Ethereal User's Guide

V1.1 for Ethereal 0.8.19

Table of Contents

Foreword

Ethereal is one of those packages that many network managers would love to be able to use, but they are often prevented from getting what they would like from Ethereal because of the lack of documentation.

This document is part of an effort on the part of the Ethereal team to improve the accessibility of Ethereal.

We hope that you find it useful, and look forward to your comments.

Acknowledgments

I would like to thank the whole Ethereal team for their assistance. In particular, I would like to thank:

- Gerald Combs, for initiating the Ethereal project and funding me to do this documentation.
- Guy Harris, for many helpful hints and a great deal of patience in reviewing this document.
- Gilbert Ramirez, for general encouragement and helpful hints along the way.

I would also like to thank the following people for their helpful feedback on this document:

- Pat Eyler, for his suggestions on improving the example on generating a **backtrace**.

I would like to acknowledge those man page and README authors for the ethereal project from who sections of this document borrow heavily:

- Scott Renfro from whose **mergecap** man page Section 5.4 derived.
- Ashok Narayanan from whose **text2pcap** man page Section 5.5 derived.
- Frank Singleton from whose `README.idl2eth` Section 5.6 derived.

1. Introduction

1.1. About this manual

This manual was originally developed by Richard Sharpe (mailto:rsharpe@ns.aus.com) with funds provided from the Ethereal Fund. More recently, it was updated by Ed Warnicke (mailto:hagbard@physics.rutgers.edu).

It is written in DocBook/SGML for the moment.

1.2. What is Ethereal?

Every network manager at some time or other needs a tool that can capture packets off the network and analyze them. In the past, such tools were either very expensive, propietary, or both. However, with the advent of Ethereal, all that has changed.

Ethereal is perhaps one the best open source packet sniffers available today. The following are some of the features Ethereal provides:

- Available for UNIX and Windows.
- Capture and display packets from any interface on a UNIX system.
- Display packets captured under a number of other capture programs:
  - tcpdump
  - Network Associates Sniffer and Sniffer Pro
  - NetXray
  - LANalyzer
  - Shomiti
  - AIX’s iptrace
  - RADCOM’s WAN/LAN Analyzer
  - Lucent/Ascend access products
  - HP-UX’s nettl
  - Toshiba’s ISDN routers
  - ISDN4BSD *i4btrace* utility
  - Microsoft Network Monitor
  - Sun snoop

- Save captures to a number of formats:
  - libpcap (tcpdump)
  - Sun snoop
  - Microsoft Network Monitor
  - Network Associates Sniffer
- Filter packets on many criteria.
- Search for packets using filters.
- Colorize packet display based on filters

However, to really appreciate its power, you have to start using it.

Figure 1-1 shows Ethereal having captured some packets and waiting for you to examine the packets.

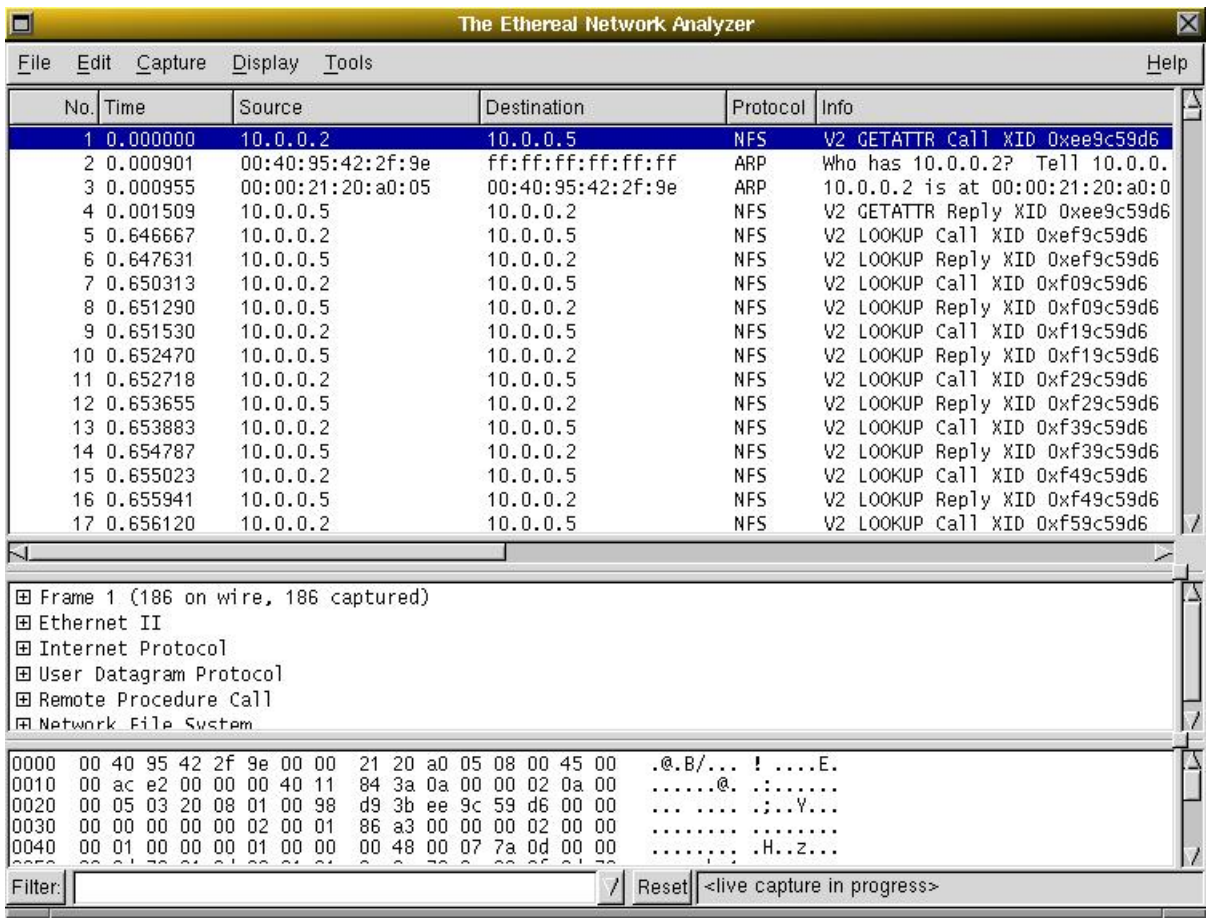

Figure 1-1. Ethereal captures packets and allows you to examine their content.

In addition, because all the source code for Ethereal is freely available, it is very easy for people to add new protocols to Ethereal, either as modules, or built into the source.

There are currently protocol decoders (or dissectors, as they are known in Ethereal), for a great many protocols, including:

- 802.1q Virtual LAN
- AOL Instant Messenger
- ATM
- ATM LAN Emulation
- Address Resolution Protocol
- Andrew File System (AFS)
- Appletalk Address Resolution Protocol
- Async data over ISDN (V.120)
- Authentication Header
- BACnet Virtual Link Control
- Banyan Vines
- Banyan Vines Fragmentation Protocol
- Banyan Vines SPP
- Blocks eXtensible eXchange Protocol
- Boot Parameters
- Bootstrap Protocol
- Border Gateway Protocol
- Building Automation and Control Network APDU
- Building Automation and Control Network NPDU
- Cisco Auto-RP
- Cisco Discovery Protocol
- Cisco Group Management Protocol
- Cisco HDLC
- Cisco Hot Standby Router Protocol
- Cisco ISL
- Cisco Interior Gateway Routing Protocol

- Cisco SLARP
- Common Open Policy Service
- Common Unix Printing System (CUPS) Browsing Protocol
- DCE RPC
- DCE/RPC Conversation Manager
- DCE/RPC Endpoint Mapper
- DCE/RPC Remote Management
- DCOM OXID Resolver
- DCOM Remote Activation
- DEC Spanning Tree Protocol
- DG Gryphon Protocol
- Data
- Data Stream Interface
- Datagram Delivery Protocol
- Diameter Protocol
- Distance Vector Multicast Routing Protocol
- Domain Name Service
- Dynamic DNS Tools Protocol
- Encapsulating Security Payload
- Enhanced Interior Gateway Routing Protocol
- Ethernet
- FTP Data
- Fiber Distributed Data Interface
- File Transfer Protocol (FTP)
- Frame
- Frame Relay
- GARP VLAN Registration Protocol
- GPRS Tunneling Protocol
- General Inter-ORB Protocol
- Generic Routing Encapsulation
- Gnutella Protocol

- Hummingbird NFS Daemon
- Hypertext Transfer Protocol
- ICQ Protocol
- IEEE 802.11 wireless LAN
- IEEE 802.11 wireless LAN management frame
- ILMI
- IP Payload Compression
- IPX Message
- IPX Routing Information Protocol
- ISDN Q.921-User Adaptation Layer
- ISDN User Part
- ISIS HELLO
- ISO 10589 ISIS Complete Sequence Numbers Protocol Data Unit
- ISO 10589 ISIS InTRA Domain Routeing Information Exchange Protocol
- ISO 10589 ISIS Link State Protocol Data Unit
- ISO 10589 ISIS Partial Sequence Numbers Protocol Data Unit
- ISO 8073 COTP Connection-Oriented Transport Protocol
- ISO 8473 CLNP ConnectionLess Network Protocol
- ISO 8602 CLTP ConnectionLess Transport Protocol
- ISO 9542 ESIS Routeing Information Exchange Protocol
- ITU-T Recommendation H.261
- Internet Cache Protocol
- Internet Control Message Protocol
- Internet Control Message Protocol v6
- Internet Group Management Protocol
- Internet Message Access Protocol
- Internet Printing Protocol
- Internet Protocol
- Internet Protocol Version 6
- Internet Relay Chat
- Internet Security Association and Key Management Protocol

- Internetwork Packet eXchange
- Kerberos
- Kernel Lock Manager
- Label Distribution Protocol
- Layer 2 Tunneling Protocol
- Lightweight Directory Access Protocol
- Line Printer Daemon Protocol
- Link Access Procedure Balanced (LAPB)
- Link Access Procedure Balanced Ethernet (LAPBETHER)
- Link Access Procedure, Channel D (LAPD)
- Linux cooked-mode capture
- Local Management Interface
- Logical-Link Control
- Lucent/Ascend debug output
- MAPI
- MS Proxy Protocol
- MSNIP : Multicast Source Notification of Interest Protocol
- MTP 3 User Adaptation Layer
- MTP2 Peer Adaptation Layer
- Malformed Frame
- Media Gateway Control Protocol
- Message Transfer Part Level 3
- Microsoft Windows Browser Protocol
- Microsoft Windows Lanman Protocol
- Microsoft Windows Logon Protocol
- Mobile IP
- Modbus/TCP
- Mount Service
- MultiProtocol Label Switching Header
- Multicast Router DISCovery protocol
- Multicast Source Discovery Protocol

- NIS+
- NIS+ Callback
- Name Binding Protocol
- Name Management Protocol over IPX
- NetBIOS
- NetBIOS Datagram Service
- NetBIOS Name Service
- NetBIOS Session Service
- NetBIOS over IPX
- NetWare Core Protocol
- Network File System
- Network Lock Manager Protocol
- Network News Transfer Protocol
- Network Status Monitor CallBack Protocol
- Network Status Monitor Protocol
- Network Time Protocol
- Null/Loopback
- Open Shortest Path First
- PPP IP Control Protocol
- PPP Link Control Protocol
- PPP Multilink Protocol
- PPP Password Authentication Protocol
- PPP-over-Ethernet Discovery
- PPP-over-Ethernet Session
- Point-to-Point Protocol
- Point-to-Point Tunnelling Protocol
- Portmap
- Post Office Protocol
- Pragmatic General Multicast
- Protocol Independent Multicast
- Q.2931

- Q.931
- Quake II Network Protocol
- Quake Network Protocol
- QuakeWorld Network Protocol
- RFC 2250 MPEG1
- RIPng
- RX Protocol
- Radio Access Network Application Part
- Radius Protocol
- Real Time Streaming Protocol
- Real-Time Transport Protocol
- Real-time Transport Control Protocol
- Remote Procedure Call
- Remote Quota
- Remote Shell
- Remote Wall protocol
- Resource ReserVation Protocol (RSVP)
- Rlogin Protocol
- Routing Information Protocol
- Routing Table Maintenance Protocol
- SCCP user adaptation layer light
- SMB (Server Message Block Protocol)
- SMB MailSlot Protocol
- SNMP Multiplex Protocol
- SPRAY
- SSCOP
- Secure Socket Layer
- Sequenced Packet eXchange
- Service Advertisement Protocol
- Service Location Protocol
- Session Announcement Protocol

- Session Description Protocol
- Session Initiation Protocol
- Short Frame
- Simple Mail Transfer Protocol
- Simple Network Management Protocol
- Sinec H1 Protocol
- Socks Protocol
- Spanning Tree Protocol
- Stream Control Transmission Protocol
- Syslog message
- Systems Network Architecture
- TACACS
- TACACS+
- TPKT
- Telnet
- Time Protocol
- Token-Ring
- Token-Ring Media Access Control
- Transmission Control Protocol
- Transparent Network Substrate Protocol
- Trivial File Transfer Protocol
- User Datagram Protocol
- Virtual Router Redundancy Protocol
- Virtual Trunking Protocol
- Web Cache Coordination Protocol
- Wellfleet Compression
- Who
- Wireless Session Protocol
- Wireless Transaction Protocol
- Wireless Transport Layer Security
- X.25

- X.25 over TCP
- X11
- Yahoo Messenger Protocol
- Yellow Pages Bind
- Yellow Pages Passwd
- Yellow Pages Service
- Yellow Pages Transfer
- Zebra Protocol
- iSCSI

1.3. The status of Ethereal

Ethereal is an open source software project, and is released under the Gnu Public Licence (http://www.gnu.org/copyleft/gpl.html) (GPL). All source code is freely available under the GPL. You are welcome to modify Ethereal to suit your own needs, and it would be appreciated if you contribute your improvements back to the Ethereal team.

You gain two benefits by contributing your improvements back to the community:

- Other people who find your contributions useful will appreciate them, and you will know that you have helped people in the same way that the developers of Ethereal have helped people
- The maintainers and developers of Ethereal will maintain your code as well, fixing it when API changes or other changes are made, and generally keeping it in tune with what is happening with Ethereal.

The Ethereal source code and binary kits for some platforms are all available on the Ethereal website: http://www.ethereal.com.

1.4. Development and maintenance of Ethereal

Ethereal was initially developed by Gerald Combs. Ongoing development and maintenance of Ethereal is handled by the Ethereal team, a loose group of individuals who fix bugs and provide new functionality.

There have also been a large number of people who have contributed protocol dissectors to Ethereal, and it is expected that this will continue. You can find a list of

the people who have contributed code to Ethereal at the authors (http://www.ethereal.com/introduction.html#authors) link on the web site.

1.5. A rose by any other name

William Shakespeare wrote: *"A rose by any other name would smell as sweet."* And so it is with Ethereal, as there appears to be two different ways that people pronounce the name.

Some people pronounce it ether-real, while others pronounce it e-the-real, as in ghostly, insubstantial, etc.

You are welcome to call it what you like, as long as you find it useful.

1.6. A brief history of Ethereal

In late 1997, Gerald Combs needed a tool for tracking down networking problems and wanted to learn more about networking, so he started writing Ethereal as a way to solve both problems.

Ethereal was initially released, after several pauses in development, in July 1998 as version 0.2.0. Within days, patches, bug reports, and words of encouragement started arriving, so Ethereal was on its way to success.

Not long after that Gilbert Ramirez saw its potential and contributed a low-level dissector to it.

In October, 1998, Guy Harris, of NetApp was looking for something better than TCPview, so he started applying patches and contributing dissectors to Ethereal.

In late 1998, Richard Sharpe, who was giving TCP/IP courses, saw its potential on such courses, started looking at it to see if it supported the protocols he needed. While it didn’t at that point, new protocols could be easily added. So he started contributing dissectors and contributing patches.

The list of people who have contributed to Ethereal is long, and almost all of them started with a protocol that they needed that Ethereal did not already handle, so they copied an existing dissector and contributed the code back to the team. You can get a list of the people who have contributed by checking the man pages for ethereal, or from the website (http://www.ethereal.com).

1.7. Platforms Ethereal runs on

Ethereal currently runs on most UNIX platforms and the various Windows platforms. It requires GTK+, GLIB and libpcap in order to run.

Binary packages are available for at least the following platforms:

- AIX
- Tru64 UNIX (formerly Digital UNIX)
- Debian GNU/Linux
- Slackware Linux
- Red Hat Linux
- FreeBSD
- NetBSD
- OpenBSD
- HP/UX
- Sparc/Solaris 8
- Windows 2000, Windows NT and Windows Me/98/95

If a binary package is not available for your platform, you should download the source and try to build it.

1.8. Where to get Ethereal

You can get the latest copy of the Ethereal from the Ethereal Website: http://www.ethereal.com. The website allows you to choose from among several mirrors for downloading.

1.9. Reporting problems and getting help

If you have problems, or need help with Ethereal, there are several mailing lists that may be of interest to you:

Ethereal Users

This list is for users of Ethereal. People post with questions about building and using Ethereal. Others provide answers.

Ethereal Announce

This list is for people wanting to receive announcements about Ethereal.

Ethereal Dev

This list is for Ethereal developers. If you want to start developing a protocol dissector, join this list.

You can subscribe to each of these from the Ethereal web site: http://www.ethereal.com. Simply select the **mailing lists** link on the left hand side of the site. The lists are archived at the Ethereal web site as well.

When reporting crashes with Ethereal, it is helpful if you supply the following information:

1. The version number of Ethereal you found the problem with, eg Ethereal 0.8.10.
2. The version number of the other software linked with Ethereal, eg GTK+, etc. You can obtain this with the command **ethereal -v**.
3. A traceback if Ethereal crashed. You can obtain this with the following commands:

```
$ gdb ‘whereis ethereal | cut -f2 -d: | cut -f’ ’ -d2‘ core >& bac
backtrace
^D
$
```

Type the characters in the first line verbatim! Those are back-tics there!

backtrace is a **gdb** command. You should enter it verbatim after the first line shown above. The ^D (Control-D, that is, press the Control key and the D key together) will cause **gdb** to exit. This will leave you with a file called `backtrace.txt` in the current directory. Include the file with your bug report.

If you do not have **gdb** available, you will have to check out your operating system’s debugger. Windows users might not be able to get a traceback.

You should mail the traceback to the **ethereal-dev** mailing list.

1.10. Where to get the latest copy of this document

The latest copy of this documentation can always be found at:
http://www.ns.aus.com/ethereal/user-guide/book1.html; and at:
http://www.ethereal.com/docs/user-guide/ (http://www.ethereal.com).

In addition, you can find a PDF version of the guide at:
http://www.ns.aus.com/ethereal/user-guide/user-guide-a4.pdf in A4 and
http://www.ns.aus.com/ethereal/user-guide/user-guide-usletter.pdf in US Letter.

1.11. Providing feedback

Should you have any feedback about this document, please send them to the author at rsharpe@ns.aus.com (mailto:rsharpe@ns.aus.com).

2. Building and Installing Ethereal

2.1. Introduction

As with all things, there must be a beginning, and so it is with Ethereal. To use Ethereal, you must:

- Obtain a binary package for your operating system, or
- Obtain the source and build Ethereal for your operating system.

Currently, only two or three Linux Distributions ship ethereal, and they are commonly shipping an out-of-date version. No other versions of UNIX ship Ethereal so far, and Microsoft does not ship it with any version of Windows. For that reason, you will need to know where to get the latest version of Ethereal and how to install it. The current version of Ethereal is 0.8.19.

This chapter shows you how to obtain source and binary packages, and how to build Ethereal from source, should you choose to do so.

The following are the general steps you would use:

1. Download the relevant package for your needs, eg, source or binary distribution.
2. Build the source into a binary, if you have downloaded the source

   This may involve building and/or installing any other necessary packages.

3. Install the binaries in their final destinations.

2.2. Obtaining the source and binary distributions

You can obtain both source and binary distributions from the Ethereal web site: http://www.ethereal.com. Simply select the download link, and then select either the source package or binary package of your choice from the mirror site closest to you.

In general, unless you have already downloaded Ethereal before, you will most likely need to down load several source packages if you are building Ethereal from source. This is covered in more detail below.

Once you have downloaded the relevant files, you can go on to the next step.

While you will find a number of binary packages available on the Ethereal web site, you might not find one for your platform, and they often tend to be several versions behind the current released version, as they are contributed by people who have the platforms they are built for.
For this reason, you might want to pull down the source distribution and build it, as the process is relatively simple.

2.3. Before you build Ethereal

Before you build Ethereal from sources, or install a binary package, you must ensure that you have the following other packages installed:

- GTK+, The GIMP Tool Kit.

  You will also need Glib. Both can be obtained from www.gtk.org (http://www.gtk.org)

- libpcap, the packet capture software that Ethereal uses.

  You can obtain libpcap from www.tcpdump.org (http://www.tcpdump.org)

Depending on your system, you may be able to install these from binaries, eg RPMs, or you may need to obtain them in source code form and build them.

If you have downloaded the source for GTK+, the instructions shown in Example 2-1 may provide some help in building it:

```
gzip -dc gtk+-1.2.8.tar.gz | tar xvf -
<much output removed>
cd gtk+-1.2.8
./configure
<much output removed>
make
<much output removed>
make install
<much output removed>
```

Example 2-1. Building GTK+ from source

You may need to change the version number of gtk+ in Example 2-1 to match the version of GTK+ you have downloaded. The directory you change to will change if the version of GTK+ changes, and in all cases, **tar xvf -** will show you the name of the directory you should change to.

If you use Linux, or have GNU **tar** installed, you can use **tar zxvf gtk+-1.2.8.tar.gz**. It is also possible to use **gunzip -c** or **gzcat** rather than **gzip -dc** on many UNIX systems.

If you downloaded gtk+ or any other tar file using Windows, you may find your file called gtk+-1_2_8_tar.gz.

You should consult the GTK+ web site if any errors occur in carrying out the instructions in Example 2-1.

If you have downloaded the source to libpcap, the general instructions shown in Example 2-2 will assist in building it. Also, if your operating system does not support **tcpdump**, you might also want to download it from the tcpdump (http://www.tcpdump.org) web site and install it.

```
gzip -dc libpcap-0.5.tar.Z | tar xvf -
<much output removed>
cd libpcap_0_5rel2
./configure
<much output removed>
make
<much output removed>
make install
<much output removed>
make install-incl
<much output removed>
```

Example 2-2. Building and installing libpcap

The directory you should change to will depend on the version of libpcap you have downloaded. In all cases, **tar xvf -** will show you the name of the directory that has been unpacked.

When installing the include files, you might get the error shown in Example 2-3 when you submit the command **make install-incl**.

```
   /usr/local/include/pcap.h
/usr/bin/install -c -m 444 -o bin -g bin ./pcap-namedb.h \
   /usr/local/include/pcap-namedb.h
/usr/bin/install -c -m 444 -o bin -g bin ./net/bpf.h \
   /usr/local/include/net/bpf.h
/usr/bin/install: cannot create regular file \
```

‘/usr/local/include/net/bpf.h’: No such file or directory

make: *** [install-incl] Error 1

Example 2-3. Errors while installing the libpcap include files

If you do, simply create the missing directory with the following command:

```
mkdir /usr/local/include/net
```

and rerun the command **make install-incl**.

Under RedHat 6.x and beyond (and distributions based on it, like Mandrake) you can simply install each of the packages you need from RPMs. Most Linux systems will install GTK+ and Glib in anycase, however, you will probably need to install the devel versions of each of these packages. The commands shown in Example 2-4 will install all the needed RPMs if they are not already installed.

```
cd /mnt/cdrom/RedHat/RPMS
rpm -ivh glib-1.2.6-3.i386.rpm
rpm -ivh glib-devel-1.2.6-3.i386.rpm
rpm -ivh gtk+-1.2.6-7.i386.rpm
rpm -ivh gtk+-devel-1.2.6-7.i386.rpm
rpm -ivh libpcap-0.4-19.i386.rpm
```

Example 2-4. Installing required RPMs under RedHat Linux 6.2 and beyond

If you are using a version of RedHat later than 6.2, the required RPMs have most likely changed. Simply use the correct RPMs from your distribution.

Under Debian you can install ethereal using apt-get. apt-get will handle any dependency issues for you. Example 2-5 shows how to do this.

```
apt-get install ethereal
```

Example 2-5. Installing debs under Debian

2.4. Building from Source under UNIX

Use the following general steps if you are building Ethereal from source under a

UNIX operating system:

1. Unpack the source from its **gzip**'d **tar** file. If you are using Linux, or your version of UNIX uses GNU **tar**, you can use the following command:

   ```
   tar zxvf ethereal-0.8.19-tar.gz
   ```

   For other versions of UNIX, You will want to use the following commands:

   ```
   gzip -d ethereal-0.8.19-tar.gz
   tar xvf ethereal-0.8.19-tar
   ```

   

   The pipeline **gzip -dc ethereal-0.8.19-tar.gz | tar xvf -** will work here as well.

   

   If you have downloaded the Ethereal tarball under Windows, you may find that your browser has created a file with underscores rather than periods in its file name.

2. Change directory to the ethereal source directory.
3. Configure your source so it will build correctly for your version of UNIX. You can do this with the following command:

   ```
   ./configure
   ```

   If this step fails, you will have to rectify the problems and rerun **configure**. Troubleshooting hints are provided in Section 2.10.

4. Build the sources into a binary, with the **make** command. For example:

   ```
   make
   ```

5. Install the software in its final destination, using the command:

   ```
   make install
   ```

Once you have installed Ethereal with **make install** above, you should be able to run it by entering **ethereal**.

2.5. Installing the binaries under UNIX

In general, installing the binary under your version of UNIX will be specific to the

installation methods used with your version of UNIX. For example, under AIX, you would use **smit** to install the Ethereal binary package, while under Tru64 UNIX (formerly Digital UNIX) you would use **setld**.

2.6. Installing from RPMs under Linux

Use the following command to install the Ethereal RPM that you have downloaded from the Ethereal web site:

```
rpm -ivh ethereal-0.8.10-1.i386.rpm
```

If the above step fails because of missing dependencies, install the dependencies first, and then retry the step above. See Example 2-4 for information on what RPMs you will need to have installed.

2.7. Installing from debs under Debian

Use the following command to install Ethereal under Debian:

```
apt-get install ethereal
```

apt-get should take care of all of the dependency issues for you.

2.8. Building from source under Windows

Unfortunately the current revisor of this document has never built Ethereal under Windows and is thus not competent to write this section. Hopefully this will be remedied in the future.

2.9. Installing Ethereal under Windows

In this section we explore installing Ethereal under Windows from the binary packages. You must follow two steps:

1. Install WinPcap. There are instructions at the WinPcap web site for installing it under Windows 9X, Windows NT and Windows 2000. These are located at: http://netgroup-serv.polito.it/winpcap/install/Default.htm.

2. Install Ethereal. You may acquire a binary installable of Ethereal at http://www.ethereal.com/download.html#binaries. Download the installer (after installing WinPcap) and execute it.

2.10. Troubleshooting during the install

A number of errors can occur during the installation process. Some hints on solving these are provided here.

If the **configure** stage fails, you will need to find out why. You can check the file `config.log` in the source directory to find out what failed. The last few lines of this file should help in determining the problems.

The standard problems are that you do not have GTK+ on your system, or you do not have a recent enough version of GTK+. The **configure** will also fail if you do not have libpcap (at least the required include files) on your system.

Another common problem is for the final compile and link stage to terminate with a complaint of: Output to long. This is likely being caused by an antiquated **sed** (like that shipped with Solaris). Since **sed** is used by the **libtool** script to construct the final link command, this leads to mysterious problems. This can be resolved by downloading sed from http://www.gnu.org/directory/sed.html.

If you cannot determine what the problems are, send mail to the **ethereal-dev** mailing list explaining your problem, and including the output from `config.log` and anything else you think is relevant, like a trace of the **make** stage.

3. Using Ethereal

3.1. Introduction

By now you have installed Ethereal and are most likely keen to get started capturing your first packets. In this chapter we explore:

- How to start Ethereal
- How to capture packets in Ethereal
- How to view packets Ethereal
- How to filter packets in Ethereal

In fact, most of the functionality of Ethereal is explored in this chapter.

3.2. Starting Ethereal

You can start Ethereal from the command line under UNIX, but it can also be started from most Window managers as well. In this section we will look at starting it from the command line.

Before looking at the command line parameters Ethereal understands, lets look at Ethereal itself. Figure 3-1 shows Ethereal as you would usually see it.

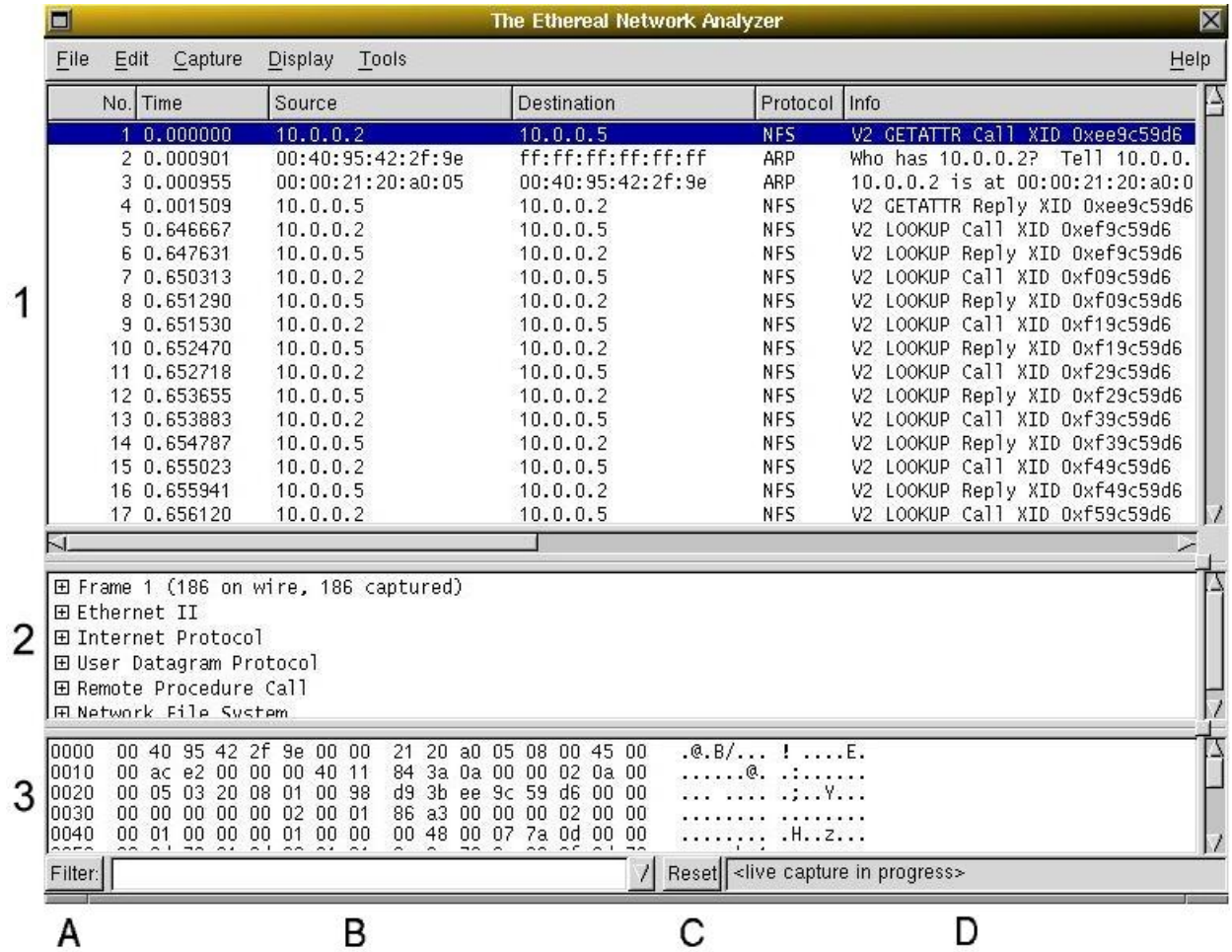

Figure 3-1. Ethereal is comprised of three main windows

Ethereal is comprised of three main windows, or panes.

1. The top pane is the packet list pane. It displays a summary of each packet captured. By clicking on packets in this pane your control what is displayed in the other two panes.
2. The middle pane is the tree view pane. It displays the packet selected in the top pane in more detail.
3. The bottom pane is the data view pane. It displays the data from the packet selected in the top pane, and highlights the field selected in the tree view pane.

In addition to the three main panes, there are four elements of interest on the bottom of the Ethereal main window.

A. The lower leftmost button labeled "Filter:" can be clicked to bring up the filter construction dialog.

B. The left middle text box provides an area to enter or edit filter strings. This is also where the current filter in effect it displayed. You can click on the pull down arrow to select past filter string from a list. More information on display filter strings is available in Section 3.10

C. The right middle button labeled "Reset" clears the current filter.

D. The right text box displays informational messages. These message may indicate whether or not you are capturing, what file you have read into the packet list pane if you are not capturing. If you have selected a protocol field from the tree view pane and it is possible to filter on that field then the filter label for that protocol field will be displayed.

Ethereal supports a large number of command line parameters. To see what they are, simply enter the command **ethereal -h** and the help information shown in Example 3-1 should be printed.

```
This is GNU ethereal 0.8.19, compiled with GTK+ 1.2.10, with GLib 1.2
cap 0.6, with libz 1.1.3, with UCD SNMP 4.2.1
ethereal [ -vh ] [ -klpQS ] [ -B <byte view height> ] [ -c <count> ]
 [ -f <capture filter> ] [ -i <interface> ] [ -m <medium font> ]
 [ -n ] [ -N <resolving> ]
 [ -o <preference setting> ] ... [ -P <packet list height> ]
 [ -r <infile> ] [ -R <read filter> ] [ -s <snaplen> ]
 [ -t <time stamp format> ] [ -T <tree view height> ] [ -w <savefile>
```

Example 3-1. Help information available from Ethereal

We will examine each of these possible command line options in turn.

The first thing to notice is that issuing the command **ethereal** by itself will bring up Ethereal. However, you can include as many of the command line parameters as you like. Their meanings are as follows (in alphabetical order):

-B <byte view height>

This option sets the initial height of the byte view pane. This pane is the bottom pane in the Ethereal display

-c <count>

This option specifies the number of packets to capture when capturing live data. It would be used in conjunction with the **-k** option.

-b <bold font>

This option sets the name of the bold font that Ethereal uses for data in the byte view pane when it is highlighted (ie, selected in the protocol pane

-D

This option changes the way Ethereal deals with the original IPv4 TOS field, so that rather than treating it as the Differentiated Services Field, it is treated as a Type of Service field.

-f <capture filter>

This option sets the initial capture filter expression to be used when capturing packets.

-h

The **-h** option requests Ethereal to print its version and usage instructions and exit.

-i <interface>

The **-i** option allows you to specify, from the command line, which interface packet capture should occur on if capturing packets.

An example would be: **ethereal -i eth0**.

To get a listing of all the interfaces you can capture on, use the command **ifconfig -a** or **netstat -i**. Unfortunately, some versions of UNIX do not support **ifconfig -a**, so you will have to use **netstat -i** in these cases.

-k

The **-k** option specifies that Ethereal should start capturing packets immediately. This option requires the use of the **-i** parameter to specify the interface that packet capture will occur from.

-l

This option turns on automatic scrolling if the packet list pane is being updated automatically as packets arrive during a capture (as specified by the **-S** flag).

-m <medium font>

This option sets the name of the font used for most text displayed by Ethereal.

-n

This option specifies that Ethereal not perform address to name translation nor to translate TCP and UDP ports into names.

-N <resolving>

Turns on name resolving for particular types of addresses and port numbers; the argument is a string that may contain the letters **m** to enable MAC address

resolution, **n** to enable network address resolution, and **t** to enable transport-layer port number resolution. This overrides **-n** if both **-N** and **-n** are present.

-o <preference settings>

Sets a preference value, overriding the default value and any value read from a preference file. The argument to the flag is a string of the form prefname:value, where prefname is the name of the preference (which is the same name that would appear in the preference file), and value is the value to which it should be set. Multiple instances of **-o <preference settings>** can be given on a single command line.

An example of setting a single preference would be:

ethereal -o mgcp.display_dissect_tree:TRUE

An example of setting multiple preferences would be:

ethereal -o mgcp.display_dissect_tree:TRUE -o mgcp.udp.callagent_port:2627

-p

Don’t put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, -p cannot be used to ensure that the only traffic that is captured is traffic sent to or from the machine on which Ethereal is running, broadcast traffic, and multicast traffic to addresses received by that machine.

-P <packet list height>

This option sets the initial height of the packet list pane, ie, the top pane.

-Q

This option forces Ethereal to exit when capturing is complete. It can be used with the **-c** option. It must be used in conjunction with the **-i** and **-w** options.

-r <infile>

This option provides the name of a capture file for Ethereal to read and display. This capture file can be in one of the formats Ethereal understands, including:

- libpcap
- Net Mon
- Snoop
- NetXray

For a complete list, see the Ethereal man pages (**man ethereal**).

-R <read filter>

This option specifies a capture filter to be applied when reading packets from a capture file. The syntax of this filter is that of the display filters discussed in Section 3.10. Packets not matching the filter are discarded.

-s <snaplen>

This option specifies the snapshot length to use when capturing packets. Ethereal will only capture **<snaplen>** bytes of data for each packet.

-S

This option specifies that Ethereal will display packets as it captures them. This is done by capturing in one process and displaying them in a separate process.

-t <time stamp format>

This option sets the format of packet timestamps that are displayed in the packet list window. The format can be one of:

- **r**, which specifies timestamps are displayed relative to the first packet captured.
- **a**, which specifies that actual dates and times be displayed for all packets.
- **d**, which specifies that timestamps are relative to the previous packet.

-T <tree view height>

This option sets the initial height of the tree view pane.

-v

The **-v** option requests Ethereal to print out its version information and exit.

-w <savefile>

This option sets the name of the **savefile** to be used when saving a capture file.

3.3. The Ethereal menus

The Ethereal menu sits across the top of the Ethereal window. An example is shown

in Figure 3-2.

Figure 3-2. The Ethereal Menu

It contains the following items:

File

This menu contains menu-items to open and reread capture files, save capture files, print capture files, print packets, and to quit from Ethereal.

Edit

This menu contains menu-items to find a frame and goto a frame,mark one or more frames, set your preferences, create filters, and enable or disable the dissection of protocols (cut, copy, and paste are not presently implemented).

Capture

This menu allows you to start and stop captures.

Display

This menu contains menu-items to modify display options, match selected frames, colorize frames, expand all frames, collapse all frames, show a packet in a separate window, and configure user specified decodes.

Tools

This menu contains menu-items to display loaded plugins, follow a TCP stream, obtain a summary of the packets that have been captured, and display protocol hierarchy statistics.

Help

This menu contains the About Ethereal... menu item and access to some basic Help.

Each of these are described in more detail in the sections that follow.

3.3.1. The Ethereal File menu

The Ethereal file menu contains the fields shown in Table 3-1.

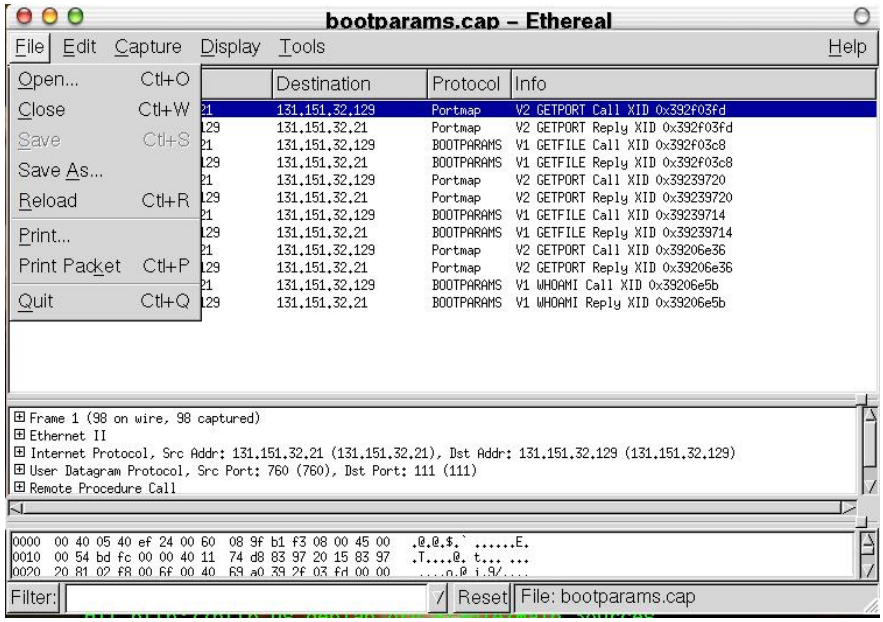

Figure 3-3. Ethereal File Menu

Menu Item	Accelerator	Description
Open...	Ctrl-O	This menu item brings up the file open dialog box that allows you to load a capture file for viewing. It is discussed in more detail in Section 3.9.1.
Close	Ctrl-W	This menu item closes the current capture. If you have not saved the capture, it is lost.
Save	Ctrl-S	This menu item saves the current capture. If you have not set a default capture file name (perhaps with the -w <capfile> option), Ethereal pops up the Save Capture File As dialog box (which is discussed further in Section 3.8.1).
		If you have already saved the current capture, this menu will
		You cannot save a live capture while it is in progress. You mu save.
Save As...		This menu item allows you to save the current capture file to whatever file you would like. It pops up the Save Capture File As dialog box (which is discussed further in Section 3.8.1).
Reload	Ctrl-R	This menu item allows you to reload the current capture file. This menu item is no longer needed, and may be removed in future releases of Ethereal

Menu Item	Accelerator	Description
Print...		This menu item allows you to print all the packets in the capture file. It pops up the Ethereal Print dialog box (which is discussed further in Section 3.16).
Print Packet	Ctrl-P	This menu item allows you to print the current packet.
Quit	Ctrl-Q	This menu item allows you to quit from Ethereal. In the current release of Ethereal (0.8.19), Ethereal silently exits even if you have not saved the current capture file. This may be changed in a future release of Ethereal.

Table 3-1. File menu

3.3.2. The Ethereal Edit menu

The Ethereal Edit menu contains the fields shown in Table 3-2.

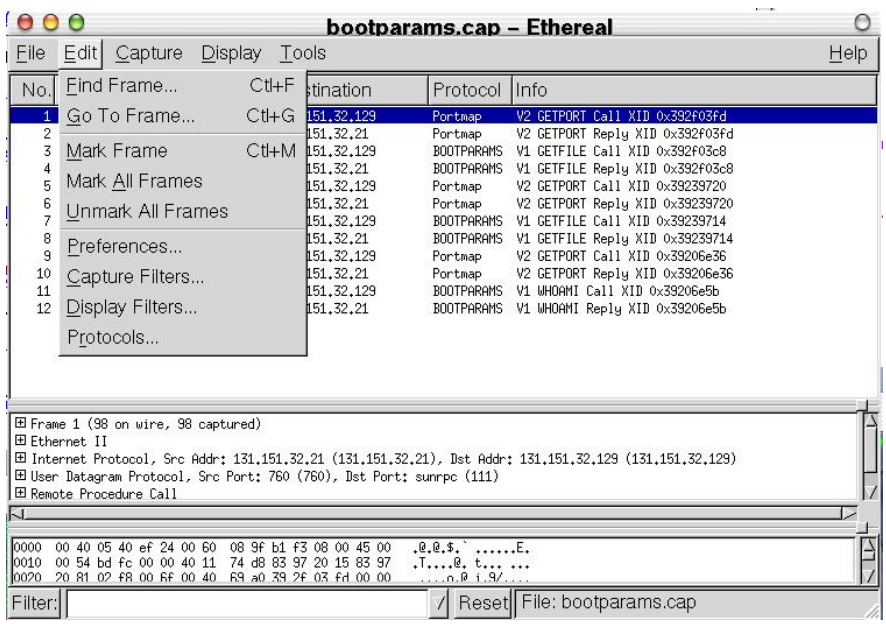

Figure 3-4. Ethereal Edit Menu

Menu Item	Accelerator	Description

Menu Item	Accelerator	Description
Find Frame...	Ctrl-F	This menu item brings up a dialog box that allows you to find a frame by entering an Ethereal display filter. There is further information on finding frames in Section 3.12.
Go to Frame...	Ctrl-G	This menu item brings up a dialog box that allows you to specify a frame to goto by frame number.
Mark Frame	Ctrl-M	This menu item "marks" the currently selected frame. See Section 3.8.1 for more information about saving marked frames.
Mark All Frames		This menu item "marks" all frames. See Section 3.8.1 for more information about saving marked frames.
Unmark All Frames		This menu item "unmarks" all marked frames.
Preferences...		This menu item brings up a dialog box that allows you to set preferences for many parameters that control Ethereal. You can also save your preferences so Ethereal will use them the next time you start it. More detail is provided in Section 3.17
Capture Filters...		This menu item brings up a dialog box that allows you to create and edit capture filters. You can name filters, and you can save them for future use. More detail on this subject is provided in Section 3.14
Display Filters...		This menu item brings up a dialog box that allows you to create and edit display filters. You can name filters, and you can save them for future use. More detail on this subject is provided in Section 3.14
Protocols...		This menu item brings up a dialog box that allows you to enable or disable the dissection of individual protocols edit.

Table 3-2. Edit menu

3.3.3. The Ethereal Capture menu

The Ethereal Capture menu contains the fields shown in Table 3-3.

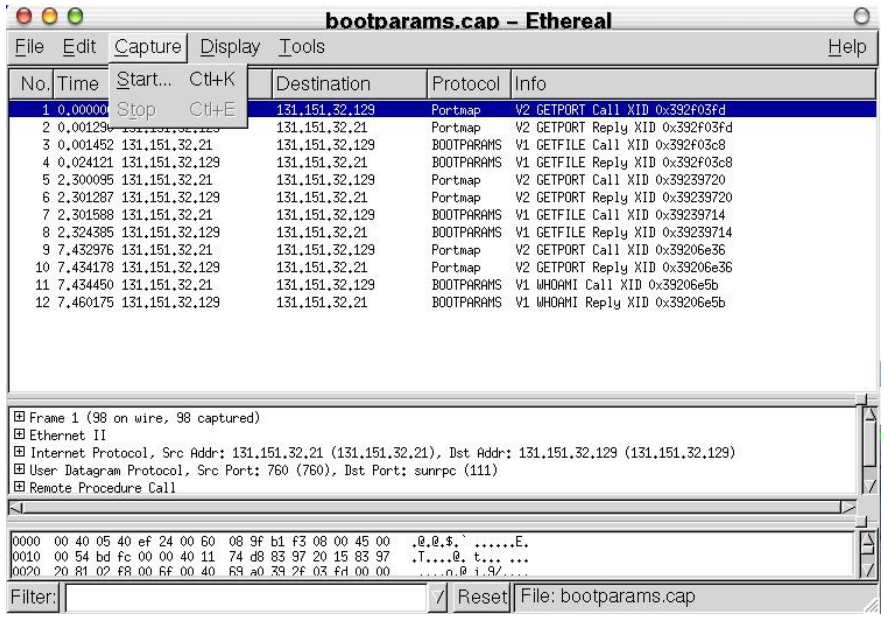

Figure 3-5. Ethereal Capture Menu

Menu Item	Accelerator	Description
Start...	Ctrl-K	This menu item brings up the Capture Preferences dialog box (discussed further in Section 3.4) and allows you to start capturing packets.
Stop	Ctrl-E	This menu item stops the currently running capture.

Table 3-3. Capture menu

3.3.4. The Ethereal Display menu

The Ethereal Display menu contains the fields shown in Table 3-4.

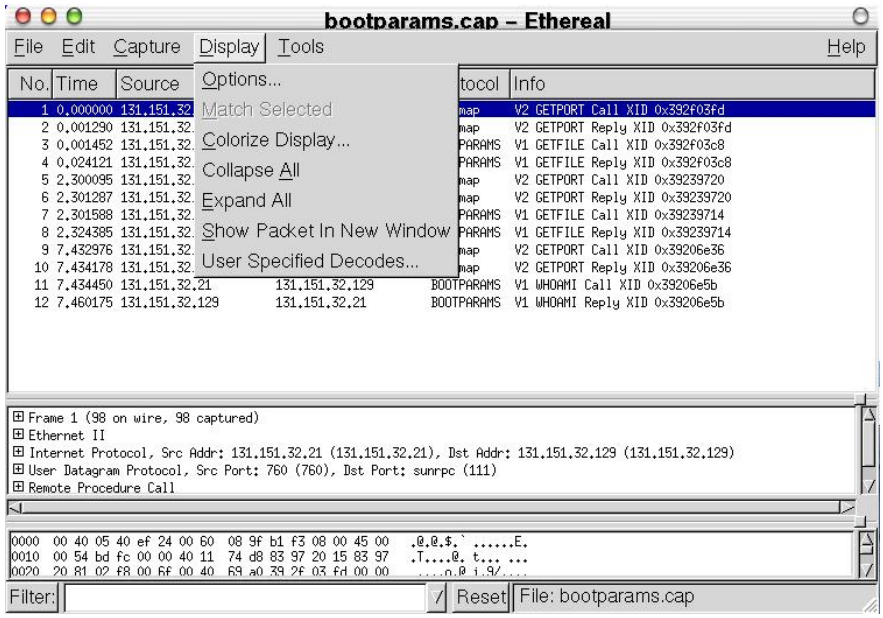

Figure 3-6. Ethereal Display Menu

Menu Item	Accelerator	Description
Options...		This menu item brings up a dialog box that controls the way that Ethereal displays some information about packets. Examples include the way timestamps are handled, whether addresses and other numbers are translated, and so forth. This is further discussed in Section 3.7.
Match Selected		This menu item allows you to select all packets that have a matching value in the field selected in the tree view pane (middle pane).
Colorize Display		This menu item brings up a dialog box that allows you color packets in the packet list pane according to filter expressions you choose. It can be very useful for spotting certain types of packets.
Collapse All		Ethereal keeps a list of all the protocol subtrees that are expanded, and uses it to ensure that the correct subtrees are expanded when you display a packet. This menu item collapses the tree view of all packets in the capture list.
Expand All		This menu item expands all subtrees in all packets in the capture.
Show Packet in New Window		This menu item brings up the selected packet in a separate window. The separate window shows only the tree view and byte view panes.

