

TCP/IP Packet Descriptor Software Requirements Specification

**Welcome
Mr. Ken Swarner**

EdgeTech Members

- | | | |
|-------------------------|---|---------------------|
| Matt DeCrescente | - | Team Leader |
| Jill Foster | - | Librarian |
| Eric Fish | - | System Admin |
| John Mooney | - | Consultant |
| Das Nobel | - | Webmaster |

Software Requirements Specification

Roger Bacon Room 328

November 5, 2004

- Matt DeCrescente Introduction & Conclusion
- Eric Fish Functional Requirements
- John Mooney Data Flow Diagrams & Prototypes

Overview

- **Introduction**
- Functional Requirements
- Data Flow Diagrams
- Prototypes
- Summary

Overview

- Introduction
- **Functional Requirements**
- Data Flow Diagrams
- Prototypes
- Summary

Functional Requirements

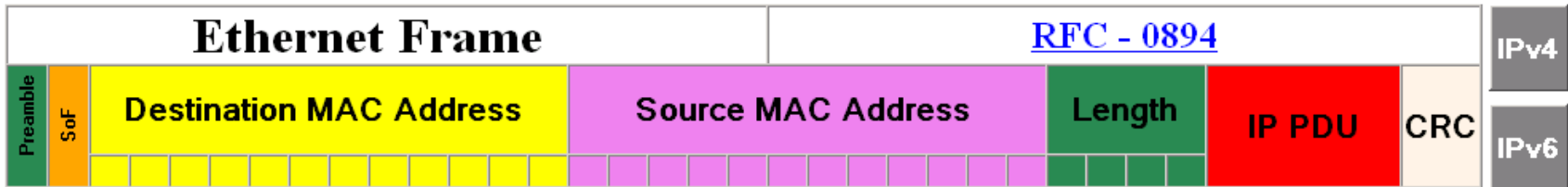
Previously completed by Blue Technologies & Mirage Inc.

- TCP/IP Packet Descriptor interface is web-based and describes a packet
- Portions of packet are displayed and labeled in diagram form
- Information in the TCP frame fields of the packet can be displayed in binary, octal, or hexadecimal form
- Displays each field of a packet and gives information about each field
- Handles different amounts of packets, each at varying lengths

Functional Requirements

To be completed by EdgeTech Development

- Clean up the interface by eliminating recently selected protocol list and replacing it with protocol information
- Coordinate all colors in relevant fields and each field's appropriate detailed information
- Screens must be full screen upon loading



Select Packet Type

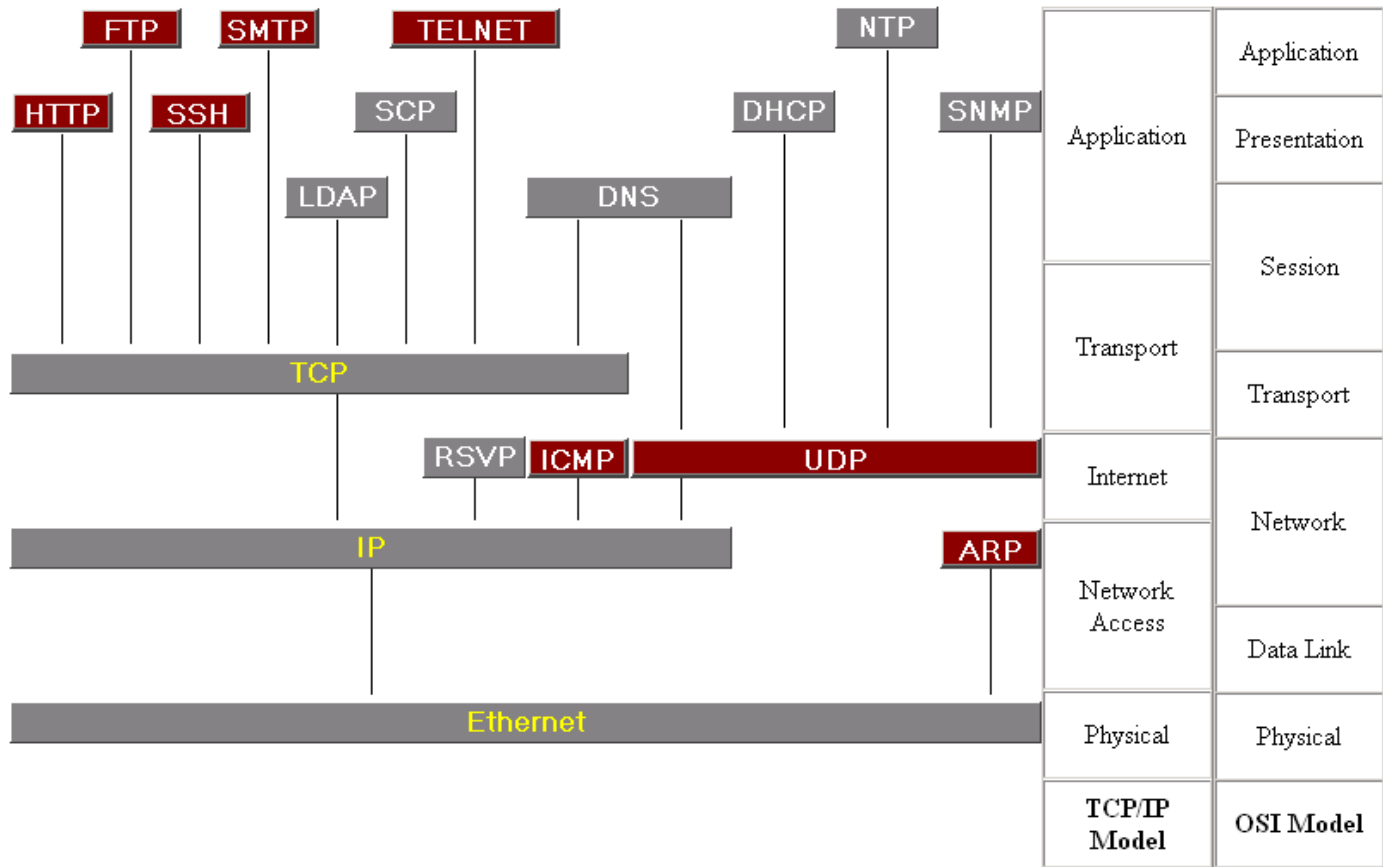
Choose Protocol

Recently Viewed Protocols:

