

Linux Cross Reference

[Free Electrons](#)

Embedded Linux Experts

• [source navigation](#) • [diff markup](#) • [identifier search](#) • [freetext search](#) •

Version: [2.0.40](#) [2.2.26](#) [2.4.37](#) [3.1](#) [3.2](#) [3.3](#) [3.4](#) [3.5](#) [3.6](#) [3.7](#) [3.8](#) [3.9](#) [3.10](#) [3.11](#) [3.12](#) [3.13](#) [3.14](#) [3.15](#) [3.16](#) [3.17](#)

[Linux/net/ipv4/tcp_diag.c](#)

```

1  /*
2  *  tcp_diag.c  Module for monitoring TCP transport protocols sockets.
3  *
4  *  Authors:    Alexey Kuznetsov, <kuznet@ms2.inr.ac.ru>
5  *
6  *      This program is free software; you can redistribute it and/or
7  *      modify it under the terms of the GNU General Public License
8  *      as published by the Free Software Foundation; either version
9  *      2 of the License, or (at your option) any later version.
10 */
11
12
13 #include <linux/module.h>
14 #include <linux/inet_diag.h>
15
16 #include <linux/tcp.h>
17
18 #include <net/tcp.h>
19
20 static void tcp\_diag\_get\_info(struct sock *sk, struct inet\_diag\_msg *r,
21                               void *_info)
22 {
23     const struct tcp\_sock *tp = tcp\_sk(sk);
24     struct tcp\_info *info = _info;
25
26     if (sk->sk\_state == TCP_LISTEN) {
27         r->idiag_rqueue = sk->sk_ack_backlog;
28         r->idiag_wqueue = sk->sk_max_ack_backlog;
29     } else {
30         r->idiag_rqueue = max\_t(int, tp->rcv_nxt - tp->copied_seq, 0);
31         r->idiag_wqueue = tp->write_seq - tp->snd_una;
32     }
33     if (info != NULL)
34         tcp\_get\_info(sk, info);
35 }
36
37 static void tcp\_diag\_dump(struct sk\_buff *skb, struct netlink\_callback *cb,
38                           struct inet\_diag\_req\_v2 *r, struct nlattr *bc)
39 {
40     inet\_diag\_dump\_icsk(&tcp\_hashinfo, skb, cb, r, bc);
41 }
42
43 static int tcp\_diag\_dump\_one(struct sk\_buff *in_skb, const struct nlmsg\_hdr *nlh,
44                              struct inet\_diag\_req\_v2 *req)
45 {
46     return inet\_diag\_dump\_one\_icsk(&tcp\_hashinfo, in_skb, nlh, req);
47 }
48
49 static const struct inet\_diag\_handler tcp\_diag\_handler = {

```

```
50     .dump                = tcp_diag_dump,
51     .dump_one            = tcp_diag_dump_one,
52     .idiag_get_info      = tcp_diag_get_info,
53     .idiag_type          = IPPROTO_TCP,
54 };
55
56 static int __init tcp_diag_init(void)
57 {
58     return inet_diag_register(&tcp_diag_handler);
59 }
60
61 static void __exit tcp_diag_exit(void)
62 {
63     inet_diag_unregister(&tcp_diag_handler);
64 }
65
66 module_init(tcp_diag_init);
67 module_exit(tcp_diag_exit);
68 MODULE_LICENSE("GPL");
69 MODULE_ALIAS_NET_PF_PROTO_TYPE(PF_NETLINK, NETLINK_SOCK_DIAG, 2-6 /* AF_INET - IPPROTO_TCP */);
70
```

This page was automatically generated by [LXR](#) 0.3.1 ([source](#)). • Linux is a registered trademark of Linus Torvalds • [Contact us](#)

- [Home](#)
- [Development](#)
- [Services](#)
- [Training](#)
- [Docs](#)
- [Community](#)
- [Company](#)
- [Blog](#)