Subject: [PATCH 0/11 net-2.6.16] unify control socket usage for protocols
Posted by den on Fri, 28 Mar 2008 09:04:31 GMT
View Forum Message <> Reply to Message

Hello, Dave!

This set provides a unification for control socket usage/creation for
all protocols. The idea is to use sock rather than socket all around the
protocols. (This has been partly done previously for ICMP/IPv6 staff).

Additionally, the creation of such a sockets now performed uniformly via
inet_csk_ctl_sock_create.

After that TCP control socket is created and used on a per/namespace
basis.

Regards,
Den

P.S. This set includes the patch discussed yesterday with Vlad Yasevich.

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

Subject: [PATCH 1/11 net-2.6.26] [TCP]: Replace socket with sock for reset sending.
Posted by den on Fri, 28 Mar 2008 09:08:48 GMT
View Forum Message <> Reply to Message

This is more effective (less derefferences on fast paths). Additionally,
the approach is unified to one used in ICMP.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
net/ipv4/tcp_ipv4.c | 10 ++++++----
1 files changed, 6 insertions(+), 4 deletions(-)

diff --git a/net/ipv4/tcp_ipv4.c b/net/ipv4/tcp_ipv4.c
index 2a5881c..9d47870 100644
--- a/net/ipv4/tcp_ipv4.c
+++ b/net/ipv4/tcp_ipv4.c
@@ -89,7 +89,7 @@ int sysctl_tcp_low_latency __read_mostly;
#define ICMP_MIN_LENGTH 8

/* Socket used for sending RSTs */

```
-static struct socket *tcp_socket __read_mostly;
+static struct sock *tcp_sock __read_mostly;

 void tcp_v4_send_check(struct sock *sk, int len, struct sk_buff *skb);

@@ -598,7 +598,7 @@ static void tcp_v4_send_reset(struct sock *sk, struct sk_buff *skb)
          sizeof(struct tcphdr), IPPROTO_TCP, 0);
  arg.csumoffset = offsetof(struct tcphdr, check) / 2;

- ip_send_reply(tcp_socket->sk, skb, &arg, arg.iov[0].iov_len);
+ ip_send_reply(tcp_sock, skb, &arg, arg.iov[0].iov_len);

  TCP_INC_STATS_BH(TCP_MIB_OUTSEGS);
  TCP_INC_STATS_BH(TCP_MIB_OUTRSTS);
@@ -693,7 +693,7 @@ static void tcp_v4_send_ack(struct tcp_timewait_sock *twsk,
  if (twsk)
   arg.bound_dev_if = twsk->tw_sk.tw_bound_dev_if;

- ip_send_reply(tcp_socket->sk, skb, &arg, arg.iov[0].iov_len);
+ ip_send_reply(tcp_sock, skb, &arg, arg.iov[0].iov_len);

  TCP_INC_STATS_BH(TCP_MIB_OUTSEGS);
 }
@@ -2493,9 +2493,11 @@ struct proto tcp_prot = {

 void __init tcp_v4_init(void)
 {
- if (inet_csk_ctl_sock_create(&tcp_socket, PF_INET, SOCK_RAW,
+ struct socket *__tcp_socket;
+ if (inet_csk_ctl_sock_create(&__tcp_socket, PF_INET, SOCK_RAW,
          IPPROTO_TCP) < 0)
   panic("Failed to create the TCP control socket.\n");
+ tcp_sock = __tcp_socket->sk;
 }

 EXPORT_SYMBOL(ipv4_specific);
--
1.5.3.rc5
```

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

---

Subject: [PATCH 2/11 net-2.6.26] [DCCP]: Replace socket with sock for reset sending.

