Detailed Design

Requested by: Ken Swarner
System Administrator
Siena College
Computer Science Department

TCP/IP Packet Descriptor

Paradigm Solutions

Prepared by: Jonathan Baker, Team Leader
Ryan Fischer
Mike Sebast
Justin Waterman
Jim DeSario
Mark Mossman

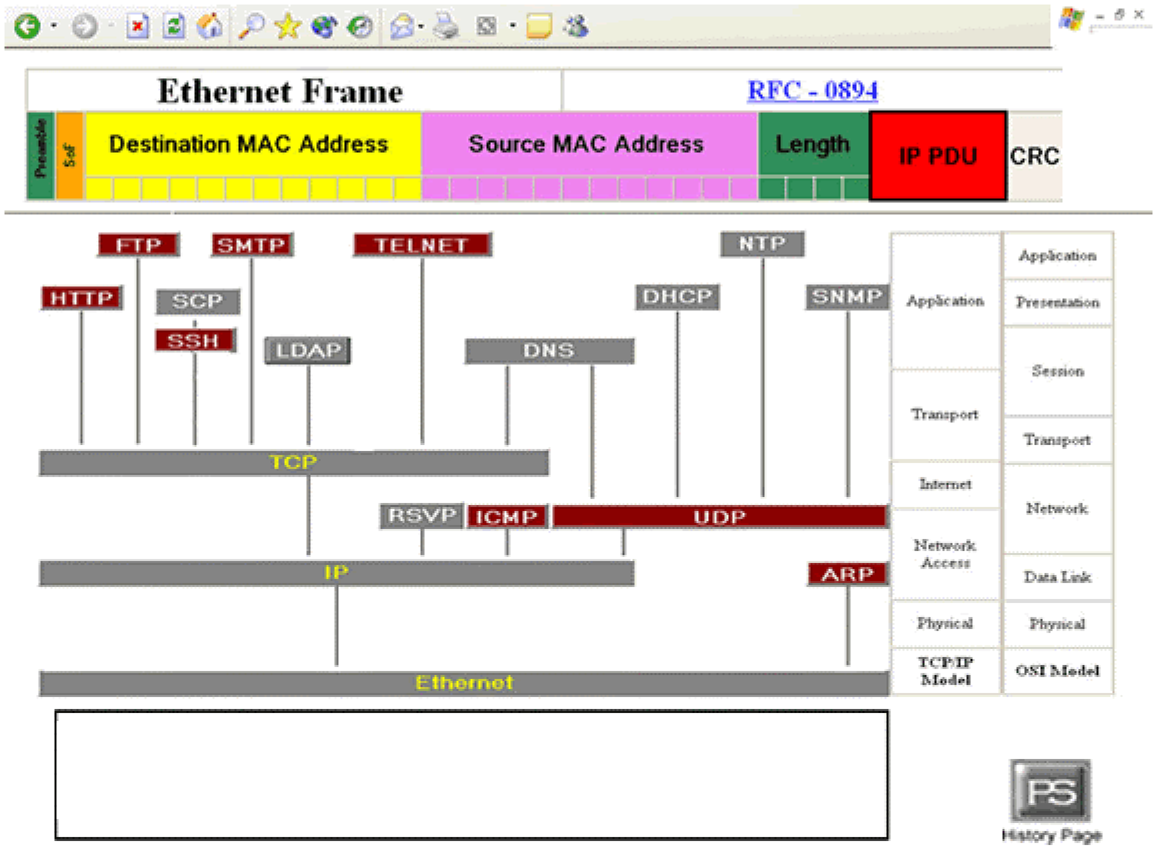
February 28, 2005

Detailed Design
Table of Contents

1.0 External Design Specifications

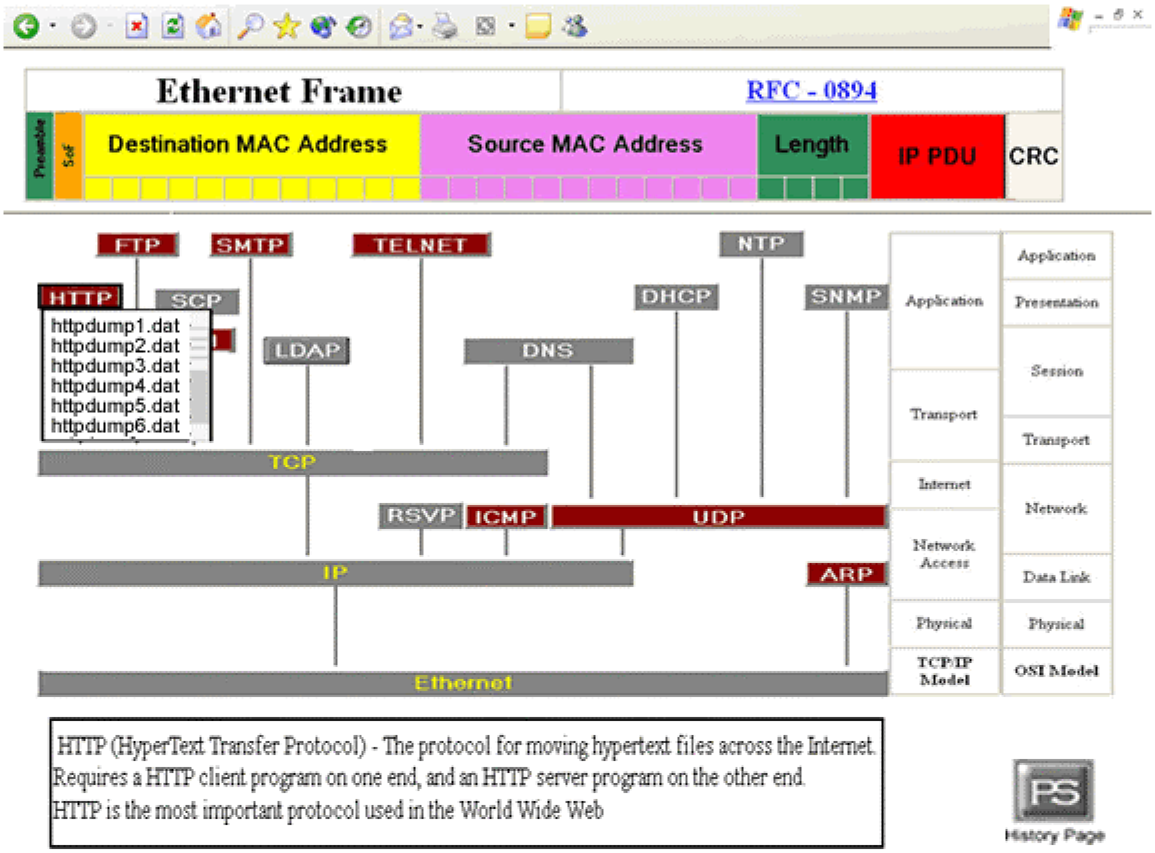
1.1 User displays

In order to effectively portray the user displays, the following pages contain a pictorial representation of a user navigating through the HTTP protocol selection. After each picture there is a brief summary of what the user would have selected in order to make each screen appear.

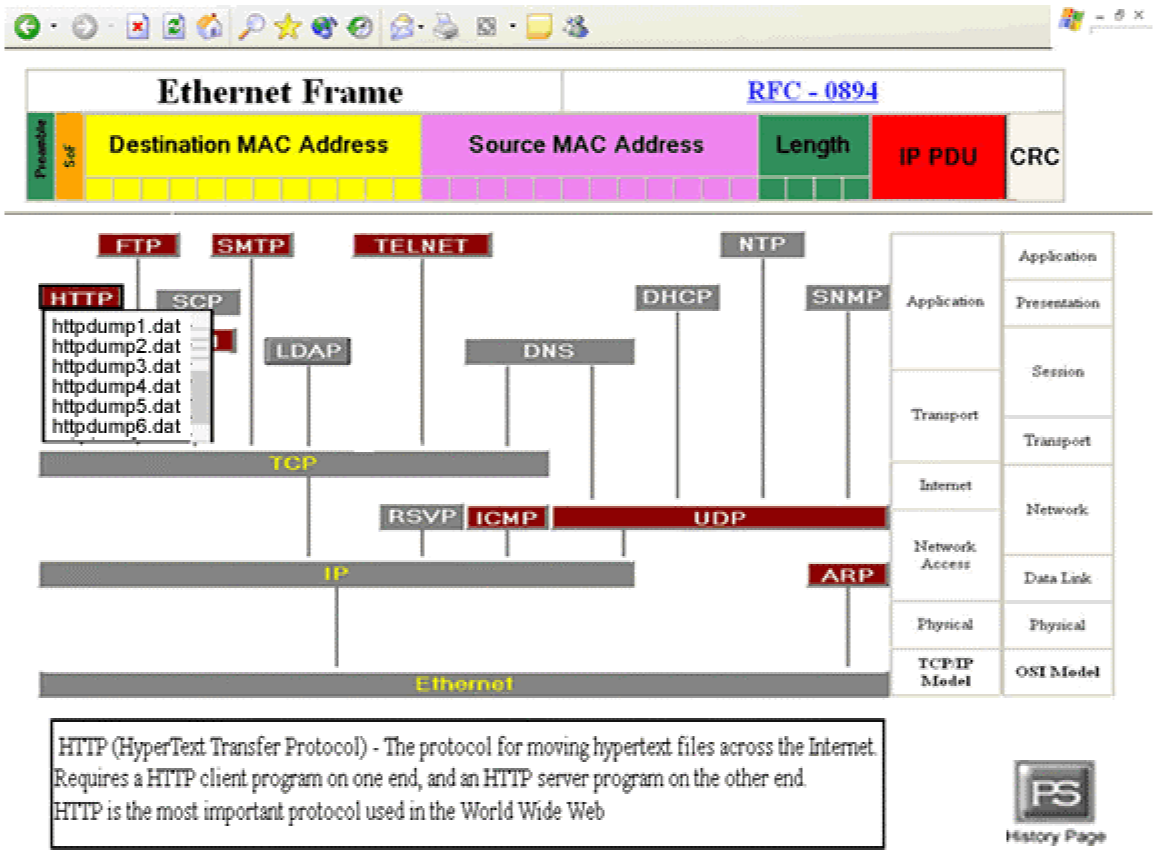


This is the first screen that the user will see when he logs onto the program. This screen consists of a protocol tree in the middle, with an accompanying Ethernet stack to the right of it, measuring where each protocol lies in the tree. Additionally, we have an Ethernet frame at the top of the screen that displays information about the destination, source, and other information about it. Any button when selected will cause the box at the bottom to display topic information about what was selected. Also there is a history page link in the bottom right corner, which upon clicking will bring the user to a history page with previous teams web pages. The history page link is a uniform feature throughout the screens.

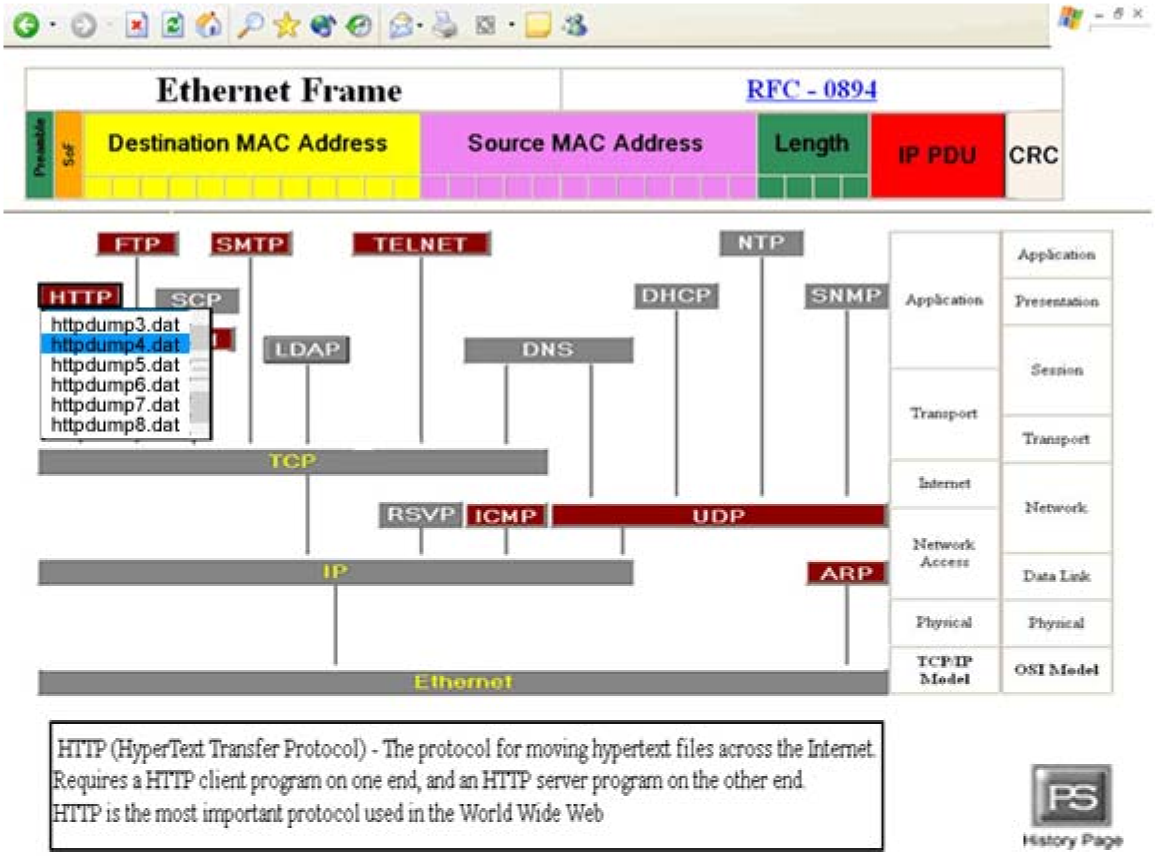
This is the history page that the user will be sent to if the icon is selected. The user can click on any previous teams link to view their homepages, to return to the previous screen the user clicks on the back button.



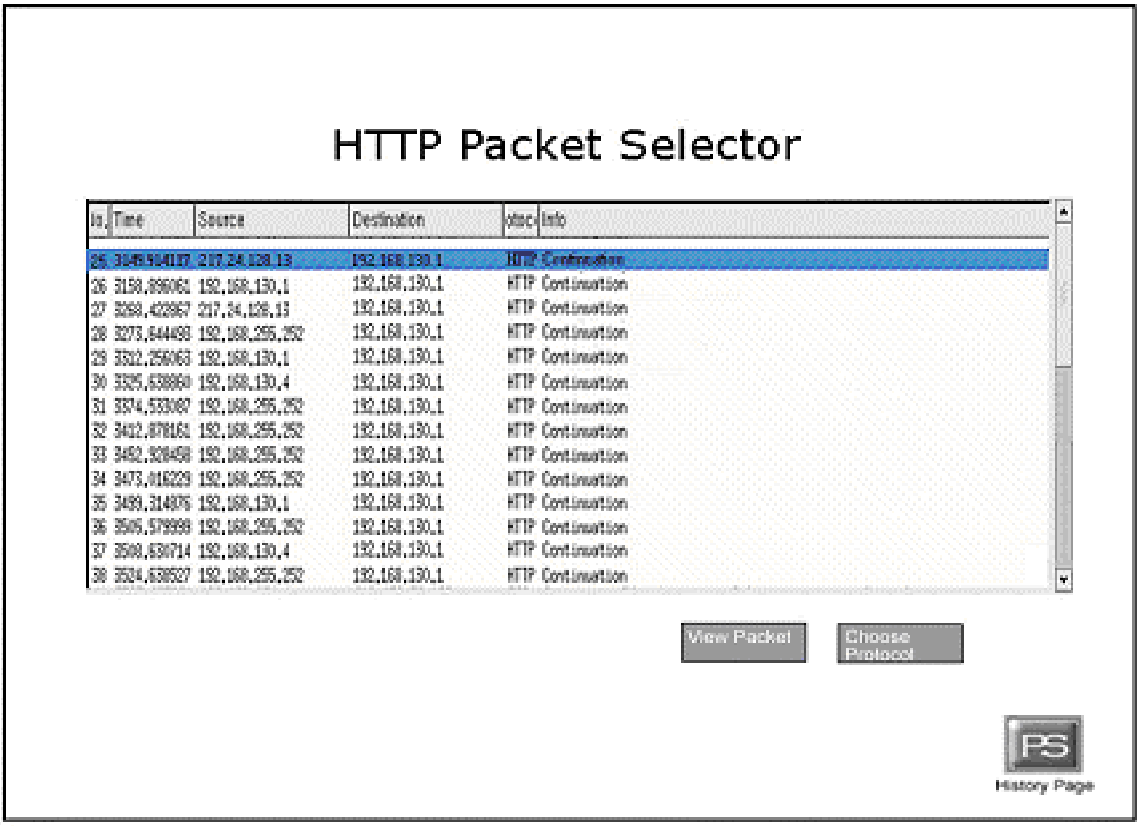
In the middle of the screen is a protocol tree, which the user may select any of the non-shaded out protocols to display a list of data sessions. In this example, the HTTP protocol was selected, and a drop-down menu of data sessions has appeared. The user then scrolls to the particular data field that he is interested in, and then clicks on it to take him or her to the Packet Selection screen.



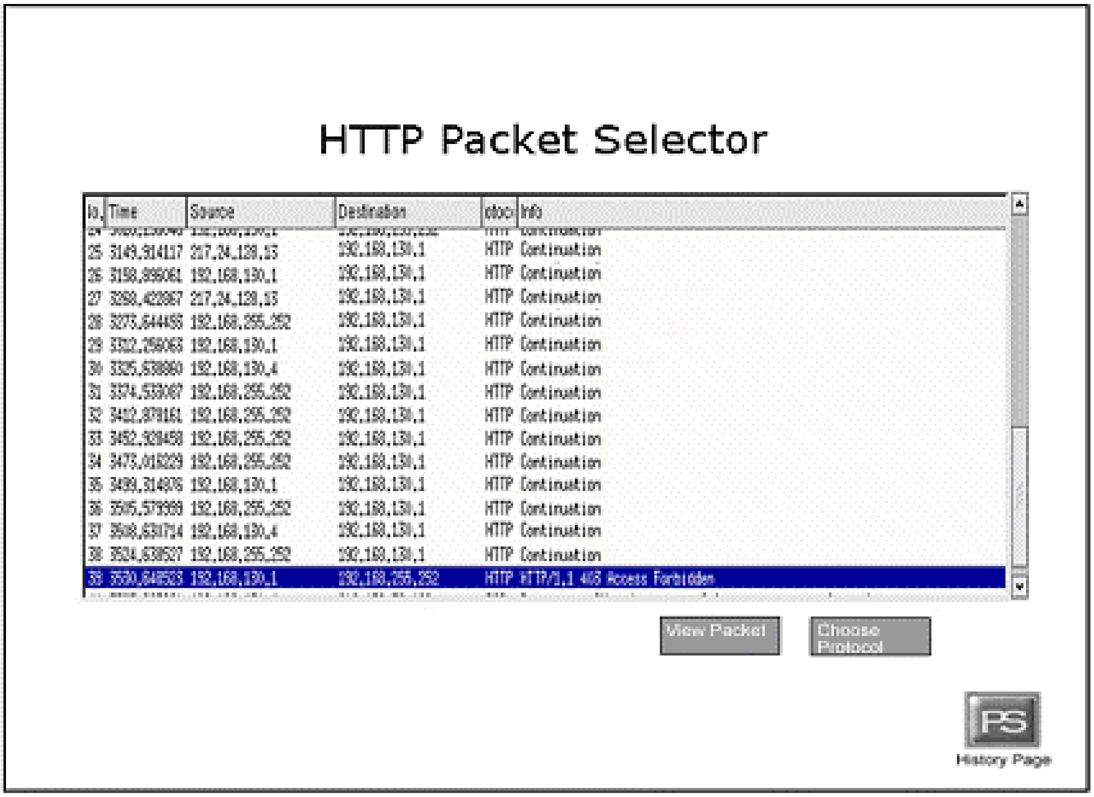
The user can scroll down the list using the scroll bar. This screen shows the user had scrolled down to view other data sessions.



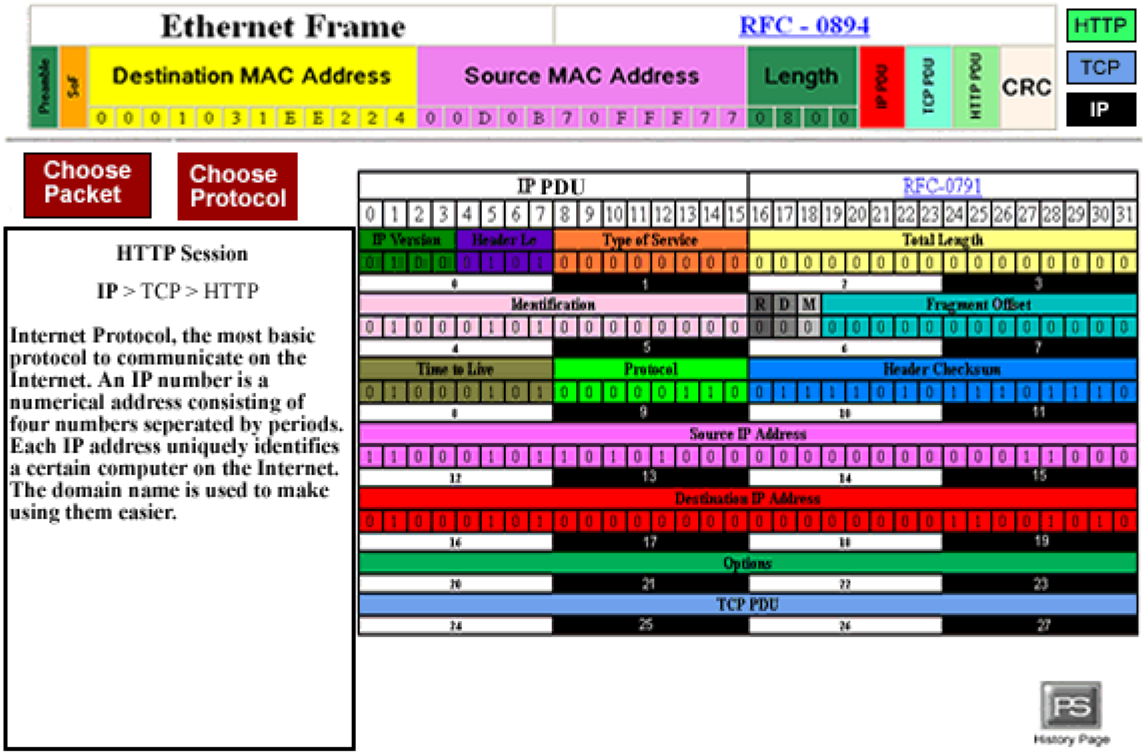
When the user moves the mouse over a data session, the session becomes highlighted. The data session then becomes un-highlighted when the mouse moves off of the data session. This screen shows the mouse highlighting a particular data session. The user would click the highlighted data session to advance to the Packet Selection screen.



