 {
+ struct sock *rtnl = per_net(rtnl, net);
  int report = 0;

  if (nlh)
@@ -180,8 +192,10 @@ int rtnl_notify(struct sk_buff *skb, u32 pid, u32 group,
  return nlmsg_notify(rtnl, skb, pid, group, report, flags);
 }

-void rtnl_set_sk_err(u32 group, int error)
+void rtnl_set_sk_err(net_t net, u32 group, int error)
 {
+ struct sock *rtnl = per_net(rtnl, net);
+
  netlink_set_err(rtnl, 0, group, error);
 }

@@ -649,7 +663,7 @@ static int rtnl_getlink(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
  /* failure impilies BUG in if_nlmsg_size or wireless_rtnetlink_get */
  BUG_ON(err < 0);

- err = rtnl_unicast(nskb, NETLINK_CB(skb).pid);
+ err = rtnl_unicast(nskb, net, NETLINK_CB(skb).pid);
 errout:
  kfree(iw_buf);
  dev_put(dev);
@@ -698,10 +712,10 @@ void rtmsg_ifinfo(int type, struct net_device *dev, unsigned change)
  /* failure implies BUG in if_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, 0, RTNLGRP_LINK, NULL, GFP_KERNEL);
+ err = rtnl_notify(skb, init_net(), 0, RTNLGRP_LINK, NULL, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_LINK, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_LINK, err);
 }

 /* Protected by RTNL sempahore.  */
@@ -713,6 +727,7 @@ static int rtattr_max;
 static __inline__ int
 rtnetlink_rcv_msg(struct sk_buff *skb, struct nlmsghdr *nlh, int *errp)
 {
+ net_t net = skb->sk->sk_net;
  struct rtnetlink_link *link;
```

```
  struct rtnetlink_link *link_tab;
  int sz_idx, kind;
@@ -767,7 +782,7 @@ rtnetlink_rcv_msg(struct sk_buff *skb, struct nlmsghdr *nlh, int *errp)
   if (link->dumpit == NULL)
    goto err_inval;

-  if ((*errp = netlink_dump_start(rtnl, skb, nlh,
+  if ((*errp = netlink_dump_start(per_net(rtnl, net), skb, nlh,
        link->dumpit, NULL)) != 0) {
    return -1;
   }
@@ -875,6 +890,36 @@ static struct notifier_block rtnetlink_dev_notifier = {
  .notifier_call = rtnetlink_event,
 };

+
+static int rtnetlink_net_init(net_t net)
+{
+ struct sock *sk;
+ sk = netlink_kernel_create(net, NETLINK_ROUTE, RTNLGRP_MAX,
+        rtnetlink_rcv, THIS_MODULE);
+ if (!sk)
+  return -ENOMEM;
+
+ /* Don't hold an extra reference on the namespace */
+ put_net(sk->sk_net);
+ per_net(rtnl, net) = sk;
+ return 0;
+}
+
+static void rtnetlink_net_exit(net_t net)
+{
+ /* At the last minute lie and say this is a socket for the
+  * initial network namespace.  So the socket will be safe to
+  * free.
+  */
+ per_net(rtnl, net)->sk_net = get_net(init_net());
+ sock_put(per_net(rtnl, net));
+}
+
+static struct pernet_operations rtnetlink_net_ops = {
+ .init = rtnetlink_net_init,
+ .exit = rtnetlink_net_exit,
+};
+
 void __init rtnetlink_init(void)
 {
  int i;
```

```
@@ -887,10 +932,9 @@ void __init rtnetlink_init(void)
  if (!rta_buf)
   panic("rtnetlink_init: cannot allocate rta_buf\n");

- rtnl = netlink_kernel_create(init_net(), NETLINK_ROUTE, RTNLGRP_MAX,
-         rtnetlink_rcv, THIS_MODULE);
- if (rtnl == NULL)
-  panic("rtnetlink_init: cannot initialize rtnetlink\n");
+ if (register_pernet_subsys(&rtnetlink_net_ops))
+    panic("rtnetlink_init: cannot initialize rtnetlink\n");
+
  netlink_set_nonroot(NETLINK_ROUTE, NL_NONROOT_RECV);
  register_netdevice_notifier(&rtnetlink_dev_notifier);
  rtnetlink_links[PF_UNSPEC] = link_rtnetlink_table;
diff --git a/net/core/wireless.c b/net/core/wireless.c
index d1418bf..9036359 100644
--- a/net/core/wireless.c
+++ b/net/core/wireless.c
@@ -1935,7 +1935,7 @@ static void wireless_nlevent_process(unsigned long data)
  struct sk_buff *skb;

  while ((skb = skb_dequeue(&wireless_nlevent_queue)))
-  rtnl_notify(skb, 0, RTNLGRP_LINK, NULL, GFP_ATOMIC);
+  rtnl_notify(skb, init_net(), 0, RTNLGRP_LINK, NULL, GFP_ATOMIC);
 }

 static DECLARE_TASKLET(wireless_nlevent_tasklet, wireless_nlevent_process, 0);
@@ -1992,6 +1992,9 @@ static inline void rtmsg_iwinfo(struct net_device * dev,
  struct sk_buff *skb;
  int size = NLMSG_GOODSIZE;

+ if (!net_eq(dev->nd_net, init_net()))
+  return;
+
  skb = alloc_skb(size, GFP_ATOMIC);
  if (!skb)
   return;
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index a09275b..bad972d 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -788,10 +788,10 @@ static void dn_ifaddr_notify(int event, struct dn_ifaddr *ifa)
  /* failure implies BUG in dn_ifaddr_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, 0, RTNLGRP_DECnet_IFADDR, NULL, GFP_KERNEL);
+ err = rtnl_notify(skb, init_net(), 0, RTNLGRP_DECnet_IFADDR, NULL, GFP_KERNEL);
 errout:
```

```
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_DECnet_IFADDR, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_DECnet_IFADDR, err);
 }

 static int dn_nl_dump_ifaddr(struct sk_buff *skb, struct netlink_callback *cb)
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index d942ea0..4b353d4 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -1604,7 +1604,7 @@ int dn_cache_getroute(struct sk_buff *in_skb, struct nlmsghdr *nlh,
void *arg)
   goto out_free;
  }

- return rtnl_unicast(skb, NETLINK_CB(in_skb).pid);
+ return rtnl_unicast(skb, init_net(), NETLINK_CB(in_skb).pid);

 out_free:
  kfree_skb(skb);
diff --git a/net/decnet/dn_table.c b/net/decnet/dn_table.c
index 3ff151c..4090ab5 100644
--- a/net/decnet/dn_table.c
+++ b/net/decnet/dn_table.c
@@ -371,10 +371,10 @@ static void dn_rtmsg_fib(int event, struct dn_fib_node *f, int z, u32
tb_id,
  /* failure implies BUG in dn_fib_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, pid, RTNLGRP_DECnet_ROUTE, nlh, GFP_KERNEL);
+ err = rtnl_notify(skb, init_net(), pid, RTNLGRP_DECnet_ROUTE, nlh, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_DECnet_ROUTE, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_DECnet_ROUTE, err);
 }

 static __inline__ int dn_hash_dump_bucket(struct sk_buff *skb,
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 7769b1c..59acce2 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1241,10 +1241,10 @@ static void rtmsg_ifa(int event, struct in_ifaddr* ifa, struct nlmsghdr
*nlh,
  /* failure implies BUG in inet_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, pid, RTNLGRP_IPV4_IFADDR, nlh, GFP_KERNEL);
```

```
+ err = rtnl_notify(skb, init_net(), pid, RTNLGRP_IPV4_IFADDR, nlh, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_IPV4_IFADDR, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_IPV4_IFADDR, err);
 }

 static struct rtnetlink_link inet_rtnetlink_table[RTM_NR_MSGTYPES] = {
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index 76218e5..8c64334 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -317,11 +317,11 @@ void rtmsg_fib(int event, __be32 key, struct fib_alias *fa,
  /* failure implies BUG in fib_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, info->pid, RTNLGRP_IPV4_ROUTE,
+ err = rtnl_notify(skb, init_net(), info->pid, RTNLGRP_IPV4_ROUTE,
       info->nlh, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_IPV4_ROUTE, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_IPV4_ROUTE, err);
 }

 /* Return the first fib alias matching TOS with
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index d2e7e55..15e0eb4 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -314,7 +314,7 @@ static void ipmr_destroy_unres(struct mfc_cache *c)
    e->error = -ETIMEDOUT;
    memset(&e->msg, 0, sizeof(e->msg));

-   rtnl_unicast(skb, NETLINK_CB(skb).pid);
+   rtnl_unicast(skb, init_net(), NETLINK_CB(skb).pid);
   } else
    kfree_skb(skb);
  }
@@ -527,7 +527,7 @@ static void ipmr_cache_resolve(struct mfc_cache *uc, struct mfc_cache
*c)
     memset(&e->msg, 0, sizeof(e->msg));
    }

-   rtnl_unicast(skb, NETLINK_CB(skb).pid);
+   rtnl_unicast(skb, init_net(), NETLINK_CB(skb).pid);
   } else
    ip_mr_forward(skb, c, 0);
```

```
 }
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 509bfb1..5f8592e 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2802,7 +2802,7 @@ int inet_rtm_getroute(struct sk_buff *in_skb, struct nlmsghdr* nlh, void
*arg)
  if (err <= 0)
   goto errout_free;

- err = rtnl_unicast(skb, NETLINK_CB(in_skb).pid);
+ err = rtnl_unicast(skb, init_net(), NETLINK_CB(in_skb).pid);
 errout:
  return err;

diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 83b7312..597bc10 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -3362,7 +3362,7 @@ static int inet6_rtm_getaddr(struct sk_buff *in_skb, struct nlmsghdr*
nlh,
  /* failure implies BUG in inet6_ifaddr_msgsize() */
  BUG_ON(err < 0);

- err = rtnl_unicast(skb, NETLINK_CB(in_skb).pid);
+ err = rtnl_unicast(skb, init_net(), NETLINK_CB(in_skb).pid);
 errout_ifa:
  in6_ifa_put(ifa);
 errout:
@@ -3382,10 +3382,10 @@ static void inet6_ifa_notify(int event, struct inet6_ifaddr *ifa)
  /* failure implies BUG in inet6_ifaddr_msgsize() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, 0, RTNLGRP_IPV6_IFADDR, NULL, GFP_ATOMIC);
+ err = rtnl_notify(skb, init_net(), 0, RTNLGRP_IPV6_IFADDR, NULL, GFP_ATOMIC);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_IPV6_IFADDR, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_IPV6_IFADDR, err);
 }

 static void inline ipv6_store_devconf(struct ipv6_devconf *cnf,
@@ -3539,10 +3539,10 @@ void inet6_ifinfo_notify(int event, struct inet6_dev *idev)
  /* failure implies BUG in inet6_if_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, 0, RTNLGRP_IPV6_IFADDR, NULL, GFP_ATOMIC);
+ err = rtnl_notify(skb, init_net(), 0, RTNLGRP_IPV6_IFADDR, NULL, GFP_ATOMIC);
```

```
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_IPV6_IFADDR, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_IPV6_IFADDR, err);
 }

 static inline size_t inet6_prefix_nlmsg_size(void)
@@ -3604,10 +3604,10 @@ static void inet6_prefix_notify(int event, struct inet6_dev *idev,
  /* failure implies BUG in inet6_prefix_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, 0, RTNLGRP_IPV6_PREFIX, NULL, GFP_ATOMIC);
+ err = rtnl_notify(skb, init_net(), 0, RTNLGRP_IPV6_PREFIX, NULL, GFP_ATOMIC);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_IPV6_PREFIX, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_IPV6_PREFIX, err);
 }

 static struct rtnetlink_link inet6_rtnetlink_table[RTM_NR_MSGTYPES] = {
diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 02fd8ae..cf568f6 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -2210,7 +2210,7 @@ int inet6_rtm_getroute(struct sk_buff *in_skb, struct nlmsghdr* nlh,
void *arg)
   goto errout;
  }

- err = rtnl_unicast(skb, NETLINK_CB(in_skb).pid);
+ err = rtnl_unicast(skb, init_net(), NETLINK_CB(in_skb).pid);
 errout:
  return err;
 }
@@ -2237,10 +2237,10 @@ void inet6_rt_notify(int event, struct rt6_info *rt, struct nl_info *info)
  /* failure implies BUG in rt6_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, pid, RTNLGRP_IPV6_ROUTE, nlh, gfp_any());
+ err = rtnl_notify(skb, init_net(), pid, RTNLGRP_IPV6_ROUTE, nlh, gfp_any());
 errout:
  if (err < 0)
-  rtnl_set_sk_err(RTNLGRP_IPV6_ROUTE, err);
+  rtnl_set_sk_err(init_net(), RTNLGRP_IPV6_ROUTE, err);
 }