Replace dccp_v(4|6)_ctl_socket with sock to unify a code with TCP/ICMP.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/dccp/ipv4.c | 16 +++++++++-------
 net/dccp/ipv6.c | 10 ++++++----
 2 files changed, 15 insertions(+), 11 deletions(-)

```
diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
index 4ca8b0c..79a9a4a 100644
--- a/net/dccp/ipv4.c
+++ b/net/dccp/ipv4.c
@@ -36,7 +36,7 @@
  * the Out-of-the-blue (OOTB) packets. A control sock will be created
  * for this socket at the initialization time.
  */
-static struct socket *dccp_v4_ctl_socket;
+static struct sock *dccp_v4_ctl_sk;

 int dccp_v4_connect(struct sock *sk, struct sockaddr *uaddr, int addr_len)
 {
@@ -514,11 +514,11 @@ static void dccp_v4_ctl_send_reset(struct sock *sk, struct sk_buff
*rxskb)
  if (rxskb->rtable->rt_type != RTN_LOCAL)
   return;

- dst = dccp_v4_route_skb(dccp_v4_ctl_socket->sk, rxskb);
+ dst = dccp_v4_route_skb(dccp_v4_ctl_sk, rxskb);
  if (dst == NULL)
   return;

- skb = dccp_ctl_make_reset(dccp_v4_ctl_socket, rxskb);
+ skb = dccp_ctl_make_reset(dccp_v4_ctl_sk->sk_socket, rxskb);
  if (skb == NULL)
   goto out;

@@ -527,10 +527,10 @@ static void dccp_v4_ctl_send_reset(struct sock *sk, struct sk_buff
*rxskb)
         rxiph->daddr);
  skb->dst = dst_clone(dst);

- bh_lock_sock(dccp_v4_ctl_socket->sk);
- err = ip_build_and_send_pkt(skb, dccp_v4_ctl_socket->sk,
+ bh_lock_sock(dccp_v4_ctl_sk);
+ err = ip_build_and_send_pkt(skb, dccp_v4_ctl_sk,
         rxiph->daddr, rxiph->saddr, NULL);
```

```
- bh_unlock_sock(dccp_v4_ctl_socket->sk);
+ bh_unlock_sock(dccp_v4_ctl_sk);

  if (net_xmit_eval(err) == 0) {
   DCCP_INC_STATS_BH(DCCP_MIB_OUTSEGS);
@@ -994,6 +994,7 @@ static struct inet_protosw dccp_v4_protosw = {

 static int __init dccp_v4_init(void)
 {
+ struct socket *socket;
  int err = proto_register(&dccp_v4_prot, 1);

  if (err != 0)
@@ -1005,10 +1006,11 @@ static int __init dccp_v4_init(void)

  inet_register_protosw(&dccp_v4_protosw);

- err = inet_csk_ctl_sock_create(&dccp_v4_ctl_socket, PF_INET,
+ err = inet_csk_ctl_sock_create(&socket, PF_INET,
            SOCK_DCCP, IPPROTO_DCCP);
  if (err)
   goto out_unregister_protosw;
+ dccp_v4_ctl_sk = socket->sk;
 out:
  return err;
 out_unregister_protosw:
diff --git a/net/dccp/ipv6.c b/net/dccp/ipv6.c
index 2fec1af..249640d 100644
--- a/net/dccp/ipv6.c
+++ b/net/dccp/ipv6.c
@@ -34,7 +34,7 @@
 #include "feat.h"

 /* Socket used for sending RSTs and ACKs */
-static struct socket *dccp_v6_ctl_socket;
+static struct sock *dccp_v6_ctl_sk;

 static struct inet_connection_sock_af_ops dccp_ipv6_mapped;
 static struct inet_connection_sock_af_ops dccp_ipv6_af_ops;
@@ -303,7 +303,7 @@ static void dccp_v6_ctl_send_reset(struct sock *sk, struct sk_buff *rxskb)
  if (!ipv6_unicast_destination(rxskb))
   return;

- skb = dccp_ctl_make_reset(dccp_v6_ctl_socket, rxskb);
+ skb = dccp_ctl_make_reset(dccp_v6_ctl_sk->sk_socket, rxskb);
  if (skb == NULL)
   return;
```

```
@@ -324,7 +324,7 @@ static void dccp_v6_ctl_send_reset(struct sock *sk, struct sk_buff *rxskb)
  /* sk = NULL, but it is safe for now. RST socket required. */
  if (!ip6_dst_lookup(NULL, &skb->dst, &fl)) {
   if (xfrm_lookup(&skb->dst, &fl, NULL, 0) >= 0) {
-   ip6_xmit(dccp_v6_ctl_socket->sk, skb, &fl, NULL, 0);
+   ip6_xmit(dccp_v6_ctl_sk, skb, &fl, NULL, 0);
    DCCP_INC_STATS_BH(DCCP_MIB_OUTSEGS);
    DCCP_INC_STATS_BH(DCCP_MIB_OUTRSTS);
    return;
@@ -1176,6 +1176,7 @@ static struct inet_protosw dccp_v6_protosw = {

 static int __init dccp_v6_init(void)
 {
+ struct socket *socket;
  int err = proto_register(&dccp_v6_prot, 1);

  if (err != 0)
@@ -1187,10 +1188,11 @@ static int __init dccp_v6_init(void)

  inet6_register_protosw(&dccp_v6_protosw);

- err = inet_csk_ctl_sock_create(&dccp_v6_ctl_socket, PF_INET6,
+ err = inet_csk_ctl_sock_create(&socket, PF_INET6,
            SOCK_DCCP, IPPROTO_DCCP);
  if (err != 0)
   goto out_unregister_protosw;
+ dccp_v6_ctl_sk = socket->sk;
 out:
  return err;
 out_unregister_protosw:
--
1.5.3.rc5
```

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

---

Subject: [PATCH 3/11 net-2.6.26] [DCCP]: dccp_v(4|6)_ctl_socket is leaked.
Posted by den on Fri, 28 Mar 2008 09:08:50 GMT
View Forum Message <> Reply to Message

This seems a purism as module can't be unloaded, but though if cleanup method
is present it should be correct and clean all staff created.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
 net/dccp/ipv4.c |    1 +
 net/dccp/ipv6.c |    1 +
 2 files changed, 2 insertions(+), 0 deletions(-)

diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
index 79a9a4a..4ee34db 100644
--- a/net/dccp/ipv4.c
+++ b/net/dccp/ipv4.c
@@ -1023,6 +1023,7 @@ out_proto_unregister:

 static void __exit dccp_v4_exit(void)
 {
+ sock_release(dccp_v4_ctl_sk->sk_socket);
  inet_unregister_protosw(&dccp_v4_protosw);
  inet_del_protocol(&dccp_v4_protocol, IPPROTO_DCCP);
  proto_unregister(&dccp_v4_prot);
diff --git a/net/dccp/ipv6.c b/net/dccp/ipv6.c
index 249640d..85665d5 100644
--- a/net/dccp/ipv6.c
+++ b/net/dccp/ipv6.c
@@ -1205,6 +1205,7 @@ out_unregister_proto:

 static void __exit dccp_v6_exit(void)
 {
+ sock_release(dccp_v6_ctl_sk->sk_socket);
  inet6_del_protocol(&dccp_v6_protocol, IPPROTO_DCCP);
  inet6_unregister_protosw(&dccp_v6_protosw);
  proto_unregister(&dccp_v6_prot);
--
1.5.3.rc5


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

Subject: [PATCH 4/11 net-2.6.26] [SCTP]: Use inet_csk_ctl_sock_create for control socket creation.

