Cisco TCP/IP Suite Online Help

 TCPDump Options

 TCPDump Procedures

 TCPDump Concepts

Cisco TCP/IP Suite Online Help

 TCPDump Options

- TCPDump Application Window
- File Menu
- Capture Menu
- Options Menu
- Help Menu

 TCPDump Procedures

 TCPDump Concepts

Cisco TCP/IP Suite Online Help

 TCPDump Options

 TCPDump Procedures

- Capturing and Displaying Packets
- Setting TCPDump Capture Options
- Setting TCPDump Display Options
- Filtering TCPDump Output
- Viewing Previously Saved TCPDump Output
- Changing the TCPDump Display Font
- Sending TCPDump Output to a File
- Saving Existing TCPDump Output in a File
- Printing TCPDump Output
- Starting Net Tools
- Exiting TCPDump

 TCPDump Concepts

Cisco TCP/IP Suite Online Help

 TCPDump Options

 TCPDump Procedures

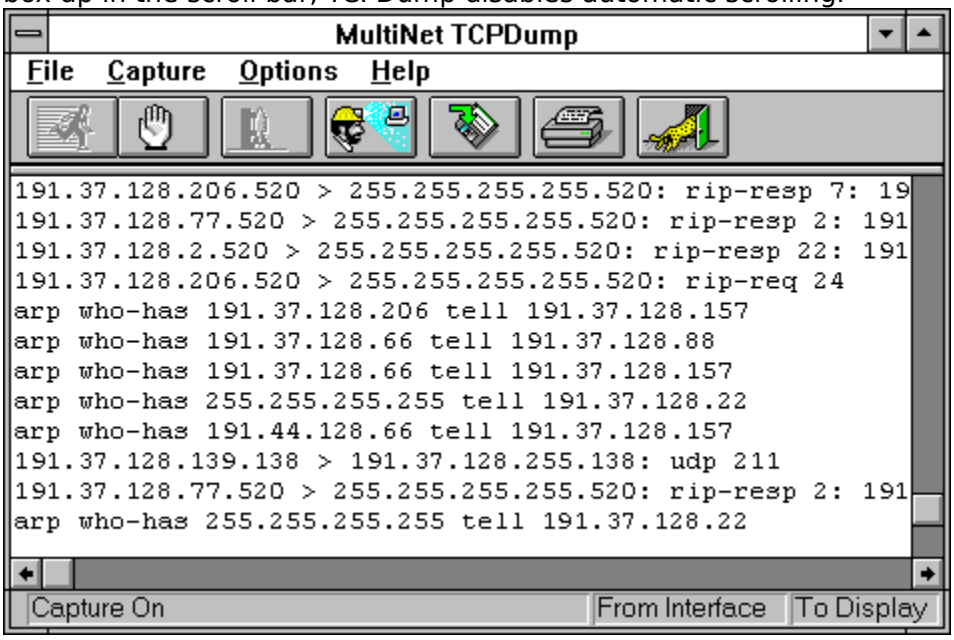
 TCPDump Concepts

- How to Read TCPDump Output
- Filter Expressions
- Filter Expression Examples

TCPDump Application Window

TCPDump lets you view the contents of IP packets received from the network. Packets can be displayed in the output window, or saved to a file for later viewing. TCPDump can capture all packets received from the network, or TCPDump can be configured to capture only specific packets.

When you start a capture, TCPDump data appears in the Output window. You can set the Output window to automatically scroll, letting you always see the most current data. If you move the scroll box (the slider in scroll bar) to the bottom of the scroll bar, data automatically scrolls in the Output window, displaying the most current data at the bottom of the window, and pushing the older data off the top of the window. If you move the scroll box up in the scroll bar, TCPDump disables automatic scrolling.



Click graphic fields for more information.

See Also

- Capturing and Displaying Packets
- How to Read TCPDump Output

File Menu

The File menu lets you save and print TCPDump output, clear the Output window, start Net Tools, and exit TCPDump.

Capture Menu

The Capture menu lets you start and stop capturing packets, delay the start of capturing packets, determine whether TCPDump displays data from the network interface or a previously saved file, and determine whether TCPDump sends output to the display or to a file.

