

CS 716: Introduction to communication networks

- 16th class; 28th Sept 2011

Instructor: Sridhar Iyer

Demo by: Swati Patil

IIT Bombay

What is IP address

An identifier for a computer or device on a TCP/IP network.

Networks using the TCP/IP protocol route messages based on the IP address of the destination.

The format of an **IP address** is a **32-bit** numeric address written as four numbers separated by periods. Each number can be zero to 255.

For example: 10.129.50.94 could be an IP address.

You can assign IP addresses at random as long as each one is unique.

The four numbers in an IP address are used in different ways to identify a particular network and a host on that network.

Classes of IP

There are five classes of available IP ranges:

Class A, Class B, Class C, Class D and Class E, while only A, B and C are commonly used.

**Class A : Supports 16 million hosts on each of 127 networks.
1.0.0.1 to 126.255.255.254**

**Class B : Supports 65,000 hosts on each of 16,00 networks.
128.1.0.1 to 191.255.255.254**

**Class C : Supports 254 hosts on each of 2 million networks.
192.0.1.1 to 223.255.254.254**

**Class D : Reserved for multicast groups.
224.0.0.0 to 239.255.255.255**

**Class E : Reserved for future use, or Research and Development Purposes.
240.0.0.0 to 254.255.255.254**

Network Configuration

GUI : Graphical User Interface

System----Preferences-----Network Settings

i) DHCP : Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer from a defined range of numbers configured for a given network.

ii) Manual

Netmask : The Netmask, together with the IP address, defines the network the computer belongs to, that is which other IP addresses the computer can touch directly in the same LAN.

Gateway: A node on a network that serves as an entrance to another network. In enterprises, the gateway is the computer that routes the traffic from a workstation to the outside network that is serving the Web pages.

Using Terminal

- a) `ifconfig`
- b) `sudo vi /etc/network/interfaces`
- c) `sudo /etc/init.d/networking restart`
- d) `sudo /etc/hostname`
- e) `sudo /etc/host`
- f) `vi /etc/resolv.conf`

PING : COMMAND

Ping is a basic Internet program that allows a user to verify that a particular IP address exists and can accept requests.

Ping works by sending an Internet Control Message Protocol (ICMP) Echo Request to a specified interface on the network and waiting for a reply.

PING PACKET SNIFFING USING WIRESHARK

The image shows the Wireshark network protocol analyzer interface. The title bar reads "Capturing from Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler) - Wireshark". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, and Help. The toolbar contains various icons for file operations, capture control, and analysis. The filter bar is set to "icmp". The main display area shows a list of captured packets, with the following data:

No.	Time	Source	Destination	Protocol	Info
104	3.405239	10.129.12.6	10.129.1.1	ICMP	Echo (ping) request (id=0x0200, seq(be/le)=1536/6, ttl=)
105	3.405503	10.129.1.1	10.129.12.6	ICMP	Echo (ping) reply (id=0x0200, seq(be/le)=1536/6, ttl=)
131	4.400905	10.129.12.6	10.129.1.1	ICMP	Echo (ping) request (id=0x0200, seq(be/le)=1792/7, ttl=)
132	4.401136	10.129.1.1	10.129.12.6	ICMP	Echo (ping) reply (id=0x0200, seq(be/le)=1792/7, ttl=)
163	5.400909	10.129.12.6	10.129.1.1	ICMP	Echo (ping) request (id=0x0200, seq(be/le)=2048/8, ttl=)
164	5.401143	10.129.1.1	10.129.12.6	ICMP	Echo (ping) reply (id=0x0200, seq(be/le)=2048/8, ttl=)
181	6.400907	10.129.12.6	10.129.1.1	ICMP	Echo (ping) request (id=0x0200, seq(be/le)=2304/9, ttl=)
182	6.401157	10.129.1.1	10.129.12.6	ICMP	Echo (ping) reply (id=0x0200, seq(be/le)=2304/9, ttl=)

Below the packet list, the details pane for packet 104 is expanded, showing the following layers:

- Frame 104: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
- Ethernet II, src: Intel_1e:66:82 (00:19:d1:1e:66:82), dst: Intel_b4:40:86 (00:04:23:b4:40:86)
- Internet Protocol, Src: 10.129.12.6 (10.129.12.6), Dst: 10.129.1.1 (10.129.1.1)
- Internet Control Message Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 00 04 23 b4 40 86 00 19 d1 1e 66 82 08 00 45 00  ..#.@... ..f...E.
0010 00 3c 5c 94 00 00 80 01 bc 24 0a 81 0c 06 0a 81  .<\...... .$.....
0020 01 01 08 00 45 5c 02 00 06 00 61 62 63 64 65 66  ....E\.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69                    wabcdefg hi
```

Telnet

Telnet is a protocol that allows you to connect to remote computers over a TCP/IP network.

Command for telnet `telnet <host>`

To create a connection with the remote host. The Telnet client will send a request to the Telnet server (remote host). The server will reply asking for a user name and password. If accepted, the Telnet client will establish a connection to the host and allows you to access the host's computer.

Telnet Packet Sniffing using Wireshark

The screenshot displays the Wireshark interface with the following details:

- Title Bar:** Capturing from Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler) - Wireshark
- Menu Bar:** File Edit View Go Capture Analyze Statistics Telephony Tools Help
- Toolbar:** Standard network analysis icons including capture, stop, refresh, and zoom.
- Filter:** telnet
- Packet List:** A table of captured packets, all identified as TELNET.
- Packet Details:** Hierarchical view for packet 443, showing Ethernet II, Internet Protocol, and Transmission Control Protocol layers.
- Packet Bytes:** Hexadecimal and ASCII representation of the captured data.

No.	Time	Source	Destination	Protocol	Info
661	23.045606	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
670	23.300844	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
674	23.619793	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
703	23.820258	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
705	24.024384	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
709	24.220560	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
714	24.381064	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
722	24.624557	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
728	24.820625	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
730	25.060294	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
736	25.260747	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
739	25.421555	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
743	25.664940	10.129.12.6	10.129.50.190	TELNET	Telnet Data ...
745	25.665279	10.129.50.190	10.129.12.6	TELNET	Telnet Data ...
754	25.962001	10.129.50.190	10.129.12.6	TELNET	Telnet Data ...
760	26.163441	10.129.50.190	10.129.12.6	TELNET	Telnet Data ...

Frame 443: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

- Ethernet II, Src: IntelCor_ed:91:89 (00:1c:c0:ed:91:89), Dst: Intel_1e:66:82 (00:19:d1:1e:66:82)
- Internet Protocol, Src: 10.129.50.190 (10.129.50.190), Dst: 10.129.12.6 (10.129.12.6)
- Transmission Control Protocol, Src Port: telnet (23), Dst Port: 3324 (3324), Seq: 1, Ack: 1, Len: 12
- Telnet

```
0000 00 19 d1 1e 66 82 00 1c c0 ed 91 89 08 00 45 10    ....f... ..E.
0010 00 34 f1 e1 40 00 40 06 f5 0c 0a 81 32 be 0a 81    .4...@.@...2...
0020 0c 06 00 17 0c fc b4 3f 1b 1b d7 fd b7 81 50 18    .....?.....P.
0030 16 d0 a1 fa 00 00 ff fd 18 ff fd 20 ff fd 23 ff    ..... ..#.
0040 fd 27
```

How to send email using Telnet and sense SMTP with Wireshark

Start a Telnet session from a command line by entering:

```
Telnet your.mailserver.com 25
```

```
220 a.mail.server.com Microsoft ESMTP MAIL Service,  
Version: 6.0.3790.2499 ready at Thu, 29 Jun 2006  
15:59:02 -0600
```

