NAME

dtrace_ip - a DTrace provider for tracing events related to the IPv4 and IPv6 protocols

SYNOPSIS

ip:::receive(*pktinfo_t* *, *csinfo_t* *, *ipinfo_t* *, *ifinfo_t* *, *ipv4info_t* *, *ipv6info_t* *);

ip:::send(*pktinfo_t* *, *csinfo_t* *, *ipinfo_t* *, *ifinfo_t* *, *ipv4info_t* *, *ipv6info_t* *);

DESCRIPTION

The DTrace **ip** provider allows users to trace events in the ip(4) and ip6(4) protocol implementations. The **ip:::send**() probe fires whenever the kernel prepares to transmit an IP packet, and the **ip:::receive**() probe fires whenever the kernel receives an IP packet. The arguments to these probes can be used to obtain detailed information about the IP headers of the corresponding packet, as well as the network interface on which the packet was sent or received. Unlike the dtrace_tcp(4) and dtrace_udp(4) providers, **ip** provider probes are triggered by forwarded packets. That is, the probes will fire on packets that are not destined to the local host.

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 *ifinfo_t* argument describes the outgoing and incoming interfaces for the packet in the **ip:::send**() and **ip:::receive**() probes respectively. Its fields are:

string if_name	The interface name.
int8_t if_local	A boolean value indicating whether or not the interface is a loopback interface.
uintptr_t if_addr	A pointer to the *struct ifnet* which describes the interface. See the ifnet(9) manual page.

The *ipv4info_t* argument contains the fields of the IP header for IPv4 packets. This argument is NULL for IPv6 packets. DTrace scripts should use the **ip_ver**() field in the *ipinfo_t* argument to determine whether to use this argument. Its fields are:

uint8_t ipv4_ver	IP version. This will always be 4 for IPv4 packets.
uint8_t ipv4_ihl	The IP header length, including options, in 32-bit words.
uint8_t ipv4_tos	IP type of service field.
uint16_t ipv4_length	The total packet length, including the header, in bytes.
uint16_t ipv4_ident	Identification field.
uint8_t ipv4_flags	The IP flags.
uint16_t ipv4_offset	The fragment offset of the packet.
uint8_t ipv4_ttl	Time to live field.
uint8_t ipv4_protocol	Next-level protocol ID.
string ipv4_protostr	A string containing the name of the encapsulated protocol. The protocol strings are defined in the *protocol* array in */usr/lib/dtrace/ip.d*
uint16_t ipv4_checksum	The IP checksum.
ipaddr_t ipv4_src	IPv4 source address.

ipaddr_t ipv4_dst	IPv4 destination address.
string ipv4_saddr	A string representation of the source address.
string ipv4_daddr	A string representation of the destination address.
*ipha_t *ipv4_hdr*	A pointer to the raw IPv4 header.

The *ipv6info_t* argument contains the fields of the IP header for IPv6 packets. Its fields are not set for IPv4 packets; as with the *ipv4info_t* argument, the **ip_ver**() field should be used to determine whether this argument is valid. Its fields are:

uint8_t ipv6_ver	IP version. This will always be 6 for IPv6 packets.
uint8_t ipv6_tclass	The traffic class, used to set the differentiated services codepoint and extended congestion notification flags.
uint32_t ipv6_flow	The flow label of the packet.
uint16_t ipv6_plen	The IP payload size, including extension headers, in bytes.
uint8_t ipv6_nexthdr	An identifier for the type of the next header.
string ipv6_nextstr	A string representation of the type of the next header.
uint8_t ipv6_hlim	The hop limit.
*ip6_addr_t *ipv6_src*	IPv6 source address.
*ip6_addr_t *ipv6_dst*	IPv6 destination address.
string ipv6_saddr	A string representation of the source address.
string ipv6_daddr	A string representation of the destination address.
*struct ip6_hdr *ipv6_hdr*	A pointer to the raw IPv6 header.

FILES

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

EXAMPLES

The following script counts received packets by remote host address.

```
ip:::receive
{
        @num[args[2]->ip_saddr] = count();
}
```

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

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

dtrace:::BEGIN
{
        printf(" %10s %30s    %-30s %8s %6s\n", "DELTA(us)", "SOURCE",
            "DEST", "INT", "BYTES");
        last = timestamp;
}

ip:::send
{
        this->elapsed = (timestamp - last) / 1000;
        printf(" %10d %30s -> %-30s %8s %6d\n", this->elapsed,
            args[2]->ip_saddr, args[2]->ip_daddr, args[3]->if_name,
            args[2]->ip_plength);
        last = timestamp;
}

ip:::receive
{
        this->elapsed = (timestamp - last) / 1000;
        printf(" %10d %30s <- %-30s %8s %6d\n", this->elapsed,
            args[2]->ip_daddr, args[2]->ip_saddr, args[3]->if_name,
            args[2]->ip_plength);
        last = timestamp;
}
```

COMPATIBILITY

This provider is compatible with the **ip** providers found in Solaris and Darwin.

SEE ALSO

dtrace(1), dtrace_tcp(4), dtrace_udp(4), ip(4), ip6(4), ifnet(9), SDT(9)

HISTORY

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

AUTHORS

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