 /*
diff --git a/net/sched/cls_api.c b/net/sched/cls_api.c
```

```
index 09a3ec8..c69b4fc 100644
--- a/net/sched/cls_api.c
+++ b/net/sched/cls_api.c
@@ -369,7 +369,7 @@ static int tfilter_notify(struct sk_buff *oskb, struct nlmsghdr *n,
   return -EINVAL;
  }

- return rtnetlink_send(skb, pid, RTNLGRP_TC, n->nlmsg_flags&NLM_F_ECHO);
+ return rtnetlink_send(skb, init_net(), pid, RTNLGRP_TC, n->nlmsg_flags&NLM_F_ECHO);
 }

 struct tcf_dump_args
diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index 7e33f73..ae55988 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -850,7 +850,7 @@ static int qdisc_notify(struct sk_buff *oskb, struct nlmsghdr *n,
  }

  if (skb->len)
-  return rtnetlink_send(skb, pid, RTNLGRP_TC, n->nlmsg_flags&NLM_F_ECHO);
+  return rtnetlink_send(skb, init_net(), pid, RTNLGRP_TC, n->nlmsg_flags&NLM_F_ECHO);

 err_out:
  kfree_skb(skb);
@@ -1082,7 +1082,7 @@ static int tclass_notify(struct sk_buff *oskb, struct nlmsghdr *n,
   return -EINVAL;
  }

- return rtnetlink_send(skb, pid, RTNLGRP_TC, n->nlmsg_flags&NLM_F_ECHO);
+ return rtnetlink_send(skb, init_net(), pid, RTNLGRP_TC, n->nlmsg_flags&NLM_F_ECHO);
 }

 struct qdisc_dump_args
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 25/31] net: Make wireless netlink event generation handle multiple network namespaces
Posted by ebiederm on Thu, 25 Jan 2007 19:00:27 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/core/wireless.c | 15 ++++++++++-----
 1 files changed, 10 insertions(+), 5 deletions(-)

```
diff --git a/net/core/wireless.c b/net/core/wireless.c
index 9036359..d534617 100644
--- a/net/core/wireless.c
+++ b/net/core/wireless.c
@@ -1934,8 +1934,13 @@ static void wireless_nlevent_process(unsigned long data)
 {
  struct sk_buff *skb;

- while ((skb = skb_dequeue(&wireless_nlevent_queue)))
-  rtnl_notify(skb, init_net(), 0, RTNLGRP_LINK, NULL, GFP_ATOMIC);
+ while ((skb = skb_dequeue(&wireless_nlevent_queue))) {
+  struct net_device *dev = skb->dev;
+  net_t net = dev->nd_net;
+  skb->dev = NULL;
+  rtnl_notify(skb, net, 0, RTNLGRP_LINK, NULL, GFP_ATOMIC);
+  dev_put(dev);
+ }
 }

 static DECLARE_TASKLET(wireless_nlevent_tasklet, wireless_nlevent_process, 0);
@@ -1992,9 +1997,6 @@ static inline void rtmsg_iwinfo(struct net_device * dev,
  struct sk_buff *skb;
  int size = NLMSG_GOODSIZE;

- if (!net_eq(dev->nd_net, init_net()))
-  return;
-
  skb = alloc_skb(size, GFP_ATOMIC);
  if (!skb)
   return;
@@ -2004,6 +2006,9 @@ static inline void rtmsg_iwinfo(struct net_device * dev,
   kfree_skb(skb);
   return;
  }
+ /* Remember the device until we are in process context */
+ dev_hold(dev);
+ skb->dev = dev;
  NETLINK_CB(skb).dst_group = RTNLGRP_LINK;
  skb_queue_tail(&wireless_nlevent_queue, skb);
  tasklet_schedule(&wireless_nlevent_tasklet);
--
```

1.4.4.1.g278f

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 26/31] net: Make the netlink methods in rtnetlink handle multiple network namespaces
Posted by ebiederm on Thu, 25 Jan 2007 19:00:28 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

It turns out after a quick audit that except for removing the checks
there is really nothing to do here.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/core/rtnetlink.c | 21 +++------------------
 1 files changed, 3 insertions(+), 18 deletions(-)

diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index 29a81bf..0a42258 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -409,9 +409,6 @@ static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
  int s_idx = cb->args[0];
  struct net_device *dev;

- if (!net_eq(net, init_net()))
- return 0;
-
  read_lock(&per_net(dev_base_lock, net));
  for (dev=per_net(dev_base, net), idx=0; dev; dev = dev->next, idx++) {
   if (idx < s_idx)
@@ -446,9 +443,6 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
  struct nlattr *tb[IFLA_MAX+1];
  char ifname[IFNAMSIZ];

- if (!net_eq(net, init_net()))
- return -EINVAL;
-
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFLA_MAX, ifla_policy);
  if (err < 0)
   goto errout;
@@ -622,9 +616,6 @@ static int rtnl_getlink(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)

```
  int iw_buf_len = 0;
  int err;

- if (!net_eq(net, init_net()))
-  return -EINVAL;
-
  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFLA_MAX, ifla_policy);
  if (err < 0)
   return err;
@@ -673,13 +664,9 @@ errout:

 static int rtnl_dump_all(struct sk_buff *skb, struct netlink_callback *cb)
 {
- net_t net = skb->sk->sk_net;
  int idx;
  int s_idx = cb->family;

- if (!net_eq(net, init_net()))
-  return 0;
-
  if (s_idx == 0)
   s_idx = 1;
  for (idx=1; idx<NPROTO; idx++) {
@@ -701,6 +688,7 @@ static int rtnl_dump_all(struct sk_buff *skb, struct netlink_callback *cb)

 void rtmsg_ifinfo(int type, struct net_device *dev, unsigned change)
 {
+ net_t net = dev->nd_net;
  struct sk_buff *skb;
  int err = -ENOBUFS;

@@ -712,10 +700,10 @@ void rtmsg_ifinfo(int type, struct net_device *dev, unsigned change)
  /* failure implies BUG in if_nlmsg_size() */
  BUG_ON(err < 0);

- err = rtnl_notify(skb, init_net(), 0, RTNLGRP_LINK, NULL, GFP_KERNEL);
+ err = rtnl_notify(skb, net, 0, RTNLGRP_LINK, NULL, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(init_net(), RTNLGRP_LINK, err);
+  rtnl_set_sk_err(net, RTNLGRP_LINK, err);
 }

 /* Protected by RTNL sempahore.  */
@@ -862,9 +850,6 @@ static int rtnetlink_event(struct notifier_block *this, unsigned long event,
voi
 {
  struct net_device *dev = ptr;
```

- if (!net_eq(dev->nd_net, init_net()))
- return NOTIFY_DONE;
-
  switch (event) {
  case NETDEV_UNREGISTER:
   rtmsg_ifinfo(RTM_DELLINK, dev, ~0U);
--
1.4.4.1.g278f

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 27/31] net: Make the xfrm sysctls per network namespace.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:29 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

In particalure I moved:
/proc/sys/net/core/xfrm_aevent_etime
/proc/sys/net/core/xfrm_aevent_rseqth

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/net/xfrm.h | 4 ++--
 net/core/sysctl_net_core.c | 37 +++++++++++++++++++-------------------
 net/xfrm/xfrm_state.c | 8 ++++----
 net/xfrm/xfrm_user.c | 10 ++++++----
 4 files changed, 30 insertions(+), 29 deletions(-)

diff --git a/include/net/xfrm.h b/include/net/xfrm.h
index e476541..9b2e727 100644
--- a/include/net/xfrm.h
+++ b/include/net/xfrm.h
@@ -24,8 +24,8 @@
  MODULE_ALIAS("xfrm-mode-" __stringify(family) "-" __stringify(encap))

 extern struct sock *xfrm_nl;
-extern u32 sysctl_xfrm_aevent_etime;
-extern u32 sysctl_xfrm_aevent_rseqth;
+DECLARE_PER_NET(u32, sysctl_xfrm_aevent_etime);
+DECLARE_PER_NET(u32, sysctl_xfrm_aevent_rseqth);

 extern struct mutex xfrm_cfg_mutex;

```
diff --git a/net/core/sysctl_net_core.c b/net/core/sysctl_net_core.c
index 76f7a29..90f2a39 100644
--- a/net/core/sysctl_net_core.c
+++ b/net/core/sysctl_net_core.c
@@ -88,24 +88,6 @@ ctl_table core_table[] = {
   .mode  = 0644,
   .proc_handler = &proc_dointvec
  },
-#ifdef CONFIG_XFRM
- {
-  .ctl_name = NET_CORE_AEVENT_ETIME,
-  .procname = "xfrm_aevent_etime",
-  .data  = &sysctl_xfrm_aevent_etime,
-  .maxlen  = sizeof(u32),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_CORE_AEVENT_RSEQTH,
-  .procname = "xfrm_aevent_rseqth",
-  .data  = &sysctl_xfrm_aevent_rseqth,
-  .maxlen  = sizeof(u32),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
-#endif /* CONFIG_XFRM */
 #endif /* CONFIG_NET */
  {
   .ctl_name = NET_CORE_SOMAXCONN,
@@ -127,6 +109,23 @@ ctl_table core_table[] = {
 };

 DEFINE_PER_NET(struct ctl_table, multi_core_table[]) = {
- /* Stub for holding per network namespace sysctls */
+#ifdef CONFIG_XFRM
+ {
+  .ctl_name = NET_CORE_AEVENT_ETIME,
+  .procname = "xfrm_aevent_etime",
+  .data  = &__per_net_base(sysctl_xfrm_aevent_etime),
+  .maxlen  = sizeof(u32),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_CORE_AEVENT_RSEQTH,
+  .procname = "xfrm_aevent_rseqth",
+  .data  = &__per_net_base(sysctl_xfrm_aevent_rseqth),
```

```
+  .maxlen  = sizeof(u32),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+#endif /* CONFIG_XFRM */
  {}
 };
diff --git a/net/xfrm/xfrm_state.c b/net/xfrm/xfrm_state.c
index fdb08d9..3304a2d 100644
--- a/net/xfrm/xfrm_state.c
+++ b/net/xfrm/xfrm_state.c
@@ -27,11 +27,11 @@
 struct sock *xfrm_nl;
 EXPORT_SYMBOL(xfrm_nl);

-u32 sysctl_xfrm_aevent_etime = XFRM_AE_ETIME;
-EXPORT_SYMBOL(sysctl_xfrm_aevent_etime);
+DEFINE_PER_NET(u32, sysctl_xfrm_aevent_etime) = XFRM_AE_ETIME;
+EXPORT_PER_NET_SYMBOL(sysctl_xfrm_aevent_etime);

-u32 sysctl_xfrm_aevent_rseqth = XFRM_AE_SEQT_SIZE;
-EXPORT_SYMBOL(sysctl_xfrm_aevent_rseqth);
+DEFINE_PER_NET(u32, sysctl_xfrm_aevent_rseqth) = XFRM_AE_SEQT_SIZE;
+EXPORT_PER_NET_SYMBOL(sysctl_xfrm_aevent_rseqth);

 /* Each xfrm_state may be linked to two tables:

diff --git a/net/xfrm/xfrm_user.c b/net/xfrm/xfrm_user.c
index 55affa7..15e962b 100644
--- a/net/xfrm/xfrm_user.c
+++ b/net/xfrm/xfrm_user.c
@@ -375,7 +375,8 @@ error:
  return err;
 }

-static struct xfrm_state *xfrm_state_construct(struct xfrm_usersa_info *p,
+static struct xfrm_state *xfrm_state_construct(net_t net,
+            struct xfrm_usersa_info *p,
             struct rtattr **xfrma,
             int *errp)
 {
@@ -411,9 +412,9 @@ static struct xfrm_state *xfrm_state_construct(struct xfrm_usersa_info *p,
   goto error;

  x->km.seq = p->seq;
- x->replay_maxdiff = sysctl_xfrm_aevent_rseqth;
+ x->replay_maxdiff = per_net(sysctl_xfrm_aevent_rseqth, net);
  /* sysctl_xfrm_aevent_etime is in 100ms units */
```

```
- x->replay_maxage = (sysctl_xfrm_aevent_etime*HZ)/XFRM_AE_ETH_M;
+ x->replay_maxage = (per_net(sysctl_xfrm_aevent_etime, net)*HZ)/XFRM_AE_ETH_M;
  x->preplay.bitmap = 0;
  x->preplay.seq = x->replay.seq+x->replay_maxdiff;
  x->preplay.oseq = x->replay.oseq +x->replay_maxdiff;
@@ -437,6 +438,7 @@ error_no_put:
 static int xfrm_add_sa(struct sk_buff *skb, struct nlmsghdr *nlh,
    struct rtattr **xfrma)
 {
+ net_t net = skb->sk->sk_net;
  struct xfrm_usersa_info *p = NLMSG_DATA(nlh);
  struct xfrm_state *x;
  int err;
@@ -446,7 +448,7 @@ static int xfrm_add_sa(struct sk_buff *skb, struct nlmsghdr *nlh,
  if (err)
   return err;

- x = xfrm_state_construct(p, xfrma, &err);
+ x = xfrm_state_construct(net, p, xfrma, &err);
  if (!x)
   return err;

--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 28/31] net: Make the SOMAXCONN sysctl per network namespace

Posted by ebiederm on Thu, 25 Jan 2007 19:00:30 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 include/linux/socket.h     |    3 ++-
 net/core/sysctl_net_core.c |   16 ++++++++--------
 net/socket.c               |    7 ++++---
 3 files changed, 14 insertions(+), 12 deletions(-)

diff --git a/include/linux/socket.h b/include/linux/socket.h
index 92cd38e..aa159ea 100644
--- a/include/linux/socket.h
```

```
+++ b/include/linux/socket.h
@@ -23,8 +23,9 @@ struct __kernel_sockaddr_storage {
 #include <linux/uio.h>   /* iovec support  */
 #include <linux/types.h>  /* pid_t   */
 #include <linux/compiler.h>  /* __user   */
+#include <linux/net_namespace_type.h>

-extern int sysctl_somaxconn;
+DECLARE_PER_NET(int, sysctl_somaxconn);
 #ifdef CONFIG_PROC_FS
 struct seq_file;
 extern void socket_seq_show(struct seq_file *seq);
diff --git a/net/core/sysctl_net_core.c b/net/core/sysctl_net_core.c
index 90f2a39..14eca68 100644
--- a/net/core/sysctl_net_core.c
+++ b/net/core/sysctl_net_core.c
@@ -90,14 +90,6 @@ ctl_table core_table[] = {
  },
 #endif /* CONFIG_NET */
  {
-  .ctl_name = NET_CORE_SOMAXCONN,
-  .procname = "somaxconn",
-  .data  = &sysctl_somaxconn,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
   .ctl_name = NET_CORE_BUDGET,
   .procname = "netdev_budget",
   .data  = &netdev_budget,
@@ -127,5 +119,13 @@ DEFINE_PER_NET(struct ctl_table, multi_core_table[]) = {
   .proc_handler = &proc_dointvec
  },
 #endif /* CONFIG_XFRM */
+ {
+  .ctl_name = NET_CORE_SOMAXCONN,
+  .procname = "somaxconn",
+  .data  = &__per_net_base(sysctl_somaxconn),
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
  {}
 };
diff --git a/net/socket.c b/net/socket.c
index 7371654..ab2aeea 100644
--- a/net/socket.c
```

```
+++ b/net/socket.c
@@ -1305,7 +1305,7 @@ asmlinkage long sys_bind(int fd, struct sockaddr __user *umyaddr, int
addrlen)
  * ready for listening.
  */

-int sysctl_somaxconn __read_mostly = SOMAXCONN;
+DEFINE_PER_NET(int, sysctl_somaxconn)= SOMAXCONN;

 asmlinkage long sys_listen(int fd, int backlog)
 {
@@ -1314,8 +1314,9 @@ asmlinkage long sys_listen(int fd, int backlog)

  sock = sockfd_lookup_light(fd, &err, &fput_needed);
  if (sock) {
-  if ((unsigned)backlog > sysctl_somaxconn)
-   backlog = sysctl_somaxconn;
+  net_t net = sock->sk->sk_net;
+  if ((unsigned)backlog > per_net(sysctl_somaxconn, net))
+   backlog = per_net(sysctl_somaxconn, net);

   err = security_socket_listen(sock, backlog);
   if (!err)
--
1.4.4.1.g278f
```