Menu Item	Accelerator	Description
User Specified Decodes...		This menu item allows the user to force ethereal to decode certain packets as a particular protocol.

Table 3-4. Display menu

3.3.5. The Ethereal Tools menu

The Ethereal Tools menu contains the fields shown in Table 3-5.

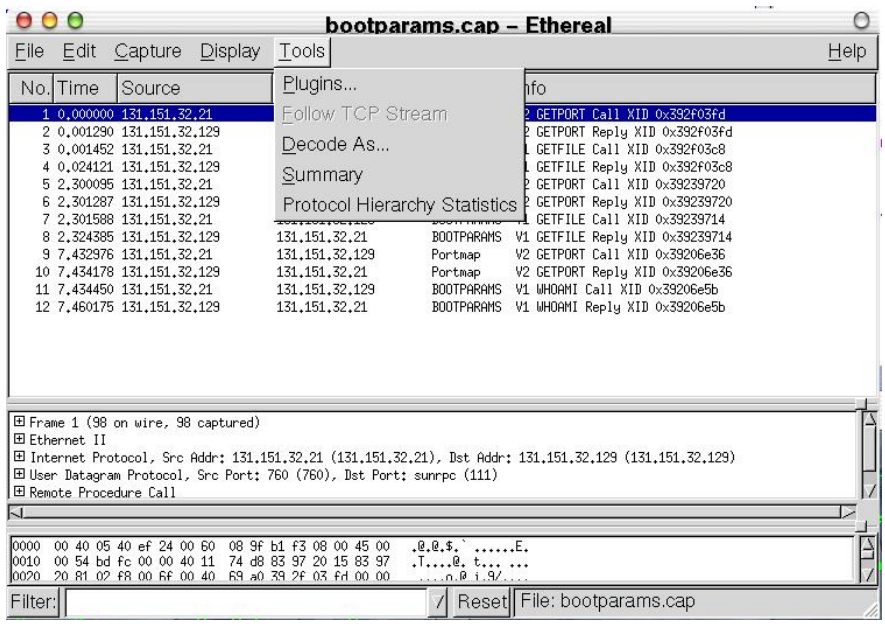

Figure 3-7. Ethereal Tools Menu

Menu Item	Accelerator	Description
Plugins...		This menu item brings up a dialog box that allows you to manage Ethereal plugins. There are very few plugins todate.
Follow TCP Stream		This menu item brings up a separate window and displays all the TCP segments captured that are on the same TCP connection as a selected packet. The data in the TCP stream is sorted into order, with duplicate segments removed, and it is then displayed in ascii. You can change the format is you desire.

Menu Item	Accelerator	Description
Decode As...		This menu item allows the user to force ethereal to decode certain packets as a particular protocol.
Summary		This menu item brings up a statistics window that shows information about the packets captured.
Protocol Hierarchy Statistics		This menu item displays a hierarchical tree of packet statistics.

Table 3-5. Tools menu

3.3.6. The Ethereal Help menu

The Ethereal Help menu contains the fields shown in Table 3-6.

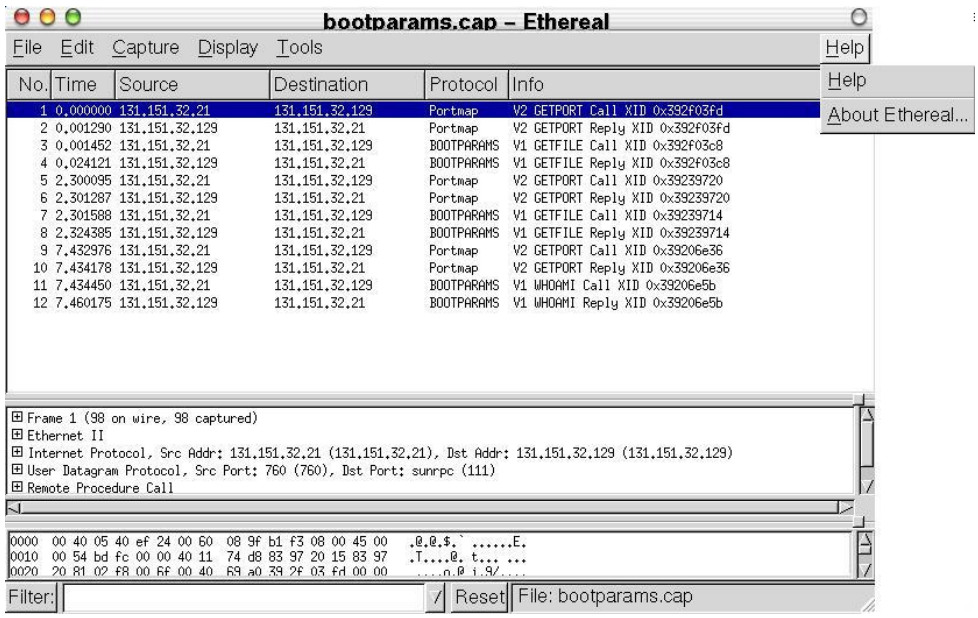

Figure 3-8. Ethereal Help Menu

Menu Item	Accelerator	Description
Help		This menu item brings up a basic help system.
About Ethereal...		This menu item brings up an information window that provides some simple information on Ethereal.

Table 3-6. Help menu

3.4. Capturing packets with Ethereal

There are two methods you can use to capture packets with Ethereal:

1. From the command line using the following:

   ```
   ethereal -i eth0 -k
   ```

2. By starting Ethereal and then selecting Start... from the Capture menu. This brings up the Capture Preferences dialog box and will be dealt with in more detail in Section 3.4.1.

3.4.1. The Capture Preferences dialog box

When you select Start... from the Capture menu, Ethereal pops up the Capture Preferences dialog box as shown in Figure 3-9.

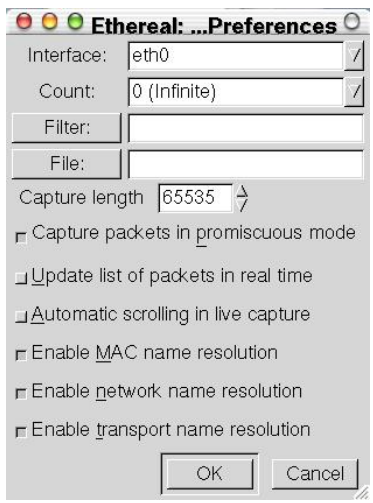

Figure 3-9. The Capture Preferences dialog box

You can set the following fields in this dialog box:

Interface

This field specifies the interface you want to capture on. You can only capture on one interface, and you can only capture on interfaces that the Ethereal has found on the system. It is a drop-down list, so simply click on the button on the right hand side and select the interface you want. It defaults to the first non-loopback interface that supports capturing, and if there are none, the first

loopback interface. On some systems, loopback interfaces cannot be used for capturing.

This field performs the same function as the **-i <interface>** command line option.

Count

This field specifies the number of packets that you want to capture. It defaults to 0, which means do not stop capturing. Enter the value that you want in here, or leave it blank.

Filter

This field allows you to specify a capture filter. Capture filters are discussed in more details in Section 3.5. It defaults to empty, or no filter.

You can also click on the Filter button/label, and Ethereal will bring up the Filters dialog box and allow you to create and/or select a filter. Please see Section 3.14

File

This field allows you to specify the file name that will be used for the capture when you later choose Save... or Save As... from the Ethereal File menu. There is no default for this value.

Capture length

This field allows you to specify the maximum amount of data that will be captured for each packet, and is sometimes referred to as the **snaplen**. The default is 65535, which will be sufficient for most protocols. It should be at least the MTU for the interface you are capturing on.

Capture packets in promiscuous mode

This radio button allows you to specify that Ethereal should set the interface in promiscuous mode when capturing. If you do not specify this, Ethereal will only capture the packets going to or from your computer (not all packets going by your interface).

If some other process has put the interface in promiscuous mode you may be capturing in promiscous mode even if you turn off this option

Update list of packets in real time

This radio button allows you to specify that Ethereal should update the packet list pane in real time. If you do not specify this, Ethereal does not display any packets until you cancel the capture. When you click on this radio button, Ethereal captures in a separate process and feeds the captures to the display process. [Is this true for Windows?]

Automatic scrolling in live capture

This radio button allows you to specify that Ethereal should scroll the packet list pane as new packets come in, so you are always looking at the last packet. If you do not specify this, Ethereal simply adds new packets onto the end of the list, but does not scroll the packet list pane.

Enable MAC name resolution

This radio button allows you to control whether or not Ethereal translates the first three octets of a MAC addresses into the name of the manufacturer to whom that prefix has been assigned by the IETF.

Enable network name resolution

This radio button allows you to control whether or not Ethereal translates IP addresses into DNS domain names. By clicking on this radio button, the packet list pane will have more useful information, but you will also cause name lookup requests to occur, which might disturb the capture.

If you cannot reach the name server, you may find that Ethereal takes a long time in updating the packet list pane as it waits for name translation to time out.

Enable transport name resolution

This radio button allows you to control whether or not Ethereal translates port numbers into protocols.

Once you have set the values you desire and have selected the radio buttons you need, simply click on OK to commence the capture, or Cancel to cancel the capture.

If you start a capture, Ethereal pops up a dialog box that shows you the progress of the capture and allows you to stop capturing when you have enough packets captured.

3.5. Filtering while capturing

Ethereal uses the libpcap filter language for capture filters. This is explained in the tcpdump man page. If you can understand it, you are a better man that I am, Gunga Din!

You enter the capture filter into the Filter field of the Ethereal Capture Preferences dialog box, as shown in Figure 3-9. The following is an outline of the syntax of the **tcpdump** capture filter language.

A capture filter takes the form of a series of primitive expressions connected by conjuctions (**and/or**) and optionally preceeded by **not**:

```
[not] primitive [and|or [not] primitive ...]
```

An example is shown in Example 3-2.

```
tcp port 23 and host 10.0.0.5
```

Example 3-2. A capture filter for telnet than captures traffic to and from a particular host

This example captures telnet traffic to and from the host 10.0.0.5, and shows how to use two primitives and the **and** conjunction. Another example is shown in Example 3-3, and shows how to capture all telnet traffic except that from 10.0.0.5.

```
tcp port 23 and not host 10.0.0.5
```

Example 3-3. Capturing all telnet traffic not from 10.0.0.5

A primitive is simply one of the following:

[src|dst] host <host>

This primitive allows you to filter on a host IP address or name. You can optionally preceed the primitive with the keyword **src|dst** to specify that you are only interested in source or destination addresses. If these are not present, packets where the specified address appears as either the source or the destination address will be selected.

ether [src|dst] host <ehost>

This primitive allows you to filter on Ethernet host addresses. You can optionally includethe keyword **src|dst** between the keywords **ether** and **host** to specify that you are only interested in source or destination addresses. If these are not present, packets where the specified address appears in either the source or destination address will be selected.

gateway host <host>

This primitive allows you to filter on packets that used **host** as a gateway. That is, where the Ethernet source or destination was **host** but neither the source nor destination IP address was **host**.

[src|dst] net <net> [{mask <mask>}|{len <len>}]

This primitive allows you to filter on network numbers. You can optionally preceed this primitive with the keyword **src|dst** to specify that you are only interested in a source or destination network. If neither of these are present, packets will be selected that have the specified network in either the source or destination address. In addition, you can specify either the netmask or the CIDR prefix for the network if they are different from your own.

[tcp|udp] [src|dst] port <port>

This primitive allows you to filter on TCP and UDP port numbers. You can optionally preceed this primitive with the keywords **src|dst** and **tcp|udp** which allow you to specify that you are only interested in source or destination ports and TCP or UDP packets respectively. The keywords **tcp|udp** must appear before **src|dst**.

If these are not specified, packets will be selected for both the TCP and UDP protocols and when the specified address appears in either the source or destination port field.

less|greater <length>

This primitive allows you to filter on packets whose length was less than or equal to the specified length, or greater than or equal to the specified length, respectively.

ip|ether proto <protocol>

This primitive allows you to filter on the specified protocol at either the Ethernet layer or the IP layer.

ether|ip broadcast|multicast

This primitive allows you to filter on either Ethernet or IP broadcasts or multicasts.

<expr> relop <expr>

This primitive allows you to create complex filter expressions that select bytes or ranges of bytes in packets. Please see the tcpdump man pages for more details.

3.6. Viewing packets you have captured

Once you have captured some packets, or you have opened a previously saved capture file, you can view the packets that are displayed in the packet list pane by simply clicking on that packet in the packet list pane, which will bring up the selected packet in the tree view and byte view panes.

You can then expand any part of the tree view by clicking on the **plus** sign to the left of that part of the payload, and you can select individual fields by clicking on them in the tree view pane. An example with a TCP segment selected is shown in Figure 3-10. It also has the Acknowledgment number in the TCP header selected, which shows up in the byte view as the selected bytes.

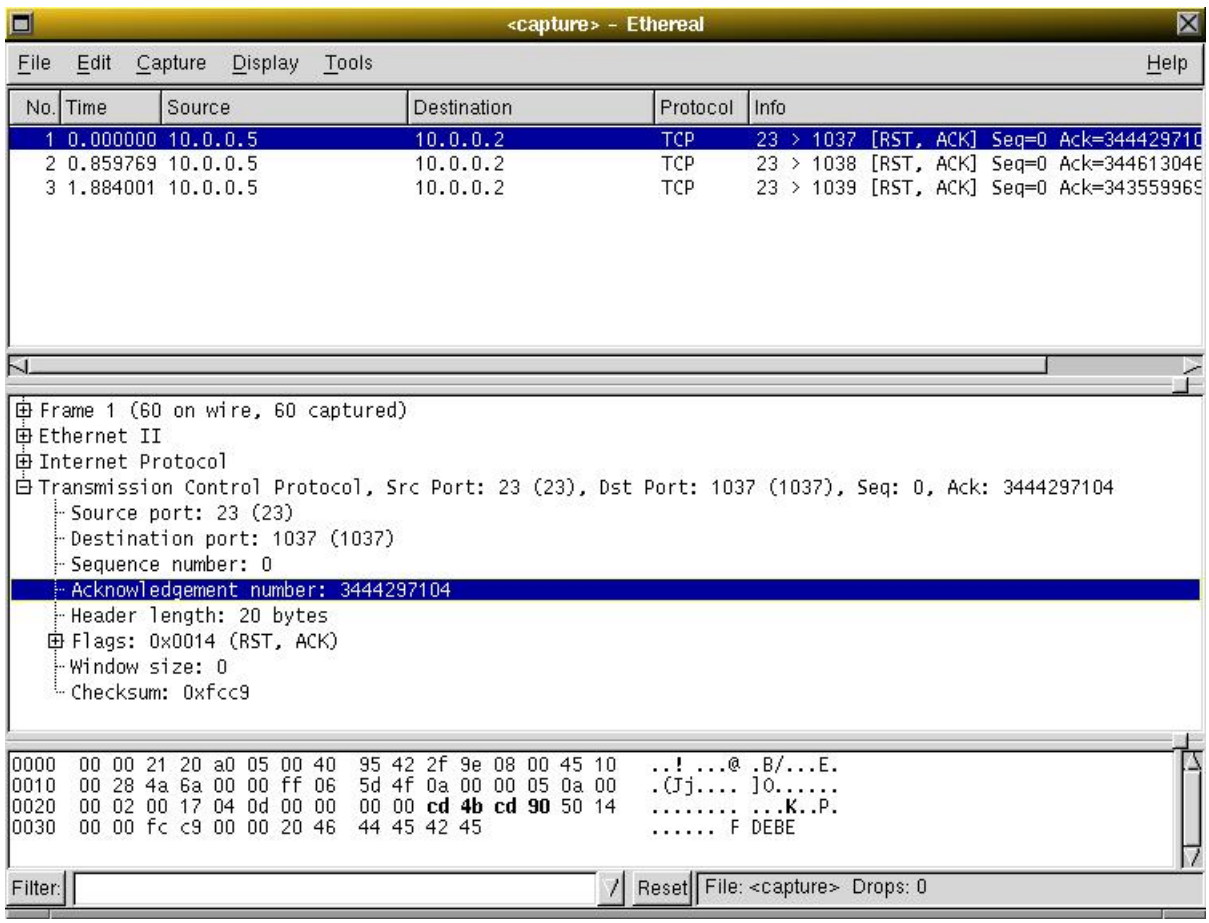

Figure 3-10. Ethereal with a TCP segment selected for viewing

You can also select and view packets when Ethereal is capturing if you selected "Update list of packets in real time" in the Ethereal Capture Preferences dialog box.

In addition, you can view individual packets in a separate window as shown in Figure 3-11. Do this by selecting the packet you are interested in in the packet list pane, and then select "Show Packet in New Windows" from the Display menu. This allows you to easily compare two or more packets.

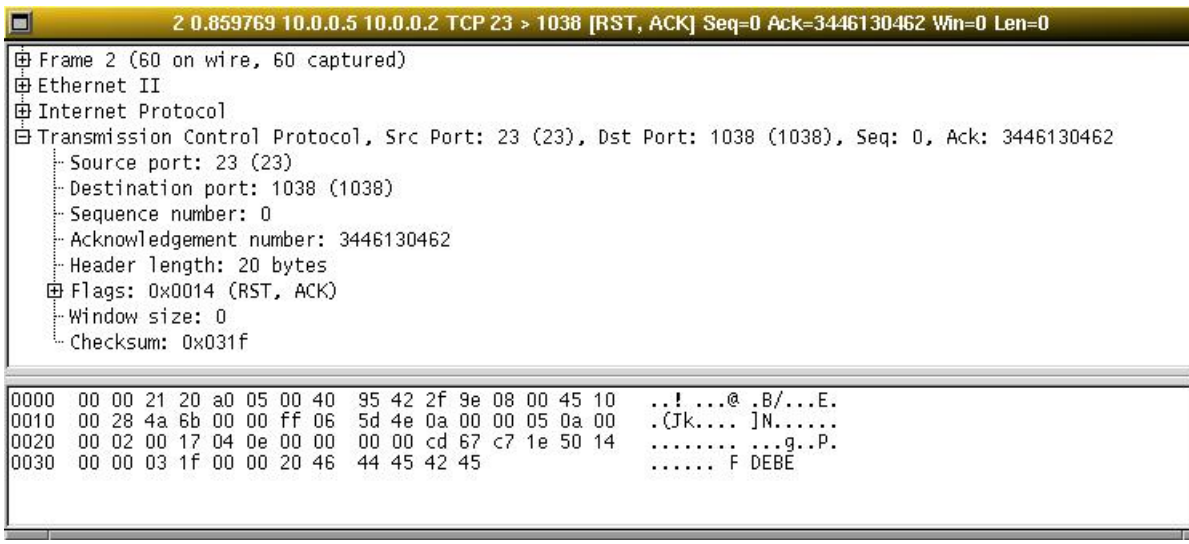

Figure 3-11. Viewing a packet in a separate window

Finally, you can bring up a pop-up menu over either the packet list pane or the tree view pane by clicking your right mouse button. The menus that is popped up contains the following items:

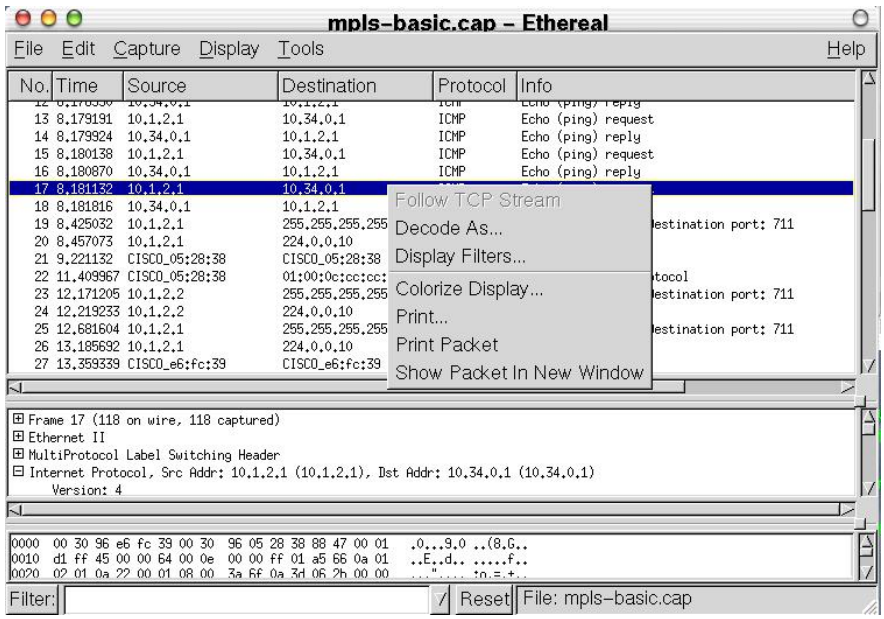

Figure 3-12. Packet Pane pop-up menu

Follow TCP Stream

This menu item is the same as the Display menu item of the same name. It allows you to view all the data on a TCP stream between a pair of nodes.

Decode As...

This menu item is the same as the Display menu item of the same name.

Display Filters...

This menu item is the same as the Edit menu item of the same name. It allows you to specify and manage filters.

Colorize Display...

This menu item is the same as the Display menu item of the same name. It allows you to colorize packets in the packet list pane.

Print...

This menu item is the same as the File menu item of the same name. It allows you to print packets.

Print Packet

This menu item is the same as the File menu item of the same name. It allows you to print the currently selected packet.

Show Packet in New Window

This menu item is the same as the Display menu item of the same name. It allows you to display the selected packet in another window.

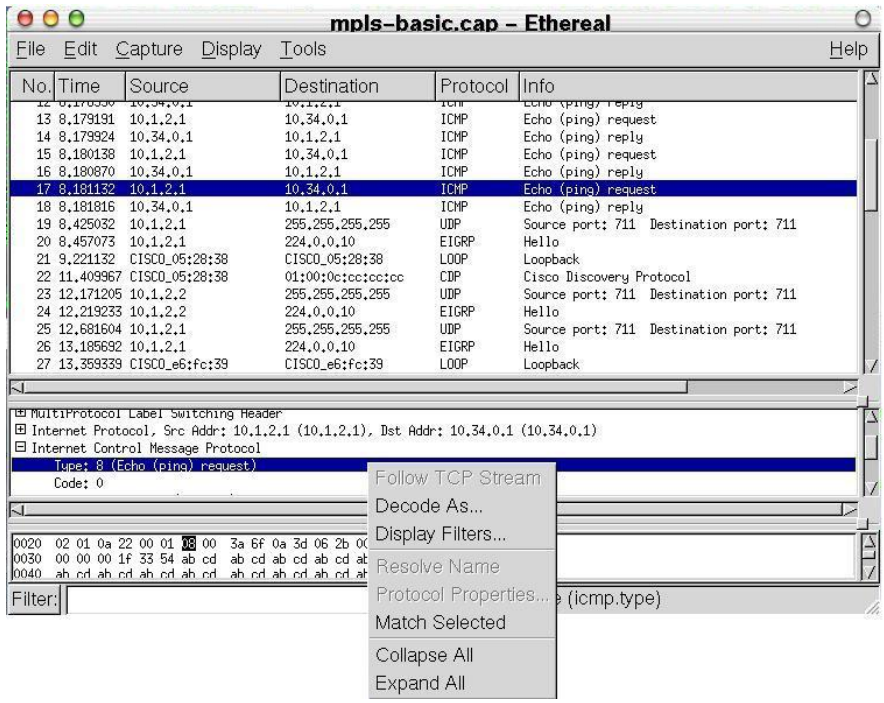

Figure 3-13. Treeview Pane pop-up menu

Follow TCP Stream

This menu item is the same as the Display menu item of the same name. It allows you to view all the data on a TCP stream between a pair of nodes.

Decode As...

This menu item is the same as the Display menu item of the same name.

Display Filters...

This menu item is the same as the Edit menu item of the same name. It allows you to specify and manage filters.

Resolve Name

This menu item causes name resolution to be performed for the selected packet, but NOT every packet in the capture.

Protocol Properties...

The menu item takes you to the protocol properties dialog if there are properties associated with the highlighted fields. More information on preferences can be found in Figure 3-29.

Match Selected

This menu item allows you to select all packets that have a matching value in the field selected in the tree view pane (middle pane).

Collapse All

Ethereal keeps a list of all the protocol subtrees that are expanded, and uses it to ensure that the correct subtrees are expanded when you display a packet. This menu item collapses the tree view of all packets in the capture list.

Expand All

This menu item expands all subtrees in all packets in the capture.

3.7. Display Options

You can control the way that Ethereal displays a number of items. You manage these by selecting the **Options** menu item from the **Display** menu. When you do this, Ethereal pops up the **Display Options** dialog box, as shown in Figure 3-14.

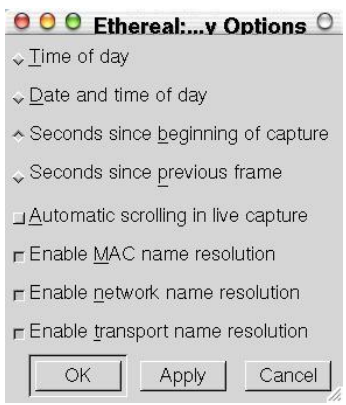

Figure 3-14. Ethereal Display Options dialog box

The following are the items on this dialog box and their meanings:

Time of day

Selecting this radio button tells Ethereal to display time stamps in time of day format. This field, Date and time of day, Seconds since beginning of capture and Seconds since previous frame are mutually exclusive.

Date and time of day

Selecting this radia button tells Ethereal to display the time stamps in date and time of day format. Time of day, this field, Seconds since beginning of capture and Seconds since previous frame are mutually exclusive.

Seconds since beginning of capture

Selecting this radio button tells Ethereal to display time stamps in seconds since beginning of capture format. Time of day, Date and time of day, this field, and Seconds since previous frame are mutually exclusive.

Seconds since previous frame

This radio button tells Ethereal to display time stamps in seconds since previous frame format. Time of day, Date and time of day, Seconds since beginning of capture and this field are mutually exclusive.

Automatic scrolling in live capture

This field, when selected, tells Ethereal to scroll the packet list pane when new packets are captured.

Enable MAC name resolution

This field, when selected, tells Ethereal to translate the first three octets of MAC addresses (the vendor identifier) into names (where it can) when displaying packets.

Enable network name resolution

This field, when selected, tells Ethereal to translate ip addresses into domain names (where it can) when displaying packets.

If you select this option and your DNS server is unavailable then ethereal will be very slow as it times out waiting for responses from your DNS server.

Enable transport name resolution

This field, when selected, tells Ethereal to translate the transport layer addresses (TCP/UDP port numbers) into well known service names (where it can) when displaying packets.

3.8. Saving captured packets

You can save captured packets simply by using the Save As... menu item from the File menu under Ethereal. You can choose to save all packets that were captured or only the packets currently being displayed.

3.8.1. The Save Capture File As dialog box

The Ethereal Save Capture File As dialog box allows you to save the current capture to a file. Figure 3-15 shows an example of this dialog box.

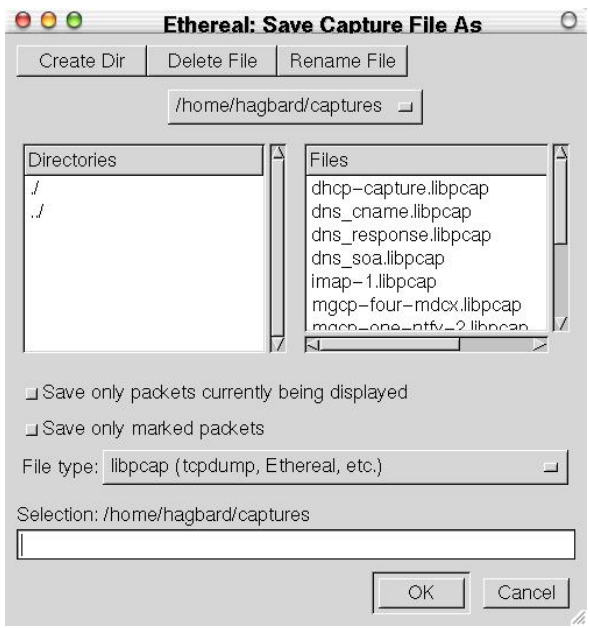

Figure 3-15. The Ethereal Save Capture File As dialog box

With this dialog box, you can perform the following actions:

1. Create directories with the **Create Dir** button.
2. Delete files with the **Delete File** button.
3. Rename files with the **Rename File** button.
4. Select files and directories with the directories and files list boxes and the file system heirarchy drop down box.

5. Save only the packets currently being displayed (as apposed to all the packets captured) by clicking on the "Save only packets currently being displayed" radio button.
6. Save only the marked packets (as apposed to all the packets captured) by clicking on the "Save only marked packets" radio button. More on Marking packets can be found in Section 3.3.2.
7. Specify the format of the saved capture file by clicking on the File type drop down box. You can choose from among the following types:
   a. libpcap (tcpdump, Ethereal, etc.)
   b. modified libpcap (tcpdump)
   c. RedHat Linux libpcap (tcpdump)
   d. Network Associates Sniffer (DOS based)
   e. Sun Snoop
   f. Microsoft Network Monitor 1.x
   g. Network Associates Sniffer (Windows based) 1.1

Some capture formats may not be available, depending on the frame types captured.

You can convert capture files from one format to another by reading in a capture file and writing it out using a different format.

8. Type in the name of the file you wish to save the captured packets in, as a standard file name in your file system.
9. Click on OK to accept your selected file and save to it. If Ethereal has a problem saving the captured packets to the file you specified, it will display an error dialog box. After clicking OK, you can try another file.
10. Click on Cancel to go back to Ethereal and not save the captured packets.

3.9. Reading capture files

Ethereal can read in previously saved capture files, and in addition, becuase it is built with a subroutine library called **libwiretap**, it can read capture files from a number of other packet capture programs as well. The following is the list of

capture formats it understands:

- tcpdump and Ethereal
- snoop (including Shomiti) and atmsnoop
- LanAlyzer
- Sniffer (compressed or uncompressed)
- Microsoft Network Monitor
- AIX's iptrace
- NetXray
- Sniffer Pro
- RADCOM's WAN/LAN analyzer
- Lucent/Ascend router debug output
- HP-UX's nettl
- the dump output from Toshiba's ISDN routers
- i4btrace from the ISDN4BSD project

You only need to get these files onto your system and Ethereal can read them. To read them, simply select the **Open** menu item from the **File** menu. Ethereal will then pop up the File Open dialog box, which is discussed in more detail in Section 3.9.1

3.9.1. The File Open dialog box

The Ethereal File Open dialog box allows you to search for a capture file containing previously captured packets for display in Ethereal. Figure 3-16 shows an example of the Ethereal Open File Dialog box.

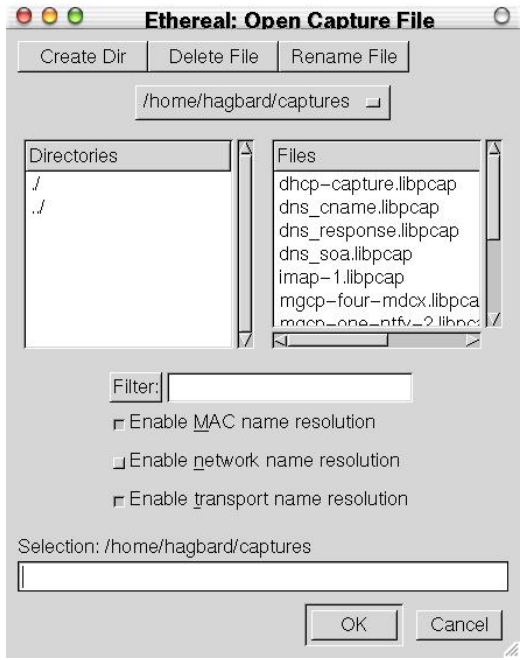

Figure 3-16. The Ethereal Open File Dialog box

With this dialog box, you can perform the following actions:

1. Create directories with the **Create Dir** button.
2. Delete files with the **Delete File** button.
3. Rename files with the **Rename File** button.
4. Select files and directories with the directories and files list boxes and the file system heirarchy drop down box.
5. Specify a display filter with the Filter button and filter field. Clicking on the Filter button causes Ethereal to pop up the Filters dialog box (while is discussed further in Section 3.10).
6. Specify that MAC name resolution is to be performed for all MAC addresses in packets by clicking on the "Enable MAC name resolution" check button.
7. Specify that DNS name resolution is to be performed for all ip addresses in packets by clicking on the "Enable network name resolution" check button.

   Enabling network name resolution when your DNS server is unavailable may significantly slow ethereal while it waits for all of the DNS requests to time out

8. Specify that transport name resolution is to be performed for all transport (TCP/UDP port) addresses in packets by clicking on the "Enable transport name resolution" check button.

9. Type in the name of the capture file you wish to open, as a standard file name in your file system.
10. Click on OK to accept your selected file and open it. If Ethereal recognizes the capture format, it will display the packets read from the capture file in the packet list pane. If it does not recognize the capture format, it will display an error dialog box. After clicking OK, you can try another file.
11. Click on Cancel to go back to Ethereal and not load a capture file.

3.10. Filtering packets while viewing

Ethereal has two filtering languages: One used when capturing packets, and one used when displaying packets. In this section we explore that second type of filters: Display filters. The first one has already been dealt with in Section 3.5.

Display filters allow you to concentrate on the packets you are interested in. They allow you to select packets by:

- Protocol
- The presence of a field
- The values of fields
- A comparison between fields

To select packets based on protocol type, simply type the protocol you are interested in in the **Filter:** field on the bottom left hand corner of the Ethereal window and press enter to initiate the filter. Figure 3-17 shown an example of what happens when you type **smb** in the filter field.

All filter expressions are entered in lowercase. Also, don’t forget to press enter after entering the filter expression.

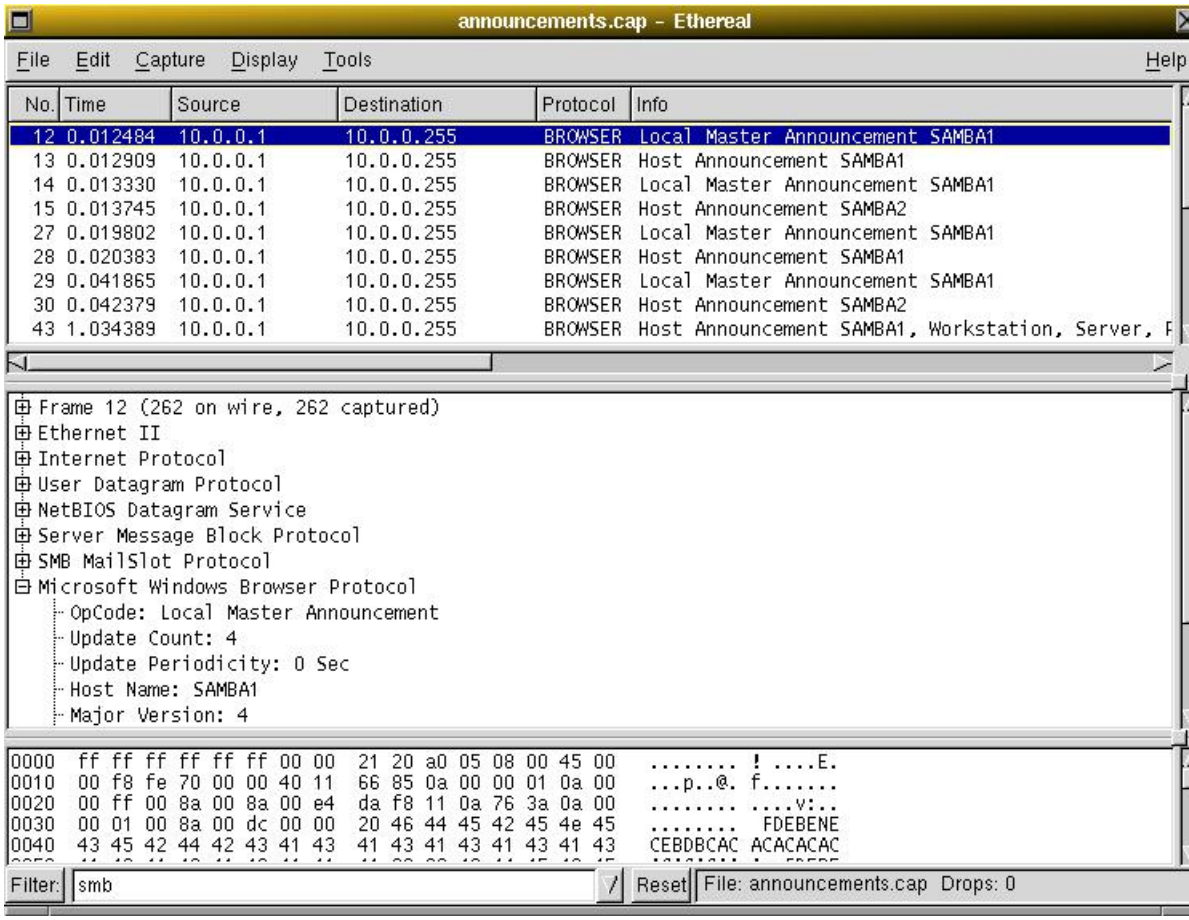

Figure 3-17. Filtering on the SMB protocol

The packets selected in Figure 3-17 all show up as **BROWSER** packets but they are carried in SMB packets.

You can filter on any protocol that Ethereal understands. However, you can also filter on any field that a dissector adds to the tree view, but only if the dissector has added an abbreviation for the field. A list of such fields is available in the Ethereal in the **Add Expression...** dialog box. You can find more information on the **Add Expression...** dialog box in Section 3.15. You may also find a list of the fields in Appendix A

For example, to narrow the packet list pane down to only those packets to or from 10.0.0.5, use **ip.addr==10.0.0.5.**

To remove the filter, click on the **Reset** button to the right of the filter field.

3.10.1. Building filter expressions

Ethereal provides a simple display filter language that you can build quite complex

filter expressions with. You can compare values in packets as well as combine expressions into more specific expressions. The following sections provide more information on doing this.

3.10.1.1. Comparing values

You can build display filters that compare values using a number of different comparison operators. They are shown in Table 3-7.

English	C-like	Description and example
eq	`==`	**Equal** `ip.addr==10.0.0.5`
ne	`!=`	**Not equal** `ip.addr!=10.0.0.5`
gt	`>`	**Greater than** `frame.pkt_len > 10`
lt	`<`	**Less than** `frame.pkt_len < 128`
ge	`>=`	**Greater than or equal to** `frame.pkt_len ge 0x100`
le	`<=`	**Less than or equal to** `frame.pkt_len <= 0x20`

Table 3-7. Display filter comparison operators

In addition, all protocol fields are typed. Table 3-8 provides a list of the types and example of how to express them.

Type	Example

Type	Example

Unsigned integer (8-bit, 16-bit, 24-bit, 32-bit)	You can express integers in decimal, octal, or hexadecimal. The following display filters are equivalent:
	`ip.len le 1500`
	`ip.len le 02734`
	`ip.len le 0x436`

Signed integer (8-bit, 16-bit, 24-bit, 32-bit)	
Boolean	A boolean field is present in the protocol decode only if its value is true. For example, **tcp.flags.syn** is present, and thus true, only if the SYN flag is present in a TCP segment header. Thus the filter expression **tcp.flags.syn** will select only those packets for which this flag exists, that is, TCP segments where the segment header contains the SYN flag. Similarly, to find source-routed token ring packets, use a filter expression of **tr.sr**.
Ethernet address (6 bytes)	
IPv4 address	
IPv6 address	
IPX network number	
String (text)	
Double-precision floating point number	

Table 3-8. Field Types

3.10.1.2. Combining expressions

You can combine filter expressions in Ethereal using the logical operators shown in Table 3-9

English	C-like	Description and example

English	C-like	Description and example
and	&&	**Logical AND** `ip.addr==10.0.0.5 and tcp.flags.f`
or	||	**Logical OR** `ip.addr==10.0.0.5 or ip.addr==192`
xor	^^	**Logical XOR** `tr.dst[0:3] == 0.6.29 xor tr.src[`
not	!	**Logical NOT** `not llc`

English	C-like	Description and example

[...]		**Substring Operator**
		Ethereal will allow you to select subsequences of a sequence in rather elaborate ways. After a label you can place a pair of brackes [] containing a comma separated list of range specifiers.
		
		`eth.src[0:3] == 00:00:83`
		
		The example above uses the n:m format to specify a single range. In this case n is the beginning offset and m is the length of the range being specified.
		
		`eth.src[1-2] == 00:83`
		
		The example above uses the n-m format to specify a single range. In this case n is the beginning offset and m is the ending offset.
		
		`eth.src[:4] == 00:00:83:00`
		
		The example above uses the :m format, which takes everything from the beginning of a sequence to offset m. It is equivalent to 0:m
		
		`eth.src[4:] == 20:20`

English	C-like	Description and example

Table 3-9. Display Filter Logical Operations

3.11. Packet colorization

A very useful mechanism available in Ethereal is packet colorization. You can set Ethereal up so that it colorizes packets according to a filter. This allows you to emphasize the packets you are interested in.

To colorize packets, select the Colorize Display... menu item from the Display menu, and Ethereal will pop up the Add Color to Protocols dialog box as shown in Figure 3-18.

Figure 3-18. The Ethereal Add Color to Protocols dialog box

Once the Add Color to Protocol dialog box is up, there are a number of buttons you can use, depending on whether or not you have any color filters installed already. If this is the first time you have used Add Color to Protocol, click on New which will bring up the Edit color filter dialog box as shown in Figure 3-19.

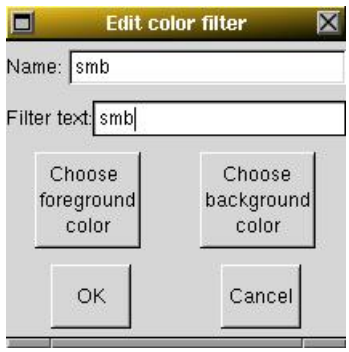

Figure 3-19. The Ethereal Edit color filter dialog box

In the Edit Color dialog box, simply enter a name for the color filter, and enter a filter sting in the Filter text field. Figure 3-19 shows the values **smb** and **smb** which means that the name of the color filter is **smb** and the filter will select protocols of type **smb**. Once you have entered these values, you can choose a background and foreground color for packets that match the filter expression. Click on **Choose background color** or **Choose foreground color** to do achieve this and Ethereal will pop up the Choose foreground/background color for protocol dialog box as shown in Figure 3-20.

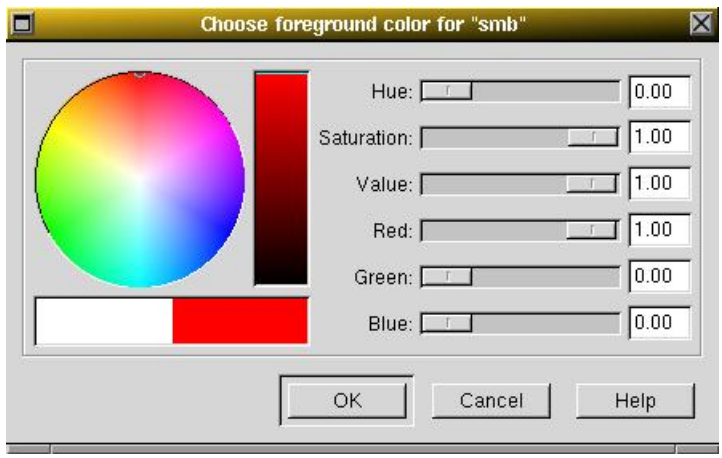

Figure 3-20. Ethereal Choose color dialog box

Select the color you desire for the selected packets and click on OK.

You must select a color in the colorbar next to the colorwheel to load values into the RGB sliders. Alternatively, you can use the sliders to select the color you want.

You will need to carefully select the order that filters are listed (and thus applied) as they are applied in order. So, more specific filters need to be listed before more general filters. For example, if you have a color filter for UDP before the one for DNS, the color filter for DNS will never be applied.

Figure 3-21 shows an example of several color filters being used in Ethereal. You may not like the color choices, however, so feel free to choose your own.

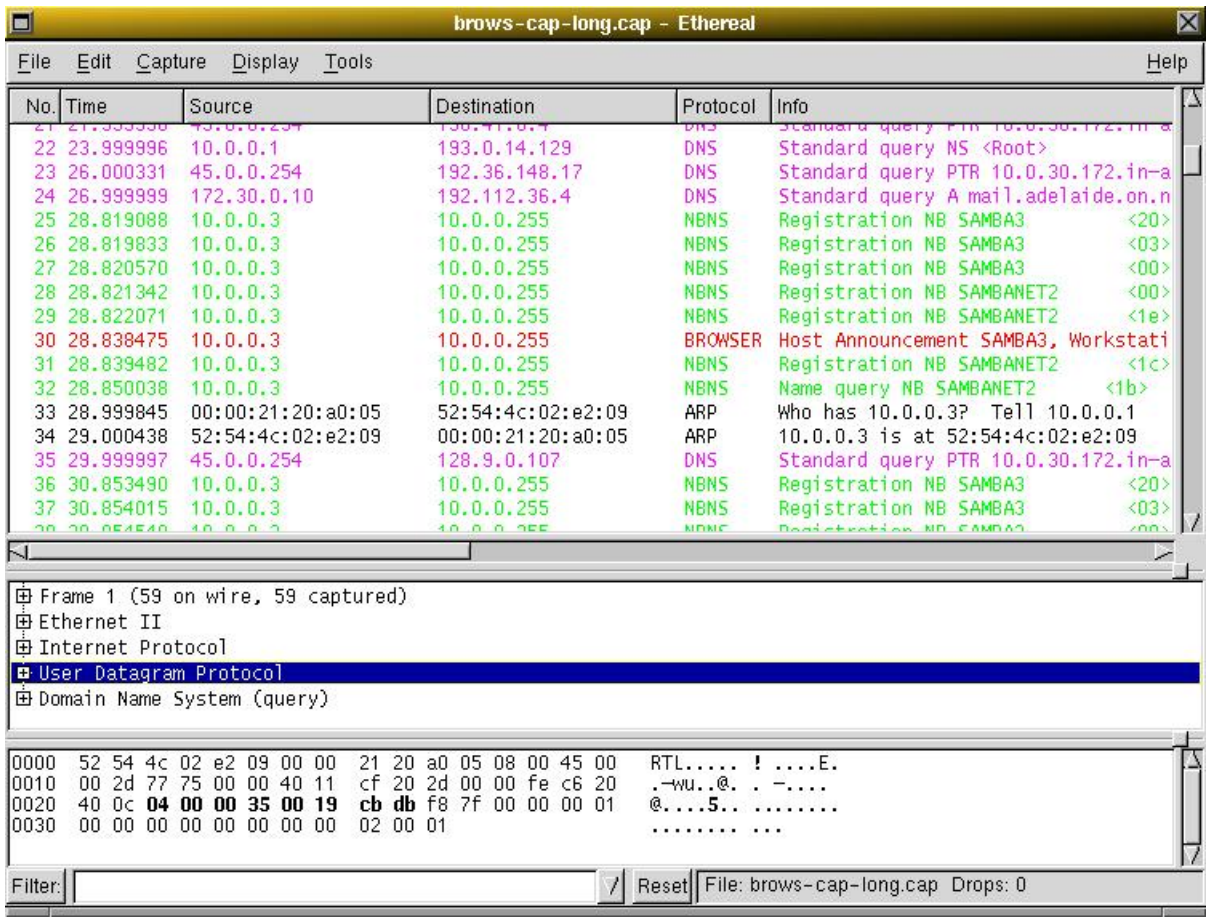

Figure 3-21. Using color filters with Ethereal

3.12. Finding frames

You can easily find frames once you have captured some packets or have read in a previously saved capture file. Simply select the **Find Frame...** menu item from the **Edit** menu. Ethereal will pop up the dialog box shown in Figure 3-22.

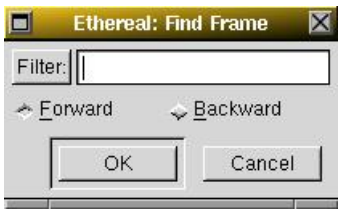

Figure 3-22. The Ethereal Find Frame dialog box

Simply enter a display filter string into the **Filter:** field, select a direction, and click on OK.

For example, to find the three way handshake for a connection from host 10.0.0.5, use the following filter string:

```
ip.addr==10.0.0.5 and tcp.flags.syn
```

For more details on display filters, see Section 3.10

3.13. Following TCP streams

There will be occasions when you would like to see the data on a TCP session in the order that the application layer would see it. Perhaps you are looking for passwords in a Telnet stream, or perhaps you are trying to make sense of a data stream. If so, Ethereal's ability to follow a TCP stream will be useful to you.

Simply select a TCP segment on the stream/connection you are interested in and then select the Follow TCP Stream menu item from the Ethereal Tools menu. Ethereal will pop up a separate window with all the data from the TCP stream layed out in order, as shown in Figure 3-23.