Options Menu

The Options menu lets you specify display options, filter packets TCPDump captures, choose the display font, enable or disable the toolbar, bubble help, and status bar, and split the output window into two panes.

Help Menu

The Help menu lets you view this online help, Windows help, and TCPDump version information.

Start Button

The Start button, or Start from the Capture menu, causes TCPDump to capture and display the contents of packets received. The output appears in the Output window.

Stop Button

The Stop button, or Stop from the Capture menu, causes TCPDump to stop capturing and displaying packets.

Start At Button

The Start At button, or Start At from the Capture menu, displays the Start At dialog box, letting you delay the start of TCPDump and specify how long TCPDump should capture packets.

Start Net Tools Button

The Start Net Tools button, or Start Net Tools from the File menu, launches the Net Tools application.

Save Button

The Save button, or Save As from the File menu, displays the Save As dialog box, letting you save TCPDump output in a file.

Print Button

The Print button, or Print from the File menu, displays the Print dialog box, letting you print TCPDump output.

Exit Button

The Exit button, or Exit from the File menu, closes TCPDump.

Output Window

The Output window displays the packet information captured by TCPDump. The output window can be split into two panes by selecting the Split... option from the Options menu, or by dragging the horizontal boundary located at the top of the window.

You can set the Output window to automatically scroll, letting you always see the most current data. If you move the scroll box (the slider in scroll bar) to the bottom of the scroll bar, data automatically scrolls in the Output window, displaying the most current data at the bottom of the window, and pushing the older data off the top of the window. If you move the scroll box up in the scroll bar, TCPDump disables automatic scrolling.

You can set the display font used in the Output window with the Font dialog box.

Status Bar

The Status bar displays a short description of the selected menu item or toolbar button, as well as whether the network interface or a file is the input device, and whether the display or a file is the output device.

Start At Dialog Box

The Start At dialog box lets you delay the start of TCPDump and determine how long TCPDump should continue capturing packets. Both the starting and stopping of TCPDump can be set by either seconds or number of packets received.

The Start At dialog box contains these fields and buttons:

Count
The number of seconds you want to pass before TCPDump starts capturing packets.

Capture
The number of seconds you want TCPDump to capture packets before stopping.

Packets
The number of packets you want TCPDump to capture.

Start/Stop Alarm
Determines whether TCPDump beeps three seconds before it starts and stops capturing packets.

Start button
Displays the TCPDump Status dialog box, letting you see the time remaining before TCPDump begins capturing packets.

See Also

- Setting TCPDump Capture Options

Filter Dialog Box

The Filter dialog box lets you filter TCPDump output so that only the specified packets are captured. Enter a filter expression, an interface, or both. Only the packets meeting the requirements of the filter expression, and that are received from or sent to the specified interface, are displayed by TCPDump.

The Filter dialog box contains these fields:

Filter Expression

A filter expression. Only packets that meet the requirements of the filter expression are captured. Once a filter expression is entered, it is saved and can be chosen from the drop-down list for future captures.

Interface

An interface to filter packets. Only packets received from and sent to the specified interface are captured. Valid devices are:

- Ethernet
- the loopback connection
- Serial
- FDDI

Maximum Packet

The maximum amount of data that is captured from any one packet.

See Also

- Filtering TCPDump Output
- Filter Expression Examples

Open Dialog Box

The Open dialog box lets you load previously saved TCPDump output. The dialog box contains these fields:

File Name
The name of the TCPDump file you want to open. Choose it from the list of files.

Save File As Type
The file format for the file.

Directories
The directory for the file.

Drives
The drive for the file.

Network
Connects to another network location so you can assign it a new drive letter.

See Also

- Viewing Previously Saved TCPDump Output

Display Options Dialog Box

The Display Options dialog box lets you determine the format in which captured packets are displayed, and which information about the packets is displayed. The dialog box contains these fields:

Symbolic Names
Determines whether TCPDump displays host name information in addition to the IP address for each packet.

Hex Dump
Determines whether packets are displayed in hexadecimal format.

Append Domain Name to Host Name
Determines whether TCPDump displays the domain information with the host name information for each packet.

Ethernet Header Information
Determines whether the Ethernet header (source, destination, protocol, and length) appears on each line of output.