Posted by den on Fri, 28 Mar 2008 09:08:51 GMT

View Forum Message <> Reply to Message

sk->sk_proc->(un)hash is noop right now, so the unification is correct.

```
Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/sctp/protocol.c |    7 ++-----
 1 files changed, 2 insertions(+), 5 deletions(-)
```

```
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 5aea911..5b0323c 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -680,16 +680,13 @@ static int sctp_ctl_sock_init(void)
  else
   family = PF_INET;

- err = sock_create_kern(family, SOCK_SEQPACKET, IPPROTO_SCTP,
-          &sctp_ctl_socket);
+ err = inet_csk_ctl_sock_create(&sctp_ctl_socket,
+           family, SOCK_SEQPACKET, IPPROTO_SCTP);
  if (err < 0) {
   printk(KERN_ERR
          "SCTP: Failed to create the SCTP control socket.\n");
   return err;
  }
- sctp_ctl_socket->sk->sk_allocation = GFP_ATOMIC;
- inet_sk(sctp_ctl_socket->sk)->uc_ttl = -1;
-
  return 0;
 }

--
1.5.3.rc5
```

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

---

Subject: [PATCH 5/11 net-2.6.26] [SCTP]: Replace socket with sock for sctp_ctl_sock.

Posted by den on Fri, 28 Mar 2008 09:08:52 GMT

View Forum Message <> Reply to Message

```
Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/sctp/protocol.c |   12 +++++++-----
 1 files changed, 7 insertions(+), 5 deletions(-)

diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 5b0323c..2715aca 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -74,7 +74,7 @@ DEFINE_SPINLOCK(sctp_assocs_id_lock);
```

```
  * the Out-of-the-blue (OOTB) packets.  A control sock will be created
  * for this socket at the initialization time.
  */
-static struct socket *sctp_ctl_socket;
+static struct sock *sctp_ctl_sock;

 static struct sctp_pf *sctp_pf_inet6_specific;
 static struct sctp_pf *sctp_pf_inet_specific;
@@ -91,7 +91,7 @@ int sysctl_sctp_wmem[3];
 /* Return the address of the control sock. */
 struct sock *sctp_get_ctl_sock(void)
 {
- return sctp_ctl_socket->sk;
+ return sctp_ctl_sock;
 }

 /* Set up the proc fs entry for the SCTP protocol. */
@@ -674,19 +674,21 @@ static int sctp_ctl_sock_init(void)
 {
  int err;
  sa_family_t family;
+ struct socket *socket;

  if (sctp_get_pf_specific(PF_INET6))
   family = PF_INET6;
  else
   family = PF_INET;

- err = inet_csk_ctl_sock_create(&sctp_ctl_socket,
+ err = inet_csk_ctl_sock_create(&socket,
           family, SOCK_SEQPACKET, IPPROTO_SCTP);
  if (err < 0) {
   printk(KERN_ERR
          "SCTP: Failed to create the SCTP control socket.\n");
   return err;
  }
+ sctp_ctl_sock = socket->sk;
  return 0;
 }

@@ -1284,7 +1286,7 @@ err_v6_add_protocol:
  sctp_v6_del_protocol();
 err_add_protocol:
  sctp_v4_del_protocol();
- sock_release(sctp_ctl_socket);
+ sock_release(sctp_ctl_sock->sk_socket);
 err_ctl_sock_init:
  sctp_v6_protosw_exit();
```

```
 err_v6_protosw_init:
@@ -1328,7 +1330,7 @@ SCTP_STATIC __exit void sctp_exit(void)
  sctp_v4_del_protocol();

  /* Free the control endpoint.  */
- sock_release(sctp_ctl_socket);
+ sock_release(sctp_ctl_sock->sk_socket);

  /* Free protosw registrations */
  sctp_v6_protosw_exit();
--
1.5.3.rc5
```

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

---

Subject: [PATCH 6/11 net-2.6.26] [INET]: Let inet_csk_ctl_sock_create return sock rather than socket.

Posted by den on Fri, 28 Mar 2008 09:08:53 GMT
View Forum Message <> Reply to Message

All upper protocol layers are already use sock internally.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 include/net/inet_connection_sock.h |    2 +-
 net/dccp/ipv4.c                    |    4 +---
 net/dccp/ipv6.c                    |    4 +---
 net/ipv4/inet_connection_sock.c    |   12 +++++++-----
 net/ipv4/tcp_ipv4.c                |    4 +---
 net/ipv6/tcp_ipv6.c                |   11 ++++++-------
 net/sctp/protocol.c                |    4 +---
 7 files changed, 16 insertions(+), 25 deletions(-)

diff --git a/include/net/inet_connection_sock.h b/include/net/inet_connection_sock.h
index f00f057..a5d4f05 100644
--- a/include/net/inet_connection_sock.h
+++ b/include/net/inet_connection_sock.h
@@ -327,7 +327,7 @@ extern void inet_csk_listen_stop(struct sock *sk);

 extern void inet_csk_addr2sockaddr(struct sock *sk, struct sockaddr *uaddr);

-extern int inet_csk_ctl_sock_create(struct socket **sock,
+extern int inet_csk_ctl_sock_create(struct sock **sk,
         unsigned short family,
```

```
        unsigned short type,
        unsigned char protocol);
diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
index 4ee34db..52566ad 100644
--- a/net/dccp/ipv4.c
+++ b/net/dccp/ipv4.c
@@ -994,7 +994,6 @@ static struct inet_protosw dccp_v4_protosw = {

 static int __init dccp_v4_init(void)
 {
- struct socket *socket;
  int err = proto_register(&dccp_v4_prot, 1);

  if (err != 0)
@@ -1006,11 +1005,10 @@ static int __init dccp_v4_init(void)

  inet_register_protosw(&dccp_v4_protosw);

- err = inet_csk_ctl_sock_create(&socket, PF_INET,
+ err = inet_csk_ctl_sock_create(&dccp_v4_ctl_sk, PF_INET,
            SOCK_DCCP, IPPROTO_DCCP);
  if (err)
   goto out_unregister_protosw;
- dccp_v4_ctl_sk = socket->sk;
 out:
  return err;
 out_unregister_protosw:
diff --git a/net/dccp/ipv6.c b/net/dccp/ipv6.c
index 85665d5..a2ffc98 100644
--- a/net/dccp/ipv6.c
+++ b/net/dccp/ipv6.c
@@ -1176,7 +1176,6 @@ static struct inet_protosw dccp_v6_protosw = {

 static int __init dccp_v6_init(void)
 {
- struct socket *socket;
  int err = proto_register(&dccp_v6_prot, 1);

  if (err != 0)
@@ -1188,11 +1187,10 @@ static int __init dccp_v6_init(void)

  inet6_register_protosw(&dccp_v6_protosw);