Subject: [PATCH RFC 29/31] net: Make AF_PACKET handle multiple network namespaces

Posted by ebiederm on Thu, 25 Jan 2007 19:00:31 GMT

View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This is done by making all of the relevant global variables
per network namespace.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/packet/af_packet.c | 125 ++++++++++++++++++++++++++++++++++++++----------
 1 files changed, 81 insertions(+), 44 deletions(-)

diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c

```
index 4ac9f9f..c772491 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -152,8 +152,8 @@ dev->hard_header == NULL (ll header is added by device, we cannot
control it)
  */

 /* List of all packet sockets. */
-static HLIST_HEAD(packet_sklist);
-static DEFINE_RWLOCK(packet_sklist_lock);
+static DEFINE_PER_NET(rwlock_t, packet_sklist_lock);
+static DEFINE_PER_NET(struct hlist_head, packet_sklist);

 static atomic_t packet_socks_nr;

@@ -264,9 +264,6 @@ static int packet_rcv_spkt(struct sk_buff *skb, struct packet_type *pt,
struct n
  struct sock *sk;
  struct sockaddr_pkt *spkt;

- if (!net_eq(dev->nd_net, init_net()))
-  goto out;
-
  /*
   * When we registered the protocol we saved the socket in the data
   * field for just this event.
@@ -288,6 +285,9 @@ static int packet_rcv_spkt(struct sk_buff *skb, struct packet_type *pt,
struct n
  if (skb->pkt_type == PACKET_LOOPBACK)
   goto out;

+ if (!net_eq(dev->nd_net, sk->sk_net))
+  goto out;
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   goto oom;

@@ -359,7 +359,7 @@ static int packet_sendmsg_spkt(struct kiocb *iocb, struct socket *sock,
   */

  saddr->spkt_device[13] = 0;
- dev = dev_get_by_name(init_net(), saddr->spkt_device);
+ dev = dev_get_by_name(sk->sk_net, saddr->spkt_device);
  err = -ENODEV;
  if (dev == NULL)
   goto out_unlock;
@@ -475,15 +475,15 @@ static int packet_rcv(struct sk_buff *skb, struct packet_type *pt, struct
net_de
```

```
  int skb_len = skb->len;
  unsigned snaplen;

- if (!net_eq(dev->nd_net, init_net()))
-  goto drop;
-
  if (skb->pkt_type == PACKET_LOOPBACK)
   goto drop;

  sk = pt->af_packet_priv;
  po = pkt_sk(sk);

+ if (!net_eq(dev->nd_net, sk->sk_net))
+  goto drop;
+
  skb->dev = dev;

  if (dev->hard_header) {
@@ -583,15 +583,15 @@ static int tpacket_rcv(struct sk_buff *skb, struct packet_type *pt, struct
net_d
  unsigned short macoff, netoff;
  struct sk_buff *copy_skb = NULL;

- if (!net_eq(dev->nd_net, init_net()))
-  goto drop;
-
  if (skb->pkt_type == PACKET_LOOPBACK)
   goto drop;

  sk = pt->af_packet_priv;
  po = pkt_sk(sk);

+ if (!net_eq(dev->nd_net, sk->sk_net))
+  goto drop;
+
  if (dev->hard_header) {
   if (sk->sk_type != SOCK_DGRAM)
    skb_push(skb, skb->data - skb->mac.raw);
@@ -744,7 +744,7 @@ static int packet_sendmsg(struct kiocb *iocb, struct socket *sock,
  }


- dev = dev_get_by_index(init_net(), ifindex);
+ dev = dev_get_by_index(sk->sk_net, ifindex);
  err = -ENXIO;
  if (dev == NULL)
   goto out_unlock;
@@ -817,15 +817,17 @@ static int packet_release(struct socket *sock)
```

```
 {
  struct sock *sk = sock->sk;
  struct packet_sock *po;
+ net_t net;

  if (!sk)
   return 0;

+ net = sk->sk_net;
  po = pkt_sk(sk);

- write_lock_bh(&packet_sklist_lock);
+ write_lock_bh(&per_net(packet_sklist_lock, net));
  sk_del_node_init(sk);
- write_unlock_bh(&packet_sklist_lock);
+ write_unlock_bh(&per_net(packet_sklist_lock, net));

  /*
   * Unhook packet receive handler.
@@ -943,7 +945,7 @@ static int packet_bind_spkt(struct socket *sock, struct sockaddr *uaddr,
int add
   return -EINVAL;
  strlcpy(name,uaddr->sa_data,sizeof(name));

- dev = dev_get_by_name(init_net(), name);
+ dev = dev_get_by_name(sk->sk_net, name);
  if (dev) {
   err = packet_do_bind(sk, dev, pkt_sk(sk)->num);
   dev_put(dev);
@@ -971,7 +973,7 @@ static int packet_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len

  if (sll->sll_ifindex) {
   err = -ENODEV;
-  dev = dev_get_by_index(init_net(), sll->sll_ifindex);
+  dev = dev_get_by_index(sk->sk_net, sll->sll_ifindex);
   if (dev == NULL)
    goto out;
  }
@@ -1000,9 +1002,6 @@ static int packet_create(net_t net, struct socket *sock, int protocol)
  __be16 proto = (__force __be16)protocol; /* weird, but documented */
  int err;

- if (!net_eq(net, init_net()))
-  return -EAFNOSUPPORT;
-
  if (!capable(CAP_NET_RAW))
   return -EPERM;
```

```
  if (sock->type != SOCK_DGRAM && sock->type != SOCK_RAW
@@ -1052,9 +1051,9 @@ static int packet_create(net_t net, struct socket *sock, int protocol)
   po->running = 1;
  }

- write_lock_bh(&packet_sklist_lock);
- sk_add_node(sk, &packet_sklist);
- write_unlock_bh(&packet_sklist_lock);
+ write_lock_bh(&per_net(packet_sklist_lock, net));
+ sk_add_node(sk, &per_net(packet_sklist, net));
+ write_unlock_bh(&per_net(packet_sklist_lock, net));
  return(0);
 out:
  return err;
@@ -1158,7 +1157,7 @@ static int packet_getname_spkt(struct socket *sock, struct sockaddr
*uaddr,
   return -EOPNOTSUPP;

  uaddr->sa_family = AF_PACKET;
- dev = dev_get_by_index(init_net(), pkt_sk(sk)->ifindex);
+ dev = dev_get_by_index(sk->sk_net, pkt_sk(sk)->ifindex);
  if (dev) {
   strlcpy(uaddr->sa_data, dev->name, 15);
   dev_put(dev);
@@ -1184,7 +1183,7 @@ static int packet_getname(struct socket *sock, struct sockaddr *uaddr,
  sll->sll_family = AF_PACKET;
  sll->sll_ifindex = po->ifindex;
  sll->sll_protocol = po->num;
- dev = dev_get_by_index(init_net(), po->ifindex);
+ dev = dev_get_by_index(sk->sk_net, po->ifindex);
  if (dev) {
   sll->sll_hatype = dev->type;
   sll->sll_halen = dev->addr_len;
@@ -1237,7 +1236,7 @@ static int packet_mc_add(struct sock *sk, struct packet_mreq_max
*mreq)
  rtnl_lock();

  err = -ENODEV;
- dev = __dev_get_by_index(init_net(), mreq->mr_ifindex);
+ dev = __dev_get_by_index(sk->sk_net, mreq->mr_ifindex);
  if (!dev)
   goto done;

@@ -1291,7 +1290,7 @@ static int packet_mc_drop(struct sock *sk, struct packet_mreq_max
*mreq)
    if (--ml->count == 0) {
     struct net_device *dev;
     *mlp = ml->next;
```

```
-    dev = dev_get_by_index(init_net(), ml->ifindex);
+    dev = dev_get_by_index(sk->sk_net, ml->ifindex);
     if (dev) {
      packet_dev_mc(dev, ml, -1);
      dev_put(dev);
@@ -1319,7 +1318,7 @@ static void packet_flush_mclist(struct sock *sk)
   struct net_device *dev;

   po->mclist = ml->next;
-  if ((dev = dev_get_by_index(init_net(), ml->ifindex)) != NULL) {
+  if ((dev = dev_get_by_index(sk->sk_net, ml->ifindex)) != NULL) {
    packet_dev_mc(dev, ml, -1);
    dev_put(dev);
   }
@@ -1438,12 +1437,10 @@ static int packet_notifier(struct notifier_block *this, unsigned long
msg, void
  struct sock *sk;
  struct hlist_node *node;
  struct net_device *dev = (struct net_device*)data;
+ net_t net = dev->nd_net;

- if (!net_eq(dev->nd_net, init_net()))
-  return NOTIFY_DONE;
-
- read_lock(&packet_sklist_lock);
- sk_for_each(sk, node, &packet_sklist) {
+ read_lock(&per_net(packet_sklist_lock, net));
+ sk_for_each(sk, node, &per_net(packet_sklist, net)) {
   struct packet_sock *po = pkt_sk(sk);

   switch (msg) {
@@ -1483,7 +1480,7 @@ static int packet_notifier(struct notifier_block *this, unsigned long msg,
void
    break;
   }
  }
- read_unlock(&packet_sklist_lock);
+ read_unlock(&per_net(packet_sklist_lock, net));
  return NOTIFY_DONE;
 }

@@ -1851,12 +1848,12 @@ static struct notifier_block packet_netdev_notifier = {
 };

 #ifdef CONFIG_PROC_FS
-static inline struct sock *packet_seq_idx(loff_t off)
+static inline struct sock *packet_seq_idx(net_t net, loff_t off)
 {
```

```
  struct sock *s;
  struct hlist_node *node;

- sk_for_each(s, node, &packet_sklist) {
+ sk_for_each(s, node, &per_net(packet_sklist, net)) {
   if (!off--)
    return s;
  }
@@ -1865,21 +1862,24 @@ static inline struct sock *packet_seq_idx(loff_t off)

 static void *packet_seq_start(struct seq_file *seq, loff_t *pos)
 {
- read_lock(&packet_sklist_lock);
- return *pos ? packet_seq_idx(*pos - 1) : SEQ_START_TOKEN;
+ net_t net = net_from_voidp(seq->private);
+ read_lock(&per_net(packet_sklist_lock, net));
+ return *pos ? packet_seq_idx(net, *pos - 1) : SEQ_START_TOKEN;
 }

 static void *packet_seq_next(struct seq_file *seq, void *v, loff_t *pos)
 {
+ net_t net = net_from_voidp(seq->private);
  ++*pos;
  return  (v == SEQ_START_TOKEN)
-  ? sk_head(&packet_sklist)
+  ? sk_head(&per_net(packet_sklist, net))
   : sk_next((struct sock*)v) ;
 }

 static void packet_seq_stop(struct seq_file *seq, void *v)
 {
- read_unlock(&packet_sklist_lock);
+ net_t net = net_from_voidp(seq->private);
+ read_unlock(&per_net(packet_sklist_lock, net));
 }

 static int packet_seq_show(struct seq_file *seq, void *v)
@@ -1915,7 +1915,22 @@ static struct seq_operations packet_seq_ops = {

 static int packet_seq_open(struct inode *inode, struct file *file)
 {
- return seq_open(file, &packet_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res = seq_open(file, &packet_seq_ops);
+ if (!res) {
+  seq = file->private_data;
+  seq->private = net_to_voidp(get_net(PROC_NET(inode)));
```

```
+ }
+ return res;
+}
+
+static int packet_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq=  file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
 }

 static struct file_operations packet_seq_fops = {
@@ -1923,15 +1938,37 @@ static struct file_operations packet_seq_fops = {
  .open  = packet_seq_open,
  .read  = seq_read,
  .llseek  = seq_lseek,
- .release = seq_release,
+ .release = packet_seq_release,
 };

 #endif

+static int packet_net_init(net_t net)
+{
+ rwlock_init(&per_net(packet_sklist_lock, net));
+ INIT_HLIST_HEAD(&per_net(packet_sklist, net));
+
+ if (!proc_net_fops_create(net, "packet", 0, &packet_seq_fops))
+  return -ENOMEM;
+
+ return 0;
+}
+
+static void packet_net_exit(net_t net)
+{
+ proc_net_remove(net, "packet");
+}
+
+static struct pernet_operations packet_net_ops = {
+ .init = packet_net_init,
+ .exit = packet_net_exit,
+};
+
+
 static void __exit packet_exit(void)
 {
- proc_net_remove(init_net(), "packet");
```

```
 	unregister_netdevice_notifier(&packet_netdev_notifier);
+	unregister_pernet_subsys(&packet_net_ops);
 	sock_unregister(PF_PACKET);
 	proto_unregister(&packet_proto);
 }
@@ -1944,8 +1981,8 @@ static int __init packet_init(void)
 		goto out;

 	sock_register(&packet_family_ops);
+	register_pernet_subsys(&packet_net_ops);
 	register_netdevice_notifier(&packet_netdev_notifier);
-	proc_net_fops_create(init_net(), "packet", 0, &packet_seq_fops);
 out:
 	return rc;
 }
--
1.4.4.1.g278f
```