Timestamp Information
Determines whether a timestamp appears on each line of output.

Verbose
Determines whether additional information appears on each line of output.

Interpret RPC Calls
Determines whether TCPDump displays extra information about TCP and UDP packets.

Print Raw TCP Sequence Numbers
Determines whether TCPDump prints the raw TCP sequence numbers.

Translate Remote Address
Determines whether TCPDump translates remote addresses by displaying the remote host names (if available) in place of the remote host IP addresses.

EBCDIC Data
Determines whether TCPDump displays a third column of data, next to the main data and hex columns. The third column of data shows the same data as the other two columns using the IBM EBCDIC encoding standard instead of ASCII.

See Also

- Setting TCPDump Display Options
- How to Read TCPDump Output

Output File Dialog Box

The Output File dialog box lets you choose the file to which TCPDump output is saved. The dialog box contains these fields and buttons:

Filename
The name of the file where TCPDump output will be stored.

Auto Increment
Determines whether TCPDump automatically assigns a new filename for each TCPDump capture. The filenames are automatically incremented by one for each capture.

Sniffer Format
Determines whether TCPDump stores data in the specified file using the Sniffer format.

Browse button
Displays the Open dialog box, letting you choose a file in which to store TCPDump output.

See Also

- Sending TCPDump Output to a File

Save As Dialog Box

The Save As dialog box lets you save the contents of the TCPDump output window in a file.

File Name
The name of the file in which to save the output.

Save File As Type
The file format for the file.

Directories
The directory for the file.

Drives
The drive for storing the file.

Network
Connects to another network location so you can assign it a new drive letter.

See Also

- Saving Existing TCPDump Output in a File

Print Dialog Box

The Print dialog box lets you print the contents of the TCPDump output window.

Printer
Shows the active printer and printer connection. Click the Setup button to change the printer and printer connection.

Print Range group
Lets you specify the pages you want to print: All pages; just the data you selected in the window; or a specific page range.

Print Quality
Determines how clear the printout is, based on the capabilities of your printer.

Copies
The number of copies you want to print.

Collate Copies
Whether you want each copy printed in page number order (checked), or whether you want all copies of each page printed together, so that, for example, all page ones are printed first before printing page two (unchecked).

Setup button
Lets you select a printer and set other printer options using the Print Setup dialog box.

See Also

- Printing TCPDump Output

Print Setup Dialog Box

The Print Setup dialog box lets you select a printer, page orientation and dimensions, paper source and other options:

Printer group
Lets you select either the default printer, or another printer that you have installed.

Orientation group
Determines whether the printout is in portrait or landscape orientation.

Paper group
Determines the size of the paper and the paper tray that is used.

Options button
Additional options that you can set for the selected printer.

Network button
Lets you connect to a printer on your network.

About Dialog Box

The About dialog box displays TCPDump version and copyright information.

TCPDump Status Dialog Box

The TCPDump Status dialog box displays the status of TCPDump.

Click the Abort button to stop the dump and erase any output that has been produced.

Click the Stop button to stop the dump without erasing already captured information.

See Also

- Start At Dialog Box
- Setting TCPDump Capture Options

Font Dialog Box

The Font dialog box lets you change the type font displayed in the Output window and in printed output. A sample indicator displays the font attributes you select. The Font dialog box contains:

Font
The name of the font you want to use. Select an available font from the associated list.

Font Style
The styles available with the selected font. Select a style from the associated list.

Font Size
The type sizes available for the selected font. Select a size from the associated list.

See Also

- Changing the TCPDump Display Font

File Menu

The File menu contains these commands:

Save As
Saves TCPDump output in a file. The shortcut button is:

Clear
Erases all TCPDump output from the window.

Print
Prints TCPDump output. The shortcut button is:

Net Tools
Starts the Net Tools application. The shortcut button is:

Exit
Closes the TCPDump application. The shortcut button is:

Capture Menu

The Capture menu contains these commands:

Start
Starts the capture of IP packets. The shortcut button is:

Stop
Stops the capture of IP packets. The shortcut button is:

Start At
Displays the Start At dialog box, so you can delay the start of and determine the length of capturing packets.