```
helo
```

250 a.mail.server.com Hello [192.168.125.237]

mail from: test@test.org

250 2.1.0 email@test.org... Sender OK

rcpt to: test@test.com

250 2.1.5 test@test.com

data

354 Start mail input; end with .

This is a test.

. (enter a dot/period to end the data)

250 2.6.0 Queued mail for delivery

quit

Connection to host lost.

Secure Shell

Secure Shell is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels.

When using ssh, the entire login session, including transmission of password, is encrypted, therefore it is almost impossible for an outsider to collect passwords.

SSH Packet Sniffing using Wireshark

Capturing from Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler) - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: ssh Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
212	11.961740	10.129.50.190	10.129.12.6	SSHv2	Server Protocol: SSH-2.0-OpenSSH_4.7p1 Debian
213	12.080875	10.129.12.6	10.129.50.190	SSHv2	Client Protocol: SSH-1.99-3.2.9 SSH Secure S
215	12.082102	10.129.50.190	10.129.12.6	SSHv2	Server: Key Exchange Init
216	12.088339	10.129.12.6	10.129.50.190	SSHv2	Client: Ignore[Malformed Packet]
220	12.125405	10.129.12.6	10.129.50.190	SSHv2	Client: Diffie-Hellman Key Exchange Init
222	12.128384	10.129.50.190	10.129.12.6	SSHv2	Server: New Keys
223	12.200459	10.129.12.6	10.129.50.190	SSHv2	Client: New Keys
228	12.237389	10.129.12.6	10.129.50.190	TCP	[TCP segment of a reassembled PDU]
230	12.237728	10.129.50.190	10.129.12.6	TCP	[TCP segment of a reassembled PDU]

⊕ Frame 212: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)

⊕ Ethernet II, Src: IntelCor_ed:91:89 (00:1c:c0:ed:91:89), Dst: Intel_1e:66:82 (00:19:d1:1e:66:82)

⊕ Internet Protocol, Src: 10.129.50.190 (10.129.50.190), Dst: 10.129.12.6 (10.129.12.6)

⊕ Transmission Control Protocol, Src Port: ssh (22), Dst Port: cs-remote-db (3630), seq: 1, Ack: 1, Len: 40

⊕ SSH Protocol

Telnet Vs SSH

TELNET, by default, does not encrypt any data sent over the connection including password, and so it is often practical to eavesdrop on the communications and use the password later for malicious purposes.

SSH by default encrypts password and traffic. SSH is recommended for all use.

HTTP

HTTP - the Hypertext Transfer Protocol - provides a standard for Web browsers and servers to communicate.

HTTP is an application layer network protocol built on top of TCP. HTTP clients (such as Web browsers) and servers communicate via HTTP request and response messages.

HTTP utilizes TCP port 80 by default

HTTP Packet Sniffing using Wireshark

The screenshot displays the Wireshark interface with a packet capture of an HTTP GET request. The top bar shows the network interface as 'Realtek 10/100/1000 Ethernet NIC' and the application as '(Microsoft's Packet Scheduler) - Wireshark'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, and Help. The toolbar contains various icons for file operations, search, and capture control. The filter bar is set to 'http'. The main packet list pane shows a series of packets, with packet 54 selected, which is an HTTP GET request to 'http://www.google.com/'. The packet details pane shows the structure of the selected packet: Internet Protocol, Transmission Control Protocol, and Hypertext Transfer Protocol. The Hypertext Transfer Protocol section is expanded to show the request details, including the method (GET), URI, version, and various headers like Accept, Accept-Language, Cookie, and User-Agent. The packet bytes pane at the bottom shows the raw data of the selected packet in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
13	1.209637	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
21	2.044905	fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
