Overview of TCP/IP

Dr. Ehab S. Al-Shaer

Link Layer: Ethernet

- Ethernet (actually IEEE 802.3) is widely used
- Multi-access (shared medium).
- Every Ethernet interface has a unique 48 bit address (a.k.a. hardware address).
- Example: C0:B3:44:17:21:17
- The broadcast address is all 1’s.
- Addresses are assigned to vendors by a central authority.
CSMA/CD
Carrier Sense Multiple Access with Collision Detection

- **Carrier Sense**: can tell when another host is transmitting
- **Multiple Access**: many hosts on 1 wire
- **Collision Detection**: can tell when another host transmits at the same time.

An Ethernet Frame

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Destination Address</th>
<th>Source Address</th>
<th>Length</th>
<th>DATA</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bytes</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>0-1500</td>
<td>4</td>
</tr>
</tbody>
</table>

- The preamble is a sequence of alternating 1s and 0s used for synchronization.
- CRC is Cyclic Redundancy Check
Ethernet Addressing

Each interface looks at every frame and inspects the destination address. If the address does not match the hardware address of the interface or the broadcast address, the frame is discarded.

Some interfaces are programmed to recognize mcast addresses (mcast-enabled).

---

Loopback Interface

- client and server in the same host can communicate without using the “real” IP address
- “localhost” or 127.0.0.1 (try telnet)
- Loopback driver is used when
  - a pkt is sent to localhost/127.0.0.1
  - a pkt is unicast and its destination is the same as IP src
  - a pkt is mcast or broadcast
- MTU and Path MTU
IP: Internet Protocol

- IP is the network layer
  - packet delivery service (host-to-host).
  - Relay pkts between different DL protocols.
- IP provides connectionless, unreliable delivery of IP datagrams.
- Connectionless: each datagram is independent of all others.
- Unreliable: there is no guarantee that datagrams are delivered correctly or at all.

IP Datagram

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERS</td>
<td>1 byte</td>
<td>Version number</td>
</tr>
<tr>
<td>HL</td>
<td>1 byte</td>
<td>Header Length</td>
</tr>
<tr>
<td>TOS</td>
<td>1 byte</td>
<td>Type of Service</td>
</tr>
<tr>
<td>Total Length</td>
<td>4 bytes</td>
<td>Total length of the IP datagram</td>
</tr>
<tr>
<td>16-bit Datagram ID</td>
<td>2 bytes</td>
<td>Datagram ID</td>
</tr>
<tr>
<td>FLAG</td>
<td>1 bit</td>
<td>Don't Fragment</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>12 bits</td>
<td>Offset of fragment</td>
</tr>
<tr>
<td>TTL</td>
<td>8 bits</td>
<td>Time to Live</td>
</tr>
<tr>
<td>Protocol</td>
<td>1 byte</td>
<td>Protocol Type</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>1 byte</td>
<td>Checksum of header</td>
</tr>
<tr>
<td>32-bit Source Address</td>
<td>4 bytes</td>
<td>Source Address</td>
</tr>
<tr>
<td>32-bit Destination Address</td>
<td>4 bytes</td>
<td>Destination Address</td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
<td>Options</td>
</tr>
<tr>
<td>Data</td>
<td>variable</td>
<td>Payload data</td>
</tr>
</tbody>
</table>
IP Addresses

IP addresses are not the same as the underlying data-link (MAC) addresses.

Why?

The formats of IP Addresses

<table>
<thead>
<tr>
<th>Class</th>
<th>NetID</th>
<th>HostID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>HostID 24bit</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>NetID 14bit HostID 16bit</td>
</tr>
<tr>
<td>C</td>
<td>110</td>
<td>NetID 21bit HostID</td>
</tr>
<tr>
<td>D</td>
<td>1110</td>
<td>Multicast Address 28bit</td>
</tr>
<tr>
<td>E</td>
<td>11110</td>
<td>(reserved for future) 27bit</td>
</tr>
</tbody>
</table>

Figure 1.6/P. 8 gives the range

Dr. Ehab Al-Shaer
Network and Host IDs

- A Network ID is assigned to an organization by a global authority.
- Host IDs are assigned locally by a system administrator.
- Both the Network ID and the Host ID are used for routing.

IP Addresses

- IP Addresses are usually shown in dotted decimal notation:
  - 1.2.3.4   00000000 00000000 00000000 00000001
  - 128.213.1.1 10000000 11010011 00000000 00000000

  CS has a class B network

- acamar.cyberdna.uncc.edu is 152.15.96.141
Host and Network Addresses

- A single network interface is assigned a single IP address called the host address.
- A host may have multiple interfaces, and therefore multiple host addresses.
- Hosts that share a network all have the same IP network address (the network ID).
- IP Broadcast address
  - An IP broadcast addresses has a host ID of all 1s.

