

---

Subject: [PATCH 4/6 net-2.6.25][RAW] Consolidate proto->hash callback  
Posted by [Pavel Emelianov](#) on Fri, 16 Nov 2007 14:14:20 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Having the raw\_hashinfo it's easy to consolidate the  
raw[46]\_hash functions.

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

---

```
diff --git a/include/net/raw.h b/include/net/raw.h
index 70b27c7..43f1e3c 100644
```

```
--- a/include/net/raw.h
```

```
+++ b/include/net/raw.h
```

```
@@ -39,4 +39,6 @@ extern int raw_proc_init(void);
extern void raw_proc_exit(void);
#endif
```

```
+void raw_hash_sk(struct sock *sk, struct raw_hashinfo *h);
```

```
+
```

```
#endif /* _RAW_H */
```

```
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
```

```
index 536aae1..3ed4350 100644
```

```
--- a/net/ipv4/raw.c
```

```
+++ b/net/ipv4/raw.c
```

```
@@ -84,15 +84,22 @@ static struct raw_hashinfo raw_v4_hashinfo = {
    .lock = __RW_LOCK_UNLOCKED(),
};
```

```
-static void raw_v4_hash(struct sock *sk)
```

```
+void raw_hash_sk(struct sock *sk, struct raw_hashinfo *h)
```

```
{
```

```
- struct hlist_head *head = &raw_v4_hashinfo.ht[inet_sk(sk)->num &
-    (RAW_HTABLE_SIZE - 1)];
```

```
+ struct hlist_head *head;
```

```
- write_lock_bh(&raw_v4_hashinfo.lock);
```

```
+ head = &h->ht[inet_sk(sk)->num & (RAW_HTABLE_SIZE - 1)];
```

```
+
```

```
+ write_lock_bh(&h->lock);
```

```
    sk_add_node(sk, head);
```

```
    sock_prot_inc_use(sk->sk_prot);
```

```
- write_unlock_bh(&raw_v4_hashinfo.lock);
```

```
+ write_unlock_bh(&h->lock);
```

```
+
```

```
+EXPORT_SYMBOL_GPL(raw_hash_sk);
```

```
+
```

```

+static void raw_v4_hash(struct sock *sk)
+{
+ raw_hash_sk(sk, &raw_v4_hashinfo);
+}

static void raw_v4_unhash(struct sock *sk)
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index afcbc47..62b8bc7 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -67,13 +67,7 @@ static struct raw_hashinfo raw_v6_hashinfo = {

static void raw_v6_hash(struct sock *sk)
{
- struct hlist_head *list = &raw_v6_hashinfo.ht[inet_sk(sk)->num &
- (RAW_HTABLE_SIZE - 1)];
-
- write_lock_bh(&raw_v6_hashinfo.lock);
- sk_add_node(sk, list);
- sock_prot_inc_use(sk->sk_prot);
- write_unlock_bh(&raw_v6_hashinfo.lock);
+ raw_hash_sk(sk, &raw_v6_hashinfo);
+}

static void raw_v6_unhash(struct sock *sk)

```

---