From
Determines whether displayed output is from a new capture of packets from the interface (**Interface**), or whether you display a previously captured dump (**File**).

To
Determines whether captured packets are sent to the TCPDump window (**Display**), or whether they are sent directly to a file (**File**).

Options Menu

The Options menu contains these commands:

Display
Lets you determine which information is displayed by TCPDump, and how that information is displayed and stored, using the Display Options dialog box.

Filter
Lets you filter TCPDump output using the Filter dialog box.

Font
Changes the font for displayed and printed output.

Program

Always On Top
Determines whether the TCPDump window always remains on top of any other active windows.

Bubble Help
Enables or disables bubble help, which is the quick popup help you get when moving the cursor over the toolbar buttons.

Status Bar
Displays or hides the status bar at the bottom of the main window.

Toolbar
Displays or hides the toolbar, which is the bar of shortcut buttons beneath the menu.

Split
Splits the output window into two different windows. The two windows can be sized by dragging the horizontal boundary between them.

Help Menu

The Help menu contains these commands:

Contents
Displays the contents page of this help file.

Search for Help on
Searches this help files index.

How to Use Help
Displays help information for using the Windows help system.

About TCPDump
Displays version information for the application.

Start Capturing Packets

To start capturing packets:

1. Click the Start button, or choose Start from the Capture Menu. Packet information appears in the output window.
2. To stop capturing packets, click the Stop button, or choose Stop from the Capture menu.

See Also

- TCPDump Application Window
- How to Read TCPDump Output

Setting TCPDump Display Options

To set TCPDump display options:

1. Choose Display... from the Options menu. The Display Options dialog box appears.
2. Set the desired display options. For more information about the display options, see Display Options dialog box.
3. Click the OK button.

To start capturing packets, see Capturing Packets, or Setting TCPDump Capture Options.

Setting TCPDump Capture Options

To set TCPDump capture options:

1. Click the Start At button, or choose Start At... from the Capture Menu. The Start At dialog box appears.
2. To set TCPDump to start capturing packets in a number of seconds, enter the number of seconds in the Count Down edit box.
3. To set TCPDump to stop capturing packets after a number of seconds, enter the number of seconds in the Capture edit box.
4. To set TCPDump to stop capturing packets after a number of packets have been captured, enter the number of packets in the Packets edit box.
5. To set TCPDump to beep for three seconds before the start and stop of capturing packets, click the Start / Stop Alarm check box.
6. Click the Start button. The TCPDump Status dialog box appears while TCPDump is idle.
7. If you have specified a condition for TCPDump to stop, it will stop when the condition is fulfilled. Otherwise, click the Stop button to stop capturing packets.

Filtering TCPDump Output

To filter TCPDump output:

1. Choose the Filter... option from the Options Menu. The Filter dialog box appears.
2. Enter a filter expression into the Filter Expression combo box.
3. If you want to capture packets from a specific interface, enter the interface into the Interface combo box.
4. If you want to set a maximum amount of data to be captured from any one packet, enter the amount of data in the Maximum Packet Size edit box.
5. Click the OK button.

To start capturing packets, see Capturing Packets, or Setting TCPDump Capture Options.

See Also

- Filter Expression Examples

Viewing Previously Saved TCPDump Output

To view previously saved TCPDump output:

1. Choose From>File... from the Capture menu.
2. Enter the name of the file you want to open in the File Name edit box.
3. Click the OK button. The output stored in the file appears in the output window.

Sending TCPDump Output to a File

To send TCPDump output to a file:

1. Choose To>File... from the Capture menu. The Output File dialog box appears.
2. Enter the name of the file in which you want to store TCPDump output in the Filename edit box.
3. To cause TCPDump to increment the file name you entered by one every time a new session is started, check the Auto Increment Filename check box.
4. To save the TCPDump output in Sniffer format, check the Sniffer format check box.
5. Click the OK button. The Output File dialog box closes. All output is now stored in the specified file.

Saving Existing TCPDump Output in a File

To save existing TCPDump output in a file:

1. Click the Save button, or choose Save As... from the File menu.
2. Enter the name of the file where you want to save the TCPDump output in the File Name edit box.
3. Click the OK button.