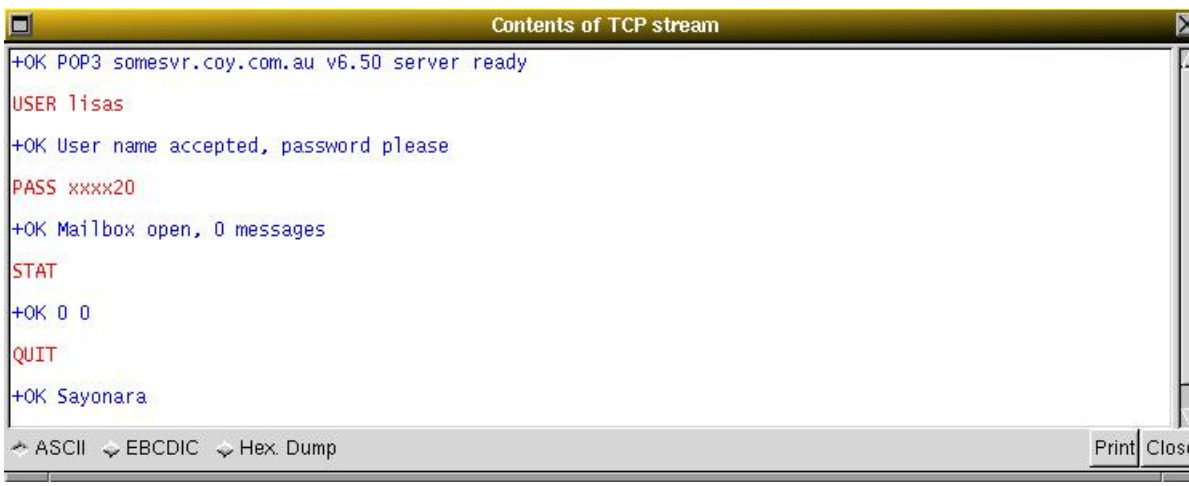

Figure 3-23. Following a TCP Stream

You can then select to view the data in one of three formats:

1. **ASCII**. In this view you see the data from each end in ASCII, but alternating according to when each end sent data. Unfortunately, non-printing characters do not print.
2. **EBCDIC**. For the big-iron freaks out there.
3. **HEX Dump**. This allows you to see all the data, but you lose the ability to read it in ASCII.

Note!
It is worthwhile noting that Follow TCP Stream installs a filter to select all the packets on the TCP stream you have selected.

3.14. Defining and saving filters

You can define filters with Ethereal and give them labels for later use. This can save time in remembering and retyping some of the more complex filters you use.

To define a new filter or edit an existing filter, select the Filters... menu item from the Edit menu. Ethereal will then pop up the Filters dialog as shown in Figure 3-24.

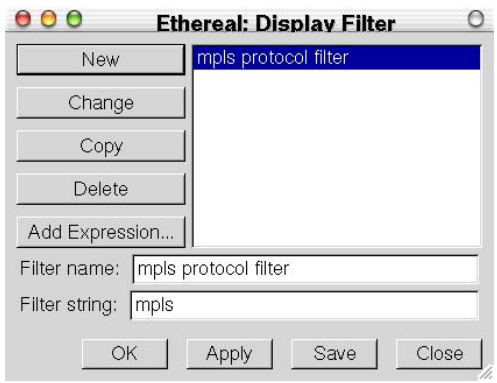

Figure 3-24. The Ethereal Filters dialog box

You would enter a filter name in the Filter name field, and a filter string in the Filter string field. However, for most other actions, you would select a filter from the list box (which will fill in the name and sting in the fields down the bottom of the dialog box), and make what ever changes you want to. Then you should choose one of the buttons down the left hand side of the dialog box. The buttons have the following meanings:

New

This button adds the filter string entered in the Filter string field with the name supplied in the Filter name field.

You can add multiple filters with the same name. This is not very useful.

Change

This button changes the filter named in the Filter name string by replacing its filter string with the string in the Filter string field.

Copy

This button copies the selected filter and calls it "Copy of <orig>", where <orig> is the name of the original filter.

Delete

This button deletes the selected filter.

Apply

This button applies the selected filter to the current display.

Add Expression...

This button brings up the Add Expression dialog box which assists in building filter strings. You can find more information about the Add Expression dialog in Section 3.15

3.15. The Add Expression Dialog

When you are accustomed to Ethereal's filtering system and know what labels you wish to use in your filters it can be very quick to simply type a filter string. However if you are new to Ethereal or are working with a slightly unfamiliar protocol it can be very confusing to try to figure out what to type. The Add Expression dialog box helps with this.

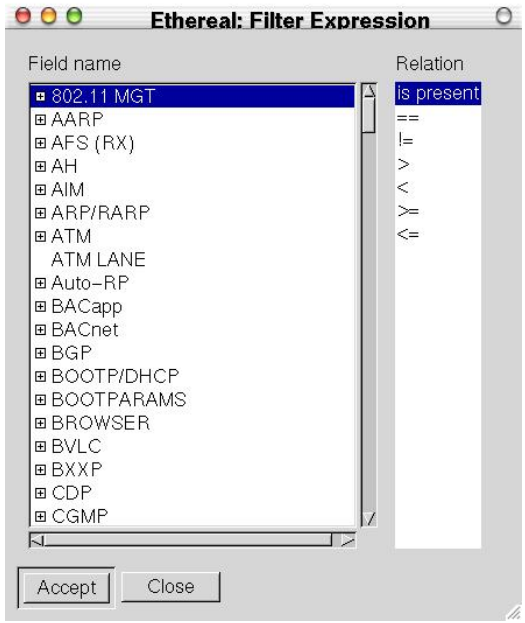

Figure 3-25. The Ethereal Add Expression dialog box, view 1

When you first bring up the Add Expression dialog box you are shown a tree list of field names, organized by protocol, and a box for selecting a relation.

Field Name

Select a protocols field from the protocol field tree. Every protocol with filterable fields is listed at the top level. By clicking on the "+" next to a protocol name you can get a list of the field names available for filtering for that protocol.

Relation

Select a relation from the list of available relation. The **is present** is a unary relation which is true if the selected field is present in a packet. All other listed relations are binary relations require additional data (ie a **Value** to match) to complete.

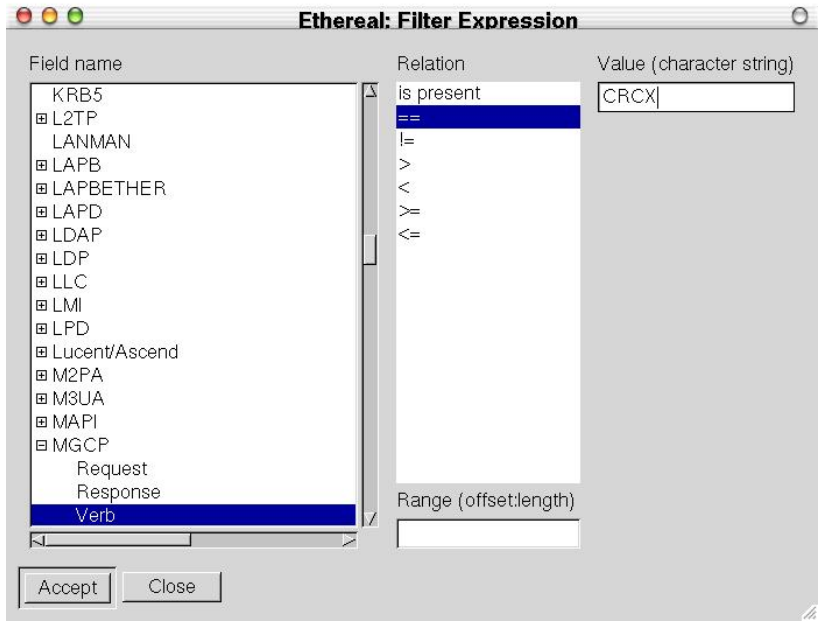

Figure 3-26. The Ethereal Add Expression dialog box, view 2

When you select a field from the field name list and select a binary relation (like the equality relation ==) you will be given the opportunity to enter a value, and possible some range information.

Value

You may enter an appropriate value in the **Value** text box. The **Value** will also indicate the type of value for the **field name** you have selected (like character string).

Accept

When you have built a satisfactory expression click **Accept** and a filter string will be built for you.

Close

You can leave the **Add Expression...** dialog box without any effect by clicking the **Close**

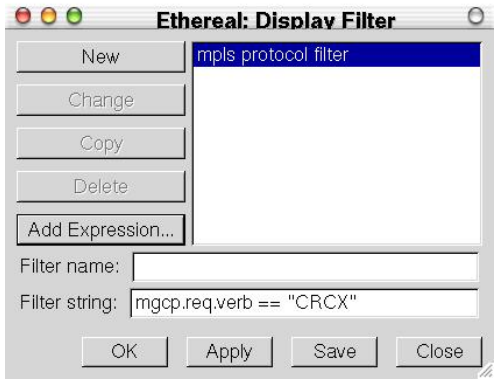

Figure 3-27. The result of building a filter string using the Add Expression dialog box.

The Add Expression dialog box is an excellent way to learn to write Ethereal display filter strings.

3.16. Printing packets

Ethereal provides two methods for printing packets:

1. Select the Print... menu item from the File menu. When you do this, Ethereal pops up the Print dialog box as shown in Figure 3-28.
2. Select the Print Packet menu item from the File menu (or type Ctrl-P) and Ethereal will print the currently selected packet.

We present more detail on the Print dialog box below.

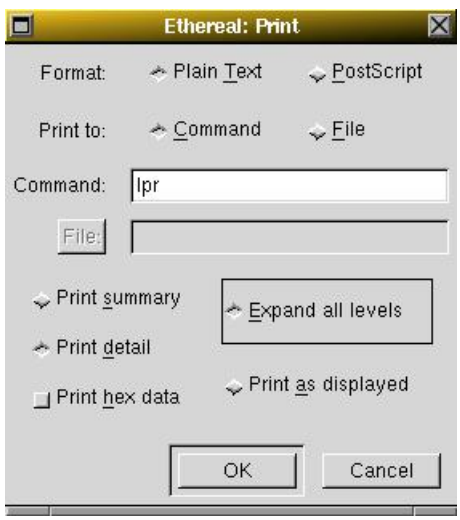

Figure 3-28. The Ethereal Print dialog box

Currently, there is no simple way with the Print dialog box to print only a range of packets, or to print a single packet. To do this, first select a range of packets with a display filter, then select Print... from the File menu. You could even select a single packet with something like **frame.number == 10** or a range by frame number with something like **frame.number >= 10 && frame.number <= 20**.

The following fields are available in the Print dialog box:

Format

This field contains a pair of mutually exclusive radio buttons:

- **Plain Text**, which specifies that the packet print should be in plain text.
- **PostScipt**, which specifies that the packet print process should use Postscript to generate a better print.

Print to

This field contains another pair of mutually exclusive radio buttons:

- **Command**, which specifies that a command be used for printing.
- **File**, which specifies that printing be done to a file.

Command

This field specifies the command to use for printing. It is typically **lpr**. You would change it to specify a particular queue if you need to print to a queue other than the default. An example might be:

```
lpr -Pmypostscript
```

This field is greyed out if **Command** is not specified above.

File

This field is where you enter the **file** to print to if you have selected Print to a file. It is greyed out if Print to a file is not selected.

Print summary and Print detail

This pair of mutually exclusive radio boxes select whether or not Ethereal prints a summary or the detail for each packet printed.

Expand all details and Print as displayed

This pair of mutially exclusive radio boxes select whether or not Ethereal expands all details for all packets printed, or prints them as displayed (ie, with only the currently expanded protocol trees expanded.

Print hex data

This radio box controls whether or not Ethereal prints the hex data for each packet selected.

3.17. Ethereal preferences

There are a number of preferences you can set from one place. Simply select the Preferences... menu item from the Edit menu, and Ethereal will pop up the Preferences dialog box as shown in Figure 3-29.

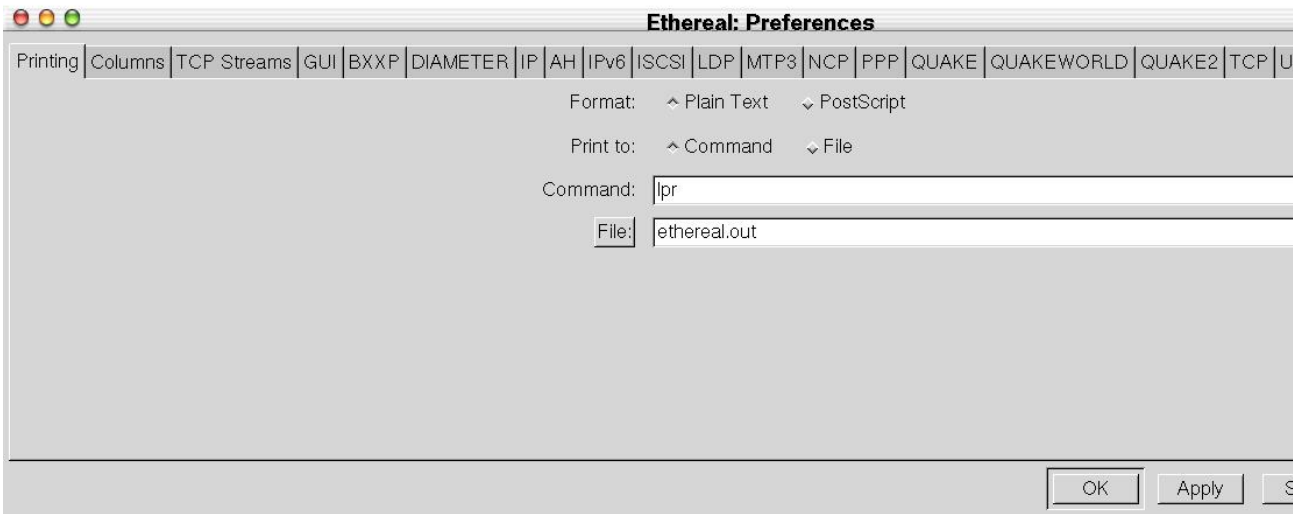

Figure 3-29. The Ethereal Preferences dialog box

The Ethereal Preferences dialog box is a tabbed dialog box that allows you to set preferences for each of the following elements:

Printing

This tab allows you to define the default printing command that Ethereal will use as well as the default output file name when you print to a file. These are discussed in more detail in Section 3.16

Columns

This tab allows you to select which columns appear in the Packet List Pane.

TCP Streams

This tab allows you to change the foreground and background colors used by the **Follow TCP Stream** described in Section 3.3.5.

GUI

This tab allows you to configure various characteristics of the GUI.

Other tabs

The remaining tabs allow you to configure various preferences for the dissection of various network protocols.

3.18. Files used by Ethereal

Ethereal uses a number of files while it is running. Some of these reside in $HOME/.ethereal and are used to maintain information between runs of Ethereal, while some of them are maintained in system areas.

The following are some of the files accessed by Ethereal:

$HOME/.ethereal/preferences

This file contains all your Ethereal preferences, including defaults for capturing and displaying packets. It is a simple text file containing statements of the form **variable: value**.

$HOME/.ethereal/filters

This file contains all the filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
"<filter name>" <filter string>
```

$HOME/.ethereal/colorfilters

This file contains all the color filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
@<filter name>@<filter string>@[<bg RGB(16-bit)>][<fg RGB(16-bit)>]
```

/usr/share/ethereal/plugins, /usr/local/share/ethereals/plugins,$HOME/.ethereal/plugins

Ethereal searches for plugins in the directories listed above. They are searched in the order listed.

/etc/ethers, $HOME/.ethereal/ethers

When Ethereal is trying to translate Ethernet hardware addresses to names, it consunts the files listed above in the order listed. If an address is not found in /etc/ethers, Etherereal looks in $HOME/.ethereal/etheres

Each line in these files consists of one hardware address and name separated by whitespace. The digits of hardware addressses are spearated by colons (:), dashes (-) or periods(.). The following are some examples:

```
ff-ff-ff-ff-ff-ff    Broadcast
c0-00-ff-ff-ff-ff    TR_broadcast
00.2b.08.93.4b.a1    Freds_machine
```

/usr/local/etc/manuf

Ethereal uses the file listed above to translate the first three bytes of an Ethernet address into a manufacturers name. This file has the same format as the ethers file, except addresses are three bytes long.

$HOME/.ethereal/ipxnets

Ethereal uses the above file to translate IPX network numbers into names.

An example is:

```
C0.A8.2C.00      HR
c0-a8-1c-00      CEO
00:00:BE:EF      IT_Server1
110f             FileServer3
```

4. Troubleshooting with Ethereal

4.1. An approach to troubleshooting with Ethereal

Ethereal is a very useful tool for network troubleshooting, since it contains a number of features that allow you to quickly focus on problems in your networkfor several reasons:

- It allows you to focus in on specific packets and protocols, as you can see a large amount of detail associated with various protocols.
- It supports a large number of protocols, and the list of protocols supported is growing as more people contribute dissectors
- By giving you a visual view of traffic in parts of your network, and providing tools to filter and colorize that information, you can get a better feel for your network traffic, and can understand your network better.

The following general approach is suggested:

- Determine that the problem looks like a networking problem. There is no point in capturing packets if the problem is not networking related.
- Figure out where to capture packets. You will have to capture packets from a part of the network where you can actually get network traffic related to the problem. This is especially important in the presence of switches and routers. See Section 4.2 for more details.

  Because Ethereal can read many capture file formats, you can capture using any conventient tool. One useful approach is to use **tcpdump** to capture on remote systems and then copy the capture file to your system for later analysis. For more details on capturing with **tcpdump**, see Section 5.1.

- Once you have captured packets that you think relate to the problem, load them into Ethereal and look for your problem. Using Ethereal's filtering and colorization capabilities, you can quickly narrow down the capture to the area of interest.
- Examine the appropriate fields within the packets where the problem appears to be. These can often help to reveal the problem.

4.2. Capturing in the presence of switches and routers

Many vendor’s switches support a feature known as "port spanning" or "port mirroring" in which all of the traffic to and from port A are also sent out port B. An excellent reference on the "port spanning" feature of Cisco switches can be found at Configuring the Catalyst Switched Port Analyzer (SPAN) Feature (http://www.cisco.com/warp/public/473/41.html)

4.3. Examples of troubleshooting

Troubleshooting often requires a reasonable knowledge of the protocols in question, however, you can often get a good idea of what might be going wrong simply by looking in the packets being exchanged.

5. Related tools

5.1. Capturing with tcpdump for viewing with Ethereal

There are occasions when you want to capture packets using **tcpdump** rather than **ethereal**, especially when you want to do a remote capture and do not want the network load associated with running Ethereal remotely (not to mention all the X traffic polluting your capture).

However, the default **tcpdump** parameters result in a capture file where each packet is truncated, because **tcpdump**, by default, does not capture full packets.

To ensure that you capture complete packets, use the following command:

```
tcpdump -i <interface> -s 1500 -w <some-file>
```

You will have to specify the correct **interface** and the name of a **file** to save into. In addition, you will have to terminate the capture with ^C when you believe you have captured enough packets.

5.2. Tethereal, for terminal-based capturing

Tethereal is a terminal oriented version of ethereal designed for capturing and displaying packets when you do not have a graphical environment available. It supports the same option set that **ethereal** does. For more information on **tethereal**, see the manual pages (**man tethereal**).

5.3. Using editcap

Included with Ethereal is a small utility called **editcap**, which is a command-line utility for working with capture files. Its main function is to remove packets from capture file, but it can also be used to convert capture files from one format to another, as well as print information about capture files.

editcap has the following format:

editcap [-r] [-h] [-v] [-T {encap type}] [-F {capture type}] {infile} {outfile} [record# [-] [record#] ...]

Where each option has the following meaning:

-r

This option specifies that the frames listed should be kept, not deleted. The default is to delete the listed frames.

-h

This option provides help.

-v

This option specifies verbose operation. The default is silent operation.

-T {encap type}

This option specifies the frame encapsulation type to use. It can take one of the following values:

- ether - Ethernet
- tr - Token Ring
- slip - SLIP
- ppp - PPP
- fddi - FDDI
- fddi-swapped - FDDI with bit-swapped MAC addresses
- rawip - Raw IP
- arcnet - ARCNET
- atm-rfc1483 - RFC 1483 ATM
- linux-atm-clip - Linux ATM CLIP
- lapb - LAPB
- atm-sniffer - ATM Sniffer
- null - NULL
- ascend - Lucent/Ascend access equipment
- lapd - LAPD
- v120 - V.120

It is mainly for converting funny captures to something that Ethereal can deal with. The default frame encapsulation type is the same as the input encapsulation.

-F {capture type}

This option specifies the capture file format to write the output file in. You can choose from the following values:

- libpcap - libpcap (tcpdump, Ethereal, etc.)
- modlibpcap - modified libpcap (tcpdump)
- rh6_1libpcap - Red Hat Linux 6.1 libpcap (tcpdump)
- ngsniffer - Network Associates Sniffer (DOS-based)
- snoop - Sun snoop
- netmon1 - Microsoft Network Monitor 1.x
- ngwsniffer_1_1 - Network Associates Sniffer (Windows-based) 1.1

The default is libpcap format.

{infile}

This parameter specifies the input file to use. It must be present.

{outfile}

This parameter specifies the output file to use. It must be present.

[record#[-][record# ...]]

This optional parameter specifies the records to include or exclude (depending on the **-r** option. You can specify individual records or a range of records.

5.4. Merging multiple capture files into a single capture file with mergecap

Mergecap is a program that combines multiple saved capture files into a single output file specified by the -w argument. Mergecap knows how to read libpcap capture files, including those of tcpdump. In addition, Mergecap can read capture files from snoop (including Shomiti) and atmsnoop, LanAlyzer, Sniffer (compressed or uncompressed), Microsoft Network Monitor, AIX’s iptrace, NetXray, Sniffer Pro, RADCOM’s WAN/LAN analyzer, Lucent/Ascend router debug output, HP-UX’s nettl, and the dump output from Toshiba’s ISDN routers. There is no need to tell Mergecap what type of file you are reading; it will determine the file type by itself. Mergecap is also capable of reading any of these file formats if they are compressed using gzip. Mergecap recognizes this directly from the file; the ’.gz’ extension is not required for this purpose.

By default, it writes the capture file in libpcap format, and writes all of the packets in both input capture files to the output file. The -F flag can be used to specify the format in which to write the capture file; it can write the file in libpcap format (standard libpcap format, a modified format used by some patched versions of libpcap, the format used by Red Hat Linux 6.1, or the format used by SuSE Linux 6.3), snoop format, uncompressed Sniffer format, Microsoft Network Monitor 1.x format, and the format used by Windows-based versions of the Sniffer software.

Packets from the input files are merged in chronological order based on each frame's timestamp, unless the -a flag is specified. Mergecap assumes that frames within a single capture file are already stored in chronological order. When the -a flag is specified, packets are copied directly from each input file to the output file, independent of each frame's timestamp.

If the -s flag is used to specify a snapshot length, frames in the input file with more captured data than the specified snapshot length will have only the amount of data specified by the snapshot length written to the output file. This may be useful if the program that is to read the output file cannot handle packets larger than a certain size (for example, the versions of snoop in Solaris 2.5.1 and Solaris 2.6 appear to reject Ethernet frames larger than the standard Ethernet MTU, making them incapable of handling gigabit Ethernet captures if jumbo frames were used).

If the -T flag is used to specify an encapsulation type, the encapsulation type of the output capture file will be forced to the specified type, rather than being the type appropriate to the encapsulation type of the input capture file. Note that this merely forces the encapsulation type of the output file to be the specified type; the packet headers of the packets will not be translated from the encapsulation type of the input capture file to the specified encapsulation type (for example, it will not translate an Ethernet capture to an FDDI capture if an Ethernet capture is read and '-T fddi' is specified).

```
hagbard@hagbard:~/build/src/ethereal/doc$ mergecap -h
mergecap version 0.8.19
Usage: mergecap [-h] [-v] [-a] [-s <snaplen>] [-T <encap type>]
          [-F <capture type>] -w <outfile> <infile> [...]

  where -h produces this help listing.
        -v verbose operation, default is silent
        -a files should be concatenated, not merged
             Default merges based on frame timestamps
        -s <snaplen>: truncate packets to <snaplen> bytes of data
        -w <outfile>: sets output filename to <outfile>
        -T <encap type> encapsulation type to use:
             ether - Ethernet
             tr - Token Ring
```

```
                slip - SLIP

                ppp - PPP
                fddi - FDDI
                fddi-swapped - FDDI with bit-swapped MAC addresses
                rawip - Raw IP
                arcnet - ARCNET
                atm-rfc1483 - RFC 1483 ATM
                linux-atm-clip - Linux ATM CLIP
                lapb - LAPB
                atm-sniffer - ATM Sniffer
                null - NULL
                ascend - Lucent/Ascend access equipment
                lapd - LAPD
                v120 - V.120
                ppp-with-direction - PPP with Directional Info
                ieee-802-11 - IEEE 802.11 Wireless LAN
                linux-sll - Linux cooked-mode capture
                frelay - Frame Relay
                chdlc - Cisco HDLC
                default is the same as the first input file
          -F <capture type> capture file type to write:
                libpcap - libpcap (tcpdump, Ethereal, etc.)
                rh6_1libpcap - Red Hat Linux 6.1 libpcap (tcpdump)
                suse6_3libpcap - SuSE Linux 6.3 libpcap (tcpdump)
                modlibpcap - modified libpcap (tcpdump)
                nokialibpcap - Nokia libpcap (tcpdump)
                ngsniffer - Network Associates Sniffer (DOS-based)
                snoop - Sun snoop
                netmon1 - Microsoft Network Monitor 1.x
                netmon2 - Microsoft Network Monitor 2.x
                ngwsniffer_1_1 - Network Associates Sniffer (Windows-
based) 1.1
                default is libpcap
```

Example 5-1. Help information available from mergecap

-h

Prints the version and options and exits.

-v

Causes **mergecap** to print a number of messages while it’s working.

-a

Causes the frame timestamps to be ignored, writing all packets from the first input file followed by all packets from the second input file. By default, when **-a** is not specified, the contents of the input files are merged in chronological order based on each frame's timestamp. Note: when merging, mergecap assumes that packets within a capture file are already in chronological order.

-s

Sets the snapshot length to use when writing the data.

-w

Sets the output filename.

-T

Sets the packet encapsulation type of the output capture file.

-F

Sets the file format of the output capture file.

A simple example merging `dhcp-capture.libpcap` and `imap-1.libpcap` into `outfile.libpcap` is shown below.

```
hagbard@hagbard:~/captures$ mergecap -w outfile.libpcap dhcp-capture.libpca
1.libpcap
```

Example 5-2. Simple example of using mergecap

5.5. Converting ASCII hexdumps to network captures with text2pcap

There may be some occasions when you wish to convert a hex dump of some network traffic into a libpcap file.

Text2pcap is a program that reads in an ASCII hex dump and writes the data described into a libpcap-style capture file. text2pcap can read hexdumps withmultiple packets in them, and build a capture file of multiple packets. text2pcap is also capable of generating dummy Ethernet, IP and UDP headers, in order to build fully processable packet dumps from hexdumps of application-level data only.

Text2pcap understands a hexdump of the form generated by od -t x1. In other words, each byte is individually displayed and surrounded with a space. Each line begins with an offset describing the position in the file. The offset is a hex number

(can also be octal - see -o), of more than two hex digits. Here is a sample dump that text2pcap can recognize:

```
000000 00 e0 1e a7 05 6f 00 10 ........
000008 5a a0 b9 12 08 00 46 00 ........
000010 03 68 00 00 00 00 0a 2e ........
000018 ee 33 0f 19 08 7f 0f 19 ........
000020 03 80 94 04 00 00 10 01 ........
000028 16 a2 0a 00 03 50 00 0c ........
000030 01 01 0f 19 03 80 11 01 ........
```

There is no limit on the width or number of bytes per line. Also the text dump at the end of the line is ignored. Bytes/hex numbers can be uppercase or lowercase. Any text before the offset is ignored, including email forwarding characters '>'. Any lines of text between the bytestring lines is ignored. The offsets are used to track the bytes, so offsets must be correct. Any line which has only bytes without a leading offset is ignored. An offset is recognized as being a hex number longer than two characters. Any text after the bytes is ignored (e.g. the character dump). Any hex numbers in this text are also ignored. An offset of zero is indicative of starting a new packet, so a single text file with a series of hexdumps can be converted into a packet capture with multiple packets. Multiple packets are read in with timestamps differing by one second each. In general, short of these restrictions, text2pcap is pretty liberal about reading in hexdumps and has been tested with a variety of mangled outputs (including being forwarded through email multiple times, with limited line wrap etc.)

There are a couple of other special features to note. Any line where the first non-whitespace character is '#' will be ignored as a comment. Any line beginning with #TEXT2PCAP is a directive and options can be inserted after this command to be processed by text2pcap. Currently there are no directives implemented; in the future, these may be used to give more fine grained control on the dump and the way it should be processed e.g. timestamps, encapsulation type etc.

Text2pcap also allows the user to read in dumps of application-level data, by inserting dummy L2, L3 and L4 headers before each packet. The user can elect to insert Ethernet headers, Ethernet and IP, or Ethernet, IP and UDP headers before each packet. This allows Ethereal or any other full-packet decoder to handle these dumps.

```
hagbard@hagbard:~/build/src/ethereal/doc$ text2pcap -h
text2pcap: invalid option -- h

Usage: text2pcap [-d] [-q] [-o h|o] [-l typenum] [-e l3pid] [-i proto
          [-u srcp destp] <input-filename> <output-filename>
```

```
where <input-filename> specifies input filename (use - for stan-
dard input)

      <output-filename> specifies output filename (use - for stan-
dard output)

[options] are one or more of the following

 -w filename  : Write capfile to <filename>. Default is standard output
 -h           : Display this help message
 -d           : Generate detailed debug of parser states
 -o hex|oct   : Parse offsets as (h)ex or (o)ctal. Default is hex
 -l typenum   : Specify link-layer type number. Default is 1 (Eth-
ernet).
                See net/bpf.h for list of numbers.
 -q           : Generate no output at all (automatically turns off -
d)
 -e l3pid     : Prepend dummy Ethernet II header with specified L3PID (in H
                Example: -e 0x800
 -i proto     : Prepend dummy IP header with specified IP proto-
col (in DECIMAL).
                Automatically prepends Ethernet header as well. Ex-
ample: -i 46
 -u srcp destp: Prepend dummy UDP header with specified dest and source por
                Automatically prepends Ethernet and IP headers as well
                Example: -u 30 40
```

Example 5-3. Help information available for text2pcap

-w <filename>

Write the capture file generated by **text2pcap** to <filename>. The default is to write to standard output.

-h

Display the help message

-d

Displays debugging information during the process. Can be used multiple times to generate more debugging information.

-q

Be completely quiet during the process.

-o hex|oct

Specify the radix for the offsets (hex or octal). Defaults to hex. This corresponds to the **-A** option for od.

-l

Specify the link-layer type of this packet. Default is Ethernet(1). See net/bpf.h for the complete list of possible encapsulations. Note that this option should be used if your dump is a complete hex dump of an encapsulated packet and you wish to specify the exact type of encapsulation. Example: -l 7 for ARCNet packets.

-e l3pid

Include a dummy Ethernet header before each packet. Specify the L3PID for the Ethernet header in hex. Use this option if your dump has Layer 3 header and payload (e.g. IP header), but no Layer 2 encapsulation. Example: -e 0x806 to specify an ARP packet.

For IP packets, instead of generating a fake Ethernet header you can also use -l 12 to indicate a raw IP packet to Ethereal. Note that -l 12 does not work for any non-IP Layer 3 packet (e.g. ARP), whereas generating a dummy Ethernet header with -e works for any sort of L3 packet.

-u srcport destport

Include dummy UDP headers before each packet. Specify the source and destination UDP ports for the packet in decimal. Use this option if your dump is the UDP payload of a packet but does not include any UDP, IP or Ethernet headers. Note that this automatically includes appropriate Ethernet and IP headers with each packet. Example: -u 1000 69 to make the packets look like TFTP/UDP packets.

5.6. Creating dissectors from Corba IDL files with idl2eth

In an ideal world idl2eth would be mentioned in the users guide in passing and documented in the developers guide. As the developers guide has not yet been completed it will be documented here.

5.6.1. What is it?

As you have probably guessed from the name, **idl2eth** takes a user specified IDL file and attempts to build a dissector that can decode the IDL traffic over GIOP. The resulting file is "C" code, that should compile okay as an ethereal dissector.

idl2ethbasically parses the data struct given to it by the omniidl compiler, and using the GIOP API available in packet-giop.[ch], generates get_CDR_xxx calls to decode the CORBA traffic on the wire.

It consists of 4 main files.

`README.idl2eth`

This document

`ethereal_be.py`

The main compiler backend

`ethereal_gen.py`

A helper class, that generates the C code.

`idl2eth`

A simple shell script wrapper that the end user should use to generate the dissector from the IDL file(s).

5.6.2. Why do this?

It is important to understand how CORBA traffic looks like over GIOP/IIOP, and to help build a tool that can assist in troubleshooting CORBA interworking. This was especially the case after seeing a lot of discussions about how particular IDL types are represented inside an octet stream.

I have also had comments/feedback that this tool would be good for say a CORBA class when teaching students how CORBA traffic looks like "on the wire".

It is also COOL to work on a great Open Source project such as the case with "Ethereal" (http://www.ethereal.com)

5.6.3. How to use idl2eth

To use the idl2eth to generate ethereal dissectors, you need the following:

Prerequisites to using idl2eth

1. Python must be installed. See http://python.org/

2. omniidl from the the omniORB package must be available. http://www.uk.research.att.com/omniORB/omniORB.html
3. Of course you need ethereal installed to compile the code an tweak it if required. idl2eth is part of the standard Ethereal distribution

To use idl2eth to generate an ethereal dissector from an idl file use the following proceedure:

Proceedure for converting a Corba idl file into an ethereal dissector

1. To write the C code to stdout.

   ```
   idl2eth  <your file.idl>
   ```

   eg:

   ```
   idl2eth echo.idl
   ```

2. To write to a file, just redirect the output.

   ```
   idl2eth echo.idl > packet-test-idl.c
   ```

   You may wish to comment out the register_giop_user_module() code and that will leave you with heuristic dissection.

If you dont want to use the shell script wrapper, then try steps 3 or 4 instead.

3. To write the C code to stdout.

   ```
   Usage: omniidl  -p ./ -b ethereal_be <your file.idl>
   ```

   eg:

   ```
   omniidl  -p ./ -b ethereal_be echo.idl
   ```

4. To write to a file, just redirect the output.

   ```
   omniidl  -p ./ -b ethereal_be echo.idl > packet-test-idl.c
   ```

   You may wish to comment out the register_giop_user_module() code and that will leave you with heuristic dissection.

5. Copy the resulting C code to your ethereal src directory, edit the 2 make files to include the packet-test-idl.c

   ```
   cp packet-test-idl.c /dir/where/ethereal/lives/
   edit Makefile.am
   edit Makefile.nmake
   ```

6. Run configure

   ```
   ./configure (or ./autogen.sh)
   ```

7. Compile the code

```
make
```

8. Good Luck !!

5.6.4. TODO

1. Exception code not generated (yet), but can be added manually.
2. Enums not converted to symbolic values (yet), but can be added manually.
3. Add command line options etc
4. More I am sure :-)

5.6.5. Limitations

See the TODO list inside `packet-giop.c`

5.6.6. Notes

1. The "-p ./" option passed to omniidl indicates that the ethereal_be.py and ethereal_gen.py are residing in the current directory. This may need tweaking if you place these files somewhere else.
2. If it complains about being unable to find some modules (eg tempfile.py), you may want to check if PYTHONPATH is set correctly. On my Linux box, it is PYTHONPATH=/usr/lib/python1.5/

A. Ethereal Display Filter Fields

A.1. 802.1q Virtual LAN (vlan)

Field	Field Name	Type
vlan.cfi	CFI	Unsigned 16-bit integer
vlan.etype	Type	Unsigned 16-bit integer
vlan.id	ID	Unsigned 16-bit integer
vlan.len	Length	Unsigned 16-bit integer
vlan.priority	Priority	Unsigned 16-bit integer
vlan.trailer	Trailer	Byte array

Table A-1. 802.1q Virtual LAN (vlan)

A.2. AOL Instant Messenger (aim)

Field	Field Name	Type
aim.channel	Channel ID	Unsigned 8-bit integer
aim.cmd_start	Command Start	Unsigned 8-bit integer
aim.datalen	Data Field Length	Unsigned 16-bit integer
aim.fnac.family	FNAC Family ID	Unsigned 16-bit integer
aim.fnac.subtype	FNAC Subtype ID	Unsigned 16-bit integer
aim.seqno	Sequence Number	Unsigned 16-bit integer

Table A-2. AOL Instant Messenger (aim)

A.3. ATM (atm)

Field	Field Name	Type
atm.vci	VCI	Unsigned 16-bit integer
atm.vpi	VPI	Unsigned 8-bit integer

Table A-3. ATM (atm)

A.4. ATM LAN Emulation (lane)

Field	Field Name	Type

Table A-4. ATM LAN Emulation (lane)

A.5. Address Resolution Protocol (arp)

Field	Field Name	Type
arp.dst.atm_num_e164	Target ATM number (E.164)	String
arp.dst.atm_num_nsap	Target ATM number (NSAP)	Byte array
arp.dst.atm_subaddr	Target ATM subaddress	Byte array
arp.dst.hlen	Target ATM number length	Unsigned 8-bit integer
arp.dst.htype	Target ATM number type	Boolean
arp.dst.hw	Target hardware address	Byte array
arp.dst.pln	Target protocol size	Unsigned 8-bit integer
arp.dst.proto	Target protocol address	Byte array
arp.dst.slen	Target ATM subaddress length	Unsigned 8-bit integer
arp.dst.stype	Target ATM subaddress type	Boolean
arp.hw.size	Hardware size	Unsigned 8-bit integer
arp.hw.type	Hardware type	Unsigned 16-bit integer
arp.opcode	Opcode	Unsigned 16-bit integer
arp.proto.size	Protocol size	Unsigned 8-bit integer
arp.proto.type	Protocol type	Unsigned 16-bit integer
arp.src.atm_num_e164	Sender ATM number (E.164)	String
arp.src.atm_num_nsap	Sender ATM number (NSAP)	Byte array
arp.src.atm_subaddr	Sender ATM subaddress	Byte array

Field	Field Name	Type
arp.src.hlen	Sender ATM number length	Unsigned 8-bit integer
arp.src.htype	Sender ATM number type	Boolean
arp.src.hw	Sender hardware address	Byte array
arp.src.pln	Sender protocol size	Unsigned 8-bit integer
arp.src.proto	Sender protocol address	Byte array
arp.src.slen	Sender ATM subaddress length	Unsigned 8-bit integer
arp.src.stype	Sender ATM subaddress type	Boolean

Table A-5. Address Resolution Protocol (arp)

A.6. Andrew File System (AFS) (afs)

Field	Field Name	Type
afs.backup	Backup	Boolean
afs.backup.errcode	Error Code	Unsigned 32-bit integer
afs.backup.opcode	Operation	Unsigned 32-bit integer
afs.bos	BOS	Boolean
afs.bos.baktime	Backup Time	Date/Time stamp
afs.bos.cell	Cell	String
afs.bos.cmd	Command	String
afs.bos.content	Content	String
afs.bos.data	Data	Byte array
afs.bos.date	Date	Unsigned 32-bit integer
afs.bos.errcode	Error Code	Unsigned 32-bit integer
afs.bos.error	Error	String
afs.bos.file	File	String
afs.bos.flags	Flags	Unsigned 32-bit integer
afs.bos.host	Host	String
afs.bos.instance	Instance	String
afs.bos.key	Key	Byte array
afs.bos.keychecksum	Key Checksum	Unsigned 32-bit integer
afs.bos.keymodtime	Key Modification Time	Date/Time stamp

Field	Field Name	Type
afs.bos.keyspare2	Key Spare 2	Unsigned 32-bit integer
afs.bos.kvno	Key Version Number	Unsigned 32-bit integer
afs.bos.newtime	New Time	Date/Time stamp
afs.bos.number	Number	Unsigned 32-bit integer
afs.bos.oldtime	Old Time	Date/Time stamp
afs.bos.opcode	Operation	Unsigned 32-bit integer
afs.bos.parm	Parm	String
afs.bos.path	Path	String
afs.bos.size	Size	Unsigned 32-bit integer
afs.bos.spare1	Spare1	String
afs.bos.spare2	Spare2	String
afs.bos.spare3	Spare3	String
afs.bos.status	Status	Signed 32-bit integer
afs.bos.statusdesc	Status Description	String
afs.bos.type	Type	String
afs.bos.user	User	String
afs.cb	Callback	Boolean
afs.cb.callback.expires	Expires	Date/Time stamp
afs.cb.callback.type	Type	Unsigned 32-bit integer
afs.cb.callback.version	Version	Unsigned 32-bit integer
afs.cb.errcode	Error Code	Unsigned 32-bit integer
afs.cb.fid.uniq	FileID (Uniqifier)	Unsigned 32-bit integer
afs.cb.fid.vnode	FileID (VNode)	Unsigned 32-bit integer
afs.cb.fid.volume	FileID (Volume)	Unsigned 32-bit integer
afs.cb.opcode	Operation	Unsigned 32-bit integer
afs.error	Error	Boolean
afs.error.opcode	Operation	Unsigned 32-bit integer
afs.fs	File Server	Boolean
afs.fs.acl.a	_A_dminister	Unsigned 8-bit integer
afs.fs.acl.count.negative	ACL Count (Negative)	Unsigned 32-bit integer
afs.fs.acl.count.positive	ACL Count (Positive)	Unsigned 32-bit integer
afs.fs.acl.d	_D_elete	Unsigned 8-bit integer
afs.fs.acl.datasize	ACL Size	Unsigned 32-bit integer
afs.fs.acl.entity	Entity (User/Group)	String
afs.fs.acl.i	_I_nsert	Unsigned 8-bit integer
afs.fs.acl.k	_L_ock	Unsigned 8-bit integer
afs.fs.acl.l	_L_ookup	Unsigned 8-bit integer

Field	Field Name	Type
afs.fs.acl.r	_R_ead	Unsigned 8-bit integer
afs.fs.acl.w	_W_rite	Unsigned 8-bit integer
afs.fs.callback.expires	Expires	Date/Time stamp
afs.fs.callback.type	Type	Unsigned 32-bit integer
afs.fs.callback.version	Version	Unsigned 32-bit integer
afs.fs.cps.spare1	CPS Spare1	Unsigned 32-bit integer
afs.fs.cps.spare2	CPS Spare2	Unsigned 32-bit integer
afs.fs.cps.spare3	CPS Spare3	Unsigned 32-bit integer
afs.fs.data	Data	Byte array
afs.fs.errcode	Error Code	Unsigned 32-bit integer
afs.fs.fid.uniq	FileID (Uniqifier)	Unsigned 32-bit integer
afs.fs.fid.vnode	FileID (VNode)	Unsigned 32-bit integer
afs.fs.fid.volume	FileID (Volume)	Unsigned 32-bit integer
afs.fs.flength	FLength	Unsigned 32-bit integer
afs.fs.ipaddr	IP Address	IPv4 address
afs.fs.length	Length	Unsigned 32-bit integer
afs.fs.motd	Message of the Day	String
afs.fs.name	Name	String
afs.fs.newname	New Name	String
afs.fs.offlinemsg	Offline Message	String
afs.fs.offset	Offset	Unsigned 32-bit integer
afs.fs.oldname	Old Name	String
afs.fs.opcode	Operation	Unsigned 32-bit integer
afs.fs.status.anonymousaccess	Anonymous Access	Unsigned 32-bit integer
afs.fs.status.author	Author	Unsigned 32-bit integer
afs.fs.status.calleraccess	Caller Access	Unsigned 32-bit integer
afs.fs.status.clientmodtime	Client Modification Time	Date/Time stamp
afs.fs.status.dataversion	Data Version	Unsigned 32-bit integer
afs.fs.status.dataversionhigh	Data Version (High)	Unsigned 32-bit integer
afs.fs.status.filetype	File Type	Unsigned 32-bit integer
afs.fs.status.group	Group	Unsigned 32-bit integer
afs.fs.status.interfaceversion	Interface Version	Unsigned 32-bit integer
afs.fs.status.length	Length	Unsigned 32-bit integer

Field	Field Name	Type
afs.fs.status.linkcount	Link Count	Unsigned 32-bit integer
afs.fs.status.mask	Mask	Unsigned 32-bit integer
afs.fs.status.mask.fsync	FSync	Unsigned 32-bit integer
afs.fs.status.mask.setgroup	Set Group	Unsigned 32-bit integer
afs.fs.status.mask.setmode	Set Mode	Unsigned 32-bit integer
afs.fs.status.mask.setmodtime	Set Modification Time	Unsigned 32-bit integer
afs.fs.status.mask.setowner	Set Owner	Unsigned 32-bit integer
afs.fs.status.mask.setsegsize	Set Segment Size	Unsigned 32-bit integer
afs.fs.status.mode	Unix Mode	Unsigned 32-bit integer
afs.fs.status.owner	Owner	Unsigned 32-bit integer
afs.fs.status.parentunique	Parent Unique	Unsigned 32-bit integer
afs.fs.status.parentvnode	Parent VNode	Unsigned 32-bit integer
afs.fs.status.segsize	Segment Size	Unsigned 32-bit integer
afs.fs.status.servermodtime	Server Modification Time	Date/Time stamp
afs.fs.status.spare2	Spare 2	Unsigned 32-bit integer
afs.fs.status.spare3	Spare 3	Unsigned 32-bit integer
afs.fs.status.spare4	Spare 4	Unsigned 32-bit integer
afs.fs.status.synccounter	Sync Counter	Unsigned 32-bit integer
afs.fs.symlink.content	Symlink Content	String
afs.fs.symlink.name	Symlink Name	String
afs.fs.timestamp	Timestamp	Date/Time stamp
afs.fs.token	Token	Byte array
afs.fs.viceid	Vice ID	Unsigned 32-bit integer
afs.fs.vicelocktype	Vice Lock Type	Unsigned 32-bit integer
afs.fs.volid	Volume ID	Unsigned 32-bit integer
afs.fs.volname	Volume Name	String
afs.fs.volsync.spare1	Spare 1	Unsigned 32-bit integer
afs.fs.volsync.spare2	Spare 2	Unsigned 32-bit integer
afs.fs.volsync.spare3	Spare 3	Unsigned 32-bit integer
afs.fs.volsync.spare4	Spare 4	Unsigned 32-bit integer
afs.fs.volsync.spare5	Spare 5	Unsigned 32-bit integer

Field	Field Name	Type
afs.fs.volsync.spare6	Spare 6	Unsigned 32-bit integer
afs.fs.xstats.clientversion	Client Version	Unsigned 32-bit integer
afs.fs.xstats.collnumber	Collection Number	Unsigned 32-bit integer
afs.fs.xstats.timestamp	XStats Timestamp	Unsigned 32-bit integer
afs.fs.xstats.version	XStats Version	Unsigned 32-bit integer
afs.kauth	KAuth	Boolean
afs.kauth.data	Data	Byte array
afs.kauth.domain	Domain	String
afs.kauth.errcode	Error Code	Unsigned 32-bit integer
afs.kauth.kvno	Key Version Number	Unsigned 32-bit integer
afs.kauth.name	Name	String
afs.kauth.opcode	Operation	Unsigned 32-bit integer
afs.kauth.princ	Principal	String
afs.kauth.realm	Realm	String
afs.prot	Protection	Boolean
afs.prot.count	Count	Unsigned 32-bit integer
afs.prot.errcode	Error Code	Unsigned 32-bit integer
afs.prot.flag	Flag	Unsigned 32-bit integer
afs.prot.gid	Group ID	Unsigned 32-bit integer
afs.prot.id	ID	Unsigned 32-bit integer
afs.prot.maxgid	Maximum Group ID	Unsigned 32-bit integer
afs.prot.maxuid	Maximum User ID	Unsigned 32-bit integer
afs.prot.name	Name	String
afs.prot.newid	New ID	Unsigned 32-bit integer
afs.prot.oldid	Old ID	Unsigned 32-bit integer
afs.prot.opcode	Operation	Unsigned 32-bit integer
afs.prot.pos	Position	Unsigned 32-bit integer
afs.prot.uid	User ID	Unsigned 32-bit integer
afs.rmtsys	Rmtsys	Boolean
afs.rmtsys.opcode	Operation	Unsigned 32-bit integer
afs.ubik	Ubik	Boolean
afs.ubik.activewrite	Active Write	Unsigned 32-bit integer
afs.ubik.addr	Address	IPv4 address
afs.ubik.amsyncsite	Am Sync Site	Unsigned 32-bit integer
afs.ubik.anyreadlocks	Any Read Locks	Unsigned 32-bit integer
afs.ubik.anywritelocks	Any Write Locks	Unsigned 32-bit integer

Field	Field Name	Type
afs.ubik.beaconsincedown	Beacon Since Down	Unsigned 32-bit integer
afs.ubik.currentdb	Current DB	Unsigned 32-bit integer
afs.ubik.currenttran	Current Transaction	Unsigned 32-bit integer
afs.ubik.epochtime	Epoch Time	Date/Time stamp
afs.ubik.errcode	Error Code	Unsigned 32-bit integer
afs.ubik.file	File	Unsigned 32-bit integer
afs.ubik.interface	Interface Address	IPv4 address
afs.ubik.isclone	Is Clone	Unsigned 32-bit integer
afs.ubik.lastbeaconsent	Last Beacon Sent	Date/Time stamp
afs.ubik.lastvote	Last Vote	Unsigned 32-bit integer
afs.ubik.lastvotetime	Last Vote Time	Date/Time stamp
afs.ubik.lastyesclaim	Last Yes Claim	Date/Time stamp
afs.ubik.lastyeshost	Last Yes Host	IPv4 address
afs.ubik.lastyesstate	Last Yes State	Unsigned 32-bit integer
afs.ubik.lastyesttime	Last Yes Time	Date/Time stamp
afs.ubik.length	Length	Unsigned 32-bit integer
afs.ubik.lockedpages	Locked Pages	Unsigned 32-bit integer
afs.ubik.locktype	Lock Type	Unsigned 32-bit integer
afs.ubik.lowesthost	Lowest Host	IPv4 address
afs.ubik.lowesttime	Lowest Time	Date/Time stamp
afs.ubik.now	Now	Date/Time stamp
afs.ubik.nservers	Number of Servers	Unsigned 32-bit integer
afs.ubik.opcode	Operation	Unsigned 32-bit integer
afs.ubik.position	Position	Unsigned 32-bit integer
afs.ubik.recoverystate	Recovery State	Unsigned 32-bit integer
afs.ubik.site	Site	IPv4 address
afs.ubik.state	State	Unsigned 32-bit integer
afs.ubik.synchost	Sync Host	IPv4 address
afs.ubik.syncsiteuntil	Sync Site Until	Date/Time stamp
afs.ubik.synctime	Sync Time	Date/Time stamp
afs.ubik.tidcounter	TID Counter	Unsigned 32-bit integer
afs.ubik.up	Up	Unsigned 32-bit integer
afs.ubik.version.counter	Counter	Unsigned 32-bit integer
afs.ubik.version.epoch	Epoch	Date/Time stamp
afs.ubik.voteend	Vote Ends	Date/Time stamp
afs.ubik.votestart	Vote Started	Date/Time stamp