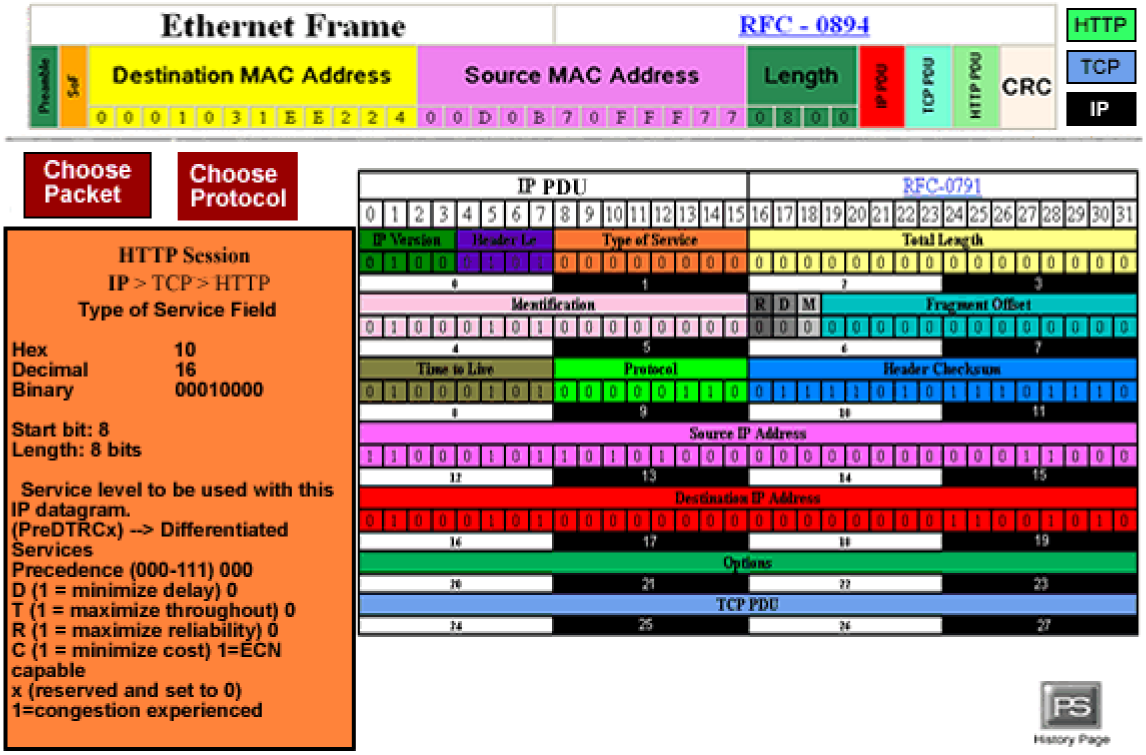
This is a Packet Selection Screen as the user first sees it. Its title at the top reflects the protocol that was selected. From before, the HTTP protocol was selected. The box in the middle of the screen displays the different packets of the selected protocol as well as other relevant information about the packet. The screen is initialized to highlight the topmost packet, which is the default packet. Above this, we have repeated the data session name for the data session that was selected from the Protocol Selector screen. There are two buttons below this packet box: a View Packet Button, which takes you to the Information Display Screen and display the highlighted packet; and a Choose Protocol Button, which when clicked returns you to the Protocol Selection Screen. Additionally the user can display the contents of a packet by double clicking the packet on this screen.



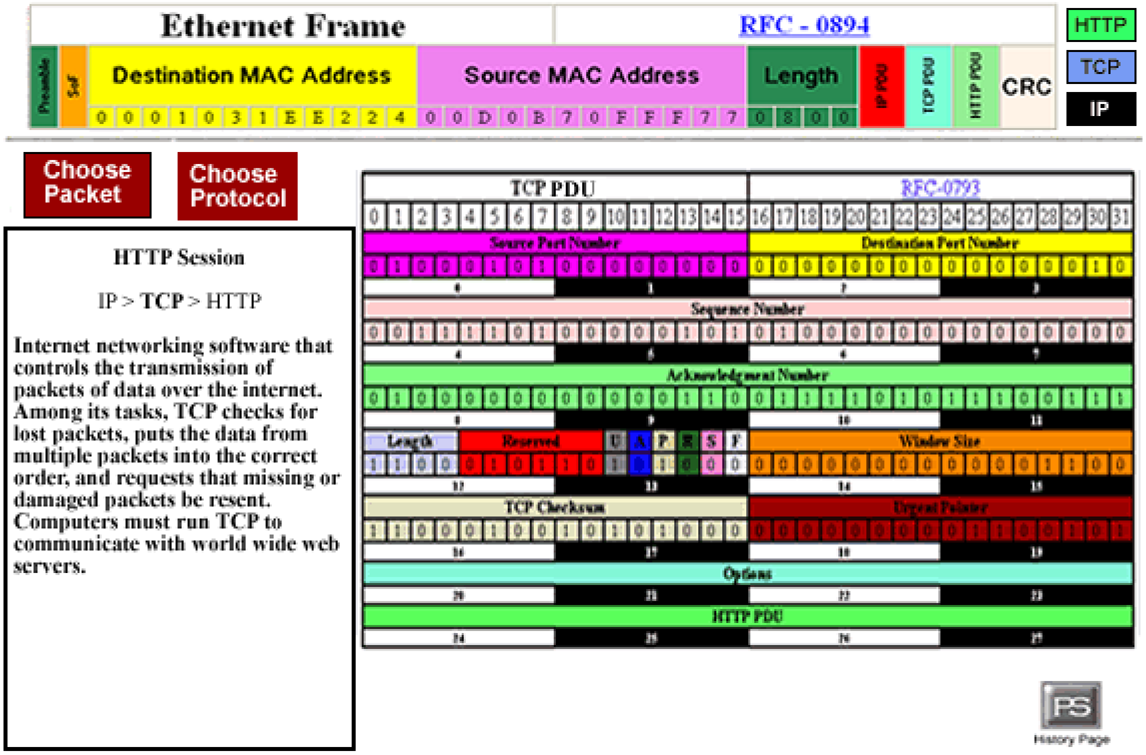
If there are more packets in the data session than the box is capable of displaying, the scroll bar on the right allows the user to scroll through the different packets available in the data session. This is an example of the user scrolling to see more packets. The user can highlight a particular packet by clicking on its row. This action will un-highlight the previously highlighted packet. Clicking on a currently highlighted packet does nothing. This screen demonstrates the highlighting of a packet other than the default packet.



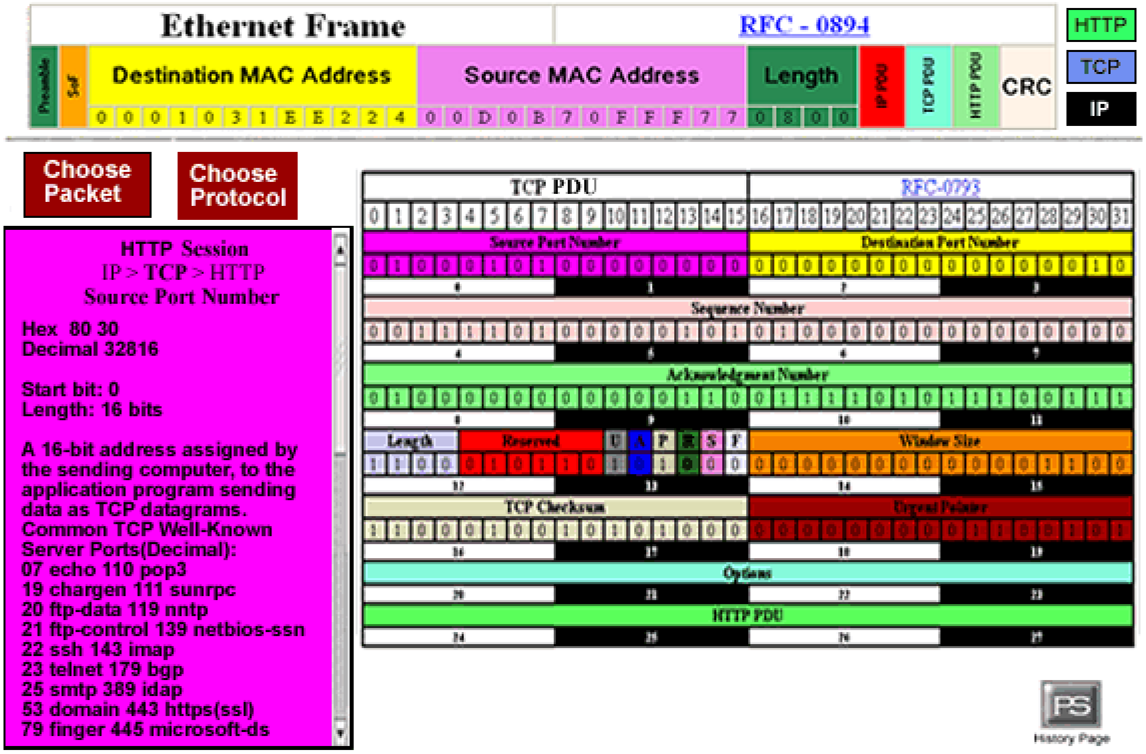
This is the Information Display Screen. This screen displays the contents of the packet that was selected in the Packet Selector Screen. The two buttons on the upper left, the Choose Packet Button and the Choose Protocol Button, bring the user to the Packet Selector Screen and the Protocol Selection Screen, respectively. There are tabs that would allow the user to traverse the protocol tree, less the Ethernet root. Since, in this example the user had selected an HTTP packet; the tabs at the bottom of the screen are IP, TCP, and HTTP. Everything on this screen except for the packet fields is geographically static, that is they will not change in size or location on this screen. The user first sees the highest level of protocol, in this case HTTP. This is the screen as the user first sees it. The protocol stack in the upper right corner allows the user to navigate through the different protocols within the protocol hierarchy. At the top of the screen there is an Ethernet Frame that is now filled with data.



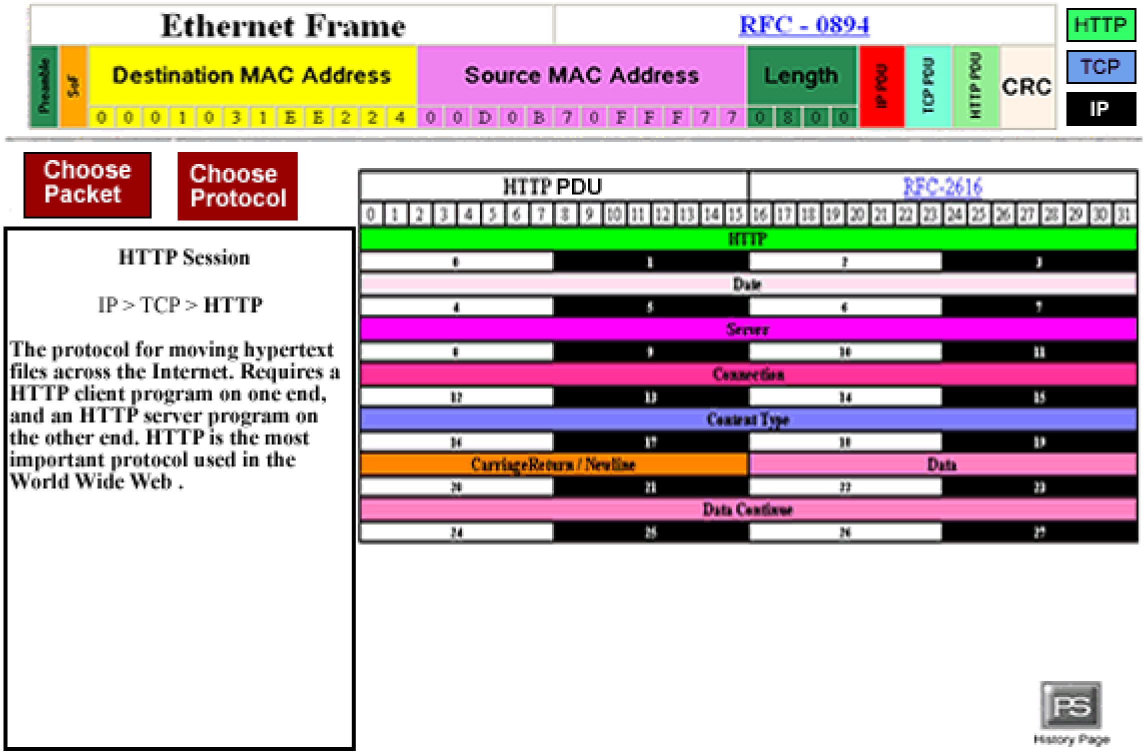
Clicking on any of the fields' causes the program to display both information about the field and the data contained in that field in the Information Box, located on the left of the screen. This is an example of the user clicking on the Type of Service field.



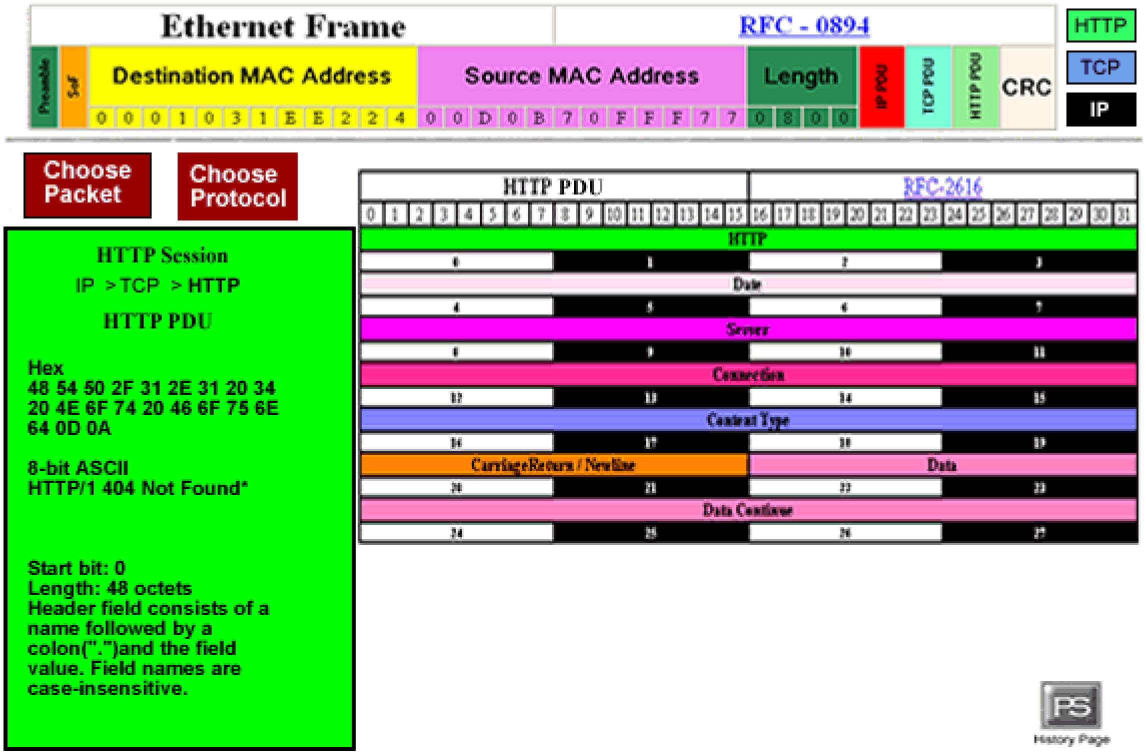
This is an example of the user selecting the TCP protocol, as the user first sees it. The information box displays the selected packet's TCP protocol information.



Clicking on a different field clears out the information box and displays the information of the newly selected field. This is an example of the user selecting Source Port Number field.



The user went to the HTTP protocol, the information box displays information about the HTTP protocol.



The user selected the HTTP PDU field. The information box displays the information within that field.

1.2 Detailed Data Flow Diagrams

<u>Key</u>:

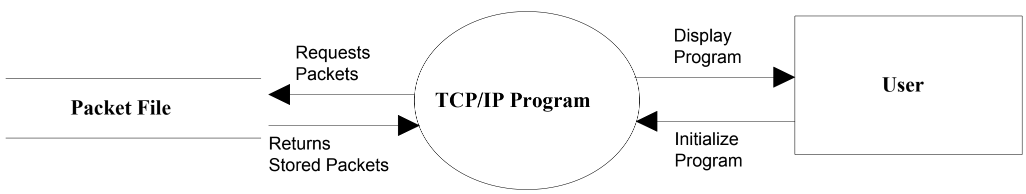
☐ — Source/Sink

○ — Process

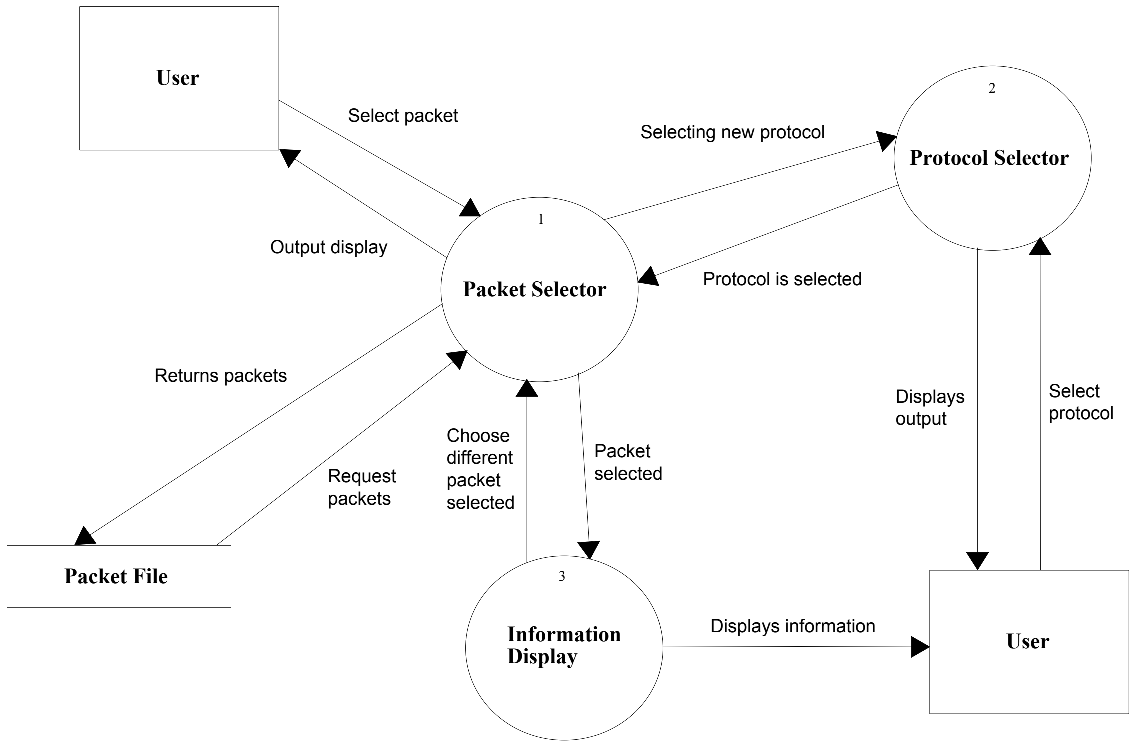
═ — Data Stores

⟶ — Data Flow

Context Diagram

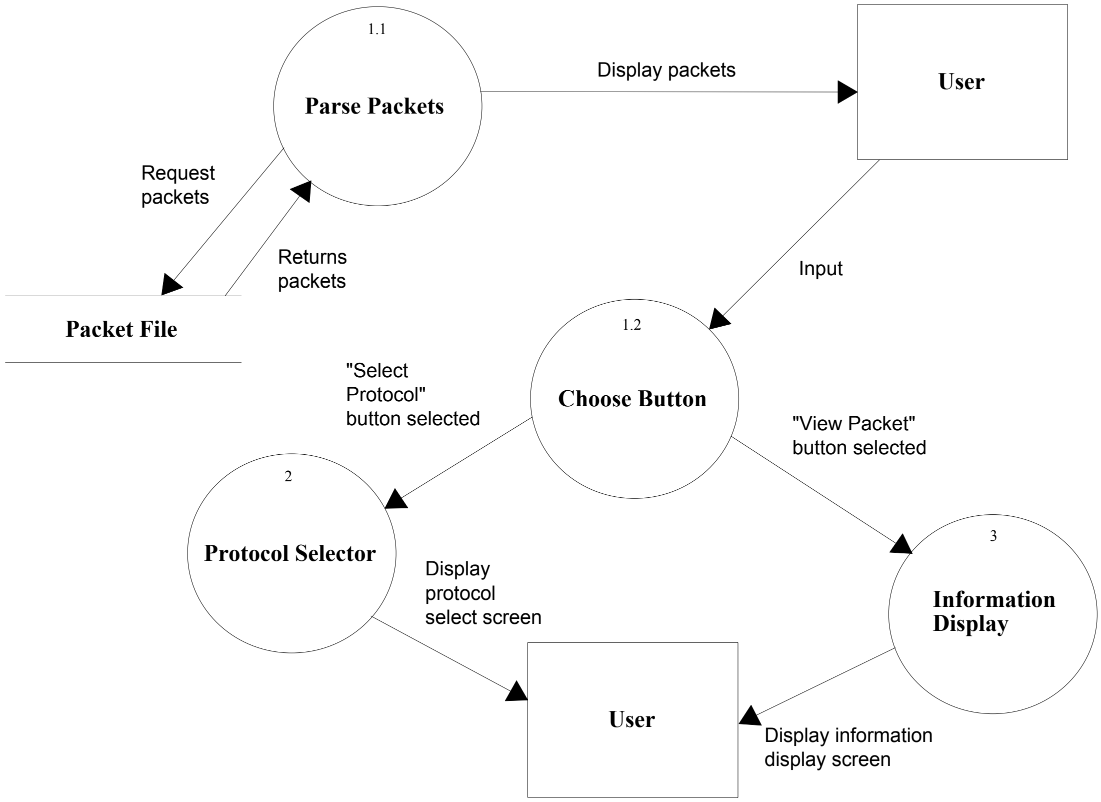


Level 0 Diagram



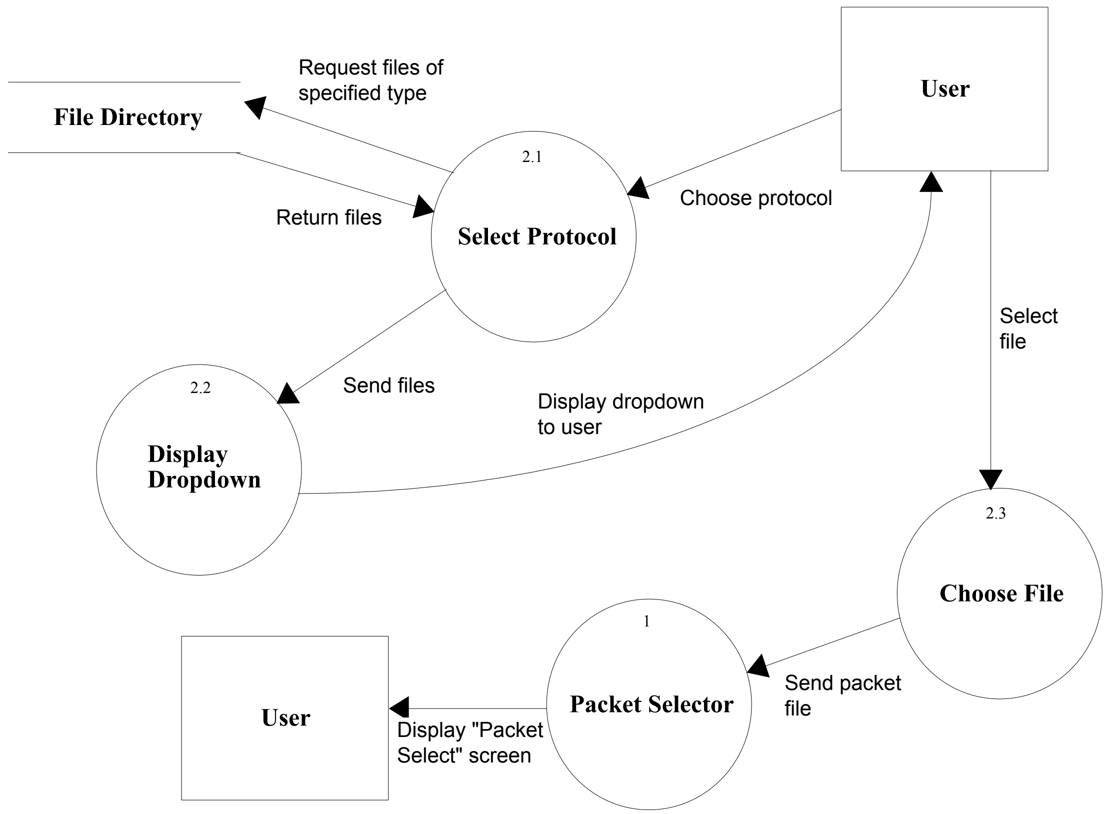
Level 1 Diagram

Packet Selector



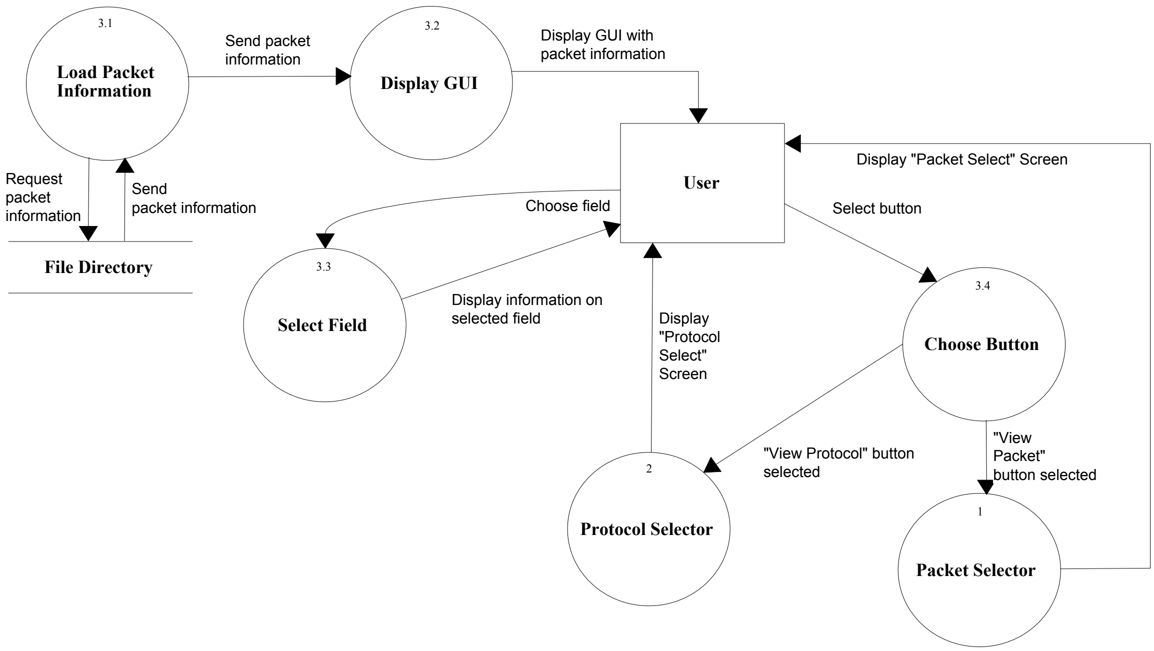
Level 2 Diagram

Protocol Selector



Level 3 Diagram

Information Display



1.3 Hardware, Software and Human Interfaces

The prototype was developed and designed using Macromedia Fireworks and Adobe Photoshop graphic design programs.

The program will be written in HTML and PHP using the Macromedia Studio MX Suite of web design tools. Other Linux based code editing tools may be used.

The TCP/IP Packet Descriptor program will be hosted as a web site on the Siena College Computer Science Department's Oraserv Linux server (Red Hat version 7.1), running the Apache web server (version 1.3.19) and PHP (version 4.1.2).

The program will be compatible with any Netscape Navigator 7.x or greater, Internet Explorer 5.x or greater, Firefox 1.x or greater, and Mozilla 1.7.x or greater web browsers.

2.0 Architectural Design Specification

2.1 User Commands

Active Protocols (those that the user can select) –

File Transfer Protocol (FTP)
Simple Mail Transfer Protocol (SMTP)
Hyper Text Transfer Protocol (HTTP)
Terminal Emulation Protocol (TELNET)
Address Resolution Protocol (ARP)
Secure Shell (SSH)
Internet Control Message Protocol (ICMP)
User Datagram Protocol (UDP)

Inactive Protocols (those that show up in the protocol hierarchy, but cannot be chosen for viewing) –

Simple Network Management Protocol (SNMP)
Secure Control Protocol (SCP)
Dynamic Host Configuration Protocol (DHCP)
Domain Name Service (DNS)
Resource Reservation Protocol (RSVP)
Lightweight Directory Access Protocol (LDAP)
Network Time Protocol (NTP)

Choose Protocol- The "Choose Protocol" function will display a hierarchical tree of available protocols. Each node of the tree will be a link to display that particular protocol.

Protocol Fields- Each field will be a link. When selected it will be highlighted and the Information of that field will be shown in an information box to the left of the PDU.

Team Logos – Each logo will be a link that, when selected, will take the user to that team's website.

View Packet – This command is given from the Packet Selector screen by pressing the View Packet button. Here, it will bring up the Information Display screen, displaying the highlighted packet.

Choose Protocol – This button in on both the Packet Selector Screen and the Information Display screen. This command will take the user back to the Protocol Selection screen.

Choose Packet – This command is executed when the user presses the Choose Packet button on the Information Display screen. This allows the user to select a different packet from within the current data session.

Packet Select – On the Packet Selector screen, there will be a display of all the packets within the current data session. The user can select the row that they wish to view by clicking on the row that contains that packet. Note that this does not bring up the Information Display screen, but rather indicates which packet will be displayed when the user selects the View Packet button.

Data Session Select – After the user clicks on their desired protocol, a drop down menu will appear displaying all the data sessions that have been stored in a folder to be specified later by our client, Mr. Swarner. The user then clicks on the data session that they would like to view. The act of selecting takes them to the Packet Selector screen.

Request for Comments Link – Each PDU will have a link to a web site with extensive information about the selected protocol.

2.2 Functional Descriptions

2.2.0 Ethernet PDU for the selected FTP PDU

Hex Dump:

00	01	03	1e	e2	24	00	00
f8	1f	00	85	08	00	45	10
00	45	AA	41	40	00	40	06
0e	85	c0	a8	00	27	c0	a8
00	65	80	30	00	15	81	a5
16	6c	87	a3	53	5d	80	18
16	d0	11	f4	00	00	01	01
08	0a	1b	25	f3	a1	0b	dd
73	58	50	41	53	53	20	66
31	61	32	6b	33	75	73	65
72	0d	0a					

Ethernet PDU > *Preamble* for the selected FTP PDU

Field Name: *Preamble*

Description: Repeating bit sequence (10101010…) used to determine clock synchronization between transmitting and receiving stations

Data value (hexadecimal): AAAAAAAAAAAAAA

Data values in other bases: ***Not applicable***

Start Bit: Pre-Packet

Length: 56

Ethernet PDU > *Start of Frame (Sof)* for the selected FTP PDU

Field Name: *Start of Frame (Sof)*

Description: Marks the beginning of the Ethernet Frame

Data value (decimal): 171

Data values in other bases:

Hexadecimal	A	B
Binary	1010	1011

Start Bit: Pre-Packet

Length: 8

Ethernet PDU > *Destination MAC Address* for the selected FTP PDU

Field Name: *Destination MAC Address*

Description: 48 bit destination node address, specifying the station(s) for which the frame is intended. It may be an individual or multicast (including broadcast) address.

The LSB of 48-bit Ethernet addresses will be 0 if the frame is single cast (meant for a specific node) or 1 if it is multicast (broadcast to multiple nodes). Individual nodes on a network determine whether they need to participate in a multicast, and either receive or ignore the frame

Data value (hexadecimal): 00 : 01 : 03 : 1E : E2 : 24

Data values in other bases: *Not applicable*

Start Bit: 0

Length: 48

Ethernet PDU > *Source MAC Address* for the selected FTP PDU

Field Name: *Source MAC Address*

Description: 48 bit source node address, specifying the station sending the frame. The Source Address field is not interpreted by the CSMA/CD MAC sublayer

Data value (hexadecimal): 00 : 00 : F8 : 1F : 00 : 85

Data values in other bases: *Not applicable*

Start Bit: 48

Length: 48

Ethernet PDU > *Length* for the selected FTP PDU

Field Name: *Length*

Description: The Length field is a 2-octet field whose value indicates the number of LLC data octets in the data field. If the value is less than the minimum required for proper operation of the protocol, a PAD field (a sequence of octets) will be added at the end of the data field but prior to the FCS field. The length field is transmitted and received with the high order octet first.

Data value (decimal): 2048

Data values in other bases:

Hexadecimal	0	8	0	0
Binary	0000	1000	0000	0000

Start Bit: 96

Length: 16

Ethernet PDU > *Cyclic Redundancy Check (CRC)* for the selected FTP PDU

Field Name: *Cyclic Redundancy Check (CRC)*

Description: Calculation to determine if any bit errors occurred to the packet during transmission

Data value (hexadecimal): FFFF

Data values in other bases: *Not applicable*

Start Bit: Post-packet

Length: 32

2.2.1 IP PDU for the selected FTP PDU

IP PDU > *IP Version* for the selected FTP PDU

Field Name: *IP Version*

Description: IP Version is a 4-bit field that indicates the format of the Internet header

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 4

IP PDU > *Header Length* for the selected FTP PDU

Field Name: *Header Length*

Description: The Header Length field is a 4-bit field indicating the length of the Internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5 (20 bytes) and the maximum value is 15 (60 bytes).

Data value (decimal): 5

Data values in other bases:

Hexadecimal	0	5
Binary	0000	0101

Start Bit: 4

Length: 4

IP PDU > *Type of Service* for the selected FTP PDU

Field Name: *Type of Service*

Description: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a data gram through a particular network.

The major choice is a three-way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay 1 = Low Delay
Bit 4: (T) 0 = Normal Throughput 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability 1 = High Reliability

Precedence:

111 = Network Control
110 = Inter-network Control
101 = CRITIC/ECP
100 = Flash Overridden
011 = Flash
010 = Immediate
001 = Priority
000 = Routine

Data value (decimal): 16

Data values in other bases:

Hexadecimal	1	0
Binary	0001	0000

Start Bit: 8

Length: 8

IP PDU > *Total Length* for the selected FTP PDU

Field Name: *Total Length*

Description: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets.

Data values (decimal): 69

Data values in other bases:

Hexadecimal	0	0	4	5
Binary	0000	0000	0100	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 69 in hexadecimal

Start Bit: 16

Length: 16

IP PDU > *Identification* for the selected FTP PDU

Field Name: *Identification*

Description: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet.

Data value (decimal): 43585

Data values in other bases:

Hexadecimal	A	A	4	1
Binary	1010	1010	0100	0001

Start Bit: 32

Length: 16

IP PDU > *Flags* for the selected FTP PDU

<u>Field Name</u>: *Flags*

<u>Description</u>: Flags is a 3-bit field that indicates directions for fragmentation.

Bit 0: reserved, must be 0
Bit 1: (DF) 0 = May Fragment 1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment 1 = More Fragment

<u>Data value (binary)</u>: 010

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 48

<u>Length</u>: 3

IP PDU > *Fragment Offset* for the selected FTP PDU

<u>Field Name</u>: *Fragment Offset*

<u>Description</u>: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

<u>Data value (decimal)</u>: 0

<u>Data values in other bases</u>:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

<u>Start Bit</u>: 51

<u>Length</u>: 13

IP PDU > *Time to Live* for the selected FTP PDU

<u>Field Name</u>: *Time to Live*

<u>Description</u>: Time to Live is an 8-bit field that indicates the maximum time the data ram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

<u>Data value (decimal)</u>: 64

<u>Data values in other bases</u>:

Hexadecimal	4	0
Binary	0100	0000

<u>Start Bit</u>: 64

<u>Length</u>: 8

IP PDU > *Protocol* for the selected FTP PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	Hex	Protocol	Dec	Hex	Protocol
0	00	Reserved	22	16	Multiplexing
1	01	ICMP	23	17	DCN
2	02	Unassigned	24	18	TAC Monitoring
3	03	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	04	CMCC Gateway Monitoring Message	77	4D	Any local network
5	05	ST	100	64	SATNET and Backroom EXPAK
6	06	TCP	101	65	MIT Subnet Support
7	07	UCL	102-104	66-68	Unassigned
10	0A	Unassigned	105	69	SATNET Monitoring
11	0B	Secure	106	6A	Unassigned
12	0C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	0D	NVP	110-113	6E-71	Unassigned
14	0E	PUP	114	72	Backroom SATNET Monitoring
15	0F	Pluribus	115	73	Unassigned
16	10	Telnet	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-0178	Unassigned
21	15	User Data gram	377	0179	Reserved

Data value (decimal): 6

Data values in other bases:

Hexadecimal	0	6
Binary	0000	0110

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the selected FTP PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. The Checksum is the 16-bit one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the initial value of its field is zero. When both header checksums are equal, then the header bits are correct. If either checksums vary, then a new, correct packet will need to be sent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 3717

Data values in other bases:

Hexadecimal	0	E	8	5
Binary	0000	1110	1000	0101

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the selected FTP PDU

Field Name: *Source IP Address*

Description: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.

Data value (decimal): 192.168.0.39

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	2	7
Binary	1100	0000	1010	1000	0000	0000	0010	0111

Start Bit: 96

Length: 32

IP PDU > *Destination IP Address* for the selected FTP PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.

Data value (decimal): 192.168.0.101

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

Start Bit: 128

Length: 32

IP PDU > *Options* for the selected FTP PDU

Field Name: *Options*

Description: The options may or may not appear in Ethernet packets. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular packet, not their implementation.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option.

Case 1: A single octet of option type

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

Some example options are:

0 End of Options list	68 Timestamp
1 No operation (pad)	131 Loose source route
7 Record route	137 Strict source route

Data values: *Not applicable*

Data values in other bases: *Not applicable*

Start Bit: 160

Length: Variable (0-40 bytes)

2.2.2 TCP PDU for the selected FTP PDU

IP > TCP PDU > *Source Port Number* for the selected FTP PDU

Field Name: *Source Port Number*

Description: A 16-bit address assigned by the sending computer that represents the name of the application that sent the data in the IP packet.

Common TCP Well-Known Server Ports (Decimal):

7	echo	110	pop3
19	chargen	111	sunrpc
20	ftp-data	119	nntp
21	ftp-control	139	netbios-ssn
22	ssh	143	imap
23	telnet	179	bgp
25	smtp	389	ldap
53	domain	443	https (ssl)
79	finger	445	microsoft-ds
80	http	1080	socks

Data value (decimal): 32816

Data values in other bases:

Hexadecimal	8	0	3	0
Binary	1000	0000	0011	0000

Start Bit: 0

Length: 16

IP > TCP PDU > *Destination Port Number* for the selected FTP PDU

Field Name: *Destination Port Number*

Description: This 16-bit number represents the name of the application that is to receive the data contained within the IP packet. This is one of the major differences between a Layer 3 and a Layer 4 header: the Layer 3 header contains the IP address of the computer that is to receive the IP packet; once that packet has been received, the port address in the Layer 4 header ensures that the data contained within that IP packet is passed to the correct application on that computer.

Common TCP Well-Known Server Ports (Decimal):

7	echo	110	pop3
19	chargen	111	sunrpc
20	ftp-data	119	nntp
21	ftp-control	139	netbios-ssn
22	ssh	143	imap
23	telnet	179	bgp
25	smtp	389	ldap
53	domain	443	https (ssl)
79	finger	445	microsoft-ds
80	http	1080	socks

This key indicates assigned port number values:

Dec	**Port Numbers**
0	Reserved
1-32767	Internet registered ("well-known") protocols
32768-98303	Reserved, to allow TCPv7-TCPv4 conversion
98304 & up	Dynamic assignment

Data value (decimal): 21 (indicates FTP)

Data values in other bases:

Hexadecimal	0	0	1	5
Binary	0000	0000	0001	0101

Start Bit: 16

Length: 16

Source: http://www.zvon.org/tmRFC/RFC1475/Output/chapter4.html

IP > TCP PDU > *Sequence Number* for the selected FTP PDU

Field Name: *Sequence Number*

Description: TCP is responsible for ensuring that all IP packets sent are actually received. When an application's data is packaged into IP packets, TCP will give each IP packet a sequence number. Once all the packets have arrived at the receiving computer, TCP uses the number in this 32-bit field to ensure that all of the packets actually arrived and are in the correct sequence.

Data value (decimal): 2175080044

Data values in other bases:

Hexadecimal	8	1	A	5	1	6	6	C
Binary	1000	0001	1010	0101	0001	0110	0110	1100

Start Bit: 32

Length: 32

IP > TCP PDU > *Acknowledgement Number* for the selected FTP PDU

Field Name: *Acknowledgement Number*

Description: This number is used by the receiving computer to acknowledge which packets have successfully arrived. This number will be the sequence number of the next packet the receiver is ready to receive.

Data value (decimal): 2275627869

Data values in other bases:

Hexadecimal	8	7	A	3	5	3	5	D
Binary	1000	0111	1010	0011	0101	0011	0101	1101

Start Bit: 64

Length: 32

IP > TCP PDU > *Length* for the selected FTP PDU

Field Name: *Length*

Description: This is identical in concept to the header length in an IP packet, except this time it indicates the length of the TCP header. The minimum value of a correct header is 5 (20 bytes) and the maximum value is 15 (60 bytes).

Data value (decimal): 8

Data values in other bases:

Hexadecimal	0	8
Binary	0000	1000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 128 in decimal

Start Bit: 96

Length: 4

IP > TCP PDU > *Reserved* for the selected FTP PDU

<u>Field Name</u>: *Reserved*

<u>Description</u>:
These 6 bits are unused and are always set to 0.

<u>Data value (decimal)</u>: 0

<u>Data values in other bases</u>:

Hexadecimal	0	0	0	0	0	0
Binary	0000	0000	0000	0000	0000	0000

<u>Start Bit</u>: 100

<u>Length</u>: 6

IP > TCP PDU > *Flags* for the selected FTP PDU

<u>Field Name</u>: *Flags*

<u>Description</u>: Every TCP packet contains this 6-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

Flags: U A P R S F
- U (1 = Urgent pointer valid)
- A (1 = Acknowledgement field value valid)
- P (1 = Push data)
- R (1 = Reset connection)
- S (1 = Synchronize sequence numbers)
- F (1 = no more data; Finish connection)}

<u>Data value (binary)</u>: 01 1000

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 106

<u>Length</u>: 6

IP > TCP PDU > *Window Size* for the selected FTP PDU

<u>Field Name</u>: *Window Size*

<u>Description</u>: Every TCP packet contains this 16-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

<u>Data value (decimal)</u>: 5840

<u>Data values in other bases</u>:

Hexadecimal	1	6	D	0
Binary	0001	0110	1101	000

<u>Start Bit</u>: 112

<u>Length</u>: 16

IP > TCP PDU > *TCP Checksum* for the selected FTP PDU

Field Name: TCP *Checksum*

Description: Unlike IP, TCP is responsible for ensuring that the entire IP packet arrived intact. TCP will run a CRC on the entire IP packet (not just the header) and place the resulting checksum in this field. When the IP packet is received, TCP re-runs the CRC on the entire packet to ensure the checksum is the same.

Data value (decimal): 4596

Data values in other bases:

Hexadecimal	1	1	F	4
Binary	0001	0001	1111	0100

Start Bit: 128

Length: 16

IP > TCP PDU > *Urgent Pointer* for the selected FTP PDU

Field Name: *Urgent Pointer*

Description: If the Urgent flag is set to on, this value indicates where the urgent data is located.

Data value (decimal): 0

Data values in other bases: *Not applicable*

Start Bit: 144

Length: 16

IP > TCP PDU > *Options* for the selected FTP PDU

<u>Field Name</u>: *Options*

<u>Description</u>: Like IP options, this field is optional and represents additional instructions not covered in the other TCP fields. Again, if an option does not fill up a 32-bit word, it will be filled in with padding bits. If present, may be used in negotiating a connection. The Options field must fit the 32-bit boundary and is padded with 0s if it does not

Some example options (Decimal):

0 End of Options list	3 Window scale
1 No operation (pad)	4 Selective ACK ok
2 Maximum segment size	8 Timestamp

<u>Data value (hexadecimal)</u>: 01 01 08 0A 1B 25 F3 A1 0B DD 73 58

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 160

<u>Length</u>: Variable

2.2.3 FTP PDU for the selected FTP PDU

IP >FTP > *Request/Response* for the selected FTP PDU

Description: Request: PASS (Password)
The argument field is a Telnet string specifying the user's password. This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control,

Data Values (hexadecimal): 50 41 53 53

Data Values in Other Bases:

ASCII	P	A	S	S
Binary	0101 0000	0100 0001	0101 0011	0101 0011

Start Bit: 144

Length: 16

RFC Link: http://www.ietf.org/rfc/rfc0959.txt?number=959

IP >FTP > *Data* for the selected FTP PDU

Description: Payload for the FTP data gram
Request Arg: f1a2k3user

Data Value (hexadecimal): 20 66 31 61 32 6B 33 75 73 65 72 0D 0A

Data Values in Other Bases:

ASCII	SPC	f	1	a	2	k	3
Binary	0010 0000	0110 0110	0011 0001	0110 0001	0011 0010	0110 1011	0011 0011

ASCII	u	s	e	r	\r	\n
Binary	0111 0101	0111 0011	0110 0101	0111 0010	0000 1101	0000 1010

Start Bit: 144

Length: 16

2.2.4 IP PDU for the selected ICMP PDU

IP PDU > *IP Version* for the selected ICMP PDU

Field Name: *IP Version*

Description: Version is a 4-bit field that indicates the format of the Internet header.

