Subject: [PATCH 4/9] Consolidate the xxx_frag_kill
Posted by Pavel Emelianov on Fri, 12 Oct 2007 13:12:43 GMT
View Forum Message <> Reply to Message

Since now all the xxx_frag_kill functions now work
with the generic inet_frag_queue data type, this can
be moved into a common place.

The xxx_unlink() code is moved as well.

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

```
diff --git a/include/net/inet_frag.h b/include/net/inet_frag.h
index ada03ba..9902363 100644
--- a/include/net/inet_frag.h
+++ b/include/net/inet_frag.h
@@ -41,4 +41,6 @@ struct inet_frags {
 void inet_frags_init(struct inet_frags *);
 void inet_frags_fini(struct inet_frags *);

+void inet_frag_kill(struct inet_frag_queue *q, struct inet_frags *f);
+
 #endif
diff --git a/net/ipv4/inet_fragment.c b/net/ipv4/inet_fragment.c
index 69623ff..534eaa8 100644
--- a/net/ipv4/inet_fragment.c
+++ b/net/ipv4/inet_fragment.c
@@ -42,3 +42,26 @@ void inet_frags_fini(struct inet_frags *f)
 {
 }
 EXPORT_SYMBOL(inet_frags_fini);
+
+static inline void fq_unlink(struct inet_frag_queue *fq, struct inet_frags *f)
+{
+ write_lock(&f->lock);
+ hlist_del(&fq->list);
+ list_del(&fq->lru_list);
+ f->nqueues--;
+ write_unlock(&f->lock);
+}
+
+void inet_frag_kill(struct inet_frag_queue *fq, struct inet_frags *f)
+{
+ if (del_timer(&fq->timer))
+  atomic_dec(&fq->refcnt);
+
```

```
+ if (!(fq->last_in & COMPLETE)) {
+  fq_unlink(fq, f);
+  atomic_dec(&fq->refcnt);
+  fq->last_in |= COMPLETE;
+ }
+}
+
+EXPORT_SYMBOL(inet_frag_kill);
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 61035a8..5b376c4 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -105,20 +105,6 @@ int ip_frag_mem(void)
  return atomic_read(&ip4_frags.mem);
 }

-static __inline__ void __ipq_unlink(struct ipq *qp)
-{
- hlist_del(&qp->q.list);
- list_del(&qp->q.lru_list);
- ip4_frags.nqueues--;
-}
-
-static __inline__ void ipq_unlink(struct ipq *ipq)
-{
- write_lock(&ip4_frags.lock);
- __ipq_unlink(ipq);
- write_unlock(&ip4_frags.lock);
-}
-
 static unsigned int ipqhashfn(__be16 id, __be32 saddr, __be32 daddr, u8 prot)
 {
  return jhash_3words((__force u32)id << 16 | prot,
@@ -219,14 +205,7 @@ static __inline__ void ipq_put(struct ipq *ipq, int *work)
  */
 static void ipq_kill(struct ipq *ipq)
 {
- if (del_timer(&ipq->q.timer))
-  atomic_dec(&ipq->q.refcnt);
-
- if (!(ipq->q.last_in & COMPLETE)) {
-  ipq_unlink(ipq);
-  atomic_dec(&ipq->q.refcnt);
-  ipq->q.last_in |= COMPLETE;
- }
+ inet_frag_kill(&ipq->q, &ip4_frags);
 }
```

```
 /* Memory limiting on fragments.  Evictor trashes the oldest
diff --git a/net/ipv6/netfilter/nf_conntrack_reasm.c b/net/ipv6/netfilter/nf_conntrack_reasm.c
index 966a888..2ebe515 100644
--- a/net/ipv6/netfilter/nf_conntrack_reasm.c
+++ b/net/ipv6/netfilter/nf_conntrack_reasm.c
@@ -79,20 +79,6 @@ struct inet_frags_ctl nf_frags_ctl __read_mostly = {

 static struct inet_frags nf_frags;

-static __inline__ void __fq_unlink(struct nf_ct_frag6_queue *fq)
-{
- hlist_del(&fq->q.list);
- list_del(&fq->q.lru_list);
- nf_frags.nqueues--;
-}
-
-static __inline__ void fq_unlink(struct nf_ct_frag6_queue *fq)
-{
- write_lock(&nf_frags.lock);
- __fq_unlink(fq);
- write_unlock(&nf_frags.lock);
-}
-
 static unsigned int ip6qhashfn(__be32 id, struct in6_addr *saddr,
           struct in6_addr *daddr)
 {
@@ -213,14 +199,7 @@ static __inline__ void fq_put(struct nf_ct_frag6_queue *fq, unsigned int
*work)
  */
 static __inline__ void fq_kill(struct nf_ct_frag6_queue *fq)
 {
- if (del_timer(&fq->q.timer))
-  atomic_dec(&fq->q.refcnt);
-
- if (!(fq->q.last_in & COMPLETE)) {
-  fq_unlink(fq);
-  atomic_dec(&fq->q.refcnt);
-  fq->q.last_in |= COMPLETE;
- }
+ inet_frag_kill(&fq->q, &nf_frags);
 }

 static void nf_ct_frag6_evictor(void)
diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index f0e22be..57e32f4 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -100,20 +100,6 @@ int ip6_frag_mem(void)
```

```
  return atomic_read(&ip6_frags.mem);
 }

-static __inline__ void __fq_unlink(struct frag_queue *fq)
-{
- hlist_del(&fq->q.list);
- list_del(&fq->q.lru_list);
- ip6_frags.nqueues--;
-}
-
-static __inline__ void fq_unlink(struct frag_queue *fq)
-{
- write_lock(&ip6_frags.lock);
- __fq_unlink(fq);
- write_unlock(&ip6_frags.lock);
-}
-
 /*
  * callers should be careful not to use the hash value outside the ipfrag_lock
  * as doing so could race with ipfrag_hash_rnd being recalculated.
@@ -236,14 +222,7 @@ static __inline__ void fq_put(struct frag_queue *fq, int *work)
  */
 static __inline__ void fq_kill(struct frag_queue *fq)
 {
- if (del_timer(&fq->q.timer))
-  atomic_dec(&fq->q.refcnt);
-
- if (!(fq->q.last_in & COMPLETE)) {
-  fq_unlink(fq);
-  atomic_dec(&fq->q.refcnt);
-  fq->q.last_in |= COMPLETE;
- }
+ inet_frag_kill(&fq->q, &ip6_frags);
 }

 static void ip6_evictor(struct inet6_dev *idev)
--
1.5.3.4
```