Printing TCPDump Output

To print TCPDump output:

1. Click the Print button, or choose Print... from the File menu.
2. Choose the printer you want to use and click the OK button.

Exiting TCPDump

To exit TCPDump:

- Click the Exit button, or choose Exit from the File menu.

Starting Net Tools

To start Net Tools:

- Click the Start Net Tools button, or choose Net Tools... from the File menu.

Changing the TCPDump Display Font

To change the TCPDump display font:

1. Choose Font... from the Options menu.
2. To change the font, choose the new font from the Font list box.
3. To change the font style, choose the new font style from the Font Style list box.
4. To change the font size, choose the new size from the Size list box.
5. Click the OK button.

Filter Expressions

Filter expressions let you select which packets TCPDump displays. If no expression is given, TCPDump displays all packets originating from or destined for your machine, as well as all broadcast packets. Otherwise, it only displays packets when the expression is "true."

The expression consists of one or more primitives. Primitives usually consist of an id (name or number) preceded by one or more qualifiers. There are three different kinds of qualifier:

- type qualifiers say what kind of thing the id name or number refers to. Possible types are host, net and port. For example, "host foo", "net 128.3", "port 20." If there is no type qualifier, host is assumed.
- dir qualifiers specify a particular transfer direction to or from id. Possible directions are src (source), dst (destination), src or dst, and src and dst. For example, "src foo", "dst net 128.3", "src or dst port ftp-data." If there is no dir qualifier, src or dst is assumed.
- proto qualifiers restrict the match to a particular protocol. Possible protos are: ether, ip, arp, rarp, tcp and udp. For example, "ether src foo", "arp net 128.3", "tcp port 21." If there is no proto qualifier, all protocols consistent with the type are assumed. For example, "src foo" means "(ip or arp or rarp) src foo" (except the latter is not legal syntax), "net bar" means "(ip or arp or rarp) net bar" and "port 53" means "(tcp or udp) port 53."

If an identifier is given without a keyword, the most recent keyword is assumed. For example, host vs and ace is short for host vs and host ace.

More complex filter expressions are built up by combining primitives. For example, "host foo and not port ftp and not port ftp-data. To save typing, identical qualifier lists can be omitted. For example, "tcp dst port ftp or ftp-data or domain" is exactly the same as "tcp dst port ftp or tcp dst port ftp-data or tcp dst port domain.

Primitives may be combined using:

- A parenthesized group of primitives and operators.
- Negation ("!" or "not").
- Concatenation ("and").
- Alternation ("or").

Negation has highest precedence. Alternation and concatenation have equal precedence and associate left to right.

In addition to the above, there are some special "primitive" keywords that do not follow the pattern: gateway, broadcast, less, greater, and arithmetic expressions. For more information on these primitives, see the list below:

For examples see the topic Filter Expression Examples.

Special primitives are:

broadcast
dst host host
dst net net
dst port port
src host host
host host
ether dst ehost
ether host ehost
ether proto protocol
ether src ehost
expr relop expr

gateway host
greater length
ip, arp, rarp
ip proto protocol
less length
net net
port port
src net net
src port port
tcp, udp

See Also

- Filter Dialog Box
- Filtering TCPDump Output

broadcast

True if the packet is a broadcast packet.

dst host host

True if the IP destination field of the packet is host, which may be either an address or a name.

dst net net

True if the IP destination address of the packet has a network number of net, which may be either an address or a name.

dst port port

True if the packet is ip/tcp or ip/udp and has a destination port value of port. The port can be a number or a name. If a name is used, both the port number and protocol are checked. If a number or ambiguous name is used, only the port number is checked (for example, dst port 513 prints both tcp/login traffic and udp/who traffic, and port domain prints both tcp/domain and udp/domain traffic).

src host host

True if the IP source field of the packet is host.

host host

True if either the IP source or destination of the packet is host. Any of the above host expressions can be prefaced with the keywords, ip, arp, or rarp as in:

```
ip host host
```

which is equivalent to:

```
ether proto \ip and host host
```

If host is a name with multiple IP addresses, each address is checked for a match.

ether dst ehost

True if the Ethernet destination address is ehost. Ehost may be either a name or a number.

