
Subject: [PATCH][NETNS] Make ifindex generation per-namespace

Posted by [Pavel Emelianov](#) on Tue, 09 Oct 2007 12:19:25 GMT

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

Currently indexes for netdevices come sequentially one by one, and the same stays true even for devices that are created for namespaces.

Side effects of this are:

- * lo device has not 1 index in a namespace. This may break some userspace that relies on it (and AFAIR something really broke in OpenVZ VEs without this);
- * after some time namespaces will have devices with indexes like 1000000 or similar. This might be confusing for a human (tools will not mind).

So move the (currently "global" and static) ifindex variable on the struct net, making the indexes allocation look more like on a standalone machine.

Moreover - when we have indexes intersect between namespaces, we may catch more BUGs in the future related to "wrong device was found for a given index".

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
```

```
index 93aa87d..83a18d0 100644
```

```
--- a/include/net/net_namespace.h
```

```
+++ b/include/net/net_namespace.h
```

```
@@ -29,6 +29,8 @@ struct net {  
    struct list_head dev_base_head;  
    struct hlist_head *dev_name_head;  
    struct hlist_head *dev_index_head;
```

```
+
```

```
+ int ifindex;
```

```
};
```

```
#ifdef CONFIG_NET
```

```
diff --git a/net/core/dev.c b/net/core/dev.c
```

```
index e7e728a..a08ed8c 100644
```

```
--- a/net/core/dev.c
```

```
+++ b/net/core/dev.c
```

```
@@ -3443,12 +3443,11 @@ int dev_ioctl(struct net *net, unsigned  
    */
```

```
static int dev_new_index(struct net *net)
```

```
{  
- static int ifindex;  
  for (;;) {  
-   if (++ifindex <= 0)  
-     ifindex = 1;  
-   if (!__dev_get_by_index(net, ifindex))  
-     return ifindex;  
+   if (++net->ifindex <= 0)  
+     net->ifindex = 1;  
+   if (!__dev_get_by_index(net, net->ifindex))  
+     return net->ifindex;  
  }  
}
```