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 30/31] net: Make AF_UNIX per network namespace safe.
Posted by ebiederm on Thu, 25 Jan 2007 19:00:32 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Because of the global nature of garbage collection, and
because of the cost of per namespace hash tables
unix_socket_table has been kept global. With a filter
added on lookups so we don't see sockets from the wrong
namespace.

Currently I don't fold the namesapce into the hash so
multiple namespaces using the same socket name will be
guaranateed a hash collision.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 include/net/af_unix.h      |   10 ++--
 net/unix/af_unix.c         |  116 +++++++++++++++++++++++++++++++-------------
 net/unix/sysctl_net_unix.c |   24 +++++----
 3 files changed, 103 insertions(+), 47 deletions(-)

diff --git a/include/net/af_unix.h b/include/net/af_unix.h
```

```
index c0398f5..1f40dd2 100644
--- a/include/net/af_unix.h
+++ b/include/net/af_unix.h
@@ -89,12 +89,12 @@ struct unix_sock {
 #define unix_sk(__sk) ((struct unix_sock *)__sk)

 #ifdef CONFIG_SYSCTL
-extern int sysctl_unix_max_dgram_qlen;
-extern void unix_sysctl_register(void);
-extern void unix_sysctl_unregister(void);
+DECLARE_PER_NET(int, sysctl_unix_max_dgram_qlen);
+extern void unix_sysctl_register(net_t net);
+extern void unix_sysctl_unregister(net_t net);
 #else
-static inline void unix_sysctl_register(void) {}
-static inline void unix_sysctl_unregister(void) {}
+static inline void unix_sysctl_register(net_t net) {}
+static inline void unix_sysctl_unregister(net_t net) {}
 #endif
 #endif
 #endif
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 8015a03..3f57cb2 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -118,7 +118,7 @@
 #include <linux/security.h>
 #include <net/net_namespace.h>

-int sysctl_unix_max_dgram_qlen __read_mostly = 10;
+DEFINE_PER_NET(int, sysctl_unix_max_dgram_qlen) = 10;

 struct hlist_head unix_socket_table[UNIX_HASH_SIZE + 1];
 DEFINE_SPINLOCK(unix_table_lock);
@@ -245,7 +245,8 @@ static inline void unix_insert_socket(struct hlist_head *list, struct sock
*sk)
  spin_unlock(&unix_table_lock);
 }

-static struct sock *__unix_find_socket_byname(struct sockaddr_un *sunname,
+static struct sock *__unix_find_socket_byname(net_t net,
+           struct sockaddr_un *sunname,
           int len, int type, unsigned hash)
 {
  struct sock *s;
@@ -254,6 +255,9 @@ static struct sock *__unix_find_socket_byname(struct sockaddr_un
*sunname,
  sk_for_each(s, node, &unix_socket_table[hash ^ type]) {
```

```
   struct unix_sock *u = unix_sk(s);

+  if (!net_eq(s->sk_net, net))
+   continue;
+
   if (u->addr->len == len &&
       !memcmp(u->addr->name, sunname, len))
    goto found;
@@ -263,21 +267,22 @@ found:
  return s;
 }

-static inline struct sock *unix_find_socket_byname(struct sockaddr_un *sunname,
+static inline struct sock *unix_find_socket_byname(net_t net,
+         struct sockaddr_un *sunname,
          int len, int type,
          unsigned hash)
 {
  struct sock *s;

  spin_lock(&unix_table_lock);
- s = __unix_find_socket_byname(sunname, len, type, hash);
+ s = __unix_find_socket_byname(net, sunname, len, type, hash);
  if (s)
   sock_hold(s);
  spin_unlock(&unix_table_lock);
  return s;
 }

-static struct sock *unix_find_socket_byinode(struct inode *i)
+static struct sock *unix_find_socket_byinode(net_t net, struct inode *i)
 {
  struct sock *s;
  struct hlist_node *node;
@@ -287,6 +292,9 @@ static struct sock *unix_find_socket_byinode(struct inode *i)
       &unix_socket_table[i->i_ino & (UNIX_HASH_SIZE - 1)]) {
   struct dentry *dentry = unix_sk(s)->dentry;

+  if (!net_eq(s->sk_net, net))
+   continue;
+
   if(dentry && dentry->d_inode == i)
   {
    sock_hold(s);
@@ -588,7 +596,7 @@ static struct sock * unix_create1(net_t net, struct socket *sock)
     &af_unix_sk_receive_queue_lock_key);

  sk->sk_write_space = unix_write_space;
```

```
- sk->sk_max_ack_backlog = sysctl_unix_max_dgram_qlen;
+ sk->sk_max_ack_backlog = per_net(sysctl_unix_max_dgram_qlen, net);
  sk->sk_destruct  = unix_sock_destructor;
  u   = unix_sk(sk);
  u->dentry = NULL;
@@ -604,9 +612,6 @@ out:

 static int unix_create(net_t net, struct socket *sock, int protocol)
 {
- if (!net_eq(net, init_net()))
-  return -EAFNOSUPPORT;
-
  if (protocol && protocol != PF_UNIX)
   return -EPROTONOSUPPORT;

@@ -650,6 +655,7 @@ static int unix_release(struct socket *sock)
 static int unix_autobind(struct socket *sock)
 {
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct unix_sock *u = unix_sk(sk);
  static u32 ordernum = 1;
  struct unix_address * addr;
@@ -676,7 +682,7 @@ retry:
  spin_lock(&unix_table_lock);
  ordernum = (ordernum+1)&0xFFFFF;

- if (__unix_find_socket_byname(addr->name, addr->len, sock->type,
+ if (__unix_find_socket_byname(net, addr->name, addr->len, sock->type,
           addr->hash)) {
   spin_unlock(&unix_table_lock);
   /* Sanity yield. It is unusual case, but yet... */
@@ -696,7 +702,8 @@ out: mutex_unlock(&u->readlock);
  return err;
 }

-static struct sock *unix_find_other(struct sockaddr_un *sunname, int len,
+static struct sock *unix_find_other(net_t net,
+        struct sockaddr_un *sunname, int len,
         int type, unsigned hash, int *error)
 {
  struct sock *u;
@@ -714,7 +721,7 @@ static struct sock *unix_find_other(struct sockaddr_un *sunname, int len,
   err = -ECONNREFUSED;
   if (!S_ISSOCK(nd.dentry->d_inode->i_mode))
    goto put_fail;
-  u=unix_find_socket_byinode(nd.dentry->d_inode);
+  u=unix_find_socket_byinode(net, nd.dentry->d_inode);
```

```
   if (!u)
    goto put_fail;

@@ -730,7 +737,7 @@ static struct sock *unix_find_other(struct sockaddr_un *sunname, int len,
    }
   } else {
    err = -ECONNREFUSED;
-   u=unix_find_socket_byname(sunname, len, type, hash);
+   u=unix_find_socket_byname(net, sunname, len, type, hash);
    if (u) {
     struct dentry *dentry;
     dentry = unix_sk(u)->dentry;
@@ -752,6 +759,7 @@ fail:
 static int unix_bind(struct socket *sock, struct sockaddr *uaddr, int addr_len)
 {
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct unix_sock *u = unix_sk(sk);
  struct sockaddr_un *sunaddr=(struct sockaddr_un *)uaddr;
  struct dentry * dentry = NULL;
@@ -826,7 +834,7 @@ static int unix_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len)

  if (!sunaddr->sun_path[0]) {
   err = -EADDRINUSE;
-  if (__unix_find_socket_byname(sunaddr, addr_len,
+  if (__unix_find_socket_byname(net, sunaddr, addr_len,
          sk->sk_type, hash)) {
    unix_release_addr(addr);
    goto out_unlock;
@@ -867,6 +875,7 @@ static int unix_dgram_connect(struct socket *sock, struct sockaddr *addr,
         int alen, int flags)
 {
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct sockaddr_un *sunaddr=(struct sockaddr_un*)addr;
  struct sock *other;
  unsigned hash;
@@ -882,7 +891,7 @@ static int unix_dgram_connect(struct socket *sock, struct sockaddr *addr,
       !unix_sk(sk)->addr && (err = unix_autobind(sock)) != 0)
    goto out;

-  other=unix_find_other(sunaddr, alen, sock->type, hash, &err);
+  other=unix_find_other(net, sunaddr, alen, sock->type, hash, &err);
   if (!other)
    goto out;

@@ -955,6 +964,7 @@ static int unix_stream_connect(struct socket *sock, struct sockaddr
```