- err = inet_csk_ctl_sock_create(&socket, PF_INET6,
+ err = inet_csk_ctl_sock_create(&dccp_v6_ctl_sk, PF_INET6,
            SOCK_DCCP, IPPROTO_DCCP);
  if (err != 0)
   goto out_unregister_protosw;
```

```
- dccp_v6_ctl_sk = socket->sk;
 out:
  return err;
 out_unregister_protosw:
diff --git a/net/ipv4/inet_connection_sock.c b/net/ipv4/inet_connection_sock.c
index a7fcaf2..2cb3552 100644
--- a/net/ipv4/inet_connection_sock.c
+++ b/net/ipv4/inet_connection_sock.c
@@ -651,19 +651,21 @@ void inet_csk_addr2sockaddr(struct sock *sk, struct sockaddr *uaddr)

 EXPORT_SYMBOL_GPL(inet_csk_addr2sockaddr);

-int inet_csk_ctl_sock_create(struct socket **sock, unsigned short family,
+int inet_csk_ctl_sock_create(struct sock **sk, unsigned short family,
         unsigned short type, unsigned char protocol)
 {
- int rc = sock_create_kern(family, type, protocol, sock);
+ struct socket *socket;
+ int rc = sock_create_kern(family, type, protocol, &socket);

   if (rc == 0) {
-   (*sock)->sk->sk_allocation = GFP_ATOMIC;
-   inet_sk((*sock)->sk)->uc_ttl = -1;
+   *sk = socket->sk;
+   (*sk)->sk_allocation = GFP_ATOMIC;
+   inet_sk(*sk)->uc_ttl = -1;
    /*
     * Unhash it so that IP input processing does not even see it,
     * we do not wish this socket to see incoming packets.
     */
-   (*sock)->sk->sk_prot->unhash((*sock)->sk);
+   (*sk)->sk_prot->unhash(*sk);
   }
   return rc;
 }
diff --git a/net/ipv4/tcp_ipv4.c b/net/ipv4/tcp_ipv4.c
index 9d47870..2ee7fb9 100644
--- a/net/ipv4/tcp_ipv4.c
+++ b/net/ipv4/tcp_ipv4.c
@@ -2493,11 +2493,9 @@ struct proto tcp_prot = {

 void __init tcp_v4_init(void)
 {
- struct socket *__tcp_socket;
- if (inet_csk_ctl_sock_create(&__tcp_socket, PF_INET, SOCK_RAW,
+ if (inet_csk_ctl_sock_create(&tcp_sock, PF_INET, SOCK_RAW,
          IPPROTO_TCP) < 0)
    panic("Failed to create the TCP control socket.\n");
```

```
- tcp_sock = __tcp_socket->sk;
 }

 EXPORT_SYMBOL(ipv4_specific);
diff --git a/net/ipv6/tcp_ipv6.c b/net/ipv6/tcp_ipv6.c
index 323c7e0..fdb670f 100644
--- a/net/ipv6/tcp_ipv6.c
+++ b/net/ipv6/tcp_ipv6.c
@@ -2202,16 +2202,13 @@ static struct inet_protosw tcpv6_protosw = {
 static int tcpv6_net_init(struct net *net)
 {
  int err;
- struct socket *sock;
- struct sock *sk;

- err = inet_csk_ctl_sock_create(&sock, PF_INET6, SOCK_RAW, IPPROTO_TCP);
+ err = inet_csk_ctl_sock_create(&net->ipv6.tcp_sk,
+           PF_INET6, SOCK_RAW, IPPROTO_TCP);
  if (err)
   return err;
-
- net->ipv6.tcp_sk = sk = sock->sk;
- sk_change_net(sk, net);
- return err;
+ sk_change_net(net->ipv6.tcp_sk, net);
+ return 0;
 }

 static void tcpv6_net_exit(struct net *net)
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 2715aca..58c1d84 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -674,21 +674,19 @@ static int sctp_ctl_sock_init(void)
 {
  int err;
  sa_family_t family;
- struct socket *socket;

  if (sctp_get_pf_specific(PF_INET6))
   family = PF_INET6;
  else
   family = PF_INET;

- err = inet_csk_ctl_sock_create(&socket,
+ err = inet_csk_ctl_sock_create(&sctp_ctl_sock,
           family, SOCK_SEQPACKET, IPPROTO_SCTP);
  if (err < 0) {
```

```
   printk(KERN_ERR
          "SCTP: Failed to create the SCTP control socket.\n");
   return err;
  }
- sctp_ctl_sock = socket->sk;
  return 0;
 }