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 5

IP PDU > *Header Length* for the selected ICMP PDU

Field Name: *Header Length*

Description: The IHL field is a 4 bit field indicating the length of the Internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5 (20 bytes) and the maximum value is 15 (60 bytes).

Data value (decimal): 5

Data values in other bases:

Hexadecimal	0	5
Binary	0000	0101

Start Bit: 4

Length: 4

IP PDU > *Type of Service* for the selected ICMP PDU

Field Name: *Type of Service*

Description: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a data gram through a particular network.

The major choice is a three-way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay 1 = Low Delay

Bit 4: (T) 0 = Normal Throughput 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability 1 = High Reliability

Precedence:

111 = Network Control	011 = Flash
110 = Internetwork Control	010 = Immediate
101 = CRITIC/ECP	001 = Priority
100 = Flash Overridden	000 = Routine

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0
Binary	0000	0000

Start Bit: 8

Length: 8

IP PDU > *Total Length* for the selected ICMP PDU

Field Name: *Total Length*

Description: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets.

Data values (decimal): 84

Data values in other bases:

Hexadecimal	0	0	5	4
Binary	0000	0000	0101	0100

Start Bit: 16

Length: 16

IP PDU > *Identification* for the selected ICMP PDU

Field Name: *Identification*

Description: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet

Data value (decimal): 21929

Data values in other bases:

Hexadecimal	5	5	A	9
Binary	0101	0101	1010	1001

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary data value was 0000 0000 0000 0000

Start Bit: 32

Length: 16

IP PDU > *Flags* for the selected ICMP PDU

<u>Field Name</u>: *Flags*

<u>Description</u>: Flags is a 3-bit field that indicates directions for fragmentation.

Bit 0: reserved, must be 0
Bit 1: (DF) 0 = May Fragment 1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment 1 = More Fragment

<u>Data value (binary)</u>: 000

<u>Data values in other bases</u>: *Not applicable*

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary data value was 010

<u>Start Bit</u>: 48

<u>Length</u>: 3

IP PDU > ***Fragment Offset*** for the selected ICMP PDU

Field Name: *Fragment Offset*

Description: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

Start Bit: 51

Length: 13

IP PDU > *Time to Live* for the selected ICMP PDU

Field Name: *Time to Live*

Description: Time to Live is an 8-bit field that indicates the maximum time the data gram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

Data value (decimal): 255

Data values in other bases:

Hexadecimal	F	F
Binary	1111	1111

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 64 in decimal

Start Bit: 64

Length: 8

IP PDU > *Protocol* for the selected ICMP PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	Hex	Protocol	Dec	Hex	Protocol
0	00	Reserved	22	16	Multiplexing
1	01	ICMP	23	17	DCN
2	02	Unassigned	24	18	TAC Monitoring
3	03	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	04	CMCC Gateway Monitoring Message	77	4D	Any local network
5	05	ST	100	64	SATNET and Backroom EXPAK
6	06	TCP	101	65	MIT Subnet Support
7	07	UCL	102-104	66-68	Unassigned
10	0A	Unassigned	105	69	SATNET Monitoring
11	0B	Secure	106	6A	Unassigned
12	0C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	0D	NVP	110-113	6E-71	Unassigned
14	0E	PUP	114	72	Backroom SATNET Monitoring
15	0F	Pluribus	115	73	Unassigned
16	10	Telnet	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-0178	Unassigned
21	15	User Data gram	377	0179	Reserved

Data value (decimal): 1

Data values in other bases:

Hexadecimal	0	1
Binary	0000	0001

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the Selected ICMP PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. This CRC algorithm is the 16-bit one's complement sum of all the 16-bit words in the header. For purposes of computing the checksum, the value of the checksum field is initially zero. When both header checksums are the same, then the header bits are correct. If either checksums vary, then a packet will need to be resent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 58402

Data values in other bases:

Hexadecimal	E	4	2	2
Binary	1110	0100	0010	0010

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was B8 CC in hexadecimal

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the Selected ICMP PDU

Field Name: *Source IP Address*

Description: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.

Data value: 192.168.0.101

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 192.168.0.39

Start Bit: 96

Length: 32

IP PDU > *Destination IP Address* for the selected ICMP PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.

Data value: 192.168.0.39

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	2	7
Binary	1100	0000	1010	1000	0000	0000	0010	0111

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 192.168.0.101

Start Bit: 128

Length: 32

IP PDU > *Options* for the selected ICMP PDU

Field Name: *Options*

Description: The options may or may not appear in Ethernet packets. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular packet, not their implementation.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option.

Case 1: A single octet of option type

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

Some example options are:

0 End of Options list	68 Timestamp
1 No operation (pad)	131 Loose source route
7 Record route	137 Strict source route

Data values: *Not applicable*

Data values in other bases: *Not applicable*

Start Bit: 160

Length: Variable (0-40 bytes)

2.2.5 ICMP PDU for the selected ICMP PDU

IP > ICMP Header > *Type* for the selected ICMP PDU

<u>Field Name</u>: *Type*

<u>Description</u>: The type is an 8-bit field that identifies what sort of message the ICMP protocol is sending.

ICMP message types are:

0 Echo Reply
3 Destination Unreachable
4 Source Quench
5 Redirect
8 Echo
11 Time Exceeded
12 Parameter Problem
13 Timestamp
14 Timestamp Reply
15 Information Request
16 Information Reply

Dec	Hex	Message Type	Dec	Hex	Message Type
0	00	Echo Reply	16	10	Information Reply
1	01	Unassigned	17	11	Address Mask Request
2	02	Unassigned	18	12	Address Mask Reply
3	03	Destination Unreachable	19	13	Reserved (for Security)
4	04	Source Quench	20-29	14-1D	Reserved (for Robustness Experiment)
5	05	Redirect	30	1E	Traceroute
6	06	Alternate Host Address	31	1F	Data gram Conversion Error
7	07	Unassigned	32	20	Mobile Host Redirect
8	08	Echo	33	21	IPv6 Where-Are-You
9	09	Router Advertisement	34	22	IPv6 I-Am-Here
10	0A	Router Solicitation	35	23	Mobile Registration Request
11	0B	Time Exceeded	36	24	Mobile Registration Reply
12	0C	Parameter Problem	37	25	Domain Name Request
13	0D	Timestamp	38	26	Domain Name Reply
14	0E	Timestamp Reply	39	27	SKIP
15	0F	Information Request	40	28	Photuris
			41-255	29-FF	Reserved

<u>Data value</u>: 0

Data values in other bases:

Hexadecimal	0	0
Binary	0000	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 8 in decimal

Start Bit: 0

Length: 8

RFC Link: http://www.iana.org/assignments/icmp-parameters

IP > ICMP Header > *Code* for the selected ICMP PDU

<u>Field Name</u>: *Code*

<u>Description</u>: Code is an 8-bit field that provides further information about the associated type field. Based on what type the ICMP PDU is, the code can have a number of meanings.

For example, with ICMP type 3 (Destination Unreachable) is as follows:
0 = net unreachable;
1 = host unreachable;
2 = protocol unreachable;
3 = port unreachable;
4 = fragmentation needed and DF set;
5 = source route failed.

Type	Code	Name
0		Echo Reply (used by "PING")
	0	No Code
1		Unassigned
2		Unassigned
3		Destination Unreachable
	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Fragmentation needed and Don't Fragment was Set
	5	Source Route Failed
	6	Destination Network Unknown
	7	Destination Host Unknown
	8	Source Host Isolated
	9	Communication with Destination Network is Administratively Prohibited
	10	Communication with Destination Host is Administratively Prohibited
	11	Destination Network Unreachable for Type of Service
	12	Destination Host Unreachable for Type of Service
4		Source Quench
	0	No Code
5		Redirect
	0	Redirect Data gram for the Network
	1	Redirect Data gram for the Host
	2	Redirect Data gram for the Type of Service and Network
	3	Redirect Data gram for the Type of Service and Host
6		Alternate Host Address
	0	Alternate Address for Host
7		Unassigned
8		Echo (used by "PING")
	0	No Code
9		Router Advertisement
	0	No Code
10		Router Selection
	0	No Code
11		Time Exceeded
	0	Time to Live exceeded in Transit
	1	Fragment Reassembly Time Exceeded
12		Parameter Problem
	0	Pointer indicates the error
	1	Missing a Required Option
	2	Bad Length
13		Timestamp
	0	No Code
14		Timestamp Reply
	0	No Code
15		Information Request
	0	No Code
16		Information Reply
	0	No Code
17		Address Mask Request
	0	No Code
18		Address Mask Reply
	0	No Code
19		Reserved (for Security)
20-29		Reserved (for Robustness Experiment)
30		Trace route
31		Data gram Conversion Error
32		Mobile Host Redirect
33		IPv6 Where-Are-You
34		IPv6 I-Am-Here
35		Mobile Registration Request
36		Mobile Registration Reply

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0
Binary	0000	0000

Start Bit: 8

Length: 8

IP > ICMP Header > *ICMP Checksum* for the selected ICMP PDU

Field Name: *ICMP Checksum*

Description: The checksum is the 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP type. For computing the checksum, the checksum field should initially be zero.

Data value (decimal): 17173

Data values in other bases:

Hexadecimal	4	3	1	5
Binary	0100	0011	0001	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was C9 15 in hexadecimal

Start Bit: 16

Length: 16

IP > ICMP Header > *Identifier* for the selected ICMP PDU

Field Name: Identifier

Description: The identifier is a 16-bit field that is used in matching echoes and replies for when the code field is zero.

Data value (decimal): 28768

Data values in other bases:

Hexadecimal	7	0	6	0
Binary	0111	0000	0110	0000

Start Bit: 32

Length: 16

IP > ICMP Header > *Sequence Number* for the selected ICMP PDU

Field Name: *Sequence Number*

Description: The sequence is a 16-bit field that is used in matching echoes and replies for when the code field is zero.

Data value (decimal): 256

Data values in other bases:

Hexadecimal	0	1	0	0
Binary	000	0001	0000	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 70 60 in hexadecimal

Start Bit: 48

Length: 16

IP > ICMP Header > *Data* for the selected ICMP PDU

Field Name: *Data*

Description: The data is a variable-length field that contains the actual information that is sent in the ping packet.

Data value (hexadecimal): 43 B1 89 3F 00 00 00 00 B8 C6 07 00 00 00 00 00 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37

Data values in other bases: *Not applicable*

Start Bit: 64

Length: Variable

2.2.6 IP PDU for the selected SMTP PDU

IP PDU > *IP Version* for the selected SMTP PDU

Field Name: *IP Version*

Description: Version is a 4-bit field that indicates the format of the Internet header

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 4

IP PDU > *Header Length* for the selected SMTP PDU

Field Name: *Header Length*

Description: The IHL field is a 4-bit field indicating the length of the Internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5 (20 bytes) and the maximum value is 15 (60 bytes).

Data value: 5

Data values in other bases:

Hexadecimal	0	5
Binary	0000	0101

Start Bit: 4

Length: 4

IP PDU > *Type of Service* for the selected SMTP PDU

Field Name: *Type of Service*

Description: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a data gram through a particular network.

The major choice is a three-way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay — 1 = Low Delay
Bit 4: (T) 0 = Normal Throughput — 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability — 1 = High Reliability

Precedence:

111 = Network Control	011 = Flash
110 = Internetwork Control	010 = Immediate
101 = CRITIC/ECP	001 = Priority
100 = Flash Overridden	000 = Routine

Data value (decimal): 16

Data values in other bases:

Hexadecimal	1	0
Binary	0001	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 0 in decimal

Start Bit: 8

Length: 8

IP PDU > *Total Length* for the selected SMTP PDU

Field Name: *Total Length*

Description: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets

Data values (decimal): 70

Data values in other bases:

Hexadecimal	0	0	4	6
Binary	0000	0010	0100	0110

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 02 12 in hexadecimal

Start Bit: 16

Length: 16

IP PDU > *Identification* for the selected SMTP PDU

Field Name: *Identification*

Description: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet.

Data value (decimal): 46047

Data values in other bases:

Hexadecimal	B	3	D	F
Binary	1011	0011	1101	1111

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 61 28 in hexadecimal

Start Bit: 32

Length: 16

IP PDU > *Flags* for the selected SMTP PDU

Field Name: *Flags*

Description: Flags is a 3-bit field that indicates directions for fragmentation.

Bit 0: reserved, must be 0	
Bit 1: (DF) 0 = May Fragment	1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment	1 = More Fragment

Data value (binary): 010

Data values in other bases: *Not applicable*

Start Bit: 48

Length: 3

IP PDU > *Fragment Offset* for the selected SMTP PDU

Field Name: *Fragment Offset*

Description: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

Start Bit: 51

Length: 13

IP PDU > *Time to Live* for the selected SMTP PDU

Field Name: *Time to Live*

Description: Time to Live is an 8-bit field that indicates the maximum time the data gram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

Data value (decimal): 128

Data values in other bases:

Hexadecimal	8	0
Binary	1000	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 64 in decimal

Start Bit: 64

Length: 8

IP PDU > *Protocol* for the selected SMTP PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	**Hex**	**Protocol**	**Dec**	**Hex**	**Protocol**
0	00	Reserved	22	16	Multiplexing
1	01	ICMP	23	17	DCN
2	02	Unassigned	24	18	TAC Monitoring
3	03	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	04	CMCC Gateway Monitoring Message	77	4D	Any local network
5	05	ST	100	64	SATNET and Backroom EXPAK
6	06	TCP	101	65	MIT Subnet Support
7	07	UCL	102-104	66-68	Unassigned
10	0A	Unassigned	105	69	SATNET Monitoring
11	0B	Secure	106	6A	Unassigned
12	0C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	0D	NVP	110-113	6E-71	Unassigned
14	0E	PUP	114	72	Backroom SATNET Monitoring
15	0F	Pluribus	115	73	Unassigned
16	10	Telnet	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-0178	Unassigned
21	15	User Data gram	377	0179	Reserved

Data value (decimal): 6

Data values in other bases:

Hexadecimal	0	6
Binary	0000	0110

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the selected SMTP PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. The Checksum is the 16-bit one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the initial value of its field is zero. When both header checksums are equal, then the header bits are correct. If either checksums vary, then a new, correct packet will need to be sent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 24840

Data values in other bases:

Hexadecimal	6	1	0	8
Binary	0110	0001	0000	1000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was F1 F3 in hexadecimal

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the selected SMTP PDU

Field Name: *Source IP Address*

Description: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.

Data value (decimal): 192.168.100.20

Data values in other bases:

Hexadecimal	C	0	A	8	6	4	1	4
Binary	1100	0000	1010	1000	0110	0100	0001	0100

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 192.168.0.101 in decimal

Start Bit: 96

Length: 32

IP PDU > *Destination IP Address* for the selected SMTP PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.

Data value (decimal): 192.168.0.101

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 192.168.100.20 in decimal

Start Bit: 128

Length: 32

IP PDU > *Options* for the selected SMTP PDU

<u>Field Name</u>: *Options*

<u>Description</u>: The options may or may not appear in Ethernet packets. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular packet, not their implementation.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option.

Case 1: A single octet of option type

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

Some example options are:

0 End of Options list	68 Timestamp
1 No operation (pad)	131 Loose source route
7 Record route	137 Strict source route

<u>Data values</u>: *Not applicable*

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 160

<u>Length</u>: Variable (0-40 bytes)

2.2.7 TCP PDU for the selected SMTP PDU

IP > TCP PDU > *Source Port Number* for the selected SMTP PDU

<u>Field Name</u>: *Source Port Number*

<u>Description</u>: A 16-bit address assigned by the sending computer, to the application program sending data as TCP data grams.

Common TCP Well-Known Server Ports (Decimal):

07 echo	110 pop3
19 chargen	111 sunrpc
20 ftp-data	119 nntp
21 ftp-control	139 netbios-ssn
22 ssh	143 imap
23 telnet	179 bgp
25 smtp	389 idap
53 domain	443 https(ssl)
79 finger	445 microsoft-ds
80 http	1080 socks

<u>Data value (decimal)</u>: 32816

<u>Data values in other bases</u>:

Hexadecimal	8	0	3	0
Binary	1000	0000	0011	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 3651 in decimal

<u>Start Bit</u>: 0

<u>Length</u>: 16

IP > TCP PDU > *Destination Port Number* for the selected SMTP PDU

Field Name: *Destination Port Number*

Description: A 16-bit address assigned by the receiving computer, to the destination application program for this message.

Common TCP Well-Known Server Ports (Decimal):

07 echo	110 pop3
19 chargen	111 sunrpc
20 ftp-data	119 nntp
21 ftp-control	139 netbios-ssn
22 ssh	143 imap
23 telnet	179 bgp
25 smtp	389 idap
53 domain	443 https (ssl)
79 finger	445 microsoft-ds
80 http	1080 socks

Data value (decimal): 21 (indicates SMTP)

Data values in other bases:

Hexadecimal	0	0	1	5
Binary	0000	0000	0001	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 25 in decimal

Start Bit: 16

Length: 16

IP > TCP PDU > *Sequence Number* for the selected SMTP PDU

Field Name: *Sequence Number*

Description: TCP is responsible for ensuring that all IP packets sent are actually received. When an application's data is packaged into IP packets, TCP will give each IP packet a sequence number. Once all the packets have arrived at the receiving computer, TCP uses the number in this 32-bit field to ensure that all of the packets actually arrived and are in the correct sequence.

Data value (decimal): 2175080044

Data values in other bases:

Hexadecimal	8	1	A	5	1	6	6	C
Binary	1000	0001	1010	0101	0001	0110	0110	1100

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 2069207327 in decimal

Start Bit: 32

Length: 32

IP > TCP PDU > *Acknowledgement Number* for the selected SMTP PDU

Field Name: *Acknowledgement Number*

Description: This number is used by the receiving computer to acknowledge which packets have successfully arrived. This number will be the sequence number of the next packet the receiver is ready to receive.

Data value (decimal): 2275627869

Data values in other bases:

Hexadecimal	8	7	A	3	5	3	5	D
Binary	1000	0111	1010	0011	0101	0011	0101	1101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 3827794966 in decimal

Start Bit: 64

Length: 32

IP > TCP PDU > *Length* for the selected SMTP PDU

Field Name: *Length*

Description: The Header Length field is a 4-bit field indicating the length of the TCP header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5 (20 bytes) and the maximum value is 15 (60 bytes).

Data value (decimal): 8

Data values in other bases:

Hexadecimal	0	8
Binary	0000	1000

- The data value used here is that of Blue Technology’s data. Blue’s data matches up with the dump file from ethereal while Mirage’s doesn’t.
- Mirage’s data value was 128 in decimal

Start Bit: 96

Length: 4

IP > TCP PDU > ***Reserved*** for the selected SMTP PDU

Field Name: *Reserved*

Description: These 6 bits are unused and are always set to 0.

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0	0	0	0	0
Binary	0000	0000	0000	0000	0000	0000

Start Bit: 100

Length: 6

IP > TCP PDU > *Control Flags* for the selected SMTP PDU

Field Name: *Control Flags*

Description: Every TCP packet contains this 6-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

Flags: U A P R S F
- U (1 = Urgent pointer valid)
- A (1 = Acknowledgement field value valid)
- P (1 = Push data)
- R (1 = Reset connection)
- S (1 = Synchronize sequence numbers)
- F (1 = no more data; Finish connection)

Data value (binary): 01 1000

Data values in other bases: *Not applicable*

Start Bit: 106

Length: 6

IP > TCP PDU > *Window Size* for the selected SMTP PDU

<u>Field Name</u>: *Window Size*

<u>Description</u>: Every TCP packet contains this 16-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

<u>Data value (decimal)</u>: 5840

<u>Data values in other bases</u>:

Hexadecimal	1	6	D	0
Binary	0001	0110	1101	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 32120 in decimal

<u>Start Bit</u>: 112

<u>Length</u>: 16

IP > TCP PDU > *TCP Checksum* for the selected SMTP PDU

Field Name: *TCP Checksum*

Description: Unlike IP, TCP is responsible for ensuring that the entire IP packet arrived intact. TCP will run a CRC on the entire IP packet (not just the header) and place the resulting checksum in this field. When the IP packet is received, TCP re-runs the CRC on the entire packet to ensure the checksum is the same.

Data value (decimal): 4596

Data values in other bases:

Hexadecimal	1	1	F	4
Binary	0001	0001	1111	0100

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 72 B5 in hexadecimal

Start Bit: 128

Length: 16

IP > TCP PDU > *Urgent Pointer* for the selected SMTP PDU

Field Name: *Urgent Pointer*

Description: If the Urgent flag is set to on, this value indicates where the urgent data is located.

Data value: *Not applicable*

Data values in other bases: *Not applicable*

Start Bit: 144

Length: 16

IP > TCP PDU > *Options* for the selected SMTP PDU

Field Name: *Options*

Description: Like IP options, this field is optional and represents additional instructions not covered in the other TCP fields. If present may be used in negotiating a connection. The options field must fit the 32 bit boundary and is padded with 0's if it does not.

Some example options (In decimal):
0 - End of options list
1 - No operation (pad)
2 - Maximum segment size
3 - Window scale
4 - Selective Acknowledgement OK
8 - Timestamp

Data value (hexadecimal): 01 01 08 0A 1B 25 F3 A1 0B DD 73 58

Data values in other bases: *Not applicable*

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 01 01 08 0A 07 AE F6 75 00 21 66 A4 in hexadecimal

Start Bit: 160

Length: Variable

2.2.8 SMTP PDU for the selected SMTP PDU

IP > SMTP Header > *Data* for the selected SMTP PDU

Field Name: *Data*

Description: ASCII messages sent between SMTP hosts.

Command	Description
DATA	Begins message composition.
EXPN <string>	Returns names on the specified mail list.
HELO <domain>	Returns identity of mail server.
HELP <command>	Returns information on the specified command.
MAIL FROM <host>	Initiates a mail session from host.
NOOP	Causes no action, except acknowledgement from server.
QUIT	Terminates the mail session.
RCPT TO <user>	Designates who receives mail.
RSET	Resets mail connection.
SAML FROM <host>	Sends mail to user terminal and mailbox.
SEND FROM <host>	Sends mail to user terminal.
SOML FROM <host>	Sends mail to user terminal or mailbox.
TURN	Switches role of receiver and sender.
VRFY <user>	Verifies the identity of a user.

