

---

Subject: Re: [PATCH 12/12] network namespace: L2 and L3 intro  
Posted by [Cedric Le Goater](#) on Thu, 07 Dec 2006 09:34:24 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I have issues with the hunks below. they don't apply :(

Can I have an updated version ? and I'll release the full patchset ASAP.

Thanks,

C.

```
> --- linux-2.6.19-rc6-mm2.orig/net/core/net_namespace.c
> +++ linux-2.6.19-rc6-mm2/net/core/net_namespace.c
> @@ -32,14 +32,20 @@ struct net_namespace init_net_ns = {
>
> /*
> * Clone a new ns copying an original net ns, setting refcount to 1
> + * @level: level of namespace to create
> * @old_ns: namespace to clone
> - * Return NULL on error (failure to kmalloc), new ns otherwise
> + * Return ERR_PTR on error, new ns otherwise
> */
> -static struct net_namespace *clone_net_ns(struct net_namespace *old_ns)
> +static struct net_namespace *clone_net_ns(unsigned int level,
> + struct net_namespace *old_ns)
> {
> struct net_namespace *ns, *orig_ns;
> static unsigned int id = 1;
>
> + /* level 3 namespaces are incomplete in order to have childs */
> + if (current_net_ns->level == NET_NS_LEVEL3)
> + return ERR_PTR(-EPERM);
> +
> ns = kmalloc(sizeof(struct net_namespace), GFP_KERNEL);
> if (!ns)
> return NULL;
> @@ -59,20 +65,25 @@ static struct net_namespace *clone_net_n
> spin_unlock_irq(&net_ns_list_lock);
>
> orig_ns = push_net_ns(ns);
> + if (level == NET_NS_LEVEL2) {
> #ifdef CONFIG_IP_MULTIPLE_TABLES
> - INIT_LIST_HEAD(&ns->fib_rules_ops_list);
> + INIT_LIST_HEAD(&ns->fib_rules_ops_list);
> #endif
> - if (ip_fib_struct_init())
> - goto out_fib4;
```

```

> + if (ip_fib_struct_init())
> + goto out_fib4;
> + }
> + ns->level = level;
> if (loopback_init())
> goto out_loopback;
> put_net_ns(orig_ns);
> - printk(KERN_DEBUG "NET_NS: created new netcontext %p (%u) for %s "
> - "(pid=%d)\n", ns, ns->id, current->comm, current->tgid);
> + printk(KERN_DEBUG "NET_NS: created new netcontext %p (%u), level %u, "
> + "for %s (pid=%d)\n", ns, ns->id, (ns->level == NET_NS_LEVEL2) ?
> + 2 : 3, current->comm, current->tgid);
> return ns;
>
> out_loopback:
> - ip_fib_struct_cleanup();
> + if (level == NET_NS_LEVEL2)
> + ip_fib_struct_cleanup();
> out_fib4:
> put_net_ns(orig_ns);
> BUG_ON(atomic_read(&ns->kref.refcount) != 1);
> @@ -86,17 +97,22 @@ out_fib4:
> int unshare_net_ns(unsigned long unshare_flags,
> struct net_namespace **new_net)
> {
> - if (unshare_flags & NS_NET) {
> + unsigned int level;
> +
> + if (unshare_flags & (NS_NET2 | NS_NET3)) {
> if (!capable(CAP_SYS_ADMIN))
> return -EPERM;
>
> - *new_net = clone_net_ns(current->nsproxy->net_ns);
> - if (!*new_net)
> - return -ENOMEM;
> + level = (unshare_flags & NS_NET2) ? NET_NS_LEVEL2 :
> + NET_NS_LEVEL3;
> + *new_net = clone_net_ns(level, current->nsproxy->net_ns);
> + if (IS_ERR(*new_net))
> + return PTR_ERR(*new_net);
> }
>
> return 0;
> }
> +EXPORT_SYMBOL(unshare_net_ns);
>
> int copy_net_ns(int flags, struct task_struct *tsk)
> {

```

```
> @@ -157,7 +173,8 @@ void free_net_ns(struct kref *kref)
> ns, atomic_read(&ns->kref.refcount));
> return;
> }
> - ip_fib_struct_cleanup();
> + if (ns->level == NET_NS_LEVEL2)
> + ip_fib_struct_cleanup();
> printk(KERN_DEBUG "NET_NS: netcontext %p (%u) destroyed\n",
> ns, ns->id);
> kfree(ns);
>


---


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

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

---