ether host ehost

True if either the Ethernet source or destination address is ehost.

ether proto protocol

True if the packet is of ether type protocol. Protocol can be a number or a name like ip, arp, or rarp. Note these identifiers are also keywords and must be escaped via backslash (\).

ether src ehost

True if the Ethernet source address is ehost.

expr relop expr

True if the relation holds, where relop is one of >, <, >=, <=, =, !=, and expr is an arithmetic expression composed of integer constants (expressed in standard C syntax), the normal binary operators [+, -, *, /, &, |], a length operator, and special packet data accessors. To access data inside the packet, use the following syntax:

```
proto [ expr : size ]
```

Proto is one of ether, ip, arp, rarp, tcp, or udp, and indicates the protocol layer for the index operation. The byte offset, relative to the indicated protocol layer, is given by expr.

Size is optional and indicates the number of bytes in the field of interest; it can be either one, two, or four, and defaults to one. The length operator, indicated by the keyword len, gives the length of the packet.

For example, "ether[0] & 1 != 0" catches all multicast traffic. The expression "ip[0] & 0xf != 5" catches all IP packets with options. The expression "ip[2:2] & 0x1fff = 0" catches only unfragmented datagrams and frag zero of fragmented datagrams. This check is implicitly applied to the tcp and udp index operations. For instance, tcp[0] always means the first byte of the TCP header, and never means the first byte of an intervening fragment.

gateway host

True if the packet used host as a gateway. That is, the Ethernet source or destination address was host but neither the IP source nor the IP destination was host. Host must be a name and must be found in the host table or DNS. (An equivalent expression is ether host ehost and not host host which can be used with either names or numbers for host / ehost.)

greater length

True if the packet has a length greater than or equal to length. This is equivalent to:
`len >= length`.

ip, arp, rarp

Abbreviations for:

```
ether proto p
```

where p is one of the above protocols.

ip proto protocol

True if the packet is an ip packet of protocol type protocol. Protocol can be a number or one of the names icmp, udp, nd, or tcp. Note that the identifiers tcp and udp are also keywords and must be escaped via backslash (\).

less length

True if the packet has a length less than or equal to length. This is equivalent to:
`len <= length.`

net net

True if either the IP source or destination address of the packet has a network number of net.

port port

True if either the source or destination port of the packet is port. Any of the above port expressions can be prefaced with the keywords, tcp or udp, as in:

```
tcp src port port
```

which matches only tcp packets.

src net net

True if the IP source address of the packet has a network number of net.

src port port

True if the packet has a source port value of port.

tcp, udp

Abbreviations for:

```
ip proto p
```

where p is one of the above protocols.

Filter Expression Examples

To print all packets arriving at or departing from sundown and this computer:

```
HOST SUNDOWN
```

To print all IP packets between this computer and any host except helios:

```
IP NOT HELIOS
```

To print all ftp traffic through gateway snup:

```
GATEWAY SNUP AND (PORT FTP OR FTP-DATA)
```

To print the start and end packets (the SYN and FIN packets) of each TCP conversation

```
TCP[13] & 3 != 0
```

To print IP packets longer than 576 bytes sent through gateway snup:

```
GATEWAY SNUP AND IP[2:2] > 576
```

To print IP broadcast or multicast packets that were not sent via Ethernet broadcast or multicast:

```
ETHER[0] & 1 = 0 AND IP[16] >= 224
```

See Also

- Filter Expressions
- Filter Dialog Box
- Filtering TCPDump Output

How to Read TCPDump Output

The output of TCPDump is protocol dependent. Below is a list of several common protocols. For information on reading TCPDump output for a specific protocol, click that protocol.

- ARP/RARP Packets
- TCP Packets
- UDP Packets
- UDP Name Server Requests
- UDP Name Server Responses
- NFS Requests
- IP Fragments

See Also

- TCPDump Application Window
- Capturing and Displaying Packets

ARP/RARP Packets

ARP/RARP output shows the type of request and its arguments. The format is intended to be self explanatory. Here is a short sample taken from the start of an "RLOGIN" from host rtsg to host csam:

```
arp who-has csam tell rtsg
arp reply csam is-at CSAM
```

The first line says that rtsg sent an arp packet asking for the Ethernet address of internet host csam. Csam replies with its Ethernet address (in this example, Ethernet addresses are in caps and internet addresses in lower case).