41	4.221713	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
54	4.953165	10.129.178.125	10.200.13.50	HTTP	GET http://www.google.com/ HTTP/1.0
58	5.045210	fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
59	5.098288	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 302 Moved Temporarily (text/html)
67	5.101318	10.129.178.125	10.200.13.50	HTTP	GET http://www.google.co.in/ HTTP/1.0
82	5.321579	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 200 OK (text/html)
89	5.328227	10.129.178.125	10.200.13.50	HTTP	GET http://www.google.co.in/gen_204?atyp=i&ghp=fbg HTTP/1.0
96	5.355544	10.129.178.125	10.200.13.50	HTTP	GET http://clients1.google.co.in/generate_204 HTTP/1.0
101	5.367175	10.129.178.125	10.200.13.50	HTTP	GET http://www.google.co.in/cs1?v=3&s=webhp&action=&e=17259,26637,274
104	5.493613	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 204 No Content
109	5.563162	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 204 No Content
151	7.929331	10.129.178.125	10.200.13.50	HTTP	GET http://clients1.google.co.in/complete/search?hl=en&client=hp&expI
153	8.044594	fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
157	8.085491	10.129.178.125	10.200.13.50	HTTP	GET http://clients1.google.co.in/complete/search?hl=en&client=hp&expI
160	8.129736	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 200 OK (text/javascript)
164	8.247816	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
166	8.287803	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 200 OK (text/javascript)
174	8.402151	10.129.178.125	10.200.13.50	HTTP	GET http://clients1.google.co.in/complete/search?hl=en&client=hp&expI
181	8.557459	10.129.178.125	10.200.13.50	HTTP	GET http://clients1.google.co.in/complete/search?hl=en&client=hp&expI
184	8.609258	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 200 OK (text/javascript)
193	8.765583	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 200 OK (text/javascript)
200	8.871058	10.129.178.125	10.200.13.50	HTTP	GET http://clients1.google.co.in/complete/search?hl=en&client=hp&expI

Internet Protocol, Src: 10.129.178.125 (10.129.178.125), Dst: 10.200.13.50 (10.200.13.50)
Transmission Control Protocol, Src Port: sm-pas-5 (2942), Dst Port: http (80), Seq: 1, Ack: 1, Len: 673
Hypertext Transfer Protocol
GET http://www.google.com/ HTTP/1.0\r\n
[Expert Info (Chat/Sequence): GET http://www.google.com/ HTTP/1.0\r\n]
Request Method: GET
Request URI: http://www.google.com/
Request Version: HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, image/png, application/x-shockwave-flash, application/vnd.ms-excel, application
Accept-Language: en-us\r\n
[truncated] Cookie: PREF=ID=4e39cd81988162ee:U=6c821b2e45d9eb9e:TM=1281184973:LM=1281608835:GM=1:S=ezza6ty2wtahPxvm; NID=37=Ecns-
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; windows NT 5.1; SV1; InfoPath.2)\r\n

0000 00 04 96 10 a6 60 00 27 0e 2e 7c 5d 08 00 45 00
0010 02 c9 bb d8 40 00 80 06 67 5e 0a 81 b2 7d 0a c8
0020 0d 32 0b 7e 00 50 34 22 67 68 b8 b2 5e 9a 50 18
0030 ff ff d7 b3 00 00 47 45 54 20 68 74 74 70 3a 2f
0040 2f 77 77 77 2e 67 6f 6f 67 6c 65 2e 63 6f 6d 2f
0050 59 43 54 54 2e 2f 21 2e 20 0d 03 41 62 62 65 70

Frame (frame), 727 bytes

Packets: 514 Displayed: 60 Marked: 0 Dropped: 0

Profile: Default

HTTPS

HTTPS stands for Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL.

HTTPS encrypts and decrypts the page requests and page information between the client browser and the web server using a Secure Socket Layer (SSL).

HTTPS by default uses port 443

HTTPS Packet Sniffing using Wireshark