```
*uaddr,
 {
  struct sockaddr_un *sunaddr=(struct sockaddr_un *)uaddr;
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct unix_sock *u = unix_sk(sk), *newu, *otheru;
  struct sock *newsk = NULL;
  struct sock *other = NULL;
@@ -994,7 +1004,7 @@ static int unix_stream_connect(struct socket *sock, struct sockaddr
*uaddr,

 restart:
  /*  Find listening sock. */
- other = unix_find_other(sunaddr, addr_len, sk->sk_type, hash, &err);
+ other = unix_find_other(net, sunaddr, addr_len, sk->sk_type, hash, &err);
  if (!other)
   goto out;

@@ -1273,6 +1283,7 @@ static int unix_dgram_sendmsg(struct kiocb *kiocb, struct socket
*sock,
 {
  struct sock_iocb *siocb = kiocb_to_siocb(kiocb);
  struct sock *sk = sock->sk;
+ net_t net = sk->sk_net;
  struct unix_sock *u = unix_sk(sk);
  struct sockaddr_un *sunaddr=msg->msg_name;
  struct sock *other = NULL;
@@ -1336,7 +1347,7 @@ restart:
   if (sunaddr == NULL)
    goto out_free;

-  other = unix_find_other(sunaddr, namelen, sk->sk_type,
+  other = unix_find_other(net, sunaddr, namelen, sk->sk_type,
       hash, &err);
   if (other==NULL)
    goto out_free;
@@ -1935,12 +1946,18 @@ static unsigned int unix_poll(struct file * file, struct socket *sock,
poll_tabl


 #ifdef CONFIG_PROC_FS
-static struct sock *unix_seq_idx(int *iter, loff_t pos)
+struct unix_iter_state {
+ net_t net;
+ int i;
+};
+static struct sock *unix_seq_idx(struct unix_iter_state *iter, loff_t pos)
 {
```

```
  loff_t off = 0;
  struct sock *s;

- for (s = first_unix_socket(iter); s; s = next_unix_socket(iter, s)) {
+ for (s = first_unix_socket(&iter->i); s; s = next_unix_socket(&iter->i, s)) {
+  if (!net_eq(s->sk_net, iter->net))
+   continue;
   if (off == pos)
    return s;
   ++off;
@@ -1951,17 +1968,24 @@ static struct sock *unix_seq_idx(int *iter, loff_t pos)

 static void *unix_seq_start(struct seq_file *seq, loff_t *pos)
 {
+ struct unix_iter_state *iter = seq->private;
  spin_lock(&unix_table_lock);
- return *pos ? unix_seq_idx(seq->private, *pos - 1) : ((void *) 1);
+ return *pos ? unix_seq_idx(iter, *pos - 1) : ((void *) 1);
 }

 static void *unix_seq_next(struct seq_file *seq, void *v, loff_t *pos)
 {
+ struct unix_iter_state *iter = seq->private;
+ struct sock *sk = v;
  ++*pos;

  if (v == (void *)1)
-  return first_unix_socket(seq->private);
- return next_unix_socket(seq->private, v);
+  sk = first_unix_socket(&iter->i);
+ else
+  sk = next_unix_socket(&iter->i, sk);
+ while (sk && !net_eq(sk->sk_net, iter->net))
+  sk = next_unix_socket(&iter->i, sk);
+ return sk;
 }

 static void unix_seq_stop(struct seq_file *seq, void *v)
@@ -2025,7 +2049,7 @@ static int unix_seq_open(struct inode *inode, struct file *file)
 {
  struct seq_file *seq;
  int rc = -ENOMEM;
- int *iter = kmalloc(sizeof(int), GFP_KERNEL);
+ struct unix_iter_state *iter = kmalloc(sizeof(*iter), GFP_KERNEL);

  if (!iter)
   goto out;
@@ -2036,7 +2060,8 @@ static int unix_seq_open(struct inode *inode, struct file *file)
```

```
  seq      = file->private_data;
  seq->private = iter;
- *iter = 0;
+ iter->net = get_net(PROC_NET(inode));
+ iter->i = 0;
 out:
  return rc;
 out_kfree:
@@ -2044,12 +2069,20 @@ out_kfree:
  goto out;
 }

+static int unix_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ struct unix_iter_state *iter = seq->private;
+ put_net(iter->net);
+ return seq_release_private(inode, file);
+}
+
 static struct file_operations unix_seq_fops = {
  .owner  = THIS_MODULE,
  .open  = unix_seq_open,
  .read  = seq_read,
  .llseek  = seq_lseek,
- .release = seq_release_private,
+ .release = unix_seq_release,
 };

 #endif
@@ -2060,6 +2093,31 @@ static struct net_proto_family unix_family_ops = {
  .owner = THIS_MODULE,
 };

+
+static int unix_net_init(net_t net)
+{
+ int error = -ENOMEM;
+#ifdef CONFIG_PROC_FS
+ if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops))
+  goto out;
+#endif
+ unix_sysctl_register(net);
+ error = 0;
+out:
+ return 0;
+}
```

```
+
+static void unix_net_exit(net_t net)
+{
+ unix_sysctl_unregister(net);
+ proc_net_remove(net, "unix");
+}
+
+static struct pernet_operations unix_net_ops = {
+ .init = unix_net_init,
+ .exit = unix_net_exit,
+};
+
 static int __init af_unix_init(void)
 {
  int rc = -1;
@@ -2075,10 +2133,7 @@ static int __init af_unix_init(void)
  }

  sock_register(&unix_family_ops);
-#ifdef CONFIG_PROC_FS
- proc_net_fops_create(init_net(), "unix", 0, &unix_seq_fops);
-#endif
- unix_sysctl_register();
+ register_pernet_subsys(&unix_net_ops);
 out:
  return rc;
 }
@@ -2086,9 +2141,8 @@ out:
 static void __exit af_unix_exit(void)
 {
  sock_unregister(PF_UNIX);
- unix_sysctl_unregister();
- proc_net_remove(init_net(), "unix");
  proto_unregister(&unix_proto);
+ unregister_pernet_subsys(&unix_net_ops);
 }

 module_init(af_unix_init);
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
index eb0bd57..4b59da8 100644
--- a/net/unix/sysctl_net_unix.c
+++ b/net/unix/sysctl_net_unix.c
@@ -14,11 +14,11 @@

 #include <net/af_unix.h>

-static ctl_table unix_table[] = {
+static DEFINE_PER_NET(ctl_table, unix_table[]) = {
```

```
  {
   .ctl_name = NET_UNIX_MAX_DGRAM_QLEN,
   .procname = "max_dgram_qlen",
-  .data  = &sysctl_unix_max_dgram_qlen,
+  .data  = &__per_net_base(sysctl_unix_max_dgram_qlen),
   .maxlen  = sizeof(int),
   .mode  = 0644,
   .proc_handler = &proc_dointvec
@@ -26,35 +26,37 @@ static ctl_table unix_table[] = {
  { .ctl_name = 0 }
 };

-static ctl_table unix_net_table[] = {
+static DEFINE_PER_NET(ctl_table, unix_net_table[]) = {
  {
   .ctl_name = NET_UNIX,
   .procname = "unix",
   .mode  = 0555,
-  .child  = unix_table
+  .child  = __per_net_base(unix_table)
  },
  { .ctl_name = 0 }
 };

-static ctl_table unix_root_table[] = {
+static DEFINE_PER_NET(ctl_table, unix_root_table[]) = {
  {
   .ctl_name = CTL_NET,
   .procname = "net",
   .mode  = 0555,
-  .child  = unix_net_table
+  .child  = __per_net_base(unix_net_table)
  },
  { .ctl_name = 0 }
 };

-static struct ctl_table_header * unix_sysctl_header;
+static DEFINE_PER_NET(struct ctl_table_header *, unix_sysctl_header);

-void unix_sysctl_register(void)
+void unix_sysctl_register(net_t net)
 {
- unix_sysctl_header = register_sysctl_table(unix_root_table);
+ ctl_table *table = per_net(unix_root_table, net);
+ per_net(unix_sysctl_header, net) =
+  register_net_sysctl_table(net, table);
 }
```

```
-void unix_sysctl_unregister(void)
+void unix_sysctl_unregister(net_t net)
 {
- unregister_sysctl_table(unix_sysctl_header);
+ unregister_net_sysctl_table(per_net(unix_sysctl_header, net));
 }

--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH RFC 31/31] net: Add etun driver
Posted by ebiederm on Thu, 25 Jan 2007 19:00:33 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