Field	Field Name	Type
afs.ubik.votetype	Vote Type	Unsigned 32-bit integer
afs.ubik.writelockedpages	Write Locked Pages	Unsigned 32-bit integer
afs.ubik.writetran	Write Transaction	Unsigned 32-bit integer
afs.update	Update	Boolean
afs.update.opcode	Operation	Unsigned 32-bit integer
afs.vldb	VLDB	Boolean
afs.vldb.bkvol	Backup Volume ID	Unsigned 32-bit integer
afs.vldb.bump	Bumped Volume ID	Unsigned 32-bit integer
afs.vldb.count	Volume Count	Unsigned 32-bit integer
afs.vldb.errcode	Error Code	Unsigned 32-bit integer
afs.vldb.id	Volume ID	Unsigned 32-bit integer
afs.vldb.index	Volume Index	Unsigned 32-bit integer
afs.vldb.name	Volume Name	String
afs.vldb.nextindex	Next Volume Index	Unsigned 32-bit integer
afs.vldb.numservers	Number of Servers	Unsigned 32-bit integer
afs.vldb.opcode	Operation	Unsigned 32-bit integer
afs.vldb.partition	Partition	String
afs.vldb.rovol	Read-Only Volume ID	Unsigned 32-bit integer
afs.vldb.rwvol	Read-Write Volume ID	Unsigned 32-bit integer
afs.vldb.server	Server	IPv4 address
afs.vldb.serveruuid	Server UUID	Byte array
afs.vldb.type	Volume Type	Unsigned 32-bit integer
afs.vol	Volume Server	Boolean
afs.vol.count	Volume Count	Unsigned 32-bit integer
afs.vol.errcode	Error Code	Unsigned 32-bit integer
afs.vol.id	Volume ID	Unsigned 32-bit integer
afs.vol.name	Volume Name	String
afs.vol.opcode	Operation	Unsigned 32-bit integer

Table A-6. Andrew File System (AFS) (afs)

A.7. Appletalk Address Resolution Protocol (aarp)

Field	Field Name	Type
aarp.dst.ether	Target ether	Byte array
aarp.dst.id	Target ID	Byte array
aarp.hard.size	Hardware size	Unsigned 8-bit integer
aarp.hard.type	Hardware type	Unsigned 16-bit integer
aarp.opcode	Opcode	Unsigned 16-bit integer
aarp.proto.size	Protocol size	Unsigned 8-bit integer
aarp.proto.type	Protocol type	Unsigned 16-bit integer
aarp.src.ether	Sender ether	Byte array
aarp.src.id	Sender ID	Byte array

Table A-7. Appletalk Address Resolution Protocol (aarp)

A.8. Async data over ISDN (V.120) (v120)

Field	Field Name	Type
v120.address	Link Address	Unsigned 16-bit integer
v120.control	Control Field	Unsigned 16-bit integer
v120.header	Header Field	String

Table A-8. Async data over ISDN (V.120) (v120)

A.9. Authentication Header (ah)

Field	Field Name	Type
ah.sequence	Sequence	Unsigned 32-bit integer
ah.spi	SPI	Unsigned 32-bit integer

Table A-9. Authentication Header (ah)

A.10. BACnet Virtual Link Control (bvlc)

Field	Field Name	Type

Field	Field Name	Type
bvlc.bdt_ip	IP	IPv4 address
bvlc.bdt_mask	Mask	Byte array
bvlc.bdt_port	Port	Unsigned 16-bit integer
bvlc.fdt_ip	IP	IPv4 address
bvlc.fdt_port	Port	Unsigned 16-bit integer
bvlc.fdt_timeout	Timeout	Unsigned 16-bit integer
bvlc.fdt_ttl	TTL	Unsigned 16-bit integer
bvlc.function	Function	Unsigned 8-bit integer
bvlc.fwd_ip	IP	IPv4 address
bvlc.fwd_port	Port	Unsigned 16-bit integer
bvlc.length	Length	Unsigned 16-bit integer
bvlc.reg_ttl	TTL	Unsigned 16-bit integer
bvlc.result	Result	Unsigned 16-bit integer
bvlc.type	Type	Unsigned 8-bit integer

Table A-10. BACnet Virtual Link Control (bvlc)

A.11. Banyan Vines (vines)

Field	Field Name	Type
vines.protocol	Protocol	Unsigned 8-bit integer

Table A-11. Banyan Vines (vines)

A.12. Banyan Vines Fragmentation Protocol (vines_frp)

Field	Field Name	Type

Table A-12. Banyan Vines Fragmentation Protocol (vines_frp)

A.13. Banyan Vines SPP (vines_spp)

Field	Field Name	Type

Table A-13. Banyan Vines SPP (vines_spp)

A.14. Blocks eXtensible eXchange Protocol (bxxp)

Field	Field Name	Type
bxxp.channel	Channel	Unsigned 32-bit integer
bxxp.end	End	Boolean
bxxp.more.complete	Complete	Boolean
bxxp.more.intermediate	Intermediate	Boolean
bxxp.req	Request	Boolean
bxxp.req.channel	Request Channel Number	Unsigned 32-bit integer
bxxp.rsp	Response	Boolean
bxxp.rsp.channel	Response Channel Number	Unsigned 32-bit integer
bxxp.seq	Sequence	Boolean
bxxp.seq.ackno	Ackno	Unsigned 32-bit integer
bxxp.seq.channel	Sequence Channel Number	Unsigned 32-bit integer
bxxp.seq.window	Window	Unsigned 32-bit integer
bxxp.seqno	Seqno	Unsigned 32-bit integer
bxxp.serial	Serial	Unsigned 32-bit integer
bxxp.size	Size	Unsigned 32-bit integer
bxxp.status.negative	Negative	Boolean
bxxp.status.positive	Positive	Boolean
bxxp.violation	Protocol Violation	Boolean

Table A-14. Blocks eXtensible eXchange Protocol (bxxp)

A.15. Boot Parameters (bootparams)

Field	Field Name	Type
bootparams.domain	Client Domain	String
bootparams.fileid	File ID	String
bootparams.filepath	File Path	String
bootparams.host	Client Host	String
bootparams.hostaddr	Client Address	IPv4 address
bootparams.routeraddr	Router Address	IPv4 address
bootparams.type	Address Type	Unsigned 32-bit integer

Table A-15. Boot Parameters (bootparams)

A.16. Bootstrap Protocol (bootp)

Field	Field Name	Type
bootp.cookie	Magic cookie	IPv4 address
bootp.dhcp	Frame is DHCP	Boolean
bootp.file	Boot file name	String
bootp.flag	Broadcast flag	Unsigned 16-bit integer
bootp.hops	Hops	Unsigned 8-bit integer
bootp.hw.addr	Client hardware address	Byte array
bootp.hw.len	Hardware address length	Unsigned 8-bit integer
bootp.hw.type	Hardware type	Unsigned 8-bit integer
bootp.id	Transaction ID	Unsigned 32-bit integer
bootp.ip.client	Client IP address	IPv4 address
bootp.ip.relay	Relay agent IP address	IPv4 address
bootp.ip.server	Next server IP address	IPv4 address
bootp.ip.your	Your (client) IP address	IPv4 address
bootp.secs	Seconds elapsed	Unsigned 16-bit integer
bootp.server	Server host name	String
bootp.type	Message type	Unsigned 8-bit integer

Table A-16. Bootstrap Protocol (bootp)

A.17. Border Gateway Protocol (bgp)

Field	Field Name	Type
bgp.type	BGP message type	Unsigned 8-bit integer

Table A-17. Border Gateway Protocol (bgp)

A.18. Building Automation and Control Network APDU (bacapp)

Field	Field Name	Type
bacapp.bacapp_type	APDU Type	Unsigned 8-bit integer

Table A-18. Building Automation and Control Network APDU (bacapp)

A.19. Building Automation and Control Network NPDU (bacnet)

Field	Field Name	Type
bacnet.control	Control	Unsigned 8-bit integer
bacnet.control_dest	Destination Specifier	Boolean
bacnet.control_expect	Expecting Reply	Boolean
bacnet.control_net	NSDU contains	Boolean
bacnet.control_prio_high	Priority	Boolean
bacnet.control_prio_low	Priority	Boolean
bacnet.control_res1	Reserved	Boolean
bacnet.control_res2	Reserved	Boolean
bacnet.control_src	Source specifier	Boolean
bacnet.dadr_eth	Destination ISO 8802-3 MAC Address	6-byte Hardware (MAC) Address
bacnet.dadr_tmp	Unknown Destination MAC	Byte array

Field	Field Name	Type
bacnet.dlen	Destination MAC Layer Address Length	Unsigned 8-bit integer
bacnet.dnet	Destination Network Address	Unsigned 16-bit integer
bacnet.hopc	Hop Count	Unsigned 8-bit integer
bacnet.mesgtyp	Message Type	Unsigned 8-bit integer
bacnet.perf	Performance Index	Unsigned 8-bit integer
bacnet.pinfo	Port Info	Unsigned 8-bit integer
bacnet.pinfolen	Port Info Length	Unsigned 8-bit integer
bacnet.portid	Port ID	Unsigned 8-bit integer
bacnet.rejectreason	Reject Reason	Unsigned 8-bit integer
bacnet.rportnum	Number of Port Mappings	Unsigned 8-bit integer
bacnet.sadr_eth	SADR	6-byte Hardware (MAC) Address
bacnet.sadr_tmp	Unknown Source MAC	Byte array
bacnet.slen	Source MAC Layer Address Length	Unsigned 8-bit integer
bacnet.snet	Source Network Address	Unsigned 16-bit integer
bacnet.vendor	Vendor ID	Unsigned 16-bit integer
bacnet.version	Version	Unsigned 8-bit integer

Table A-19. Building Automation and Control Network NPDU (bacnet)

A.20. Cisco Auto-RP (auto_rp)

Field	Field Name	Type
auto_rp.group_prefix	Prefix	IPv4 address
auto_rp.holdtime	Holdtime	Unsigned 16-bit integer
auto_rp.mask_len	Mask length	Unsigned 8-bit integer
auto_rp.pim_ver	Version	Unsigned 8-bit integer
auto_rp.prefix_sign	Sign	Unsigned 8-bit integer
auto_rp.rp_addr	RP address	IPv4 address
auto_rp.rp_count	RP count	Unsigned 8-bit integer
auto_rp.type	Packet type	Unsigned 8-bit integer
auto_rp.version	Protocol version	Unsigned 8-bit integer

Table A-20. Cisco Auto-RP (auto_rp)

A.21. Cisco Discovery Protocol (cdp)

Field	Field Name	Type
cdp.checksum	Checksum	Unsigned 16-bit integer
cdp.tlv.len	Length	Unsigned 16-bit integer
cdp.tlv.type	Type	Unsigned 16-bit integer
cdp.ttl	TTL	Unsigned 16-bit integer
cdp.version	Version	Unsigned 8-bit integer

Table A-21. Cisco Discovery Protocol (cdp)

A.22. Cisco Group Management Protocol (cgmp)

Field	Field Name	Type
cgmp.count	Count	Unsigned 8-bit integer
cgmp.gda	Group Destination Address	6-byte Hardware (MAC) Address
cgmp.type	Type	Unsigned 8-bit integer
cgmp.usa	Unicast Source Address	6-byte Hardware (MAC) Address
cgmp.version	Version	Unsigned 8-bit integer

Table A-22. Cisco Group Management Protocol (cgmp)

A.23. Cisco HDLC (chdlc)

Field	Field Name	Type
chdlc.address	Address	Unsigned 8-bit integer
chdlc.protocol	Protocol	Unsigned 16-bit integer

Table A-23. Cisco HDLC (chdlc)

A.24. Cisco Hot Standby Router Protocol (hsrp)

Field	Field Name	Type
hsrp.auth_data	Authentication Data	String
hsrp.group	Group	Unsigned 8-bit integer
hsrp.hellotime	Hellotime	Unsigned 8-bit integer
hsrp.holdtime	Holdtime	Unsigned 8-bit integer
hsrp.opcode	Op Code	Unsigned 8-bit integer
hsrp.priority	Priority	Unsigned 8-bit integer
hsrp.reserved	Reserved	Unsigned 8-bit integer
hsrp.state	State	Unsigned 8-bit integer
hsrp.version	Version	Unsigned 8-bit integer
hsrp.virt_ip	Virtual IP Address	IPv4 address

Table A-24. Cisco Hot Standby Router Protocol (hsrp)

A.25. Cisco ISL (isl)

Field	Field Name	Type
isl.addr	Source or Destination Address	6-byte Hardware (MAC) Address
isl.bpdu	BPDU	Boolean
isl.crc	CRC	Unsigned 32-bit integer
isl.dst	Destination	6-byte Hardware (MAC) Address
isl.dst_route_desc	Destination route descriptor	Unsigned 16-bit integer
isl.esize	Esize	Unsigned 8-bit integer
isl.explorer	Explorer	Boolean
isl.fcs_not_incl	FCS Not Included	Boolean
isl.hsa	HSA	Unsigned 24-bit integer
isl.index	Index	Unsigned 16-bit integer
isl.len	Length	Unsigned 16-bit integer
isl.src	Source	6-byte Hardware (MAC) Address
isl.src_route_desc	Source-route descriptor	Unsigned 16-bit integer

Field	Field Name	Type
isl.src_vlan_id	Source VLAN ID	Unsigned 16-bit integer
isl.type	Type	Unsigned 8-bit integer
isl.user	User	Unsigned 8-bit integer
isl.user_eth	User	Unsigned 8-bit integer
isl.vlan_id	VLAN ID	Unsigned 16-bit integer

Table A-25. Cisco ISL (isl)

A.26. Cisco Interior Gateway Routing Protocol (igrp)

Field	Field Name	Type
igrp.as	Autonomous System	Unsigned 16-bit integer
igrp.update	Update Release	Unsigned 8-bit integer

Table A-26. Cisco Interior Gateway Routing Protocol (igrp)

A.27. Cisco SLARP (slarp)

Field	Field Name	Type
slarp.address	Address	IPv4 address
slarp.mysequence	Outgoing sequence number	Unsigned 32-bit integer
slarp.ptype	Packet type	Unsigned 32-bit integer
slarp.yoursequence	Returned sequence number	Unsigned 32-bit integer

Table A-27. Cisco SLARP (slarp)

A.28. Common Open Policy Service (cops)

Field	Field Name	Type
cops.accttimer.value	Contents: ACCT Timer Value	Unsigned 16-bit integer
cops.c_num	C-Num	Unsigned 8-bit integer
cops.c_type	C-Type	Unsigned 8-bit integer
cops.client_type	Client Type	Unsigned 16-bit integer
cops.context.m_type	M-Type	Unsigned 16-bit integer
cops.context.r_type	R-Type	Unsigned 16-bit integer
cops.decision.cmd	Command-Code	Unsigned 16-bit integer
cops.decision.flags	Flags	Unsigned 16-bit integer
cops.error	Error	Unsigned 16-bit integer
cops.error_sub	Error Sub-code	Unsigned 16-bit integer
cops.flags	Flags	Unsigned 8-bit integer
cops.in-int.ipv4	IPv4 address	IPv4 address
cops.in-int.ipv6	IPv6 address	IPv6 address
cops.in-out-int.ifindex	ifIndex	Unsigned 32-bit integer
cops.integrity.key_id	Contents: Key ID	Unsigned 32-bit integer
cops.integrity.seq_num	Contents: Sequence Number	Unsigned 32-bit integer
cops.katimer.value	Contents: KA Timer Value	Unsigned 16-bit integer
cops.lastpdpaddr.ipv4	IPv4 address	IPv4 address
cops.lastpdpaddr.ipv6	IPv6 address	IPv6 address
cops.msg_len	Message Length	Unsigned 32-bit integer
cops.obj.len	Object Length	Unsigned 32-bit integer
cops.op_code	Op Code	Unsigned 8-bit integer
cops.out-int.ipv4	IPv4 address	IPv4 address
cops.out-int.ipv6	IPv6 address	IPv6 address
cops.pdp.tcp_port	TCP Port Number	Unsigned 32-bit integer
cops.pdprediraddr.ipv4	IPv4 address	IPv4 address
cops.pdprediraddr.ipv6	IPv6 address	IPv6 address
cops.pepid.id	Contents: PEP Id	String
cops.reason	Reason	Unsigned 16-bit integer
cops.reason_sub	Reason Sub-code	Unsigned 16-bit integer
cops.report_type	Contents: Report-Type	Unsigned 16-bit integer
cops.ver_flags	Version and Flags	Unsigned 8-bit integer
cops.version	Version	Unsigned 8-bit integer

Table A-28. Common Open Policy Service (cops)

A.29. Common Unix Printing System (CUPS) Browsing Protocol (cups)

Field	Field Name	Type
cups.ptype	Type	Unsigned 32-bit integer
cups.state	State	Unsigned 8-bit integer

Table A-29. Common Unix Printing System (CUPS) Browsing Protocol (cups)

A.30. DCE RPC (dcerpc)

Field	Field Name	Type
dcerpc.cn_ack_reason	Ack reason	Unsigned 16-bit integer
dcerpc.cn_ack_result	Ack result	Unsigned 16-bit integer
dcerpc.cn_alloc_hint	Alloc hint	Unsigned 32-bit integer
dcerpc.cn_assoc_group	Assoc Group	Unsigned 32-bit integer
dcerpc.cn_auth_len	Auth Length	Unsigned 16-bit integer
dcerpc.cn_bind_if_ver	Interface Ver	Unsigned 16-bit integer
dcerpc.cn_bind_if_ver_minor	Interface Ver Minor	Unsigned 16-bit integer
dcerpc.cn_bind_to_uuid	Interface UUID	String
dcerpc.cn_bind_trans_id	Transfer Syntax	String
dcerpc.cn_bind_trans_ver	Syntax ver	Unsigned 32-bit integer
dcerpc.cn_call_id	Call ID	Unsigned 32-bit integer
dcerpc.cn_cancel_count	Cancel count	Unsigned 8-bit integer
dcerpc.cn_ctx_id	Context ID	Unsigned 16-bit integer
dcerpc.cn_flags	Packet Flags	Unsigned 8-bit integer
dcerpc.cn_flags.cancel_pending	Cancel Pending	Boolean
dcerpc.cn_flags.dne	Did Not Execute	Boolean
dcerpc.cn_flags.first_frag	First Frag	Boolean
dcerpc.cn_flags.last_frag	Last Frag	Boolean
dcerpc.cn_flags.maybe	Maybe	Boolean
dcerpc.cn_flags.mpx	Multiplex	Boolean
dcerpc.cn_flags.object	Object	Boolean

Field	Field Name	Type
dcerpc.cn_flags.reserved	Reserved	Boolean
dcerpc.cn_frag_len	Frag Length	Unsigned 16-bit integer
dcerpc.cn_max_recv	Max Recv Frag	Unsigned 16-bit integer
dcerpc.cn_max_xmit	Max Xmit Frag	Unsigned 16-bit integer
dcerpc.cn_num_ctx_items	Num Ctx Items	Unsigned 8-bit integer
dcerpc.cn_num_results	Num results	Unsigned 8-bit integer
dcerpc.cn_num_trans_items	Num Trans Items	Unsigned 16-bit integer
dcerpc.cn_sec_addr_len	Scndry Addr len	Unsigned 16-bit integer
dcerpc.dg_act_id	Activitiy	String
dcerpc.dg_ahint	Activity Hint	Unsigned 16-bit integer
dcerpc.dg_auth_proto	Auth proto	Unsigned 8-bit integer
dcerpc.dg_flags1	Flags1	Unsigned 8-bit integer
dcerpc.dg_flags1_broadcast	Broadcast	Boolean
dcerpc.dg_flags1_frag	Fragment	Boolean
dcerpc.dg_flags1_idempotent	Idempotent	Boolean
dcerpc.dg_flags1_last_frag	Last Fragment	Boolean
dcerpc.dg_flags1_maybe	Maybe	Boolean
dcerpc.dg_flags1_nofack	No Fack	Boolean
dcerpc.dg_flags1_rsrvd_01	Reserved	Boolean
dcerpc.dg_flags1_rsrvd_80	Reserved	Boolean
dcerpc.dg_flags2	Flags2	Unsigned 8-bit integer
dcerpc.dg_flags2_cancel_pending	Cancel Pending	Boolean
dcerpc.dg_flags2_rsrvd_01	Reserved	Boolean
dcerpc.dg_flags2_rsrvd_04	Reserved	Boolean
dcerpc.dg_flags2_rsrvd_08	Reserved	Boolean
dcerpc.dg_flags2_rsrvd_10	Reserved	Boolean

Field	Field Name	Type
dcerpc.dg_flags2_rsrvd_20	Reserved	Boolean
dcerpc.dg_flags2_rsrvd_40	Reserved	Boolean
dcerpc.dg_flags2_rsrvd_80	Reserved	Boolean
dcerpc.dg_frag_len	Fragment len	Unsigned 16-bit integer
dcerpc.dg_frag_num	Fragment num	Unsigned 16-bit integer
dcerpc.dg_if_id	Interface	String
dcerpc.dg_if_ver	Interface Ver	Unsigned 32-bit integer
dcerpc.dg_ihint	Interface Hint	Unsigned 16-bit integer
dcerpc.dg_seqnum	Sequence num	Unsigned 32-bit integer
dcerpc.dg_serial_hi	Serial High	Unsigned 8-bit integer
dcerpc.dg_serial_lo	Serial Low	Unsigned 8-bit integer
dcerpc.dg_server_boot	Server boot time	Unsigned 32-bit integer
dcerpc.obj_id	Object	String
dcerpc.opnum	Opnum	Unsigned 16-bit integer
dcerpc.pkt_type	Packet type	Unsigned 8-bit integer
dcerpc.ver	Version	Unsigned 8-bit integer
dcerpc.ver_minor	Version (minor)	Unsigned 8-bit integer

Table A-30. DCE RPC (dcerpc)

A.31. DCE/RPC Conversation Manager (conv)

Field	Field Name	Type

Table A-31. DCE/RPC Conversation Manager (conv)

A.32. DCE/RPC Endpoint Mapper (epm)

Field	Field Name	Type

Table A-32. DCE/RPC Endpoint Mapper (epm)

A.33. DCE/RPC Remote Management (mgmt)

Field	Field Name	Type

Table A-33. DCE/RPC Remote Management (mgmt)

A.34. DCOM OXID Resolver (oxid)

Field	Field Name	Type

Table A-34. DCOM OXID Resolver (oxid)

A.35. DCOM Remote Activation (remact)

Field	Field Name	Type

Table A-35. DCOM Remote Activation (remact)

A.36. DEC Spanning Tree Protocol (dec_stp)

Field	Field Name	Type

Table A-36. DEC Spanning Tree Protocol (dec_stp)

A.37. DG Gryphon Protocol (gryphon)

Field	Field Name	Type
gryph.cmd.cmd	Command	Unsigned 8-bit integer
gryph.dest	Destination	Unsigned 8-bit integer
gryph.dstchan	Destination channel	Unsigned 8-bit integer
gryph.src	Source	Unsigned 8-bit integer
gryph.srcchan	Source channel	Unsigned 8-bit integer
gryph.type	Frame type	Unsigned 8-bit integer

Table A-37. DG Gryphon Protocol (gryphon)

A.38. Data (data)

Field	Field Name	Type

Table A-38. Data (data)

A.39. Data Stream Interface (dsi)

Field	Field Name	Type
dsi.code	Code	Unsigned 32-bit integer
dsi.command	Command	Unsigned 8-bit integer
dsi.flags	Flags	Unsigned 8-bit integer
dsi.length	Length	Unsigned 32-bit integer
dsi.requestid	Request ID	Unsigned 16-bit integer
dsi.reserved	Reserved	Unsigned 32-bit integer

Table A-39. Data Stream Interface (dsi)

A.40. Datagram Delivery Protocol (ddp)

Field	Field Name	Type
ddp.checksum	Checksum	Unsigned 16-bit integer
ddp.dst.net	Destination Net	Unsigned 16-bit integer
ddp.dst.node	Destination Node	Unsigned 8-bit integer
ddp.dst.socket	Destination Socket	Unsigned 8-bit integer
ddp.hopcount	Hop count	Unsigned 8-bit integer
ddp.len	Datagram length	Unsigned 16-bit integer
ddp.src.net	Source Net	Unsigned 16-bit integer
ddp.src.node	Source Node	Unsigned 8-bit integer
ddp.src.socket	Source Socket	Unsigned 8-bit integer
ddp.type	Protocol type	Unsigned 8-bit integer

Table A-40. Datagram Delivery Protocol (ddp)

A.41. Diameter Protocol (diameter)

Field	Field Name	Type
diameter.avp.code	AVP Code	Unsigned 32-bit integer
diameter.avp.data.bytes	AVP Data	Byte array
diameter.avp.data.int32	AVP Data	Signed 32-bit integer
diameter.avp.data.string	AVP Data	String
diameter.avp.data.time	AVP Data	Date/Time stamp
diameter.avp.data.uint32	AVP Data	Unsigned 32-bit integer
diameter.avp.data.v4addr	AVP Data	IPv4 address
diameter.avp.data.v6addr	AVP Data	IPv6 address
diameter.avp.flags	AVP Flags	Unsigned 8-bit integer
diameter.avp.length	AVP length	Unsigned 16-bit integer
diameter.avp.reserved	AVP Reserved	Unsigned 8-bit integer
diameter.avp.vendorId	AVP Vendor Id	Unsigned 32-bit integer
diameter.code	Command Code	Unsigned 32-bit integer
diameter.endtoendid	End-to-End Identifier	Unsigned 32-bit integer
diameter.flags	Flags	Unsigned 8-bit integer
diameter.hopbyhopid	Hop-by-Hop Identifier	Unsigned 32-bit integer

Field	Field Name	Type
diameter.length	Length	Unsigned 16-bit integer
diameter.reserved	Reserved	Unsigned 8-bit integer
diameter.vendorId	VendorId	Unsigned 32-bit integer
diameter.version	Version	Unsigned 8-bit integer

Table A-41. Diameter Protocol (diameter)

A.42. Distance Vector Multicast Routing Protocol (dvmrp)

Field	Field Name	Type
dvmrp.afi	Address Family	Unsigned 8-bit integer
dvmrp.cap.genid	Genid	Boolean
dvmrp.cap.leaf	Leaf	Boolean
dvmrp.cap.mtrace	Mtrace	Boolean
dvmrp.cap.netmask	Netmask	Boolean
dvmrp.cap.prune	Prune	Boolean
dvmrp.cap.snmp	SNMP	Boolean
dvmrp.capabilities	Capabilities	No value
dvmrp.checksum	Checksum	Unsigned 16-bit integer
dvmrp.checksum_bad	Bad Checksum	Boolean
dvmrp.command	Command	Unsigned 8-bit integer
dvmrp.commands	Commands	No value
dvmrp.count	Count	Unsigned 8-bit integer
dvmrp.dest_unreach	Destination Unreachable	Boolean
dvmrp.genid	Generation ID	Unsigned 32-bit integer
dvmrp.hold	Hold Time	Unsigned 32-bit integer
dvmrp.infinity	Infinity	Unsigned 8-bit integer
dvmrp.lifetime	Prune lifetime	Unsigned 32-bit integer
dvmrp.maj_ver	Major Version	Unsigned 8-bit integer
dvmrp.metric	Metric	Unsigned 8-bit integer
dvmrp.min_ver	Minor Version	Unsigned 8-bit integer
dvmrp.route	Route	No value
dvmrp.split_horiz	Split Horizon	Boolean

Field	Field Name	Type
dvmrp.type	Type	Unsigned 8-bit integer
dvmrp.v1.code	Code	Unsigned 8-bit integer
dvmrp.v3.code	Code	Unsigned 8-bit integer
dvmrp.version	DVMRP Version	Unsigned 8-bit integer
igmp.daddr	Dest Addr	IPv4 address
igmp.maddr	Multicast Addr	IPv4 address
igmp.naddr	Neighbor Addr	IPv4 address
igmp.neighbor	Neighbor Addr	IPv4 address
igmp.netmask	Netmask	IPv4 address
igmp.saddr	Source Addr	IPv4 address

Table A-42. Distance Vector Multicast Routing Protocol (dvmrp)

A.43. Domain Name Service (dns)

Field	Field Name	Type
dns.count.add_rr	Additional RRs	Unsigned 16-bit integer
dns.count.answers	Answer RRs	Unsigned 16-bit integer
dns.count.auth_rr	Authority RRs	Unsigned 16-bit integer
dns.count.queries	Questions	Unsigned 16-bit integer
dns.flags	Flags	Unsigned 16-bit integer
dns.id	Transaction ID	Unsigned 16-bit integer
dns.length	Length	Unsigned 16-bit integer
dns.query	Query	Boolean
dns.response	Response	Boolean

Table A-43. Domain Name Service (dns)

A.44. Dynamic DNS Tools Protocol (ddtp)

Field	Field Name	Type
ddtp.encrypt	Encryption	Unsigned 32-bit integer
ddtp.hostid	Hostid	Unsigned 32-bit integer

Field	Field Name	Type
ddtp.ipaddr	IP address	IPv4 address
ddtp.msgtype	Message type	Unsigned 32-bit integer
ddtp.opcode	Opcode	Unsigned 32-bit integer
ddtp.status	Status	Unsigned 32-bit integer
ddtp.version	Version	Unsigned 32-bit integer

Table A-44. Dynamic DNS Tools Protocol (ddtp)

A.45. Encapsulating Security Payload (esp)

Field	Field Name	Type
esp.sequence	Sequence	Unsigned 32-bit integer
esp.spi	SPI	Unsigned 32-bit integer

Table A-45. Encapsulating Security Payload (esp)

A.46. Enhanced Interior Gateway Routing Protocol (eigrp)

Field	Field Name	Type
eigrp.as	Autonomous System	Unsigned 16-bit integer
eigrp.opcode	Opcode	Unsigned 8-bit integer
eigrp.tlv	Entry	Unsigned 16-bit integer

Table A-46. Enhanced Interior Gateway Routing Protocol (eigrp)

A.47. Ethernet (eth)

Field	Field Name	Type
eth.addr	Source or Destination Address	6-byte Hardware (MAC) Address

Field	Field Name	Type
eth.dst	Destination	6-byte Hardware (MAC) Address
eth.len	Length	Unsigned 16-bit integer
eth.src	Source	6-byte Hardware (MAC) Address
eth.trailer	Trailer	Byte array
eth.type	Type	Unsigned 16-bit integer

Table A-47. Ethernet (eth)

A.48. FTP Data (ftp-data)

Field	Field Name	Type

Table A-48. FTP Data (ftp-data)

A.49. Fiber Distributed Data Interface (fddi)

Field	Field Name	Type
fddi.addr	Source or Destination Address	6-byte Hardware (MAC) Address
fddi.dst	Destination	6-byte Hardware (MAC) Address
fddi.fc	Frame Control	Unsigned 8-bit integer
fddi.fc.clf	Class/Length/Format	Unsigned 8-bit integer
fddi.fc.mac_subtype	MAC Subtype	Unsigned 8-bit integer
fddi.fc.prio	Priority	Unsigned 8-bit integer
fddi.fc.smt_subtype	SMT Subtype	Unsigned 8-bit integer
fddi.src	Source	6-byte Hardware (MAC) Address

Table A-49. Fiber Distributed Data Interface (fddi)

A.50. File Transfer Protocol (FTP) (ftp)

Field	Field Name	Type
ftp.reponse.data	Response data	String
ftp.request	Request	Boolean
ftp.request.command	Request command	String
ftp.request.data	Request data	String
ftp.response	Response	Boolean
ftp.response.code	Response code	Unsigned 8-bit integer

Table A-50. File Transfer Protocol (FTP) (ftp)

A.51. Frame (frame)

Field	Field Name	Type
frame.cap_len	Capture Frame Length	Unsigned 32-bit integer
frame.number	Frame Number	Unsigned 32-bit integer
frame.p2p_dir	Point-to-Point Direction	Unsigned 8-bit integer
frame.pkt_len	Total Frame Length	Unsigned 32-bit integer
frame.time	Arrival Time	Date/Time stamp
frame.time_delta	Time delta from previous packet	Time duration
frame.time_relative	Time relative to first packet	Time duration

Table A-51. Frame (frame)

A.52. Frame Relay (fr)

Field	Field Name	Type
fr.becn	BECN	Boolean
fr.chdlctype	Type	Unsigned 16-bit integer
fr.cr	CR	Boolean
fr.dc	DC	Boolean

Field	Field Name	Type
fr.de	DE	Boolean
fr.dlci	DLCI	Unsigned 16-bit integer
fr.ea	EA	Boolean
fr.fecn	FECN	Boolean
fr.nlpid	NLPID	Unsigned 8-bit integer
fr.snap.oui	Organization Code	Unsigned 24-bit integer
fr.snap.pid	Protocol ID	Unsigned 16-bit integer
fr.snaptype	Type	Unsigned 16-bit integer

Table A-52. Frame Relay (fr)

A.53. GARP VLAN Registration Protocol (gvrp)

Field	Field Name	Type
garp.attribute_event	Event	Unsigned 8-bit integer
garp.attribute_length	Length	Unsigned 8-bit integer
garp.attribute_type	Type	Unsigned 8-bit integer
garp.attribute_value	Value	Unsigned 16-bit integer
garp.protocol_id	Protocol ID	Unsigned 16-bit integer

Table A-53. GARP VLAN Registration Protocol (gvrp)

A.54. GPRS Tunneling Protocol (gtp)

Field	Field Name	Type
gtp.ext	Extension header	Unsigned 8-bit integer
gtp.ext.apn	APN	String
gtp.ext.cause	Cause	Unsigned 8-bit integer
gtp.ext.chrg_addr	CG address	IPv4 address
gtp.ext.chrg_id	Charging ID	Unsigned 32-bit integer
gtp.ext.ext_id	Ext id	Unsigned 16-bit integer
gtp.ext.ext_val	Ext val	String
gtp.ext.flow_ii	Downlink flow label data	Unsigned 16-bit integer

Field	Field Name	Type
gtp.ext.flow_ii_nsapi	NSAPI	Unsigned 8-bit integer
gtp.ext.flow_label	Flow label	Unsigned 16-bit integer
gtp.ext.flow_sig	Flow label signature	Unsigned 16-bit integer
gtp.ext.gsn_addr	GSN address	IPv4 address
gtp.ext.imsi	IMSI	String
gtp.ext.lac	LAC	Unsigned 16-bit integer
gtp.ext.map	Ext type	Unsigned 8-bit integer
gtp.ext.mcc	MCC	Unsigned 16-bit integer
gtp.ext.mnc	MNC	Unsigned 8-bit integer
gtp.ext.ms	MS validated	Boolean
gtp.ext.msisdn	MSISDN	String
gtp.ext.node_addr	Node address	IPv4 address
gtp.ext.proto_conf	Protocol configuration	String
gtp.ext.ptmsi	P-TMSI	Unsigned 32-bit integer
gtp.ext.ptmsi_sig	P-TMSI signature	Unsigned 24-bit integer
gtp.ext.qos_delay	QoS delay	Unsigned 8-bit integer
gtp.ext.qos_mean	QoS mean	Unsigned 8-bit integer
gtp.ext.qos_peak	QoS peak	Unsigned 8-bit integer
gtp.ext.qos_precedence	QoS precedence	Unsigned 8-bit integer
gtp.ext.qos_reliabilty	QoS reliability	Unsigned 8-bit integer
gtp.ext.rac	RAC	Unsigned 8-bit integer
gtp.ext.recover	Restart counter	Unsigned 8-bit integer
gtp.ext.reorder	Reordering required	Boolean
gtp.ext.sel_mode	Selection mode	Unsigned 8-bit integer
gtp.ext.tlli	TLLI	Unsigned 32-bit integer
gtp.ext.tr_comm	Transfer command	Unsigned 8-bit integer
gtp.ext.unknown	Unknown data (length)	Unsigned 16-bit integer
gtp.ext.user_addr	End user address	IPv4 address
gtp.ext.user_addr_pdp_org	PDP type organization	Unsigned 8-bit integer
gtp.ext.user_addr_pdp_type	PDP type	Unsigned 8-bit integer
gtp.flags	Flags	Unsigned 8-bit integer
gtp.flags.payload_type	Payload Type	Unsigned 8-bit integer
gtp.flags.snn	Is seq number	Unsigned 8-bit integer
gtp.flags.spare	Reserved	Unsigned 8-bit integer
gtp.flags.version	Version	Unsigned 8-bit integer

Field	Field Name	Type
gtp.flow_label	Flow label	Unsigned 16-bit integer
gtp.length	Length	Unsigned 16-bit integer
gtp.message_type	Message type	Unsigned 8-bit integer
gtp.seq_number	Sequence number	Unsigned 16-bit integer
gtp.sndcp_number	SNDCP N-PDU LLC Number	Unsigned 8-bit integer
gtp.tid	Tunnel ID	String

Table A-54. GPRS Tunneling Protocol (gtp)

A.55. General Inter-ORB Protocol (giop)

Field	Field Name	Type
giop.TCKind	TypeCode enum	Unsigned 32-bit integer
giop.endianess	Endianess	Unsigned 8-bit integer
giop.iiop.host	IIOP::Profile_host	String
giop.iiop.port	IIOP::Profile_port	Unsigned 16-bit integer
giop.iiop.scid	SCID	Unsigned 32-bit integer
giop.iiop.vscid	VSCID	Unsigned 32-bit integer
giop.iiop_vmaj	IIOP Major Version	Unsigned 8-bit integer
giop.iiop_vmin	IIOP Minor Version	Unsigned 8-bit integer
giop.iioptag	IIOP Component TAG	Unsigned 32-bit integer
giop.iortag	IOR Profile TAG	Unsigned 8-bit integer
giop.len	Message size	Unsigned 32-bit integer
giop.profid	Profile ID	Unsigned 32-bit integer
giop.repoid	Repository ID	String
giop.seqlen	Sequence Length	Unsigned 32-bit integer
giop.strlen	String Length	Unsigned 32-bit integer
giop.tcValueModifier	ValueModifier	Signed 16-bit integer
giop.tcVisibility	Visibility	Signed 16-bit integer
giop.tcboolean	TypeCode boolean data	Boolean
giop.tcchar	TypeCode char data	Unsigned 8-bit integer
giop.tccount	TypeCode count	Unsigned 32-bit integer
giop.tcdefault_used	default_used	Signed 32-bit integer
giop.tcdigits	Digits	Unsigned 16-bit integer

Field	Field Name	Type
giop.tcdouble	TypeCode double data	Double-precision floating point
giop.tcenumdata	TypeCode enum data	Unsigned 32-bit integer
giop.tcfloat	TypeCode float data	Double-precision floating point
giop.tclength	Length	Unsigned 32-bit integer
giop.tclongdata	TypeCode long data	Signed 32-bit integer
giop.tcmaxlen	Maximum length	Unsigned 32-bit integer
giop.tcmemname	TypeCode member name	String
giop.tcname	TypeCode name	String
giop.tcoctet	TypeCode octet data	Unsigned 8-bit integer
giop.tcscale	Scale	Signed 16-bit integer
giop.tcshortdata	TypeCode short data	Signed 16-bit integer
giop.tcstring	TypeCode string data	String
giop.tculongdata	TypeCode ulong data	Unsigned 32-bit integer
giop.tcushortdata	TypeCode ushort data	Unsigned 16-bit integer
giop.type	Message type	Unsigned 8-bit integer
giop.typeid	IOR::type_id	String

Table A-55. General Inter-ORB Protocol (giop)

A.56. Generic Routing Encapsulation (gre)

Field	Field Name	Type
gre.proto	Protocol Type	Unsigned 16-bit integer

Table A-56. Generic Routing Encapsulation (gre)

A.57. Gnutella Protocol (gnutella)

Field	Field Name	Type
gnutella.header	Descriptor Header	No value
gnutella.header.hops	Hops	Unsigned 8-bit integer
gnutella.header.id	ID	Byte array

Field	Field Name	Type
gnutella.header.payload	Payload	Unsigned 8-bit integer
gnutella.header.size	Length	Unsigned 8-bit integer
gnutella.header.ttl	TTL	Unsigned 8-bit integer
gnutella.pong.files	Files Shared	Unsigned 32-bit integer
gnutella.pong.ip	IP	IPv4 address
gnutella.pong.kbytes	KBytes Shared	Unsigned 32-bit integer
gnutella.pong.payload	Pong	No value
gnutella.pong.port	Port	Unsigned 16-bit integer
gnutella.push.index	Index	Unsigned 32-bit integer
gnutella.push.ip	IP	IPv4 address
gnutella.push.payload	Push	No value
gnutella.push.port	Port	Unsigned 16-bit integer
gnutella.push.servent_id	Servent ID	Byte array
gnutella.query.min_speed	Min Speed	Unsigned 32-bit integer
gnutella.query.payload	Query	No value
gnutella.query.search	Search	String
gnutella.queryhit.count	Count	Unsigned 8-bit integer
gnutella.queryhit.extra	Extra	Byte array
gnutella.queryhit.hit	Hit	No value
gnutella.queryhit.hit.extra	Extra	Byte array
gnutella.queryhit.hit.index	Index	Unsigned 32-bit integer
gnutella.queryhit.hit.name	Name	String
gnutella.queryhit.hit.size	Size	Unsigned 32-bit integer
gnutella.queryhit.ip	IP	IPv4 address
gnutella.queryhit.payload	QueryHit	No value
gnutella.queryhit.port	Port	Unsigned 16-bit integer
gnutella.queryhit.servent_id	Servent ID	Byte array
gnutella.queryhit.speed	Speed	Unsigned 32-bit integer
gnutella.stream	Gnutella Upload / Download Stream	No value
gnutella.truncated	Truncated Frame	No value

Table A-57. Gnutella Protocol (gnutella)

A.58. Hummingbird NFS Daemon (hclnfsd)

Field	Field Name	Type
hclnfsd.access	Access	Unsigned 32-bit integer
hclnfsd.cookie	Cookie	Unsigned 32-bit integer
hclnfsd.copies	Copies	Unsigned 32-bit integer
hclnfsd.device	Device	String
hclnfsd.exclusive	Exclusive	Unsigned 32-bit integer
hclnfsd.fileext	File Extension	Unsigned 32-bit integer
hclnfsd.filename	Filename	String
hclnfsd.gid	GID	Unsigned 32-bit integer
hclnfsd.group	Group	String
hclnfsd.host_ip	Host IP	IPv4 address
hclnfsd.hostname	Hostname	String
hclnfsd.jobstatus	Job Status	Unsigned 32-bit integer
hclnfsd.length	Length	Unsigned 32-bit integer
hclnfsd.lockname	Lockname	String
hclnfsd.lockowner	Lockowner	Byte array
hclnfsd.logintext	Login Text	String
hclnfsd.mode	Mode	Unsigned 32-bit integer
hclnfsd.npp	Number of Physical Printers	Unsigned 32-bit integer
hclnfsd.offset	Offset	Unsigned 32-bit integer
hclnfsd.pqn	Print Queue Number	Unsigned 32-bit integer
hclnfsd.printername	Printer Name	String
hclnfsd.printparameters	Print Parameters	String
hclnfsd.printqueuecomment	Comment	String
hclnfsd.printqueuename	Name	String
hclnfsd.queuestatus	Queue Status	Unsigned 32-bit integer
hclnfsd.request_type	Request Type	Unsigned 32-bit integer
hclnfsd.sequence	Sequence	Unsigned 32-bit integer
hclnfsd.server_ip	Server IP	IPv4 address
hclnfsd.size	Size	Unsigned 32-bit integer
hclnfsd.status	Status	Unsigned 32-bit integer
hclnfsd.timesubmitted	Time Submitted	Unsigned 32-bit integer
hclnfsd.uid	UID	Unsigned 32-bit integer

Field	Field Name	Type
hclnfsd.unknown_data	Unknown	Byte array
hclnfsd.username	Username	String

Table A-58. Hummingbird NFS Daemon (hclnfsd)

A.59. Hypertext Transfer Protocol (http)

Field	Field Name	Type
http.notification	Notification	Boolean
http.request	Request	Boolean
http.response	Response	Boolean

Table A-59. Hypertext Transfer Protocol (http)

A.60. ICQ Protocol (icq)

Field	Field Name	Type
icq.checkcode	Checkcode	Unsigned 32-bit integer
icq.client_cmd	Client command	Unsigned 16-bit integer
icq.decode	Decode	String
icq.server_cmd	Server command	Unsigned 16-bit integer
icq.sessionid	Session ID	Unsigned 32-bit integer
icq.type	Type	Unsigned 16-bit integer
icq.uin	UIN	Unsigned 32-bit integer

Table A-60. ICQ Protocol (icq)

A.61. IEEE 802.11 wireless LAN (wlan)

Field	Field Name	Type
wlan.aid	Association ID	Unsigned 16-bit integer

Field	Field Name	Type
wlan.bssid	BSS Id	6-byte Hardware (MAC) Address
wlan.da	Destination address	6-byte Hardware (MAC) Address
wlan.duration	Duration	Unsigned 16-bit integer
wlan.fc	Frame Control Field	Unsigned 16-bit integer
wlan.fc.ds	DS status	Unsigned 8-bit integer
wlan.fc.frag	Fragments	Boolean
wlan.fc.fromds	From DS	Boolean
wlan.fc.moredata	More Data	Boolean
wlan.fc.order	Order flag	Boolean
wlan.fc.pwrmgt	PWR MGT	Boolean
wlan.fc.retry	Retry	Boolean
wlan.fc.subtype	Subtype	Unsigned 8-bit integer
wlan.fc.tods	To DS	Boolean
wlan.fc.type	Type	Unsigned 8-bit integer
wlan.fc.type_subtype	Type/Subtype	Unsigned 16-bit integer
wlan.fc.version	Version	Unsigned 8-bit integer
wlan.fc.wep	WEP flag	Boolean
wlan.fcs	Frame Check Sequence (not verified)	Unsigned 32-bit integer
wlan.flags	Protocol Flags	Unsigned 8-bit integer
wlan.frag	Fragment number	Unsigned 16-bit integer
wlan.ra	Receiver address	6-byte Hardware (MAC) Address
wlan.sa	Source address	6-byte Hardware (MAC) Address
wlan.seq	Sequence number	Unsigned 16-bit integer
wlan.ta	Transmitter address	6-byte Hardware (MAC) Address
wlan.wep.crc	WEP CRC (not verified)	Unsigned 32-bit integer
wlan.wep.iv	Initialization Vector	Unsigned 24-bit integer
wlan.wep.key	Key	Unsigned 8-bit integer

Table A-61. IEEE 802.11 wireless LAN (wlan)

A.62. IEEE 802.11 wireless LAN management

frame (wlan_mgt)

Field	Field Name	Type
wlan_mgt.fixed.aid	Association ID	Unsigned 16-bit integer
wlan_mgt.fixed.all	Fixed parameters	Unsigned 16-bit integer
wlan_mgt.fixed.auth.alg	Authentication Algorithm	Unsigned 16-bit integer
wlan_mgt.fixed.auth_seq	Authentication SEQ	Unsigned 16-bit integer
wlan_mgt.fixed.beacon	Beacon Interval	Double-precision floating point
wlan_mgt.fixed.capabilities	Capabilities	Unsigned 16-bit integer
wlan_mgt.fixed.capabilities.agility	Channel Agility	Boolean
wlan_mgt.fixed.capabilities.cfpoll.ap	CF Pollable capabilities	Unsigned 16-bit integer
wlan_mgt.fixed.capabilities.cfpoll.sta	CF Poll participation capabilities	Unsigned 16-bit integer
wlan_mgt.fixed.capabilities.ess	ESS capabilities	Boolean
wlan_mgt.fixed.capabilities.ibss	IBSS status	Boolean
wlan_mgt.fixed.capabilities.pbcc	PBCC	Boolean
wlan_mgt.fixed.capabilities.preamble	Short Preamble	Boolean
wlan_mgt.fixed.capabilities.privacy	Privacy	Boolean
wlan_mgt.fixed.current_ap	Current AP	6-byte Hardware (MAC) Address
wlan_mgt.fixed.listen_ival	Listen Interval	Unsigned 16-bit integer
wlan_mgt.fixed.reason_code	Reason code	Unsigned 16-bit integer
wlan_mgt.fixed.status_code	Status code	Unsigned 16-bit integer
wlan_mgt.fixed.timestamp	Timestamp	String

Field	Field Name	Type
wlan_mgt.tag.interpretation	Tag interpretation	String
wlan_mgt.tag.length	Tag length	Unsigned 16-bit integer
wlan_mgt.tag.number	Tag	Unsigned 16-bit integer
wlan_mgt.tagged.all	Tagged parameters	Unsigned 16-bit integer

Table A-62. IEEE 802.11 wireless LAN management frame (wlan_mgt)

A.63. ILMI (ilmi)

Field	Field Name	Type

Table A-63. ILMI (ilmi)

A.64. IP Payload Compression (ipcomp)

Field	Field Name	Type
ipcomp.cpi	CPI	Unsigned 16-bit integer
ipcomp.flags	Flags	Unsigned 8-bit integer

Table A-64. IP Payload Compression (ipcomp)

A.65. IPX Message (ipxmsg)

Field	Field Name	Type
ipxmsg.conn	Connection Number	Unsigned 8-bit integer
ipxmsg.sigchar	Signature Char	Unsigned 8-bit integer

Table A-65. IPX Message (ipxmsg)

A.66. IPX Routing Information Protocol (ipxrip)

Field	Field Name	Type
ipxrip.request	Request	Boolean
ipxrip.response	Response	Boolean

Table A-66. IPX Routing Information Protocol (ipxrip)

A.67. ISDN Q.921-User Adaptation Layer (iua)

