NAME

dtrace_udp **-** a DTrace provider for tracing events related to the UDP protocol

SYNOPSIS

udp:::receive(*pktinfo_t* *, *csinfo_t* *, *ipinfo_t* *, *udpsinfo_t* *, *udpinfo_t* *);

udp:::send(*pktinfo_t* *, *csinfo_t* *, *ipinfo_t* *, *udpsinfo_t* *, *udpinfo_t* *);

DESCRIPTION

The DTrace **udp** provider allows users to trace events in the udp(4) protocol implementation. The **udp:::send**() probe fires whenever the kernel prepares to transmit a UDP packet, and the **udp:::receive**() probe fires whenever the kernel receives a UDP packet, unless the UDP 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 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 *udpsinfo_t* argument contains the state of the UDP connection associated with the packet. Its fields are:

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

uint16_t udps_lport
Local UDP port.

uint16_t udps_rport
Remote UDP port.

string udps_laddr Local IPv4 or IPv6 address.

string udps_raddr Remote IPv4 or IPv6 address.

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

uint16_t udp_sport Source UDP port.

uint16_t udp_dport Destination UDP port.

uint16_t udp_length Length of the UDP header and payload, in bytes.

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

*struct udphdr *udp_hdr* A pointer to the raw UDP header.

FILES

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

EXAMPLES

The following script counts transmitted packets by destination port.

```
udp:::send
{
        @num[args[4]->udp_dport] = count();
}
```

This script will print some details of each UDP 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", "BYTES");
        last = timestamp;
}

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

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

COMPATIBILITY

This provider is compatible with the **udp** provider in Solaris.

SEE ALSO

dtrace(1), dtrace_ip(4), dtrace_sctp(4), dtrace_tcp(4), dtrace_udplite(4), udp(4), SDT(9)

HISTORY

The **udp** provider first appeared in FreeBSD 10.0.

AUTHORS

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