--
1.5.3.rc5
```

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

---

Subject: [PATCH 7/11 net-2.6.26] [ICMP]: Simplify ICMP control socket creation.
Posted by den on Fri, 28 Mar 2008 09:08:54 GMT
View Forum Message <> Reply to Message

Replace sock_create_kern with inet_csk_ctl_sock_create.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 net/ipv4/icmp.c |   23 +++++------------------
 1 files changed, 5 insertions(+), 18 deletions(-)

diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index 803bc9f..b00bc20 100644
--- a/net/ipv4/icmp.c
+++ b/net/ipv4/icmp.c
@@ -1152,33 +1152,20 @@ int __net_init icmp_sk_init(struct net *net)

  for_each_possible_cpu(i) {
   struct sock *sk;
-  struct socket *sock;
-  struct inet_sock *inet;

-  err = sock_create_kern(PF_INET, SOCK_RAW, IPPROTO_ICMP, &sock);
+  err = inet_csk_ctl_sock_create(&sk,
+           PF_INET, SOCK_RAW, IPPROTO_ICMP);
   if (err < 0)
    goto fail;

-  net->ipv4.icmp_sk[i] = sk = sock->sk;
+  net->ipv4.icmp_sk[i] = sk;
   sk_change_net(sk, net);
```

```
-  sk->sk_allocation = GFP_ATOMIC;
-
   /* Enough space for 2 64K ICMP packets, including
    * sk_buff struct overhead.
    */
-  sk->sk_sndbuf =
-   (2 * ((64 * 1024) + sizeof(struct sk_buff)));
-
-  inet = inet_sk(sk);
-  inet->uc_ttl = -1;
-  inet->pmtudisc = IP_PMTUDISC_DONT;
-
-  /* Unhash it so that IP input processing does not even
-   * see it, we do not wish this socket to see incoming
-   * packets.
-   */
-  sk->sk_prot->unhash(sk);
+  sk->sk_sndbuf = (2 * ((64 * 1024) + sizeof(struct sk_buff)));
+  inet_sk(sk)->pmtudisc = IP_PMTUDISC_DONT;
  }

  /* Control parameters for ECHO replies. */
--
1.5.3.rc5
```

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

Subject: [PATCH 8/11 net-2.6.26] [INET]: uc_ttl assignment in inet_csk_ctl_sock_create is extra.

Posted by den on Fri, 28 Mar 2008 09:08:55 GMT

View Forum Message <> Reply to Message

uc_ttl is initialized in inet(6)_create and never changed except exact
setsockopt ioctl. Remove it.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 net/ipv4/inet_connection_sock.c |    1 -
 1 files changed, 0 insertions(+), 1 deletions(-)

diff --git a/net/ipv4/inet_connection_sock.c b/net/ipv4/inet_connection_sock.c
index 2cb3552..bfa6274 100644
--- a/net/ipv4/inet_connection_sock.c
```

```
+++ b/net/ipv4/inet_connection_sock.c
@@ -660,7 +660,6 @@ int inet_csk_ctl_sock_create(struct sock **sk, unsigned short family,
  if (rc == 0) {
   *sk = socket->sk;
   (*sk)->sk_allocation = GFP_ATOMIC;
-  inet_sk(*sk)->uc_ttl = -1;
   /*
    * Unhash it so that IP input processing does not even see it,
    * we do not wish this socket to see incoming packets.
--
1.5.3.rc5
```

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

---

Subject: [PATCH 9/11 net-2.6.26] [IPV6]: Simplify IPv6 control socket creation.
Posted by den on Fri, 28 Mar 2008 09:08:56 GMT
View Forum Message <> Reply to Message

Replace sock_create_kern with inet_csk_ctl_sock_create.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv6/icmp.c | 10 ++++-------
 net/ipv6/mcast.c | 7 ++-----
 net/ipv6/ndisc.c | 7 ++-----
 3 files changed, 7 insertions(+), 17 deletions(-)

```
diff --git a/net/ipv6/icmp.c b/net/ipv6/icmp.c
index 63309d1..fe8ddab 100644
--- a/net/ipv6/icmp.c
+++ b/net/ipv6/icmp.c
@@ -808,9 +808,8 @@ static int __net_init icmpv6_sk_init(struct net *net)
   return -ENOMEM;

  for_each_possible_cpu(i) {
-  struct socket *sock;
-  err = sock_create_kern(PF_INET6, SOCK_RAW, IPPROTO_ICMPV6,
-          &sock);
+  err = inet_csk_ctl_sock_create(&sk, PF_INET6,
+           SOCK_RAW, IPPROTO_ICMPV6);
   if (err < 0) {
    printk(KERN_ERR
           "Failed to initialize the ICMP6 control socket "
@@ -819,10 +818,9 @@ static int __net_init icmpv6_sk_init(struct net *net)
```

```
    goto fail;
   }

-  net->ipv6.icmp_sk[i] = sk = sock->sk;
+  net->ipv6.icmp_sk[i] = sk;
   sk_change_net(sk, net);

-  sk->sk_allocation = GFP_ATOMIC;
   /*
    * Split off their lock-class, because sk->sk_dst_lock
    * gets used from softirqs, which is safe for
@@ -837,8 +835,6 @@ static int __net_init icmpv6_sk_init(struct net *net)
    */
   sk->sk_sndbuf =
    (2 * ((64 * 1024) + sizeof(struct sk_buff)));
-
-  sk->sk_prot->unhash(sk);
  }
  return 0;

diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index d810cff..4092d2e 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -2673,11 +2673,10 @@ static void igmp6_proc_exit(struct net *net)
 static int igmp6_net_init(struct net *net)
 {
  struct ipv6_pinfo *np;
- struct socket *sock;
  struct sock *sk;
  int err;

- err = sock_create_kern(PF_INET6, SOCK_RAW, IPPROTO_ICMPV6, &sock);
+ err = inet_csk_ctl_sock_create(&sk, PF_INET6, SOCK_RAW, IPPROTO_ICMPV6);
  if (err < 0) {
   printk(KERN_ERR
          "Failed to initialize the IGMP6 control socket (err %d).\n",
@@ -2685,10 +2684,8 @@ static int igmp6_net_init(struct net *net)
   goto out;
  }

- net->ipv6.igmp_sk = sk = sock->sk;
+ net->ipv6.igmp_sk = sk;
  sk_change_net(sk, net);
- sk->sk_allocation = GFP_ATOMIC;
- sk->sk_prot->unhash(sk);

  np = inet6_sk(sk);
```

```
  np->hop_limit = 1;
diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index e77b74e..9d8c552 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -1731,12 +1731,11 @@ static int ndisc_ifinfo_sysctl_strategy(ctl_table *ctl, int __user
*name,

 static int ndisc_net_init(struct net *net)
 {
- struct socket *sock;
  struct ipv6_pinfo *np;
  struct sock *sk;
  int err;