Field	Field Name	Type
hf.iua.one_bit	One bit	Boolean
hf.iua.sapi	SAPI	Unsigned 8-bit integer
hf.iua.spare_bit	Spare bit	Boolean
hf.iua.tei	TEI	Unsigned 8-bit integer
hf.iua.zero_bit	Zero bit	Boolean
iua.asp_reason	Reason	Unsigned 32-bit integer
iua.error_code	Error code	Unsigned 32-bit integer
iua.info_string	Info string	String
iua.int_interface_identifier	Integer interface identifier	Unsigned 32-bit integer
iua.interface_range_end	End	Unsigned 32-bit integer
iua.interface_range_start	Start	Unsigned 32-bit integer
iua.message_class	Message class	Unsigned 8-bit integer
iua.message_length	Message length	Unsigned 32-bit integer
iua.message_type	Message Type	Unsigned 8-bit integer
iua.parameter_length	Parameter length	Unsigned 16-bit integer
iua.parameter_tag	Parameter Tag	Unsigned 16-bit integer
iua.release_reason	Reason	Unsigned 32-bit integer
iua.reserved	Reserved	Unsigned 8-bit integer
iua.status_identification	Status identification	Unsigned 16-bit integer
iua.status_type	Status type	Unsigned 16-bit integer
iua.tei_status	TEI status	Unsigned 32-bit integer
iua.text_interface_identifier	Text interface identifier	String

Field	Field Name	Type
iua.traffic_mode_type	Traffic mode type	Unsigned 32-bit integer
iua.version	Version	Unsigned 8-bit integer

Table A-67. ISDN Q.921-User Adaptation Layer (iua)

A.68. ISDN User Part (isup)

Field	Field Name	Type
isup.access_delivery_ind	Access delivery indicator	Boolean
isup.address_presentation_restricted_indicator	Address presentation restricted indicator	Unsigned 8-bit integer
isup.automatic_congestion_level	Automatic congestion level	Unsigned 8-bit integer
isup.backw_call_echo_control_device_ind	Echo Control Device Indicator	Boolean
isup.backw_call_end_to_end_info_ind	End-to-end information indicator	Boolean
isup.backw_call_end_to_end_method_ind	End-to-end method indicator	Unsigned 16-bit integer
isup.backw_call_holding_ind	Holding indicator	Boolean
isup.backw_call_interworking_ind	Interworking indicator	Boolean
isup.backw_call_isdn_access_ind	ISDN access indicator	Boolean
isup.backw_call_isdn_user_part_ind	ISDN user part indicator	Boolean
isup.backw_call_sccp_method_ind	SCCP method indicator	Unsigned 16-bit integer
isup.call_diversion_may_occur_ind	Call diversion may occur indicator	Boolean
isup.call_processing_state	Call processing state	Unsigned 8-bit integer
isup.call_to_be_diverted_ind	Call to be diverted indicator	Unsigned 8-bit integer
isup.call_to_be_offered_ind	Call to be offered indicator	Unsigned 8-bit integer

Field	Field Name	Type
isup.called_party_even_address_signal_digit	Address signal digit	Unsigned 8-bit integer
isup.called_party_nature_of_address_indicator	Nature of address indicator	Unsigned 8-bit integer
isup.called_party_odd_address_signal_digit	Address signal digit	Unsigned 8-bit integer
isup.called_partys_category_indicator	Called party's category indicator	Unsigned 16-bit integer
isup.called_partys_status_indicator	Called party's status indicator	Unsigned 16-bit integer
isup.calling_party_address_request_indicator	Calling party address request indicator	Boolean
isup.calling_party_address_response_indicator	Calling party address response indicator	Unsigned 16-bit integer
isup.calling_party_even_address_signal_digit	Address signal digit	Unsigned 8-bit integer
isup.calling_party_nature_of_address_indicator	Nature of address indicator	Unsigned 8-bit integer
isup.calling_party_odd_address_signal_digit	Address signal digit	Unsigned 8-bit integer
isup.calling_partys_category	Calling Party's category	Unsigned 8-bit integer
isup.calling_partys_category_request_indicator	Calling party's category request indicator	Boolean
isup.calling_partys_category_response_indicator	Calling party's category response indicator	Boolean
isup.cgs_message_type	Circuit group supervision message type	Unsigned 8-bit integer
isup.charge_indicator	Charge indicator	Unsigned 16-bit integer
isup.charge_information_request_indicator	Charge information request indicator	Boolean
isup.charge_information_response_indicator	Charge information response indicator	Boolean
isup.cic	CIC	Unsigned 16-bit integer
isup.clg_call_ind	Closed user group call indicator	Unsigned 8-bit integer
isup.conference_acceptance_ind	Conference acceptance indicator	Unsigned 8-bit integer

Field	Field Name	Type
isup.connected_line_identity_request_ind	Connected line identity request indicator	Boolean
isup.continuity_check_indicator	Continuity Check Indicator	Unsigned 8-bit integer
isup.continuity_indicator	Continuity indicator	Boolean
isup.echo_control_device_indicator	Echo Control Device Indicator	Boolean
isup.event_ind	Event indicator	Unsigned 8-bit integer
isup.event_presentatiation_restr_ind	Event presentation restricted indicator	Boolean
isup.extension_ind	Extension indicator	Boolean
isup.forw_call_end_to_end_info_ind	End-to-end information indicator	Boolean
isup.forw_call_end_to_end_method_ind	End-to-end method indicator	Unsigned 16-bit integer
isup.forw_call_interworking_ind	Interworking indicator	Boolean
isup.forw_call_isdn_access_ind	ISDN access indicator	Boolean
isup.forw_call_isdn_user_part_ind	ISDN user part indicator	Boolean
isup.forw_call_natnl_inatnl_call_ind	National/international call indicator	Boolean
isup.forw_call_preferences_ind	ISDN user part preference indicator	Unsigned 16-bit integer
isup.forw_call_sccp_method_ind	SCCP method indicator	Unsigned 16-bit integer
isup.hold_provided_indicator	Hold provided indicator	Boolean
isup.hw_blocking_state	HW blocking state	Unsigned 8-bit integer
isup.inband_information_ind	In-band information indicator	Boolean
isup.info_req_holding_indicator	Holding indicator	Boolean
isup.inn_indicator	INN indicator	Boolean
isup.isdn_odd_even_indicator	Odd/even indicator	Boolean

Field	Field Name	Type
isup.loop_prevention_response_ind	Response indicator	Unsigned 8-bit integer
isup.malicious_call_ident_request_ind	Malicious call identification request indicator (ISUP'88)	Boolean
isup.mandatory_variable_parameter_pointer	Pointer to Parameter	Unsigned 8-bit integer
isup.map_type	Map Type	Unsigned 8-bit integer
isup.message_type	Message Type	Unsigned 8-bit integer
isup.mlpp_user	MLPP user indicator	Boolean
isup.mtc_blocking_state	Maintenance blocking state	Unsigned 8-bit integer
isup.network_identification_plan	Network identification plan	Unsigned 8-bit integer
isup.ni_indicator	NI indicator	Boolean
isup.numbering_plan_indicator	Numbering plan indicator	Unsigned 8-bit integer
isup.optional_parameter_part_pointer	Pointer to optional parameter part	Unsigned 8-bit integer
isup.original_redirection_reason	Original redirection reason	Unsigned 16-bit integer
isup.parameter_length	Parameter Length	Unsigned 8-bit integer
isup.parameter_type	Parameter Type	Unsigned 8-bit integer
isup.range_indicator	Range indicator	Unsigned 8-bit integer
isup.redirecting_ind	Redirection indicator	Unsigned 16-bit integer
isup.redirection_counter	Redirection counter	Unsigned 16-bit integer
isup.redirection_reason	Redirection reason	Unsigned 16-bit integer
isup.satellite_indicator	Satellite Indicator	Unsigned 8-bit integer
isup.screening_indicator	Screening indicator	Unsigned 8-bit integer
isup.screening_indicator_enhanced	Screening indicator	Unsigned 8-bit integer
isup.simple_segmentation_ind	Simple segmentation indicator	Boolean
isup.solicided_indicator	Solicited indicator	Boolean
isup.suspend_resume_indicator	Suspend/Resume indicator	Boolean
isup.temporary_alternative_routing_ind	Temporary alternative routing indicator	Boolean

Field	Field Name	Type
isup.transmission_medium_requirement	Transmission medium requirement	Unsigned 8-bit integer
isup.transmission_medium_requirement_prime	Transmission medium requirement prime	Unsigned 8-bit integer
isup.type_of_network_identification	Type of network identification	Unsigned 8-bit integer

Table A-68. ISDN User Part (isup)

A.69. ISIS HELLO (isis_hello)

Field	Field Name	Type
isis_hello.circuite_type	Circuit type	Unsigned 8-bit integer
isis_hello.clv_ipv4_int_addr	IPv4 interface address	IPv4 address
isis_hello.clv_ipv6_int_addr	IPv6 interface address	IPv6 address
isis_hello.clv_ptp_adj	point-to-point Adjacency	Unsigned 8-bit integer
isis_hello.holding_timer	Holding timer	Unsigned 16-bit integer
isis_hello.lan_id	SystemID{ Designated IS }	Byte array
isis_hello.local_circuit_id	Local circuit ID	Unsigned 8-bit integer
isis_hello.pdu_length	PDU length	Unsigned 16-bit integer
isis_hello.priority	Priority	Unsigned 8-bit integer
isis_hello.source_id	SystemID{ Sender of PDU }	Byte array

Table A-69. ISIS HELLO (isis_hello)

A.70. ISO 10589 ISIS Complete Sequence Numbers Protocol Data Unit (isis_csnp)

Field	Field Name	Type
isis_csnp.pdu_length	PDU length	Unsigned 16-bit integer

Table A-70. ISO 10589 ISIS Complete Sequence Numbers Protocol Data Unit (isis_csnp)

A.71. ISO 10589 ISIS InTRA Domain Routeing Information Exchange Protocol (isis)

Field	Field Name	Type
isis.irpd	Intra Domain Routing Protocol Discriminator	Unsigned 8-bit integer
isis.len	PDU Header Length	Unsigned 8-bit integer
isis.max_area_adr	Max.AREAs: (0==3)	Unsigned 8-bit integer
isis.reserved	Reserved (==0)	Unsigned 8-bit integer
isis.sysid_len	System ID Length	Unsigned 8-bit integer
isis.type	PDU Type	Unsigned 8-bit integer
isis.version	Version (==1)	Unsigned 8-bit integer
isis.version2	Version2 (==1)	Unsigned 8-bit integer

Table A-71. ISO 10589 ISIS InTRA Domain Routeing Information Exchange Protocol (isis)

A.72. ISO 10589 ISIS Link State Protocol Data Unit (isis_lsp)

Field	Field Name	Type
isis_lsp.checksum	Checksum	Unsigned 16-bit integer
isis_lsp.clv_ipv4_int_addr	IPv4 interface address	IPv4 address
isis_lsp.clv_ipv6_int_addr	IPv6 interface address	IPv6 address
isis_lsp.clv_te_router_id	Traffic Engineering Router ID	IPv4 address
isis_lsp.pdu_length	PDU length	Unsigned 16-bit integer
isis_lsp.remaining_life	Remaining lifetime	Unsigned 16-bit integer

Field	Field Name	Type
isis_lsp.sequence_number	Sequence number	Unsigned 32-bit integer

Table A-72. ISO 10589 ISIS Link State Protocol Data Unit (isis_lsp)

A.73. ISO 10589 ISIS Partial Sequence Numbers Protocol Data Unit (isis_psnp)

Field	Field Name	Type
isis_psnp.pdu_length	PDU length	Unsigned 16-bit integer

Table A-73. ISO 10589 ISIS Partial Sequence Numbers Protocol Data Unit (isis_psnp)

A.74. ISO 8073 COTP Connection-Oriented Transport Protocol (cotp)

Field	Field Name	Type

Table A-74. ISO 8073 COTP Connection-Oriented Transport Protocol (cotp)

A.75. ISO 8473 CLNP ConnectionLess Network Protocol (clnp)

Field	Field Name	Type
clnp.checksum	Checksum	Unsigned 16-bit integer
clnp.dsap	DA	Byte array
clnp.dsap.len	DAL	Unsigned 8-bit integer
clnp.len	HDR Length	Unsigned 8-bit integer

Field	Field Name	Type
clnp.nlpi	Network Layer Protocol Identifier	Unsigned 8-bit integer
clnp.pdu.len	PDU length	Unsigned 16-bit integer
clnp.segment	CLNP Segment	No value
clnp.segment.error	Reassembly error	No value
clnp.segment.multipletails	Multiple tail segments found	Boolean
clnp.segment.overlap	Segment overlap	Boolean
clnp.segment.overlap.conflict	Conflicting data in segment overlap	Boolean
clnp.segment.toolongsegment	Segment too long	Boolean
clnp.segments	CLNP Segments	No value
clnp.ssap	SA	Byte array
clnp.ssap.len	SAL	Unsigned 8-bit integer
clnp.ttl	Holding Time	Unsigned 8-bit integer
clnp.type	PDU Type	Unsigned 8-bit integer
clnp.version	Version	Unsigned 8-bit integer

Table A-75. ISO 8473 CLNP ConnectionLess Network Protocol (clnp)

A.76. ISO 8602 CLTP ConnectionLess Transport Protocol (cltp)

Field	Field Name	Type

Table A-76. ISO 8602 CLTP ConnectionLess Transport Protocol (cltp)

A.77. ISO 9542 ESIS Routeing Information Exchange Protocol (esis)

Field	Field Name	Type

Field	Field Name	Type
esis.chksum	Checksum	Unsigned 16-bit integer
esis.htime	Holding Time	Unsigned 16-bit integer
esis.length	PDU Length	Unsigned 8-bit integer
esis.nlpi	Network Layer Protocol Identifier	Unsigned 8-bit integer
esis.res	Reserved(==0)	Unsigned 8-bit integer
esis.type	PDU Type	Unsigned 8-bit integer
esis.ver	Version (==1)	Unsigned 8-bit integer

Table A-77. ISO 9542 ESIS Routeing Information Exchange Protocol (esis)

A.78. ITU-T Recommendation H.261 (h261)

Field	Field Name	Type
h261.ebit	End bit position	Unsigned 8-bit integer
h261.gobn	GOB Number	Unsigned 8-bit integer
h261.hmvd	Horizontal motion vctor data	Unsigned 8-bit integer
h261.i	Intra frame encoded data flag	Boolean
h261.mbap	Macroblock address predictor	Unsigned 8-bit integer
h261.quant	Quantizer	Unsigned 8-bit integer
h261.sbit	Start bit position	Unsigned 8-bit integer
h261.stream	H.261 stream	Byte array
h261.v	Motion vector flag	Boolean
h261.vmvd	Vertical motion vector data	Unsigned 8-bit integer

Table A-78. ITU-T Recommendation H.261 (h261)

A.79. Internet Cache Protocol (icp)

Field	Field Name	Type

Field	Field Name	Type
icp.length	Length	Unsigned 16-bit integer
icp.nr	Request Number	Unsigned 32-bit integer
icp.opcode	Opcode	Unsigned 8-bit integer
icp.version	Version	Unsigned 8-bit integer

Table A-79. Internet Cache Protocol (icp)

A.80. Internet Control Message Protocol (icmp)

Field	Field Name	Type
icmp.checksum	Checksum	Unsigned 16-bit integer
icmp.checksum_bad	Bad Checksum	Boolean
icmp.code	Code	Unsigned 8-bit integer
icmp.type	Type	Unsigned 8-bit integer

Table A-80. Internet Control Message Protocol (icmp)

A.81. Internet Control Message Protocol v6 (icmpv6)

Field	Field Name	Type
icmpv6.checksum	Checksum	Unsigned 16-bit integer
icmpv6.checksum_bad	Bad Checksum	Boolean
icmpv6.code	Code	Unsigned 8-bit integer
icmpv6.type	Type	Unsigned 8-bit integer

Table A-81. Internet Control Message Protocol v6 (icmpv6)

A.82. Internet Group Management Protocol (igmp)

Field	Field Name	Type
igmp.access_key	Access Key	Byte array
igmp.aux_data	Aux Data	Byte array
igmp.aux_data_len	Aux Data Len	Unsigned 8-bit integer
igmp.checksum	Checksum	Unsigned 16-bit integer
igmp.checksum_bad	Bad Checksum	Boolean
igmp.group_type	Type Of Group	Unsigned 8-bit integer
igmp.identifier	Identifier	Unsigned 32-bit integer
igmp.max_resp	Max Resp Time	Unsigned 8-bit integer
igmp.max_resp.exp	Exponent	Unsigned 8-bit integer
igmp.max_resp.mant	Mantissa	Unsigned 8-bit integer
igmp.mtrace.max_hops	# hops	Unsigned 8-bit integer
igmp.mtrace.q_arrival	Query Arrival	Unsigned 32-bit integer
igmp.mtrace.q_fwd_code	Forwarding Code	Unsigned 8-bit integer
igmp.mtrace.q_fwd_ttl	FwdTTL	Unsigned 8-bit integer
igmp.mtrace.q_id	Query ID	Unsigned 24-bit integer
igmp.mtrace.q_inaddr	In itf addr	IPv4 address
igmp.mtrace.q_inpkt	In pkts	Unsigned 32-bit integer
igmp.mtrace.q_mbz	MBZ	Unsigned 8-bit integer
igmp.mtrace.q_outaddr	Out itf addr	IPv4 address
igmp.mtrace.q_outpkt	Out pkts	Unsigned 32-bit integer
igmp.mtrace.q_prevrtr	Previous rtr addr	IPv4 address
igmp.mtrace.q_rtg_proto	Rtg Protocol	Unsigned 8-bit integer
igmp.mtrace.q_s	S	Unsigned 8-bit integer
igmp.mtrace.q_src_mask	Src Mask	Unsigned 8-bit integer
igmp.mtrace.q_total	S,G pkt count	Unsigned 32-bit integer
igmp.mtrace.raddr	Receiver Address	IPv4 address
igmp.mtrace.resp_ttl	Response TTL	Unsigned 8-bit integer
igmp.mtrace.rspaddr	Response Address	IPv4 address
igmp.mtrace.saddr	Source Address	IPv4 address
igmp.num_grp_recs	Num Group Records	Unsigned 16-bit integer
igmp.num_src	Num Src	Unsigned 16-bit integer
igmp.qqic	QQIC	Unsigned 8-bit integer
igmp.qrv	QRV	Unsigned 8-bit integer
igmp.record_type	Record Type	Unsigned 8-bit integer
igmp.reply	Reply	Unsigned 8-bit integer
igmp.reply.pending	Reply Pending	Unsigned 8-bit integer
igmp.s	S	Boolean

Field	Field Name	Type
igmp.type	Type	Unsigned 8-bit integer
igmp.version	IGMP Version	Unsigned 8-bit integer

Table A-82. Internet Group Management Protocol (igmp)

A.83. Internet Message Access Protocol (imap)

Field	Field Name	Type
imap.request	Request	Boolean
imap.response	Response	Boolean

Table A-83. Internet Message Access Protocol (imap)

A.84. Internet Printing Protocol (ipp)

Field	Field Name	Type

Table A-84. Internet Printing Protocol (ipp)

A.85. Internet Protocol (ip)

Field	Field Name	Type
ip.addr	Source or Destination Address	IPv4 address
ip.checksum	Header checksum	Unsigned 16-bit integer
ip.checksum_bad	Bad Header checksum	Boolean
ip.dsfield	Differentiated Services field	Unsigned 8-bit integer
ip.dsfield.ce	ECN-CE	Unsigned 8-bit integer
ip.dsfield.dscp	Differentiated Services Codepoint	Unsigned 8-bit integer

Field	Field Name	Type
ip.dsfield.ect	ECN-Capable Transport (ECT)	Unsigned 8-bit integer
ip.dst	Destination	IPv4 address
ip.flags	Flags	Unsigned 8-bit integer
ip.flags.df	Don’t fragment	Boolean
ip.flags.mf	More fragments	Boolean
ip.frag_offset	Fragment offset	Unsigned 16-bit integer
ip.fragment	IP Fragment	No value
ip.fragment.error	Defragmentation error	No value
ip.fragment.multipletails	Multiple tail fragments found	Boolean
ip.fragment.overlap	Fragment overlap	Boolean
ip.fragment.overlap.conflict	Conflicting data in fragment overlap	Boolean
ip.fragment.toolongfragment	Fragment too long	Boolean
ip.fragments	IP Fragments	No value
ip.hdr_len	Header Length	Unsigned 8-bit integer
ip.id	Identification	Unsigned 16-bit integer
ip.len	Total Length	Unsigned 16-bit integer
ip.proto	Protocol	Unsigned 8-bit integer
ip.src	Source	IPv4 address
ip.tos	Type of Service	Unsigned 8-bit integer
ip.tos.cost	Cost	Boolean
ip.tos.delay	Delay	Boolean
ip.tos.precedence	Precedence	Unsigned 8-bit integer
ip.tos.reliability	Reliability	Boolean
ip.tos.throughput	Throughput	Boolean
ip.ttl	Time to live	Unsigned 8-bit integer
ip.version	Version	Unsigned 8-bit integer

Table A-85. Internet Protocol (ip)

A.86. Internet Protocol Version 6 (ipv6)

Field	Field Name	Type
ipv6.addr	Address	IPv6 address
ipv6.class	Traffic class	Unsigned 8-bit integer
ipv6.dst	Destination	IPv6 address
ipv6.flow	Flowlabel	Unsigned 32-bit integer
ipv6.fragment	IPv6 Fragment	No value
ipv6.fragment.error	Defragmentation error	No value
ipv6.fragment.multipletails	Multiple tail fragments found	Boolean
ipv6.fragment.overlap	Fragment overlap	Boolean
ipv6.fragment.overlap.conflict	Conflicting data in fragment overlap	Boolean
ipv6.fragment.toolongfragment	Fragment too long	Boolean
ipv6.fragments	IPv6 Fragments	No value
ipv6.hlim	Hop limit	Unsigned 8-bit integer
ipv6.mipv6_a_flag	Acknowledge (A)	Boolean
ipv6.mipv6_b_flag	Bicasting all (B)	Boolean
ipv6.mipv6_d_flag	Duplicate Address Detection (D)	Boolean
ipv6.mipv6_h_flag	Home Registration (H)	Boolean
ipv6.mipv6_home_address	Home Address	IPv6 address
ipv6.mipv6_length	Option Length	Unsigned 8-bit integer
ipv6.mipv6_life_time	Life Time	Unsigned 32-bit integer
ipv6.mipv6_m_flag	MAP Registration (M)	Boolean
ipv6.mipv6_prefix_length	Prefix Length	Unsigned 8-bit integer
ipv6.mipv6_r_flag	Router (R)	Boolean
ipv6.mipv6_refresh	Refresh	Unsigned 32-bit integer
ipv6.mipv6_sequence_number	Sequence Number	Unsigned 16-bit integer
ipv6.mipv6_status	Status	Unsigned 8-bit integer
ipv6.mipv6_sub_alternative_COA	Alternative Care of Address	IPv6 address
ipv6.mipv6_sub_length	Sub-Option Length	Unsigned 8-bit integer
ipv6.mipv6_sub_type	Sub-Option Type	Unsigned 8-bit integer
ipv6.mipv6_sub_unique_ID	Unique Identifier	Unsigned 16-bit integer

Field	Field Name	Type
ipv6.mipv6_type	Option Type	Unsigned 8-bit integer
ipv6.nxt	Next header	Unsigned 8-bit integer
ipv6.plen	Payload length	Unsigned 16-bit integer
ipv6.src	Source	IPv6 address
ipv6.version	Version	Unsigned 8-bit integer

Table A-86. Internet Protocol Version 6 (ipv6)

A.87. Internet Relay Chat (irc)

Field	Field Name	Type
irc.command	Command	String
irc.request	Request	Boolean
irc.response	Response	Boolean

Table A-87. Internet Relay Chat (irc)

A.88. Internet Security Association and Key Management Protocol (isakmp)

Field	Field Name	Type

Table A-88. Internet Security Association and Key Management Protocol (isakmp)

A.89. Internetwork Packet eXchange (ipx)

Field	Field Name	Type
ipx.checksum	Checksum	Unsigned 16-bit integer

Field	Field Name	Type
ipx.dst.net	Destination Network	IPX network or server name
ipx.dst.node	Destination Node	6-byte Hardware (MAC) Address
ipx.dst.socket	Destination Socket	Unsigned 16-bit integer
ipx.hops	Transport Control (Hops)	Unsigned 8-bit integer
ipx.len	Length	Unsigned 16-bit integer
ipx.packet_type	Packet Type	Unsigned 8-bit integer
ipx.src.net	Source Network	IPX network or server name
ipx.src.node	Source Node	6-byte Hardware (MAC) Address
ipx.src.socket	Source Socket	Unsigned 16-bit integer

Table A-89. Internetwork Packet eXchange (ipx)

A.90. Kerberos (kerberos)

Field	Field Name	Type

Table A-90. Kerberos (kerberos)

A.91. Kernel Lock Manager (klm)

Field	Field Name	Type
klm.block	block	Boolean
klm.exclusive	exclusive	Boolean
klm.holder	holder	No value
klm.len	length	Unsigned 32-bit integer
klm.lock	lock	No value
klm.offset	offset	Unsigned 32-bit integer
klm.pid	pid	Unsigned 32-bit integer
klm.servername	server name	String

Field	Field Name	Type
klm.stats	stats	Unsigned 32-bit integer

Table A-91. Kernel Lock Manager (klm)

A.92. Label Distribution Protocol (ldp)

Field	Field Name	Type
ldp,msg.tlv.hello.requested	Hello Requested	Boolean
ldp.hdr.ldpid.lsid	Label Space ID	Unsigned 16-bit integer
ldp.hdr.ldpid.lsr	LSR ID	Unsigned 32-bit integer
ldp.hdr.pdu_len	PDU Length	Unsigned 16-bit integer
ldp.hdr.version	Version	Unsigned 16-bit integer
ldp.msg.id	Message ID	Unsigned 32-bit integer
ldp.msg.len	Message Length	Unsigned 16-bit integer
ldp.msg.tlv.fec.af	FEC Element Address Type	Unsigned 16-bit integer
ldp.msg.tlv.fec.len	FEC Element Length	Unsigned 8-bit integer
ldp.msg.tlv.fec.pfval	FEC Element Prefix Value	IPv4 address
ldp.msg.tlv.fec.type	FEC Element Type	Unsigned 8-bit integer
ldp.msg.tlv.hello.cnf_seqno	Configuration Sequence Number	Unsigned 32-bit integer
ldp.msg.tlv.hello.hold	Hold Time	Unsigned 16-bit integer
ldp.msg.tlv.hello.res	Reserved	Unsigned 16-bit integer
ldp.msg.tlv.hello.targeted	Targeted Hello	Boolean
ldp.msg.tlv.label	Generic Label	Unsigned 32-bit integer
ldp.msg.tlv.len	TLV Length	Unsigned 16-bit integer
ldp.msg.tlv.type	TLV Type	Unsigned 16-bit integer
ldp.msg.tlv.value	TLV Value	Byte array
ldp.msg.type	Message Type	Unsigned 16-bit integer
ldp.req	Request	Boolean
ldp.rsp	Response	Boolean

Table A-92. Label Distribution Protocol (ldp)

A.93. Layer 2 Tunneling Protocol (l2tp)

Field	Field Name	Type
l2tp.Nr	Nr	Unsigned 16-bit integer
l2tp.Ns	Ns	Unsigned 16-bit integer
l2tp.length	Length	Unsigned 16-bit integer
l2tp.offset	Offset	Unsigned 16-bit integer
l2tp.session	Session ID	Unsigned 16-bit integer
l2tp.tunnel	Tunnel ID	Unsigned 16-bit integer
lt2p.avp.hidden	Hidden	Boolean
lt2p.avp.length	Length	Unsigned 16-bit integer
lt2p.avp.mandatory	Mandatory	Boolean
lt2p.avp.type	Type	Unsigned 16-bit integer
lt2p.avp.vendor_id	Vendor ID	Unsigned 16-bit integer
lt2p.length_bit	Length Bit	Boolean
lt2p.offset_bit	Offset bit	Boolean
lt2p.priority	Priority	Boolean
lt2p.seq_bit	Sequence Bit	Boolean
lt2p.type	Type	Unsigned 16-bit integer
lt2p.version	Version	Unsigned 16-bit integer

Table A-93. Layer 2 Tunneling Protocol (l2tp)

A.94. Lightweight Directory Access Protocol (ldap)

Field	Field Name	Type
ldap.abandon.msgid	Abandon Msg Id	Unsigned 32-bit integer
ldap.attribute	Attribute	String
ldap.bind.auth_type	Auth Type	Unsigned 8-bit integer
ldap.bind.dn	DN	String
ldap.bind.password	Password	String
ldap.bind.version	Version	Unsigned 32-bit integer
ldap.compare.test	Test	String
ldap.dn	Distinguished Name	String

Field	Field Name	Type
ldap.length	Length	Unsigned 32-bit integer
ldap.message_id	Message Id	Unsigned 32-bit integer
ldap.message_length	Message Length	Unsigned 32-bit integer
ldap.message_type	Message Type	Unsigned 8-bit integer
ldap.modify.add	Add	String
ldap.modify.delete	Delete	String
ldap.modify.replace	Replace	String
ldap.modrdn.delete	Delete Values	Boolean
ldap.modrdn.name	New Name	String
ldap.modrdn.superior	New Location	String
ldap.result.code	Result Code	Unsigned 8-bit integer
ldap.result.errormsg	Error Message	String
ldap.result.matcheddn	Matched DN	String
ldap.result.referral	Referral	String
ldap.search.basedn	Base DN	String
ldap.search.dereference	Dereference	Unsigned 8-bit integer
ldap.search.filter	Filter	String
ldap.search.scope	Scope	Unsigned 8-bit integer
ldap.search.sizelimit	Size Limit	Unsigned 32-bit integer
ldap.search.timelimit	Time Limit	Unsigned 32-bit integer
ldap.search.typesonly	Attributes Only	Boolean
ldap.value	Value	String

Table A-94. Lightweight Directory Access Protocol (ldap)

A.95. Line Printer Daemon Protocol (lpd)

Field	Field Name	Type
lpd.request	Request	Boolean
lpd.response	Response	Boolean

Table A-95. Line Printer Daemon Protocol (lpd)

A.96. Link Access Procedure Balanced (LAPB)

(lapb)

Field	Field Name	Type
lapb.address	Address Field	Unsigned 8-bit integer
lapb.control	Control Field	Unsigned 8-bit integer

Table A-96. Link Access Procedure Balanced (LAPB) (lapb)

A.97. Link Access Procedure Balanced Ethernet (LAPBETHER) (lapbether)

Field	Field Name	Type
lapbether.length	Length Field	Unsigned 16-bit integer

Table A-97. Link Access Procedure Balanced Ethernet (LAPBETHER) (lapbether)

A.98. Link Access Procedure, Channel D (LAPD) (lapd)

Field	Field Name	Type
lapd.address	Address Field	Unsigned 16-bit integer
lapd.control	Control Field	Unsigned 16-bit integer
lapd.cr	C/R	Unsigned 16-bit integer
lapd.ea1	EA1	Unsigned 16-bit integer
lapd.ea2	EA2	Unsigned 16-bit integer
lapd.sapi	SAPI	Unsigned 16-bit integer
lapd.tei	TEI	Unsigned 16-bit integer

Table A-98. Link Access Procedure, Channel D (LAPD) (lapd)

A.99. Linux cooked-mode capture (sll)

Field	Field Name	Type
sll.etype	Protocol	Unsigned 16-bit integer
sll.halen	Link-layer address length	Unsigned 16-bit integer
sll.hatype	Link-layer address type	Unsigned 16-bit integer
sll.ltype	Protocol	Unsigned 16-bit integer
sll.pkttype	Packet type	Unsigned 16-bit integer
sll.src.eth	Source	6-byte Hardware (MAC) Address
sll.src.other	Source	Byte array
sll.trailer	Trailer	Byte array

Table A-99. Linux cooked-mode capture (sll)

A.100. Local Management Interface (lmi)

Field	Field Name	Type
lmi.cmd	Call reference	Unsigned 8-bit integer
lmi.dlci_act	DLCI Active	Unsigned 8-bit integer
lmi.dlci_hi	DLCI High	Unsigned 8-bit integer
lmi.dlci_low	DLCI Low	Unsigned 8-bit integer
lmi.dlci_new	DLCI New	Unsigned 8-bit integer
lmi.ele_rcd_type	Record Type	Unsigned 8-bit integer
lmi.inf_ele	Information Element	Unsigned 8-bit integer
lmi.inf_ele_len	Length	Unsigned 8-bit integer
lmi.inf_ele_type	Type	Unsigned 8-bit integer
lmi.msg_type	Message Type	Unsigned 8-bit integer
lmi.recv_seq	Recv Seq	Unsigned 8-bit integer
lmi.send_seq	Send Seq	Unsigned 8-bit integer

Table A-100. Local Management Interface (lmi)

A.101. Logical-Link Control (llc)

Field	Field Name	Type
llc.control	Control	Unsigned 16-bit integer
llc.dsap	DSAP	Unsigned 8-bit integer
llc.dsap.ig	IG Bit	Boolean
llc.oui	Organization Code	Unsigned 24-bit integer
llc.pid	Protocol ID	Unsigned 16-bit integer
llc.ssap	SSAP	Unsigned 8-bit integer
llc.ssap.cr	CR Bit	Boolean
llc.type	Type	Unsigned 16-bit integer

Table A-101. Logical-Link Control (llc)

A.102. Lucent/Ascend debug output (ascend)

Field	Field Name	Type
ascend.chunk	WDD Chunk	Unsigned 32-bit integer
ascend.number	Called number	String
ascend.sess	Session ID	Unsigned 32-bit integer
ascend.task	Task	Unsigned 32-bit integer
ascend.type	Link type	Unsigned 32-bit integer
ascend.user	User name	String

Table A-102. Lucent/Ascend debug output (ascend)

A.103. MAPI (mapi)

Field	Field Name	Type
mapi.request	Request	Boolean
mapi.response	Response	Boolean

Table A-103. MAPI (mapi)

A.104. MS Proxy Protocol (msproxy)

Field	Field Name	Type
msproxy.bindaddr	Destination	IPv4 address
msproxy.bindid	Bound Port Id	Unsigned 32-bit integer
msproxy.bindport	Bind Port	Unsigned 16-bit integer
msproxy.boundport	Bound Port	Unsigned 16-bit integer
msproxy.clntport	Client Port	Unsigned 16-bit integer
msproxy.command	Command	Unsigned 16-bit integer
msproxy.dstaddr	Destination Address	IPv4 address
msproxy.dstport	Destination Port	Unsigned 16-bit integer
msproxy.resolvaddr	Address	IPv4 address
msproxy.server_ext_addr	Server External Address	IPv4 address
msproxy.server_ext_port	Server External Port	Unsigned 16-bit integer
msproxy.server_int_addr	Server Internal Address	IPv4 address
msproxy.server_int_port	Server Internal Port	Unsigned 16-bit integer
msproxy.serveraddr	Server Address	IPv4 address
msproxy.serverport	Server Port	Unsigned 16-bit integer
msproxy.srcport	Source Port	Unsigned 16-bit integer

Table A-104. MS Proxy Protocol (msproxy)

A.105. MSNIP : Multicast Source Notification of Interest Protocol (msnip)

Field	Field Name	Type
msnip.checksum	Checksum	Unsigned 16-bit integer
msnip.checksum_bad	Bad Checksum	Boolean
msnip.count	Count	Unsigned 8-bit integer
msnip.genid	Generation ID	Unsigned 16-bit integer
msnip.groups	Groups	No value
msnip.holdtime	Holdtime	Unsigned 32-bit integer
msnip.holdtime16	Holdtime	Unsigned 16-bit integer
msnip.maddr	Multicast group	IPv4 address
msnip.netmask	Netmask	Unsigned 8-bit integer

Field	Field Name	Type
msnip.rec_type	Record Type	Unsigned 8-bit integer
msnip.type	Type	Unsigned 8-bit integer

Table A-105. MSNIP : Multicast Source Notification of Interest Protocol (msnip)

A.106. MTP 3 User Adaptation Layer (m3ua)

Field	Field Name	Type
m3ua.affected_dpc	Affected DPC	Unsigned 24-bit integer
m3ua.congestion_level	Congestion level	Unsigned 8-bit integer
m3ua.error_code	Error code	Unsigned 32-bit integer
m3ua.info_string	Info string	String
m3ua.mask	Mask	Unsigned 8-bit integer
m3ua.message_class	Message class	Unsigned 8-bit integer
m3ua.message_length	Message length	Unsigned 32-bit integer
m3ua.message_type	Message Type	Unsigned 8-bit integer
m3ua.network_appearance	Network appearance	Unsigned 32-bit integer
m3ua.parameter_length	Parameter length	Unsigned 16-bit integer
m3ua.parameter_tag	Parameter Tag	Unsigned 16-bit integer
m3ua.reason	Reason	Unsigned 32-bit integer
m3ua.reserved	Reserved	Unsigned 8-bit integer
m3ua.routing_context	Routing context	Unsigned 32-bit integer
m3ua.status_info	Status info	Unsigned 16-bit integer
m3ua.status_type	Status type	Unsigned 16-bit integer
m3ua.traffic_mode_type	Traffic mode Type	Unsigned 32-bit integer
m3ua.unavailability_cause	Unavailability cause	Unsigned 16-bit integer
m3ua.user_identity	User Identity	Unsigned 16-bit integer
m3ua.version	Version	Unsigned 8-bit integer

Table A-106. MTP 3 User Adaptation Layer (m3ua)

A.107. MTP2 Peer Adaptation Layer (m2pa)

Field	Field Name	Type
m2pa.li	Length Indicator	Unsigned 8-bit integer
m2pa.li.prio	Priority	Unsigned 8-bit integer
m2pa.li.spare	Spare	Unsigned 8-bit integer
m2pa.message_length	Message length	Unsigned 32-bit integer
m2pa.message_type	Message Type	Unsigned 16-bit integer
m2pa.spare	Spare	Unsigned 8-bit integer
m2pa.status	Link Status Status	Unsigned 32-bit integer
m2pa.version	Version	Unsigned 8-bit integer

Table A-107. MTP2 Peer Adaptation Layer (m2pa)

A.108. Malformed Frame (malformed)

Field	Field Name	Type

Table A-108. Malformed Frame (malformed)

A.109. Media Gateway Control Protocol (mgcp)

Field	Field Name	Type
mgcp.messagecount	MGCP Message Count	Unsigned 32-bit integer
mgcp.param.bearerinfo	BearerInformation (B)	String
mgcp.param.callid	CallId (C)	String
mgcp.param.capabilities	Capabilities (A)	String
mgcp.param.connectionid	ConnectionIdentifier (I)	String
mgcp.param.connectionmod	ConnectionMode (M)	String
mgcp.param.connectionparam	ConnectionParameters (P)	String

Field	Field Name	Type
mgcp.param.detectedevents	DetectedEvents (T)	String
mgcp.param.digitmap	DigitMap (D)	String
mgcp.param.eventstates	EventStates (ES)	String
mgcp.param.extention	Extention Parameter (X-*)	String
mgcp.param.invalid	Invalid Parameter	String
mgcp.param.localconnectionoptions	LocalConnectionOptions (L)	String
mgcp.param.notifiedentity	NotifiedEntity (N)	String
mgcp.param.observedevents	ObservedEvents (O)	String
mgcp.param.quarantinehandling	QuarantineHandling (Q)	String
mgcp.param.reasoncode	ReasonCode (E)	String
mgcp.param.reqevents	RequestedEvents (R)	String
mgcp.param.reqinfo	RequestedInfo (F)	String
mgcp.param.requestid	RequestIdentifier (X)	String
mgcp.param.restartdelay	RestartDelay (RD)	String
mgcp.param.restartmethod	RestartMethod (RM)	String
mgcp.param.rspack	ResponseAck (K)	String
mgcp.param.secondconnectionid	SecondConnectionID (I2)	String
mgcp.param.secondendpointid	SecondEndpointID (Z2)	String
mgcp.param.signalreq	SignalRequests (S)	String
mgcp.param.specificendpointid	SpecificEndpointID (Z)	String
mgcp.req	Request	Boolean
mgcp.req.endpoint	Endpoint	String
mgcp.req.verb	Verb	String
mgcp.rsp	Response	Boolean
mgcp.rsp.rspcode	Response Code	String
mgcp.rsp.rspstring	Response String	String
mgcp.transid	Transaction ID	String
mgcp.version	Version	String

Table A-109. Media Gateway Control Protocol (mgcp)

A.110. Message Transfer Part Level 3 (mtp3)

Field	Field Name	Type
mtp3.dpc	DPC	Unsigned 32-bit integer
mtp3.dpc.cluster	DPC Cluster	Unsigned 24-bit integer
mtp3.dpc.member	DPC Member	Unsigned 24-bit integer
mtp3.dpc.network	DPC Network	Unsigned 24-bit integer
mtp3.network_indicator	Network indicator	Unsigned 8-bit integer
mtp3.opc	OPC	Unsigned 32-bit integer
mtp3.opc.cluster	OPC Cluster	Unsigned 24-bit integer
mtp3.opc.member	OPC Member	Unsigned 24-bit integer
mtp3.opc.network	OPC Network	Unsigned 24-bit integer
mtp3.priority	Priority	Unsigned 8-bit integer
mtp3.service_indicator	Service indicator	Unsigned 8-bit integer
mtp3.sls	Signalling Link Selector	Unsigned 32-bit integer
mtp3.spare	Spare	Unsigned 8-bit integer

Table A-110. Message Transfer Part Level 3 (mtp3)

A.111. Microsoft Windows Browser Protocol (browser)

Field	Field Name	Type
browser.backup.count	Backup List Requested Count	Unsigned 8-bit integer
browser.backup.server	Backup Server	String
browser.backup.token	Backup Request Token	Unsigned 32-bit integer
browser.browser_to_promote	Browser to Promote	String
browser.command	Command	Unsigned 8-bit integer
browser.comment	Host Comment	String
browser.election.criteria	Election Criteria	Unsigned 32-bit integer

Field	Field Name	Type
browser.election.desire	Election Desire	Unsigned 8-bit integer
browser.election.desire.backup	Backup	Boolean
browser.election.desire.domain_master	Domain Master	Boolean
browser.election.desire.master	Master	Boolean
browser.election.desire.nt	NT	Boolean
browser.election.desire.standby	Standby	Boolean
browser.election.desire.wins	WINS	Boolean
browser.election.os	Election OS	Unsigned 8-bit integer
browser.election.os.nts	NT Server	Boolean
browser.election.os.ntw	NT Workstation	Boolean
browser.election.os.wfw	WfW	Boolean
browser.election.revision	Election Revision	Unsigned 16-bit integer
browser.election.version	Election Version	Unsigned 8-bit integer
browser.mb_server	Master Browser Server Name	String
browser.os_major	OS Major Version	Unsigned 8-bit integer
browser.os_minor	OS Minor Version	Unsigned 8-bit integer
browser.period	Update Periodicity	Unsigned 32-bit integer
browser.proto_major	Browser Protocol Major Version	Unsigned 8-bit integer
browser.proto_minor	Browser Protocol Minor Version	Unsigned 8-bit integer
browser.response_computer_name	Response Computer Name	String
browser.server	Server Name	String
browser.server_type	Server Type	Unsigned 32-bit integer
browser.server_type.apple	Apple	Boolean
browser.server_type.backup_controller	Backup Controller	Boolean
browser.server_type.browser	Backup Browser	Boolean
browser.server_type.domain_master	Domain Master Browser	Boolean

Field	Field Name	Type
browser.server_type.browser	Master Browser	Boolean
browser.server_type.browser_potential	Potential Browser	Boolean
browser.server_type.dialin	Dialin	Boolean
browser.server_type.domain_ctrl	Domain Controller	Boolean
browser.server_type.domain_enum	Domain Enum	Boolean
browser.server_type.local	Local	Boolean
browser.server_type.member	Member	Boolean
browser.server_type.novell	Novell	Boolean
browser.server_type.nts	NT Server	Boolean
browser.server_type.ntw	NT Workstation	Boolean
browser.server_type.osf	OSF	Boolean
browser.server_type.print	Print	Boolean
browser.server_type.server	Server	Boolean
browser.server_type.sql	SQL	Boolean
browser.server_type.time	Time Source	Boolean
browser.server_type.vms	VMS	Boolean
browser.server_type.w95	Windows 95+	Boolean
browser.server_type.wfw	WfW	Boolean
browser.server_type.workstation	Workstation	Boolean
browser.server_type.xenix	Xenix	Boolean
browser.sig	Signature	Unsigned 16-bit integer
browser.unused	Unused flags	Unsigned 8-bit integer
browser.update_count	Update Count	Unsigned 8-bit integer
browser.uptime	Uptime	Unsigned 32-bit integer

Table A-111. Microsoft Windows Browser Protocol (browser)

A.112. Microsoft Windows Lanman Protocol (lanman)

Field	Field Name	Type

Table A-112. Microsoft Windows Lanman Protocol (lanman)

A.113. Microsoft Windows Logon Protocol (netlogon)

Field	Field Name	Type
netlogon.command	Command	Unsigned 8-bit integer
netlogon.computer_name	Computer Name	String
netlogon.date_time	Date/Time	Unsigned 32-bit integer
netlogon.db_count	DB Count	Unsigned 32-bit integer
netlogon.db_index	Database Index	Unsigned 32-bit integer
netlogon.domain_name	Domain Name	String
netlogon.domain_sid	Domain SID	Byte array
netlogon.domain_sid_size	Domain SID Size	Unsigned 32-bit integer
netlogon.flags.autolock	Autolock	Boolean
netlogon.flags.enabled	Enabled	Boolean
netlogon.flags.expire	Expire	Boolean
netlogon.flags.homedir	Homedir	Boolean
netlogon.flags.interdomain	Interdomain Trust	Boolean
netlogon.flags.mns	MNS User	Boolean
netlogon.flags.normal	Normal User	Boolean
netlogon.flags.password	Password	Boolean
netlogon.flags.server	Server Trust	Boolean
netlogon.flags.temp_dup	Temp Duplicate User	Boolean
netlogon.flags.workstation	Workstation Trust	Boolean
netlogon.large_serial	Large Serial Number	Byte array

Field	Field Name	Type
netlogon.lm_token	LM Token	Unsigned 16-bit integer
netlogon.lmnt_token	LMNT Token	Unsigned 16-bit integer
netlogon.low_serial	Low Serial Number	Unsigned 32-bit integer
netlogon.mailslot_name	Mailslot Name	String
netlogon.major_version	Workstation Major Version	Unsigned 8-bit integer
netlogon.minor_version	Workstation Minor Version	Unsigned 8-bit integer
netlogon.nt_date_time	NT Date/Time	Byte array
netlogon.nt_version	NT Version	Unsigned 32-bit integer
netlogon.os_version	Workstation OS Version	Unsigned 8-bit integer
netlogon.pdc_name	PDC Name	String
netlogon.pulse	Pulse	Unsigned 32-bit integer
netlogon.random	Random	Unsigned 32-bit integer
netlogon.request_count	Request Count	Unsigned 16-bit integer
netlogon.script_name	Script Name	String
netlogon.server_name	Server Name	String
netlogon.unicode_computer_name	Unicode Computer Name	String
netlogon.unicode_pdc_name	Unicode PDC Name	String
netlogon.update	Update Type	Unsigned 16-bit integer
netlogon.user_name	User Name	String

Table A-113. Microsoft Windows Logon Protocol (netlogon)

A.114. Mobile IP (mip)

Field	Field Name	Type
mip.auth.auth	Authenticator	Byte array
mip.auth.spi	SPI	Signed 32-bit integer
mip.b	Broadcast Datagrams	Boolean
mip.coa	Care of Address	IPv4 address
mip.code	Reply Code	Unsigned 8-bit integer
mip.d	Co-lcated Care-of Address	Boolean

Field	Field Name	Type
mip.ext.len	Extension Length	Signed 8-bit integer
mip.ext.type	Extension Type	Signed 8-bit integer
mip.g	GRE	Boolean
mip.haaddr	Home Agent	IPv4 address
mip.homeaddr	Home Address	IPv4 address
mip.ident	Identification	Date/Time stamp
mip.life	Lifetime	Unsigned 16-bit integer
mip.m	Minimal Encapsulation	Boolean
mip.nai	NAI	String
mip.s	Simultaneous Bindings	Boolean
mip.t	Reverse Tunneling	Boolean
mip.type	Message Type	Signed 8-bit integer
mip.v	Van Jacobson	Boolean

Table A-114. Mobile IP (mip)

A.115. Modbus/TCP (mbtcp)

Field	Field Name	Type
modbus_tcp.func_code	function code	Unsigned 8-bit integer
modbus_tcp.len	length	Unsigned 16-bit integer
modbus_tcp.prot_id	protocol identifier	Unsigned 16-bit integer
modbus_tcp.trans_id	transaction identifier	Unsigned 16-bit integer
modbus_tcp.unit_id	unit identifier	Unsigned 8-bit integer

Table A-115. Modbus/TCP (mbtcp)

A.116. Mount Service (mount)

Field	Field Name	Type
mount.dump.directory	Directory	String
mount.dump.entry	Mount List Entry	No value
mount.dump.hostname	Hostname	String

Field	Field Name	Type
mount.export.directory	Directory	String
mount.export.entry	Export List Entry	No value
mount.export.group	Group	String
mount.export.groups	Groups	No value
mount.flavor	Flavor	Unsigned 32-bit integer
mount.flavors	Flavors	Unsigned 32-bit integer
mount.path	Path	String
mount.pathconf.link_max	Maximum number of links to a file	Unsigned 32-bit integer
mount.pathconf.mask	Reply error/status bits	Unsigned 16-bit integer
mount.pathconf.mask.chown_restricted	CHOWN_RESTRICTED	Boolean
mount.pathconf.mask.error_all	ERROR_ALL	Boolean
mount.pathconf.mask.error_link_max	ERROR_LINK_MAX	Boolean
mount.pathconf.mask.error_max_canon	ERROR_MAX_CANON	Boolean
mount.pathconf.mask.error_max_input	ERROR_MAX_INPUT	Boolean
mount.pathconf.mask.error_name_max	ERROR_NAME_MAX	Boolean
mount.pathconf.mask.error_path_max	ERROR_PATH_MAX	Boolean
mount.pathconf.mask.error_pipe_buf	ERROR_PIPE_BUF	Boolean
mount.pathconf.mask.error_vdisable	ERROR_VDISABLE	Boolean
mount.pathconf.mask.no_trunc	NO_TRUNC	Boolean
mount.pathconf.max_canon	Maximum terminal input line length	Unsigned 16-bit integer
mount.pathconf.max_input	Terminal input buffer size	Unsigned 16-bit integer
mount.pathconf.name_max	Maximum file name length	Unsigned 16-bit integer
mount.pathconf.path_max	Maximum path name length	Unsigned 16-bit integer