Data value: Content_TEXT\Plain;name="mimetest.txt"

Data values in other bases:

Hexadecimal	4	3	6	F	6	E	7	4
Binary	0100	0011	0110	1111	0110	1110	0111	0100

Hexadecimal	6	5	6	E	7	4	2	D
Binary	0110	0101	0110	1110	0111	0100	0010	1101

Hexadecimal	5	4	4	5	5	8	5	4
Binary	0101	0100	0100	0101	0101	1000	0101	0100

Hexadecimal	2	F	5	0	6	C	6	1
Binary	0010	1111	0101	0000	0110	1100	0110	0001

Hexadecimal	6	9	6	E	3	B	6	9
Binary	0110	1001	0110	1110	0011	1011	0110	1001

Hexadecimal	6	1	6	D	6	5	3	D
Binary	0110	0001	0110	1101	0110	0101	0011	1101

Hexadecimal	2	0	6	3	6	8	6	1
Binary	0010	0000	0110	0011	0110	1000	0110	0001

Hexadecimal	2	2	7	4	6	5	7	3
Binary	0010	0010	0111	0100	0110	0101	0111	0011

Hexadecimal	7	4	2	E	7	4	7	8
Binary	0111	0100	0010	1110	0111	0100	0111	1000

Hexadecimal	7	4	2	0
Binary	0111	0100	0010	0000

Start Bit: 0

Length: 152

RFC Link: http://www.ietf.org/rfc/rfc0821.txt?number=821

IP > SMTP Header > *Message* for the selected SMTP PDU

Field Name: *Message*

Description: Response messages consist of a response code followed by explanatory text

Response Code	Explanatory Text
211	(Response to system status or help request).
214	(Response to help request).
220	Mail service ready.
221	Mail service closing connection.
250	Mail transfer completed.
251	User not local, forward to <path>.
354	Start mail message, end with <CRLF><CRLF>.
421	Mail service unavailable.
450	Mailbox unavailable.
451	Local error in processing command.
452	Insufficient system storage.
500	Unknown command.
501	Bad parameter.
502	Command not implemented.
503	Bad command sequence.
504	Parameter not implemented.
550	Mailbox not found.
551	User not local, try <path>.
552	Storage allocation exceeded.
553	Mailbox name not allowed.
554	Mail transaction failed.

Data value: *Not applicable.*

Start Bit: 152

Length: Variable

2.2.9 IP PDU for the selected HTTP PDU

IP PDU > *IP Version* for the selected HTTP PDU

Field Name: *IP Version*

Description: Version is a 4-bit field that indicates the format of the Internet header.

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 4

IP PDU> *Header Length* for the selected HTTP PDU

Field Name: *Header Length*

Description: The IHL field is a 4-bit field indicating the length of the Internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5.

Data value: The value contained in our field is 20 bytes. This is the hexadecimal and decimal value of 5 multiplied by 4 bits.

Data values in other bases:

Hexadecimal	0	5
Binary	0000	0101

Start Bit: 4

Length: 4

IP PDU > *Type of Service* for the selected HTTP PDU

Field Name: *Type of Service*

Description: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a data gram through a particular network.

The major choice is a three-way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay 1 = Low Delay
Bit 4: (T) 0 = Normal Throughput 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability 1 = High Reliability

Precedence:

111 = Network Control
110 = Internetwork Control
101 = CRITIC/ECP
100 = Flash Overrided
011 = Flash
010 = Immediate
001 = Priority
000 = Routine

Data value (decimal): 16

Data values in other bases:

Hexadecimal	1	0
Binary	0001	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 0 in decimal.

Start Bit: 8

Length: 8

IP PDU > *Total Length* for the selected HTTP PDU

Field Name: *Total Length*

Description: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets.

Data values (decimal): 69

Data values in other bases:

Hexadecimal	0	0	4	5
Binary	0000	0000	0100	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 570 in decimal.

Start Bit: 16

Length: 16

IP PDU > *Identification* for the selected HTTP PDU

<u>Field Name</u>: *Identification*

<u>Description</u>: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet.

<u>Data value (decimal)</u>: 43585

<u>Data values in other bases</u>:

Hexadecimal	A	A	4	1
Binary	1010	1010	0100	0001

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 15365 in decimal.

<u>Start Bit</u>: 32

<u>Length</u>: 16

IP PDU > *Flags* for the selected HTTP PDU

Field Name: *Flags*

Description: Flags is a 3-bit field that indicates directions for fragmentation.

Bit 0: reserved, must be 0
Bit 1: (DF) 0 = May Fragment 1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment 1 = More Fragment

Data value (binary): 010

Data values in other bases: *Not applicable*

Start Bit: 48

Length: 3

IP PDU > *Fragment Offset* for the selected HTTP PDU

Field Name: *Fragment Offset*

Description: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

Start Bit: 51

Length: 13

IP PDU > *Time to Live* for the selected HTTP PDU

Field Name: *Time to Live*

Description: Time to Live is an 8-bit field that indicates the maximum time the data gram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

Data value (decimal): 64

Data values in other bases:

Hexadecimal	4	0
Binary	0100	0000

Start Bit: 64

Length: 8

IP PDU > *Protocol* for the selected HTTP PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	Hex	Protocol	Dec	Hex	Protocol
0	0	Reserved	22	16	Multiplexing
1	1	ICMP	23	17	DCN
2	2	Unassigned	24	18	TAC Monitoring
3	3	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	4	CMCC Gateway Monitoring Message	77	4D	Any local network
5	5	ST	100	64	SATNET and Backroom EXPAK
6	6	TCP	101	65	MIT Subnet Support
7	7	UCL	102-104	66-68	Unassigned
10	A	Unassigned	105	69	SATNET Monitoring
11	B	Secure	106	6A	Unassigned
12	C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	D	NVP	110-113	6E-71	Unassigned
14	E	PUP	114	72	Backroom SATNET Monitoring
15	F	Pluribus	115	73	Unassigned
16	10	Telnet	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-178	Unassigned
21	15	User Datagram	377	179	Reserved

Data value (decimal): 6

Data values in other bases:

Hexadecimal	0	6
Binary	0000	0110

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the selected HTTP PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. The Checksum is the 16-bit one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the initial value of its field is zero. When both header checksums are equal, then the header bits are correct. If either checksums vary, then a new, correct packet will need to be sent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 3717

Data values in other bases:

Hexadecimal	0	E	8	5
Binary	0000	1110	1000	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 31319 in decimal.

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the selected HTTP PDU

Field Name: *Source IP Address*

Description: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.

Data value (decimal): 192.168.0.39

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	2	7
Binary	1100	0000	1010	1000	0000	0000	0010	0111

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 192.168.0.012 in decimal.

Start Bit: 96

Length: 32

IP PDU > *Destination IP Address* for the selected HTTP PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.

Data value (decimal): 192.168.0.101

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

Start Bit: 128

Length: 32

IP PDU > *Options* for the selected HTTP PDU

Field Name: *Options*

Description: The options may or may not appear in Ethernet packets. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular packet, not their implementation.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option.

Case 1: A single octet of option type

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

Data values: *Not applicable*

Data values in other bases: *Not applicable*

Start Bit: 160

Length: 0-40

IP PDU > *Data* for the selected HTTP PDU

Field Name: *Data*

Description: The Data is a variable length field, which contains the actual data that is being sent from one host to another. The data field may start with a Layer 4 header, which will give additional instructions to the application that will be receiving the data; alternately, it may be an ICMP header and not contain any user data at all.

Data values (hexadecimal): (TCP) 80 30 00 15 81 A5 16 6C 87 A3 53 5D 80 18 16 D0 11 F4 00 00 01 01 08 0A 1B 25 F3 A1 0b DD 73 58
(FTP) 50 41 53 53 20 66 31 61 32 6B 33 75 73 65 72 0D 0A

Data values in other bases: ***Not applicable***

2.2.10 TCP PDU for the selected HTTP PDU

IP > TCP PDU > *Source Port Number* for the selected HTTP PDU

Field Name: *Source Port Number*

Description: This 16-bit number represents the name of the application that sent the data in the IP packet.

Data value (decimal): 32816

Data values in other bases:

Hexadecimal	8	0	3	0
Binary	1000	0000	0011	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 50 in decimal.

Start Bit: 0

Length: 16

IP > TCP PDU > *Destination Port Number* for the selected HTTP PDU

Field Name: *Destination Port Number*

Description: This 16-bit number represents the name of the application that is to receive the data contained within the IP packet. This is one of the major differences between a Layer 3 and a Layer 4 header: the Layer 3 header contains the IP address of the computer that is to receive the IP packet; once that packet has been received, the port address in the Layer 4 header ensures that the data contained within that IP packet is passed to the correct application on that computer.

This key indicates assigned port number values:

Dec	Port Numbers
0	Reserved
1-32767	Internet registered ("well-known") protocols
32768-98303	Reserved, to allow TCPv7-TCPv4 conversion
98304 & up	Dynamic assignment

Data value (decimal): 21

Data values in other bases:

Hexadecimal	0	0	1	5
Binary	0000	0000	0001	0101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 4255 in decimal.

Start Bit: 16

Length: 16

Source: http://www.zvon.org/tmRFC/RFC1475/Output/chapter4.html

IP > TCP PDU > *Sequence Number* for the selected HTTP PDU

Field Name: *Sequence Number*

Description: TCP is responsible for ensuring that all IP packets sent are actually received. When an application's data is packaged into IP packets, TCP will give each IP packet a sequence number. Once all the packets have arrived at the receiving computer, TCP uses the number in this 32-bit field to ensure that all of the packets actually arrived and are in the correct sequence.

Data value (decimal): 2175080044

Data values in other bases:

Hexadecimal	8	1	A	5	1	6	6	C
Binary	1000	0001	1010	0101	0001	0110	0110	1100

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 988014608 in decimal.

Start Bit: 32

Length: 32

IP > TCP PDU > *Acknowledgement Number* for the selected HTTP PDU

Field Name: *Acknowledgement Number*

Description: This number is used by the receiving computer to acknowledge which packets have successfully arrived. This number will be the sequence number of the next packet the receiver is ready to receive.

Data value (decimal): 2275627869

Data values in other bases:

Hexadecimal	8	7	A	3	5	3	5	D
Binary	1000	0111	1010	0011	0101	0011	0101	1101

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 1398299764 in decimal.

Start Bit: 64

Length: 32

IP > TCP PDU > *Length* for the selected HTTP PDU

Field Name: *Length*

Description: This is identical in concept to the header length in an IP packet, except this time it indicates the length of the TCP header.

Data value (decimal): 8

Data values in other bases:

Hexadecimal	0	8
Binary	0000	1000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 128 in decimal.

Start Bit: 96

Length: 4

IP > TCP PDU > *Reserved* for the selected HTTP PDU

<u>Field Name</u>: *Reserved*

<u>Description</u>: 4 bits; set to 0
ECN bits (used when ECN employed; else 00)
CWR (1=sender has cut congestion window in half
ECN-Echo (1=receiver has cut congestion window in half

<u>Data value (decimal)</u>: 0

<u>Data values in other bases</u>:

Hexadecimal	0	0
Binary	0000	0000

<u>Start Bit</u>: 100

<u>Length</u>: 6 Bits

IP > TCP PDU > *Control Flags* for the selected HTTP PDU

<u>Field Name</u>: *Control Flags*

<u>Description</u>: Every TCP packet contains this 6-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

- Urgent (URG)
- Acknowledgement (ACK)
- Push (PSH)
- Reset (RST)
- Synchronize (SYN)
- Finish (FIN)

<u>Data value (binary)</u>: 01 1000

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 106

<u>Length</u>: 6

IP > TCP PDU > *Window Size* for the selected HTTP PDU

Field Name: *Window Size*

Description: Every TCP packet contains this 16-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

Data value (decimal): 5840

Data values in other bases:

Hexadecimal	1	6	D	0
Binary	0001	0110	1101	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 7504 in decimal.

Start Bit: 112

Length: 16

IP > TCP PDU > *Urgent Pointer* for the selected HTTP PDU

<u>Field Name</u>: *Urgent Pointer*

<u>Description</u>: If the Urgent flag is set to on, this value indicates where the urgent data is located.

<u>Information Key</u>: *Not applicable*

<u>Data value</u>: *Not applicable*

<u>Data values in other bases</u>: *Not applicable*

IP > TCP PDU > *TCP* *Checksum* for the selected HTTP PDU

Field Name: *TCP Checksum*

Description: Unlike IP, TCP is responsible for ensuring that the entire IP packet arrived intact. TCP will run a CRC on the entire IP packet (not just the header) and place the resulting checksum in this field. When the IP packet is received, TCP re-runs the CRC on the entire packet to ensure the checksum is the same.

Data value (decimal): 4596

Data values in other bases:

Hexadecimal	1	1	F	4
Binary	0001	0001	1111	0100

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 61686 in decimal.

Start Bit: 128

Length: 16

IP > TCP PDU > *Options* for the selected HTTP PDU

<u>Field Name</u>: *Options*

<u>Description</u>: Like IP options, this field is optional and represents additional instructions not covered in the other TCP fields. Again, if an option does not fill up a 32-bit word, it will be filled in with padding bits.

<u>Data value (hexadecimal)</u>: 01 01 0A 1B 25 F3 A1 0B DD 73 58

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 160

<u>Length</u>: Variable

2.2.11 HTTP PDU for the selected HTTP PDU

IP > TCP > HTTP PDU > *Content Type* for the selected HTTP PDU

Field Name: *Content Type*

Description: The Content-Type entity-header field indicates the media type of the Entity-Body sent to the recipient.

Data value (ASCII): text/html; charset=iso – 8859-1\r\n

Data values in other bases:

Hexadecimal	4	3	6	F	6	E	7	4
Binary	0100	0011	0110	1111	0110	1110	0111	0100
ASCII	C		o		n		t	

Hexadecimal	6	5	6	E	7	4	2	D
Binary	0110	0101	0110	1110	0111	0100	0010	1101
ASCII	e		n		t		-	

Hexadecimal	5	4	7	9	7	0	6	5
Binary	0101	0100	0111	1001	0111	0000	0110	0101
ASCII	T		y		p		e	

Hexadecimal	3	A	2	0	7	4	6	5
Binary	0110	1010	0010	0000	0111	0100	0110	0101
ASCII	:				t		e	

Hexadecimal	7	8	7	4	2	F	6	8
Binary	0111	1000	0111	0100	0010	1111	0110	1000
ASCII	x		t		/		h	

Hexadecimal	7	4	6	D	6	C	3	B
Binary	0111	0100	0110	1101	0110	1100	0011	1011
ASCII	t		m		l		;	

Hexadecimal	2	0	6	3	6	8	6	1
Binary	0010	0000	0110	0011	0110	1000	0110	0001
ASCII			c		h		a	

Hexadecimal	7	2	7	3	6	5	7	4
Binary	0111	0010	0111	0011	0110	0101	0111	0100
ASCII	r		s		e		t	

Hexadecimal	3	D	6	9	7	3	6	F
Binary	0011	1101	0110	1001	0111	0011	0110	1111
ASCII	=		i		s		o	

Hexadecimal	2	D	3	8	3	8	3	5
Binary	0010	1101	0011	1000	0011	1000	0011	0101
ASCII	-		8		8		5	

Hexadecimal	3	9	2	D	3	1	0	D
Binary	0011	1001	0010	1101	0011	0001	0000	1101
ASCII	9		-		1		\r	

Hexadecimal	0	A
Binary	0000	1010
ASCII	\n	

Start Bit: 0

Length: 460

IP > TCP > HTTP PDU > *Date* for the selected HTTP PDU

Field Name: *Date*

Description: This field contains the date and time on which the web page was accessed.

Data value (ASCII): Date: Tue, 03 Feb 2004 23:08:10 GMT\r\n

Data values in other bases:

Hexadecimal	4	6	6	1	7	4	6	5
Binary	0110	0110	0110	0001	0111	0100	0110	0101
ASCII	D		a		t		e	

Hexadecimal	3	A	2	0	5	4	7	5
Binary	0010	1010	0010	0000	0101	0100	0111	0101
ASCII	:				T		u	

Hexadecimal	6	5	2	C	2	0	3	0
Binary	0110	0101	0010	1100	0010	0000	0011	0000
ASCII	e		,				0	

Hexadecimal	3	3	2	0	4	6	6	5
Binary	0011	0011	0010	0000	0100	0110	0110	0101
ASCII	3				F		e	

Hexadecimal	6	2	2	0	3	2	3	0
Binary	0110	0010	0010	0000	0011	0010	0011	0000
ASCII	b				2		0	

Hexadecimal	3	0	3	4	3	2	3	3
Binary	0011	0000	0011	0100	0011	0010	0011	0011
ASCII	0		4		2		3	

Hexadecimal	3	A	3	0	3	8	3	A
Binary	0011	1010	0011	0000	0011	1000	0011	1010
ASCII	:		0		8		:	

Hexadecimal	3	1	3	0	2	0	4	7
Binary	0011	0001	0011	0000	0010	0000	0100	0111
ASCII	1		0				G	

Hexadecimal	4	D	5	4	0	D	0	A
Binary	0100	1101	0101	0100	0000	1101	0000	1010
ASCII	M		T		\r		\n	

Start Bit: 0

Length: 296

IP > TCP > HTTP PDU > *HTTP* for the selected HTTP PDU

Field Name: *HTTP*

Description: This field displays the category of the page that is being displayed.

Data value (ASCII): HTTP/1.1 404 Not Found\r\n

Data values in other bases:

Hexadecimal	4	8	5	4	5	4	5	0
Binary	0100	1000	0101	0100	0101	0100	0101	0000
ASCII	H		T		T		P	

Hexadecimal	2	F	3	1	2	E	3	1
Binary	0010	1111	0011	0001	0010	1110	0011	0001
ASCII	/		1		.		1	

Hexadecimal	2	0	3	4	3	0	3	4
Binary	0010	0000	0011	0100	0011	0000	0011	0100
ASCII			4		0		4	

Hexadecimal	2	0	4	E	6	F	7	4
Binary	0010	0000	0100	1110	0110	1111	0111	0100
ASCII			N		o		t	

Hexadecimal	2	0	4	6	6	F	7	5
Binary	0010	0000	0100	0110	0110	1111	0111	0101
ASCII			F		o		u	

Hexadecimal	6	E	6	4	0	D	0	A
Binary	0110	1110	0110	0100	0000	1101	0000	1010
ASCII	n		d		\r		\n	

Start Bit: 0

Length: 40 octets

IP > TCP > HTTP PDU > *Server* for the selected HTTP PDU

<u>Field Name</u>: *Server*

<u>Description</u>: The Server response-header field contains information about the software used by the origin server to handle the request.

<u>Field Key</u>: *Not applicable*

<u>Data value (ASCII)</u>: Server: Apache/1.3.24 (Unix) PHP/4.2.1\r\n

<u>Data values in other bases</u>:

Hexadecimal	5	3	6	5	7	2	7	6
Binary	0101	0011	0110	0101	0111	0010	0111	0110
ASCII	S		e		r		v	

Hexadecimal	6	5	7	2	3	A	2	0
Binary	0110	0101	0111	0010	0011	1010	0010	0000
ASCII	e		r		:			

Hexadecimal	4	1	7	0	6	1	6	3
Binary	0110	0001	0111	0000	0110	0001	0110	0011
ASCII	A		p		a		c	

Hexadecimal	6	8	6	5	2	F	3	1
Binary	0110	1000	0110	0101	0010	1111	0011	0001
ASCII	h		e		/		1	

Hexadecimal	2	E	3	3	2	E	3	2
Binary	0010	1110	0011	0011	0010	1110	0011	0010
ASCII	.		3		.		2	

Hexadecimal	3	4	2	0	2	8	5	5
Binary	0011	0100	0010	0000	0010	1000	0101	0101
ASCII	4				(		U	

Hexadecimal	6	E	6	9	7	8	2	9
Binary	0110	1110	0110	1001	0111	1000	0010	1001
ASCII	n		i		x		)	

Hexadecimal	2	0	5	0	4	8	5	0
Binary	0010	0000	0101	0000	0100	1000	0101	0000
ASCII			P		H		P	

Hexadecimal	2	5	3	4	2	E	3	2
Binary	0010	0101	0011	0100	0010	1110	0011	0010
ASCII	/		4		.		2	

Hexadecimal	2	E	3	1	0	D	0	A
Binary	0010	1110	0011	0001	0000	1101	0000	1010
ASCII	.		1		\r		\n	

Start Bit: 0

Length: 312

IP > TCP > HTTP PDU > *Data* for the selected HTTP PDU

<u>Field Name</u>: *Data*

<u>Description</u>: This field stores the information that is actually contained in the HTTP Protocol.