- FTP
- UDP
- ICMP
- TELNET
- ARP



[Blue Technologies, Inc.](#)



Functional Requirements

To be completed by EdgeTech Development

- Allow users to select captured Ethereal sessions. Ethereal sessions will be located in a static directory chosen by client
- List of all available packets will be present within a session
- By selecting a packet, the information will be parsed into the previously chosen protocol

Functional Requirements

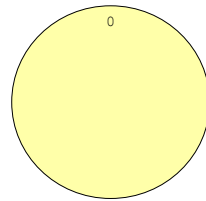
To be completed by EdgeTech Development (cont'd)

- Software must be adaptable
- Supply a button that links the user to a history EdgeTech and the TCP/IP Packet Descriptor, including Mirage Inc. and Blue Technology's websites, and the previous versions of the software
- Allow future programmers to implement a Cyclic Redundancy Check (CRC), a dynamic directory selector, and provide a demo

Overview

- Introduction
- Functional Requirements
- **Data Flow Diagrams**
- Prototypes
- Summary

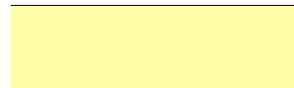
Data Flow Diagram



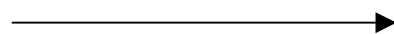
Process



Source/Sink

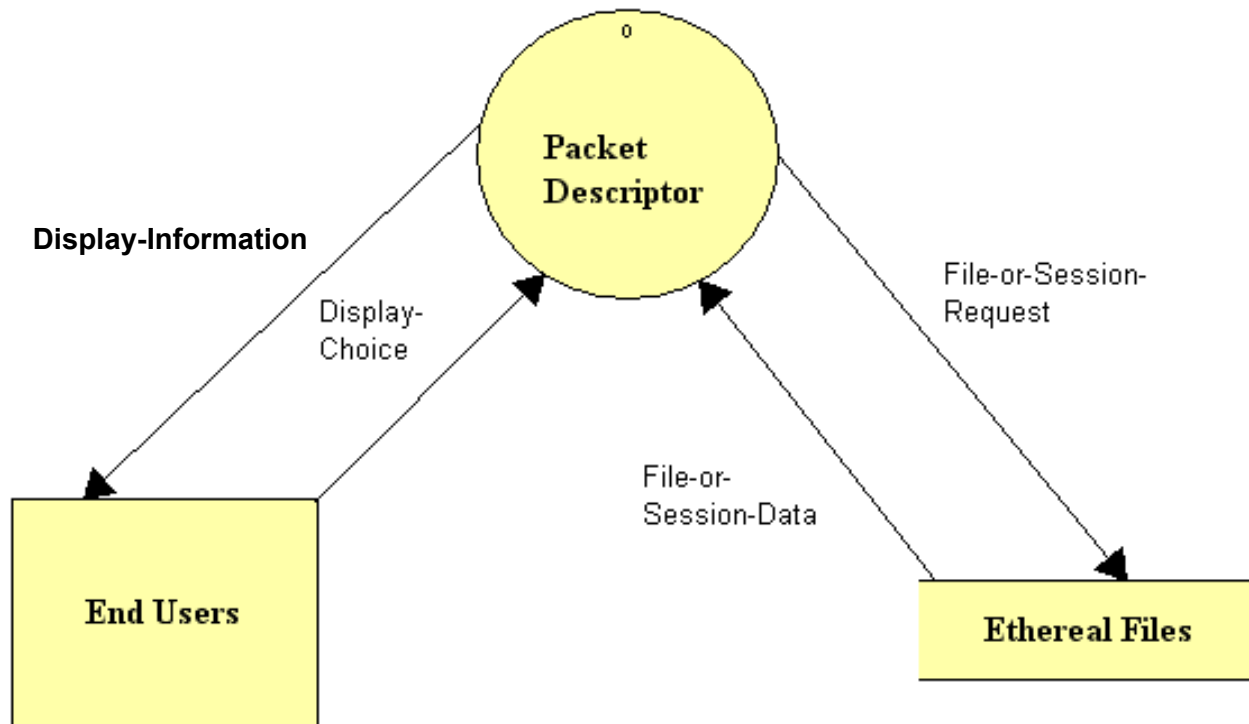


File

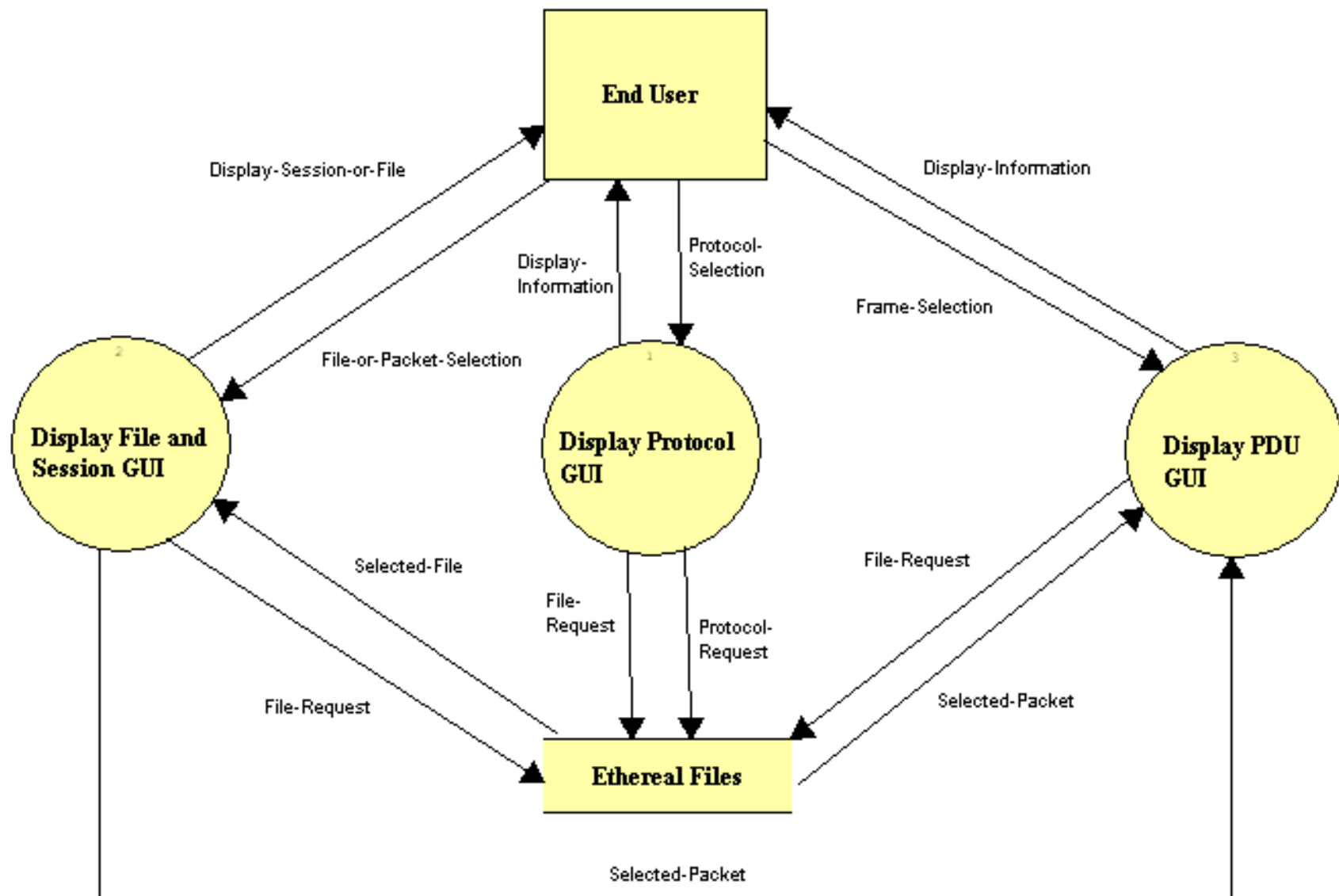


Data Flow

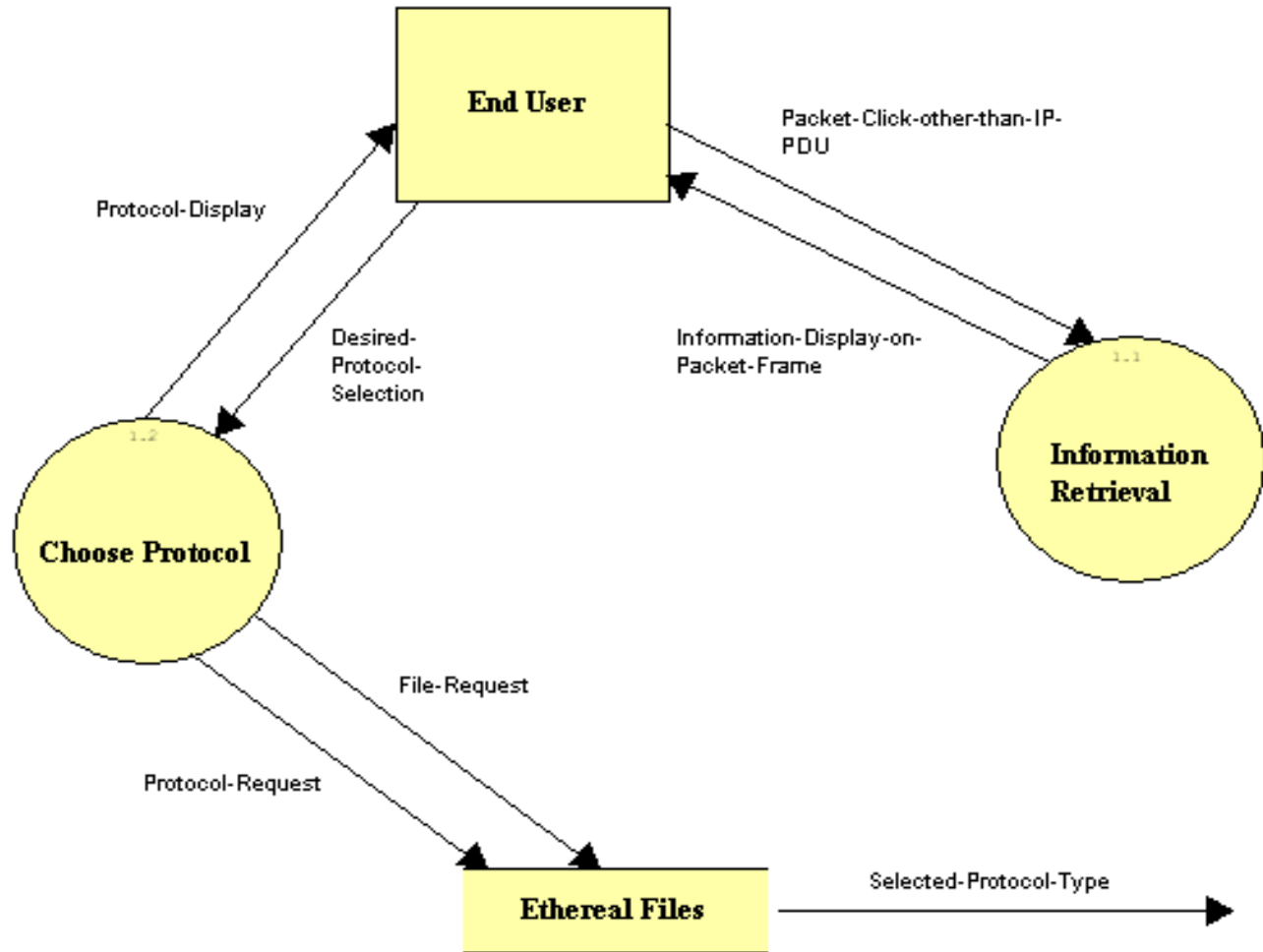
Context Diagram



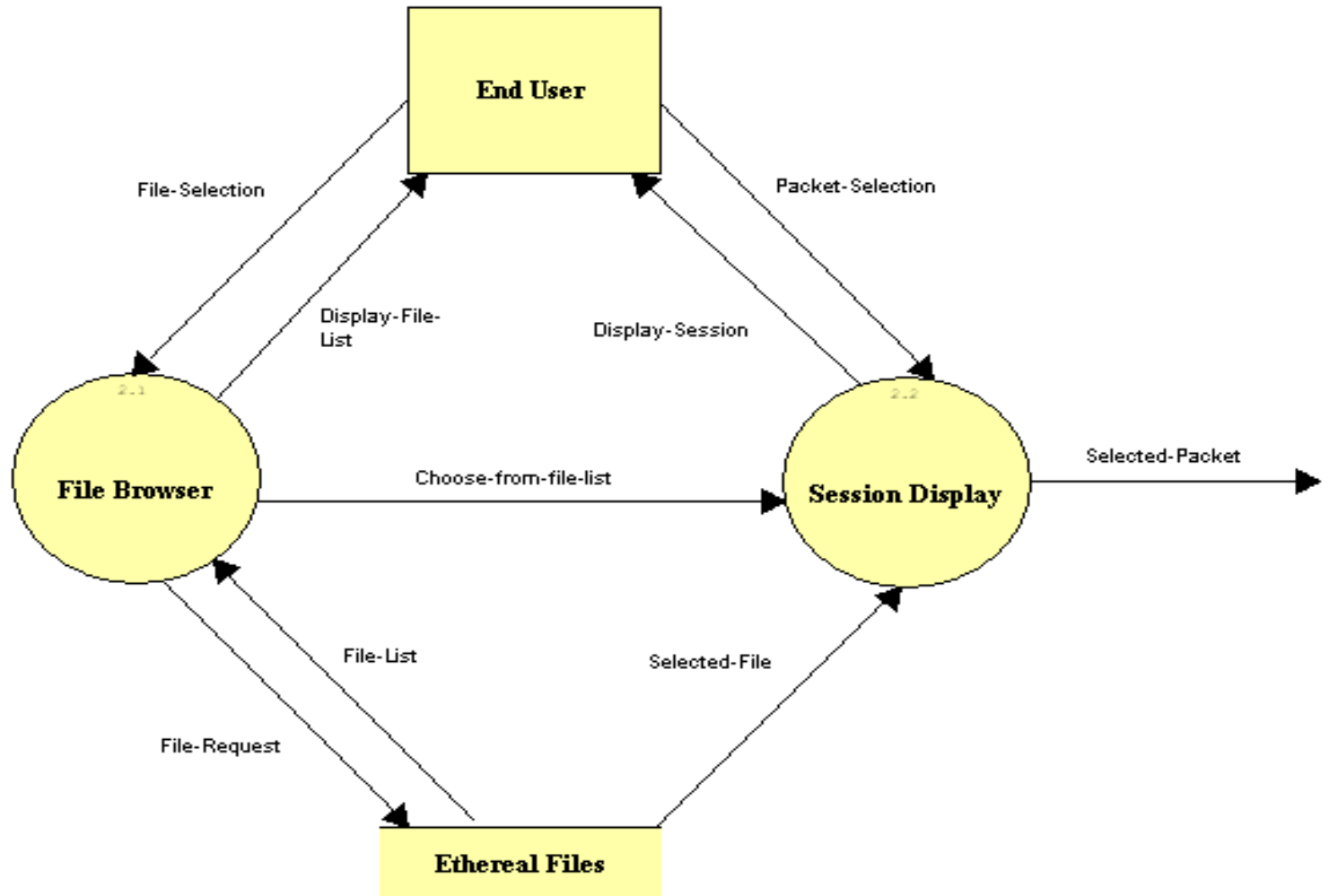
Level 0



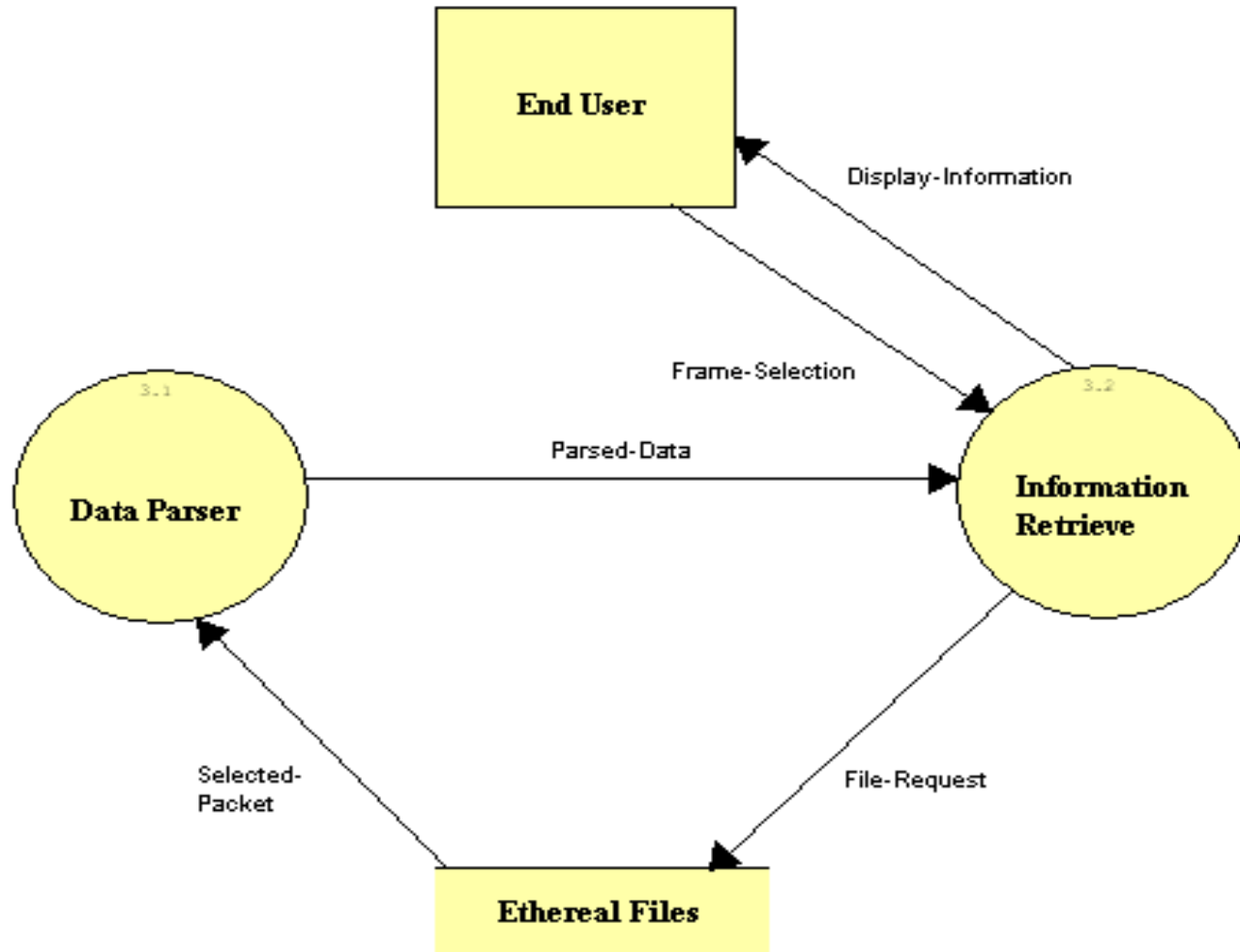
Level 1
Display Protocol GUI
(1)



Level 1
Display File and Session GUI
(2)



Level 1
Display PDU GUI
(3)



