Subject: [PATCH 6/7] Remove no longer needed ->equal callback
Posted by Pavel Emelianov on Tue, 16 Oct 2007 14:05:31 GMT
View Forum Message <> Reply to Message

Since this callback is used to check for conflicts in
hashtable when inserting a newly created frag queue, we can
do the same by checking for matching the queue with the
argument, used to create one.

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

---

```
diff --git a/include/net/inet_frag.h b/include/net/inet_frag.h
index 6429926..954def4 100644
--- a/include/net/inet_frag.h
+++ b/include/net/inet_frag.h
@@ -43,8 +43,6 @@ struct inet_frags {
       void *arg);
  void   (*destructor)(struct inet_frag_queue *);
  void   (*skb_free)(struct sk_buff *);
- int   (*equal)(struct inet_frag_queue *q1,
-       struct inet_frag_queue *q2);
  int   (*match)(struct inet_frag_queue *q,
       void *arg);
  void   (*frag_expire)(unsigned long data);
diff --git a/include/net/ipv6.h b/include/net/ipv6.h
index 005853a..ae328b6 100644
--- a/include/net/ipv6.h
+++ b/include/net/ipv6.h
@@ -378,7 +378,6 @@ static inline int ipv6_prefix_equal(const struct in6_addr *a1,
 }

 struct inet_frag_queue;
-int ip6_frag_equal(struct inet_frag_queue *q1, struct inet_frag_queue *q2);

 struct ip6_create_arg {
  __be32 id;
diff --git a/net/ipv4/inet_fragment.c b/net/ipv4/inet_fragment.c
index 08901b4..470b056 100644
--- a/net/ipv4/inet_fragment.c
+++ b/net/ipv4/inet_fragment.c
@@ -174,7 +174,7 @@ int inet_frag_evictor(struct inet_frags *f)
 EXPORT_SYMBOL(inet_frag_evictor);

 static struct inet_frag_queue *inet_frag_intern(struct inet_frag_queue *qp_in,
-  struct inet_frags *f, unsigned int hash)
+  struct inet_frags *f, unsigned int hash, void *arg)
```

```
 {
  struct inet_frag_queue *qp;
 #ifdef CONFIG_SMP
@@ -188,7 +188,7 @@ static struct inet_frag_queue *inet_frag_intern(struct inet_frag_queue
*qp_in,
   * promoted read lock to write lock.
   */
  hlist_for_each_entry(qp, n, &f->hash[hash], list) {
-  if (f->equal(qp, qp_in)) {
+  if (f->match(qp, arg)) {
    atomic_inc(&qp->refcnt);
    write_unlock(&f->lock);
    qp_in->last_in |= COMPLETE;
@@ -235,7 +235,7 @@ static struct inet_frag_queue *inet_frag_create(struct inet_frags *f,
  if (q == NULL)
   return NULL;

- return inet_frag_intern(q, f, hash);
+ return inet_frag_intern(q, f, hash, arg);
 }

 struct inet_frag_queue *inet_frag_find(struct inet_frags *f, void *key,
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 46f8de6..62dcb79 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -128,20 +128,6 @@ static unsigned int ip4_hashfn(struct inet_frag_queue *q)
  return ipqhashfn(ipq->id, ipq->saddr, ipq->daddr, ipq->protocol);
 }

-static int ip4_frag_equal(struct inet_frag_queue *q1,
-  struct inet_frag_queue *q2)
-{
- struct ipq *qp1, *qp2;
-
- qp1 = container_of(q1, struct ipq, q);
- qp2 = container_of(q2, struct ipq, q);
- return (qp1->id == qp2->id &&
-   qp1->saddr == qp2->saddr &&
-   qp1->daddr == qp2->daddr &&
-   qp1->protocol == qp2->protocol &&
-   qp1->user == qp2->user);
-}
-
 static int ip4_frag_match(struct inet_frag_queue *q, void *a)
 {
  struct ipq *qp;
@@ -631,7 +617,6 @@ void __init ipfrag_init(void)
```

```
  ip4_frags.destructor = ip4_frag_free;
  ip4_frags.skb_free = NULL;
  ip4_frags.qsize = sizeof(struct ipq);
- ip4_frags.equal = ip4_frag_equal;
  ip4_frags.match = ip4_frag_match;
  ip4_frags.frag_expire = ip_expire;
  inet_frags_init(&ip4_frags);
diff --git a/net/ipv6/netfilter/nf_conntrack_reasm.c b/net/ipv6/netfilter/nf_conntrack_reasm.c
index 1ab52ef..411daf5 100644
--- a/net/ipv6/netfilter/nf_conntrack_reasm.c
+++ b/net/ipv6/netfilter/nf_conntrack_reasm.c
@@ -686,7 +686,6 @@ int nf_ct_frag6_init(void)
  nf_frags.skb_free = nf_skb_free;
  nf_frags.qsize = sizeof(struct nf_ct_frag6_queue);
  nf_frags.match = ip6_frag_match;
- nf_frags.equal = ip6_frag_equal;
  nf_frags.frag_expire = nf_ct_frag6_expire;
  inet_frags_init(&nf_frags);

diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 11fffe7..01766bc 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -143,18 +143,6 @@ static unsigned int ip6_hashfn(struct inet_frag_queue *q)
  return ip6qhashfn(fq->id, &fq->saddr, &fq->daddr);
 }

-int ip6_frag_equal(struct inet_frag_queue *q1, struct inet_frag_queue *q2)
-{
- struct frag_queue *fq1, *fq2;
-
- fq1 = container_of(q1, struct frag_queue, q);
- fq2 = container_of(q2, struct frag_queue, q);
- return (fq1->id == fq2->id &&
-   ipv6_addr_equal(&fq2->saddr, &fq1->saddr) &&
-   ipv6_addr_equal(&fq2->daddr, &fq1->daddr));
-}
-EXPORT_SYMBOL(ip6_frag_equal);
-
 int ip6_frag_match(struct inet_frag_queue *q, void *a)
 {
  struct frag_queue *fq;
@@ -661,7 +649,6 @@ void __init ipv6_frag_init(void)
  ip6_frags.skb_free = NULL;
  ip6_frags.qsize = sizeof(struct frag_queue);
  ip6_frags.match = ip6_frag_match;
- ip6_frags.equal = ip6_frag_equal;
  ip6_frags.frag_expire = ip6_frag_expire;
```

```
 inet_frags_init(&ip6_frags);
}
--
1.5.3.4
```