<u>Data value (ASCII)</u>: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
<HTML><HEAD>\n
<TITLE>404 Not Found</TITLE>\n
</HEAD><BODY>\n
<H1>Not Found</H1>\n
The requested URL /~csis410/2003/bluetech/Requirements Specification Document Final-files/image002.gif was not found on this server.<p>\n
<HR>\n
<ADDRESS>Apache/1.3.24 Server at ares.cs.siena.edu Port 80</ADDRESS>\n
</BODY></HTML>\n

<u>Data values in other bases</u>:

Hexadecimal	3	C	2	1	4	4	4	5
Binary	0011	1100	0010	0001	0100	0100	0100	0101
ASCII	<		!		D		O	

Hexadecimal	4	3	5	4	5	9	5	0
Binary	0100	0011	0101	0100	0101	1001	0101	0000
ASCII	C		T		Y		P	

Hexadecimal	4	5	2	0	4	8	5	4
Binary	0100	0101	0010	0000	0100	1000	0101	0100
ASCII	E				H		T	

Hexadecimal	4	D	4	C	2	0	5	0
Binary	0100	1101	0110	1100	0010	0000	0101	0000
ASCII	M		L				P	

Hexadecimal	5	5	4	2	4	C	4	9
Binary	0101	0101	0100	0010	0100	1100	0100	1001
ASCII	U		B		L		I	

Hexadecimal	4	3	2	0	2	2	2	D
Binary	0100	0011	0010	0000	0010	0010	0010	1101
ASCII	C				"		-	

Hexadecimal	2	F	2	F	4	9	4	5
Binary	0010	1111	0010	1111	0100	1001	0100	0101
ASCII	/		/		I		E	

Hexadecimal	5	4	4	6	2	F	2	F
Binary	0101	0100	0100	0110	0010	1111	0010	1111
ASCII	T		F		/		/	

Hexadecimal	4	4	5	4	4	4	2	0
Binary	0100	0100	0101	0100	0100	0100	0010	0000
ASCII	D		T		D			

Hexadecimal	4	8	5	4	4	D	4	C
Binary	0100	1000	0101	0100	0100	1101	0100	1100
ASCII	H		T		M		L	

Hexadecimal	2	0	3	2	2	E	3	0
Binary	0010	0000	0011	0010	0010	1110	0011	0000
ASCII			2		.		0	

Hexadecimal	2	F	2	F	4	5	4	E
Binary	0010	1111	0010	1111	0100	0101	0100	1110
ASCII	/		/		E		N	

Hexadecimal	2	2	3	E	0	A	3	C
Binary	0010	0010	0011	1110	0000	1010	0011	1100
ASCII	“		>		\n		<	

Hexadecimal	4	8	5	4	4	D	4	C
Binary	0100	1000	0101	0100	0100	1101	0100	1100
ASCII	H		T		M		L	

Hexadecimal	3	E	3	C	4	8	4	5
Binary	0011	1110	0011	1100	0100	1000	0100	0101
ASCII	>		<		H		E	

Hexadecimal	4	1	4	4	3	E	0	A
Binary	0100	0001	0100	0100	0011	1110	0000	1010
ASCII	A		D		>		\n	

Hexadecimal	3	C	5	4	4	9	5	4
Binary	0011	1100	0101	0100	0100	1001	0101	0100
ASCII	<		T		I		T	

Hexadecimal	4	C	4	5	3	E	3	4
Binary	0100	1100	0100	0101	0011	1110	0011	0100
ASCII	L		E		>		4	

Hexadecimal	3	0	3	4	2	0	4	E
Binary	0011	0000	0011	0100	0010	0000	0100	1110
ASCII	0		4				N	

Hexadecimal	6	F	7	4	2	0	4	6
Binary	0110	1111	0111	0100	0010	0000	0100	0110
ASCII	o		t				F	

Hexadecimal	6	F	7	5	6	E	6	4
Binary	0110	1111	0111	0101	0110	1110	0110	0100
ASCII	o		u		n		d	

Hexadecimal	3	C	2	F	5	4	4	9
Binary	0011	1100	0010	1111	0101	0100	0100	1001
ASCII	<		/		T		I	

Hexadecimal	5	4	4	C	4	5	3	E
Binary	0101	0100	0100	1100	0100	0101	0011	1110
ASCII	T		L		E		>	

Hexadecimal	0	A	3	C	2	F	4	8
Binary	0000	1010	0011	1100	0010	1111	0100	1000
ASCII	\n		<		/		H	

Hexadecimal	4	5	4	1	4	4	3	E
Binary	0100	0101	0100	0001	0100	0100	0011	1110
ASCII	E		A		D		>	

Hexadecimal	3	C	4	2	4	F	4	4
Binary	0011	1100	0100	0010	0100	1111	0100	0100
ASCII	<		B		O		D	

Hexadecimal	5	9	3	E	0	A	3	C
Binary	0101	1001	0011	1110	0000	1010	0011	1100
ASCII	Y		>		\n		<	

Hexadecimal	4	8	3	1	3	E	4	E
Binary	0100	1000	0011	0001	0011	1110	0100	1110
ASCII	H		1		>		N	

Hexadecimal	6	F	7	4	2	0	4	6
Binary	0110	1111	0111	0100	0010	0000	0100	0110
ASCII	o		t				F	

Hexadecimal	6	F	7	5	6	E	6	4
Binary	0100	1111	0111	0101	0110	1110	0110	0100
ASCII	o		u		n		d	

Hexadecimal	3	C	2	F	4	8	3	1
Binary	0011	1100	0010	1111	0100	1000	0011	0001
ASCII	<		/		H		1	

Hexadecimal	3	E	0	A	5	4	6	8
Binary	0011	1110	0000	1010	0101	0100	0110	1000
ASCII	>		\n		T		h	

Hexadecimal	6	5	2	0	7	2	6	5
Binary	0110	0101	0010	0000	0111	0010	0110	0101
ASCII	e				r		e	

Hexadecimal	7	1	7	5	6	5	7	3
Binary	0111	0001	0111	0101	0110	0101	0111	0011
ASCII	q		u		e		s	

Hexadecimal	7	4	6	5	6	4	2	0
Binary	0111	0100	0110	0101	0110	0100	0010	0000
ASCII	t		e		d			

Hexadecimal	5	5	5	2	4	C	2	0
Binary	0101	0101	0101	0010	0100	1100	0010	0000
ASCII	U		R		L			

Hexadecimal	2	F	7	E	6	3	7	3
Binary	0010	1111	0111	1110	0110	0011	0111	0011
ASCII	/		~		c		s	

Hexadecimal	6	9	7	3	3	4	3	1
Binary	0110	1001	0111	0011	0011	0100	0011	0001
ASCII	i		s		4		1	

Hexadecimal	3	0	2	F	3	2	3	0
Binary	0011	0000	0010	1111	0011	0010	0011	0000
ASCII	0		/		2		0	

Hexadecimal	3	0	3	3	2	F	6	2
Binary	0011	0000	0011	0011	0010	1111	0110	0010
ASCII	0		3		/		b	

Hexadecimal	6	C	7	5	6	5	7	4
Binary	0110	1100	0111	0101	0110	0101	0111	0100
ASCII	l		u		e		t	

Hexadecimal	6	5	6	3	6	8	2	F
Binary	0110	0101	0110	0011	0110	1000	0010	1111
ASCII	e		c		h		/	

Hexadecimal	5	2	6	5	7	1	7	5
Binary	0101	0010	0110	0101	0111	0001	0111	0101
ASCII	R		e		q		u	

Hexadecimal	6	9	7	2	6	5	6	D
Binary	0110	1001	0111	0010	0110	0101	0110	1101
ASCII	i		r		e		m	

Hexadecimal	6	5	6	E	7	4	7	3
Binary	0110	0101	0110	1110	0111	0100	0111	0011
ASCII	e		n		t		s	

Hexadecimal	2	0	5	3	7	0	6	5
Binary	0010	0000	0101	0011	0111	0000	0110	0101
ASCII			S		p		e	

Hexadecimal	6	3	6	9	6	6	6	9
Binary	0110	0011	0110	1001	0110	0110	0110	1001
ASCII	c		i		f		i	

Hexadecimal	6	3	6	1	7	4	6	9
Binary	0110	0011	0110	0001	0111	0100	0110	1001
ASCII	c		a		t		i	

Hexadecimal	6	F	6	E	2	0	4	4
Binary	0110	1111	0110	1110	0010	0000	0100	0100
ASCII	o		n				D	

Hexadecimal	6	F	6	3	7	5	6	D
Binary	0110	1111	0110	0011	0111	0101	0110	1101
ASCII	o		c		u		m	

Hexadecimal	6	5	6	E	7	4	2	0
Binary	0110	0101	0110	1110	0111	0100	0010	0000
ASCII	e		n		t			

Hexadecimal	4	6	6	9	6	E	6	1
Binary	0100	0110	0110	1001	0110	1110	0110	0001
ASCII	F		i		n		a	

Hexadecimal	6	C	5	F	6	6	6	9
Binary	0110	1100	0101	1111	0110	0110	0110	1001
ASCII	l		-		f		i	

Hexadecimal	6	C	6	5	7	3	2	F
Binary	0110	1100	0110	0101	0111	0011	0010	1111
ASCII	l		e		s		/	

Hexadecimal	6	9	6	D	6	1	6	7
Binary	0110	1001	0110	1101	0110	0001	0110	0111
ASCII	i		m		a		g	

Hexadecimal	6	5	3	0	3	0	3	2
Binary	0110	0101	0011	0000	0011	0000	0011	0010
ASCII	e		0		0		2	

Hexadecimal	2	E	6	7	6	9	6	6
Binary	0010	1110	0110	0111	0110	1001	0110	0110
ASCII	.		g		i		f	

Hexadecimal	2	0	7	7	6	1	7	3
Binary	0001	0000	0111	0111	0110	0001	0111	0011
ASCII			w		a		s	

Hexadecimal	2	0	6	E	6	F	7	4
Binary	0010	0000	0110	1110	0110	1111	0111	0100
ASCII			n		o		t	

Hexadecimal	2	0	6	6	6	F	7	5
Binary	0010	0000	0110	0110	0110	1111	0111	0101
ASCII			f		o		u	

Hexadecimal	6	E	6	4	2	0	6	F
Binary	0110	1110	0110	0100	0010	0000	0110	1111
ASCII	n		d				o	

Hexadecimal	6	E	2	0	7	4	6	8
Binary	0110	1110	0010	0000	0111	0100	0110	1000
ASCII	n				t		h	

Hexadecimal	6	9	7	3	2	0	7	3
Binary	0110	1001	0111	0011	0010	0000	0111	0011
ASCII	i		s				s	

Hexadecimal	6	5	7	2	7	6	6	5
Binary	0110	0101	0111	0010	0111	0110	0110	0101
ASCII	e		r		v		e	

Hexadecimal	7	2	2	E	3	C	5	0
Binary	0111	0010	0010	1110	0011	1100	0101	0000
ASCII	r		.		<		p	

Hexadecimal	3	E	0	A	3	C	4	8
Binary	0011	1110	0000	1010	0011	1100	0100	1000
ASCII	>		\n		<		H	

Hexadecimal	5	2	3	E	0	4	3	C
Binary	0101	0010	0011	1110	0000	0100	0011	1100
ASCII	R		>		\n		<	

Hexadecimal	4	1	4	4	4	4	5	2
Binary	0100	0001	0100	0100	0100	0100	0101	0010
ASCII	A		D		D		R	

Hexadecimal	4	5	5	3	5	3	3	E
Binary	0100	0101	0101	0011	0101	0011	0011	1110
ASCII	E		S		S		>	

Hexadecimal	4	1	7	0	6	1	6	3
Binary	0100	0001	0111	0000	0110	0001	0110	0011
ASCII	A		p		a		c	

Hexadecimal	6	8	6	5	2	F	3	1
Binary	0110	1000	0110	0101	0010	1111	0011	0001
ASCII	h		e		/		1	

Hexadecimal	2	E	3	3	2	E	3	2
Binary	0010	1110	0011	0011	0010	1110	0011	0010
ASCII	.		3		.		2	

Hexadecimal	3	4	2	0	5	3	6	5
Binary	0011	0100	0010	0000	0101	0011	0110	0101
ASCII	4				S		e	

Hexadecimal	7	2	7	6	6	5	7	2
Binary	0111	0010	0111	0110	0110	0101	0111	0010
ASCII	r		v		e		r	

Hexadecimal	2	0	6	1	7	4	2	0
Binary	0010	0000	0110	0001	0111	0100	0010	0000
ASCII			a		t			

Hexadecimal	3	1	7	2	6	5	7	3
Binary	0011	0001	0111	0010	0110	0101	0111	0011
ASCII	a		r		e		s	

Hexadecimal	2	E	6	3	7	3	2	E
Binary	0010	1110	0110	0011	0111	0011	0010	1110
ASCII	.		c		s		.	

Hexadecimal	7	3	6	9	6	5	6	E
Binary	0111	0011	0110	1001	0110	0101	0110	1110
ASCII	s		i		e		n	

Hexadecimal	6	1	2	E	6	5	6	4
Binary	0110	0001	0010	1110	0110	0101	0110	0100
ASCII	a		.		e		d	

Hexadecimal	7	5	2	0	5	0	6	F
Binary	0111	0101	0010	0000	0101	0000	0110	1111
ASCII	u				P		o	

Hexadecimal	7	2	7	4	2	0	3	8
Binary	0111	0010	0111	0100	0010	0000	0011	1000
ASCII	r		t				8	

Hexadecimal	3	0	3	C	2	F	4	1
Binary	0011	0000	0010	1100	0010	1111	0100	0001
ASCII	0		<		/		A	

Hexadecimal	4	4	4	4	5	2	4	5
Binary	0100	0100	0100	0100	0101	0010	0100	0101
ASCII	D		D		R		E	

Hexadecimal	5	3	5	3	3	E	0	A
Binary	0101	0011	0101	0011	0011	1110	0000	1010
ASCII	S		S		>		\n	

Hexadecimal	3	C	2	F	4	2	4	F
Binary	0011	1100	0010	1111	0100	0010	0100	1111
ASCII	<		/		B		O	

Hexadecimal	4	4	5	9	3	E	3	C
Binary	0100	0100	0101	1001	0011	1110	0011	1100
ASCII	D		Y		>		<	

Hexadecimal	2	F	4	8	5	4	4	D
Binary	0010	1111	0100	1000	0101	0100	0100	1101
ASCII	/		H		T		M	

Hexadecimal	4	C	3	E	0	A
Binary	0100	1100	0011	1110	0000	1010
ASCII	L		>		\n	

Start Bit: 0

Length: 351

2.2.12 IP PDU for the selected SSH PDU

IP PDU > *IP Version* for the selected SSH PDU

Field Name: *IP Version*

Description: Version is a 4-bit field that indicates the format of the Internet header.

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 4

IP PDU > *Header Length* for the selected SSH PDU

<u>Field Name</u>: *Header Length*

<u>Description</u>: The IHL field is a 4-bit field indicating the length of the Internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5.

<u>Data value</u>: The value contained in our field is 20 bytes. This is the hexadecimal and decimal value of 5 multiplied by 4 bits.

<u>Data values in other bases</u>:

Hexadecimal	5
Binary	0101

<u>Start Bit</u>: 4

<u>Length</u>: 4

IP PDU > *Type of Service* for the selected SSH PDU

<u>Field Name</u>: *Type of Service*

<u>Description</u>: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a data gram through a particular network.

The major choice is a three-way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay 1 = Low Delay
Bit 4: (T) 0 = Normal Throughput 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability 1 = High Reliability

<u>Precedence</u>:

111 = Network Control
110 = Internetwork Control
101 = CRITIC/ECP
100 = Flash Overrided
011 = Flash
010 = Immediate
001 = Priority
000 = Routine

<u>Data value (decimal)</u>: 16

<u>Data values in other bases</u>:

Hexadecimal	1	0
Binary	0001	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 0 in decimal.

<u>Start Bit</u>: 8

<u>Length</u>: 8

IP PDU > *Total Length of Ethernet Frame* for the selected SSH PDU

<u>Field Name</u>: *Total Length of Ethernet Frame*

<u>Description</u>: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets.

<u>Data values (decimal)</u>: 100

<u>Data values in other bases</u>:

Hexadecimal	0	0	6	4
Binary	0000	0000	0110	0100

<u>Start Bit</u>: 16

<u>Length</u>: 16

IP PDU > *Identification* for the selected SSH PDU

Field Name: *Identification*

Description: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet.

Data value (decimal): 12490

Data values in other bases:

Hexadecimal	3	0	C	A
Binary	0011	0000	1100	1010

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 12942 in decimal.

Start Bit: 32

Length: 16

IP PDU > *Flags* for the selected SSH PDU

Field Name: *Flags*

Description: Flags is a 3-bit field that indicates directions for fragmentation.

Bit 0: reserved, must be 0	
Bit 1: (DF) 0 = May Fragment	1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment	1 = More Fragment

Data value (binary): 010

Data values in other bases: *Not applicable*

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 001 in binary.

Start Bit: 4

Length: 4

IP PDU > *Fragment Offset* for the selected SSH PDU

Field Name: *Fragment Offset*

Description: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

Start Bit: 51

Length: 13

IP PDU > *Time to Live* for the selected SSH PDU

Field Name: *Time to Live*

Description: Time to Live is an 8-bit field that indicates the maximum time the data gram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

Data value (decimal): 64

Data values in other bases:

Hexadecimal	4	0
Binary	0100	0000

Start Bit: 64

Length: 8

IP PDU > *Protocol* for the selected SSH PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	Hex	Protocol	Dec	Hex	Protocol
0	00	Reserved	22	16	Multiplexing
1	01	ICMP	23	17	DCN
2	02	Unassigned	24	18	TAC Monitoring
3	03	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	04	CMCC Gateway Monitoring Message	77	4D	Any local network
5	05	ST	100	64	SATNET and Backroom EXPAK
6	06	TCP	101	65	MIT Subnet Support
7	07	UCL	102-104	66-68	Unassigned
10	0A	Unassigned	105	69	SATNET Monitoring
11	0B	Secure	106	6A	Unassigned
12	0C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	0D	NVP	110-113	6E-71	Unassigned
14	0E	PUP	114	72	Backroom SATNET Monitoring
15	0F	Pluribus	115	73	Unassigned
16	10	SSH	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-0178	Unassigned
21	15	User Datagram	377	0179	Reserved

Data value (decimal): 6

Data values in other bases:

Hexadecimal	0	6
Binary	0000	0110

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the selected SSH PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. The Checksum is the 16-bit one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the initial value of its field is zero. When both header checksums are equal, then the header bits are correct. If either checksums vary, then a new, correct packet will need to be sent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 34734

Data values in other bases:

Hexadecimal	8	7	A	E
Binary	1000	0111	1010	1110

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the selected SSH PDU

<u>Field Name</u>: *Source IP Address*

<u>Description</u>: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.

<u>Data value (decimal)</u>: 192.168.0.101

<u>Data values in other bases</u>:

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

<u>Start Bit</u>: 96

<u>Length</u>: 32

IP PDU > *Destination IP Address* for the selected SSH PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.

Data value (decimal): 192.168.0.102

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	6	6
Binary	1100	0000	1010	1000	0000	0000	0110	0110

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 192.168.0.39 in decimal.

Start Bit: 128

Length: 32

2.2.13 TCP PDU for the selected SSH PDU

IP > TCP PDU > *Source Port Number* for the selected SSH PDU

<u>Field Name</u>: *Source Port Number*

<u>Description</u>: This 16-bit number represents the name of the application that sent the data in the IP packet.

<u>Data value (decimal)</u>: 1243

<u>Data values in other bases</u>:

Hexadecimal	0	4	D	B
Binary	0000	0100	1101	1011

<u>Start Bit</u>: 0

<u>Length</u>: 16

IP > TCP PDU > *Destination Port Number* for the selected SSH PDU

Field Name: *Destination Port Number*

Description:
This 16-bit number represents the name of the application that is to receive the data contained within the IP packet. This is one of the major differences between a Layer 3 and a Layer 4 header: the Layer 3 header contains the IP address of the computer that is to receive the IP packet; once that packet has been received, the port address in the Layer 4 header ensures that the data contained within that IP packet is passed to the correct application on that computer.

This key indicates assigned port number values:

Dec	Port Numbers
0	Reserved
1-32767	Internet registered ("well-known") protocols
32768-98303	Reserved, to allow TCPv7-TCPv4 conversion
98304 & up	Dynamic assignment

Data value (decimal): 22

Data values in other bases:

Hexadecimal	0	0	1	6
Binary	0000	0000	0001	0000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 1243 in decimal.

Start Bit: 16

Length: 16

Source: http://www.zvon.org/tmRFC/RFC1475/Output/chapter4.html

IP > TCP PDU > *Sequence Number* for the selected SSH PDU

<u>Field Name</u>: *Sequence Number*

<u>Description</u>: TCP is responsible for ensuring that all IP packets sent are actually received. When an application's data is packaged into IP packets, TCP will give each IP packet a sequence number. Once all the packets have arrived at the receiving computer, TCP uses the number in this 32-bit field to ensure that all of the packets actually arrived and are in the correct sequence.

<u>Data value (decimal)</u>: 4008673261

<u>Data values in other bases</u>:

Hexadecimal	E	E	E	F	7	F	E	D
Binary	1110	1110	1110	1111	0111	1111	1110	1101

<u>Start Bit</u>: 32

<u>Length</u>: 32

IP > TCP PDU > *Acknowledgement Number* for the selected SSH PDU

<u>Field Name</u>: *Acknowledgement Number*

<u>Description</u>: This number is used by the receiving computer to acknowledge which packets have successfully arrived. This number will be the sequence number of the next packet the receiver is ready to receive.

<u>Data value (decimal)</u>: 3798775616

<u>Data values in other bases</u>:

Hexadecimal	E	2	6	C	B	7	4	0
Binary	1110	0010	0110	1100	1011	0111	0100	0000

<u>Start Bit</u>: 64

<u>Length</u>: 32

IP > TCP PDU > *Length or Offset* for the selected SSH PDU

<u>Field Name</u>: *Length or Offset*

<u>Description</u>: This is identical in concept to the header length in an IP packet, except this time it indicates the length of the TCP header.

<u>Data value (decimal)</u>: 8

<u>Data values in other bases</u>:

Hexadecimal	0	8
Binary	0000	1000

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 32 in decimal.

<u>Start Bit</u>: 96

<u>Length</u>: 4

IP > TCP PDU > *Control Flags* for the selected SSH PDU

Field Name: *Control Flags*

Description: Every TCP packet contains this 6-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

- Urgent (URG)
- Acknowledgement (ACK)
- Push (PSH)
- Reset (RST)
- Synchronize (SYN)
- Finish (FIN)

Data value (decimal): 24

Data values in other bases:

Hexadecimal	1	8
Binary	0001	1000

Start Bit: 106

Length: 6

IP > TCP PDU > *Window Size* for the selected SSH PDU

Field Name: *Window Size*

Description: Every TCP packet contains this 16-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

Data value (decimal): 32120

Data values in other bases:

Hexadecimal	7	D	7	8
Binary	0111	1101	0111	1000

Start Bit: 112

Length: 16

IP > TCP PDU > ***TCP Checksum*** **for the selected SSH PDU**

Field Name: *TCP Checksum*

Description: Unlike IP, TCP is responsible for ensuring that the entire IP packet arrived intact. TCP will run a CRC on the entire IP packet (not just the header) and place the resulting checksum in this field. When the IP packet is received, TCP re-runs the CRC on the entire packet to ensure the checksum is the same.

Data value (decimal): 35786

Data values in other bases:

Hexadecimal	8	B	C	A
Binary	1000	1011	1100	1010

Start Bit: 128

Length: 16

IP > TCP PDU > *Options* for the selected SSH PDU

<u>Field Name</u>: *Options*

<u>Description</u>: Like IP options, this field is optional and represents additional instructions not covered in the other TCP fields. Again, if an option does not fill up a 32-bit word, it will be filled in with padding bits.

<u>Data value</u>: 01 01 08 0A 1B 25 F3 A1 0B DD 73 58

- The data value used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's data value was 01 01 08 0A 0B D1 8D EC 1A AC 06 AB in hexidecimal.

<u>Start Bit</u>: 160

<u>Length</u>: Variable

2.2.14 IP PDU for the selected TELNET PDU

IP PDU > *IP Version* for the selected TELNET PDU

Field Name: *IP Version*

Description: Version is a 4-bit field that indicates the format of the Internet header.

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 4

IP PDU > *Header Length* for the selected TELNET PDU

Field Name: *Header Length*

Description: The IHL field is a 4-bit field indicating the length of the Internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5.

Data value: The value contained in our field is 20 bytes. This is the hexadecimal and decimal value of 5 multiplied by 4 bits.

Data value (decimal): 5

Data values in other bases:

Hexadecimal	5
Binary	0101

Start Bit: 4

Length: 4

IP PDU > *Type of Service* for the selected TELNET PDU

<u>Field Name</u>: *Type of Service*

<u>Description</u>: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a data gram through a particular network.
The major choice is a three-way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay 1 = Low Delay
Bit 4: (T) 0 = Normal Throughput 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability 1 = High Reliability

<u>Precedence</u>:
111 = Network Control
110 = Internet work Control
101 = CRITIC/ECP
100 = Flash Override
011 = Flash
010 = Immediate
001 = Priority
000 = Routine

<u>Data value (decimal)</u>: 00

<u>Data values in other bases</u>:

Hexadecimal	0	0
Binary	0000	0000

<u>Start Bit</u>: 8

<u>Length</u>: 8

IP PDU > *Total Length* for the selected TELNET PDU

Field Name: *Total Length*

Description: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets.

Data values (decimal): 125

Data values in other bases:

Hexadecimal	00	7D
Binary	1111	1101

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 0011 1110

Start Bit: 16

Length: 16

IP PDU > *Identification* for the selected TELNET PDU

<u>Field Name</u>: *Identification*

<u>Description</u>: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet.

<u>Data value (decimal)</u>: 51102

<u>Data values in other bases</u>:

Hexadecimal	C	7	9	E
Binary	1100	0111	1001	1110

.