IP Routing

- IP datagram may be sent
  - directly to a remote host which is directly connected, or
  - to a default router that will deliver it to the destination
- What is the difference between host-IP and router-IP?
- Routing Table consists of
  - Destination IP address
  - IP address of a next-hop router
  - Flags
  - Port specification (which interface should be used)
IP Routing

IP Routing Actions:
- Search routing Table for the complete IP address, else
- Search routing Table for destination network ID in IP, else
- Search routing Table for “default” entry
- In entry is found for any of these cases, the flag is used to forward the pkt to next host or hop
- Otherwise, error = “host/network unreachable”
- Examples (fig 3.3 and 3.4)

Most host (and some routers) use “default” rout, if destination is not in local net
IP destination never get changed
Different DL/Ethernet addresses are used in each link
Subnet Addresses

- An organization can subdivide its host address space into groups called subnets.
- The subnet ID is generally used to group hosts based on the physical network topology.

<table>
<thead>
<tr>
<th>NetID</th>
<th>SubnetID</th>
<th>HostID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Class A and B has too many hosts:
  - Class A = $2^{24} - 2$, Class B = $2^{16} - 2$
- Class B with 6 bit subnet; how many hosts and subnets does it have?

Subnet Addresses

- Subnetting reduces the routing table size
- Host needs to know <IP + Subnet mask> to determine if the destination is
  - in local subnet
  - in local net
  - remote net
- Subnet mask: a hex notation used with IP
  - Example: 142.255.5.5 & mask: 255.255.255.0 => class B and subnet ID = 5, hostID = 5
  - receives 142.255.1.5 => same net but diff subnet
  - receives 142.255.5.3 => same net but and subnet
  - receives 195.44.235.6 => different net
Subnet Addresses

Example
- 140.225.13.33 & mask:255.255.255.224 =>
  - class B and subnet 27 bits, host 5 bit =>
  - Net/SubnetID = 140.225.13.32, HostID=1

Configuration Commands
- `ifconfig` to configure interfaces during bootstrap
  - `ifconfig -a` to get interface information
- `netstat` provides information about the network status of the interface

Subnetting

Dr. Ehab Al-Shaer
Services provided by IP

- Connectionless Delivery (each datagram is treated individually).
- Unreliable (delivery is not guaranteed).
- Fragmentation / Reassembly (based on hardware MTU).
- Routing.
- Error detection.

IP Datagram Fragmentation

- Each fragment (packet) has the same structure as the IP datagram.
- IP specifies that datagram reassembly is done only at the destination (not on a hop-by-hop basis).
- If any of the fragments are lost - the entire datagram is discarded (and an ICMP message is sent to the sender).
Mapping IP Addresses to Hardware Addresses

- IP Addresses are not recognized by hardware.
- If we know the IP address of a host, how do we find out the hardware address?
- The process of finding the hardware address of a host given the IP address is called *Address Resolution*.

Reverse Address Resolution

- The process of finding out the IP address of a host given a hardware address is called *Reverse Address Resolution*.
- Reverse address resolution is needed by diskless workstations when booting.
ARP

The Address Resolution Protocol is used by a sending host when it knows the IP address of the destination but needs the Ethernet address.

ARP is a broadcast protocol - every host on the network receives the request.

Each host checks the request against its IP address - the right one responds.

Dr. Ehab Al-Shaer
ARP

ARP does not need to be done every time an IP datagram is sent - hosts remember (cache) the hardware addresses of each other.

Part of the ARP protocol specifies that the receiving host should also remember (cache) the IP and hardware addresses of the sending host.

Figure 4.2

RARP

HEY - Everyone please listen!
My Ethernet address is 22:BC:66:17:01:75. Does anyone know my IP address?

Hi Red! Your IP address is 128.213.1.17.
IP Routing

- Routing Table (RT) Search
  - Host (complete march)
  - Network (netid or subnet)
  - default
- Routing mechanism
  - which port/interface to use for forwarding
  - performed by IP
- Routing Policy
  - which routing should be added to the RT
  - performed by routing daemon

- netstat -rn => routing table (P.113)
  - G (indirect) => IP addr is used as dest but the MAC addr is the MAC addr of the Gateway
  - H (host) => “Destination” is a host and must completely match the target O=IP. Otherwise, Destination is a network address and only NetID (or SubnetID) must match.
- What MAC and IP are used when
  - svr4 -> sun
  - svr4 -> slip
  - svr4 -> www.cs.depaul.edu
ICMP: Internet Control Message Protocol

- It is an adjunct to IP!
- Functions: (exchange control information)
  - communications errors
  - important (useful) events
  - special request
- ICMP uses IP to deliver messages (encapsulated):

  ![Diagram of ICMP message format]

  - IP datagram
  - IP header: 20 bytes
  - ICMP message

ICMP Message Format

- ICMP Msg includes
  - type => 15 different types
  - code => (subtypes) for each type
  - data (depends on type and code)
    - IP header to know the protocol type (tcp or udp)
    - first 8 bytes of IP datagram (src and dest ports)

<table>
<thead>
<tr>
<th>8 bits</th>
<th>8 bits</th>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Code</td>
<td>16 bit Checksum</td>
</tr>
</tbody>
</table>

(Contents depends on type & code)
ICMP Messages

- Echo Request
- Echo Response
- Destination(host/network/port) Unreachable
- Redirect
- Time Exceeded
- Redirect (route change)
- there are more … Table 6.3/P.71

