Topic 01: Overview

Kameswari Chebrolu
Room 402
Department of CSE, IIT Bombay
http://www.cse.iitb.ac.in/~chebrolu

Jan 13, 2009
Internet protocol stack

- **Application**: supporting network applications
  - FTP, SMTP, HTTP
- **Transport**: process-process data transfer
  - TCP, UDP
- **Network**: routing of datagrams from source to destination
  - IP, routing protocols
- **Link**: data transfer between neighboring network elements
  - PPP, Ethernet
- **Physical**: bits “on the wire”
Encapsulation

- The application data is sent down
- Each layer adds a header to the data (PDU) from its higher layer.
Multiplexing and Demultiplexing

- Different higher layer protocols can use the service by the same lower layer protocol.
Naming and Addressing

- Uniquely identify processes in different computers for communications.
- Domain name
- IP address
- Port number
- MAC address
Domain Name

• A user friendly name to identify a host
• Domain Name System (DNS): resolves a domain name to the corresponding IP address.
  – E.g: www.cse.iitb.ac.in -> 121.241.28.216
  – distributed database maintained by DNS servers
  – DNS query and reply exchange between client and server
  – A host first contacts its local DNS server to get the mapping
IP Address

• Each interface in a host is assigned an IP address.
• IPv4, 32 bits, dotted-decimal notation
  
  128.238.42.112 means
  
  10000000 in 1st Byte
  11101110 in 2nd Byte
  00101010 in 3rd Byte
  01110000 in 4th Byte

• IPv6, 128-bit address
IP Address Cont...

- An IP address can be divided into a subnet and host part
  - the subnet part identifies a given network
  - the host part identifies a given host within a network
- A subnet mask specifies the number of bits that correspond to the subnet part
  - E.g 255.255.255.192 -> /26 (26 bits)
Media Access Control Address

- Apart from IP address, each interface in a host also has a hardware address (MAC address)
- Ethernet MAC address is 48 bits long
  - E.g 00:18:F3:96:C2:A7
- ARP protocol is used to translate an IP address to MAC address
Port Numbers

• Address for the application layer user process.
• **Port Number** field specified in TCP or UDP header.
• Well-known port numbers
  – 1 to 255: Internet wide services
  – 256 to 1023: preserved for Unix specific services
  – 1024 and up: ephemeral port numbers
Ethernet Frame Format

- Source Ethernet (MAC) Address
- Destination Ethernet Address
- Frame Type: used to identify the payload
- CRC: used for error control

<table>
<thead>
<tr>
<th>Destination Address</th>
<th>Source Address</th>
<th>Frame Type</th>
<th>Data</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 bytes</td>
<td>6 bytes</td>
<td>2 bytes</td>
<td>46–1500 bytes</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>
IP Header Format

- Size: 20 bytes without options.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>Differentiated Services</td>
<td>Total Length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identification</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time to Live</td>
<td>Fragment Offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protocol</td>
<td>Header Checksum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source IP Address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Destination IP Address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options (if any, &lt;= 40 bytes)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>
UDP Header Format

<table>
<thead>
<tr>
<th>0</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port Number</td>
<td>Destination Port Number</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Checksum</td>
<td></td>
</tr>
</tbody>
</table>
## TCP Header Format

<table>
<thead>
<tr>
<th>Field</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port Number</td>
<td>0</td>
</tr>
<tr>
<td>Destination Port Number</td>
<td>16</td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
</tr>
<tr>
<td>Acknowledgement Number</td>
<td></td>
</tr>
<tr>
<td>Hdr Len.</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
</tr>
<tr>
<td>TCP Checksum</td>
<td></td>
</tr>
<tr>
<td>Window Size</td>
<td></td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
</tr>
<tr>
<td>Data (optional)</td>
<td></td>
</tr>
</tbody>
</table>
Example of TCP/IP
Networking Code Organization

- Most applications are implemented as *user space* processes.
- Protocols are implemented in the system kernel
  - Socket layer
  - Protocol layer
  - Interface layer
Network Configurations Files

- When a host is configured to boot locally, TCP/IP configuration parameters are stored in files.
- When the system boots up, parameters are read from the files and used to configure the daemons and the network interface.
- A parameter may be changed by editing the corresponding configuration file.
Diagnostic Tools

- **Tcpdump**
  - E.g `tcpdump -enx host 10.129.5.181 -w exe3.out`

- **Wireshark**
  - `wireshark -r exe3.out`
Packet Sniffer

• Sniffs messages being sent/received from/by your computer

• Store and display the contents of the various protocol fields in the messages

• Passive program
  - never sends packets itself
  - no packets addressed to it
  - receives a copy of all packets (sent/received)
Packet Sniffer Structure

Figure 1: Packet sniffer structure
Figure 1.22  An Ethereal screen shot