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 1100 0111 0101 0111

<u>Start Bit</u>: 32

<u>Length</u>: 16

IP PDU > *Flags* for the selected TELNET PDU

<u>Field Name</u>: *Flags*

<u>Description</u>: Flags is a 3-bit field that indicates directions for fragmentation.
Bit 0: reserved, must be 0
Bit 1: (DF) 0 = May Fragment 1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment 1 = More Fragment

<u>Data value (binary)</u>: 010

<u>Data values in other bases</u>: *Not applicable*

<u>Start Bit</u>: 48

<u>Length</u>: 3

IP PDU > ***Fragment Offset*** for the selected TELNET PDU

Field Name: *Fragment Offset*

Description: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

Data value (decimal): 0

Data values in other bases:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

Start Bit: 51

Length: 13

IP PDU > *Time to Live* for the selected TELNET PDU

Field Name: *Time to Live*

Description: Time to Live is an 8-bit field that indicates the maximum time the data gram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

Data value (decimal): 64

Data values in other bases:

Hexadecimal	4	0
Binary	0100	0000

Start Bit: 64

Length: 8

IP PDU > *Protocol* for the selected TELNET PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	Hex	Protocol	Dec	Hex	Protocol
0	00	Reserved	22	16	Multiplexing
1	01	ICMP	23	17	DCN
2	02	Unassigned	24	18	TAC Monitoring
3	03	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	04	CMCC Gateway Monitoring Message	77	4D	Any local network
5	05	ST	100	64	SATNET and Backroom EXPAK
6	06	TCP	101	65	MIT Subnet Support
7	07	UCL	102-104	66-68	Unassigned
10	0A	Unassigned	105	69	SATNET Monitoring
11	0B	Secure	106	6A	Unassigned
12	0C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	0D	NVP	110-113	6E-71	Unassigned
14	0E	PUP	114	72	Backroom SATNET Monitoring
15	0F	Pluribus	115	73	Unassigned
16	10	Telnet	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-0178	Unassigned
21	15	User Datagram	377	0179	Reserved

Data value (decimal): 6

Data values in other bases:

Hexadecimal	0	6
Binary	0000	0110

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the selected TELNET PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. The Checksum is the 16-bit one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the initial value of its field is zero. When both header checksums are equal, then the header bits are correct. If either checksums vary, then a new, correct packet will need to be sent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 61695

Data values in other bases:

Hexadecimal	F	0	F	F
Binary	1111	0000	1111	1111

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 1111 0001 1000 0101

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the selected TELNET PDU

Field Name: *Source IP Address*

Description: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.
The source address of the sender of the IP data gram:
NET ID ADDRESS RANGE
000-127 Class A 10.0.0.0-10.225.225
128-191 Class B 172.16.0.0-172.31.255.255
192-223 Class C 192.168.0.0-192.168.255.255
224-239 Class D (multicast)
240-255 Class E (experimental)
HOST ID
0 Network value; broadcast(old)
255 Broadcast

Data value (decimal): 192.168.0.101

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

Start Bit: 96

Length: 32

IP PDU > *Destination IP Address* for the selected TELNET PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.
The IP address of the host where this data gram is being sent:
NET ID ADDRESS RANGE
000-127 Class A 10.0.0.0-10.225.225
128-191 Class B 172.16.0.0-172.31.255.255
192-223 Class C 192.168.0.0-192.168.255.255
224-239 Class D (multicast)
240-255 Class E (experimental)
HOST ID
0 Network value; broadcast(old)
255 Broadcast

Data value (decimal): 192.168.0.39

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	2	7
Binary	1100	0000	1010	1000	0000	0000	0010	0111

Start Bit: 128

Length: 32

IP > TCP PDU > *Options* for the selected TELNET PDU

Field Name: *Options*

<u>Description</u>: The options may or may not appear in Ethernet packets. Options have to be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular packet, not their implementation.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option.

Case 1: A single octet of option type
Case 2: An option-type octet, an option-length octet, and the actual option-data octets

Data value: *Not applicable*

<u>Start Bit</u>: 160

<u>Length</u>: Variable (0-40)

2.2.15 TCP PDU for the selected TELNET PDU

IP > TCP PDU > *Source Port Number* for the selected TELNET PDU

Field Name: *Source Port Number*

Description: A 16-bit address assigned by the sending computer, to the application program sending data as TCP data grams.

Data value (ASCII): TELNET (23)

Data values in other bases:

Hexadecimal	0	0	1	7
Binary	0000	0000	0001	0111

Start Bit: 0

Length: 16

IP > TCP PDU > *Destination Port Number* for the selected TELNET PDU

<u>Field Name</u>: *Destination Port Number*

<u>Description</u>: This 16-bit number represents the name of the application that is to receive the data contained within the IP packet. This is one of the major differences between a Layer 3 and a Layer 4 header: the Layer 3 header contains the IP address of the computer that is to receive the IP packet; once that packet has been received, the port address in the Layer 4 header ensures that the data contained within that IP packet is passed to the correct application on that computer.

Dec	**Port Numbers**
0	Reserved
1-32767	Internet registered ("well-known") protocols
32768-98303	Reserved, to allow TCPv7-TCPv4 conversion
98304 & up	Dynamic assignment

<u>Data value (decimal)</u>: 32805

<u>Data values in other bases:</u>

Hexadecimal	8	0	2	5
Binary	1000	0000	0010	0101

<u>Start Bit</u>: 16

<u>Length</u>: 16

<u>Source</u>: http://www.zvon.org/tmRFC/RFC1475/Output/chapter4.html

IP > TCP PDU > *Sequence Number* for the selected TELNET PDU

Field Name: *Sequence Number*

Description: TCP is responsible for ensuring that all IP packets sent are actually received. When an application's data is packaged into IP packets, TCP will give each IP packet a sequence number. Once all the packets have arrived at the receiving computer, TCP uses the number in this 32-bit field to ensure that all of the packets actually arrived and are in the correct sequence.

Data value (decimal): 263530302

Data values in other bases:

Hexadecimal	9	D	1	3	8	8	6	D
Binary	1001	1101	0001	0011	1000	1000	0110	1101

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 1001 1101 0001 0011 1000 1000 0000 1000

Start Bit: 32

Length: 32

IP > TCP PDU > *Acknowledgement Number* for the selected TELNET PDU

Field Name: *Acknowledgement Number*

Description: The receiving computer to acknowledge which packets have successfully arrived uses this number. This number will be the sequence number of the next packet the receiver is ready to receive.

Data value: 2526101273

Data values in other bases:

Hexadecimal	9	6	9	1	3	F	1	9
Binary	1001	0110	1001	0001	0011	1111	0001	1001

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 1001 0110 1001 0001 0011 1111 0000 0101

Start Bit: 64

Length: 32

IP > TCP PDU > *Length* for the selected TELNET PDU

Field Name: *Length*

Description: This is identical in concept to the header length in an IP packet, except this time it indicates the length of the TCP header.

Data value (decimal): 128

Data values in other bases:

Hexadecimal	8	0
Binary	1000	0000

Start Bit: 96

Length: 4

IP > TCP PDU > *Reserved* for the selected TELNET PDU

<u>Field Name</u>: *Reserved*

<u>Description</u>: 4 bits; set to 0
ECN bits (used when ECN employed; else 00)
CWR (1=sender has cut congestion window in half
ECN-Echo (1=receiver has cut congestion window in half

<u>Data value (decimal)</u>: 0

<u>Data values in other bases</u>:

Hexadecimal	0	0
Binary	0000	0000

<u>Start Bit</u>: 100

<u>Length</u>: 6

IP > TCP PDU > *Control Flags* for the selected TELNET PDU

Field Name: *Control Flags*

Description: Every TCP packet contains this 6-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

- Urgent (URG)
- Acknowledgement (ACK)
- Push (PSH)
- Reset (RST)
- Synchronize (SYN)
- Finish (FIN)

Data value (decimal): 24

Data values in other bases:

Hexadecimal	1	8
Binary	0001	1000

Start Bit: 106

Length: 6

IP > TCP PDU > *Window Size* for the selected TELNET PDU

<u>Field Name</u>: *Window Size*

<u>Description</u>: Every TCP packet contains this 16-bit value that indicates how many octets it can receive at once. When IP packets are received, they are placed in a temporary area of RAM known as a buffer until the receiving computer has a chance to process them; this value represents how big a buffer the receiving host has made available for this temporary storage of IP packets.

<u>Data value (decimal)</u>: 32120

<u>Data values in other bases</u>:

Hexadecimal	7	D	7	8
Binary	0111	1101	0111	1000

<u>Start Bit</u>: 116

<u>Length</u>: 16

IP > TCP PDU > *TCP Checksum* for the selected TELNET PDU

Field Name: *TCP Checksum*

Description: Unlike IP, TCP is responsible for ensuring that the entire IP packet arrived intact. TCP will run a CRC on the entire IP packet (not just the header) and place the resulting checksum in this field. When the IP packet is received, TCP re-runs the CRC on the entire packet to ensure the checksum is the same.

Data value (decimal): 63307

Data values in other bases:

Hexadecimal	F	7	4	B
Binary	1111	0111	0100	1011

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 0101 1001 1000 1001

Start Bit: 128

Length: 16

IP > TCP PDU > *Urgent Pointer* for the selected TELNET PDU

<u>Field Name</u>: *Urgent Pointer*

<u>Description</u>: A 16-bit pointer to the last byte within the segment, which is urgent and should be expected in delivery-valid only if URG flag is set.

<u>Data value (decimal)</u>: 0

<u>Data values in other bases</u>:

Hexadecimal	0	0	0	0
Binary	0000	0000	0000	0000

<u>Start Bit</u>: 144

<u>Length</u>: 16

IP > TCP PDU > *Options* for the selected TELNET PDU

Field Name: *Options*

Description: Like IP options, this field is optional and represents additional instructions not covered in the other TCP fields. Again, if an option does not fill up a 32-bit word, it will be filled in with padding bits.

Data value (hexadecimal): 01 01 08 0A 0B D1 8D EC 1A AC 06 AB

Data values in other bases:

Hexadecimal	0	1	0	1	0	8	0	A	0	B
Binary	0000	0001	0000	0001	0000	1000	0000	1010	0000	1011

Hexadecimal	D	1	8	D	E	C	1	A	A	C
Binary	1101	0001	1000	1101	1110	1100	0001	1010	1010	1100

Hexadecimal	0	6	A	B
Binary	0000	0110	1010	1011

2.2.16 TELNET PDU for the selected TELNET PDU

IP >TCP > TELNET PDU for the TELNET Packet

Field Name: *Telnet PDU*

Description: PASS (Password)
The argument field is a Telnet string specifying the user's password. This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control.

What is contained in the Packet:
Request: PASS

Data Values (hexadecimal): 50 61 73 73 77 6F 72 64 3A 20

Data Values in Other Bases:

Hexadecimal	5	0	6	1	7	3	7	7	6	F
Binary	0101	0000	0110	0001	0111	0011	0111	0111	0110	1111
ASCII	P		a		s		w		o	

Hexadecimal	7	2	6	4	3	A	2	0
Binary	0111	0010	0110	0100	0011	1010	0010	0000
ASCII	r		d		:		©	

RFC Link: **http://www.ietf.org/rfc/rfc0959.txt?number=959**

2.2.17 ARP PDU for the selected ARP PDU

ARP PDU> *Hardware Type* > for the selected ARP PDU

Field Name: *Hardware Type*

Description: The physical media that communicates on the network.

Data value (decimal): 1

Data values in other bases:

Hexadecimal	0	0	0	1
Binary	0000	0000	0000	0001

Start Bit: 0

Length: 16

ARP PDU > *Protocol Type* > for the selected ARP PDU

Field Name: *Protocol Type*

Description: Defines the protocol that the terminals are using to connect with each other.

Data value (decimal): 2048

Data values in other bases:

Hexadecimal	0	8	0	0
Binary	0000	1000	0000	0000

Start Bit: 16

Length: 16

ARP PDU > *Hardware size* > for the selected ARP PDU

Field Name: *Hardware size*

Description: This field determines the type of hardware used.

Data value (decimal): 6

Data values in other bases:

Hexadecimal	0	6
Binary	0000	0110

Start Bit: 32

Length: 8

ARP PDU >*Protocol size*>for the selected ARP PDU

<u>Field Name</u>: *Protocol size*

<u>Description</u>: Determines the protocol used in the request or response**.** .

<u>Data value (decimal)</u>: 2

<u>Data values in other bases</u>:

Hexadecimal	0	2
Binary	0000	0010

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 0000 0100

<u>Start Bit</u>: 40

<u>Length</u>: 8

ARP PDU > *Opcode Request*> for the selected ARP PDU

Field Name: *Opcode Request*

Description: Determines weather a request or a response is being called upon.

Data value (decimal): 1

Data values in other bases:

Hexadecimal	0	0	0	1
Binary	0000	0000	0000	0001

Start Bit: 48

Length: 16

ARP PDU > *Sender Hardware Address* > for the selected ARP PDU

Field Name: *Sender Hardware Address*

Description: The Physical address or MAC address of the network adapter of the sender's terminal.
00000C Cisco
00000E Fujitsu
00000F NeXT
000010 Sytek
00001D Cabletron
000020 DIAB (Data Intdustrier AB)
000022 Visual Technology
00002A TRW
000032 GPT Limited (reassigned from GEC Computers Ltd)
00005A S & Koch
00005E IANA
000065 Network General
00006B MIPS
000077 MIPS
00007A Ardent
000089 Cayman Systems Gatorbox
000093 Proteon
00009F Ameristar Technology
0000A2 Wellfleet
0000A3 Network Application Technology
0000A6 Network General (internal assignment, not for products)
0000A7 NCD   X-terminals
0000A9 Network Systems
0000AA Xerox   Xerox machines
0000B3 CIMLinc
0000B7 Dove   Fastnet
0000BC Allen-Bradley
0000C0 Western Digital
0000C5 Farallon phone net card
0000C6 HP Intelligent Networks Operation (formerly Eon Systems)
0000C8 Altos
0000C9 Emulex   Terminal Servers
0000D7 Dartmouth College (NED Router)
0000D8 3Com? Novell?   PS/2
0000DD Gould
0000DE Unigraph
0000E2 Acer Counterpoint
0000EF Alantec
0000FD High Level Hardvare (Orion, UK)

000102 BBN BBN internal usage (not registered)
0020AF 3COM ???
001700 Kabel
008064 Wyse Technology / Link Technologies
00802B IMAC ???
00802D Xylogics, Inc. Annex terminal servers
00808C Frontier Software Development
0080C2 IEEE 802.1 Committee
0080D3 Shiva
00AA00 Intel
00DD00 Ungermann-Bass
00DD01 Ungermann-Bass
020701 Racal InterLan
020406 BBN BBN internal usage (not registered)
026086 Satelcom MegaPac (UK)
02608C 3Com IBM PC; Imagen; Valid; Cisco
02CF1F CMC Masscomp; Silicon Graphics; Prime EXL
080002 3Com (Formerly Bridge)
080003 ACC (Advanced Computer Communications)
080005 Symbolics Symbolics LISP machines
080008 BBN
080009 Hewlett-Packard
08000A Nestar Systems
08000B Unisys
080011 Tektronix, Inc.
080014 Excelan BBN Butterfly, Masscomp, Silicon Graphics
080017 NSC
08001A Data General
08001B Data General
08001E Apollo
080020 Sun Sun machines
080022 NBI
080025 CDC
080026 Norsk Data (Nord)
080027 PCS Computer Systems GmbH
080028 TI Explorer
08002B DEC
08002E Metaphor
08002F Prime Computer Prime 50-Series LHC300
080036 Intergraph CAE stations
080037 Fujitsu-Xerox
080038 Bull
080039 Spider Systems
080041 DCA Digital Comm. Assoc.
080045 ???? (maybe Xylogics, but they claim not to know this number)
080046 Sony

080047 Sequent
080049 Univation
08004C Encore
08004E BICC
080056 Stanford University
080058 ??? DECsystem-20
08005A IBM
080067 Comdesign
080068 Ridge
080069 Silicon Graphics
08006E Concurrent Masscomp
080075 DDE (Danish Data Elektronik A/S)
08007C Vitalink TransLAN III
080080 XIOS
080086 Imagen/QMS
080087 Xyplex terminal servers
080089 Kinetics AppleTalk-Ethernet interface
08008B Pyramid
08008D XyVision XyVision machines
080090 Retix Inc Bridges
484453 HDS ???
800010 AT&T
AA0000 DEC obsolete
AA0001 DEC obsolete
AA0002 DEC obsolete
AA0003 DEC Global physical address for some DEC machines
AA0004 DEC Local logical address for systems running DECNET

Data value (hexadecimal): 00:00:E6: 34:ED:A3

Data values in other bases: *Not Applicable*

Start Bit: 64

Length: 48

ARP PDU > *Sender Protocol Address* > for the selected ARP PDU

<u>Field Name</u>: *Sender Protocol Address*

<u>Description</u>: The protocol of the sender computer. This is used to identify the senders Protocol.

<u>Data value (decimal)</u>: 192.168.0.101

Hexadecimal	C	0	A	8	0	0	6	5
Binary	1100	0000	1010	1000	0000	0000	0110	0101

- The Binary sequence used here is that of Blue Technology’s data. Blue’s data matches up with the dump file from ethereal while Mirage’s doesn’t.
- Mirage’s binary sequence was 1100 0000 1010 1000 0000 0000 0001 0110

<u>Start Bit</u>: 112

<u>Length</u>: 32

ARP PDU > *Target Hardware* > for the selected ARP PDU

Field Name: *Target Hardware*

Description: The Physical address or MAC address of the network adapter of the target terminal.
00000C Cisco
00000E Fujitsu
00000F NeXT
000010 Sytek
00001D Cabletron
000020 DIAB (Data Intdustrier AB)
000022 Visual Technology
00002A TRW
000032 GPT Limited (reassigned from GEC Computers Ltd)
00005A S & Koch
00005E IANA
000065 Network General
00006B MIPS
000077 MIPS
00007A Ardent
000089 Cayman Systems Gatorbox
000093 Proteon
00009F Ameristar Technology
0000A2 Wellfleet
0000A3 Network Application Technology
0000A6 Network General (internal assignment, not for products)
0000A7 NCD X-terminals
0000A9 Network Systems
0000AA Xerox Xerox machines
0000B3 CIMLinc
0000B7 Dove Fastnet
0000BC Allen-Bradley
0000C0 Western Digital
0000C5 Farallon phone net card
0000C6 HP Intelligent Networks Operation (formerly Eon Systems)
0000C8 Altos
0000C9 Emulex Terminal Servers
0000D7 Dartmouth College (NED Router)
0000D8 3Com? Novell? PS/2
0000DD Gould
0000DE Unigraph
0000E2 Acer Counterpoint
0000EF Alantec
0000FD High Level Hardvare (Orion, UK)
000102 BBN BBN internal usage (not registered)

0020AF 3COM ???
001700 Kabel
008064 Wyse Technology / Link Technologies
00802B IMAC ???
00802D Xylogics, Inc. Annex terminal servers
00808C Frontier Software Development
0080C2 IEEE 802.1 Committee
0080D3 Shiva
00AA00 Intel
00DD00 Ungermann-Bass
00DD01 Ungermann-Bass
020701 Racal InterLan
020406 BBN BBN internal usage (not registered)
026086 Satelcom MegaPac (UK)
02608C 3Com IBM PC; Imagen; Valid; Cisco
02CF1F CMC Masscomp; Silicon Graphics; Prime EXL
080002 3Com (Formerly Bridge)
080003 ACC (Advanced Computer Communications)
080005 Symbolics Symbolics LISP machines
080008 BBN
080009 Hewlett-Packard
08000A Nestar Systems
08000B Unisys
080011 Tektronix, Inc.
080014 Excelan BBN Butterfly, Masscomp, Silicon Graphics
080017 NSC
08001A Data General
08001B Data General
08001E Apollo
080020 Sun Sun machines
080022 NBI
080025 CDC
080026 Norsk Data (Nord)
080027 PCS Computer Systems GmbH
080028 TI Explorer
08002B DEC
08002E Metaphor
08002F Prime Computer Prime 50-Series LHC300
080036 Intergraph CAE stations
080037 Fujitsu-Xerox
080038 Bull
080039 Spider Systems
080041 DCA Digital Comm. Assoc.
080045 ???? (maybe Xylogics, but they claim not to know this number)
080046 Sony
080047 Sequent

080049 Univation
08004C Encore
08004E BICC
080056 Stanford University
080058 ??? DECsystem-20
08005A IBM
080067 Comdesign
080068 Ridge
080069 Silicon Graphics
08006E Concurrent Masscomp
080075 DDE (Danish Data Elektronik A/S)
08007C Vitalink TransLAN III
080080 XIOS
080086 Imagen/QMS
080087 Xyplex terminal servers
080089 Kinetics AppleTalk-Ethernet interface
08008B Pyramid
08008D XyVision XyVision machines
080090 Retix Inc Bridges
484453 HDS ???
800010 AT&T
AA0000 DEC obsolete
AA0001 DEC obsolete
AA0002 DEC obsolete
AA0003 DEC Global physical address for some DEC machines
AA0004 DEC Local logical address for systems running DECNET

Data value (hexadecimal): 00:00:00:00:00:00

Data values in other bases: *Not applicable*

Start Bit: 144

Length: 48

ARP PDU>*Target Protocol Address*> for the selected ARP PDU

Field Name: *Target Protocol Address*

Description: The protocol of the sender computer. This is used to identify the targets Protocol.

Data value (decimal): 192.168.0.145

Hexadecimal	C	0	A	8	0	0	9	1
Binary	1100	0000	1010	1000	0000	0000	1001	0001

Start Bit: 196

Length: 32

2.2.18 IP PDU for the selected UPD PDU

IP PDU > *IP Version* for the selected UDP PDU

Field Name: *IP Version*

Description: Version is a 4-bit field that indicates the format of the Internet header.

Data value (decimal): 4

Data values in other bases:

Hexadecimal	4
Binary	0100

Start Bit: 0

Length: 4

IP PDU > *Header Length* for the selected UDP PDU

Field Name: *Header Length*

Description: The IHL field is a 4-bit field indicating the length of the internet header in 32 bit words, and thus points to the beginning of the data. The minimum value of a correct header is 5.

Data value (decimal): 5.

Data values in other bases:

Hexadecimal	0	5
Binary	0000	0101

Start Bit: 4

Length: 4

IP PDU > *Type of Service* for the selected UDP PDU

Field Name: *Type of Service*

Description: Type of Service is an 8-bit field that provides and indication of the abstract parameters of the quality of service desired. These parameters guide the selection of the actual service parameters when transmitting a datagram through a particular network.

Bits 0-2: Precedence
Bit 3: (D) 0 = Normal Delay 1 = Low Delay
Bit 4: (T) 0 = Normal Throughput 1 = High Throughput
Bit 5: (R) 0 = Normal Reliability 1 = High Reliability

Precedence:

111 = Network Control 011 = Flash
110 = Internetwork Control 010 = Immediate
101 = CRITIC/ECP 001 = Priority
100 = Flash Overrided 000 = Routine

Data value (decimal): 16

Data values in other bases:

Hexadecimal	1	0
Binary	0001	0000

Start Bit: 8

Length: 8

IP PDU > *Total Length* for the selected UDP PDU

Field Name: *Total Length*

Description: Total Length is a 16-bit field that indicates the length of the frame, measured in octets, including Internet header and data. The maximum size is 2^{16}-1 or 65,535 octets; however, the recommended maximum size is 576 octets.