TCP Packets

Note: The following description assumes familiarity with the TCP protocol described in RFC-793.

The general format of a tcp protocol line is:

```
src > dst: flags data-seqno ack window urgent options
```

Src and dst are the source and destination IP addresses and ports. Flags are some combination of S (synchronize), F (finish), P (push) or R (reset) or a single "." (no flags). Data-seqno describes the portion of sequence space covered by the data in this packet (see example below). Ack is sequence number of the next data expected the other direction on this connection. Window is the number of bytes of receive buffer space available the other direction on this connection. Urg indicates there is "urgent" data in the packet. Options are tcp options enclosed in angle brackets (for example, <mss 1024>).

Src, dst and flags are always present. The other fields depend on the contents of the packet's tcp protocol header and are output only if appropriate.

Here is the opening portion of an RLOGIN from host rtsg to host csam.

```
rtsg.1023 > csam.login: S 768512:768512(0) win 4096 <mss 1024>
csam.login > rtsg.1023: S 947648:947648(0) ack 768513 win 4096
<mss 1024>
rtsg.1023 > csam.login: . ack 1 win 4096
rtsg.1023 > csam.login: P 1:2(1) ack 1 win 4096
csam.login > rtsg.1023: . ack 2 win 4096
rtsg.1023 > csam.login: P 2:21(19) ack 1 win 4096
csam.login > rtsg.1023: P 1:2(1) ack 21 win 4077
csam.login > rtsg.1023: P 2:3(1) ack 21 win 4077 urg 1
csam.login > rtsg.1023: P 3:4(1) ack 21 win 4077 urg 1
```

The first line says that tcp port 1023 on rtsg sent a packet to port login on csam. The S indicates that the SYN flag was set. The packet sequence number was 768512 and it contained no data. (The notation is "first:last(nbytes)" which means "sequence numbers first up to but not including last which is nbytes bytes of user data.) There was no piggybacked ack, the available receive window was 4096 bytes and there was a max-segment-size option requesting an mss of 1024 bytes.

Csam replies with a similar packet except it includes a piggy-backed ack for rtsg's SYN.

Rtsg then acks csam's SYN. The "." means no flags were set. The packet contained no data so there is no data sequence number. Note that the ack sequence number is a small integer (1). The first time TCPDump sees a tcp "conversation", it prints the sequence number from the packet. On subsequent packets of the conversation, the difference between the current packet's sequence number and this initial sequence number is printed to ease readability. This means that sequence numbers after the first can be interpreted as relative byte positions in the conversation's data stream (with the first data byte each direction being "1").

On the 6th line, rtsg sends csam 19 bytes of data (bytes 2 through 20 in the rtsg -> csam side of the conversation). The PUSH flag is set in the packet. On the 7th line, csam says it's received data sent by rtsg up to but not including byte 21. Most of this data is apparently sitting in the socket buffer since csam's receive window has gotten 19 bytes smaller. Csam also sends one byte of data to rtsg in this packet. On the 8th and 9th lines, csam sends two bytes of urgent, pushed data to rtsg.

UDP Packets

UDP format is illustrated by this rwho packet:

```
actinide.who > broadcast.who: udp 84
```

This says that port who on host actinide sent a udp datagram to port who on host broadcast, the Internet broadcast address. The packet contained 84 bytes of user data.

Some UDP services are recognized (from the source or destination port number) and the higher level protocol information printed. In particular, Domain Name service requests (RFC-1034/1035) and Sun RPC calls (RFC-1050) to NFS.

UDP Name Server Requests

Note: The following description assumes familiarity with the Domain Service protocol described in RFC-1035.

Name server requests are formatted as

```
src > dst: id op?  flags qtype qclass name (len)
```

An example is:

```
h2opolo.1538 > helios.domain: 3+ A?  ucbvax.berkeley.edu. (37)
```

Host h2opolo asked the domain server on helios for an address record (qtype=A) associated with the name ucbvax.berkeley.edu. The query id was "3. The "+" indicates the recursion desired flag was set. The query length was 37 bytes, not including the UDP and IP protocol headers. The query operation was the normal one, Query, so the op field was omitted. If the op had been anything else, it would have been printed between the "3" and the "+. Similarly, the qclass was the normal one, C_IN, and omitted. Any other qclass would have been printed immediately after the "A.