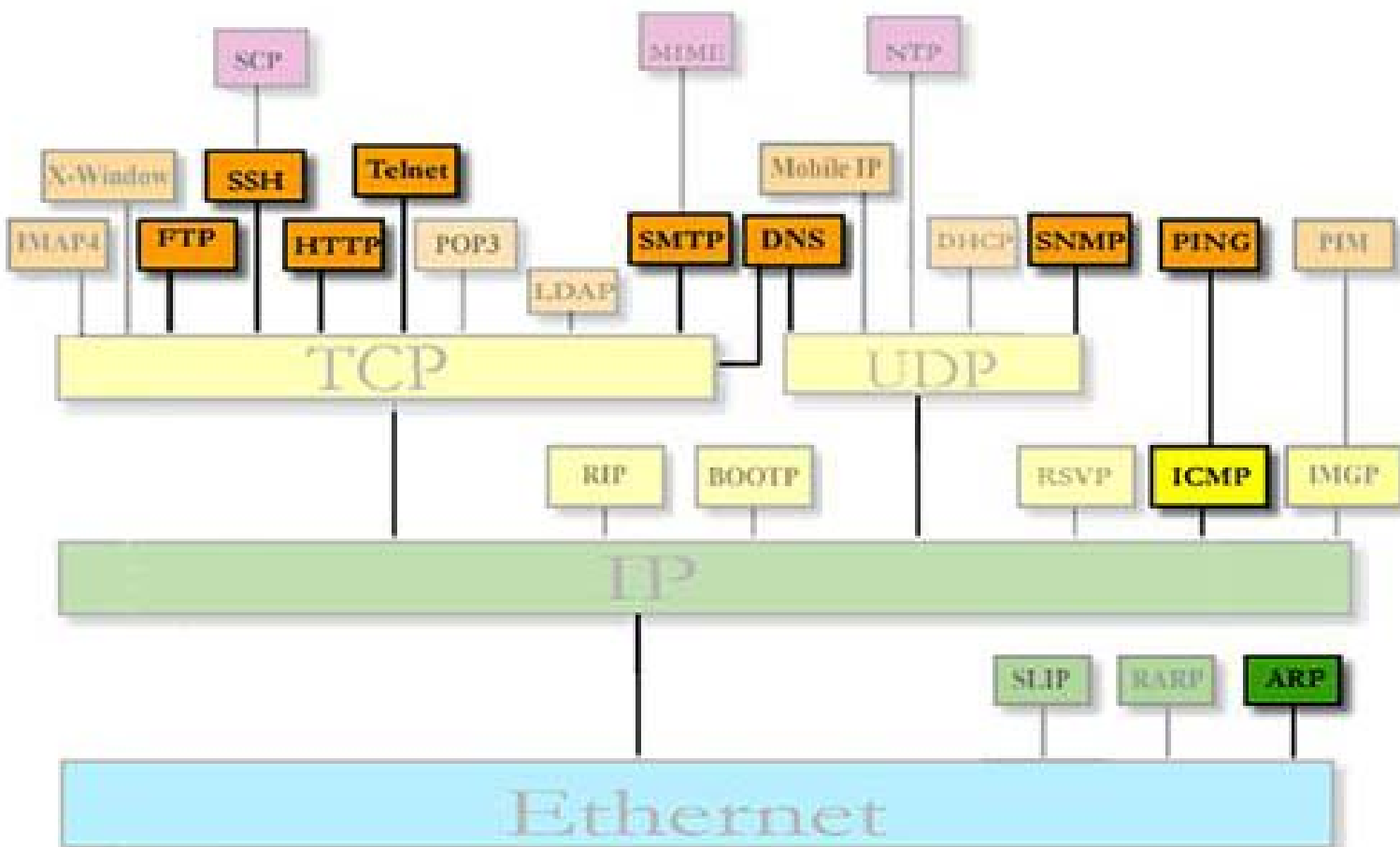
Overview

- Introduction
- Functional Requirements
- Data Flow Diagrams
- **Prototypes**
- Summary

Ethernet Packet

IPv4 IPv6

Choose A Protocol From Diagram Below



Application	Application
	Presentation
	Session
Transport	Transport
	Network
Network Access	Data Link
	Physical

TCP/IP Model OSI Model

IPv4 IPv6

Choose Protocol

Ethernet Packet



FTP

Select A Packet From The Right Or Open A New Captured Session Below

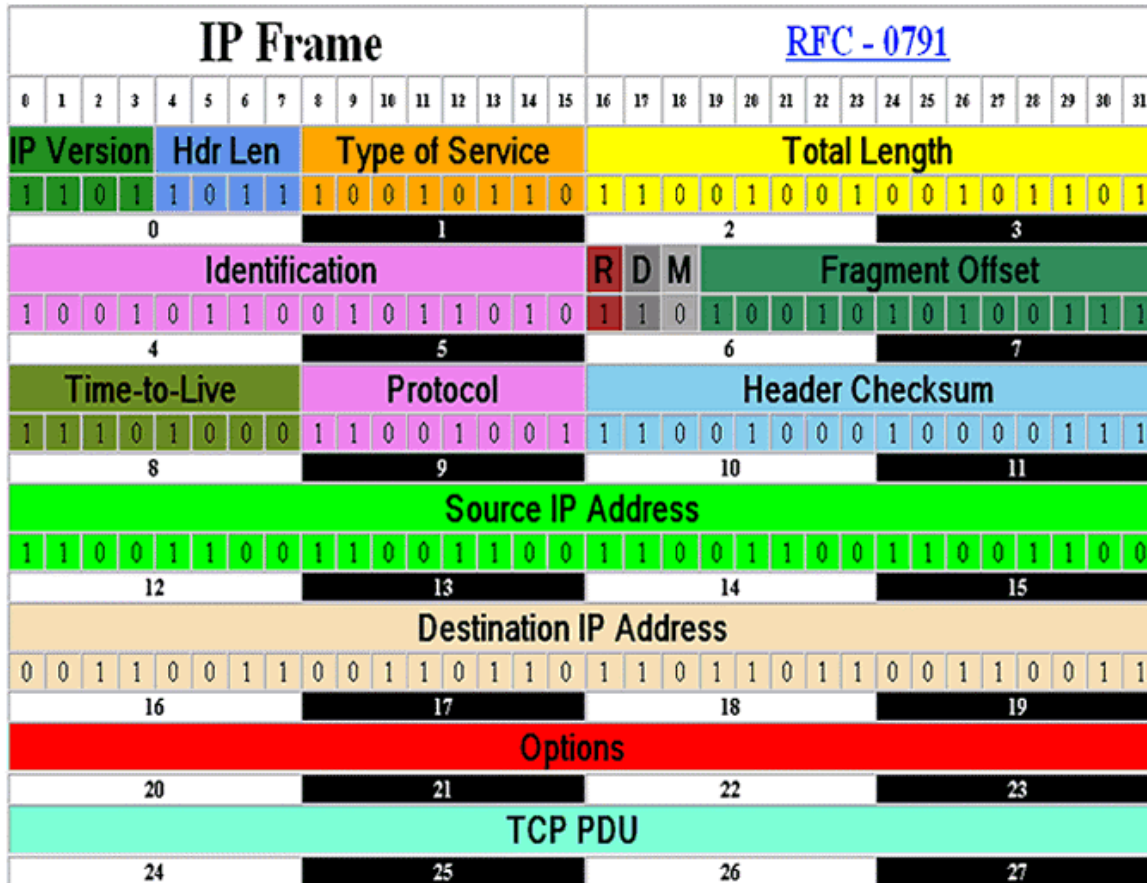
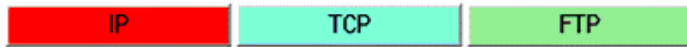
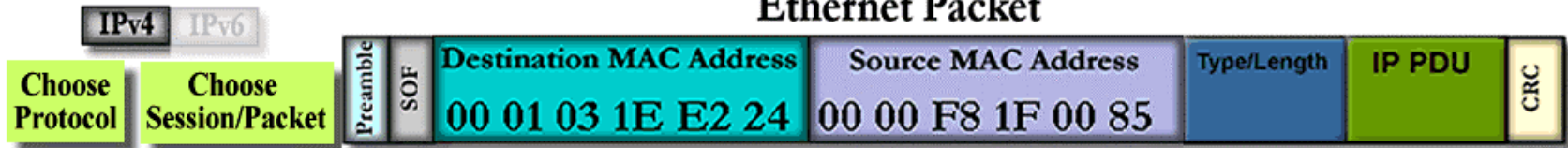
Directory:

/usr/local/EthernetDumps

- arpDump10.dat
- ftpDump.dat
- ftpDumpPut.dat
- http168Dump.dat
- icmpDump1.dat
- mimeDump.dat
- pingDump.dat

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.39	192.168.0.101	TCP	32816 > ftp [SYN] Seq=0 Ack=0 win=!
2	0.000154	192.168.0.101	192.168.0.39	TCP	ftp > 32816 [SYN, ACK] Seq=0 Ack=1
3	0.000401	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=1 Ack=1 win=!
4	0.013027	192.168.0.101	192.168.0.39	FTP	Response: 220 cb118ks.cs.siena.edu
5	0.013375	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=1 Ack=95 win=!
6	6.676401	192.168.0.39	192.168.0.101	FTP	Request: USER fakeuser
7	6.676429	192.168.0.101	192.168.0.39	TCP	ftp > 32816 [ACK] Seq=95 Ack=16 win=!
8	6.677232	192.168.0.101	192.168.0.39	FTP	Response: 331 Password required for
9	6.677417	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=16 Ack=132 win=!
10	13.813892	192.168.0.39	192.168.0.101	FTP	Request: PASS fla2k3user
11	13.827680	192.168.0.101	192.168.0.39	FTP	Response: 230 User fakeuser logged
12	13.827905	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=33 Ack=162 win=!
13	13.828369	192.168.0.39	192.168.0.101	FTP	Request: SYST
14	13.828878	192.168.0.101	192.168.0.39	FTP	Response: 215 UNIX Type: L8
15	13.868033	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=39 Ack=181 win=!
16	15.964049	192.168.0.39	192.168.0.101	FTP	Request: TYPE I
17	15.964227	192.168.0.101	192.168.0.39	FTP	Response: 200 Type set to I.
18	15.964440	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=47 Ack=201 win=!
19	21.044925	192.168.0.39	192.168.0.101	FTP	Request: PASV
20	21.046043	192.168.0.101	192.168.0.39	FTP	Response: 227 Entering Passive Mode
21	21.046293	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=53 Ack=249 win=!
22	21.047403	192.168.0.39	192.168.0.101	FTP	Request: STOR testfile.dat
23	21.060328	192.168.0.101	192.168.0.39	FTP	Response: 150 opening BINARY mode
24	21.099489	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=72 Ack=308 win=!
25	21.099568	192.168.0.101	192.168.0.39	FTP	Response: 226 Transfer complete.
26	21.099738	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=72 Ack=332 win=!
27	23.631322	192.168.0.39	192.168.0.101	FTP	Request: QUIT
28	23.631433	192.168.0.101	192.168.0.39	FTP	Response: 221-You have transferred
29	23.631752	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=78 Ack=379 win=!
30	23.631769	192.168.0.101	192.168.0.39	FTP	Response: 221-Total traffic for th
31	23.631983	192.168.0.39	192.168.0.101	TCP	32816 > ftp [ACK] Seq=78 Ack=525 win=!
32	23.632002	192.168.0.101	192.168.0.39	TCP	ftp > 32816 [FIN, ACK] Seq=525 Ack=!
33	23.632348	192.168.0.39	192.168.0.101	TCP	32816 > ftp [FIN, ACK] Seq=78 Ack=!

Ethernet Packet



FTP PDU

IP Frame

Source IP Address Field

Decimal IP Address:

Start bit: 96
Length: 32 bits

T
he source address of the sender of the IP datagram.