etun is a simple two headed tunnel driver that at the link layer
looks like ethernet. It's target audience is communicating
between network namespaces but it is general enough it may
have other uses as well.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/net/Kconfig | 14 ++
 drivers/net/Makefile | 1 +
 drivers/net/etun.c | 470 ++
 3 files changed, 485 insertions(+), 0 deletions(-)

```
diff --git a/drivers/net/Kconfig b/drivers/net/Kconfig
index 8aa8dd0..969d3df 100644
--- a/drivers/net/Kconfig
+++ b/drivers/net/Kconfig
@@ -119,6 +119,20 @@ config TUN

    If you don't know what to use this for, you don't need it.

+config ETUN
+ tristate "Ethernet tunnel device driver support"
+ depends on SYSFS
+ ---help---
+   ETUN provices a pair of network devices that can be used for
+   configuring interesting topolgies.  What one devices transmits
```

```
+   the other receives and vice versa.  The link level framing
+   is ethernet for wide compatibility with network stacks.
+
+   To compile this driver as a module, choose M here: the module
+   will be called etun.
+
+   If you don't know what to use this for, you don't need it.
+
 config NET_SB1000
  tristate "General Instruments Surfboard 1000"
  depends on PNP
diff --git a/drivers/net/Makefile b/drivers/net/Makefile
index 4c0d4e5..396af4f 100644
--- a/drivers/net/Makefile
+++ b/drivers/net/Makefile
@@ -185,6 +185,7 @@ obj-$(CONFIG_MACSONIC) += macsonic.o
 obj-$(CONFIG_MACMACE) += macmace.o
 obj-$(CONFIG_MAC89x0) += mac89x0.o
 obj-$(CONFIG_TUN) += tun.o
+obj-$(CONFIG_ETUN) += etun.o
 obj-$(CONFIG_NET_NETX) += netx-eth.o
 obj-$(CONFIG_DL2K) += dl2k.o
 obj-$(CONFIG_R8169) += r8169.o
diff --git a/drivers/net/etun.c b/drivers/net/etun.c
new file mode 100644
index 0000000..1dd8cd8
--- /dev/null
+++ b/drivers/net/etun.c
@@ -0,0 +1,470 @@
+/*
+ *  ETUN - Universal ETUN device driver.
+ *  Copyright (C) 2006 Linux Networx
+ *
+ */
+
+#define DRV_NAME "etun"
+#define DRV_VERSION "1.0"
+#define DRV_DESCRIPTION "Ethernet pseudo tunnel device driver"
+#define DRV_COPYRIGHT "(C) 2007 Linux Networx"
+
+#include <linux/module.h>
+#include <linux/kernel.h>
+#include <linux/list.h>
+#include <linux/spinlock.h>
+#include <linux/skbuff.h>
+#include <linux/netdevice.h>
+#include <linux/etherdevice.h>
+#include <linux/ethtool.h>
```

```
+#include <linux/rtnetlink.h>
+#include <linux/if.h>
+#include <linux/if_ether.h>
+#include <linux/ctype.h>
+#include <net/net_namespace.h>
+#include <net/dst.h>
+
+
+/* Device cheksum strategy.
+ *
+ * etun is designed to a be a pair of virutal devices
+ * connecting two network stack instances.
+ *
+ * Typically it will either be used with ethernet bridging or
+ * it will be used to route packets between the two stacks.
+ *
+ * The only checksum offloading I can do is to completely
+ * skip the checksumming step all together.
+ *
+ * When used for ethernet bridging I don't believe any
+ * checksum off loading is safe.
+ * - If my source is an external interface the checksum may be
+ *   invalid so I don't want to report I have already checked it.
+ * - If my destination is an external interface I don't want to put
+ *   a packet on the wire with someone computing the checksum.
+ *
+ * When used for routing between two stacks checksums should
+ * be as unnecessary as they are on the loopback device.
+ *
+ * So by default I am safe and disable checksumming and
+ * other advanced features like SG and TSO.
+ *
+ * However because I think these features could be useful
+ * I provide the ethtool functions to and enable/disable
+ * them at runtime.
+ *
+ * If you think you can correctly enable these go ahead.
+ * For checksums both the transmitter and the receiver must
+ * agree before the are actually disabled.
+ */
+
+#define ETUN_NUM_STATS 1
+static struct {
+ const char string[ETH_GSTRING_LEN];
+} ethtool_stats_keys[ETUN_NUM_STATS] = {
+ { "partner_ifindex" },
+};
+
```

```
+struct etun_info {
+ struct net_device  *rx_dev;
+ unsigned  ip_summed;
+ struct net_device_stats stats;
+ struct list_head list;
+ struct net_device *dev;
+};
+
+/*
+ * I have to hold the rtnl_lock during device delete.
+ * So I use the rtnl_lock to protect my list manipulations
+ * as well.  Crude but simple.
+ */
+static LIST_HEAD(etun_list);
+
+/*
+ * The higher levels take care of making this non-reentrant (it's
+ * called with bh's disabled).
+ */
+static int etun_xmit(struct sk_buff *skb, struct net_device *tx_dev)
+{
+ struct etun_info *tx_info = tx_dev->priv;
+ struct net_device *rx_dev = tx_info->rx_dev;
+ struct etun_info *rx_info = rx_dev->priv;
+
+ tx_info->stats.tx_packets++;
+ tx_info->stats.tx_bytes += skb->len;
+
+ /* Drop the skb state that was needed to get here */
+ skb_orphan(skb);
+ if (skb->dst)
+  skb->dst = dst_pop(skb->dst); /* Allow for smart routing */
+
+ /* Switch to the receiving device */
+ skb->pkt_type = PACKET_HOST;
+ skb->protocol = eth_type_trans(skb, rx_dev);
+ skb->dev = rx_dev;
+ skb->ip_summed = CHECKSUM_NONE;
+
+ /* If both halves agree no checksum is needed */
+ if (tx_dev->features & NETIF_F_NO_CSUM)
+  skb->ip_summed = rx_info->ip_summed;
+
+ rx_dev->last_rx = jiffies;
+ rx_info->stats.rx_packets++;
+ rx_info->stats.rx_bytes += skb->len;
+ netif_rx(skb);
+
```

```
+ return 0;
+}
+
+static struct net_device_stats *etun_get_stats(struct net_device *dev)
+{
+ struct etun_info *info = dev->priv;
+ return &info->stats;
+}
+
+/* ethtool interface */
+static int etun_get_settings(struct net_device *dev, struct ethtool_cmd *cmd)
+{
+ cmd->supported  = 0;
+ cmd->advertising = 0;
+ cmd->speed  = SPEED_10000; /* Memory is fast! */
+ cmd->duplex  = DUPLEX_FULL;
+ cmd->port  = PORT_TP;
+ cmd->phy_address = 0;
+ cmd->transceiver = XCVR_INTERNAL;
+ cmd->autoneg  = AUTONEG_DISABLE;
+ cmd->maxtxpkt  = 0;
+ cmd->maxrxpkt  = 0;
+ return 0;
+}
+
+static void etun_get_drvinfo(struct net_device *dev, struct ethtool_drvinfo *info)
+{
+ strcpy(info->driver, DRV_NAME);
+ strcpy(info->version, DRV_VERSION);
+ strcpy(info->fw_version, "N/A");
+}
+
+static void etun_get_strings(struct net_device *dev, u32 stringset, u8 *buf)
+{
+ switch(stringset) {
+ case ETH_SS_STATS:
+  memcpy(buf, &ethtool_stats_keys, sizeof(ethtool_stats_keys));
+  break;
+ case ETH_SS_TEST:
+ default:
+  break;
+ }
+}
+
+static int etun_get_stats_count(struct net_device *dev)
+{
+ return ETUN_NUM_STATS;
+}
```

```
+
+static void etun_get_ethtool_stats(struct net_device *dev,
+ struct ethtool_stats *stats, u64 *data)
+{
+ struct etun_info *info = dev->priv;
+
+ data[0] = info->rx_dev->ifindex;
+}
+
+static u32 etun_get_rx_csum(struct net_device *dev)
+{
+ struct etun_info *info = dev->priv;
+ return info->ip_summed == CHECKSUM_UNNECESSARY;
+}
+
+static int etun_set_rx_csum(struct net_device *dev, u32 data)
+{
+ struct etun_info *info = dev->priv;
+
+ info->ip_summed = data ? CHECKSUM_UNNECESSARY : CHECKSUM_NONE;
+
+ return 0;
+}
+
+static u32 etun_get_tx_csum(struct net_device *dev)
+{
+ return (dev->features & NETIF_F_NO_CSUM) != 0;
+}
+
+static int etun_set_tx_csum(struct net_device *dev, u32 data)
+{
+ dev->features &= NETIF_F_NO_CSUM;
+ if (data)
+  dev->features |= NETIF_F_NO_CSUM;
+
+ return 0;
+}
+
+static struct ethtool_ops etun_ethtool_ops = {
+ .get_settings  = etun_get_settings,
+ .get_drvinfo  = etun_get_drvinfo,
+ .get_link  = ethtool_op_get_link,
+ .get_rx_csum  = etun_get_rx_csum,
+ .set_rx_csum  = etun_set_rx_csum,
+ .get_tx_csum  = etun_get_tx_csum,
+ .set_tx_csum  = etun_set_tx_csum,
+ .get_sg   = ethtool_op_get_sg,
+ .set_sg   = ethtool_op_set_sg,
```

```
+#if 0 /* Does just setting the bit successfuly emulate tso? */
+ .get_tso  = ethtool_op_get_tso,
+ .set_tso  = ethtool_op_set_tso,
+#endif
+ .get_strings  = etun_get_strings,
+ .get_stats_count = etun_get_stats_count,
+ .get_ethtool_stats = etun_get_ethtool_stats,
+ .get_perm_addr  = ethtool_op_get_perm_addr,
+};
+
+static int etun_open(struct net_device *tx_dev)
+{
+ struct etun_info *tx_info = tx_dev->priv;
+ struct net_device *rx_dev = tx_info->rx_dev;
+ if (rx_dev->flags & IFF_UP) {
+  netif_carrier_on(tx_dev);
+  netif_carrier_on(rx_dev);
+ }
+ netif_start_queue(tx_dev);
+ return 0;
+}
+
+static int etun_stop(struct net_device *tx_dev)
+{
+ struct etun_info *tx_info = tx_dev->priv;
+ struct net_device *rx_dev = tx_info->rx_dev;
+ netif_stop_queue(tx_dev);
+ if (netif_carrier_ok(tx_dev)) {
+  netif_carrier_off(tx_dev);
+  netif_carrier_off(rx_dev);
+ }
+ return 0;
+}
+
+static void etun_set_multicast_list(struct net_device *dev)
+{
+ /* Nothing sane I can do here */
+ return;
+}
+
+static int etun_ioctl(struct net_device *dev, struct ifreq *rq, int cmd)
+{
+ return -EOPNOTSUPP;
+}
+
+/* Only allow letters and numbers in an etun device name */
+static int is_valid_name(const char *name)
+{
```

```
+ const char *ptr;
+ for (ptr = name; *ptr; ptr++) {
+  if (!isalnum(*ptr))
+   return 0;
+ }
+ return 1;
+}
+
+static struct net_device *etun_alloc(net_t net, const char *name)
+{
+ struct net_device *dev;
+ struct etun_info *info;
+ int err;
+
+ if (!name || !is_valid_name(name))
+  return ERR_PTR(-EINVAL);
+
+ dev = alloc_netdev(sizeof(struct etun_info), name, ether_setup);
+ if (!dev)
+  return ERR_PTR(-ENOMEM);
+
+ info = dev->priv;
+ info->dev = dev;
+ dev->nd_net = net;
+
+ random_ether_addr(dev->dev_addr);
+ dev->tx_queue_len = 0; /* A queue is silly for a loopback device */
+ dev->hard_start_xmit = etun_xmit;
+ dev->get_stats  = etun_get_stats;
+ dev->open  = etun_open;
+ dev->stop  = etun_stop;
+ dev->set_multicast_list = etun_set_multicast_list;
+ dev->do_ioctl  = etun_ioctl;
+ dev->features  = NETIF_F_FRAGLIST
+      | NETIF_F_HIGHDMA
+      | NETIF_F_LLTX;
+ dev->flags  = IFF_BROADCAST | IFF_MULTICAST |IFF_PROMISC;
+ dev->ethtool_ops = &etun_ethtool_ops;
+ dev->destructor  = free_netdev;
+ err = register_netdev(dev);
+ if (err) {
+  free_netdev(dev);
+  dev = ERR_PTR(err);
+  goto out;
+ }
+ netif_carrier_off(dev);
+out:
+ return dev;
```

```
+}
+
+static int etun_alloc_pair(net_t net, const char *name0, const char *name1)
+{
+ struct net_device *dev0, *dev1;
+ struct etun_info *info0, *info1;
+
+ dev0 = etun_alloc(net, name0);
+ if (IS_ERR(dev0)) {
+  return PTR_ERR(dev0);
+ }
+ info0 = dev0->priv;
+
+ dev1 = etun_alloc(net, name1);
+ if (IS_ERR(dev1)) {
+  unregister_netdev(dev0);
+  return PTR_ERR(dev1);
+ }
+ info1 = dev1->priv;
+
+ dev_hold(dev0);
+ dev_hold(dev1);
+ info0->rx_dev = dev1;
+ info1->rx_dev = dev0;
+
+ /* Only place one member of the pair on the list
+  * so I don't confuse list_for_each_entry_safe,
+  * by deleting two list entries at once.
+  */
+ rtnl_lock();
+ list_add(&info0->list, &etun_list);
+ INIT_LIST_HEAD(&info1->list);
+ rtnl_unlock();
+
+ return 0;
+}
+
+static int etun_unregister_pair(struct net_device *dev0)
+{
+ struct etun_info *info0, *info1;
+ struct net_device *dev1;
+
+ ASSERT_RTNL();
+
+ if (!dev0)
+  return -ENODEV;
+
+ info0 = dev0->priv;
```

```
+ dev1  = info0->rx_dev;
+ info1 = dev1->priv;
+
+ /* Drop the cross device references */
+ dev_put(dev0);
+ dev_put(dev1);
+
+ /* Remove from the etun list */
+ if (!list_empty(&info0->list))
+  list_del_init(&info0->list);
+ if (!list_empty(&info1->list))
+  list_del_init(&info1->list);
+
+ unregister_netdevice(dev0);
+ unregister_netdevice(dev1);
+ return 0;
+}
+
+static int etun_noget(char *buffer, struct kernel_param *kp)
+{
+ return 0;
+}
+
+static int etun_newif(const char *val, struct kernel_param *kp)
+{
+ char name0[IFNAMSIZ], name1[IFNAMSIZ];
+ const char *mid;
+ int len, len0, len1;
+ if (!capable(CAP_NET_ADMIN))
+  return -EPERM;
+
+ /* Avoid frustration by removing trailing whitespace */
+ len = strlen(val);
+ while (isspace(val[len - 1]))
+  len--;
+
+ /* Split the string into 2 names */
+ mid = memchr(val, ',', len);
+ if (!mid)
+  return -EINVAL;
+
+ /* Get the first device name */
+ len0 = mid - val;
+ if (len0 > sizeof(name0) - 1)
+  len = sizeof(name0) - 1;
+ strncpy(name0, val, len0);
+ name0[len0] = '\0';
+
```

```
+ /* And the second device name */
+ len1 = len - (len0 + 1);
+ if (len1 > sizeof(name1) - 1)
+  len1 = sizeof(name1) - 1;
+ strncpy(name1, mid + 1, len1);
+ name1[len1] = '\0';
+
+ return etun_alloc_pair(current->nsproxy->net_ns, name0, name1);
+}
+
+static int etun_delif(const char *val, struct kernel_param *kp)
+{
+ char name[IFNAMSIZ];
+ int len;
+ struct net_device *dev;
+ int err;
+ if (!capable(CAP_NET_ADMIN))
+  return -EPERM;
+
+ /* Avoid frustration by removing trailing whitespace */
+ len = strlen(val);
+ while (isspace(val[len - 1]))
+  len--;
+
+ /* Get the device name */
+ if (len > sizeof(name) - 1)
+  return -EINVAL;
+ strncpy(name, val, len);
+ name[len] = '\0';
+
+ /* Double check I don't have strange characters in my device name */
+ if (!is_valid_name(name))
+  return -EINVAL;
+
+ rtnl_lock();
+ err = -ENODEV;
+ dev = __dev_get_by_name(current->nsproxy->net_ns, name);
+ err = etun_unregister_pair(dev);
+ rtnl_unlock();
+ return err;
+}
+
+static int __init etun_init(void)
+{
+ printk(KERN_INFO "etun: %s, %s\n", DRV_DESCRIPTION, DRV_VERSION);
+ printk(KERN_INFO "etun: %s\n", DRV_COPYRIGHT);
+
+ return 0;
```

```
+}
+
+static void etun_cleanup(void)
+{
+ struct etun_info *info, *tmp;
+ rtnl_lock();
+ list_for_each_entry_safe(info, tmp, &etun_list, list) {
+  etun_unregister_pair(info->dev);
+ }
+ rtnl_unlock();
+}
+
+module_param_call(newif, etun_newif, etun_noget, NULL, S_IWUSR);
+module_param_call(delif, etun_delif, etun_noget, NULL, S_IWUSR);
+module_init(etun_init);
+module_exit(etun_cleanup);
+MODULE_DESCRIPTION(DRV_DESCRIPTION);
+MODULE_AUTHOR("Eric Biederman <ebiederm@xmission.com>");
+MODULE_LICENSE("GPL");
--
1.4.4.1.g278f
```

Subject: Re: [PATCH RFC 1/31] net: Add net_namespace_type.h to allow for per network namespace variables.
Posted by Stephen Hemminger on Thu, 25 Jan 2007 20:30:17 GMT
View Forum Message <> Reply to Message

Can all this be a nop if a CONFIG option is not selected?

```
>
> diff --git a/include/linux/net_namespace_type.h b/include/linux/net_namespace_type.h
> new file mode 100644
> index 0000000..8173f59
> --- /dev/null
> +++ b/include/linux/net_namespace_type.h
> @@ -0,0 +1,52 @@
> +/*
> + * Definition of the network namespace reference type
> + * And operations upon it.
```

> + */
> +#ifndef __LINUX_NET_NAMESPACE_TYPE_H
> +#define __LINUX_NET_NAMESPACE_TYPE_H
> +
> +#define __pernetname(name) per_net__##name

Code obfuscation, please don't do that

> +typedef struct {} net_t;

No typedef for this please.

