

---

Subject: [PATCH v2 0/4] SUNRPC: pass network namespace context to sockaddr construction routines

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 09:08:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

This patch-set was created in context of clone of git branch:

git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git.

head: devel

v2:

1) Rebased on up-to-date state ("devel" head).

2) patches change logs updated

This is another precursor patch set aimed to remove hard-coded "init\_net" reference from SUNRPC sockaddr conversion and construction routines.

The following series consists of:

---

Stanislav Kinsbursky (4):

SUNRPC: parametrize rpc\_parse\_scope\_id() by network context

SUNRPC: parametrize rpc\_pton6() by network context

SUNRPC: parametrize rpc\_pton() by network context

SUNRPC: parametrize rpc\_uaddr2sockaddr() by network context

```
fs/nfs/dns_resolve.c      |  4 +++-
fs/nfs/nfs4filelayoutdev.c |  2 +-
fs/nfs/nfs4namespace.c    |  2 +-
fs/nfs/super.c            |  4 +++-
fs/nfsd/nfs4state.c       |  2 +-
fs/nfsd/nfsctl.c          |  2 +-
include/linux/sunrpc/clnt.h |  4 +++-
net/sunrpc/addr.c         | 26 ++++++-----
net/sunrpc/rpcb_clnt.c     |  2 +-
net/sunrpc/svcauth_unix.c |  2 +-
10 files changed, 27 insertions(+), 23 deletions(-)
```

--

Signature

---

---

Subject: [PATCH v2 1/4] SUNRPC: parametrize rpc\_parse\_scope\_id() by network context

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 09:09:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Parametrize `rpc_parse_scope_id()` by network context and thus force it's caller to pass in network context instead of using hard-coded "init\_net".

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

---

net/sunrpc/addr.c | 9 +++++----

1 files changed, 5 insertions(+), 4 deletions(-)

diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c

index ee77742..cc83de1 100644

--- a/net/sunrpc/addr.c

+++ b/net/sunrpc/addr.c

@@ -156,8 +156,9 @@ static size\_t rpc\_pton4(const char \*buf, const size\_t buflen,  
{

#if IS\_ENABLED(CONFIG\_IPV6)

-static int rpc\_parse\_scope\_id(const char \*buf, const size\_t buflen,

- const char \*delim, struct sockaddr\_in6 \*sin6)

+static int rpc\_parse\_scope\_id(struct net \*net, const char \*buf,

+ const size\_t buflen, const char \*delim,

+ struct sockaddr\_in6 \*sin6)

{  
 char \*p;  
 size\_t len;  
@@ -177,7 +178,7 @@ static int rpc\_parse\_scope\_id(const char \*buf, const size\_t buflen,  
 unsigned long scope\_id = 0;  
 struct net\_device \*dev;

- dev = dev\_get\_by\_name(&init\_net, p);

+ dev = dev\_get\_by\_name(net, p);

if (dev != NULL) {

scope\_id = dev->ifindex;

dev\_put(dev);

@@ -213,7 +214,7 @@ static size\_t rpc\_pton6(const char \*buf, const size\_t buflen,

if (in6\_pton(buf, buflen, addr, IPV6\_SCOPE\_DELIMITER, &delim) == 0)

return 0;

- if (!rpc\_parse\_scope\_id(buf, buflen, delim, sin6))

+ if (!rpc\_parse\_scope\_id(&init\_net, buf, buflen, delim, sin6))

return 0;

sin6->sin6\_family = AF\_INET6;

---

Subject: [PATCH v2 2/4] SUNRPC: parametrize `rpc_pton6()` by network context  
Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 09:09:11 GMT

Parametrize rpc\_pton6() by network context and thus force it's caller to pass in network context instead of using hard-coded "init\_net".

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

---

net/sunrpc/addr.c | 8 ++++----

1 files changed, 4 insertions(+), 4 deletions(-)

diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c

index cc83de1..18c40a2 100644

--- a/net/sunrpc/addr.c

+++ b/net/sunrpc/addr.c

```
@@ -198,7 +198,7 @@ static int rpc_parse_scope_id(struct net *net, const char *buf,
    return 0;
}
```

```
-static size_t rpc_pton6(const char *buf, const size_t buflen,
```

```
+static size_t rpc_pton6(struct net *net, const char *buf, const size_t buflen,
    struct sockaddr *sap, const size_t salen)
```

```
{
    struct sockaddr_in6 *sin6 = (struct sockaddr_in6 *)sap;
```

```
@@ -214,14 +214,14 @@ static size_t rpc_pton6(const char *buf, const size_t buflen,
    if (in6_pton(buf, buflen, addr, IPV6_SCOPE_DELIMITER, &delim) == 0)
        return 0;
```

```
- if (!rpc_parse_scope_id(&init_net, buf, buflen, delim, sin6))
```

```
+ if (!rpc_parse_scope_id(net, buf, buflen, delim, sin6))
    return 0;
```

```
    sin6->sin6_family = AF_INET6;
    return sizeof(struct sockaddr_in6);
}
```

```
#else
```

```
-static size_t rpc_pton6(const char *buf, const size_t buflen,
```

```
+static size_t rpc_pton6(struct net *net, const char *buf, const size_t buflen,
    struct sockaddr *sap, const size_t salen)
```

```
{
    return 0;
```

```
@@ -249,7 +249,7 @@ size_t rpc_pton(const char *buf, const size_t buflen,
```

```
    for (i = 0; i < buflen; i++)
```

```
        if (buf[i] == ':')
```

```
-    return rpc_pton6(buf, buflen, sap, salen);
```

```
+    return rpc_pton6(&init_net, buf, buflen, sap, salen);
    return rpc_pton4(buf, buflen, sap, salen);
}
```

EXPORT\_SYMBOL\_GPL(rpc\_pton);

---

---

Subject: [PATCH v2 3/4] SUNRPC: parametrize rpc\_pton() by network context

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 09:09:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Parametrize rpc\_pton() by network context and thus force it's callers to pass in network context instead of using hard-coded "init\_net".

Signed-off-by: Stanislav Kinsbursky <[skinsbursky@parallels.com](mailto:skinsbursky@parallels.com)>

```
---
fs/nfs/dns_resolve.c      |  4 ++--
fs/nfs/nfs4filelayoutdev.c |  2 +-
fs/nfs/nfs4namespace.c    |  2 +-
fs/nfs/super.c            |  4 ++--
fs/nfsd/nfsctl.c          |  2 +-
include/linux/sunrpc/clnt.h |  2 +-
net/sunrpc/addr.c          |  7 ++++---
net/sunrpc/svcauth_unix.c |  2 +-
8 files changed, 13 insertions(+), 12 deletions(-)

diff --git a/fs/nfs/dns_resolve.c b/fs/nfs/dns_resolve.c
index 200eb67..7edc62a 100644
--- a/fs/nfs/dns_resolve.c
+++ b/fs/nfs/dns_resolve.c
@@ -20,7 +20,7 @@ static int nfs_dns_resolve_name(struct net *net, char *name, size_t namelen,
    ip_len = dns_query(NULL, name, namelen, NULL, &ip_addr, NULL);
    if (ip_len > 0)
-   ret = rpc_pton(ip_addr, ip_len, sa, salen);
+   ret = rpc_pton(&init_net, ip_addr, ip_len, sa, salen);
    else
        ret = -ESRCH;
    kfree(ip_addr);
@@ -224,7 +224,7 @@ static int nfs_dns_parse(struct cache_detail *cd, char *buf, int buflen)
    len = qword_get(&buf, buf1, sizeof(buf1));
    if (len <= 0)
        goto out;
-   key.addrlen = rpc_pton(buf1, len,
+   key.addrlen = rpc_pton(&init_net, buf1, len,
        (struct sockaddr *)&key.addr,
        sizeof(key.addr));

diff --git a/fs/nfs/nfs4filelayoutdev.c b/fs/nfs/nfs4filelayoutdev.c
index 8ae9190..0d8b952 100644
--- a/fs/nfs/nfs4filelayoutdev.c
```

```

+++ b/fs/nfs/nfs4filelayoutdev.c
@@ -457,7 +457,7 @@ decode_ds_addr(struct xdr_stream *stream, gfp_t gfp_flags)

    INIT_LIST_HEAD(&da->da_node);

- if (!rpc_pton(buf, portstr-buf, (struct sockaddr *)&da->da_addr,
+ if (!rpc_pton(&init_net, buf, portstr-buf, (struct sockaddr *)&da->da_addr,
    sizeof(da->da_addr))) {
    dprintk("%s: error parsing address %s\n", __func__, buf);
    goto out_free_da;
diff --git a/fs/nfs/nfs4namespace.c b/fs/nfs/nfs4namespace.c
index 919a369..48a9acd 100644
--- a/fs/nfs/nfs4namespace.c
+++ b/fs/nfs/nfs4namespace.c
@@ -98,7 +98,7 @@ static size_t nfs_parse_server_name(char *string, size_t len,
{
    ssize_t ret;

- ret = rpc_pton(string, len, sa, salen);
+ ret = rpc_pton(&init_net, string, len, sa, salen);
    if (ret == 0) {
        ret = nfs_dns_resolve_name(server->client->cl_xprt->xprt_net,
            string, len, sa, salen);
diff --git a/fs/nfs/super.c b/fs/nfs/super.c
index e45feb0..b79f2a1 100644
--- a/fs/nfs/super.c
+++ b/fs/nfs/super.c
@@ -1408,7 +1408,7 @@ static int nfs_parse_mount_options(char *raw,
    if (string == NULL)
        goto out_nomem;
    mnt->nfs_server.addrlen =
-   rpc_pton(string, strlen(string),
+   rpc_pton(&init_net, string, strlen(string),
        (struct sockaddr *)
            &mnt->nfs_server.address,
            sizeof(mnt->nfs_server.address));
@@ -1430,7 +1430,7 @@ static int nfs_parse_mount_options(char *raw,
    if (string == NULL)
        goto out_nomem;
    mnt->mount_server.addrlen =
-   rpc_pton(string, strlen(string),
+   rpc_pton(&init_net, string, strlen(string),
        (struct sockaddr *)
            &mnt->mount_server.address,
            sizeof(mnt->mount_server.address));
diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index bb4a11d..3bc8242 100644
--- a/fs/nfsd/nfsctl.c