Field	Field Name	Type
mount.pathconf.pipe_buf	Pipe buffer size	Unsigned 16-bit integer
mount.pathconf.vdisable_char	VDISABLE character	Unsigned 8-bit integer
mount.status	Status	Unsigned 32-bit integer

Table A-116. Mount Service (mount)

A.117. MultiProtocol Label Switching Header (mpls)

Field	Field Name	Type
mpls.bottom	MPLS Bottom Of Label Stack	Unsigned 8-bit integer
mpls.exp	MPLS Experimental Bits	Unsigned 8-bit integer
mpls.label	MPLS Label	Unsigned 32-bit integer
mpls.ttl	MPLS TTL	Unsigned 8-bit integer

Table A-117. MultiProtocol Label Switching Header (mpls)

A.118. Multicast Router DISCovery protocol (mrdisc)

Field	Field Name	Type
mrdisc.adv_int	Advertising Interval	Unsigned 8-bit integer
mrdisc.checksum	Checksum	Unsigned 16-bit integer
mrdisc.checksum_bad	Bad Checksum	Boolean
mrdisc.num_opts	Number Of Options	Unsigned 16-bit integer
mrdisc.opt_len	Length	Unsigned 8-bit integer
mrdisc.option	Option	Unsigned 8-bit integer
mrdisc.option_data	Data	Byte array
mrdisc.options	Options	No value
mrdisc.query_int	Query Interval	Unsigned 16-bit integer
mrdisc.rob_var	Robustness Variable	Unsigned 16-bit integer

Field	Field Name	Type
mrdisc.type	Type	Unsigned 8-bit integer

Table A-118. Multicast Router DISCovery protocol (mrdisc)

A.119. Multicast Source Discovery Protocol (msdp)

Field	Field Name	Type
msdp.length	Length	Unsigned 16-bit integer
msdp.not.entry_count	Entry Count	Unsigned 24-bit integer
msdp.not.error	Error Code	Unsigned 8-bit integer
msdp.not.error_sub	Error subode	Unsigned 8-bit integer
msdp.not.ipv4	IPv4 address	IPv4 address
msdp.not.o	Open-bit	Unsigned 8-bit integer
msdp.not.res	Reserved	Unsigned 24-bit integer
msdp.not.sprefix_len	Sprefix len	Unsigned 8-bit integer
msdp.sa.entry_count	Entry Count	Unsigned 8-bit integer
msdp.sa.group_addr	Group Address	IPv4 address
msdp.sa.reserved	Reserved	Unsigned 24-bit integer
msdp.sa.rp_addr	RP Address	IPv4 address
msdp.sa.sprefix_len	Sprefix len	Unsigned 8-bit integer
msdp.sa.src_addr	Source Address	IPv4 address
msdp.sa_req.group_addr	Group Address	IPv4 address
msdp.sa_req.res	Reserved	Unsigned 8-bit integer
msdp.type	Type	Unsigned 8-bit integer

Table A-119. Multicast Source Discovery Protocol (msdp)

A.120. NIS+ (nisplus)

Field	Field Name	Type
.nisplus.dummy		Byte array
nisplus.access.mask	access mask	No value

Field	Field Name	Type
nisplus.aticks	aticks	Unsigned 32-bit integer
nisplus.attr	Attribute	No value
nisplus.attr.name	name	String
nisplus.attr.val	val	Byte array
nisplus.attributes	Attributes	No value
nisplus.callback.status	status	Boolean
nisplus.checkpoint.dticks	dticks	Unsigned 32-bit integer
nisplus.checkpoint.status	status	Unsigned 32-bit integer
nisplus.checkpoint.zticks	zticks	Unsigned 32-bit integer
nisplus.cookie	cookie	Byte array
nisplus.cticks	cticks	Unsigned 32-bit integer
nisplus.ctime	ctime	Date/Time stamp
nisplus.directory	directory	No value
nisplus.directory.mask	mask	No value
nisplus.directory.mask.group_create	GROUP CREATE	Boolean
nisplus.directory.mask.group_destroy	GROUP DESTROY	Boolean
nisplus.directory.mask.group_modify	GROUP MODIFY	Boolean
nisplus.directory.mask.group_read	GROUP READ	Boolean
nisplus.directory.mask.nobody_create	NOBODY CREATE	Boolean
nisplus.directory.mask.nobody_destroy	NOBODY DESTROY	Boolean
nisplus.directory.mask.nobody_modify	NOBODY MODIFY	Boolean
nisplus.directory.mask.nobody_read	NOBODY READ	Boolean
nisplus.directory.mask.owner_create	OWNER CREATE	Boolean
nisplus.directory.mask.owner_destroy	OWNER DESTROY	Boolean
nisplus.directory.mask.owner_modify	OWNER MODIFY	Boolean
nisplus.directory.mask.owner_read	OWNER READ	Boolean

Field	Field Name	Type
nisplus.directory.mask.world_create	WORLD CREATE	Boolean
nisplus.directory.mask.world_destroy	WORLD DESTROY	Boolean
nisplus.directory.mask.world_modify	WORLD MODIFY	Boolean
nisplus.directory.mask.world_read	WORLD READ	Boolean
nisplus.directory.mask_list	mask list	No value
nisplus.directory.name	directory name	String
nisplus.directory.ttl	ttl	Unsigned 32-bit integer
nisplus.directory.type	type	Unsigned 32-bit integer
nisplus.dticks	dticks	Unsigned 32-bit integer
nisplus.dump.dir	directory	String
nisplus.dump.time	time	Date/Time stamp
nisplus.endpoint	endpoint	No value
nisplus.endpoint.family	family	String
nisplus.endpoint.proto	proto	String
nisplus.endpoint.uaddr	addr	String
nisplus.endpoints	nis endpoints	No value
nisplus.entry	entry	No value
nisplus.entry.col	column	No value
nisplus.entry.cols	columns	No value
nisplus.entry.flags	flags	Unsigned 32-bit integer
nisplus.entry.flags.asn	ASN.1	Boolean
nisplus.entry.flags.binary	BINARY	Boolean
nisplus.entry.flags.encrypted	ENCRYPTED	Boolean
nisplus.entry.flags.modified	MODIFIED	Boolean
nisplus.entry.flags.xdr	XDR	Boolean
nisplus.entry.type	type	String
nisplus.entry.val	val	String
nisplus.fd.dir.data	data	Byte array
nisplus.fd.dirname	dirname	String
nisplus.fd.requester	requester	String

Field	Field Name	Type
nisplus.fd.sig	signature	Byte array
nisplus.group	Group	No value
nisplus.group.flags	flags	Unsigned 32-bit integer
nisplus.group.name	group name	String
nisplus.grps	Groups	No value
nisplus.ib.bufsize	bufsize	Unsigned 32-bit integer
nisplus.ib.flags	flags	Unsigned 32-bit integer
nisplus.key.data	key data	Byte array
nisplus.key.type	type	Unsigned 32-bit integer
nisplus.link	link	No value
nisplus.log.entries	log entries	No value
nisplus.log.entry	log entry	No value
nisplus.log.entry.type	type	Unsigned 32-bit integer
nisplus.log.principal	principal	String
nisplus.log.time	time	Date/Time stamp
nisplus.mtime	mtime	Date/Time stamp
nisplus.object	NIS Object	No value
nisplus.object.domain	domain	String
nisplus.object.group	group	String
nisplus.object.name	name	String
nisplus.object.oid	Object Identity Verifier	No value
nisplus.object.owner	owner	String
nisplus.object.private	private	Byte array
nisplus.object.ttl	ttl	Unsigned 32-bit integer
nisplus.object.type	type	Unsigned 32-bit integer
nisplus.ping.dir	directory	String
nisplus.ping.time	time	Date/Time stamp
nisplus.server	server	No value
nisplus.server.name	name	String
nisplus.servers	nis servers	No value
nisplus.status	status	Unsigned 32-bit integer
nisplus.table	table	No value
nisplus.table.col	column	No value
nisplus.table.col.flags	flags	No value
nisplus.table.col.name	column name	String
nisplus.table.cols	columns	No value
nisplus.table.flags.asn	asn	Boolean

Field	Field Name	Type
nisplus.table.flags.binary	binary	Boolean
nisplus.table.flags.casesensitive	casesensitive	Boolean
nisplus.table.flags.encrypted	encrypted	Boolean
nisplus.table.flags.modified	modified	Boolean
nisplus.table.flags.searchable	searchable	Boolean
nisplus.table.flags.xdr	xdr	Boolean
nisplus.table.maxcol	max columns	Unsigned 16-bit integer
nisplus.table.path	path	String
nisplus.table.separator	separator	Unsigned 8-bit integer
nisplus.table.type	type	String
nisplus.tag	tag	No value
nisplus.tag.type	type	Unsigned 32-bit integer
nisplus.tag.value	value	String
nisplus.taglist	taglist	No value
nisplus.zticks	zticks	Unsigned 32-bit integer

Table A-120. NIS+ (nisplus)

A.121. NIS+ Callback (nispluscb)

Field	Field Name	Type
nispluscb.entries	entries	No value
nispluscb.entry	entry	No value

Table A-121. NIS+ Callback (nispluscb)

A.122. Name Binding Protocol (nbp)

Field	Field Name	Type
nbp.count	Count	Unsigned 8-bit integer

Field	Field Name	Type
nbp.enum	Enumerator	Unsigned 8-bit integer
nbp.info	Info	Unsigned 8-bit integer
nbp.net	Network	Unsigned 16-bit integer
nbp.node	Node	Unsigned 8-bit integer
nbp.object	Object	String
nbp.op	Operation	Unsigned 8-bit integer
nbp.port	Port	Unsigned 8-bit integer
nbp.tid	Transaction ID	Unsigned 8-bit integer
nbp.type	Type	String
nbp.zone	Zone	String

Table A-122. Name Binding Protocol (nbp)

A.123. Name Management Protocol over IPX (nmpi)

Field	Field Name	Type

Table A-123. Name Management Protocol over IPX (nmpi)

A.124. NetBIOS (netbios)

Field	Field Name	Type
netbios.ack	Acknowledge	Boolean
netbios.ack_expected	Acknowledge expected	Boolean
netbios.ack_with_data	Acknowledge with data	Boolean
netbios.call_name_type	Call Name Type	Signed 16-bit integer
netbios.command	Command	Unsigned 16-bit integer
netbios.data2	DATA2 value	Unsigned 16-bit integer
netbios.hdr_len	Header Length	Unsigned 16-bit integer
netbios.largest_frame	Largest Frame	Unsigned 8-bit integer
netbios.local_session	Local Session No.	Unsigned 8-bit integer

Field	Field Name	Type
netbios.name	Netbios Name	String
netbios.name_type	Netbios Name Type	Unsigned 16-bit integer
netbios.recv_cont_req	RECEIVE_CONTINUE requested	Boolean
netbios.remote_session	Remote Session No.	Unsigned 8-bit integer
netbios.resp_corrl	Response Correlator	Signed 16-bit integer
netbios.send_no_ack	Handle SEND.NO.ACK	Boolean
netbios.version	NetBIOS Version	Boolean
netbios.xmit_corrl	Transmit Correlator	Signed 16-bit integer

Table A-124. NetBIOS (netbios)

A.125. NetBIOS Datagram Service (nbdgm)

Field	Field Name	Type
nbdgm.dgram_id	Datagram ID	Unsigned 16-bit integer
nbdgm.first	First fragment	Boolean
nbdgm.next	Fragmented	Boolean
nbdgm.node_type	Node Type	Unsigned 8-bit integer
nbdgm.src.ip	Source IP	IPv4 address
nbdgm.src.port	Source Port	Unsigned 16-bit integer
nbdgm.type	Message Type	Unsigned 8-bit integer

Table A-125. NetBIOS Datagram Service (nbdgm)

A.126. NetBIOS Name Service (nbns)

Field	Field Name	Type
nbns.count.add_rr	Additional RRs	Unsigned 16-bit integer
nbns.count.answers	Answer RRs	Unsigned 16-bit integer
nbns.count.auth_rr	Authority RRs	Unsigned 16-bit integer
nbns.count.queries	Questions	Unsigned 16-bit integer
nbns.id	Transaction ID	Unsigned 16-bit integer
nbns.query	Query	Boolean

Field	Field Name	Type
nbns.response	Response	Boolean

Table A-126. NetBIOS Name Service (nbns)

A.127. NetBIOS Session Service (nbss)

Field	Field Name	Type
nbss.flags	Flags	Unsigned 8-bit integer
nbss.type	Message Type	Unsigned 8-bit integer

Table A-127. NetBIOS Session Service (nbss)

A.128. NetBIOS over IPX (nbipx)

Field	Field Name	Type

Table A-128. NetBIOS over IPX (nbipx)

A.129. NetWare Core Protocol (ncp)

Field	Field Name	Type
ncp.accepted_max_size	Accepted Max Size	Unsigned 16-bit integer
ncp.acct_version	Acct Version	Unsigned 8-bit integer
ncp.buffer_size	Buffer Size	Unsigned 16-bit integer
ncp.completion_code	Completion Code	Unsigned 8-bit integer
ncp.connection	Connection Number	Unsigned 16-bit integer
ncp.connection_number	Connection Number	Unsigned 32-bit integer
ncp.connection_status	Connection Status	Unsigned 8-bit integer
ncp.connections_in_use	Connections In Use	Unsigned 16-bit integer
ncp.connections_max_used	Connections Max Used	Unsigned 16-bit integer

Field	Field Name	Type
ncp.connections_supported_max	Connections Supported Max	Unsigned 16-bit integer
ncp.echo_socket	Echo Socket	Unsigned 16-bit integer
ncp.file_handle	File Handle	Byte array
ncp.file_offset	File Offset	Unsigned 32-bit integer
ncp.file_size	File Size	Unsigned 32-bit integer
ncp.func	Function	Unsigned 8-bit integer
ncp.internet_bridge_version	Internet Bridge Version	Unsigned 8-bit integer
ncp.ip.length	NCP over IP length	Unsigned 32-bit integer
ncp.ip.replybufsize	NCP over IP Reply Buffer Size	Unsigned 32-bit integer
ncp.ip.signature	NCP over IP signature	Unsigned 32-bit integer
ncp.ip.version	NCP over IP Version	Unsigned 32-bit integer
ncp.job_type	Job Type	Unsigned 16-bit integer
ncp.length	Packet Length	Unsigned 16-bit integer
ncp.local_login_info_ccode	Local Login Info C Code	Unsigned 8-bit integer
ncp.max_bytes	Maximum Number of Bytes	Unsigned 16-bit integer
ncp.mixed_mode_path_flag	Mixed Mode Path Flag	Unsigned 8-bit integer
ncp.num_bytes	Number of Bytes	Unsigned 16-bit integer
ncp.object_flags	Object Flags	Unsigned 8-bit integer
ncp.object_has_properites	Object Has Properties	Unsigned 8-bit integer
ncp.object_id	Object ID	Unsigned 32-bit integer
ncp.object_name	Object Name	
ncp.object_name1	Object Name	String
ncp.object_security	Object Security	Unsigned 8-bit integer
ncp.object_type	Object Type	Unsigned 16-bit integer
ncp.os_language_id	OS Language ID	Unsigned 8-bit integer
ncp.os_major_version	OS Major Version	Unsigned 8-bit integer
ncp.os_minor_version	OS Minor Version	Unsigned 8-bit integer
ncp.os_revision	OS Revision	Unsigned 8-bit integer
ncp.ping_version	Ping Version	Unsigned 16-bit integer
ncp.print_server_version	Print Server Version	Unsigned 8-bit integer

Field	Field Name	Type
ncp.product_major_version	Product Major Version	Unsigned 16-bit integer
ncp.product_minor_version	Product Minor Version	Unsigned 16-bit integer
ncp.product_revision_version	Product Revision Version	Unsigned 8-bit integer
ncp.property_data	Property Data	Byte array
ncp.property_has_more_segments	Property Has More Segments	Unsigned 8-bit integer
ncp.property_name	Property Name	
ncp.property_segment	Property Segment	Unsigned 8-bit integer
ncp.property_type	Property Type	Unsigned 8-bit integer
ncp.proposed_max_size	Proposed Max Size	Unsigned 16-bit integer
ncp.qms_version	QMS Version	Unsigned 8-bit integer
ncp.reserved3	Reserved	Byte array
ncp.reserved51	Reserved	Byte array
ncp.security_flag	Security Flag	Unsigned 8-bit integer
ncp.security_restriction_version	Security Restriction Version	Unsigned 8-bit integer
ncp.seq	Sequence Number	Unsigned 8-bit integer
ncp.server_name	Server Name	String
ncp.sft_level	SFT Level	Unsigned 8-bit integer
ncp.subfunc	SubFunction	Unsigned 8-bit integer
ncp.task	Task Number	Unsigned 8-bit integer
ncp.task_number	Task Number	Unsigned 32-bit integer
ncp.tts_level	TTS Level	Unsigned 8-bit integer
ncp.type	Type	Unsigned 16-bit integer
ncp.unknown_byte	Unknown Byte	Unsigned 8-bit integer
ncp.vap_version	VAP Version	Unsigned 8-bit integer
ncp.virtual_console_version	Virtual Console Version	Unsigned 8-bit integer
ncp.volumes_supported_max	Volumes Supported Max	Unsigned 16-bit integer

Table A-129. NetWare Core Protocol (ncp)

A.130. Network File System (nfs)

Field	Field Name	Type
nfs.ace	ace	String
nfs.aceflag4	aceflag	Unsigned 32-bit integer
nfs.acemask4	acemask	Unsigned 32-bit integer
nfs.acetype4	acetype	Unsigned 32-bit integer
nfs.attr	mand_attr	Unsigned 32-bit integer
nfs.bytes_per_block	bytes_per_block	Unsigned 32-bit integer
nfs.call.operation	Opcode	Unsigned 32-bit integer
nfs.cb_location	cb_location	Unsigned 32-bit integer
nfs.cb_program	cb_program	Unsigned 32-bit integer
nfs.change_info.atomic	Atomic	Boolean
nfs.changeid4	changeid	Unsigned 32-bit integer
nfs.clientid	clientid	Unsigned 32-bit integer
nfs.clientid.verifier	verifier	Unsigned 32-bit integer
nfs.cookie3	cookie	Unsigned 32-bit integer
nfs.cookie4	cookie	Unsigned 32-bit integer
nfs.cookieverf4	cookieverf	Unsigned 32-bit integer
nfs.count3	count	Unsigned 32-bit integer
nfs.count3_dircount	dircount	Unsigned 32-bit integer
nfs.count3_maxcount	maxcount	Unsigned 32-bit integer
nfs.count4	count	Unsigned 32-bit integer
nfs.createmode	Create Mode	Unsigned 32-bit integer
nfs.data	Data	Byte array
nfs.data_follows	data_follows	Boolean
nfs.delegate_stateid	delegate_stateid	Unsigned 32-bit integer
nfs.delegate_type	delegate_type	Unsigned 32-bit integer
nfs.dircount	dircount	Unsigned 32-bit integer
nfs.dirlist4.eof	eof	Boolean
nfs.eof	eof	Unsigned 32-bit integer
nfs.fattr.blocks	blocks	Unsigned 32-bit integer
nfs.fattr.blocksize	blocksize	Unsigned 32-bit integer
nfs.fattr.fileid	fileid	Unsigned 32-bit integer
nfs.fattr.fsid	fsid	Unsigned 32-bit integer
nfs.fattr.gid	gid	Unsigned 32-bit integer
nfs.fattr.nlink	nlink	Unsigned 32-bit integer

Field	Field Name	Type
nfs.fattr.rdev	rdev	Unsigned 32-bit integer
nfs.fattr.size	size	Unsigned 32-bit integer
nfs.fattr.type	type	Unsigned 32-bit integer
nfs.fattr.uid	uid	Unsigned 32-bit integer
nfs.fattr3.fileid	fileid	Unsigned 32-bit integer
nfs.fattr3.fsid	fsid	Unsigned 32-bit integer
nfs.fattr3.gid	gid	Unsigned 32-bit integer
nfs.fattr3.nlink	nlink	Unsigned 32-bit integer
nfs.fattr3.rdev	rdev	Unsigned 32-bit integer
nfs.fattr3.size	size	Unsigned 32-bit integer
nfs.fattr3.type	Type	Unsigned 32-bit integer
nfs.fattr3.uid	uid	Unsigned 32-bit integer
nfs.fattr3.used	used	Unsigned 32-bit integer
nfs.fattr4.aclsupport	aclsupport	Unsigned 32-bit integer
nfs.fattr4.attr_vals	attr_vals	Byte array
nfs.fattr4.fileid	fileid	Unsigned 32-bit integer
nfs.fattr4.files_avail	files_avail	Unsigned 32-bit integer
nfs.fattr4.files_free	files_free	Unsigned 32-bit integer
nfs.fattr4.files_total	files_total	Unsigned 32-bit integer
nfs.fattr4.lease_time	lease_time	Unsigned 32-bit integer
nfs.fattr4.maxfilesize	maxfilesize	Unsigned 32-bit integer
nfs.fattr4.maxlink	maxlink	Unsigned 32-bit integer
nfs.fattr4.maxname	maxname	Unsigned 32-bit integer
nfs.fattr4.maxread	maxread	Unsigned 32-bit integer
nfs.fattr4.maxwrite	maxwrite	Unsigned 32-bit integer
nfs.fattr4.numlinks	numlinks	Unsigned 32-bit integer
nfs.fattr4.quota_hard	quota_hard	Unsigned 32-bit integer
nfs.fattr4.quota_soft	quota_soft	Unsigned 32-bit integer
nfs.fattr4.quota_used	quota_used	Unsigned 32-bit integer
nfs.fattr4.size	size	Unsigned 32-bit integer
nfs.fattr4.space_avail	space_avail	Unsigned 32-bit integer
nfs.fattr4.space_free	space_free	Unsigned 32-bit integer
nfs.fattr4.space_total	space_total	Unsigned 32-bit integer
nfs.fattr4.space_used	space_used	Unsigned 32-bit integer
nfs.fattr4_archive	fattr4_archive	Boolean
nfs.fattr4_cansettime	fattr4_cansettime	Boolean

Field	Field Name	Type
nfs.fattr4_case_insensitive	fattr4_case_insensitive	Boolean
nfs.fattr4_case_preserving	fattr4_case_preserving	Boolean
nfs.fattr4_chown_restricted	fattr4_chown_restricted	Boolean
nfs.fattr4_expire_type	fattr4_expire_type	Unsigned 32-bit integer
nfs.fattr4_hidden	fattr4_hidden	Boolean
nfs.fattr4_homogeneous	fattr4_homogeneous	Boolean
nfs.fattr4_link_support	fattr4_link_support	Boolean
nfs.fattr4_mimetype	fattr4_mimetype	String
nfs.fattr4_named_attr	fattr4_named_attr	Boolean
nfs.fattr4_no_trunc	fattr4_no_trunc	Boolean
nfs.fattr4_owner	fattr4_owner	String
nfs.fattr4_owner_group	fattr4_owner_group	String
nfs.fattr4_symlink_support	fattr4_symlink_support	Boolean
nfs.fattr4_system	fattr4_system	Boolean
nfs.fattr4_unique_handles	fattr4_unique_handles	Boolean
nfs.fh.auth_type	auth_type	Unsigned 8-bit integer
nfs.fh.dentry	dentry	Unsigned 32-bit integer
nfs.fh.dev	device	Unsigned 32-bit integer
nfs.fh.dirinode	directory inode	Unsigned 32-bit integer
nfs.fh.fileid_type	fileid_type	Unsigned 8-bit integer
nfs.fh.fn	file number	Unsigned 32-bit integer
nfs.fh.fn.generation	generation	Unsigned 32-bit integer
nfs.fh.fn.inode	inode	Unsigned 32-bit integer
nfs.fh.fn.len	length	Unsigned 32-bit integer
nfs.fh.fsid.inode	inode	Unsigned 32-bit integer
nfs.fh.fsid.major	major	Unsigned 32-bit integer
nfs.fh.fsid.minor	minor	Unsigned 32-bit integer
nfs.fh.fsid_type	fsid_type	Unsigned 8-bit integer
nfs.fh.fstype	file system type	Unsigned 32-bit integer
nfs.fh.hp.len	length	Unsigned 32-bit integer
nfs.fh.length	length	Unsigned 32-bit integer
nfs.fh.pinode	pseudo inode	Unsigned 32-bit integer
nfs.fh.version	version	Unsigned 8-bit integer

Field	Field Name	Type
nfs.fh.xdev	exported device	Unsigned 32-bit integer
nfs.fh.xfn	exported file number	Unsigned 32-bit integer
nfs.fh.xfn.generation	generation	Unsigned 32-bit integer
nfs.fh.xfn.inode	exported inode	Unsigned 32-bit integer
nfs.fh.xfn.len	length	Unsigned 32-bit integer
nfs.fh.xfsid.major	exported major	Unsigned 32-bit integer
nfs.fh.xfsid.minor	exported minor	Unsigned 32-bit integer
nfs.filesize	filesize	Unsigned 32-bit integer
nfs.fsid4.major	fsid4.major	Unsigned 32-bit integer
nfs.fsid4.minor	fsid4.minor	Unsigned 32-bit integer
nfs.fsinfo.dtpref	dtpref	Unsigned 32-bit integer
nfs.fsinfo.maxfilesize	maxfilesize	Unsigned 32-bit integer
nfs.fsinfo.propeties	Properties	Unsigned 32-bit integer
nfs.fsinfo.rtmax	rtmax	Unsigned 32-bit integer
nfs.fsinfo.rtmult	rtmult	Unsigned 32-bit integer
nfs.fsinfo.rtpref	rtpref	Unsigned 32-bit integer
nfs.fsinfo.wtmax	wtmax	Unsigned 32-bit integer
nfs.fsinfo.wtmult	wtmult	Unsigned 32-bit integer
nfs.fsinfo.wtpref	wtpref	Unsigned 32-bit integer
nfs.fsstat.invarsec	invarsec	Unsigned 32-bit integer
nfs.fsstat3_resok.abytes	abytes	Unsigned 32-bit integer
nfs.fsstat3_resok.afiles	afiles	Unsigned 32-bit integer
nfs.fsstat3_resok.fbytes	fbytes	Unsigned 32-bit integer
nfs.fsstat3_resok.ffiles	ffiles	Unsigned 32-bit integer
nfs.fsstat3_resok.tbytes	tbytes	Unsigned 32-bit integer
nfs.fsstat3_resok.tfiles	tfiles	Unsigned 32-bit integer
nfs.gid3	gid	Unsigned 32-bit integer
nfs.length4	length	Unsigned 32-bit integer
nfs.locktype4	locktype	Unsigned 32-bit integer
nfs.maxcount	maxcount	Unsigned 32-bit integer
nfs.minorversion	minorversion	Unsigned 32-bit integer
nfs.name	Name	String
nfs.nfs_ftype4	nfs_ftype4	Unsigned 32-bit integer
nfs.nfstime4.nseconds	nseconds	Unsigned 32-bit integer
nfs.nfstime4.seconds	seconds	Unsigned 32-bit integer
nfs.num_blocks	num_blocks	Unsigned 32-bit integer
nfs.offset3	offset	Unsigned 32-bit integer

Field	Field Name	Type
nfs.offset4	offset	Unsigned 32-bit integer
nfs.open.claim_type	Claim Type	Unsigned 32-bit integer
nfs.open.delegation_type	Delegation Type	Unsigned 32-bit integer
nfs.open.limit_by	Space Limit	Unsigned 32-bit integer
nfs.open.opentype	Open Type	Unsigned 32-bit integer
nfs.open4.share_access	share_access	Unsigned 32-bit integer
nfs.open4.share_deny	share_deny	Unsigned 32-bit integer
nfs.pathconf.case_insensitive	case_insensitive	Boolean
nfs.pathconf.case_preserving	case_preserving	Boolean
nfs.pathconf.chown_restricted	chown_restricted	Boolean
nfs.pathconf.linkmax	linkmax	Unsigned 32-bit integer
nfs.pathconf.name_max	name_max	Unsigned 32-bit integer
nfs.pathconf.no_trunc	no_trunc	Boolean
nfs.pathname.component	Filename	String
nfs.read.count	Count	Unsigned 32-bit integer
nfs.read.eof	EOF	Boolean
nfs.read.offset	Offset	Unsigned 32-bit integer
nfs.read.totalcount	Total Count	Unsigned 32-bit integer
nfs.readdir.cookie	Cookie	Unsigned 32-bit integer
nfs.readdir.count	Count	Unsigned 32-bit integer
nfs.readdir.entry	Entry	No value
nfs.readdir.entry.cookie	Cookie	Unsigned 32-bit integer
nfs.readdir.entry.fileid	File ID	Unsigned 32-bit integer
nfs.readdir.entry.name	Name	String
nfs.readdir.entry3.cookie	Cookie	Unsigned 32-bit integer
nfs.readdir.entry3.fileid	File ID	Unsigned 32-bit integer
nfs.readdir.entry3.name	Name	String
nfs.readdir.eof	EOF	Unsigned 32-bit integer
nfs.readdirplus.entry.cookie	Cookie	Unsigned 32-bit integer
nfs.readdirplus.entry.fileid	File ID	Unsigned 32-bit integer
nfs.readdirplus.entry.name	Name	String

Field	Field Name	Type
nfs.readlink.data	Data	String
nfs.recall	EOF	Boolean
nfs.recall4	recall	Boolean
nfs.reclaim4	reclaim	Unsigned 32-bit integer
nfs.reply.operation	Opcode	Unsigned 32-bit integer
nfs.secinfo_flavor	secinfo_flavor	Unsigned 32-bit integer
nfs.seqid	seqid	Unsigned 32-bit integer
nfs.server	server	String
nfs.set_it	set_it	Unsigned 32-bit integer
nfs.set_size3.size	size	Unsigned 32-bit integer
nfs.specdata1	specdata1	Unsigned 32-bit integer
nfs.specdata2	specdata2	Unsigned 32-bit integer
nfs.stable_how4	stable_how4	Unsigned 32-bit integer
nfs.stateid4	stateid	Unsigned 32-bit integer
nfs.statfs.bavail	Available Blocks	Unsigned 32-bit integer
nfs.statfs.bfree	Free Blocks	Unsigned 32-bit integer
nfs.statfs.blocks	Total Blocks	Unsigned 32-bit integer
nfs.statfs.bsize	Block Size	Unsigned 32-bit integer
nfs.statfs.tsize	Transfer Size	Unsigned 32-bit integer
nfs.status	Status	Unsigned 32-bit integer
nfs.status2	Status	Unsigned 32-bit integer
nfs.symlink.linktext	Name	String
nfs.symlink.to	To	String
nfs.tag	Tag	String
nfs.type	Type	Unsigned 32-bit integer
nfs.uid3	uid	Unsigned 32-bit integer
nfs.verifier4	verifier	Unsigned 32-bit integer
nfs.wcc_attr.size	size	Unsigned 32-bit integer
nfs.who	who	String
nfs.write.beginoffset	Begin Offset	Unsigned 32-bit integer
nfs.write.committed	Committed	Unsigned 32-bit integer
nfs.write.offset	Offset	Unsigned 32-bit integer
nfs.write.stable	Stable	Unsigned 32-bit integer
nfs.write.totalcount	Total Count	Unsigned 32-bit integer

Table A-130. Network File System (nfs)

A.131. Network Lock Manager Protocol (nlm)

Field	Field Name	Type
nlm.block	block	Boolean
nlm.cookie	cookie	Byte array
nlm.exclusive	exclusive	Boolean
nlm.holder	holder	No value
nlm.lock	lock	No value
nlm.lock.caller_name	caller_name	String
nlm.lock.l_len	l_len	Unsigned 32-bit integer
nlm.lock.l_offset	l_offset	Unsigned 32-bit integer
nlm.lock.owner	owner	Byte array
nlm.lock.svid	svid	Unsigned 32-bit integer
nlm.reclaim	reclaim	Boolean
nlm.sequence	sequence	Signed 32-bit integer
nlm.share	share	No value
nlm.share.access	access	Unsigned 32-bit integer
nlm.share.mode	mode	Unsigned 32-bit integer
nlm.share.name	name	String
nlm.state	state	Unsigned 32-bit integer
nlm.test_stat	test_stat	No value
nlm.test_stat.stat	stat	Unsigned 32-bit integer

Table A-131. Network Lock Manager Protocol (nlm)

A.132. Network News Transfer Protocol (nntp)

Field	Field Name	Type
nntp.request	Request	Boolean
nntp.response	Response	Boolean

Table A-132. Network News Transfer Protocol (nntp)

A.133. Network Status Monitor CallBack Protocol

(stat-cb)

Field	Field Name	Type
statnotify.name	Name	String
statnotify.priv	Priv	Byte array
statnotify.state	State	Unsigned 32-bit integer

Table A-133. Network Status Monitor CallBack Protocol (stat-cb)

A.134. Network Status Monitor Protocol (stat)

Field	Field Name	Type
stat.mon	Monitor	No value
stat.mon_id.name	Monitor ID Name	String
stat.my_id	My ID	No value
stat.my_id.hostname	Hostname	String
stat.my_id.proc	Procedure	Unsigned 32-bit integer
stat.my_id.prog	Program	Unsigned 32-bit integer
stat.my_id.vers	Version	Unsigned 32-bit integer
stat.name	Name	String
stat.priv	Priv	Byte array
stat.stat_chge	Status Change	No value
stat.stat_res	Status Result	No value
stat.stat_res.res	Result	Unsigned 32-bit integer
stat.stat_res.state	State	Unsigned 32-bit integer
stat.state	State	Unsigned 32-bit integer

Table A-134. Network Status Monitor Protocol (stat)

A.135. Network Time Protocol (ntp)

Field	Field Name	Type
ntp.flags	Flags	Unsigned 8-bit integer
ntp.flags.li	Leap Indicator	Unsigned 8-bit integer

Field	Field Name	Type
ntp.flags.mode	Mode	Unsigned 8-bit integer
ntp.flags.vn	Version number	Unsigned 8-bit integer
ntp.keyid	Key ID	Byte array
ntp.mac	Message Authentication Code	Byte array
ntp.org	Originate Time Stamp	Byte array
ntp.ppoll	Peer Polling Interval	Unsigned 8-bit integer
ntp.precision	Peer Clock Precision	Unsigned 8-bit integer
ntp.rec	Receive Time Stamp	Byte array
ntp.refid	Reference Clock ID	Byte array
ntp.reftime	Reference Clock Update Time	Byte array
ntp.rootdelay	Root Delay	Double-precision floating point
ntp.rootdispersion	Clock Dispersion	Double-precision floating point
ntp.stratum	Peer Clock Stratum	Unsigned 8-bit integer
ntp.xmt	Transmit Time Stamp	Byte array

Table A-135. Network Time Protocol (ntp)

A.136. Null/Loopback (null)

Field	Field Name	Type
null.family	Family	Unsigned 32-bit integer
null.type	Type	Unsigned 16-bit integer

Table A-136. Null/Loopback (null)

A.137. Open Shortest Path First (ospf)

Field	Field Name	Type

Table A-137. Open Shortest Path First (ospf)

A.138. PPP IP Control Protocol (ipcp)

Field	Field Name	Type

Table A-138. PPP IP Control Protocol (ipcp)

A.139. PPP Link Control Protocol (lcp)

Field	Field Name	Type

Table A-139. PPP Link Control Protocol (lcp)

A.140. PPP Multilink Protocol (mp)

Field	Field Name	Type
mp.first	First fragment	Boolean
mp.last	Last fragment	Boolean
mp.seq	Sequence number	Unsigned 24-bit integer

Table A-140. PPP Multilink Protocol (mp)

A.141. PPP Password Authentication Protocol (pap)

Field	Field Name	Type

Table A-141. PPP Password Authentication Protocol (pap)

A.142. PPP-over-Ethernet Discovery (pppoed)

Field	Field Name	Type

Table A-142. PPP-over-Ethernet Discovery (pppoed)

A.143. PPP-over-Ethernet Session (pppoes)

Field	Field Name	Type

Table A-143. PPP-over-Ethernet Session (pppoes)

A.144. Point-to-Point Protocol (ppp)

Field	Field Name	Type

Table A-144. Point-to-Point Protocol (ppp)

A.145. Point-to-Point Tunnelling Protocol (pptp)

Field	Field Name	Type
pptp.type	Message type	Unsigned 16-bit integer

Table A-145. Point-to-Point Tunnelling Protocol (pptp)

A.146. Portmap (portmap)

Field	Field Name	Type
portmap.answer	Answer	Boolean
portmap.args	Arguments	Byte array
portmap.port	Port	Unsigned 32-bit integer
portmap.proc	Procedure	Unsigned 32-bit integer
portmap.prog	Program	Unsigned 32-bit integer
portmap.proto	Protocol	Unsigned 32-bit integer
portmap.result	Result	Byte array
portmap.rpcb	RPCB	No value
portmap.rpcb.addr	Universal Address	String
portmap.rpcb.netid	Network Id	String
portmap.rpcb.owner	Owner of this Service	String
portmap.rpcb.prog	Program	Unsigned 32-bit integer
portmap.rpcb.version	Version	Unsigned 32-bit integer
portmap.uaddr	Universal Address	String
portmap.version	Version	Unsigned 32-bit integer

Table A-146. Portmap (portmap)

A.147. Post Office Protocol (pop)

Field	Field Name	Type
pop.request	Request	Boolean
pop.response	Response	Boolean

Table A-147. Post Office Protocol (pop)

A.148. Pragmatic General Multicast (pgm)

Field	Field Name	Type
pgm.data.sqn	Data Packet Sequence Number	Unsigned 32-bit integer
pgm.data.trail	Trailing Edge Sequence Number	Unsigned 32-bit integer
pgm.genopts.len	Length	Unsigned 8-bit integer

Field	Field Name	Type
pgm.genopts.opx	Option Extensibility Bits	Unsigned 8-bit integer
pgm.genopts.type	Type	Unsigned 8-bit integer
pgm.hdr.cksum	Checksum	Unsigned 8-bit integer
pgm.hdr.dport	Destination Port	Unsigned 16-bit integer
pgm.hdr.gsi	Global Source Identifier	Byte array
pgm.hdr.opts	Options	Unsigned 8-bit integer
pgm.hdr.opts.netsig	Network Significant Options	Boolean
pgm.hdr.opts.opt	Options	Boolean
pgm.hdr.opts.parity	Parity	Boolean
pgm.hdr.opts.varlen	Variable lenght Parity Packet Option	Boolean
pgm.hdr.sport	Source Port	Unsigned 16-bit integer
pgm.hdr.tsdulen	Transport Service Data Unit	Unsigned 8-bit integer
pgm.hdr.type	Type	Unsigned 8-bit integer
pgm.nak.grp	Multicast Group NLA	Unsigned 32-bit integer
pgm.nak.grpafi	Multicast group AFI	Unsigned 16-bit integer
pgm.nak.grpres	Reserved	Unsigned 16-bit integer
pgm.nak.sqn	Requested Sequence Number	Unsigned 32-bit integer
pgm.nak.src	Source NLA	Unsigned 32-bit integer
pgm.nak.srcafi	Source Network Layer Address (Family Indicator)	Unsigned 16-bit integer
pgm.nak.srcres	Reserved	Unsigned 16-bit integer
pgm.opt.nak.type	Type	Unsigned 8-bit integer
pgm.opts.join.min_join	Minimum Sequence Number	Unsigned 32-bit integer
pgm.opts.join.opx	Option Extensibility Bits	Unsigned 8-bit integer
pgm.opts.join.res	Length	Unsigned 8-bit integer
pgm.opts.join.type	Type	Unsigned 8-bit integer
pgm.opts.len	Length	Unsigned 8-bit integer
pgm.opts.nak.len	Length	Unsigned 8-bit integer
pgm.opts.nak.list	List	Byte array
pgm.opts.nak.op	Reserved	Unsigned 8-bit integer
pgm.opts.nak.opx	Option Extensibility Bits	Unsigned 8-bit integer

Field	Field Name	Type
pgm.opts.parity_prm.op	Pro-Active Parity	Unsigned 8-bit integer
pgm.opts.parity_prm.opx	Option Extensibility Bits	Unsigned 8-bit integer
pgm.opts.parity_prm.prm_grp	Transmission Group Size	Unsigned 32-bit integer
pgm.opts.tlen	Total Length	Unsigned 16-bit integer
pgm.opts.type	Type	Unsigned 8-bit integer
pgm.parity_grp.len	Length	Unsigned 8-bit integer
pgm.parity_grp.type	Type	Unsigned 8-bit integer
pgm.parity_prm.len	Length	Unsigned 8-bit integer
pgm.parity_prm.type	Type	Unsigned 8-bit integer
pgm.spm.lead	Leading Edge Sequence Number	Unsigned 32-bit integer
pgm.spm.path	Path NLA	Unsigned 32-bit integer
pgm.spm.pathafi	NLA AFI (IPv4 is set to 1)	Unsigned 16-bit integer
pgm.spm.res	Reserved	Unsigned 16-bit integer
pgm.spm.sqn	Sequence number	Unsigned 32-bit integer
pgm.spm.trail	Trailing Edge Sequence Number	Unsigned 32-bit integer

Table A-148. Pragmatic General Multicast (pgm)

A.149. Protocol Independent Multicast (pim)

Field	Field Name	Type
pim.cksum	Checksum	Unsigned 16-bit integer
pim.code	Code	Unsigned 8-bit integer
pim.type	Type	Unsigned 8-bit integer
pim.version	Version	Unsigned 8-bit integer

Table A-149. Protocol Independent Multicast (pim)

A.150. Q.2931 (q2931)

Field	Field Name	Type
q2931.call_ref	Call reference value	Byte array
q2931.call_ref_len	Call reference value length	Unsigned 8-bit integer
q2931.disc	Protocol discriminator	Unsigned 8-bit integer
q2931.message_action_indicator	Action indicator	Unsigned 8-bit integer
q2931.message_flag	Flag	Boolean
q2931.message_len	Message length	Unsigned 16-bit integer
q2931.message_type	Message type	Unsigned 8-bit integer
q2931.message_type_ext	Message type extension	Unsigned 8-bit integer

Table A-150. Q.2931 (q2931)

A.151. Q.931 (q931)

Field	Field Name	Type
q931.call_ref	Call reference value	Byte array
q931.call_ref_len	Call reference value length	Unsigned 8-bit integer
q931.disc	Protocol discriminator	Unsigned 8-bit integer
q931.message_type	Message type	Unsigned 8-bit integer

Table A-151. Q.931 (q931)

A.152. Quake II Network Protocol (quake2)

Field	Field Name	Type
quake2.c2s	Client to Server	Unsigned 32-bit integer
quake2.connectionless	Connectionless	Unsigned 32-bit integer
quake2.connectionless.marker	Marker	Unsigned 32-bit integer
quake2.connectionless.text	Text	String
quake2.game	Game	Unsigned 32-bit integer

Field	Field Name	Type
quake2.game.qport	QPort	Unsigned 32-bit integer
quake2.game.rel1	Reliable	Boolean
quake2.game.rel2	Reliable	Boolean
quake2.game.seq1	Sequence Number	Unsigned 32-bit integer
quake2.game.seq2	Sequence Number	Unsigned 32-bit integer
quake2.s2c	Server to Client	Unsigned 32-bit integer

Table A-152. Quake II Network Protocol (quake2)

A.153. Quake Network Protocol (quake)

Field	Field Name	Type
quake.control.accept.port	Port	Unsigned 32-bit integer
quake.control.command	Command	Unsigned 8-bit integer
quake.control.connect.game	Game	String
quake.control.connect.version	Version	Unsigned 8-bit integer
quake.control.player_info.address	Address	String
quake.control.player_info.colors	Colors	Unsigned 32-bit integer
quake.control.player_info.pants	Pants	Unsigned 8-bit integer
quake.control.player_info.shirt	Shirt	Unsigned 8-bit integer
quake.control.player_info.connect_time	Connect Time	Unsigned 32-bit integer
quake.control.player_info.frags	Frags	Unsigned 32-bit integer
quake.control.player_info.name	Name	String
quake.control.player_info.player	Player	Unsigned 8-bit integer
quake.control.reject.reason	Reason	String

Field	Field Name	Type
quake.control.rule_info.lastrule	Last Rule	String
quake.control.rule_info.rule	Rule	String
quake.control.rule_info.value	Value	String
quake.control.server_info.address	Address	String
quake.control.server_info.game	Game	String
quake.control.server_info.map	Map	String
quake.control.server_info.maxplayers	Maximum Number of Players	Unsigned 8-bit integer
quake.control.server_info.numplayers	Number of Players	Unsigned 8-bit integer
quake.control.server_info.server	Server	String
quake.control.server_info.version	Version	Unsigned 8-bit integer
quake.header.flags	Flags	Unsigned 16-bit integer
quake.header.length	Length	Unsigned 16-bit integer
quake.header.sequence	Sequence	Unsigned 32-bit integer

Table A-153. Quake Network Protocol (quake)

A.154. QuakeWorld Network Protocol (quakeworld)

Field	Field Name	Type
quakeworld.c2s	Client to Server	Unsigned 32-bit integer
quakeworld.connectionless	Connectionless	Unsigned 32-bit integer
quakeworld.connectionless.marker	Marker	Unsigned 32-bit integer

Field	Field Name	Type
quakeworld.connectionless.text	Text	String
quakeworld.game	Game	Unsigned 32-bit integer
quakeworld.game.qport	QPort	Unsigned 32-bit integer
quakeworld.game.rel1	Reliable	Boolean
quakeworld.game.rel2	Reliable	Boolean
quakeworld.game.seq1	Sequence Number	Unsigned 32-bit integer
quakeworld.game.seq2	Sequence Number	Unsigned 32-bit integer
quakeworld.s2c	Server to Client	Unsigned 32-bit integer

Table A-154. QuakeWorld Network Protocol (quakeworld)

A.155. RFC 2250 MPEG1 (mpeg1)

Field	Field Name	Type
mpeg1.stream	MPEG-1 stream	Byte array
rtp.payload_mpeg_T	T	Unsigned 16-bit integer
rtp.payload_mpeg_an	AN	Unsigned 16-bit integer
rtp.payload_mpeg_b	Beginning-of-slice	Boolean
rtp.payload_mpeg_bfc	BFC	Unsigned 16-bit integer
rtp.payload_mpeg_fbv	FBV	Unsigned 16-bit integer
rtp.payload_mpeg_ffc	FFC	Unsigned 16-bit integer
rtp.payload_mpeg_ffv	FFV	Unsigned 16-bit integer
rtp.payload_mpeg_mbz	MBZ	Unsigned 16-bit integer
rtp.payload_mpeg_n	New Picture Header	Unsigned 16-bit integer
rtp.payload_mpeg_p	Picture type	Unsigned 16-bit integer
rtp.payload_mpeg_s	Sequence Header	Boolean
rtp.payload_mpeg_tr	Temporal Reference	Unsigned 16-bit integer

Table A-155. RFC 2250 MPEG1 (mpeg1)

A.156. RIPng (ripng)

Field	Field Name	Type
ripng.cmd	Command	Unsigned 8-bit integer
ripng.version	Version	Unsigned 8-bit integer

Table A-156. RIPng (ripng)

A.157. RX Protocol (rx)

Field	Field Name	Type
rx.ack	ACK Packet	No value
rx.ack_type	ACK Type	Unsigned 8-bit integer
rx.bufferspace	Bufferspace	Unsigned 16-bit integer
rx.callnumber	Call Number	Unsigned 32-bit integer
rx.challenge	CHALLENGE Packet	No value
rx.cid	CID	Unsigned 32-bit integer
rx.encrypted	Encrypted	No value
rx.epoch	Epoch	Date/Time stamp
rx.first	First Packet	Unsigned 32-bit integer
rx.flags	Flags	Unsigned 8-bit integer
rx.flags.client_init	Client Initiated	Unsigned 8-bit integer
rx.flags.free_packet	Free Packet	Unsigned 8-bit integer
rx.flags.last_packet	Last Packet	Unsigned 8-bit integer
rx.flags.more_packets	More Packets	Unsigned 8-bit integer
rx.flags.request_ack	Request Ack	Unsigned 8-bit integer
rx.if_mtu	Interface MTU	Unsigned 32-bit integer
rx.inc_nonce	Inc Nonce	Unsigned 32-bit integer
rx.kvno	kvno	Unsigned 32-bit integer
rx.level	Level	Unsigned 32-bit integer
rx.max_mtu	Max MTU	Unsigned 32-bit integer
rx.max_packets	Max Packets	Unsigned 32-bit integer
rx.maxskew	Max Skew	Unsigned 16-bit integer
rx.min_level	Min Level	Unsigned 32-bit integer
rx.nonce	Nonce	Unsigned 32-bit integer
rx.num_acks	Num ACKs	Unsigned 8-bit integer
rx.prev	Prev Packet	Unsigned 32-bit integer
rx.reason	Reason	Unsigned 8-bit integer

Field	Field Name	Type
rx.response	RESPONSE Packet	No value
rx.rwind	rwind	Unsigned 32-bit integer
rx.securityindex	Security Index	Unsigned 32-bit integer
rx.seq	Sequence Number	Unsigned 32-bit integer
rx.serial	Serial	Unsigned 32-bit integer
rx.serviceid	Service ID	Unsigned 16-bit integer
rx.spare	Spare/Checksum	Unsigned 16-bit integer
rx.ticket	ticket	Byte array
rx.ticket_len	Ticket len	Unsigned 32-bit integer
rx.type	Type	Unsigned 8-bit integer
rx.userstatus	User Status	Unsigned 32-bit integer
rx.version	Version	Unsigned 32-bit integer

Table A-157. RX Protocol (rx)

A.158. Radio Access Network Application Part (ranap)