- err = sock_create_kern(PF_INET6, SOCK_RAW, IPPROTO_ICMPV6, &sock);
+ err = inet_csk_ctl_sock_create(&sk, PF_INET6, SOCK_RAW, IPPROTO_ICMPV6);
  if (err < 0) {
   ND_PRINTK0(KERN_ERR
       "ICMPv6 NDISC: Failed to initialize the control socket (err %d).\n",
@@ -1744,15 +1743,13 @@ static int ndisc_net_init(struct net *net)
   return err;
  }

- net->ipv6.ndisc_sk = sk = sock->sk;
+ net->ipv6.ndisc_sk = sk;
  sk_change_net(sk, net);

  np = inet6_sk(sk);
- sk->sk_allocation = GFP_ATOMIC;
  np->hop_limit = 255;
  /* Do not loopback ndisc messages */
  np->mc_loop = 0;
- sk->sk_prot->unhash(sk);

  return 0;
 }
--
1.5.3.rc5
```

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

Subject: [PATCH 10/11] [NETNS]: Create tcp control socket in the each

namespace.
Posted by den on Fri, 28 Mar 2008 09:08:57 GMT
View Forum Message <> Reply to Message

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/net/netns/ipv4.h | 1 +
 net/ipv4/tcp_ipv4.c | 26 ++++++++++++++++++++++++--
 2 files changed, 25 insertions(+), 2 deletions(-)

```
diff --git a/include/net/netns/ipv4.h b/include/net/netns/ipv4.h
index af685f7..34ee348 100644
--- a/include/net/netns/ipv4.h
+++ b/include/net/netns/ipv4.h
@@ -28,6 +28,7 @@ struct netns_ipv4 {
  struct sock  *fibnl;

  struct sock  **icmp_sk;
+ struct sock  *tcp_sock;

  struct netns_frags frags;
 #ifdef CONFIG_NETFILTER
diff --git a/net/ipv4/tcp_ipv4.c b/net/ipv4/tcp_ipv4.c
index 2ee7fb9..14944af 100644
--- a/net/ipv4/tcp_ipv4.c
+++ b/net/ipv4/tcp_ipv4.c
@@ -2491,11 +2491,33 @@ struct proto tcp_prot = {
  REF_PROTO_INUSE(tcp)
 };

+
+static int __net_init tcp_sk_init(struct net *net)
+{
+ int err;
+ err = inet_csk_ctl_sock_create(&net->ipv4.tcp_sock,
+           PF_INET, SOCK_RAW, IPPROTO_TCP);
+ if (err < 0)
+  return err;
+ sk_change_net(net->ipv4.tcp_sock, net);
+ return 0;
+}
+
+static void __net_exit tcp_sk_exit(struct net *net)
+{
+ sk_release_kernel(net->ipv4.tcp_sock);
+}
+
+static struct pernet_operations __net_initdata tcp_sk_ops = {
+       .init = tcp_sk_init,
```

```
+       .exit = tcp_sk_exit,
+};
+
 void __init tcp_v4_init(void)
 {
- if (inet_csk_ctl_sock_create(&tcp_sock, PF_INET, SOCK_RAW,
-         IPPROTO_TCP) < 0)
+ if (register_pernet_device(&tcp_sk_ops))
    panic("Failed to create the TCP control socket.\n");
+ tcp_sock = init_net.ipv4.tcp_sock;
 }

 EXPORT_SYMBOL(ipv4_specific);
--
1.5.3.rc5
```

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

---

Subject: [PATCH 11/11 net-2.6.26] [NETNS]: Use TCP control socket from a correct namespace.
Posted by den on Fri, 28 Mar 2008 09:08:58 GMT
View Forum Message <> Reply to Message

```
Signed-off-by: Denis V.Lunev <den@openvz.org>
---
 net/ipv4/tcp_ipv4.c |   10 ++++------
 1 files changed, 4 insertions(+), 6 deletions(-)

diff --git a/net/ipv4/tcp_ipv4.c b/net/ipv4/tcp_ipv4.c
index 14944af..34f9174 100644
--- a/net/ipv4/tcp_ipv4.c
+++ b/net/ipv4/tcp_ipv4.c
@@ -88,9 +88,6 @@ int sysctl_tcp_low_latency __read_mostly;
 /* Check TCP sequence numbers in ICMP packets. */
 #define ICMP_MIN_LENGTH 8

-/* Socket used for sending RSTs */
-static struct sock *tcp_sock __read_mostly;
-
 void tcp_v4_send_check(struct sock *sk, int len, struct sk_buff *skb);

 #ifdef CONFIG_TCP_MD5SIG
@@ -598,7 +595,8 @@ static void tcp_v4_send_reset(struct sock *sk, struct sk_buff *skb)
         sizeof(struct tcphdr), IPPROTO_TCP, 0);
```

```
  arg.csumoffset = offsetof(struct tcphdr, check) / 2;

- ip_send_reply(tcp_sock, skb, &arg, arg.iov[0].iov_len);
+ ip_send_reply(dev_net(skb->dst->dev)->ipv4.tcp_sock, skb,
+        &arg, arg.iov[0].iov_len);

  TCP_INC_STATS_BH(TCP_MIB_OUTSEGS);
  TCP_INC_STATS_BH(TCP_MIB_OUTRSTS);
@@ -693,7 +691,8 @@ static void tcp_v4_send_ack(struct tcp_timewait_sock *twsk,
  if (twsk)
   arg.bound_dev_if = twsk->tw_sk.tw_bound_dev_if;

- ip_send_reply(tcp_sock, skb, &arg, arg.iov[0].iov_len);
+ ip_send_reply(dev_net(skb->dev)->ipv4.tcp_sock, skb,
+        &arg, arg.iov[0].iov_len);