```

```

+++ b/fs/nfsd/nfsctl.c
@@ -222,7 +222,7 @@ static ssize_t write_unlock_ip(struct file *file, char *buf, size_t size)
    if (qword_get(&buf, fo_path, size) < 0)
        return -EINVAL;

- if (rpc_pton(fo_path, size, sap, salen) == 0)
+ if (rpc_pton(&init_net, fo_path, size, sap, salen) == 0)
    return -EINVAL;

    return nlmsvc_unlock_all_by_ip(sap);
diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h
index e891a8a..7bd114f 100644
--- a/include/linux/sunrpc/clnt.h
+++ b/include/linux/sunrpc/clnt.h
@@ -163,7 +163,7 @@ size_t rpc_peeraddr(struct rpc_clnt *, struct sockaddr *, size_t);
const char *rpc_peeraddr2str(struct rpc_clnt *, enum rpc_display_format_t);

size_t rpc_ntop(const struct sockaddr *, char *, const size_t);
-size_t rpc_pton(const char *, const size_t,
+size_t rpc_pton(struct net *, const char *, const size_t,
    struct sockaddr *, const size_t);
char * rpc_sockaddr2uaddr(const struct sockaddr *, gfp_t);
size_t rpc_uaddr2sockaddr(const char *, const size_t,
diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
index 18c40a2..82b06b7 100644
--- a/net/sunrpc/addr.c
+++ b/net/sunrpc/addr.c
@@ -230,6 +230,7 @@ static size_t rpc_pton6(struct net *net, const char *buf, const size_t
    buflen,

/**
 * rpc_pton - Construct a sockaddr in @sap
+ * @net: applicable network namespace
 * @buf: C string containing presentation format IP address
 * @buflen: length of presentation address in bytes
 * @sap: buffer into which to plant socket address
@@ -242,14 +243,14 @@ static size_t rpc_pton6(struct net *net, const char *buf, const size_t
    buflen,
    * socket address, if successful. Returns zero if an error
    * occurred.
 */
-size_t rpc_pton(const char *buf, const size_t buflen,
+size_t rpc_pton(struct net *net, const char *buf, const size_t buflen,
    struct sockaddr *sap, const size_t salen)
{
    unsigned int i;

    for (i = 0; i < buflen; i++)

```

```

    if (buf[i] == ':')
-   return rpc_pton6(&init_net, buf, buflen, sap, salen);
+   return rpc_pton6(net, buf, buflen, sap, salen);
    return rpc_pton4(buf, buflen, sap, salen);
}
EXPORT_SYMBOL_GPL(rpc_pton);
@@ -340,7 +341,7 @@ size_t rpc_uaddr2sockaddr(const char *uaddr, const size_t uaddr_len,
    port = (unsigned short)((porthi << 8) | portlo);

*c = '\0';
- if (rpc_pton(buf, strlen(buf), sap, salen) == 0)
+ if (rpc_pton(&init_net, buf, strlen(buf), sap, salen) == 0)
    return 0;

switch (sap->sa_family) {
diff --git a/net/sunrpc/svcauth_unix.c b/net/sunrpc/svcauth_unix.c
index 01153ea..2f8c426 100644
--- a/net/sunrpc/svcauth_unix.c
+++ b/net/sunrpc/svcauth_unix.c
@@ -211,7 +211,7 @@ static int ip_map_parse(struct cache_detail *cd,
    len = qword_get(&mesg, buf, mlen);
    if (len <= 0) return -EINVAL;

- if (rpc_pton(buf, len, &address.sa, sizeof(address)) == 0)
+ if (rpc_pton(&init_net, buf, len, &address.sa, sizeof(address)) == 0)
    return -EINVAL;
    switch (address.sa.sa_family) {
    case AF_INET:

```

---

Subject: [PATCH v2 4/4] SUNRPC: parametrize rpc\_uaddr2sockaddr() by network context

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 09:09:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Parametrize rpc\_uaddr2sockaddr() by network context and thus force it's callers to pass in network context instead of using hard-coded "init\_net".