Field	Field Name	Type
ranap.CN_DomainIndicator	CN-DomainIndicator	Unsigned 8-bit integer
ranap.Extension_Field_Value	Extension Field Value	Byte array
ranap.IuSigConId	IuSigConId	Byte array
ranap.NAS_PDU	NAS-PDU	Byte array
ranap.PLMN_ID	PLMN-ID	Byte array
ranap.ProtocolExtensionContainer_present	ProtocolExtensionContainer	Unsigned 8-bit integer
ranap.ProtocolExtensionFields_octets	Number of octets	Unsigned 16-bit integer
ranap.RAB_ID	RAB-ID	Unsigned 8-bit integer
ranap.RAB_SetupOrModifyItemSecond_PDP_Type	PDP-Type	Unsigned 8-bit integer
ranap.RAB_SetupOrModifyItemSecond_transportLayerAddressIPv4	transportLayerAddressIPv4	Unsigned 8-bit integer

Field	Field Name	Type
ranap.RAB_SetupOrModifyItem_element	RAB_SetupOrModifyItem_element	Unsigned 16-bit integer
ranap.RAB_SetupOrModifyItem_extension	RAB_SetupOrModifyItem_extension	Unsigned 16-bit integer
ranap.RAC	RAC	Byte array
ranap.SAC	SAC	Byte array
ranap.allocationOrRetentionPriority	allocationOrRetentionPriority	Unsigned 8-bit integer
ranap.bindingID	bindingID	Byte array
ranap.cause_choice	cause choice	Unsigned 8-bit integer
ranap.cause_value	cause value	Unsigned 8-bit integer
ranap.dataVolumeReference	dataVolumeReference	Unsigned 8-bit integer
ranap.dataVolumeReference_present	dataVolumeReference_present	Unsigned 8-bit integer
ranap.dataVolumeReportingIndication	dataVolumeReportingIndication	Unsigned 8-bit integer
ranap.dl-UnsuccessfullyTransmittedDataVolume	dl-UnsuccessfullyTransmittedDataVolume	Unsigned 32-bit integer
ranap.dl_GTP_PDU_SequenceNumber	dl_GTP_PDU_SequenceNumber	Unsigned 8-bit integer
ranap.dl_N_PDU_SequenceNumber	dl_N_PDU_SequenceNumber	Unsigned 8-bit integer
ranap.dl_UnsuccessfullyTransmittedDataVolume	dl_UnsuccessfullyTransmittedDataVolume_present	Unsigned 8-bit integer
ranap.dl_dataVolumes_present	dl_dataVolumes	Unsigned 8-bit integer
ranap.gTP_TEI	gTP_TEI	Byte array
ranap.guaranteedBitRate_present	guaranteedBitRate	Unsigned 8-bit integer
ranap.iECriticality	iECriticality	Unsigned 8-bit integer
ranap.iEsCriticalityDiagnostics_present	iEsCriticalityDiagnostics	Unsigned 8-bit integer
ranap.ie.ProtocolExtensionFields.Id	ProtocolExtensionFieldId	Unsigned 16-bit integer

Field	Field Name	Type
ranap.ie.ProtocolExtensionField.criticality	Criticality of ProtocolExtensionField	Unsigned 8-bit integer
ranap.ie.criticality	Criticality of IE	Unsigned 8-bit integer
ranap.ie.iE-Extensions_present	iE-Extensions	Unsigned 8-bit integer
ranap.ie.ie_id	IE-ID	Unsigned 16-bit integer
ranap.ie.number_of_ProtocolExtensionFields	Number of Protocol Extension Fields	Unsigned 16-bit integer
ranap.ie.number_of_octets	Number of Octets in IE	Unsigned 16-bit integer
ranap.ie.protocol_extension_present	Protocol Extension	Unsigned 8-bit integer
ranap.ie_pair.first_criticality	First Criticality	Unsigned 8-bit integer
ranap.ie_pair.first_value.number_of_octets	Number of Octets in first value	Unsigned 16-bit integer
ranap.ie_pair.second_criticality	Second Criticality	Unsigned 8-bit integer
ranap.ie_pair.second_value.number_of_octets	Number of Octets in second value	Unsigned 16-bit integer
ranap.iuTransportAssociation_present	iuTransportAssociation	Unsigned 8-bit integer
ranap.msg_extension_present	Message Extension	Unsigned 8-bit integer
ranap.nAS-SynchronisationIndicator	nAS-SynchronisationIndicator	Unsigned 8-bit integer
ranap.nAS-SynchronisationIndicator_present	nAS-SynchronisationIndicator	Unsigned 8-bit integer
ranap.nas_pdu_length	length of NAS-PDU	Unsigned 16-bit integer
ranap.num_of_CriticalityDiagnostics_IEs	Number of CriticalityDiagnostics-IEs	Unsigned 16-bit integer
ranap.number_of_ProtocolExtensionFields	Number of ProtocolExtensionFields	Unsigned 16-bit integer
ranap.number_of_RABs	Number of RABs	Unsigned 8-bit integer
ranap.number_of_ies	Number of IEs in list	Unsigned 16-bit integer

Field	Field Name	Type
ranap.pDP_TypeInformation_present	pDP-TypeInformation	Unsigned 8-bit integer
ranap.pdu.criticality	Criticality of PDU	Unsigned 8-bit integer
ranap.pdu.num_of_octets	Number of Octets in PDU	Unsigned 16-bit integer
ranap.pdu.number_of_ies	Number of IEs in PDU	Unsigned 16-bit integer
ranap.procedureCode_present	procedureCode	Unsigned 8-bit integer
ranap.procedureCriticality	procedureCriticality	Unsigned 8-bit integer
ranap.procedureCriticality_present	procedureCriticality	Unsigned 8-bit integer
ranap.procedure_code	Procedure Code	Unsigned 8-bit integer
ranap.rAB_Parameters_present	rAB-Parameters	Unsigned 8-bit integer
ranap.rAB_SubflowCombinationBitRate_present	subflowSDU_Size	Unsigned 8-bit integer
ranap.rab_Parameters.allocationRetentionPriority.pre_emptionCapability	pre-emptionCapability	Unsigned 8-bit integer
ranap.rab_Parameters.allocationRetentionPriority.pre_emptionVulnerability	pre-emptionVulnerability	Unsigned 8-bit integer
ranap.rab_Parameters.allocationRetentionPriority.priorityLevel	priorityLevel	Unsigned 8-bit integer
ranap.rab_Parameters.allocationRetentionPriority.queuingAllowed	queuingAllowed	Unsigned 8-bit integer
ranap.rab_Parameters.deliveryOrder	deliveryOrder	Unsigned 8-bit integer
ranap.rab_Parameters.guaranteedBitrate	guaranteedBitrate	Unsigned 32-bit integer
ranap.rab_Parameters.maxBitrate	maxBitrate	Unsigned 32-bit integer
ranap.rab_Parameters.maxSDU_Size	maxSDU_Size	Unsigned 16-bit integer
ranap.rab_Parameters.rAB_AsymmetryIndicator	rAB-AsymmetryIndicator	Unsigned 8-bit integer
ranap.rab_Parameters.rAB_SubflowCombinationBitRate	rAB-SubflowCombinationBitRate	Unsigned 32-bit integer
ranap.rab_Parameters.ranap_deliveryOfErroneousSDU	deliveryOfErroneousSDU	Unsigned 8-bit integer

Field	Field Name	Type
ranap.rab_Parameters.relocationRequirement	relocationRequirement	Unsigned 8-bit integer
ranap.rab_Parameters.residualBitErrorRatio.exponent	residualBitErrorRatio.exponent	Unsigned 8-bit integer
ranap.rab_Parameters.residualBitErrorRatio.mantissa	residualBitErrorRatio.mantissa	Unsigned 8-bit integer
ranap.rab_Parameters.sDU_ErrorRatio.exponent	sDU_ErrorRatio.exponent	Unsigned 8-bit integer
ranap.rab_Parameters.sDU_ErrorRatio.mantissa	sDU_ErrorRatio.mantissa	Unsigned 8-bit integer
ranap.rab_Parameters.sourceStatisticsDescriptor	sourceStatisticsDescriptor	Unsigned 8-bit integer
ranap.rab_Parameters.subflowSDU_Size	subflowSDU_Size	Unsigned 8-bit integer
ranap.rab_Parameters.trafficClass	Traffic Class	Unsigned 8-bit integer
ranap.rab_Parameters.trafficHandlingPriority	trafficHandlingPriority	Unsigned 8-bit integer
ranap.rab_Parameters.transferDelay	transferDelay	Unsigned 16-bit integer
ranap.ranap_pdu_index	RANAP-PDU Index	Unsigned 8-bit integer
ranap.relocationRequirement	relocationRequirement	Unsigned 8-bit integer
ranap.repetitionNumber	repetitionNumber	Unsigned 16-bit integer
ranap.repetitionNumber_present	repetitionNumber	Unsigned 8-bit integer
ranap.sDU_ErrorRatio_present	sDU_ErrorRatio	Unsigned 8-bit integer
ranap.sDU_FormatInformationParameters	sDU_FormatInformationParameters	Unsigned 8-bit integer
ranap.service_Handover	service-Handover	Unsigned 8-bit integer
ranap.service_Handover_present	service-Handover	Unsigned 8-bit integer
ranap.sourceStatisticsDescriptor_present	sourceStatisticsDescriptor	Unsigned 8-bit integer
ranap.subflowSDU_Size_present	subflowSDU_Size	Unsigned 8-bit integer
ranap.trafficHandlingPriority_present	trafficHandlingPriority	Unsigned 8-bit integer

Field	Field Name	Type
ranap.transferDelay_present	transferDelay	Unsigned 8-bit integer
ranap.transportLayerAddress	transportLayerAddress	Byte array
ranap.transportLayerAddress_length	length of transportLayerAddress	Unsigned 8-bit integer
ranap.transportLayerAddress_present	transportLayerAddress	Unsigned 8-bit integer
ranap.transportLayerInformation_present	transportLayerInformation	Unsigned 8-bit integer
ranap.triggeringMessage	triggeringMessage	Unsigned 8-bit integer
ranap.triggeringMessage_present	triggeringMessage	Unsigned 8-bit integer
ranap.uP_ModeVersions	uP_ModeVersions	Byte array
ranap.ul_GTP_PDU_SequenceNumber_present	ul_GTP_PDU_SequenceNumber	Unsigned 8-bit integer
ranap.ul_N_PDU_SequenceNumber_present	ul_N_PDU_SequenceNumber	Unsigned 8-bit integer
ranap.userPlaneInformation_present	userPlaneInformation	Unsigned 8-bit integer
ranap.userPlaneMode	userPlaneMode	Unsigned 8-bit integer

Table A-158. Radio Access Network Application Part (ranap)

A.159. Radius Protocol (radius)

Field	Field Name	Type
radius.code	Code	Unsigned 8-bit integer
radius.id	Identifier	Unsigned 8-bit integer
radius.length	Length	Unsigned 16-bit integer

Table A-159. Radius Protocol (radius)

A.160. Real Time Streaming Protocol (rtsp)

Field	Field Name	Type
rtsp.method	Method	String
rtsp.status	Status	Unsigned 32-bit integer
rtsp.url	URL	String

Table A-160. Real Time Streaming Protocol (rtsp)

A.161. Real-Time Transport Protocol (rtp)

Field	Field Name	Type
rtp.cc	Contributing source identifiers count	Unsigned 8-bit integer
rtp.csrc.item	CSRC item	Unsigned 32-bit integer
rtp.ext	Extension	Boolean
rtp.ext.len	Extension length	Unsigned 16-bit integer
rtp.ext.profile	Defined by profile	Unsigned 16-bit integer
rtp.hdr_ext	Header extension	Unsigned 32-bit integer
rtp.marker	Marker	Boolean
rtp.p_type	Payload type	Unsigned 8-bit integer
rtp.padding	Padding	Boolean
rtp.padding.count	Padding count	Unsigned 8-bit integer
rtp.padding.data	Padding data	Byte array
rtp.payload	Payload	Byte array
rtp.seq	Sequence number	Unsigned 16-bit integer
rtp.ssrc	Synchronization Source identifier	Unsigned 32-bit integer
rtp.timestamp	Timestamp	Unsigned 32-bit integer
rtp.version	Version	Unsigned 8-bit integer

Table A-161. Real-Time Transport Protocol (rtp)

A.162. Real-time Transport Control Protocol (rtcp)

Field	Field Name	Type
rtcp.app.data	Application specific data	Byte array
rtcp.app.name	Name (ASCII)	String
rtcp.app.subtype	Subtype	Unsigned 8-bit integer
rtcp.length	Length	Unsigned 16-bit integer
rtcp.nack.blp	Bitmask of following lost packets	Unsigned 16-bit integer
rtcp.nack.fsn	First sequence number	Unsigned 16-bit integer
rtcp.padding	Padding	Boolean
rtcp.padding.count	Padding count	Unsigned 8-bit integer
rtcp.padding.data	Padding data	Byte array
rtcp.pt	Packet type	Unsigned 8-bit integer
rtcp.rc	Reception report count	Unsigned 8-bit integer
rtcp.sc	Source count	Unsigned 8-bit integer
rtcp.sdes.length	Length	Unsigned 32-bit integer
rtcp.sdes.prefix.length	Prefix length	Unsigned 8-bit integer
rtcp.sdes.prefix.string	Prefix string	String
rtcp.sdes.ssrc_csrc	SSRC / CSRC identifier	Unsigned 32-bit integer
rtcp.sdes.text	Text	String
rtcp.sdes.type	Type	Unsigned 8-bit integer
rtcp.sender.octetcount	Sender’s octet count	Unsigned 32-bit integer
rtcp.sender.packetcount	Sender’s packet count	Unsigned 32-bit integer
rtcp.senderssrc	Sender SSRC	Unsigned 32-bit integer
rtcp.ssrc.cum_nr	Cumulative number of packets lost	Unsigned 32-bit integer
rtcp.ssrc.dlsr	Delay since last SR timestamp	Unsigned 32-bit integer
rtcp.ssrc.ext_high	Extended highest sequence number received	Unsigned 32-bit integer
rtcp.ssrc.fraction	Fraction lost	Unsigned 8-bit integer
rtcp.ssrc.high_cycles	Sequence number cycles count	Unsigned 16-bit integer
rtcp.ssrc.high_seq	Highest sequence number received	Unsigned 16-bit integer

Field	Field Name	Type
rtcp.ssrc.identifier	Identifier	Unsigned 32-bit integer
rtcp.ssrc.jitter	Interarrival jitter	Unsigned 32-bit integer
rtcp.ssrc.lsr	Last SR timestamp	Unsigned 32-bit integer
rtcp.timestamp.ntp	NTP timestamp	String
rtcp.timestamp.rtp	RTP timestamp	Unsigned 32-bit integer
rtcp.version	Version	Unsigned 8-bit integer

Table A-162. Real-time Transport Control Protocol (rtcp)

A.163. Remote Procedure Call (rpc)

Field	Field Name	Type
rpc.array.len	num	Unsigned 32-bit integer
rpc.auth.flavor	Flavor	Unsigned 32-bit integer
rpc.auth.gid	GID	Unsigned 32-bit integer
rpc.auth.length	Length	Unsigned 32-bit integer
rpc.auth.machinename	Machine Name	String
rpc.auth.stamp	Stamp	Unsigned 32-bit integer
rpc.auth.uid	UID	Unsigned 32-bit integer
rpc.authdes.convkey	Conversation Key (encrypted)	Unsigned 32-bit integer
rpc.authdes.namekind	Namekind	Unsigned 32-bit integer
rpc.authdes.netname	Netname	String
rpc.authdes.nickname	Nickname	Unsigned 32-bit integer
rpc.authdes.timestamp	Timestamp (encrypted)	Unsigned 32-bit integer
rpc.authdes.timeverf	Timestamp verifier (encrypted)	Unsigned 32-bit integer
rpc.authdes.window	Window (encrypted)	Unsigned 32-bit integer
rpc.authdes.windowverf	Window verifier (encrypted)	Unsigned 32-bit integer
rpc.authgss.checksum	GSS Checksum	Byte array
rpc.authgss.context	GSS Context	Byte array
rpc.authgss.data	GSS Data	Byte array
rpc.authgss.data.length	Length	Unsigned 32-bit integer
rpc.authgss.major	GSS Major Status	Unsigned 32-bit integer

Field	Field Name	Type
rpc.authgss.minor	GSS Minor Status	Unsigned 32-bit integer
rpc.authgss.procedure	GSS Procedure	Unsigned 32-bit integer
rpc.authgss.seqnum	GSS Sequence Number	Unsigned 32-bit integer
rpc.authgss.service	GSS Service	Unsigned 32-bit integer
rpc.authgss.token	GSS Token	Byte array
rpc.authgss.version	GSS Version	Unsigned 32-bit integer
rpc.authgss.window	GSS Sequence Window	Unsigned 32-bit integer
rpc.call.dup	Duplicate Call	Unsigned 32-bit integer
rpc.dup	Duplicate Transaction	Unsigned 32-bit integer
rpc.fraglen	Fragment Length	Unsigned 32-bit integer
rpc.lastfrag	Last Fragment	Boolean
rpc.msgtyp	Message Type	Unsigned 32-bit integer
rpc.procedure	Procedure	Unsigned 32-bit integer
rpc.program	Program	Unsigned 32-bit integer
rpc.programversion	Program Version	Unsigned 32-bit integer
rpc.programversion.max	Program Version (Maximum)	Unsigned 32-bit integer
rpc.programversion.min	Program Version (Minimum)	Unsigned 32-bit integer
rpc.reply.dup	Duplicate Reply	Unsigned 32-bit integer
rpc.replystat	Reply State	Unsigned 32-bit integer
rpc.state_accept	Accept State	Unsigned 32-bit integer
rpc.state_auth	Auth State	Unsigned 32-bit integer
rpc.state_reject	Reject State	Unsigned 32-bit integer
rpc.value_follows	Value Follows	Boolean
rpc.version	RPC Version	Unsigned 32-bit integer
rpc.version.max	RPC Version (Maximum)	Unsigned 32-bit integer
rpc.version.min	RPC Version (Minimum)	Unsigned 32-bit integer
rpc.xid	XID	Unsigned 32-bit integer

Table A-163. Remote Procedure Call (rpc)

A.164. Remote Quota (rquota)

Field	Field Name	Type

Field	Field Name	Type
rquota.active	active	Boolean
rquota.bhardlimit	bhardlimit	Unsigned 32-bit integer
rquota.bsize	bsize	Unsigned 32-bit integer
rquota.bsoftlimit	bsoftlimit	Unsigned 32-bit integer
rquota.btimeleft	btimeleft	Unsigned 32-bit integer
rquota.curblocks	curblocks	Unsigned 32-bit integer
rquota.curfiles	curfiles	Unsigned 32-bit integer
rquota.fhardlimit	fhardlimit	Unsigned 32-bit integer
rquota.fsoftlimit	fsoftlimit	Unsigned 32-bit integer
rquota.ftimeleft	ftimeleft	Unsigned 32-bit integer
rquota.pathp	pathp	String
rquota.rquota	rquota	No value
rquota.status	status	Unsigned 32-bit integer
rquota.uid	uid	Unsigned 32-bit integer

Table A-164. Remote Quota (rquota)

A.165. Remote Shell (rsh)

Field	Field Name	Type
rsh.request	Request	Boolean
rsh.response	Response	Boolean

Table A-165. Remote Shell (rsh)

A.166. Remote Wall protocol (rwall)

Field	Field Name	Type
rwall.message	Message	String

Table A-166. Remote Wall protocol (rwall)

A.167. Resource ReserVation Protocol (RSVP)

(rsvp)

Field	Field Name	Type
rsvp.ack	MESSAGE-ID ACK	No value
rsvp.adspec	ADSPEC	No value
rsvp.confirm	CONFIRM	No value
rsvp.error	ERROR	No value
rsvp.explicit_route	EXPLICIT ROUTE	No value
rsvp.filter	FILTERSPEC	No value
rsvp.flowspec	FLOWSPEC	No value
rsvp.hello	HELLO Message	Boolean
rsvp.hello_obj	HELLO Request/Ack	No value
rsvp.hop	HOP	No value
rsvp.integrity	INTEGRITY	No value
rsvp.label	LABEL	No value
rsvp.label_request	LABEL REQUEST	No value
rsvp.msg	Message Type	Unsigned 8-bit integer
rsvp.msgid	MESSAGE-ID	No value
rsvp.msgid_list	MESSAGE-ID LIST	No value
rsvp.obj_unknown	Unknown object	No value
rsvp.object	Object class	Unsigned 8-bit integer
rsvp.path	Path Message	Boolean
rsvp.perr	Path Error Message	Boolean
rsvp.policy	POLICY	No value
rsvp.ptear	Path Tear Message	Boolean
rsvp.record_route	RECORD ROUTE	No value
rsvp.rerr	Resv Error Message	Boolean
rsvp.resv	Resv Message	Boolean
rsvp.resvconf	Resv Confirm Message	Boolean
rsvp.rtear	Resv Tear Message	Boolean
rsvp.rtearconf	Resv Tear Confirm Message	Boolean
rsvp.scope	SCOPE	No value
rsvp.sender	SENDER TEMPLATE	No value
rsvp.sender.ip	Sender IPv4 address	IPv4 address
rsvp.sender.lsp_id	Sender LSP ID	Unsigned 16-bit integer
rsvp.sender.port	Sender port number	Unsigned 16-bit integer

Field	Field Name	Type
rsvp.session	SESSION	No value
rsvp.session.ext_tunnel_id	Extended tunnel ID	Unsigned 32-bit integer
rsvp.session.ip	Destination address	IPv4 address
rsvp.session.port	Port number	Unsigned 16-bit integer
rsvp.session.proto	Protocol	Unsigned 8-bit integer
rsvp.session.tunnel_id	Tunnel ID	Unsigned 16-bit integer
rsvp.session_attribute	SESSION ATTRIBUTE	No value
rsvp.style	STYLE	No value
rsvp.time	TIME VALUES	No value
rsvp.tspec	SENDER TSPEC	No value

Table A-167. Resource ReserVation Protocol (RSVP) (rsvp)

A.168. Rlogin Protocol (rlogin)

Field	Field Name	Type
rlogin.user_info	User Info	No value
rlogin.window_size	Window Info	No value
rlogin.window_size.cols	Columns	Unsigned 16-bit integer
rlogin.window_size.rows	Rows	Unsigned 16-bit integer
rlogin.window_size.x_pixels	X Pixels	Unsigned 16-bit integer
rlogin.window_size.y_pixels	Y Pixels	Unsigned 16-bit integer

Table A-168. Rlogin Protocol (rlogin)

A.169. Routing Information Protocol (rip)

Field	Field Name	Type

Table A-169. Routing Information Protocol (rip)

A.170. Routing Table Maintenance Protocol (rtmp)

Field	Field Name	Type
nbp.nodeid	Node	Unsigned 8-bit integer
nbp.nodeid.length	Node Length	Unsigned 8-bit integer
rtmp.function	Function	Unsigned 8-bit integer
rtmp.net	Net	Unsigned 16-bit integer
rtmp.tuple.dist	Distance	Unsigned 16-bit integer
rtmp.tuple.net	Net	Unsigned 16-bit integer
rtmp.tuple.range_end	Range End	Unsigned 16-bit integer
rtmp.tuple.range_start	Range Start	Unsigned 16-bit integer

Table A-170. Routing Table Maintenance Protocol (rtmp)

A.171. SCCP user adaptation layer light (sual)

Field	Field Name	Type
sual.error_code	Error Code	Unsigned 16-bit integer
sual.message_length	Message length	Unsigned 32-bit integer
sual.message_type	Message Type	Unsigned 16-bit integer
sual.spare_1	Spare	Unsigned 8-bit integer
sual.spare_2	Spare	Unsigned 16-bit integer
sual.subsystem_number	Subsystem number	Unsigned 16-bit integer
sual.version	Version	Unsigned 8-bit integer

Table A-171. SCCP user adaptation layer light (sual)

A.172. SMB (Server Message Block Protocol) (smb)

Field	Field Name	Type

Field	Field Name	Type
smb.cmd	SMB Command	Unsigned 8-bit integer

Table A-172. SMB (Server Message Block Protocol) (smb)

A.173. SMB MailSlot Protocol (mailslot)

Field	Field Name	Type

Table A-173. SMB MailSlot Protocol (mailslot)

A.174. SNMP Multiplex Protocol (smux)

Field	Field Name	Type

Table A-174. SNMP Multiplex Protocol (smux)

A.175. SPRAY (spray)

Field	Field Name	Type
spray.clock	clock	No value
spray.counter	counter	Unsigned 32-bit integer
spray.sec	sec	Unsigned 32-bit integer
spray.sprayarr	Data	Byte array
spray.usec	usec	Unsigned 32-bit integer

Table A-175. SPRAY (spray)

A.176. SSCOP (sscop)

Field	Field Name	Type

Table A-176. SSCOP (sscop)

A.177. Secure Socket Layer (ssl)

Field	Field Name	Type
ss.handshake.dname	Distinguished Name	String
ssl.alert_message	Alert Message	No value
ssl.alert_message.desc	Description	Unsigned 8-bit integer
ssl.alert_message.level	Level	Unsigned 8-bit integer
ssl.app_data	Application Data	No value
ssl.change_cipher_spec	Change Cipher Spec Message	No value
ssl.handshake	Handshake Protocol	No value
ssl.handshake.cert_type	Certificate type	Unsigned 8-bit integer
ssl.handshake.cert_types	Certificate types	String
ssl.handshake.certificate	Certificate	String
ssl.handshake.certificate_length	Certificate Length	Unsigned 24-bit integer
ssl.handshake.challenge	Challenge	No value
ssl.handshake.challenge_length	Challenge Length	Unsigned 16-bit integer
ssl.handshake.cipher_spec_length	Cipher Spec Length	Unsigned 16-bit integer
ssl.handshake.cipherspec	Cipher Spec	Unsigned 24-bit integer
ssl.handshake.cipherspecs	Cipher Suites	String
ssl.handshake.ciphersuite	Cipher Suite	Unsigned 16-bit integer
ssl.handshake.clear_key_data	Clear Key Data	No value
ssl.handshake.clear_key_length	Clear Key Data Length	Unsigned 16-bit integer

Field	Field Name	Type
ssl.handshake.comp_method	Compression Method	Unsigned 8-bit integer
ssl.handshake.comp_methods	Compression Methods	String
ssl.handshake.connection_id	Connection ID	No value
ssl.handshake.connection_id_length	Connection ID Length	Unsigned 16-bit integer
ssl.handshake.dname_len	Distinguished Name Length	Unsigned 16-bit integer
ssl.handshake.dnames	Distinguished Names	String
ssl.handshake.encrypted_key	Encrypted Key	No value
ssl.handshake.encrypted_key_length	Encrypted Key Data Length	Unsigned 16-bit integer
ssl.handshake.key_arg	Key Argument	No value
ssl.handshake.key_arg_length	Key Argument Length	Unsigned 16-bit integer
ssl.handshake.length	Length	Unsigned 24-bit integer
ssl.handshake.md5_hash	MD5 Hash	No value
ssl.handshake.random	Random.bytes	No value
ssl.handshake.random_time	Random.gmt_unix_time	Date/Time stamp
ssl.handshake.session_id	Session ID	String
ssl.handshake.session_id_hit	Session ID Hit	Boolean
ssl.handshake.session_id_length	Session ID Length	Unsigned 8-bit integer
ssl.handshake.sha_hash	SHA-1 Hash	No value
ssl.handshake.type	Handshake Message Type	Unsigned 8-bit integer
ssl.handshake.verify_data	Verify Data	No value
ssl.handshake.version	Version	Unsigned 16-bit integer
ssl.record	Record Layer	No value
ssl.record.content_type	Content Type	Unsigned 8-bit integer
ssl.record.is_escape	Is Escape	Boolean
ssl.record.length	Length	Unsigned 16-bit integer
ssl.record.padding_length	Padding Length	Unsigned 8-bit integer
ssl.record.version	Version	Unsigned 16-bit integer

Table A-177. Secure Socket Layer (ssl)

A.178. Sequenced Packet eXchange (spx)

Field	Field Name	Type
spx.ack	Acknowledgment Number	Unsigned 16-bit integer
spx.alloc	Allocation Number	Unsigned 16-bit integer
spx.ctl	Connection Control	Unsigned 8-bit integer
spx.dst	Destination Connection ID	Unsigned 16-bit integer
spx.seq	Sequence Number	Unsigned 16-bit integer
spx.src	Source Connection ID	Unsigned 16-bit integer
spx.type	Datastream type	Unsigned 8-bit integer

Table A-178. Sequenced Packet eXchange (spx)

A.179. Service Advertisement Protocol (ipxsap)

Field	Field Name	Type
ipxsap.request	Request	Boolean
ipxsap.response	Response	Boolean

Table A-179. Service Advertisement Protocol (ipxsap)

A.180. Service Location Protocol (srvloc)

Field	Field Name	Type
srvloc.err	Error Code	Unsigned 16-bit integer
srvloc.flags	Flags	Unsigned 8-bit integer
srvloc.function	Function	Unsigned 8-bit integer
srvloc.version	Version	Unsigned 8-bit integer

Table A-180. Service Location Protocol (srvloc)

A.181. Session Announcement Protocol (sap)

Field	Field Name	Type
sap.auth	Authentication data	No value
sap.auth.flags	Authentication data flags	Unsigned 8-bit integer
sap.auth.flags.p	Padding Bit	Boolean
sap.auth.flags.t	Authentication Type	Unsigned 8-bit integer
sap.auth.flags.v	Version Number	Unsigned 8-bit integer
sap.flags	Flags	Unsigned 8-bit integer
sap.flags.a	Address Type	Boolean
sap.flags.c	Compression Bit	Boolean
sap.flags.e	Encryption Bit	Boolean
sap.flags.r	Reserved	Boolean
sap.flags.t	Message Type	Boolean
sap.flags.v	Version Number	Unsigned 8-bit integer

Table A-181. Session Announcement Protocol (sap)

A.182. Session Description Protocol (sdp)

Field	Field Name	Type

Table A-182. Session Description Protocol (sdp)

A.183. Session Initiation Protocol (sip)

Field	Field Name	Type
sip.msg_hdr	Message Header	No value

Table A-183. Session Initiation Protocol (sip)

A.184. Short Frame (short)

Field	Field Name	Type

Table A-184. Short Frame (short)

A.185. Simple Mail Transfer Protocol (smtp)

Field	Field Name	Type
smtp.req	Request	Boolean
smtp.rsp	Response	Boolean

Table A-185. Simple Mail Transfer Protocol (smtp)

A.186. Simple Network Management Protocol (snmp)

Field	Field Name	Type
snmpv3.flags	SNMPv3 Flags	Unsigned 8-bit integer
snmpv3.flags.auth	Authenticated	Boolean
snmpv3.flags.crypt	Encrypted	Boolean
snmpv3.flags.report	Reportable	Boolean

Table A-186. Simple Network Management Protocol (snmp)

A.187. Sinec H1 Protocol (h1)

Field	Field Name	Type
h1.dbnr	Memory block number	Unsigned 8-bit integer
h1.dlen	Length in words	Signed 16-bit integer

Field	Field Name	Type
h1.dwnr	Address within memory block	Unsigned 16-bit integer
h1.empty	Empty field	Unsigned 8-bit integer
h1.empty_len	Empty field length	Unsigned 8-bit integer
h1.header	H1-Header	Unsigned 16-bit integer
h1.len	Length indicator	Unsigned 16-bit integer
h1.opcode	Opcode	Unsigned 8-bit integer
h1.opfield	Operation identifier	Unsigned 8-bit integer
h1.oplen	Operation length	Unsigned 8-bit integer
h1.org	Memory type	Unsigned 8-bit integer
h1.reqlen	Request length	Unsigned 8-bit integer
h1.request	Request identifier	Unsigned 8-bit integer
h1.reslen	Response length	Unsigned 8-bit integer
h1.response	Response identifier	Unsigned 8-bit integer
h1.resvalue	Response value	Unsigned 8-bit integer

Table A-187. Sinec H1 Protocol (h1)

A.188. Socks Protocol (socks)

Field	Field Name	Type
socks.command	Command	Unsigned 16-bit integer
socks.dst	Remote Address	IPv4 address
socks.dstV6	Remote Address	IPv6 address
socks.dstport	Remote Port	Unsigned 16-bit integer
socks.username	User Name	String
socks.ver	Version	Unsigned 8-bit integer

Table A-188. Socks Protocol (socks)

A.189. Spanning Tree Protocol (stp)

Field	Field Name	Type

Field	Field Name	Type
stp.bridge.hw	Bridge Identifier	6-byte Hardware (MAC) Address
stp.flags	BPDU flags	Unsigned 8-bit integer
stp.forward	Forward Delay	Double-precision floating point
stp.hello	Hello Time	Double-precision floating point
stp.max_age	Max Age	Double-precision floating point
stp.msg_age	Message Age	Double-precision floating point
stp.port	Port identifier	Unsigned 16-bit integer
stp.protocol	Protocol Identifier	Unsigned 16-bit integer
stp.root.cost	Root Path Cost	Unsigned 32-bit integer
stp.root.hw	Root Identifier	6-byte Hardware (MAC) Address
stp.type	BPDU type	Unsigned 8-bit integer
stp.version	Protocol Version Identifier	Unsigned 8-bit integer

Table A-189. Spanning Tree Protocol (stp)

A.190. Stream Control Transmission Protocol (sctp)

Field	Field Name	Type
sctp.cause.code	Cause code	Unsigned 16-bit integer
sctp.cause.length	Cause length	Unsigned 16-bit integer
sctp.cause.measure_of_staleness	Measure of staleness in usec	Unsigned 32-bit integer
sctp.cause.missing_parameter_type	Missing parameters type	Unsigned 16-bit integer
sctp.cause.nr_of_missing_parameters	Number of missing parameters	Unsigned 32-bit integer
sctp.cause.stream_identifier	Stream identifier	Unsigned 16-bit integer

Field	Field Name	Type
sctp.cause.tsn	TSN	Unsigned 32-bit integer
sctp.checksum	Adler-32 checksum	Unsigned 32-bit integer
sctp.chunk_flags	Flags	Unsigned 8-bit integer
sctp.chunk_length	Length	Unsigned 16-bit integer
sctp.chunk_type	Identifier	Unsigned 8-bit integer
sctp.cumulative.tsn.ack	Cumulative TSN Ack	Unsigned 32-bit integer
sctp.cwr.lowest_tsn	Lowest TSN	Unsigned 32-bit integer
sctp.data.b_bit	B-Bit	Boolean
sctp.data.e_bit	E-Bit	Boolean
sctp.data.u.bit	U-Bit	Boolean
sctp.dstport	Destination port	Unsigned 16-bit integer
sctp.ecne.lowest_tsn	Lowest TSN	Unsigned 32-bit integer
sctp.init.chunk.credit	Advertised reciever window credit (a_rwnd)	Unsigned 32-bit integer
sctp.init.chunk.initial.tsn	Initial TSN	Unsigned 32-bit integer
sctp.init.chunk.initiate.tag	Initiate tag	Unsigned 32-bit integer
sctp.init.chunk.nr.in.streams	Number of inbound streams	Unsigned 16-bit integer
sctp.init.chunk.nr.out.streams	Number of outbound streams	Unsigned 16-bit integer
sctp.parameter.cookie_preservative.suggested_inc	Suggested Cookie life-span increment (msec)	Unsigned 32-bit integer
sctp.parameter.hostname.hostname	Hostname	String
sctp.parameter.ipv4_address	IP Version 4 address	IPv4 address
sctp.parameter.ipv6_address	IP Version 6 address	IPv6 address
sctp.parameter.length	Parameter length	Unsigned 16-bit integer
sctp.parameter.supported_address_types.supp_type	Supported address type	Unsigned 16-bit integer
sctp.parameter.type	Parameter type	Unsigned 16-bit integer
sctp.payload_proto_id	Payload Protocol identifier	Unsigned 32-bit integer
sctp.sack.a_rwnd	Advertised receiver window credit (a_rwnd)	Unsigned 32-bit integer

Field	Field Name	Type
sctp.sack.cumulative_tsn_ack	Cumulative TSN ACK	Unsigned 32-bit integer
sctp.sack.duplicate.tsn	Duplicate TSN	Unsigned 16-bit integer
sctp.sack.gap_block_end	End	Unsigned 16-bit integer
sctp.sack.gap_block_start	Start	Unsigned 16-bit integer
sctp.sack.number_of_duplicated_tsns	Number of duplicated TSNs	Unsigned 16-bit integer
sctp.sack.number_of_gap_blocks	Number of gap acknowldgement blocks	Unsigned 16-bit integer
sctp.shutdown.cumulative_tsn_ack	Cumulative TSN Ack	Unsigned 32-bit integer
sctp.shutdown_complete.t_bit	T-Bit	Boolean
sctp.srcport	Source port	Unsigned 16-bit integer
sctp.stream_id	Stream Identifier	Unsigned 16-bit integer
sctp.stream_seq_number	Stream sequence number	Unsigned 16-bit integer
sctp.tsn	TSN	Unsigned 32-bit integer
sctp.verfication_tag	Verification tag	Unsigned 32-bit integer

Table A-190. Stream Control Transmission Protocol (sctp)

A.191. Syslog message (syslog)

Field	Field Name	Type
syslog.facility	Facility	Unsigned 8-bit integer
syslog.level	Level	Unsigned 8-bit integer
syslog.msg_len	Message length	Unsigned 32-bit integer

Table A-191. Syslog message (syslog)

A.192. Systems Network Architecture (sna)

Field	Field Name	Type
sna.rh	Request/Response Header	No value

Field	Field Name	Type
sna.rh.0	Request/Response Header Byte 0	Unsigned 8-bit integer
sna.rh.1	Request/Response Header Byte 1	Unsigned 8-bit integer
sna.rh.2	Request/Response Header Byte 2	Unsigned 8-bit integer
sna.rh.bbi	Begin Bracket Indicator	Boolean
sna.rh.bci	Begin Chain Indicator	Boolean
sna.rh.cdi	Change Direction Indicator	Boolean
sna.rh.cebi	Conditional End Bracket Indicator	Boolean
sna.rh.csi	Code Selection Indicator	Unsigned 8-bit integer
sna.rh.dr1	Definite Response 1 Indicator	Boolean
sna.rh.dr2	Definite Response 2 Indicator	Boolean
sna.rh.ebi	End Bracket Indicator	Boolean
sna.rh.eci	End Chain Indicator	Boolean
sna.rh.edi	Enciphered Data Indicator	Boolean
sna.rh.eri	Exception Response Indicator	Boolean
sna.rh.fi	Format Indicator	Boolean
sna.rh.lcci	Length-Checked Compression Indicator	Boolean
sna.rh.pdi	Padded Data Indicator	Boolean
sna.rh.pi	Pacing Indicator	Boolean
sna.rh.qri	Queued Response Indicator	Boolean
sna.rh.rlwi	Request Larger Window Indicator	Boolean
sna.rh.rri	Request/Response Indicator	Unsigned 8-bit integer
sna.rh.rti	Response Type Indicator	Boolean
sna.rh.ru_category	Request/Response Unit Category	Unsigned 8-bit integer
sna.rh.sdi	Sense Data Included	Boolean

Field	Field Name	Type
sna.th	Transmission Header	No value
sna.th.0	Transmission Header Byte 0	Unsigned 8-bit integer
sna.th.cmd_fmt	Command Format	Unsigned 8-bit integer
sna.th.cmd_sn	Command Sequence Number	Unsigned 16-bit integer
sna.th.cmd_type	Command Type	Unsigned 8-bit integer
sna.th.daf	Destination Address Field	Unsigned 16-bit integer
sna.th.dcf	Data Count Field	Unsigned 16-bit integer
sna.th.def	Destination Element Field	Unsigned 16-bit integer
sna.th.dsaf	Destination Subarea Address Field	Unsigned 32-bit integer
sna.th.efi	Expedited Flow Indicator	Unsigned 8-bit integer
sna.th.er_vr_supp_ind	ER and VR Support Indicator	Unsigned 8-bit integer
sna.th.ern	Explicit Route Number	Unsigned 8-bit integer
sna.th.fid	Format Identifer	Unsigned 8-bit integer
sna.th.iern	Initial Explicit Route Number	Unsigned 8-bit integer
sna.th.lsid	Local Session Identification	Unsigned 8-bit integer
sna.th.mft	MPR FID4 Type	Boolean
sna.th.mpf	Mapping Field	Unsigned 8-bit integer
sna.th.nlp_cp	NLP Count or Padding	Unsigned 8-bit integer
sna.th.nlpoi	NLP Offset Indicator	Unsigned 8-bit integer
sna.th.ntwk_prty	Network Priority	Unsigned 8-bit integer
sna.th.oaf	Origin Address Field	Unsigned 16-bit integer
sna.th.odai	ODAI Assignment Indicator	Unsigned 8-bit integer
sna.th.oef	Origin Element Field	Unsigned 16-bit integer
sna.th.osaf	Origin Subarea Address Field	Unsigned 32-bit integer
sna.th.piubf	PIU Blocking Field	Unsigned 8-bit integer
sna.th.sa	Session Address	Byte array
sna.th.snai	SNA Indicator	Boolean
sna.th.snf	Sequence Number Field	Unsigned 16-bit integer
sna.th.tg_nonfifo_ind	Transmission Group Non-FIFO Indicator	Boolean

Field	Field Name	Type
sna.th.tg_snf	Transmission Group Sequence Number Field	Unsigned 16-bit integer
sna.th.tg_sweep	Transmission Group Sweep	Unsigned 8-bit integer
sna.th.tgsf	Transmission Group Segmenting Field	Unsigned 8-bit integer
sna.th.tpf	Transmission Priority Field	Unsigned 8-bit integer
sna.th.vr_cwi	Virtual Route Change Window Indicator	Unsigned 16-bit integer
sna.th.vr_cwri	Virtual Route Change Window Reply Indicator	Unsigned 16-bit integer
sna.th.vr_pac_cnt_ind	Virtual Route Pacing Count Indicator	Unsigned 8-bit integer
sna.th.vr_rwi	Virtual Route Reset Window Indicator	Boolean
sna.th.vr_snf_send	Virtual Route Send Sequence Number Field	Unsigned 16-bit integer
sna.th.vr_sqti	Virtual Route Sequence and Type Indicator	Unsigned 16-bit integer
sna.th.vrn	Virtual Route Number	Unsigned 8-bit integer
sna.th.vrprq	Virtual Route Pacing Request	Boolean
sna.th.vrprs	Virtual Route Pacing Response	Boolean

Table A-192. Systems Network Architecture (sna)

A.193. TACACS (tacacs)

Field	Field Name	Type
tacacs.destaddr	Destination address	IPv4 address
tacacs.destport	Destination port	Unsigned 16-bit integer
tacacs.line	Line	Unsigned 16-bit integer
tacacs.nonce	Nonce	Unsigned 16-bit integer
tacacs.passlen	Password length	Unsigned 8-bit integer

Field	Field Name	Type
tacacs.reason	Reason	Unsigned 8-bit integer
tacacs.response	Response	Unsigned 8-bit integer
tacacs.result1	Result 1	Unsigned 32-bit integer
tacacs.result2	Result 2	Unsigned 32-bit integer
tacacs.result3	Result 3	Unsigned 16-bit integer
tacacs.type	Type	Unsigned 8-bit integer
tacacs.userlen	Username length	Unsigned 8-bit integer
tacacs.version	Version	Unsigned 8-bit integer

Table A-193. TACACS (tacacs)

A.194. TACACS+ (tacplus)

Field	Field Name	Type
tacplus.flags	Flags	Unsigned 8-bit integer
tacplus.flags.connection_type	Connection type	Boolean
tacplus.flags.payload_type	Payload type	Boolean
tacplus.majvers	Major version	Unsigned 8-bit integer
tacplus.minvers	Minor version	Unsigned 8-bit integer
tacplus.packet_len	Packet length	Unsigned 32-bit integer
tacplus.request	Request	Boolean
tacplus.response	Response	Boolean
tacplus.seqno	Sequence number	Unsigned 8-bit integer
tacplus.session_id	Session ID	Unsigned 32-bit integer
tacplus.type	Type	Unsigned 8-bit integer

Table A-194. TACACS+ (tacplus)

A.195. TPKT (tpkt)

Field	Field Name	Type
tpkt.length	Length	Unsigned 16-bit integer

Field	Field Name	Type
tpkt.reserved	Reserved	Unsigned 8-bit integer
tpkt.version	Version	Unsigned 8-bit integer

Table A-195. TPKT (tpkt)

A.196. Telnet (telnet)

Field	Field Name	Type

Table A-196. Telnet (telnet)

A.197. Time Protocol (time)

Field	Field Name	Type
time.time	Time	Unsigned 32-bit integer

Table A-197. Time Protocol (time)

A.198. Token-Ring (tr)

Field	Field Name	Type
tr.ac	Access Control	Unsigned 8-bit integer
tr.addr	Source or Destination Address	6-byte Hardware (MAC) Address
tr.broadcast	Broadcast Type	Unsigned 8-bit integer
tr.direction	Direction	Unsigned 8-bit integer
tr.dst	Destination	6-byte Hardware (MAC) Address
tr.fc	Frame Control	Unsigned 8-bit integer
tr.frame	Frame	Boolean
tr.frame_pcf	Frame PCF	Unsigned 8-bit integer

Field	Field Name	Type
tr.frame_type	Frame Type	Unsigned 8-bit integer
tr.max_frame_size	Maximum Frame Size	Unsigned 8-bit integer
tr.monitor_cnt	Monitor Count	Unsigned 8-bit integer
tr.priority	Priority	Unsigned 8-bit integer
tr.priority_reservation	Priority Reservation	Unsigned 8-bit integer
tr.rif	Ring-Bridge Pairs	String
tr.rif.bridge	RIF Bridge	Unsigned 8-bit integer
tr.rif.ring	RIF Ring	Unsigned 16-bit integer
tr.rif_bytes	RIF Bytes	Unsigned 8-bit integer
tr.sr	Source Routed	Boolean
tr.src	Source	6-byte Hardware (MAC) Address

Table A-198. Token-Ring (tr)

A.199. Token-Ring Media Access Control (trmac)

Field	Field Name	Type
trmac.dstclass	Destination Class	Unsigned 8-bit integer
trmac.errors.abort	Abort Delimiter Transmitted Errors	Unsigned 8-bit integer
trmac.errors.ac	A/C Errors	Unsigned 8-bit integer
trmac.errors.burst	Burst Errors	Unsigned 8-bit integer
trmac.errors.congestion	Receiver Congestion Errors	Unsigned 8-bit integer
trmac.errors.fc	Frame-Copied Errors	Unsigned 8-bit integer
trmac.errors.freq	Frequency Errors	Unsigned 8-bit integer
trmac.errors.internal	Internal Errors	Unsigned 8-bit integer
trmac.errors.iso	Isolating Errors	Unsigned 16-bit integer
trmac.errors.line	Line Errors	Unsigned 8-bit integer
trmac.errors.lost	Lost Frame Errors	Unsigned 8-bit integer
trmac.errors.noniso	Non-Isolating Errors	Unsigned 16-bit integer
trmac.errors.token	Token Errors	Unsigned 8-bit integer
trmac.length	Total Length	Unsigned 8-bit integer
trmac.mvec	Major Vector	Unsigned 8-bit integer

Field	Field Name	Type
trmac.naun	NAUN	6-byte Hardware (MAC) Address
trmac.srcclass	Source Class	Unsigned 8-bit integer
trmac.svec	Sub-Vector	Unsigned 8-bit integer

Table A-199. Token-Ring Media Access Control (trmac)

A.200. Transmission Control Protocol (tcp)

Field	Field Name	Type
tcp.ack	Acknowledgement number	Unsigned 32-bit integer
tcp.checksum	Checksum	Unsigned 16-bit integer
tcp.checksum_bad	Bad Checksum	Boolean
tcp.dstport	Destination Port	Unsigned 16-bit integer
tcp.flags	Flags	Unsigned 8-bit integer
tcp.flags.ack	Acknowledgment	Boolean
tcp.flags.cwr	Congestion Window Reduced (CWR)	Boolean
tcp.flags.ecn	ECN-Echo	Boolean
tcp.flags.fin	Fin	Boolean
tcp.flags.push	Push	Boolean
tcp.flags.reset	Reset	Boolean
tcp.flags.syn	Syn	Boolean
tcp.flags.urg	Urgent	Boolean
tcp.hdr_len	Header Length	Unsigned 8-bit integer
tcp.nxtseq	Next sequence number	Unsigned 32-bit integer
tcp.port	Source or Destination Port	Unsigned 16-bit integer
tcp.seq	Sequence number	Unsigned 32-bit integer
tcp.srcport	Source Port	Unsigned 16-bit integer
tcp.urgent_pointer	Urgent pointer	Unsigned 16-bit integer
tcp.window_size	Window size	Unsigned 16-bit integer

Table A-200. Transmission Control Protocol (tcp)

A.201. Transparent Network Substrate Protocol (tns)

Field	Field Name	Type
tns.compat_version	Version (Compatible)	Unsigned 16-bit integer
tns.connect	Connect	Boolean
tns.data_flag	Data Flag	Unsigned 16-bit integer
tns.header_checksum	Header Checksum	Unsigned 16-bit integer
tns.length	Packet Length	Unsigned 32-bit integer
tns.packet_checksum	Packet Checksum	Unsigned 16-bit integer
tns.request	Request	Boolean
tns.reserved_byte	Reserved Byte	Byte array
tns.response	Response	Boolean
tns.service_options	Service Options	Unsigned 16-bit integer
tns.sns	Secure Network Services	Boolean
tns.type	Packet Type	Unsigned 8-bit integer
tns.version	Version	Unsigned 16-bit integer

Table A-201. Transparent Network Substrate Protocol (tns)

A.202. Trivial File Transfer Protocol (tftp)

Field	Field Name	Type
tftp.block	Block	Unsigned 16-bit integer
tftp.destination_file	DESTINATION File	String
tftp.error.code	Error code	Unsigned 16-bit integer
tftp.error.message	Error message	String
tftp.opcode	Opcode	Unsigned 16-bit integer
tftp.source_file	Source File	String
tftp.type	Type	String

Table A-202. Trivial File Transfer Protocol (tftp)

A.203. User Datagram Protocol (udp)

