

---

Subject: [PATCH 7/7] Consolidate frag queues freeing  
Posted by [Pavel Emelianov](#) on Tue, 16 Oct 2007 14:07:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Since we now allocate the queues in inet\_fragment.c, we can safely free it in the same place. The ->destructor callback thus becomes optional for inet\_frags.

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

---

```
diff --git a/net/ipv4/inet_fragment.c b/net/ipv4/inet_fragment.c
index 470b056..a75f8cb 100644
--- a/net/ipv4/inet_fragment.c
+++ b/net/ipv4/inet_fragment.c
@@ -136,7 +136,9 @@ void inet_frag_destroy(struct inet_frag_queue *q, struct inet_frags *f,
    *work -= f->qsize;
    atomic_sub(f->qsize, &f->mem);

- f->destructor(q);
+ if (f->destructor)
+   f->destructor(q);
+ kfree(q);

}
```

```
EXPORT_SYMBOL(inet_frag_destroy);
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 62dcb79..34e790d 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -171,7 +171,6 @@ static __inline__ void ip4_frag_free(struct inet_frag_queue *q)
    qp = container_of(q, struct ipq, q);
    if (qp->peer)
        inet_putpeer(qp->peer);
- kfree(qp);
}
```

```
diff --git a/net/ipv6/netfilter/nf_conntrack_reasm.c b/net/ipv6/netfilter/nf_conntrack_reasm.c
index 411daf5..a70a482 100644
--- a/net/ipv6/netfilter/nf_conntrack_reasm.c
+++ b/net/ipv6/netfilter/nf_conntrack_reasm.c
@@ -130,11 +130,6 @@ static inline void frag_kfree_skb(struct sk_buff *skb, unsigned int *work)
    kfree_skb(skb);
}
```

```
-static void nf_frag_free(struct inet_frag_queue *q)
```

```

- {
- kfree(container_of(q, struct nf_ct_frag6_queue, q));
- }
-
/* Destruction primitives. */

static __inline__ void fq_put(struct nf_ct_frag6_queue *fq)
@@ -682,7 +677,7 @@ int nf_ct_frag6_init(void)
    nf_frags.ctl = &nf_frags_ctl;
    nf_frags.hashfn = nf_hashfn;
    nf_frags.constructor = ip6_frag_init;
- nf_frags.destructor = nf_frag_free;
+ nf_frags.destructor = NULL;
    nf_frags.skbn_free = nf_skbn_free;
    nf_frags.qsize = sizeof(struct nf_ct_frag6_queue);
    nf_frags.match = ip6_frag_match;
diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 01766bc..76c88a9 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -175,11 +175,6 @@ void ip6_frag_init(struct inet_frag_queue *q, void *a)
}
EXPORT_SYMBOL(ip6_frag_init);

-static void ip6_frag_free(struct inet_frag_queue *fq)
- {
- kfree(container_of(fq, struct frag_queue, q));
- }
-
/* Destruction primitives. */

static __inline__ void fq_put(struct frag_queue *fq)
@@ -645,7 +640,7 @@ void __init ipv6_frag_init(void)
    ip6_frags.ctl = &ip6_frags_ctl;
    ip6_frags.hashfn = ip6_hashfn;
    ip6_frags.constructor = ip6_frag_init;
- ip6_frags.destructor = ip6_frag_free;
+ ip6_frags.destructor = NULL;
    ip6_frags.skbn_free = NULL;
    ip6_frags.qsize = sizeof(struct frag_queue);
    ip6_frags.match = ip6_frag_match;
--
1.5.3.4

```