  TCP_INC_STATS_BH(TCP_MIB_OUTSEGS);
 }
@@ -2517,7 +2516,6 @@ void __init tcp_v4_init(void)
 {
  if (register_pernet_device(&tcp_sk_ops))
   panic("Failed to create the TCP control socket.\n");
- tcp_sock = init_net.ipv4.tcp_sock;
 }

 EXPORT_SYMBOL(ipv4_specific);
--
1.5.3.rc5
```

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

---

Subject: Re: [PATCH 2/11 net-2.6.26] [DCCP]: Replace socket with sock for reset sending.
Posted by Arnaldo Carvalho de M on Fri, 28 Mar 2008 14:39:59 GMT
View Forum Message <> Reply to Message

Em Fri, Mar 28, 2008 at 12:08:49PM +0300, Denis V. Lunev escreveu:
> Replace dccp_v(4|6)_ctl_socket with sock to unify a code with TCP/ICMP.
>
> Signed-off-by: Denis V. Lunev <den@openvz.org>
> ---
> net/dccp/ipv4.c | 16 +++++++++-------
> net/dccp/ipv6.c | 10 ++++++----
> 2 files changed, 15 insertions(+), 11 deletions(-)

>
> diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
> index 4ca8b0c..79a9a4a 100644
> --- a/net/dccp/ipv4.c
> +++ b/net/dccp/ipv4.c
> @@ -36,7 +36,7 @@
> * the Out-of-the-blue (OOTB) packets. A control sock will be created
> * for this socket at the initialization time.
> */
> -static struct socket *dccp_v4_ctl_socket;
> +static struct sock *dccp_v4_ctl_sk;
>
> int dccp_v4_connect(struct sock *sk, struct sockaddr *uaddr, int addr_len)
> {
> @@ -514,11 +514,11 @@ static void dccp_v4_ctl_send_reset(struct sock *sk, struct sk_buff
*rxskb)
> if (rxskb->rtable->rt_type != RTN_LOCAL)
> return;
>
> - dst = dccp_v4_route_skb(dccp_v4_ctl_socket->sk, rxskb);
> + dst = dccp_v4_route_skb(dccp_v4_ctl_sk, rxskb);
> if (dst == NULL)
> return;
>
> - skb = dccp_ctl_make_reset(dccp_v4_ctl_socket, rxskb);
> + skb = dccp_ctl_make_reset(dccp_v4_ctl_sk->sk_socket, rxskb);

Why not make dccp_ctl_make_reset receive a struct sock too? It only uses
ctl->sk anyway. But I guess that can be on a follow-up patch, could you
that?

Thanks,

- Arnaldo

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

---

Subject: Re: [PATCH 3/11 net-2.6.26] [DCCP]: dccp_v(4|6)_ctl_socket is leaked.
Posted by Arnaldo Carvalho de M on Fri, 28 Mar 2008 14:44:46 GMT
View Forum Message <> Reply to Message

Em Fri, Mar 28, 2008 at 12:08:50PM +0300, Denis V. Lunev escreveu:
> This seems a purism as module can't be unloaded, but though if cleanup method
> is present it should be correct and clean all staff created.

Its the right thing to do :-) But perhaps we could have a
inet_csk_ctl_sock_destroy() as we have inet_csk_ctl_sock_create?

And since this is not csk at all, could you please rename it to
inet_ctl_sock_{create,destroy}()? I Agree with Vlad and Pavel that this
would make it more clear, as SCTP is not (yet) a inet_connection_sock.

- Arnaldo

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

---

Subject: Re: [PATCH 7/11 net-2.6.26] [ICMP]: Simplify ICMP control socket creation.
Posted by Arnaldo Carvalho de M on Fri, 28 Mar 2008 14:47:26 GMT
View Forum Message <> Reply to Message

Em Fri, Mar 28, 2008 at 12:08:54PM +0300, Denis V. Lunev escreveu:
> Replace sock_create_kern with inet_csk_ctl_sock_create.
>
> Signed-off-by: Denis V. Lunev <den@openvz.org>
> ---
> net/ipv4/icmp.c | 23 +++++----------------
> 1 files changed, 5 insertions(+), 18 deletions(-)
>
> diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
> index 803bc9f..b00bc20 100644
> --- a/net/ipv4/icmp.c
> +++ b/net/ipv4/icmp.c
> @@ -1152,33 +1152,20 @@ int __net_init icmp_sk_init(struct net *net)
>
>   for_each_possible_cpu(i) {
>   struct sock *sk;
> - struct socket *sock;
> - struct inet_sock *inet;
>
> - err = sock_create_kern(PF_INET, SOCK_RAW, IPPROTO_ICMP, &sock);
> + err = inet_csk_ctl_sock_create(&sk,
> +   PF_INET, SOCK_RAW, IPPROTO_ICMP);

And ICMP surely is not a inet_connection_sock, so please rename it as
suggested in another message.

- Arnaldo

_____
Containers mailing list

Subject: Re: [PATCH 2/11 net-2.6.26] [DCCP]: Replace socket with sock for reset sending.
Posted by den on Fri, 28 Mar 2008 15:08:42 GMT
View Forum Message <> Reply to Message

On Fri, 2008-03-28 at 11:39 -0300, Arnaldo Carvalho de Melo wrote:
> Em Fri, Mar 28, 2008 at 12:08:49PM +0300, Denis V. Lunev escreveu:
> > Replace dccp_v(4|6)_ctl_socket with sock to unify a code with TCP/ICMP.
> >
> > Signed-off-by: Denis V. Lunev <den@openvz.org>
> > ---
> > net/dccp/ipv4.c | 16 +++++++++-------
> > net/dccp/ipv6.c | 10 ++++++----
> > 2 files changed, 15 insertions(+), 11 deletions(-)
> >
> > diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
> > index 4ca8b0c..79a9a4a 100644
> > --- a/net/dccp/ipv4.c
> > +++ b/net/dccp/ipv4.c
> > @@ -36,7 +36,7 @@
> >   * the Out-of-the-blue (OOTB) packets. A control sock will be created
> >   * for this socket at the initialization time.
> >   */
> > -static struct socket *dccp_v4_ctl_socket;
