

---

Subject: [PATCH 14/16] net: Factor out \_\_dev\_alloc\_name from dev\_alloc\_name  
Posted by [ebiederm](#) on Sat, 08 Sep 2007 21:36:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

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](mailto:ebiederm@xmission.com)>

---

net/core/dev.c | 48 ++++++-----  
1 files changed, 35 insertions(+), 13 deletions(-)

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

index c51cf40..53cdb64 100644

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

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

@@ -739,9 +739,10 @@ 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

@@ -752,18 +753,13 @@ 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(struct net \*net, const char \*name, char \*buf)

{

- int i = 0;

- char buf[IFNAMSIZ];

- const char \*p;

- const int max\_netdevices = 8\*PAGE\_SIZE;

- long \*inuse;

- struct net\_device \*d;

- struct net \*net;

-

- BUG\_ON(!dev->nd\_net);

```

- net = dev->nd_net;

p = strchr(name, IFNAMSIZ-1, '%');
if (p) {
@@ -787,7 +783,7 @@ int dev_alloc_name(struct net_device *dev, const char *name)
    continue;

    /* avoid cases where sscanf is not exact inverse of printf */
- snprintf(buf, sizeof(buf), name, i);
+ snprintf(buf, IFNAMSIZ, name, i);
    if (!strcmp(buf, dev->name, IFNAMSIZ))
        set_bit(i, inuse);
}
@@ -796,11 +792,9 @@ int dev_alloc_name(struct net_device *dev, const char *name)
    free_page((unsigned long) inuse);
}

- snprintf(buf, sizeof(buf), name, i);
- if (!__dev_get_by_name(net, buf)) {
-     strcpy(dev->name, buf, IFNAMSIZ);
+ snprintf(buf, IFNAMSIZ, name, i);
+ 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
@@ -809,6 +803,34 @@ 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];

```

```
+ struct net *net;
+ int ret;
+
+ BUG_ON(!dev->nd_net);
+ net = dev->nd_net;
+ ret = __dev_alloc_name(net, name, buf);
+ if (ret >= 0)
+   strcpy(dev->name, buf, IFNAMSIZ);
+ return ret;
+}
+

/**
 * dev_change_name - change name of a device
--
1.5.3.rc6.17.g1911
```

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---