The screenshot displays the Wireshark interface with a packet capture of an HTTP CONNECT request. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, and Help. The toolbar contains various icons for file operations and network analysis. The filter bar shows 'http' and the expression bar is empty. The packet list pane shows a list of captured packets, with packet 174 selected, which is an HTTP CONNECT request to www.google.com:443. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol, Transmission Control Protocol, and Hypertext Transfer Protocol. The Hypertext Transfer Protocol section is expanded to show the CONNECT method and the request URI 'www.google.com:443'. The packet bytes pane shows the raw data of the selected packet in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
36	2.723366	Fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
43	3.419790	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
72	5.722388	fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
81	6.419567	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
116	8.722836	fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
125	9.419626	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
160	12.723154	fe80::e02d:30c3:83cff02::c		SSDP	M-SEARCH * HTTP/1.1
168	13.419875	fe80::452d:8ccb:8d2ff02::c		SSDP	M-SEARCH * HTTP/1.1
174	13.748558	10.129.178.125	10.200.13.50	HTTP	CONNECT www.google.com:443 HTTP/1.0
178	13.826884	10.200.13.50	10.129.178.125	HTTP	HTTP/1.0 200 Connection established
179	13.827198	10.129.178.125	10.200.13.50	SSLV2	Client Hello
182	13.902884	10.200.13.50	10.129.178.125	SSLV3	Server Hello
183	13.902897	10.200.13.50	10.129.178.125	SSLV3	Certificate, Server Hello Done
185	13.903462	10.129.178.125	10.200.13.50	SSLV3	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
188	13.976465	10.200.13.50	10.129.178.125	SSLV3	Change Cipher Spec, Encrypted Handshake Message
189	13.978197	10.129.178.125	10.200.13.50	SSLV3	Application Data
192	14.065821	10.200.13.50	10.129.178.125	SSLV3	Application Data
193	14.065941	10.200.13.50	10.129.178.125	SSLV3	Application Data
195	14.066067	10.200.13.50	10.129.178.125	SSLV3	Application Data
196	14.066186	10.200.13.50	10.129.178.125	TCP	[TCP segment of a reassembled PDU]
197	14.066192	10.200.13.50	10.129.178.125	SSLV3	Application Data
199	14.070964	10.200.13.50	10.129.178.125	TCP	[TCP segment of a reassembled PDU]
200	14.070972	10.200.13.50	10.129.178.125	SSLV3	Application Data
252	15.688025	10.129.178.125	10.200.13.50	HTTP	CONNECT www.google.com:443 HTTP/1.0

Frame 174: 324 bytes on wire (2592 bits), 324 bytes captured (2592 bits)
Ethernet II, Src: IntelCor_2e:7c:5d (00:27:0e:2e:7c:5d), Dst: ExtremeN_10:a6:60 (00:04:96:10:a6:60)
Internet Protocol, Src: 10.129.178.125 (10.129.178.125), Dst: 10.200.13.50 (10.200.13.50)
Transmission Control Protocol, Src Port: hyperip (3919), Dst Port: http (80), Seq: 1, Ack: 1, Len: 270
Hypertext Transfer Protocol
CONNECT www.google.com:443 HTTP/1.0\r\n
[Expert Info (Chat/Sequence): CONNECT www.google.com:443 HTTP/1.0\r\n]
[Message: CONNECT www.google.com:443 HTTP/1.0\r\n]
[Severity level: Chat]
[Group: sequence]
Request Method: CONNECT
Request URI: www.google.com:443
Request Version: HTTP/1.0

```
0000 00 04 96 10 a6 60 00 27 0e 2e 7c 5d 08 00 45 00  ....  .]..E.  
0010 01 36 05 58 40 00 80 06 1f 72 0a 81 b2 7d 0a c8  .6.X@...r...}.  
0020 0d 32 0f 4f 00 50 bd 68 ee 21 30 e7 d7 c0 50 18  .2.O.P.h !0...P.  
0030 ff ff d6 20 00 00 43 4f 4e 4e 45 43 54 20 77 77  ... ..CO NNECT ww  
0040 77 2e 67 6f 6f 67 6c 65 2e 63 6f 6d 3a 34 34 33  w.google .com:443  
0050 20 48 54 54 50 2f 21 20 20 0d 02 55 72 65 72 2d  HTTP/1.0 User
```

HTTP Vs HTTPS

HTTP doesn't encrypt data at all with all its communication pretty much readable, with no decoding, translation or decryption required, Completely insecure

HTTPS is a secure connection, which means the data between the client and Web server is encrypted.

HTTPS uses public key encryption to secure data