> +
> +#define __data_pernet
> +
> +/* Look up a per network namespace variable */
> +static inline unsigned long __per_net_offset(net_t net) { return 0; }
> +
> +/* Like per_net but returns a pseudo variable address that must be moved
> + * __per_net_offset() bytes before it will point to a real variable.
> + * Useful for static initializers.
> + */
> +#define __per_net_base(name)   __pernetname(name)
> +
> +/* Get the network namespace reference from a per_net variable address */
> +#define net_of(ptr, name) ({ net_t net; ptr; net; })
> +
> +/* Look up a per network namespace variable */
> +#define per_net(name, net) \
> + (*(__per_net_offset(net), &__per_net_base(name)))
> +
> +/* Are the two network namespaces the same */
> +static inline int net_eq(net_t a, net_t b) { return 1; }
> +/* Get an unsigned value appropriate for hashing the network namespace */
> +static inline unsigned int net_hval(net_t net) { return 0; }
> +
> +/* Convert to and from to and from void pointers */
> +static inline void *net_to_voidp(net_t net) { return NULL; }
> +static inline net_t net_from_voidp(void *ptr) { net_t net; return net; }
> +
> +static inline int null_net(net_t net) { return 0; }
> +
> +#define DEFINE_PER_NET(type, name)   \
> + __data_pernet __typeof__(type) __pernetname(name)
> +
> +#define DECLARE_PER_NET(type, name) \
> + extern __typeof__(type) __pernetname(name)
> +

> +#define EXPORT_PER_NET_SYMBOL(var) \
> + EXPORT_SYMBOL(__pernetname(var))
> +#define EXPORT_PER_NET_SYMBOL_GPL(var) \
> + EXPORT_SYMBOL_GPL(__pernetname(var))
> +
> +#endif /* __LINUX_NET_NAMESPACE_TYPE_H */

--
Stephen Hemminger <shemminger@linux-foundation.org>

Subject: Re: [PATCH RFC 1/31] net: Add net_namespace_type.h to allow for per network namespace variables.
Posted by ebiederm on Thu, 25 Jan 2007 20:53:11 GMT
View Forum Message <> Reply to Message

Stephen Hemminger <shemminger@linux-foundation.org> writes:

> Can all this be a nop if a CONFIG option is not selected?

That is exactly what this infrastructure supports.
What you see is the version that comes into effect when
the CONFIG option is not selected.

>From using an empty structure to replace a pointer to make
that a NOP to most of the rest below.

>> diff --git a/include/linux/net_namespace_type.h
> b/include/linux/net_namespace_type.h
>> new file mode 100644
>> index 0000000..8173f59
>> --- /dev/null
>> +++ b/include/linux/net_namespace_type.h
>> @@ -0,0 +1,52 @@
>> +/*
>> + * Definition of the network namespace reference type
>> + * And operations upon it.
>> + */
>> +#ifndef __LINUX_NET_NAMESPACE_TYPE_H
>> +#define __LINUX_NET_NAMESPACE_TYPE_H
>> +
>> +#define __pernetname(name) per_net__##name

>
> Code obfuscation, please don't do that

Single point of making the naming rules, better maintenance.
The basic point is that variables that come through this path
you should not access directly. Tweaking the name enforces that
even in the compiled out state.

>> +typedef struct {} net_t;
>
> No typedef for this please.

Why. That is conventially how we do opaque types in linux
when someone is doing something sophisticated.

You probably want to look down to patch 21 to see what the compiled
in version of these look like.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH RFC 18/31] net: Implment network device movement between namespaces
Posted by Daniel Lezcano on Wed, 28 Feb 2007 14:35:41 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> This patch introduces NETIF_F_NETNS_LOCAL a flag to indicate
> a network device is local to a single network namespace and
> should never be moved. Useful for pseudo devices that we
> need an instance in each network namespace (like the loopback
> device) and for any device we find that cannot handle multiple
> network namespaces so we may trap them in the initial network
> namespace.
>
> This patch introduces the function dev_change_net_namespace
> a function used to move a network device from one network
> namespace to another. To the network device nothing
> special appears to happen, to the components of the network
> stack it appears as if the network device was unregistered
> in the network namespace it is in, and a new device

> was registered in the network namespace the device
> was moved to.
>
> This patch sets up a namespace device destructor that
> upon the exit of a network namespace moves all of the
> movable network devices to the initial network namespace
> so they are not lost.
>
If you:
 * create etun0/etun1
 * create a namespace
 * move etun1 to this namespace
 * rename the etun1 to eth0
 * kill the namespace

the former network device etun1 will be lost if you have in your parent
namespace an interface eth0 because it will conflict.
Perhaps, the first name should be restored before moving the device back
to the initial network namespace ?

  -- Daniel

ps : nice patchset

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH RFC 22/31] net: Add network namespace clone support.
Posted by Daniel Lezcano on Wed, 28 Feb 2007 14:42:08 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> This patch allows you to create a new network namespace
> using sys_clone(...).
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> ---
> include/linux/sched.h | 1 +
> kernel/nsproxy.c | 11 ++++++++++
> net/core/net_namespace.c | 38 ++++++++++++++++++++++++++++++++++++++
> 3 files changed, 50 insertions(+), 0 deletions(-)
>
> diff --git a/include/linux/sched.h b/include/linux/sched.h
> index 4463735..9e0f91a 100644

```
> --- a/include/linux/sched.h
> +++ b/include/linux/sched.h
> @@ -26,6 +26,7 @@
>  #define CLONE_STOPPED  0x02000000 /* Start in stopped state */
>  #define CLONE_NEWUTS  0x04000000 /* New utsname group? */
>  #define CLONE_NEWIPC  0x08000000 /* New ipcs */
> +#define CLONE_NEWNET  0x20000000 /* New network namespace */
>
>  /*
>   * Scheduling policies
> diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
> index 4f3c95a..7861c4c 100644
> --- a/kernel/nsproxy.c
> +++ b/kernel/nsproxy.c
> @@ -20,6 +20,7 @@
>  #include <linux/mnt_namespace.h>
>  #include <linux/utsname.h>
>  #include <linux/pid_namespace.h>
> +#include <net/net_namespace.h>
>
>  struct nsproxy init_nsproxy = INIT_NSPROXY(init_nsproxy);
>  EXPORT_SYMBOL_GPL(init_nsproxy);
> @@ -70,6 +71,7 @@ struct nsproxy *dup_namespaces(struct nsproxy *orig)
>      get_ipc_ns(ns->ipc_ns);
>    if (ns->pid_ns)
>      get_pid_ns(ns->pid_ns);
> +  get_net(ns->net_ns);
>   }
>
>   return ns;
> @@ -117,10 +119,18 @@ int copy_namespaces(int flags, struct task_struct *tsk)
>   if (err)
>    goto out_pid;
>
> + err = copy_net(flags, tsk);
> + if (err)
> +  goto out_net;
> +
>  out:
>   put_nsproxy(old_ns);
>   return err;
>
> +out_net:
> + if (new_ns->pid_ns)
> +  put_pid_ns(new_ns->pid_ns);
> +
>  out_pid:
>   if (new_ns->ipc_ns)
```

>   put_ipc_ns(new_ns->ipc_ns);
> @@ -146,5 +156,6 @@ void free_nsproxy(struct nsproxy *ns)
>   put_ipc_ns(ns->ipc_ns);
>   if (ns->pid_ns)
>   put_pid_ns(ns->pid_ns);
> + put_net(ns->net_ns);
>   kfree(ns);
> }
> diff --git a/net/core/net_namespace.c b/net/core/net_namespace.c
> index 93e3879..cc56105 100644
> --- a/net/core/net_namespace.c
> +++ b/net/core/net_namespace.c
> @@ -175,6 +175,44 @@ out_undo:
>   goto out;
> }
>
> +int copy_net(int flags, struct task_struct *tsk)
> +{
> + net_t old_net = tsk->nsproxy->net_ns;
> + net_t new_net;
> + int err;
> +
> + get_net(old_net);
> +
> + if (!(flags & CLONE_NEWNET))
> + return 0;
> +
> + err = -EPERM;
> + if (!capable(CAP_SYS_ADMIN))
> + goto out;
> +
> + err = -ENOMEM;
> + new_net = net_alloc();
> + if (null_net(new_net))
> + goto out;
> +
> + mutex_lock(&net_mutex);
> + err = setup_net(new_net);
> + if (err)
> + goto out_unlock;
>
Should we "net_free" in case of error ?
> +
> + net_lock();
> + net_list_append(new_net);
> + net_unlock();
> +
> + tsk->nsproxy->net_ns = new_net;

> +
> +out_unlock:
> + mutex_unlock(&net_mutex);
> +out:
> + put_net(old_net);
> + return err;
> +}
> +
> void pernet_modcopy(void *pnetdst, const void *src, unsigned long size)
> {
>   net_t net;
>

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH RFC 22/31] net: Add network namespace clone support.
Posted by ebiederm on Wed, 28 Feb 2007 15:05:13 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

>> +
>> + mutex_lock(&net_mutex);
>> + err = setup_net(new_net);
>> + if (err)
>> + goto out_unlock;
>>
> Should we "net_free" in case of error ?

Oops. Yes we should.
Thanks.

>> + net_lock();
>> + net_list_append(new_net);
>> + net_unlock();
>> +
>> + tsk->nsproxy->net_ns = new_net;
>> +
>> +out_unlock:
>> + mutex_unlock(&net_mutex);
 net_free(new_net);
>> +out:
>> + put_net(old_net);

>> + return err;
>> +}
>> +
>>

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH RFC 18/31] net: Implment network device movement between namespaces
Posted by ebiederm on Wed, 28 Feb 2007 15:12:16 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Eric W. Biederman wrote:
>> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>>
>> This patch introduces NETIF_F_NETNS_LOCAL a flag to indicate
>> a network device is local to a single network namespace and
>> should never be moved. Useful for pseudo devices that we
>> need an instance in each network namespace (like the loopback
>> device) and for any device we find that cannot handle multiple
>> network namespaces so we may trap them in the initial network
>> namespace.
>>
>> This patch introduces the function dev_change_net_namespace
>> a function used to move a network device from one network
>> namespace to another. To the network device nothing
>> special appears to happen, to the components of the network
>> stack it appears as if the network device was unregistered
>> in the network namespace it is in, and a new device
>> was registered in the network namespace the device
>> was moved to.
>>
>> This patch sets up a namespace device destructor that
>> upon the exit of a network namespace moves all of the
>> movable network devices to the initial network namespace
>> so they are not lost.
>>
> If you:
> * create etun0/etun1
> * create a namespace
> * move etun1 to this namespace

> * rename the etun1 to eth0
> * kill the namespace
>
> the former network device etun1 will be lost if you have in your parent
> namespace an interface eth0 because it will conflict.
> Perhaps, the first name should be restored before moving the device back to the
> initial network namespace ?

Restoration of a previous name is no guarantee of anything. Someone may have
renamed the some other interface etun1 in the original network namespace.

However if you look closely at the code. You will discover that if it can't
keep the same name it will rename the device as it switches namespaces.
In particular it will become devN where N is replaced by some unused number.