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

```

---
fs/nfsd/nfs4state.c      |  2 +-
include/linux/sunrpc/clnt.h |  2 +-
net/sunrpc/addr.c         |  8 ++++++---
net/sunrpc/rpcb_clnt.c    |  2 +-
4 files changed, 8 insertions(+), 6 deletions(-)

```

```
diff --git a/fs/nfsd/nfs4state.c b/fs/nfsd/nfs4state.c
```

```

index 9ca16dc..56ad8ca 100644
--- a/fs/nfsd/nfs4state.c
+++ b/fs/nfsd/nfs4state.c
@@ -1301,7 +1301,7 @@ gen_callback(struct nfs4_client *clp, struct nfsd4_setclientid *se, struct
svc_r
    else
        goto out_err;

- conn->cb_addrlen = rpc_uaddr2sockaddr(se->se_callback_addr_val,
+ conn->cb_addrlen = rpc_uaddr2sockaddr(&init_net, se->se_callback_addr_val,
        se->se_callback_addr_len,
        (struct sockaddr *)&conn->cb_addr,
        sizeof(conn->cb_addr));
diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h
index 7bd114f..9e754e3 100644
--- a/include/linux/sunrpc/clnt.h
+++ b/include/linux/sunrpc/clnt.h
@@ -166,7 +166,7 @@ size_t rpc_ntop(const struct sockaddr *, char *, const size_t);
size_t rpc_pton(struct net *, const char *, const size_t,
        struct sockaddr *, const size_t);
char * rpc_sockaddr2uaddr(const struct sockaddr *, gfp_t);
-size_t rpc_uaddr2sockaddr(const char *, const size_t,
+size_t rpc_uaddr2sockaddr(struct net *, const char *, const size_t,
        struct sockaddr *, const size_t);

static inline unsigned short rpc_get_port(const struct sockaddr *sap)
diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
index 82b06b7..d11418f 100644
--- a/net/sunrpc/addr.c
+++ b/net/sunrpc/addr.c
@@ -297,6 +297,7 @@ char *rpc_sockaddr2uaddr(const struct sockaddr *sap, gfp_t gfp_flags)

/**
 * rpc_uaddr2sockaddr - convert a universal address to a socket address.
+ * @net: applicable network namespace
 * @uaddr: C string containing universal address to convert
 * @uaddr_len: length of universal address string
 * @sap: buffer into which to plant socket address
@@ -308,8 +309,9 @@ char *rpc_sockaddr2uaddr(const struct sockaddr *sap, gfp_t gfp_flags)
 * Returns the size of the socket address if successful; otherwise
 * zero is returned.
 */
-size_t rpc_uaddr2sockaddr(const char *uaddr, const size_t uaddr_len,
-        struct sockaddr *sap, const size_t salen)
+size_t rpc_uaddr2sockaddr(struct net *net, const char *uaddr,
+        const size_t uaddr_len, struct sockaddr *sap,
+        const size_t salen)
{

```



```

char *c, buf[RPCBIND_MAXUADDRLEN + sizeof('\0')];
unsigned long portlo, porthi;
@@ -341,7 +343,7 @@ size_t rpc_uaddr2sockaddr(const char *uaddr, const size_t uaddr_len,
port = (unsigned short)((porthi << 8) | portlo);

*c = '\0';
- if (rpc_pton(&init_net, buf, strlen(buf), sap, salen) == 0)
+ if (rpc_pton(net, buf, strlen(buf), sap, salen) == 0)
return 0;

switch (sap->sa_family) {
diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
index 6d6a84f..7c6fac0 100644
--- a/net/sunrpc/rpcb_clnt.c
+++ b/net/sunrpc/rpcb_clnt.c
@@ -934,7 +934,7 @@ static int rpcb_dec_getaddr(struct rpc_rqst *req, struct xdr_stream *xdr,
dprintk("RPC: %5u RPCB_%s reply: %s\n", task->tk_pid,
task->tk_msg.rpc_proc->p_name, (char *)p);

- if (rpc_uaddr2sockaddr((char *)p, len, sap, sizeof(address)) == 0)
+ if (rpc_uaddr2sockaddr(&init_net, (char *)p, len, sap, sizeof(address)) == 0)
goto out_fail;
rpcb->r_port = rpc_get_port(sap);

```

---