> > +static struct sock *dccp_v4_ctl_sk;
> >
> > int dccp_v4_connect(struct sock *sk, struct sockaddr *uaddr, int addr_len)
> > {
> > @@ -514,11 +514,11 @@ static void dccp_v4_ctl_send_reset(struct sock *sk, struct sk_buff
*rxskb)
> >   if (rxskb->rtable->rt_type != RTN_LOCAL)
> >   return;
> >
> > - dst = dccp_v4_route_skb(dccp_v4_ctl_socket->sk, rxskb);
> > + dst = dccp_v4_route_skb(dccp_v4_ctl_sk, rxskb);
> >   if (dst == NULL)
> >   return;
> >
> > - skb = dccp_ctl_make_reset(dccp_v4_ctl_socket, rxskb);
> > + skb = dccp_ctl_make_reset(dccp_v4_ctl_sk->sk_socket, rxskb);
>
> Why not make dccp_ctl_make_reset receive a struct sock too? It only uses
> ctl->sk anyway. But I guess that can be on a follow-up patch, could you
> that?

sure!

Thank you for pointing this out :)

Regards,
Den

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

---

Subject: Re: [PATCH 3/11 net-2.6.26] [DCCP]: dccp_v(4|6)_ctl_socket is leaked.
Posted by den on Fri, 28 Mar 2008 15:22:58 GMT
View Forum Message <> Reply to Message

On Fri, 2008-03-28 at 11:44 -0300, Arnaldo Carvalho de Melo wrote:
> Em Fri, Mar 28, 2008 at 12:08:50PM +0300, Denis V. Lunev escreveu:
> > This seems a purism as module can't be unloaded, but though if cleanup method
> > is present it should be correct and clean all staff created.
>
> Its the right thing to do :-) But perhaps we could have a
> inet_csk_ctl_sock_destroy() as we have inet_csk_ctl_sock_create?
>
> And since this is not csk at all, could you please rename it to
> inet_ctl_sock_{create,destroy}()? I Agree with Vlad and Pavel that this
> would make it more clear, as SCTP is not (yet) a inet_connection_sock.

Basically, after these sockets will be moved into namespace we'll use
sk_release_kernel for them.

Can I send a rename as a follow up on Monday :) ?

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

---

Subject: Re: [PATCH 3/11 net-2.6.26] [DCCP]: dccp_v(4|6)_ctl_socket is leaked.
Posted by Arnaldo Carvalho de M on Fri, 28 Mar 2008 16:47:52 GMT
View Forum Message <> Reply to Message

Em Fri, Mar 28, 2008 at 06:22:58PM +0300, Denis V. Lunev escreveu:
> On Fri, 2008-03-28 at 11:44 -0300, Arnaldo Carvalho de Melo wrote:

> > Em Fri, Mar 28, 2008 at 12:08:50PM +0300, Denis V. Lunev escreveu:
> > > This seems a purism as module can't be unloaded, but though if cleanup method
> > > is present it should be correct and clean all staff created.
> >
> > Its the right thing to do :-) But perhaps we could have a
> > inet_csk_ctl_sock_destroy() as we have inet_csk_ctl_sock_create?
> >
> > And since this is not csk at all, could you please rename it to
> > inet_ctl_sock_{create,destroy}()? I Agree with Vlad and Pavel that this
> > would make it more clear, as SCTP is not (yet) a inet_connection_sock.
>
> Basically, after these sockets will be moved into namespace we'll use
> sk_release_kernel for them.
>
> Can I send a rename as a follow up on Monday :) ?

Logically the rename should be done first, then its use, but its up to
Dave/you.

- Arnaldo

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

---

Subject: Re: [PATCH 3/11 net-2.6.26] [DCCP]: dccp_v(4|6)_ctl_socket is leaked.
Posted by den on Fri, 28 Mar 2008 17:09:51 GMT
View Forum Message <> Reply to Message

On Fri, 2008-03-28 at 13:47 -0300, Arnaldo Carvalho de Melo wrote:
> Em Fri, Mar 28, 2008 at 06:22:58PM +0300, Denis V. Lunev escreveu:
> > On Fri, 2008-03-28 at 11:44 -0300, Arnaldo Carvalho de Melo wrote:
> > > Em Fri, Mar 28, 2008 at 12:08:50PM +0300, Denis V. Lunev escreveu:
> > > > This seems a purism as module can't be unloaded, but though if cleanup method
> > > > is present it should be correct and clean all staff created.
> > >
> > > Its the right thing to do :-) But perhaps we could have a
> > > inet_csk_ctl_sock_destroy() as we have inet_csk_ctl_sock_create?
> > >
> > > And since this is not csk at all, could you please rename it to
> > > inet_ctl_sock_{create,destroy}()? I Agree with Vlad and Pavel that this
> > > would make it more clear, as SCTP is not (yet) a inet_connection_sock.
> >
> > Basically, after these sockets will be moved into namespace we'll use
> > sk_release_kernel for them.
> >
> > Can I send a rename as a follow up on Monday :) ?

>
> Logically the rename should be done first, then its use, but its up to
> Dave/you.

seems logical. OK, I'll redo this.

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

---

Subject: Re: [PATCH 3/11 net-2.6.26] [DCCP]: dccp_v(4|6)_ctl_socket is leaked.
Posted by davem on Fri, 28 Mar 2008 23:34:27 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Fri, 28 Mar 2008 20:09:51 +0300

> seems logical. OK, I'll redo this.

I'm dropping this patch set for now, please make a full
resubmission when you've done the fixups.

Thanks!

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

---