

NAME

dtrace_udplite - a DTrace provider for tracing events related to the UDP-Lite protocol

SYNOPSIS

```
udplite:::receive(pktinfo_t *, csinfo_t *, ipinfo_t *, udplitesinfo_t *, udpliteinfo_t *);
```

```
udplite:::send(pktinfo_t *, csinfo_t *, ipinfo_t *, udplitesinfo_t *, udpliteinfo_t *);
```

DESCRIPTION

The DTrace **udplite** provider allows users to trace events in the `udplite(4)` protocol implementation. The **udplite:::send()** probe fires whenever the kernel prepares to transmit a UDP-Lite packet, and the **udplite:::receive()** probe fires whenever the kernel receives a UDP-Lite packet, unless the UDP-Lite header is incomplete, the destination port is 0, the length field is invalid, or the checksum is wrong. The arguments to these probes can be used to obtain detailed information about the IP and UDP-Lite headers of the corresponding packet.

ARGUMENTS

The *pktinfo_t* argument is currently unimplemented and is included for compatibility with other implementations of this provider. Its fields are:

uintptr_t *pkt_addr* Always set to 0.

The *csinfo_t* argument is currently unimplemented and is included for compatibility with other implementations of this provider. Its fields are:

uintptr_t *cs_addr* Always set to 0.

uint64_t *cs_cid* A pointer to the *struct inpcb* for this packet, or NULL.

pid_t *cs_pid* Always set to 0.

The *ipinfo_t* argument contains IP fields common to both IPv4 and IPv6 packets. Its fields are:

uint8_t *ip_ver* IP version of the packet, 4 for IPv4 packets and 6 for IPv6 packets.

uint32_t *ip_plength* IP payload size. This does not include the size of the IP header or IPv6 option headers.

string *ip_saddr* IP source address.

string ip_daddr IP destination address.

The *udpliteinfo_t* argument contains the state of the UDP-Lite connection associated with the packet. Its fields are:

uintptr_t udplites_addr Pointer to the *struct inpcb* containing the IP state for the associated socket.

uint16_t udplites_lport
Local UDP-Lite port.

uint16_t udplites_rport
Remote UDP-Lite port.

string udplites_laddr Local IPv4 or IPv6 address.

string udplites_raddr Remote IPv4 or IPv6 address.

The *udpliteinfo_t* argument is the raw UDP-Lite header of the packet, with all fields in host order. Its fields are:

uint16_t udplite_sport Source UDP-Lite port.

uint16_t udplite_dport Destination UDP-Lite port.

uint16_t udplite_coverage Checksum coverage of the UDP-Lite header, in bytes, or 0 for full coverage.

uint16_t udplite_checksum A checksum of the UDP-Lite header and payload, or 0 if no checksum was calculated.

*struct udplitehdr *udplite_hdr* A pointer to the raw UDP-Lite header.

FILES

/usr/lib/dtrace/udplite.d DTrace type and translator definitions for the **udplite** provider.

EXAMPLES

The following script counts transmitted packets by destination port.

```
udplite:::send
{
```

```

    @num[args[4]->udplite_dport] = count();
}

```

This script will print some details of each UDP-Lite packet as it is sent or received by the kernel:

```

#pragma D option quiet
#pragma D option switchrate=10Hz

dtrace:::BEGIN
{
    printf(" %10s %36s  %-36s %6s\n", "DELTA(us)", "SOURCE",
        "DEST", "COV");
    last = timestamp;
}

udplite:::send
{
    this->elapsed = (timestamp - last) / 1000;
    self->dest = strjoin(strjoin(args[2]->ip_daddr, ":"),
        lltostr(args[4]->udplite_dport));
    printf(" %10d %30s:%-5d -> %-36s %6d\n", this->elapsed,
        args[2]->ip_saddr, args[4]->udplite_sport,
        self->dest, args[4]->udplite_coverage);
    last = timestamp;
}

udplite:::receive
{
    this->elapsed = (timestamp - last) / 1000;
    self->dest = strjoin(strjoin(args[2]->ip_saddr, ":"),
        lltostr(args[4]->udplite_sport));
    printf(" %10d %30s:%-5d <- %-36s %6d\n", this->elapsed,
        args[2]->ip_daddr, args[4]->udplite_dport,
        self->dest, args[4]->udplite_coverage);
    last = timestamp;
}

```

SEE ALSO

dtrace(1), dtrace_ip(4), dtrace_sctp(4), dtrace_tcp(4), dtrace_udp(4), udplite(4), SDT(9)

HISTORY

The **udplite** provider first appeared in FreeBSD 12.0.

AUTHORS

This manual page was written by Mark Johnston <*markj@FreeBSD.org*> and Michael Tuexen <*tuexen@FreeBSD.org*>.