Data values (decimal): 296

Data values in other bases:

Hexadecimal	0	1	2	8
Binary	0000	0001	0010	1000

Start Bit: 16

Length: 16

IP PDU > *Identification* for the selected UDP PDU

Field Name: *Identification*

Description: Identification is a 16-bit field. An identifying value is assigned by the sender to aid in assembling the fragments of a data gram. The identifier is chosen based on the need to provide a way to uniquely identify the fragments and protocol for the time the data gram or any fragment could be alive in the Internet.

Data value (decimal): 48087

Data values in other bases:

Hexadecimal	B	B	D	7
Binary	1011	1011	1101	0111

Start Bit: 32

Length: 16

IP PDU > *Flags* for the selected UDP PDU

Field Name: *Flags*

Description: Flags is a 3-bit field that indicates directions for fragmentation.
Bit 0: reserved, must be 0
Bit 1: (DF) 0 = May Fragment 1 = Don't Fragment
Bit 2: (MF) 0 = Last Fragment 1 = More Fragment

Data value (binary): 000

Data values in other bases: *Not applicable*

Start Bit: 48

Length: 3

IP PDU > *Fragment Offset* for the selected UDP PDU

<u>Field Name</u>: *Fragment Offset*

<u>Description</u>: The Fragment Offset is a 13- bit field indicating where in the Ethernet frame this fragment begins. The Fragment Offset is measured in units of 8 octets, and the first fragment has offset 0.

<u>Data value (decimal)</u>: 0

<u>Data values in other bases</u>:

Hexadecimal	0	0
Binary	0000	0000

<u>Start Bit</u>: 51

<u>Length</u>: 13

IP PDU > *Time to Live* for the selected UDP PDU

Field Name: *Time to Live*

Description: Time to Live is an 8-bit field that indicates the maximum time the data gram is allowed to remain in the Internet. If this field contains the value 0, then the data gram must be destroyed. This field is modified in Internet header processing. The time is measure in units of seconds, and is set by the sender to the maximum time the data gram is allowed to be in the Internet. This field is decreased at each point that the Internet header is processed. The intention is to cause undeliverable packets to be discarded, and to bind the maximum data gram lifetime.

Data value (decimal): 60

Data values in other bases:

Hexadecimal	3	C
Binary	0011	1100

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 0010 0000

Start Bit: 64

Length: 8

IP PDU > *Protocol* for the selected UDP PDU

Field Name: *Protocol*

Description: Protocol is an 8-bit field that indicates the next level protocol that is used in the data portion of the Internet diagram.

Dec	Hex	Protocol	Dec	Hex	Protocol
0	00	Reserved	22	16	Multiplexing
1	01	ICMP	23	17	DCN
2	02	Unassigned	24	18	TAC Monitoring
3	03	Gateway-to-Gateway	25-76	19-4C	Unassigned
4	04	CMCC Gateway Monitoring Message	77	4D	Any local network
5	05	ST	100	64	SATNET and Backroom EXPAK
6	06	TCP	101	65	MIT Subnet Support
7	07	UCL	102-104	66-68	Unassigned
10	0A	Unassigned	105	69	SATNET Monitoring
11	0B	Secure	106	6A	Unassigned
12	0C	BBN RCC Monitoring	107	6B	Internet Packet Core Utility
13	0D	NVP	110-113	6E-71	Unassigned
14	0E	PUP	114	72	Backroom SATNET Monitoring
15	0F	Pluribus	115	73	Unassigned
16	10	Telnet	116	74	WIDEBAND Monitoring
17	11	XNET	117	75	WIDEBAND EXPAK
20	14	Chaos	120-376	78-0178	Unassigned
21	15	User Datagram	377	0179	Reserved

Data value (decimal): 17

Data values in other bases:

Hexadecimal	1	1
Binary	0001	0001

Start Bit: 72

Length: 8

RFC Link: http://www.faqs.org/rfcs/rfc790.html

IP PDU > *Header Checksum* for the selected UDP PDU

Field Name: *Header Checksum*

Description: The Header Checksum is a 16-bit field. The Checksum is the 16-bit one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the initial value of its field is zero. When both header checksums are equal, then the header bits are correct. If either checksums vary, then a new, correct packet will need to be sent.

This is a simple way to compute the checksum and experimental evidence indicates that it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Data value (decimal): 16199

Data values in other bases:

Hexadecimal	3	F	4	7
Binary	0011	1111	0100	0111

Start Bit: 80

Length: 16

IP PDU > *Source IP Address* for the selected UDP PDU

Field Name: *Source IP Address*

Description: The Source Address is a 32-bit field that contains the IP address of the host that sent the IP Packet.
The source address of the sender of the IP data gram:
NET ID ADDRESS RANGE
000-127 Class A 10.0.0.0-10.225.225
128-191 Class B 172.16.0.0-172.31.255.255
192-223 Class C 192.168.0.0-192.168.255.255
224-239 Class D (multicast)
240-255 Class E (experimental)
HOST ID
0 Network value; broadcast(old)
255 Broadcast

Data value (decimal): 192.168.0.71

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	4	7
Binary	1100	0000	1010	1000	0000	0000	0100	0111

Start Bit: 96

Length: 32

IP PDU > *Destination IP Address* for the selected UDP PDU

Field Name: *Destination IP Address*

Description: The Destination Address is a 32-bit field that contains the address of the host that is to receive the data contained within the IP packet.
The IP address of the host where this data gram is being sent:
NET ID ADDRESS RANGE
000-127 Class A 10.0.0.0-10.225.225
128-191 Class B 172.16.0.0-172.31.255.255
192-223 Class C 192.168.0.0-192.168.255.255
224-239 Class D (multicast)
240-255 Class E (experimental)
HOST ID
0 Network value; broadcast(old)
255 Broadcast

Data value (decimal): 192.168.0.255

Data values in other bases:

Hexadecimal	C	0	A	8	0	0	F	F
Binary	1100	0000	1010	1000	0000	0000	1111	1111

Start Bit: 128

Length: 32 bits

IP PDU > *Options* for the selected UDP PDU

Field Name: *Options*

Description: The options may or may not appear in Ethernet packets. They have to be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular packet, not their implementation.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option.

Case 1: A single octet of option type

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

Data values: *Not applicable*

Data values in other bases: *Not applicable*

Start Bit: 160

Length: Variable (0-40 bits)

IP PDU > *Data* for the selected UDP PDU

<u>Description</u>: The Data is a variable length field, which contains the actual data that is being sent from one host to another. The data field may start with a Layer 4 header, which will give additional instructions to the application that will be receiving the data; alternately, it may be an ICMP header and not contain any user data at all.

<u>Data values (hexadecimal)</u>: (UDP) FF FF FF FF FF FF 08 00 69 07 9A C5 08 00 45 10 01 28 BB D7 00 00 3C 11 3F 47 C0 A8 00 47 C0 A8 00 FF 02 0D 01 14 E9 DB 18 01 C2 5A 0A FF 13 D0 0 00 00 00 00 69 6E 64 79 00 00 00 00 7F FF 11 DB 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 7F FF 13 F0 00 00 00 00 7F FF 11 F6 00 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 7F FF 17 E3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 09 00 00 00 10 05 9D D8 00 00 00 00 10 00 DB A8 00 00 00 00 0F B5 B1 74 00 00 00 00 0F AB 01 C8 00 00 00 00 10 05 9D F8 00 00 00 00 0F B5 9D D8 00 00 00 00 10 05 9D FC 00 00 00 00 0F B5 B1 74 00 00 00 00 10 05 9B 18 00 00 00 00 0F AB 02 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00

<u>Data values in other bases</u>:
Not Applicable

<u>Start Bit</u>: 0

<u>Length</u>: 268

IP PDU > *Source Port Number* for the selected UDP PDU

Field Name: *Source Port Number*

Description: Source Port is an optional field, when meaningful, it indicates the port of the sending process, and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.

Data value (decimal): 525

Data values in other bases:

Hexadecimal	02	0D
Binary	0010	1101

Start Bit: 0

Length: 16 bits

RFC Link: http://www.ietf.org/rfc/rfc0768.txt?number=768

IP PDU > *Destination Port Number* for the selected UDP PDU

Field Name: *Destination Port Number*

Description: Destination Port has a meaning within the context of a particular Internet destination address.

Data value (decimal): 525

Data values in other bases:

Hexadecimal	02	0D
Binary	0010	1101

Start Bit: 16

Length: 16 bits

IP PDU > *Length* for the selected UDP PDU

Field Name: *Length*

Description: Length is the length in octets of this user data gram including this header and the data (This means the minimum value of the length is eight).

Data value (decimal): 276

Data values in other bases:

Hexadecimal	01	14
Binary	0000 0001	0001 0100

- The Binary sequence used here is that of Blue Technology's data. Blue's data matches up with the dump file from ethereal while Mirage's doesn't.
- Mirage's binary sequence was 0000 0001 0000 0100

Start Bit: 32

Length: 16 bits

IP PDU > *Checksum* for the selected UDP PDU

Field Name: *Checksum*

Description: Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Data value (decimal): 59867

Data values in other bases:

Hexadecimal	E	9	D	B
Binary	1110	1001	1101	1011

Start Bit: 48

Length: 16 bits

IP PDU > *Data* for the selected UDP PDU

Field Name: *Data*

Description: Source Port is an optional field, when meaningful, it indicates the port of the sending process, and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.

Data value (hexadecimal): 18 01 C2 5A 0A FF 13 D0 00 00 00 00 69 6E 64 79 00 00 00 00 7F FF 11 DB 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 7F FF 13 F0 00 00 00 00 7F FF 11 F6 00 00 00 00 00 00 00 00 00 00 00 7F FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 7F FF 17 E3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 09 00 00 00 10 05 9D D8 00 00 00 00 10 00 DB A8 00 00 00 00 0F B5 B1 74 00 00 00 00 0F AB 01 C8 00 00 00 00 10 05 9D F8 00 00 00 00 0F B5 9D D8 00 00 00 00 10 05 9D FC 00 00 00 00 0F B5 B1 74 00 00 00 00 10 05 9B 18 00 00 00 00 0F AB 02 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Data values in other bases: *Not Applicable*

Start Bit: 0

Length: 268 bits

Section 3 Data Storage

Section 3.1 Naming convention

The data session files will have a naming convention that will distinguish the type of the file and the chosen name to represent the file. The type of file will end in the *.dat format. The date does not have to be given in the file as Paradigm will be writing a script to find it using Perl as the scripting language. All files will be stored in the same directory named Files, as they will be sorted by type through the Packet Descriptor. An example of an ftp packet using the file name convention is ftp_file01.dat.

Section 3.2 Physical location of data

The files will be stored and maintained on Siena's Oraserv server in the "/home/csis4100405/EtherealSessions" directory. The files in this directory will have -rw-r--r-- permissions while the directory will have drwxrwxr-x permissions.

Section 4 Testing Requirements

During the unit and functional testing phase, Black Box testing will be used. With Black Box testing, possible inputs are inserted into the program, and a successful test includes observing expected outputs. For functionality testing, there will be no need to look at what is happening inside the program itself. For this testing phase, the testing subset of the team will test every facet of the program.

Section 4.1 Testing Specifications

- The web site will contain four screens:
    1. Protocol Selector Screen
    2. Packet Selector Screen
    3. Information Display Screen
    4. History Page

The list of "active" protocols (those that the user can select) are as follows:
1. SNMP
2. FTP
3. SMTP
4. HTTP
5. PING
6. TELNET
7. ARP
8. SSH

The list of "inactive" protocols (those that show up in the protocol hierarchy, but cannot be chosen for viewing) are as follows:
1. SCP
2. DHCP
3. DNS
4. RSVP
5. LDAP
6. NTP

The protocol hierarchy display will be available at any time while the web site is being accessed. When the hierarchy is accessed, a picture will appear with the hierarchy and active links. Once a selection is made, a drop down menu will appear to allow the user to select an Ethereal data session.

- When the user first accesses the TCP/IP Descriptor, this is the first screen he or she will see.
- Within the protocol hierarchy display, a picture of the TCP/IP and Open Systems Interconnection (OSI) layers will be displayed alongside the protocol hierarchy, with an emphasis on which protocols reside within each layer.
- There will be a button that will link the user to the History Page. This page will display all groups that have worked on the project, both past and present.
- Selecting a protocol will cause a drop down menu containing all ethereal data sessions in the /home/csis4100405/EtherealSessions folder of Oraserv.
- Selecting one of those data sessions will move the user to the Packet Selector screen.

- Selecting a protocol will cause a picture of that protocol, and any lower level protocols, to be displayed in the Protocol Display Zone of the TCP/IP Packet Descriptor.

The Packet Selector of the TCP/IP Packet Descriptor will display the selected data session's packets. Only the packets of the selected protocol will be displayed for the user to select.

- If there are more packets than our displaying window will allow the user will be able to scroll down and highlight a different packet.
- There will be a button that will link the user to the History Page. This page will display all groups that have worked on the project, both past and present.
- The user is able to return to the Protocol Selector Screen by clicking on the "Choose Protocol" button.
- To select a highlighted packet, the user will be able to either double-click the packet, or click the "View Packet" button.
- Selecting a packet will bring the user to the Information Display window where a picture of that protocol, and any lower level protocols, will be displayed.

The Information Display Screen of the TCP/IP Packet Descriptor will display the selected packet in the selected protocol, with a breakdown of the protocol's appropriate fields, in a colorful and informative matter.

- The user is able to return to the Protocol Selector Screen by clicking on the "Choose Protocol" button.
- The user is able to return to the Packet Selector Screen by clicking on the "Choose Packet" button.
- There will be a button that will link the user to the History Page. This page will display all groups that have worked on the project, both past and present.
- Each PDU picture will be broken up into its component fields. Bit and octet positions will be shown.
- Each picture of each protocol will show the Request for Comments (RFC) number. The number will be a link to a homepage containing comprehensive information about that protocol.
- There will be a protocol stack displayed in the upper right section of the Information Display Screen, allowing the user to be able to switch between the selected protocol and any of the lower level protocols.
- In each picture, field names and protocol data will be displayed.
- Selecting a field in one of the displayed units will cause information about that field to be displayed in the Information Box on the left side of the screen.

The Information Box, which is part of the Information Display Screen, will display field specific information for the various PDUs.

- When a field is selected, that field will be highlighted, and a picture of that field, with the contained data and bit positions, will be displayed.
- Along with the picture, information about the selected field will also be displayed.
- The information will be displayed in all appropriate radices.

The History Page will contain information about groups, both past and present, that have worked on the TCP/IP Descriptor.

- This page will contain links to every group's home page as well as to the Software Engineering website, and back to the TCP/IP Packed Descriptor.
- This page will also contain a description of the project as given to us by Mr. Ken Swarner.

Other Requirements:

- The Descriptor Home Page will be viewable in Microsoft Internet Explorer and Netscape Navigator in full screen mode at 1024 x 768 screen resolution.

Section 4.2 Testing Forms

Unit Testing Form

What being tested	Tested for	Expected Outcome	Pass or Fail	Comments
Protocol Selector Screen (PrSS)	Does it load?	The PrSS page loads when access is attempted		
PrSS	Is the protocol tree displayed correctly?	The connecting lines on the protocol tree connected are correctly and contains both active and inactive protocols		
PrSS	Is the TCP/IP Model present?	The TCP/IP Model appears to the right of the protocol tree		
PrSS	Is the OSI Model present?	The OSI Model appears to the right of the TCP/IP Model		
PrSS	Is the History Button present?	The History Button is on the page on the lower right hand corner		
PrSS Active Protocols*	Does drop down appear when clicked?	When the user clicks an active protocol the drop down menu containing a list of Ethereal data sessions is displayed		
PrSS Drop Down Menu*	Does drop down contain all the correct files named correctly?	The names of the Ethereal data sessions are correct		
PrSS Active Protocol Buttons*	Correct box color?	The color of the active protocol		

		boxes are consistent and different from the inactive protocol boxes		
PrSS Active Protocol Buttons*	Correct font color?	The color of the active protocol font is readable, consistent, and different from that of the inactive protocol font		
PrSS Inactive Protocols*	Correct box color?	The color of the inactive protocol boxes are consistent and different from the active protocol boxes		
PrSS Inactive Protocols*	Correct font color?	The color of the inactive protocol font is readable, consistent, and different from that of the active protocol font		
PrSS Information Box	Is it present?	The Information Box appears below the protocol screen when the PrSS loads		
PrSS Information Box*	Does it display the correct information when you select a protocol?	When a protocol is selected, information about that protocol appears in the information box		
PrSS	Is the RFC link present?	The RFC link is present in the Ethernet Frame at the top of the screen		
PrSS	Does the RFC	The RFC link		

	link bring you to the correct site?	brings the user to the appropriate RFC page		
PrSS	Does the link to the History page work?	Upon clicking the history page button we are brought to the History Page		
PrSS	Does the link to the History Page appear?	The History Page button appears on the bottom right of the screen		
Packet Selector Screen (PaSS)	Does it load?	The PaSS page loads once a protocol and data session are selected		
PaSS	Is the data session parsed correctly?	The packets for the data session are displayed properly		
PaSS	Is the default packet highlighted upon load?	The first packet in the list of packets is highlighted upon page load		
PaSS	Is the box scrollable if there is more than a page worth of packets in the data session?	If more than a page of packets is in the selected session, a scroll bar allows the user to go through the entire list		
PaSS*	Is the title correct for the selected protocol?	The title of the selected protocol appears above the packet selection box in the center of the page		
PaSS	Are we allowed to highlight different packets?	The user can change the highlighted packet from the default to others		

PaSS	Does the link to the History page work?	Upon clicking the history page button we are brought to the History Page		
PaSS	Does the link to the History Page appear?	The History Page button appears on the bottom right of the screen		
Information Display Screen (IDS)	Does it load?	The IDS page loads once a packet is selected		
IDS*	Do the correct number of fields show up in their correct positions and lengths?	All the fields for the selected protocol show up and are the proper length and in their correct positions		
IDS*	Does the top bar load correctly?	Upon page load does the Ethernet Bar load with the proper information		
IDS*	Does the link to the History Page appear?	Upon clicking the history page button we are brought to the History Page		
IDS*	Does the link to the History page work?	The History Page button appears on the bottom right of the screen		
IDS*	Does the protocol stack for the selected protocol and its lower level protocols display properly?	The selected protocol and its lower level protocols are displayed in a stack to the right of the Ethernet Bar		
IDS*	Is the data in each field displayed correctly?	The data in each field is correct and displayed in a coherent manner		

IDS*	Upon clicking a field, is the information about that field displayed in the Information Box?	When a field is selected, information about it appears in the Information Box		
IDS*	Does the background color of the Information Box match the color of the selected field?	The background of the Information Box matched that of the field it is displaying information about		
IDS*	Does the RFC link point to the correct location?	The RFC link brings the user to more information about the protocol		
IDS*	Can we navigate between the protocols using the protocol stack?	Clicking the protocol stack allows the user to change view the lower level protocols or return to the starting protocol		
History Page	Does everything appear correctly?	The project description, title, and links all appear in a neat and coherent manner		
History Page	Do the links work?	All the links bring the user to the location that that say they do		

* **Note: For the steps that require testing more than one object, separate sub-test forms will be created. These sub-test forms will be modeled to encompass each of the separate protocol's unique attributes.**

Functional Testing Form

What being tested	Tested for	Expected Outcome	Pass or Fail	Comments
PrSS	Upon click of a data session from drop down menu does it go to the PaSS?	When a data session from the drop down menu is clicked, it brings the user to the PaSS		
PaSS	Is the correct protocol from the PrSS displayed at the top of the screen?	The title of the selected protocol appears above the packet selection box in the center of the page		
PaSS	Are the packets displayed the correct ones for the data session chosen in PrSS?	The packets displayed in the PaSS are for the selected data session and protocol		
PaSS	Does Choose Protocol button take us to PrSS?	Pressing the Choose Protocol button brings the user back to the PrSS		
PaSS	Does double clicking a packet take us to IDS?	Double clicking a packet brings the user to the IDS		
PaSS	Does the View Packet button take us to IDS?	Highlighting a packet and pressing the View Packet button brings the user to the IDS		
IDS	Is the correct packet displayed as given to	The packet whose		

	us by the PaSS? (by double clicking or View Packet button)	information is displayed on the IDS is the packet that the user selected		
IDS	Does Choose Protocol button take us to PrSS?	Pressing the Choose Protocol button brings the user back to the PrSS		
IDS	Does Choose Packet button take us to PaSS? (check to make sure still in same data session)	Pressing the Choose Packer button brings the user back to the PaSS for the selected data session and protocol		
History Page Button*	Does clicking the History Page button take us there?	Upon clicking the History Page button the user is brought to the History Page		
History Page	Does the Back link on the History Page bring us back to where we left off?	The Back link brings the user back to the point from which the History Button was selected		
History Page	Does the Blue Technologies link on the History Page bring us to their website?	Clicking the Blue Technologies link brings us to their website		
History Page	Does the Mirage Inc. link on the History Page bring us to their website?	Clicking the Mirage Inc. link brings us to their website		
History Page	Does the Paradigm Solutions link on the History Page bring us to their website?	Clicking the Paradigm Solutions link brings us to their website		
History Page	Does the Software Engineering link on	Clicking the Software		

	the History Page bring us to the Software Engineering webpage?	Engineering link brings us to the Software Engineering webpage		

* **Note:** **For the steps that require testing more than one object, separate sub-test forms will be created. These sub-test forms will be modeled to encompass each of the separate protocol's unique attributes.**

5.0 Appendix

5.1 Glossary

Gantt Chart - A chart that depicts progress in relation to time, often used in planning and tracking a project.

GUI - (Graphical user interface) An interface, which uses text boxes and buttons to allow easy access of information by use of a mouse or other pointing device.

HTML – (Hypertext markup language) A markup language used to structure text and multimedia documents and to set up hypertext links between documents, used extensively on the World Wide Web.

Internet - An interconnected system of networks that connects computers around the world via the TCP/IP protocol.

Linear Sequential Model / Classic Waterfall Model – A systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and support.

Macromedia Dreamweaver MX 2004 - Website development environment. (http://www.macromedia.com)

Open Systems Interconnection Reference Model - A model of network architecture and a suite of protocols (a protocol stack) to implement it, developed by the International Organization for Standardization (ISO) in 1978 as a framework for international standards in heterogeneous computer network architecture.

PDU- (Protocol Data Unit) Information that is delivered as a unit among peer entities of a network that may contain control information, address information, and/or data.

PHP – (PHP: Hypertext Preprocessor) A server-side embedded scripting language. The PHP commands, which are embedded in the web page's HTML, are executed on the web server to generate dynamic HTML pages.

RFC- (Request for Comments) number. A formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties.

Software - Written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory.

TCP/IP – (Transmission Control Protocol / Internet Protocol) The suite of communications protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP. TCP/IP is built into the UNIX operating system and is used by the Internet, making it the primary standard for transmitting data over networks.

5.2 Gantt Chart