Field	Field Name	Type
udp.checksum	Checksum	Unsigned 16-bit integer
udp.checksum_bad	Bad Checksum	Boolean
udp.dstport	Destination Port	Unsigned 16-bit integer
udp.length	Length	Unsigned 16-bit integer
udp.port	Source or Destination Port	Unsigned 16-bit integer
udp.srcport	Source Port	Unsigned 16-bit integer

Table A-203. User Datagram Protocol (udp)

A.204. Virtual Router Redundancy Protocol (vrrp)

Field	Field Name	Type
vrrp.adver_int	Adver Int	Unsigned 8-bit integer
vrrp.auth_type	Auth Type	Unsigned 8-bit integer
vrrp.count_ip_addrs	Count IP Addrs	Unsigned 8-bit integer
vrrp.ip_addr	IP Address	IPv4 address
vrrp.prio	Priority	Unsigned 8-bit integer
vrrp.type	VRRP packet type	Unsigned 8-bit integer
vrrp.typever	VRRP message version and type	Unsigned 8-bit integer
vrrp.version	VRRP protocol version	Unsigned 8-bit integer
vrrp.virt_rtr_id	Virtual Rtr ID	Unsigned 8-bit integer

Table A-204. Virtual Router Redundancy Protocol (vrrp)

A.205. Virtual Trunking Protocol (vtp)

Field	Field Name	Type
vtp.code	Code	Unsigned 8-bit integer

Field	Field Name	Type
vtp.conf_rev_num	Configuration Revision Number	Unsigned 32-bit integer
vtp.followers	Followers	Unsigned 8-bit integer
vtp.md	Management Domain	String
vtp.md5_digest	MD5 Digest	Byte array
vtp.md_len	Management Domain Length	Unsigned 8-bit integer
vtp.seq_num	Sequence Number	Unsigned 8-bit integer
vtp.start_value	Start Value	Unsigned 16-bit integer
vtp.upd_id	Updater Identity	IPv4 address
vtp.upd_ts	Update Timestamp	String
vtp.version	Version	Unsigned 8-bit integer
vtp.vlan_info.802_10_index	802.10 Index	Unsigned 32-bit integer
vtp.vlan_info.isl_vlan_id	ISL VLAN ID	Unsigned 16-bit integer
vtp.vlan_info.len	VLAN Information Length	Unsigned 8-bit integer
vtp.vlan_info.mtu_size	MTU Size	Unsigned 16-bit integer
vtp.vlan_info.status.vlan_susp	VLAN suspended	Boolean
vtp.vlan_info.tlv_len	Length	Unsigned 8-bit integer
vtp.vlan_info.tlv_type	Type	Unsigned 8-bit integer
vtp.vlan_info.vlan_name	VLAN Name	String
vtp.vlan_info.vlan_name_len	VLAN Name Length	Unsigned 8-bit integer
vtp.vlan_info.vlan_type	VLAN Type	Unsigned 8-bit integer

Table A-205. Virtual Trunking Protocol (vtp)

A.206. Web Cache Coordination Protocol (wccp)

Field	Field Name	Type
wccp.cache_ip	Web Cache IP address	IPv4 address
wccp.change_num	Change Number	Unsigned 32-bit integer
wccp.hash_revision	Hash Revision	Unsigned 32-bit integer
wccp.message	WCCP Message Type	Unsigned 32-bit integer

Field	Field Name	Type
wccp.recvd_id	Received ID	Unsigned 32-bit integer
wccp.version	WCCP Version	Unsigned 32-bit integer

Table A-206. Web Cache Coordination Protocol (wccp)

A.207. Wellfleet Compression (wcp)

Field	Field Name	Type
wcp.alg	Alg	Unsigned 8-bit integer
wcp.alg1	Alg 1	Unsigned 8-bit integer
wcp.alg2	Alg 2	Unsigned 8-bit integer
wcp.alg3	Alg 3	Unsigned 8-bit integer
wcp.alg4	Alg 4	Unsigned 8-bit integer
wcp.alg_cnt	Alg Count	Unsigned 8-bit integer
wcp.checksum	Checksum	Unsigned 8-bit integer
wcp.cmd	Command	Unsigned 8-bit integer
wcp.ext_cmd	Extended Command	Unsigned 8-bit integer
wcp.flag	Compress Flag	Unsigned 8-bit integer
wcp.hist	History	Unsigned 8-bit integer
wcp.init	Initiator	Unsigned 8-bit integer
wcp.long_comp	Long Compression	Unsigned 16-bit integer
wcp.long_len	Compress Length	Unsigned 8-bit integer
wcp.mark	Compress Marker	Unsigned 8-bit integer
wcp.off	Source offset	Unsigned 16-bit integer
wcp.pib	PIB	Unsigned 8-bit integer
wcp.ppc	PerPackComp	Unsigned 8-bit integer
wcp.rev	Revision	Unsigned 8-bit integer
wcp.rexmit	Rexmit	Unsigned 8-bit integer
wcp.seq	SEQ	Unsigned 16-bit integer
wcp.seq_size	Seq Size	Unsigned 8-bit integer
wcp.short_comp	Short Compression	Unsigned 8-bit integer
wcp.short_len	Compress Length	Unsigned 8-bit integer
wcp.tid	TID	Unsigned 16-bit integer

Table A-207. Wellfleet Compression (wcp)

A.208. Who (who)

Field	Field Name	Type
who.boottime	Boot Time	Date/Time stamp
who.hostname	Hostname	String
who.idle	Time Idle	Unsigned 32-bit integer
who.loadav_10	Load Average Over Past 10 Minutes	Double-precision floating point
who.loadav_15	Load Average Over Past 15 Minutes	Double-precision floating point
who.loadav_5	Load Average Over Past 5 Minutes	Double-precision floating point
who.recvtime	Receive Time	Date/Time stamp
who.sendtime	Send Time	Date/Time stamp
who.timeon	Time On	Date/Time stamp
who.tty	TTY Name	String
who.type	Type	Unsigned 8-bit integer
who.uid	User ID	String
who.vers	Version	Unsigned 8-bit integer
who.whoent	Who utmp Entry	No value

Table A-208. Who (who)

A.209. Wireless Session Protocol (wap-wsp)

Field	Field Name	Type
wsp.TID	Transmission ID	Unsigned 8-bit integer
wsp.capabilities	Capabilities	No value
wsp.capabilities.aliases	Aliases	Unsigned 8-bit integer
wsp.capabilities.client_SDU	Client SDU	Unsigned 8-bit integer
wsp.capabilities.code_pages	Header Code Pages	String
wsp.capabilities.extend_methods	Extended Methods	String

Field	Field Name	Type
wsp.capabilities.method_mor	Method MOR	Unsigned 8-bit integer
wsp.capabilities.protocol_opt	Protocol Options	String
wsp.capabilities.push_mor	Push MOR	Unsigned 8-bit integer
wsp.capabilities.server_SDU	Server SDU	Unsigned 8-bit integer
wsp.capability.length	Capability Length	Unsigned 32-bit integer
wsp.content_type.parameter.charset	Charset	Unsigned 16-bit integer
wsp.content_type.type	Content Type	Unsigned 8-bit integer
wsp.header.accept	Accept	Unsigned 8-bit integer
wsp.header.accept.string	Accept	String
wsp.header.accept_charset	Accept-Charset	Unsigned 16-bit integer
wsp.header.accept_charset.string	Accept-Charset	String
wsp.header.accept_language	Accept-Language	Unsigned 8-bit integer
wsp.header.accept_language.string	Accept-Language	String
wsp.header.accept_ranges	Accept-Ranges	Unsigned 8-bit integer
wsp.header.age	Age	Unsigned 32-bit integer
wsp.header.application_header	Application Header	String
wsp.header.application_header.value	Application Header Value	String
wsp.header.cache_control	Cache-Control	Unsigned 8-bit integer
wsp.header.content_length	Content-Length	Unsigned 32-bit integer
wsp.header.date	Date	Date/Time stamp
wsp.header.etag	Etag	String
wsp.header.expires	Expires	Date/Time stamp
wsp.header.if_modified_since	If-Modified-Since	Date/Time stamp
wsp.header.last_modified	Last-Modified	Date/Time stamp
wsp.header.location	Location	String

Field	Field Name	Type
wsp.header.server	Server	String
wsp.header.shift	Shift code	Unsigned 8-bit integer
wsp.header.transfer_enc	Transfer Encoding	Unsigned 8-bit integer
wsp.header.transfer_enc_str	Transfer Encoding	String
wsp.header.user_agent	User-Agent	String
wsp.header.via	Via	String
wsp.header.x_wap_tod	X-WAP.TOD	Date/Time stamp
wsp.headers	Headers	No value
wsp.headers.header	Header	No value
wsp.headers_length	Headers Length	Unsigned 32-bit integer
wsp.pdu_type	PDU Type	Unsigned 8-bit integer
wsp.post.data	Post Data	No value
wsp.reply.data	Data	No value
wsp.reply.status	Status	Unsigned 8-bit integer
wsp.server.session_id	Server Session ID	Unsigned 32-bit integer
wsp.uri	URI	String
wsp.uri_length	URI Length	Unsigned 32-bit integer
wsp.version.major	Version (Major)	Unsigned 8-bit integer
wsp.version.minor	Version (Minor)	Unsigned 8-bit integer

Table A-209. Wireless Session Protocol (wap-wsp)

A.210. Wireless Transaction Protocol (wap-wsp-wtp)

Field	Field Name	Type
wtp.RID	Re-transmission Indicator	Boolean
wtp.TID	Transmission ID	Unsigned 16-bit integer
wtp.TID.response	TID Response	Boolean
wtp.abort.reason.provider	Abort Reason	Unsigned 8-bit integer
wtp.abort.reason.user	Abort Reason	Unsigned 8-bit integer
wtp.abort.type	Abort Type	Unsigned 8-bit integer
wtp.ack.tvetok	Tve/Tok flag	Boolean
wtp.continue_flag	Continue Flag	Boolean

Field	Field Name	Type
wtp.header.TIDNew	TIDNew	Boolean
wtp.header.UP	U/P flag	Boolean
wtp.header.missing_packets	Missing Packets	Unsigned 8-bit integer
wtp.header.sequence	Packet Sequence Number	Unsigned 8-bit integer
wtp.header.version	Version	Unsigned 8-bit integer
wtp.header_data	Data	Byte array
wtp.header_fixed_part	Header	Byte array
wtp.header_variable_part	Header: Variable part	Byte array
wtp.inv.reserved	Reserved	Unsigned 8-bit integer
wtp.inv.transaction_class	Transaction Class	Unsigned 8-bit integer
wtp.pdu_type	PDU Type	Unsigned 8-bit integer
wtp.trailer_flags	Trailer Flags	Unsigned 8-bit integer

Table A-210. Wireless Transaction Protocol (wap-wsp-wtp)

A.211. Wireless Transport Layer Security (wap-wtls)

Field	Field Name	Type
wsp.wtls.alert	Alert	No value
wsp.wtls.alert.description	Description	Unsigned 8-bit integer
wsp.wtls.alert.level	Level	Unsigned 8-bit integer
wsp.wtls.handshake	Handshake	Unsigned 8-bit integer
wsp.wtls.handshake.certificate	Certificate	No value
wsp.wtls.handshake.certificate.after	Validity not after	Date/Time stamp
wsp.wtls.handshake.certificate.before	Validity not before	Date/Time stamp
wsp.wtls.handshake.certificate.issuer.charset	Charset	Unsigned 16-bit integer
wsp.wtls.handshake.certificate.issuer.name	Name	String

Field	Field Name	Type
wsp.wtls.handshake.certificate.issuer	Issuer.type	Unsigned 8-bit integer
wsp.wtls.handshake.certificate.parameter_set	Parameter Set	String
wsp.wtls.handshake.certificate.parameter_index	Parameter Index	Unsigned 8-bit integer
wsp.wtls.handshake.certificate.public_key_type	Public Key Type	Unsigned 8-bit integer
wsp.wtls.handshake.certificate.rsa_exponent_size	RSA Exponent Size	Unsigned 32-bit integer
wsp.wtls.handshake.certificate.rsa_modulus_size	RSA Modulus Size	Unsigned 32-bit integer
wsp.wtls.handshake.certificate.signature_size	Signature Size	Unsigned 32-bit integer
wsp.wtls.handshake.certificate.signature_type	Signature Type	Unsigned 8-bit integer
wsp.wtls.handshake.certificate.subject.charset	Charset	Unsigned 16-bit integer
wsp.wtls.handshake.certificate.subject.name	Name	String
wsp.wtls.handshake.certificate.subject.type	Subject.type	Unsigned 8-bit integer
wsp.wtls.handshake.certificate.type	Type	Unsigned 8-bit integer
wsp.wtls.handshake.certificate.version	Version	Unsigned 8-bit integer
wsp.wtls.handshake.certificates	Certificates	No value
wsp.wtls.handshake.client_hello	Client Hello	No value
wsp.wtls.handshake.client_hello.cipher	Cipher	String
wsp.wtls.handshake.client_hello.cipher_suites	Cipher Suites	No value
wsp.wtls.handshake.client_hello.client_keys	Client Keys	No value
wsp.wtls.handshake.client_hello.client_keys_id	keys_id	No value
wsp.wtls.handshake.client_hello.client_keys_len	Length	Unsigned 16-bit integer

Field	Field Name	Type
wsp.wtls.handshake.client_hello.compression_methods	Compression Methods	No value
wsp.wtls.handshake.client_hello.compression	Compression	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.gmt	Time GMT	Date/Time stamp
wsp.wtls.handshake.client_hello.identifier_charset	Identifier Charset	Unsigned 16-bit integer
wsp.wtls.handshake.client_hello.identifier_name	Identifier Name	String
wsp.wtls.handshake.client_hello.identifier_size	Identifier Size	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.identifier_type	Identifier Type	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.identifier	Identifier	No value
wsp.wtls.handshake.client_hello.key_exchange	Key Exchange	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.key_exchange_suite	Suite	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.parameter_set	Parameter Set	String
wsp.wtls.handshake.client_hello.parameter_index	Parameter Index	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.random	Random	No value
wsp.wtls.handshake.client_hello.refresh	Refresh	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.sequence_mode	Sequence Mode	Unsigned 8-bit integer
wsp.wtls.handshake.client_hello.session_id.str	Session ID	String
wsp.wtls.handshake.client_hello.session_id	Session ID	Unsigned 32-bit integer
wsp.wtls.handshake.client_hello.trusted_keys	Trusted Keys	No value
wsp.wtls.handshake.client_hello.version	Version	Unsigned 8-bit integer

Field	Field Name	Type
wsp.wtls.handshake.length	Length	Unsigned 16-bit integer
wsp.wtls.handshake.server_hello	Server Hello	No value
wsp.wtls.handshake.server_hello.cipher	Cipher	No value
wsp.wtls.handshake.server_hello.cipher.bulk	Cipher Bulk	Unsigned 8-bit integer
wsp.wtls.handshake.server_hello.cipher.mac	Cipher MAC	Unsigned 8-bit integer
wsp.wtls.handshake.server_hello.compression	Compression	Unsigned 8-bit integer
wsp.wtls.handshake.server_hello.gmt	Time GMT	Date/Time stamp
wsp.wtls.handshake.server_hello.key	Client Key ID	Unsigned 8-bit integer
wsp.wtls.handshake.server_hello.random	Random	No value
wsp.wtls.handshake.server_hello.refresh	Refresh	Unsigned 8-bit integer
wsp.wtls.handshake.server_hello.sequence_mode	Sequence Mode	Unsigned 8-bit integer
wsp.wtls.handshake.server_hello.session	Session ID	String
wsp.wtls.handshake.server_hello.sessionid	Session ID	Unsigned 32-bit integer
wsp.wtls.handshake.server_hello.version	Version	Unsigned 8-bit integer
wsp.wtls.handshake.type	Type	Unsigned 8-bit integer
wsp.wtls.rec_cipher	Record Ciphered	No value
wsp.wtls.rec_length	Record Length	Unsigned 16-bit integer
wsp.wtls.rec_seq	Record Sequence	Unsigned 16-bit integer
wsp.wtls.rec_type	Record Type	Unsigned 8-bit integer
wsp.wtls.record	Record	Unsigned 8-bit integer

Table A-211. Wireless Transport Layer Security (wap-wtls)

A.212. X.25 (x.25)

Field	Field Name	Type
x.25.a	A Bit	Boolean
x.25.d	D Bit	Boolean
x.25.gfi	GFI	Unsigned 16-bit integer
x.25.lcn	Logical Channel	Unsigned 16-bit integer
x.25.m	M Bit	Boolean
x.25.mod	Modulo	Unsigned 16-bit integer
x.25.p_r	P(R)	Unsigned 8-bit integer
x.25.p_s	P(S)	Unsigned 8-bit integer
x.25.q	Q Bit	Boolean
x.25.type	Packet Type	Unsigned 8-bit integer

Table A-212. X.25 (x.25)

A.213. X.25 over TCP (xot)

Field	Field Name	Type

Table A-213. X.25 over TCP (xot)

A.214. X11 (x11)

Field	Field Name	Type
x11.acceleration-denominator	acceleration-denominator	Signed 16-bit integer
x11.acceleration-numerator	acceleration-numerator	Signed 16-bit integer
x11.access-mode	access-mode	Unsigned 8-bit integer
x11.address	address	Byte array

Field	Field Name	Type
x11.address-length	address-length	Unsigned 16-bit integer
x11.alloc	alloc	Unsigned 8-bit integer
x11.allow-events-mode	allow-events-mode	Unsigned 8-bit integer
x11.allow-exposures	allow-exposures	Unsigned 8-bit integer
x11.arc	arc	No value
x11.arc.angle1	angle1	Signed 16-bit integer
x11.arc.angle2	angle2	Signed 16-bit integer
x11.arc.height	height	Unsigned 16-bit integer
x11.arc.mode	mode	Unsigned 8-bit integer
x11.arc.width	width	Unsigned 16-bit integer
x11.arc.x	x	Signed 16-bit integer
x11.arc.y	y	Signed 16-bit integer
x11.arcs	arcs	No value
x11.atom	atom	Unsigned 32-bit integer
x11.auto-repeat-mode	auto-repeat-mode	Unsigned 8-bit integer
x11.back-blue	back-blue	Unsigned 16-bit integer
x11.back-green	back-green	Unsigned 16-bit integer
x11.back-red	back-red	Unsigned 16-bit integer
x11.background	background	Unsigned 32-bit integer
x11.background-pixel	background-pixel	Unsigned 32-bit integer
x11.background-pixmap	background-pixmap	Unsigned 32-bit integer
x11.backing-pixel	backing-pixel	Unsigned 32-bit integer
x11.backing-planes	backing-planes	Unsigned 32-bit integer
x11.backing-store	backing-store	Unsigned 8-bit integer
x11.bell-duration	bell-duration	Signed 16-bit integer
x11.bell-percent	bell-percent	Signed 8-bit integer
x11.bell-pitch	bell-pitch	Signed 16-bit integer
x11.bit-gravity	bit-gravity	Unsigned 8-bit integer
x11.bit-plane	bit-plane	Unsigned 32-bit integer
x11.blue	blue	Unsigned 16-bit integer
x11.blues	blues	Unsigned 16-bit integer
x11.border-pixel	border-pixel	Unsigned 32-bit integer
x11.border-pixmap	border-pixmap	Unsigned 32-bit integer
x11.border-width	border-width	Unsigned 16-bit integer
x11.button	button	Unsigned 8-bit integer
x11.cap-style	cap-style	Unsigned 8-bit integer
x11.change-host-mode	change-host-mode	Unsigned 8-bit integer

Field	Field Name	Type
x11.cid	cid	Unsigned 32-bit integer
x11.class	class	Unsigned 8-bit integer
x11.clip-mask	clip-mask	Unsigned 32-bit integer
x11.clip-x-origin	clip-x-origin	Signed 16-bit integer
x11.clip-y-origin	clip-y-origin	Signed 16-bit integer
x11.close-down-mode	close-down-mode	Unsigned 8-bit integer
x11.cmap	cmap	Unsigned 32-bit integer
x11.color-items	color-items	No value
x11.coloritem	coloritem	No value
x11.coloritem.blue	blue	Unsigned 16-bit integer
x11.coloritem.flags	flags	Unsigned 8-bit integer
x11.coloritem.flags.do-blue	do-blue	Boolean
x11.coloritem.flags.do-green	do-green	Boolean
x11.coloritem.flags.do-red	do-red	Boolean
x11.coloritem.flags.unused	unused	Boolean
x11.coloritem.green	green	Unsigned 16-bit integer
x11.coloritem.pixel	pixel	Unsigned 32-bit integer
x11.coloritem.red	red	Unsigned 16-bit integer
x11.coloritem.unused	unused	No value
x11.colormap	colormap	Unsigned 32-bit integer
x11.colors	colors	Unsigned 16-bit integer
x11.configure-window-mask	configure-window-mask	Unsigned 16-bit integer
x11.configure-window-mask.border-width	border-width	Boolean
x11.configure-window-mask.height	height	Boolean

Field	Field Name	Type
x11.configure-window-mask.sibling	sibling	Boolean
x11.configure-window-mask.stack-mode	stack-mode	Boolean
x11.configure-window-mask.width	width	Boolean
x11.configure-window-mask.x	x	Boolean
x11.configure-window-mask.y	y	Boolean
x11.confine-to	confine-to	Unsigned 32-bit integer
x11.contiguous	contiguous	Boolean
x11.coordinate-mode	coordinate-mode	Unsigned 8-bit integer
x11.count	count	Unsigned 8-bit integer
x11.cursor	cursor	Unsigned 32-bit integer
x11.dash-offset	dash-offset	Unsigned 16-bit integer
x11.dashes	dashes	Byte array
x11.dashes-length	dashes-length	Unsigned 16-bit integer
x11.data	data	Byte array
x11.data-length	data-length	Unsigned 32-bit integer
x11.delete	delete	Boolean
x11.delta	delta	Signed 16-bit integer
x11.depth	depth	Unsigned 8-bit integer
x11.direction	direction	Unsigned 8-bit integer
x11.do-acceleration	do-acceleration	Boolean
x11.do-not-propagate-mask	do-not-propagate-mask	Unsigned 32-bit integer
x11.do-not-propagate-mask.Button1Motion	Button1Motion	Boolean

Field	Field Name	Type
x11.do-not-propagate-mask.Button2Motion	Button2Motion	Boolean
x11.do-not-propagate-mask.Button3Motion	Button3Motion	Boolean
x11.do-not-propagate-mask.Button4Motion	Button4Motion	Boolean
x11.do-not-propagate-mask.Button5Motion	Button5Motion	Boolean
x11.do-not-propagate-mask.ButtonMotion	ButtonMotion	Boolean
x11.do-not-propagate-mask.ButtonPress	ButtonPress	Boolean
x11.do-not-propagate-mask.ButtonRelease	ButtonRelease	Boolean
x11.do-not-propagate-mask.KeyPress	KeyPress	Boolean
x11.do-not-propagate-mask.KeyRelease	KeyRelease	Boolean
x11.do-not-propagate-mask.PointerMotion	PointerMotion	Boolean
x11.do-not-propagate-mask.erroneous-bits	erroneous-bits	Boolean
x11.do-threshold	do-threshold	Boolean
x11.drawable	drawable	Unsigned 32-bit integer
x11.dst-drawable	dst-drawable	Unsigned 32-bit integer
x11.dst-gc	dst-gc	Unsigned 32-bit integer
x11.dst-window	dst-window	Unsigned 32-bit integer
x11.dst-x	dst-x	Signed 16-bit integer

Field	Field Name	Type
x11.dst-y	dst-y	Signed 16-bit integer
x11.event-mask	event-mask	Unsigned 32-bit integer
x11.event-mask.Button1Motion	Button1Motion	Boolean
x11.event-mask.Button2Motion	Button2Motion	Boolean
x11.event-mask.Button3Motion	Button3Motion	Boolean
x11.event-mask.Button4Motion	Button4Motion	Boolean
x11.event-mask.Button5Motion	Button5Motion	Boolean
x11.event-mask.ButtonMotion	ButtonMotion	Boolean
x11.event-mask.ButtonPress	ButtonPress	Boolean
x11.event-mask.ButtonRelease	ButtonRelease	Boolean
x11.event-mask.ColormapChange	ColormapChange	Boolean
x11.event-mask.EnterWindow	EnterWindow	Boolean
x11.event-mask.Exposure	Exposure	Boolean
x11.event-mask.FocusChange	FocusChange	Boolean
x11.event-mask.KeyPress	KeyPress	Boolean

Field	Field Name	Type
x11.event-mask.KeyRelease	KeyRelease	Boolean
x11.event-mask.KeymapState	KeymapState	Boolean
x11.event-mask.LeaveWindow	LeaveWindow	Boolean
x11.event-mask.OwnerGrabButton	OwnerGrabButton	Boolean
x11.event-mask.PointerMotion	PointerMotion	Boolean
x11.event-mask.PointerMotionHint	PointerMotionHint	Boolean
x11.event-mask.PropertyChange	PropertyChange	Boolean
x11.event-mask.ResizeRedirect	ResizeRedirect	Boolean
x11.event-mask.StructureNotify	StructureNotify	Boolean
x11.event-mask.SubstructureNotify	SubstructureNotify	Boolean
x11.event-mask.SubstructureRedirect	SubstructureRedirect	Boolean
x11.event-mask.VisibilityChange	VisibilityChange	Boolean
x11.event-mask.erroneous-bits	erroneous-bits	Boolean

Field	Field Name	Type
x11.exposures	exposures	Boolean
x11.family	family	Unsigned 8-bit integer
x11.fid	fid	Unsigned 32-bit integer
x11.fill-rule	fill-rule	Unsigned 8-bit integer
x11.fill-style	fill-style	Unsigned 8-bit integer
x11.first-keycode	first-keycode	Unsigned 8-bit integer
x11.focus	focus	Unsigned 8-bit integer
x11.font	font	Unsigned 32-bit integer
x11.fore-blue	fore-blue	Unsigned 16-bit integer
x11.fore-green	fore-green	Unsigned 16-bit integer
x11.fore-red	fore-red	Unsigned 16-bit integer
x11.foreground	foreground	Unsigned 32-bit integer
x11.format	format	Unsigned 8-bit integer
x11.function	function	Unsigned 8-bit integer
x11.gc	gc	Unsigned 32-bit integer
x11.gc-dashes	gc-dashes	Unsigned 8-bit integer
x11.gc-value-mask	gc-value-mask	Unsigned 32-bit integer
x11.gc-value-mask.arc-mode	arc-mode	Boolean
x11.gc-value-mask.background	background	Boolean
x11.gc-value-mask.cap-style	cap-style	Boolean
x11.gc-value-mask.clip-mask	clip-mask	Boolean
x11.gc-value-mask.clip-x-origin	clip-x-origin	Boolean
x11.gc-value-mask.clip-y-origin	clip-y-origin	Boolean
x11.gc-value-mask.dash-offset	dash-offset	Boolean

Field	Field Name	Type
x11.gc-value-mask.fill-rule	fill-rule	Boolean
x11.gc-value-mask.fill-style	fill-style	Boolean
x11.gc-value-mask.font	font	Boolean
x11.gc-value-mask.foreground	foreground	Boolean
x11.gc-value-mask.function	function	Boolean
x11.gc-value-mask.gc-dashes	gc-dashes	Boolean
x11.gc-value-mask.graphics-exposures	graphics-exposures	Boolean
x11.gc-value-mask.join-style	join-style	Boolean
x11.gc-value-mask.line-style	line-style	Boolean
x11.gc-value-mask.line-width	line-width	Boolean
x11.gc-value-mask.plane-mask	plane-mask	Boolean
x11.gc-value-mask.stipple	stipple	Boolean
x11.gc-value-mask.subwindow-mode	subwindow-mode	Boolean
x11.gc-value-mask.tile	tile	Boolean

Field	Field Name	Type
x11.gc-value-mask.tile-stipple-x-origin	tile-stipple-x-origin	Boolean
x11.gc-value-mask.tile-stipple-y-origin	tile-stipple-y-origin	Boolean
x11.get-property-type	get-property-type	Unsigned 32-bit integer
x11.grab_window	grab_window	Unsigned 32-bit integer
x11.graphics-exposures	graphics-exposures	Boolean
x11.green	green	Unsigned 16-bit integer
x11.greens	greens	Unsigned 16-bit integer
x11.height	height	Unsigned 16-bit integer
x11.image-format	image-format	Unsigned 8-bit integer
x11.image-pixmap-format	image-pixmap-format	Unsigned 8-bit integer
x11.interval	interval	Signed 16-bit integer
x11.items	items	No value
x11.join-style	join-style	Unsigned 8-bit integer
x11.key	key	Unsigned 8-bit integer
x11.key-click-percent	key-click-percent	Signed 8-bit integer
x11.keyboard-key	keyboard-key	Unsigned 8-bit integer
x11.keyboard-mode	keyboard-mode	Unsigned 8-bit integer
x11.keyboard-value-mask	keyboard-value-mask	Unsigned 32-bit integer
x11.keyboard-value-mask.auto-repeat-mode	auto-repeat-mode	Boolean
x11.keyboard-value-mask.bell-duration	bell-duration	Boolean
x11.keyboard-value-mask.bell-percent	bell-percent	Boolean
x11.keyboard-value-mask.bell-pitch	bell-pitch	Boolean
x11.keyboard-value-mask.key-click-percent	key-click-percent	Boolean

Field	Field Name	Type
x11.keyboard-value-mask.keyboard-key	keyboard-key	Boolean
x11.keyboard-value-mask.led	led	Boolean
x11.keyboard-value-mask.led-mode	led-mode	Boolean
x11.keycode-count	keycode-count	Unsigned 8-bit integer
x11.keycodes	keycodes	No value
x11.keycodes-per-modifier	keycodes-per-modifier	Unsigned 8-bit integer
x11.keycodes.item	item	Byte array
x11.keysyms	keysyms	No value
x11.keysyms-per-keycode	keysyms-per-keycode	Unsigned 8-bit integer
x11.keysyms.item	item	No value
x11.keysyms.item.keysym	keysym	Unsigned 32-bit integer
x11.led	led	Unsigned 8-bit integer
x11.led-mode	led-mode	Unsigned 8-bit integer
x11.left-pad	left-pad	Unsigned 8-bit integer
x11.line-style	line-style	Unsigned 8-bit integer
x11.line-width	line-width	Unsigned 16-bit integer
x11.long-length	long-length	Unsigned 32-bit integer
x11.long-offset	long-offset	Unsigned 32-bit integer
x11.map	map	Byte array
x11.map-length	map-length	Unsigned 8-bit integer
x11.mask	mask	Unsigned 32-bit integer
x11.mask-char	mask-char	Unsigned 16-bit integer
x11.mask-font	mask-font	Unsigned 32-bit integer
x11.max-names	max-names	Unsigned 16-bit integer
x11.mid	mid	Unsigned 32-bit integer
x11.mode	mode	Unsigned 8-bit integer
x11.modifiers-mask	modifiers-mask	Unsigned 16-bit integer

Field	Field Name	Type
x11.modifiers-mask.AnyModifier	AnyModifier	Unsigned 16-bit integer
x11.modifiers-mask.Control	Control	Boolean
x11.modifiers-mask.Lock	Lock	Boolean
x11.modifiers-mask.Mod1	Mod1	Boolean
x11.modifiers-mask.Mod2	Mod2	Boolean
x11.modifiers-mask.Mod3	Mod3	Boolean
x11.modifiers-mask.Mod4	Mod4	Boolean
x11.modifiers-mask.Mod5	Mod5	Boolean
x11.modifiers-mask.Shift	Shift	Boolean
x11.modifiers-mask.erroneous-bits	erroneous-bits	Boolean
x11.name	name	String
x11.name-length	name-length	Unsigned 16-bit integer
x11.odd-length	odd-length	Boolean
x11.only-if-exists	only-if-exists	Boolean
x11.opcode	opcode	Unsigned 8-bit integer
x11.ordering	ordering	Unsigned 8-bit integer
x11.override-redirect	override-redirect	Boolean
x11.owner	owner	Unsigned 32-bit integer
x11.owner-events	owner-events	Boolean
x11.parent	parent	Unsigned 32-bit integer
x11.path	path	No value
x11.path.string	string	String
x11.pattern	pattern	String
x11.pattern-length	pattern-length	Unsigned 16-bit integer
x11.percent	percent	Unsigned 8-bit integer
x11.pid	pid	Unsigned 32-bit integer
x11.pixel	pixel	Unsigned 32-bit integer

Field	Field Name	Type
x11.pixels	pixels	No value
x11.pixels_item	pixels_item	Unsigned 32-bit integer
x11.pixmap	pixmap	Unsigned 32-bit integer
x11.plane-mask	plane-mask	Unsigned 32-bit integer
x11.planes	planes	Unsigned 16-bit integer
x11.point	point	No value
x11.point-x	point-x	Signed 16-bit integer
x11.point-y	point-y	Signed 16-bit integer
x11.pointer-event-mask	pointer-event-mask	Unsigned 16-bit integer
x11.pointer-event-mask.Button1Motion	Button1Motion	Boolean
x11.pointer-event-mask.Button2Motion	Button2Motion	Boolean
x11.pointer-event-mask.Button3Motion	Button3Motion	Boolean
x11.pointer-event-mask.Button4Motion	Button4Motion	Boolean
x11.pointer-event-mask.Button5Motion	Button5Motion	Boolean
x11.pointer-event-mask.ButtonMotion	ButtonMotion	Boolean
x11.pointer-event-mask.ButtonPress	ButtonPress	Boolean
x11.pointer-event-mask.ButtonRelease	ButtonRelease	Boolean
x11.pointer-event-mask.EnterWindow	EnterWindow	Boolean
x11.pointer-event-mask.KeymapState	KeymapState	Boolean

Field	Field Name	Type
x11.pointer-event-mask.LeaveWindow	LeaveWindow	Boolean
x11.pointer-event-mask.PointerMotion	PointerMotion	Boolean
x11.pointer-event-mask.PointerMotionHint	PointerMotionHint	Boolean
x11.pointer-event-mask.erroneous-bits	erroneous-bits	Boolean
x11.pointer-mode	pointer-mode	Unsigned 8-bit integer
x11.points	points	No value
x11.prefer-blanking	prefer-blanking	Unsigned 8-bit integer
x11.properties	properties	No value
x11.properties.item	item	Unsigned 32-bit integer
x11.property	property	Unsigned 32-bit integer
x11.property-number	property-number	Unsigned 16-bit integer
x11.rectangle	rectangle	No value
x11.rectangle-height	rectangle-height	Unsigned 16-bit integer
x11.rectangle-width	rectangle-width	Unsigned 16-bit integer
x11.rectangle-x	rectangle-x	Signed 16-bit integer
x11.rectangle-y	rectangle-y	Signed 16-bit integer
x11.rectangles	rectangles	No value
x11.red	red	Unsigned 16-bit integer
x11.reds	reds	Unsigned 16-bit integer
x11.request	request	Unsigned 8-bit integer
x11.request-length	request-length	Unsigned 16-bit integer
x11.requestor	requestor	Unsigned 32-bit integer
x11.resource	resource	Unsigned 32-bit integer
x11.revert-to	revert-to	Unsigned 8-bit integer
x11.save-set-mode	save-set-mode	Unsigned 8-bit integer
x11.save-under	save-under	Boolean
x11.screen-saver-mode	screen-saver-mode	Unsigned 8-bit integer
x11.segment	segment	No value
x11.segment_x1	segment_x1	Signed 16-bit integer

Field	Field Name	Type
x11.segment_x2	segment_x2	Signed 16-bit integer
x11.segment_y1	segment_y1	Signed 16-bit integer
x11.segment_y2	segment_y2	Signed 16-bit integer
x11.segments	segments	No value
x11.selection	selection	Unsigned 32-bit integer
x11.shape	shape	Unsigned 8-bit integer
x11.sibling	sibling	Unsigned 32-bit integer
x11.source-char	source-char	Unsigned 16-bit integer
x11.source-font	source-font	Unsigned 32-bit integer
x11.source-pixmap	source-pixmap	Unsigned 32-bit integer
x11.src-cmap	src-cmap	Unsigned 32-bit integer
x11.src-drawable	src-drawable	Unsigned 32-bit integer
x11.src-gc	src-gc	Unsigned 32-bit integer
x11.src-height	src-height	Unsigned 16-bit integer
x11.src-width	src-width	Unsigned 16-bit integer
x11.src-window	src-window	Unsigned 32-bit integer
x11.src-x	src-x	Signed 16-bit integer
x11.src-y	src-y	Signed 16-bit integer
x11.stack-mode	stack-mode	Unsigned 8-bit integer
x11.start	start	Unsigned 32-bit integer
x11.stipple	stipple	Unsigned 32-bit integer
x11.stop	stop	Unsigned 32-bit integer
x11.str-number-in-path	str-number-in-path	Unsigned 16-bit integer
x11.string	string	String
x11.string-length	string-length	Unsigned 32-bit integer
x11.string16	string16	String
x11.string16.bytes	bytes	Byte array
x11.subwindow-mode	subwindow-mode	Unsigned 8-bit integer
x11.target	target	Unsigned 32-bit integer
x11.textitem	textitem	No value
x11.textitem.font	font	Unsigned 32-bit integer
x11.textitem.string	string	No value
x11.textitem.string.delta	delta	Signed 8-bit integer
x11.textitem.string.string16	string16	String
x11.textitem.string.string16.bytes	bytes	Byte array

Field	Field Name	Type
x11.textitem.string.string8	string8	String
x11.threshold	threshold	Signed 16-bit integer
x11.tile	tile	Unsigned 32-bit integer
x11.tile-stipple-x-origin	tile-stipple-x-origin	Signed 16-bit integer
x11.tile-stipple-y-origin	tile-stipple-y-origin	Signed 16-bit integer
x11.time	time	Unsigned 32-bit integer
x11.timeout	timeout	Signed 16-bit integer
x11.type	type	Unsigned 32-bit integer
x11.undecoded	undecoded	No value
x11.unused	unused	No value
x11.visual	visual	Unsigned 32-bit integer
x11.visualid	visualid	Unsigned 32-bit integer
x11.warp-pointer-dst-window	warp-pointer-dst-window	Unsigned 32-bit integer
x11.warp-pointer-src-window	warp-pointer-src-window	Unsigned 32-bit integer
x11.wid	wid	Unsigned 32-bit integer
x11.width	width	Unsigned 16-bit integer
x11.win-gravity	win-gravity	Unsigned 8-bit integer
x11.window	window	Unsigned 32-bit integer
x11.window-class	window-class	Unsigned 16-bit integer
x11.window-value-mask	window-value-mask	Unsigned 32-bit integer
x11.window-value-mask.background-pixel	background-pixel	Boolean
x11.window-value-mask.background-pixmap	background-pixmap	Boolean
x11.window-value-mask.backing-pixel	backing-pixel	Boolean
x11.window-value-mask.backing-planes	backing-planes	Boolean

Field	Field Name	Type
x11.window-value-mask.backing-store	backing-store	Boolean
x11.window-value-mask.bit-gravity	bit-gravity	Boolean
x11.window-value-mask.border-pixel	border-pixel	Boolean
x11.window-value-mask.border-pixmap	border-pixmap	Boolean
x11.window-value-mask.colormap	colormap	Boolean
x11.window-value-mask.cursor	cursor	Boolean
x11.window-value-mask.do-not-propagate-mask	do-not-propagate-mask	Boolean
x11.window-value-mask.event-mask	event-mask	Boolean
x11.window-value-mask.override-redirect	override-redirect	Boolean
x11.window-value-mask.save-under	save-under	Boolean
x11.window-value-mask.win-gravity	win-gravity	Boolean
x11.x	x	Signed 16-bit integer
x11.y	y	Signed 16-bit integer

Table A-214. X11 (x11)

A.215. Yahoo Messenger Protocol (yhoo)

Field	Field Name	Type
yhoo.connection_id	Connection ID	Unsigned 32-bit integer
yhoo.content	Content	String
yhoo.len	Packet Length	Unsigned 32-bit integer
yhoo.magic_id	Magic ID	Unsigned 32-bit integer
yhoo.msgtype	Message Type	Unsigned 32-bit integer
yhoo.nick1	Real Nick (nick1)	String
yhoo.nick2	Active Nick (nick2)	String
yhoo.service	Service Type	Unsigned 32-bit integer
yhoo.unknown1	Unknown 1	Unsigned 32-bit integer
yhoo.version	Version	String

Table A-215. Yahoo Messenger Protocol (yhoo)

A.216. Yellow Pages Bind (ypbind)

Field	Field Name	Type

Table A-216. Yellow Pages Bind (ypbind)

A.217. Yellow Pages Passwd (yppasswd)

Field	Field Name	Type
yppasswd.newpw	newpw	No value
yppasswd.newpw.dir	dir	String
yppasswd.newpw.gecos	gecos	String
yppasswd.newpw.gid	gid	Unsigned 32-bit integer
yppasswd.newpw.name	name	String
yppasswd.newpw.passwd	passwd	String
yppasswd.newpw.shell	shell	String

Field	Field Name	Type
yppasswd.newpw.uid	uid	Unsigned 32-bit integer
yppasswd.oldpass	oldpass	String
yppasswd.status	status	Unsigned 32-bit integer

Table A-217. Yellow Pages Passwd (yppasswd)

A.218. Yellow Pages Service (ypserv)

Field	Field Name	Type
ypserv.domain	Domain	String
ypserv.key	Key	String
ypserv.map	Map Name	String
ypserv.map_parms	YP Map Parameters	No value
ypserv.more	More	Boolean
ypserv.ordernum	Order Number	Unsigned 32-bit integer
ypserv.peer	Peer Name	String
ypserv.port	Port	Unsigned 32-bit integer
ypserv.prog	Program Number	Unsigned 32-bit integer
ypserv.servesdomain	Serves Domain	Boolean
ypserv.status	Status	Signed 32-bit integer
ypserv.transid	Host Transport ID	IPv4 address
ypserv.value	Value	String
ypserv.xfrstat	Xfrstat	Signed 32-bit integer

Table A-218. Yellow Pages Service (ypserv)

A.219. Yellow Pages Transfer (ypxfr)

Field	Field Name	Type

Table A-219. Yellow Pages Transfer (ypxfr)

A.220. Zebra Protocol (zebra)

Field	Field Name	Type
zebra.bandwidth	Bandwidth	Unsigned 32-bit integer
zebra.command	Command	Unsigned 8-bit integer
zebra.dest4	Destination	IPv4 address
zebra.dest6	Destination	IPv6 address
zebra.distance	Distance	Unsigned 8-bit integer
zebra.family	Family	Unsigned 32-bit integer
zebra.index	Index	Unsigned 32-bit integer
zebra.indexnum	Index Number	Unsigned 8-bit integer
zebra.interface	Interface	String
zebra.intflags	Flags	Unsigned 32-bit integer
zebra.len	Length	Unsigned 16-bit integer
zebra.message	Message	Unsigned 8-bit integer
zebra.message.distance	Message Distance	Boolean
zebra.message.index	Message Index	Boolean
zebra.message.metric	Message Metric	Boolean
zebra.message.nexthop	Message Nexthop	Boolean
zebra.metric	Metric	Unsigned 32-bit integer
zebra.mtu	MTU	Unsigned 32-bit integer
zebra.nexthop4	Nexthop	IPv4 address
zebra.nexthop6	Nexthop	IPv6 address
zebra.nexthopnum	Nexthop Number	Unsigned 8-bit integer
zebra.prefix4	Prefix	IPv4 address
zebra.prefix6	Prefix	IPv6 address
zebra.prefixlen	Prefix length	Unsigned 32-bit integer
zebra.request	Request	Boolean
zebra.rtflags	Flags	Unsigned 8-bit integer
zebra.type	Type	Unsigned 8-bit integer

Table A-220. Zebra Protocol (zebra)

A.221. iSCSI (iscsi)

Field	Field Name	Type
iscsi.bufferOffset	BufferOffset	Unsigned 32-bit integer
iscsi.cid	CID	Unsigned 16-bit integer
iscsi.cmdsn	CmdSN	Unsigned 32-bit integer
iscsi.commandstatus	CommandStatus	Unsigned 8-bit integer
iscsi.datasegmentlength	DataSegmentLength	Unsigned 32-bit integer
iscsi.datasn	DataSN	Unsigned 32-bit integer
iscsi.desireddatalength	DesiredDataLength	Unsigned 32-bit integer
iscsi.expcmdsn	ExpCmdSN	Unsigned 32-bit integer
iscsi.expdatasn	ExpCmdSN	Unsigned 32-bit integer
iscsi.expstatsn	ExpStatSN	Unsigned 32-bit integer
iscsi.flags	Flags	Unsigned 8-bit integer
iscsi.headerdigest	HeaderDigest	Unsigned 32-bit integer
iscsi.initcmdsn	InitCmdSN	Unsigned 32-bit integer
iscsi.initiatortasktag	InitiatorTaskTag	Unsigned 32-bit integer
iscsi.initstatsn	InitStatSN	Unsigned 32-bit integer
iscsi.iscsievent	iSCSIEvent	Unsigned 8-bit integer
iscsi.isid	ISID	Unsigned 16-bit integer
iscsi.keyvalue	KeyValue	String
iscsi.length	Length	Unsigned 32-bit integer
iscsi.login.f	F	Boolean
iscsi.login.status	Status	Unsigned 8-bit integer
iscsi.logout.reason	Reason	Unsigned 8-bit integer
iscsi.logout.response	Response	Unsigned 8-bit integer
iscsi.lun	LUN	Byte array
iscsi.maxcmdsn	MaxCmdSN	Unsigned 32-bit integer
iscsi.nop.p	P	Boolean
iscsi.opcode	Opcode	Unsigned 8-bit integer
iscsi.padding	Padding	Byte array
iscsi.parameter1	Parameter1	Unsigned 16-bit integer
iscsi.parameter2	Parameter2	Unsigned 16-bit integer
iscsi.parameter3	Parameter3	Unsigned 16-bit integer
iscsi.payload	Payload	Byte array
iscsi.r2texpdatasn	R2TExpCmdSN	Unsigned 32-bit integer
iscsi.reject.firstbadbyte	FirstBadByte	Unsigned 16-bit integer
iscsi.reject.reason	Reason	Unsigned 8-bit integer
iscsi.scsicommand.addcdb	AddCDB	Unsigned 8-bit integer

Field	Field Name	Type
iscsi.scsicommand.attr	Attr	Unsigned 8-bit integer
iscsi.scsicommand.cdb	CDB	Byte array
iscsi.scsicommand.cdb0	CDB	Unsigned 8-bit integer
iscsi.scsicommand.crn	CRN	Unsigned 8-bit integer
iscsi.scsicommand.expecteddatatransferlength	ExpectedDataTransferLength	Unsigned 32-bit integer
iscsi.scsicommand.f	F	Boolean
iscsi.scsicommand.r	R	Boolean
iscsi.scsicommand.x	X	Boolean
iscsi.scsidata.O	O	Boolean
iscsi.scsidata.U	U	Boolean
iscsi.scsidata.f	F	Boolean
iscsi.scsidata.p	P	Boolean
iscsi.scsidata.readresidualcount	ReadResidualCount	Unsigned 32-bit integer
iscsi.scsidata.s	S	Boolean
iscsi.scsievent	SCSIEvent	Unsigned 8-bit integer
iscsi.scsiresponse.O	O	Boolean
iscsi.scsiresponse.S	S	Boolean
iscsi.scsiresponse.U	U	Boolean
iscsi.scsiresponse.basicresidualcount	BasicResidualCount	Unsigned 32-bit integer
iscsi.scsiresponse.bidireadresidualcount	BidiReadResidualCount	Unsigned 32-bit integer
iscsi.scsiresponse.o	o	Boolean
iscsi.scsiresponse.senselength	SenseLength	Unsigned 16-bit integer
iscsi.scsiresponse.statusresponse	Status/Response	Unsigned 8-bit integer
iscsi.scsiresponse.u	u	Boolean
iscsi.scsitask.function	Function	Unsigned 8-bit integer
iscsi.scsitask.referencedtasktag	InitiatorTaskTag	Unsigned 32-bit integer
iscsi.scsitask.response	Response	Unsigned 8-bit integer
iscsi.snack.additionalruns	AdditionalRuns	Byte array
iscsi.snack.addruns	AddRuns	Unsigned 8-bit integer
iscsi.snack.begrun	BegRun	Unsigned 32-bit integer

Field	Field Name	Type
iscsi.snack.runlength	RunLength	Unsigned 32-bit integer
iscsi.snack.s	S	Boolean
iscsi.statsn	StatSN	Unsigned 32-bit integer
iscsi.targettransfertag	TargetTransferTag	Unsigned 32-bit integer
iscsi.text.f	F	Boolean
iscsi.totalahslength	TotalAHSLength	Unsigned 8-bit integer
iscsi.tsid	TSID	Unsigned 16-bit integer
iscsi.versionmax	VersionMax	Unsigned 8-bit integer
iscsi.versionmin	VersionMin	Unsigned 8-bit integer

Table A-221. iSCSI (iscsi)

B. Ethereal Error Messages

B.1. Capture file format not understood

If Ethereal cannot decode the capture file format of the file you have asked it to load, you will receive a warning box similar to that shown in Figure B-1.

Figure B-1. Ethereal Read Format warning

B.2. Save file error

If Ethereal cannot open the file you requested it to save captured packets in, you will receive a warning box similar to that shown in Figure B-2.

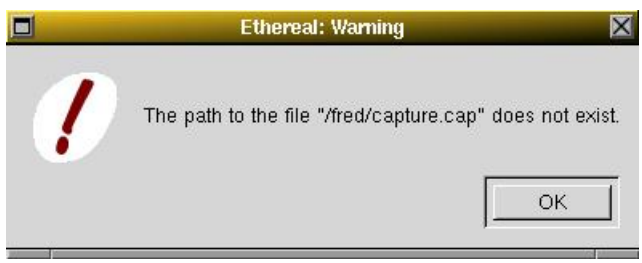

Figure B-2. Save Error warning

C. The GNU Free Document Public Licence

C.1. Copyright

Version 1.1, March 2000

C.2. Preamble

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.3. Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

C.4. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C.5. Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide

you with an updated version of the Document.

C.6. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled "Acknowledgements" or "Dedications", preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C.7. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

C.8. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C.9. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other

self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

C.10. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

C.11. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

C.12. Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as

a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.