That is what the pat parameter to dev_change_net_namespace is about.

I'm not exactly thrilled about the generic name but the code should work,
and I don't know if there is a name that makes better sense.

> -- Daniel
>
> ps : nice patchset

Thanks.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC PATCH 0/31] An introduction and A path for merging network namespace work
Posted by Daniel Lezcano on Wed, 28 Feb 2007 16:38:46 GMT
View Forum Message <> Reply to Message

Hi Eric,

Do you plan to propose to merge into mainline your patchset ?

Shouldn't we ask netdev guys what they think about the explicit network
namespace parameter into function you did versus the implicit network
context using the push_net_ns/pop_net_ns function ?

-- Daniel

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC PATCH 0/31] An introduction and A path for merging network namespace work
Posted by ebiederm on Wed, 28 Feb 2007 19:45:10 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Hi Eric,
>
> Do you plan to propose to merge into mainline your patchset ?

I'm hung up at the moment in the sysfs support. Network device renaming is broken in 2.6.21-rc2 at the moment.

Then I would like to see the best of etun/veth merged.

After that yes I would like to propose getting a network namespace implementation into mainline. Which would like be based on the patchset I posted.

> Shouldn't we ask netdev guys what they think about the explicit network
> namespace parameter into function you did versus the implicit network context
> using the push_net_ns/pop_net_ns function ?

It is an important question.

My impression is that in the larger context it seems to be a minor detail.

>From what I have seen of Dmitry's patches and from what I have seen of my own. When not talking functions parameters we make roughly the same set of code changes. For example in my git tree without referencing Dmitry's work, I made roughly the same set of changes to the fib code as he did.

I currently prefer my register_pernet_subsys infrastructure as it is much easier to deal with then gradually accumulating the code into the namespace initialization. There is nothing to clever about it, so we should be fine.

I think my current per_net() function is questionable (it borders on
being too clever) but it is very straight forward to use. In fact
I am probably over using it a bit and making my network namespace data
structures a little too big. If you look at the slab or the
initialization messages you will see I have exceeded page size. Oops.

The big advantage of my pernet work (as opposed to other techniques)
is that it makes compiling out any the effect of my code possible,
and it allows for pernet variables with file scope. If continuing to
support compiling out the pernet code isn't a requirement we could
get less clever solution, and just pass a pointer around.

If we do start passing a pointer around there becomes the question of
how do we support modules. Which is particularly important in the
IPv6 case.

So long as my per_net() function doesn't cause problems I suspect
it is easier to work with than to work without.

As long as we are supporting compiling things out I think using
net_t instead of a raw pointer makes a lot of sense. I really like
the fact that using an empty type when we compile things out so gcc
can just optimize everything away, instead of having to ifdef
everything.

The big practical difference between the approaches comes down to
push_net_ns/pop_net_ns, or doing something else to get the argument
where it belongs. The fact that push_net_ns/pop_net_ns cannot be
universally used in the network stack is a pain. It means that a
function that can used on both the receive and the transmit path has
to have a network namespace argument.

The verification that we are doing the right thing with
push_net_ns/pop_net_ns is also harder as we have to check that we have
the proper value for the entire call chain instead of a check that is
simply local. For doing the conversion it is not a big pain, as we
need to audit the call chain anyway. For dealing with future changes
it could be a problem if to verify every little patch we had to check
the entire call chain.

The biggest argument against explicit parameters that I could see in
earlier conversations was that you could not compile them out. Now
that I have gotten clever and can compile my explicit parameters out
that argument goes away.

The big downside of the explicit parameters is that in some cases
you get a lot of noise patches just to get the value where you need
it. So it more difficult to merge and a little more difficult to
maintain out of the tree.

However for long term maintenance there is a big advantage of explicit
parameters as you only need to test a patch to see if it is locally
correct.

So unless explicit parameters hurt performance my impression is that
they are the better solution.

Dmitry? Daniel? What do you think.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC PATCH 0/31] An introduction and A path for merging network namespace work
Posted by Daniel Lezcano on Thu, 01 Mar 2007 13:53:49 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:

[cut]
> Dmitry? Daniel? What do you think.
>

Hi Eric,

I agree with all the points you presented but I am still 50/50 for both
approaches.

The major argument in favor of the explicit parameter is that it allows
to keep track of the network namespace. But the argument against is it
touchs a lot of files and that can makes the patch less attractive.
Furthermore, everybody should not be aware of what a network namespace
is and should not know how to handle the parameter function.

The implicit network namespace has the advantage to reduce considerably
the impact on the code and to have network developer to be unware of the
network virtualization. But in the same way the network developer should
"forget" in which network namespace he is running. Another point is the
race condition we have while doing network namespace switching and that

can make a contention point.

Concerning the network namespace compile out, that can be done by both approaches.

In the [1/31] patch description, you mention you tryed zero sized structure on x86_64, and the optimization works for all architectures. Does it mean, you tested it with s390, PowerPC, ia64, etc ... ?

IMHO, both approaches are equivalent in terms of pros/cons. Perhaps we should ask netdev@ ...

Regards.

-- Daniel

Subject: Re: [RFC PATCH 0/31] An introduction and A path for merging network namespace work
Posted by Daniel Lezcano on Thu, 01 Mar 2007 14:22:56 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:

[cut]
> Dmitry? Daniel? What do you think.
>

Hi Eric,

I agree with all the points you presented but I am still 50/50 for both approaches.

The major argument in favor of the explicit parameter is that it allows to keep track of the network namespace. But the argument against is it touchs a lot of files and that can makes the patch less attractive. Furthermore, everybody should not be aware of what a network namespace is and should not know how to handle the parameter function.

The implicit network namespace has the advantage to reduce considerably the impact on the code and to have network developer to be unware of the network virtualization. But in the same way the network developer should

"forget" in which network namespace he is running. Another point is the race condition we have while doing network namespace switching and that can make a contention point.

Concerning the network namespace compile out, that can be done by both approaches.

In the [1/31] patch description, you mention you tryed zero sized structure on x86_64, and the optimization works for all architectures. Does it mean, you tested it with s390, PowerPC, ia64, etc ... ?

IMHO, both approaches are equivalent in terms of pros/cons. Perhaps we should ask netdev@ ...

Regards.

-- Daniel

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH RFC 17/31] net: Factor out __dev_alloc_name from dev_alloc_name
Posted by Benjamin Thery on Mon, 05 Mar 2007 15:29:49 GMT
View Forum Message <> Reply to Message

Hello Eric,

See comments about __dev_alloc_name() below.

Regards,
Benjamin

Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> When forcibly changing the network namespace of a device
> I need something that can generate a name for the device
> in the new namespace without overwriting the old name.
>
> __dev_alloc_name provides me that functionality.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

> ---
> net/core/dev.c | 44 ++----------
> 1 files changed, 33 insertions(+), 11 deletions(-)
>
> diff --git a/net/core/dev.c b/net/core/dev.c
> index 32fe905..fc0d2af 100644
> --- a/net/core/dev.c
> +++ b/net/core/dev.c
> @@ -655,9 +655,10 @@ int dev_valid_name(const char *name)
> }
>
> /**
> - * dev_alloc_name - allocate a name for a device
> - * @dev: device
> + * __dev_alloc_name - allocate a name for a device
> + * @net: network namespace to allocate the device name in
>   * @name: name format string
> + * @buf: scratch buffer and result name string
>   *
>   * Passed a format string - eg "lt%d" it will try and find a suitable
>   * id. It scans list of devices to build up a free map, then chooses
> @@ -668,18 +669,13 @@ int dev_valid_name(const char *name)
>   * Returns the number of the unit assigned or a negative errno code.
>   */
>
> -int dev_alloc_name(struct net_device *dev, const char *name)
> +static int __dev_alloc_name(net_t net, const char *name, char buf[IFNAMSIZ])

IMHO the third parameter should be: char *buf
Indeed using "char buf[IFNAMSIZ]" is misleading because later in the
routine sizeof(buf) is used (with an expected result of IFNAMSIZ).
Unfortunately this is no longer the case: sizeof(buf) value is only 4
now (buf is pointer parameter).

This corrupts the registration of network devices (now I understand
why only one of my e1000 showed up after each reboot :).

Also sizeof(buf) should be replaced by IFNAMSIZ in this new routine.
(See below)

> {
>   int i = 0;
> - char buf[IFNAMSIZ];
>   const char *p;
>   const int max_netdevices = 8*PAGE_SIZE;
>   long *inuse;
>   struct net_device *d;
> - net_t net;

> -
> - BUG_ON(null_net(dev->nd_net));
> - net = dev->nd_net;
>
> p = strnchr(name, IFNAMSIZ-1, '%');
> if (p) {
> @@ -713,10 +709,8 @@ int dev_alloc_name(struct net_device *dev, const char *name)
> }
>
> snprintf(buf, sizeof(buf), name, i);

Replace "snprintf(buf, IFNAMSIZ, name, i);" or i will never be
appended to name and all your ethernet devices will all try to
register the name "eth".

There is another occurence of "snprintf(buf, sizeof(buf), ...)" to
replace in the for loop above.

> - if (!__dev_get_by_name(net, buf)) {
> - strlcpy(dev->name, buf, IFNAMSIZ);
> + if (!__dev_get_by_name(net, buf))
> return i;
> - }
>
> /* It is possible to run out of possible slots
> * when the name is long and there isn't enough space left
> @@ -725,6 +719,34 @@ int dev_alloc_name(struct net_device *dev, const char *name)
> return -ENFILE;
> }
>
> +/**
> + * dev_alloc_name - allocate a name for a device
> + * @dev: device
> + * @name: name format string
> + *
> + * Passed a format string - eg "lt%d" it will try and find a suitable
> + * id. It scans list of devices to build up a free map, then chooses
> + * the first empty slot. The caller must hold the dev_base or rtnl lock
> + * while allocating the name and adding the device in order to avoid
> + * duplicates.
> + * Limited to bits_per_byte * page size devices (ie 32K on most platforms).
> + * Returns the number of the unit assigned or a negative errno code.
> + */
> +
> +int dev_alloc_name(struct net_device *dev, const char *name)
> +{
> + char buf[IFNAMSIZ];
> + net_t net;

> + int ret;
> +
> + BUG_ON(null_net(dev->nd_net));
> + net = dev->nd_net;
> + ret = __dev_alloc_name(net, name, buf);
> + if (ret >= 0)
> + strlcpy(dev->name, buf, IFNAMSIZ);
> + return ret;
> +}
> +
>
> /**
>   * dev_change_name - change name of a device

--
B e n j a m i n T h e r y - BULL/DT/Open Software R&D

http://www.bull.com

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC PATCH 0/31] An introduction and A path for merging network namespace work
Posted by ebiederm on Wed, 07 Mar 2007 04:53:27 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Eric W. Biederman wrote:
>
> [cut]
>> Dmitry? Daniel? What do you think.
>>
>
> Hi Eric,
>
> I agree with all the points you presented but I am still 50/50 for both
> approaches.
>
> The major argument in favor of the explicit parameter is that it allows
> to keep track of the network namespace. But the argument against is it
> touchs a lot of files and that can makes the patch less attractive.
> Furthermore, everybody should not be aware of what a network namespace
> is and should not know how to handle the parameter function.

Daniel please look at the patches and see how this interacts. What you describe is how sight unseen one would expect the situation to be however that doesn't seem to match the reality of the code.

Besides which a one time big impact is not a problem, for merging code. It is more of a problem for maintaining out of tree patches.

> The implicit network namespace has the advantage to reduce considerably
> the impact on the code and to have network developer to be unware of the
> network virtualization. But in the same way the network developer should
> "forget" in which network namespace he is running. Another point is the
> race condition we have while doing network namespace switching and that
> can make a contention point.

Actually this is largely false. The implicit parameter does not do more than remove a few patches. The bulk of the changes are the fundamental changes like the arp cache, the routing tables etc. In general all of the basic data structures.

> Concerning the network namespace compile out, that can be done by both
> approaches.

Agreed. My innovation was finding a way to compile out an explicit parameter.

> In the [1/31] patch description, you mention you tryed zero sized
> structure on x86_64, and the optimization works for all architectures.
> Does it mean, you tested it with s390, PowerPC, ia64, etc ... ?

I think my meaning was this: Every where I tested it (i386 (many compiler versions) and x86_64) the parameter was completely optimized out. And even if it isn't the code should still work. Further since passing a void parameter is explicitly allowed in C++ in the right circumstances I expect the code to work on all architectures.

> IMHO, both approaches are equivalent in terms of pros/cons. Perhaps we
> should ask netdev@ ...

Perhaps we should see if we can resolve it ourselves?

Anyway as soon as I get the stupid sysfs support fixed I'm going to look at a veth/etun driver and see if we can get that merged.

Eric