NET ID ADDRESS RANGE

- 000-127 Class A 10.0.0.0-10.255.255
- 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)

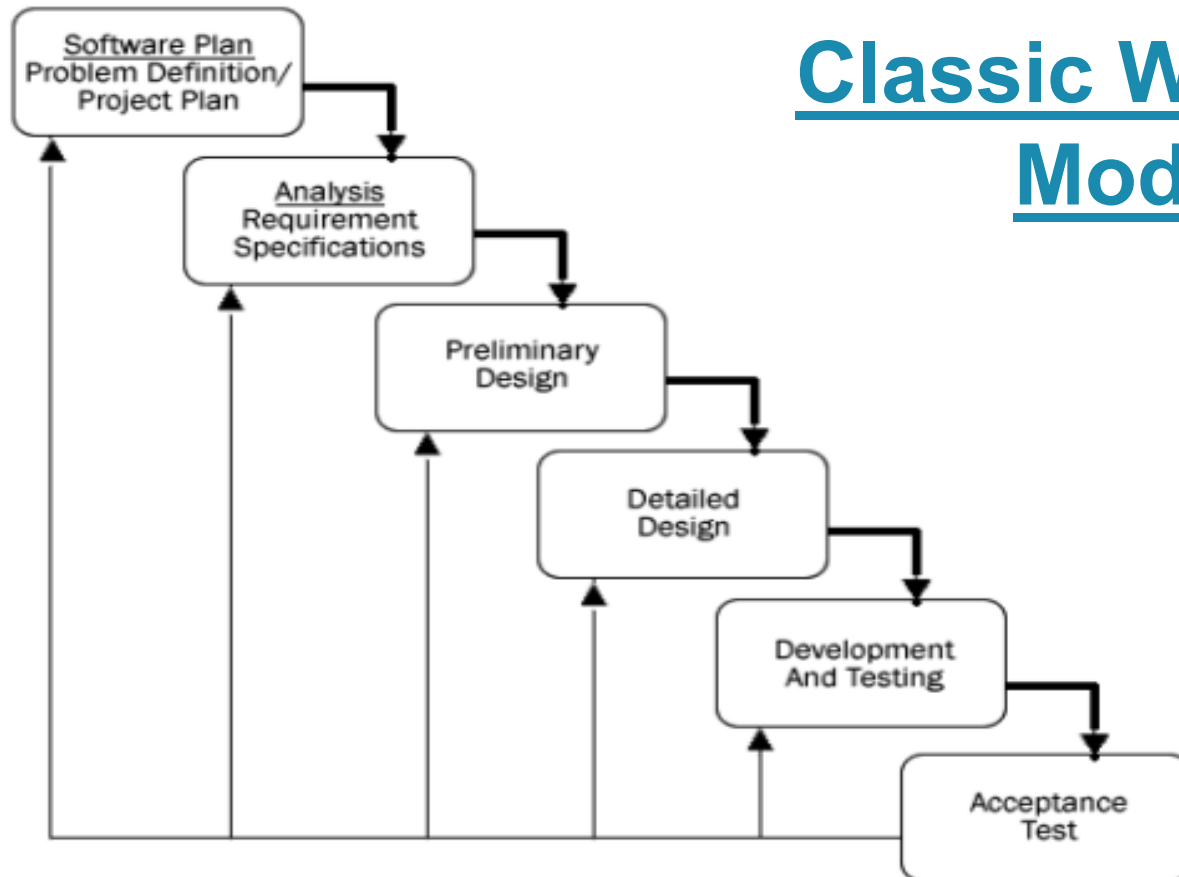
HOST ID

- 0 Network value, broadcast(old)
- 255 Broadcast

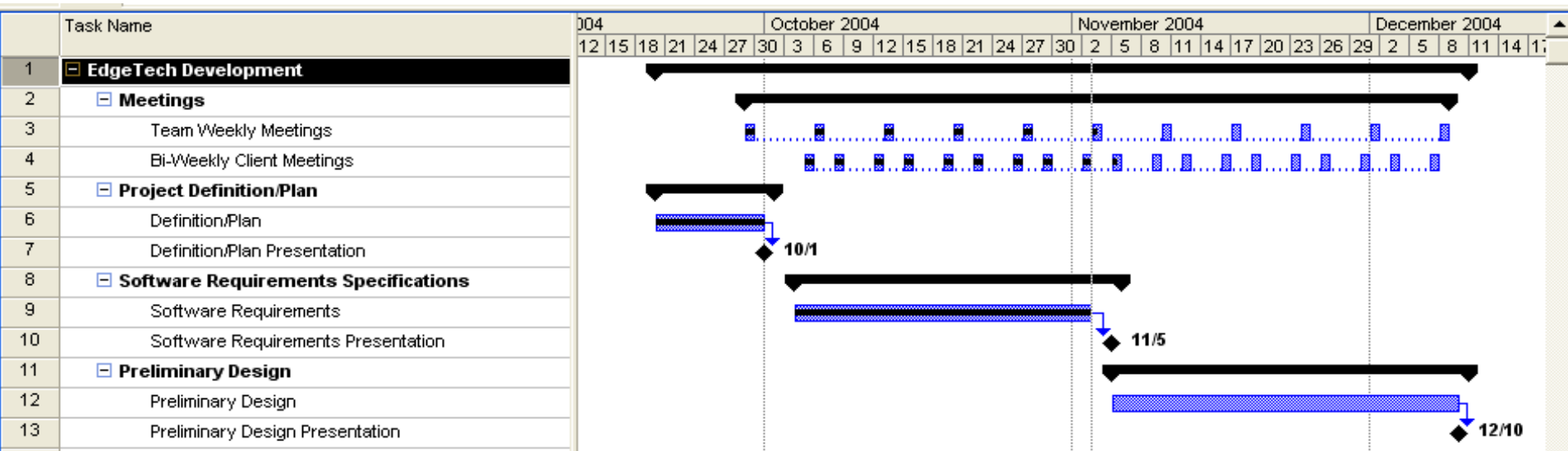
Overview

- Introduction
- Functional Requirements
- Data Flow Diagrams
- Prototypes
- **Summary**

Classic Waterfall Model



Important Dates



- **December 6th, 2004** - Preliminary Design Document due
- **December 10th, 2004 at 8:15 am** - Preliminary Design Presentation

Any Questions?