A few anomalies are checked and may result in extra fields enclosed in square brackets: If a query contains an answer, name server or authority section, ancount, nscount, or arcount are printed as "[na]", "[nn]" or "[nau]" where n is the appropriate count. If any of the response bits are set (AA, RA or rcode) or any of the "must be zero" bits are set in bytes two and three, "[b2&3=x]" is printed, where x is the hex value of header bytes two and three.

UDP Name Server Responses

Name server responses are formatted as

```
src > dst:  id op rcode flags a/n/au type class data (len)
```

Examples are:

```
helios.domain > h2opolo.1538: 3 3/3/7 A 128.32.137.3 (273)
helios.domain > h2opolo.1537: 2 NXDomain* 0/1/0 (97)
```

In the first example, helios responds to query id 3 from h2opolo with 3 answer records, 3 name server records and 7 authority records. The first answer record is type A (address) and its data is internet address 128.32.137.3. The total size of the response was 273 bytes, excluding UDP and IP headers. The op (Query) and response code (NoError) were omitted, as was the class (C_IN) of the A record.

In the second example, helios responds to query id 2 with a response code of non-existent domain (NXDomain) with no answers, one name server and no authority records. The "*" indicates that the authoritative answer bit was set. Since there were no answers, no type, class or data were printed.

Other flag characters that might appear are "-" (recursion available, RA, not set) and "|" (truncated message, TC, set). If the "question" section does not contain exactly one entry, "[nq]" is printed.

NFS Requests

Sun NFS (Network File System) requests and replies are printed as:

```
src.xid > dst.nfs: len op args
src.nfs > dst.xid: reply stat len
```

Example packets are:

```
vs.e2766 > helios.nfs: 136 readdir fh 6.5197 8192 bytes @ 0
helios.nfs > vs.e2766: reply ok 384
vs.e2767 > helios.nfs: 136 lookup fh 6.5197 "RCS"
```

In the first line, host vs sends a transaction with id e2766 to helios (note that the number following the src host is a transaction id, not the source port). The request was 136 bytes, excluding the UDP and IP headers. The operation was a readdir (read directory) on file handle (fh) 6.5197. 8192 bytes are read, starting at offset 0. Helios replies "ok" with 384 bytes of data.

In the third line, vs asks helios to look up the name "RCS" in directory file 6.5197. Note that the data printed depends on the operation type. The format is intended to be self explanatory if read in conjunction with an NFS protocol spec.

IP Fragments

Fragmented Internet datagrams are printed as:

```
(frag id:size@offset+)
(frag id:size@offset)
```

(The first form indicates there are more fragments. The second indicates this is the last fragment.)

Id is the fragment id (in hex). Size is the fragment size (in bytes) excluding the IP header. Offset is this fragment's offset (in bytes) in the original datagram.

The fragment information is output for each fragment. The first fragment contains the higher level protocol header and the frag info is printed after the protocol info. Fragments after the first contain no higher level protocol header and the frag info is printed after the source and destination addresses. For example, here is part of an ftp from arizona.edu to lbl-rtsg.arpa over a CSNET connection that does not appear to handle 576 byte datagrams:

```
arizona.ftp-data > rtsg.1170: . 1024:1332(308) ack 1 win 4096
(frag 595a:328@0+)
arizona > rtsg: (frag 595a:204@328)
rtsg.1170 > arizona.ftp-data: . ack 1536 win 2560
```

There are a couple of things to note here: First, addresses in the 2nd line do not include port numbers. This is because the TCP protocol information is all in the first fragment, so that the port or sequence numbers cannot be determined for the later fragments. Second, the tcp sequence information in the first line is printed as if there were 308 bytes of user data when, in fact, there are 512 bytes (308 in the first frag and 204 in the second). If you are looking for holes in the sequence space or trying to match up acks with packets, be sure to add up the fragments.

A packet with the IP do not-fragment flag is marked with a trailing (DF).