ICMP Port Unreachable Error

```
bsdi% tftp
 tftp> connect svr4 8888
 tftp> get temp.foo
 transfer timed out
 tftp> quit

See tcpdump output (P. 78)
```

<table>
<thead>
<tr>
<th>type</th>
<th>Code</th>
<th>16 bit Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused (must be 0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IP header of datagram that generated error (with options) + first 8 bytes of original IP datagram data

ICMP Unreachable Message Format
Host Unreachable Error

- Author subnet example
  - P. 118: dial up links is down
- How long a pkt to unknown host will travel in the Internet
  - intermediate routers use “default”
  - until it get to a router that has a global knowledge (all nets such as NFSNET)
  - Example P. 118

---

ICMP Redirect Error

- Redirects are generated by routers only and intended to be used by hosts only (routers have their own protocols)
- Generating redirect error requires (sender)
  - outgoing = incoming interfaces
  - redirect must not created/modified by redirect
  - No source routing is used in the datagram pkt
  - OS kernel must be configured for this
- Redirect is accepted if (receiver)
  - the new assigned router is directly connected
  - comes from its current/default router
  - a router does not nominate itself
  - corrected rout must be indirect (means non-optimized)
What does TCP Offer?

TCP services and components
- End-to-end connection-oriented – means what?
- Reliable Error Control
  - Goal: no loss, no duplicate and no out-of-sequence
  - Techniques: sequence numbers, acknowledgment, timers, checksum
- Flow Control
  - to avoid overrunning the receiver
- Congestion Control and Avoidance
  - to avoid overrunning the network
- Byte stream
  - no message boundaries
  - Why? (dealing with congestion)

TCP Header

- http://www.mnlab.cs.depaul.edu/TDC562/figures/ch17/fig17_02.jpg
- Min length 20 and Max 60
- Ports are always in the beginning
- TCP uses sliding window
  - no selective repeat
  - no negative ACK
- Window Size
  - number of bytes a receiver can accept
  - Max 65K but can be scaled up by option
- Checksum covers header and data
- urgent pointer + sequence num => urgent data
- Options: MSS
TCP state scenario

netstat -a gives the state of the current connections in a Unix machine

Connection termination using Three-way handshake
Aborting a connection

- Client process failed
- TCP sends reset to abort connection
- Client and server TCP throw away all the data in queue

Simultaneous open – 4 way handshake

- Not that common
- Both are client+server, both issue active open at the same time
Simultaneous close

- Not that common
- Both are client+server, both issue close at the same time

TCP Congestion Control

- Maintains three variables:
  - cwnd – congestion window (imposed on the sender)
  - flow_win – flow window; receiver advertised window
  - ssthresh – threshold size (used to control the cwnd update) – basically it is close to the knee

- For sending use:
  \[ \text{win} = \min(\text{flow\_win}, \text{cwnd}) \]
TCP: Slow Start

Goal: discover congestion quickly
- injecting packets very fast might exceed the network knee or capacity (pipeline)
- injecting packets too slow might lead to underutilization of the network capacity

How?
- quickly increase \( cwnd \) until network congested \( \rightarrow \) get a rough estimate of the optimal \( cwnd \)

How do we know when network is congested?
- packet loss (TCP, [Jac88])
  - over the cliff here \( \rightarrow \) congestion control
- congestion notification ECN (Ram & Floyd, RFC[2481])
  - set it in the packet that went thru a congestion. The receiver will use this to alarm the sender which will do congestion avoidance
  - over the knee but before the cliff \( \rightarrow \) congestion avoidance

Whenever starting traffic on a new connection, or whenever increasing traffic after congestion was experienced:
- Set \( cwnd = 1 \)
- Each time a segment is acknowledged increment \( cwnd \) by one \( (cwnd++) \).

Does Slow Start increment slowly? Not really.
In fact, the increase of \( cwnd \) is exponential.
**Slow Start Example**

- The congestion window size grows very rapidly (linearly with number of segments acked but exponentially over time or RTT)

- TCP slows down the increase of \( cwnd \) when \( cwnd \geq ssthresh \)

**Congestion Avoidance**

- Goal: maintain operating point at the left of the cliff:
- **How?**
  - additive increase: starting from the rough estimate, slowly increase \( cwnd \) to probe for additional available bandwidth
  - multiplicative decrease: cut congestion window size aggressively if a timeout occurs
Congestion Avoidance
Algorithm (Tahoe)– big picture

\[\text{cwnd} = 1 \text{ mss; ssthresh} = 64 \text{K or infinite} \quad /\!* \text{means one segment}*/\]
\[\text{sendWin} = \min(\text{cwnd, advertisedWin}) \quad /\!* \text{max to send}*/\]

if (Ack for one segment is received)
\[
\begin{align*}
&\text{if (cwnd < ssthresh)} \\
&\quad \text{cwnd}++ \\
&\text{else} \\
&\quad \text{cwnd} += 1/\text{cwnd}
\end{align*}
\]
/* slow start phase */
/* expo increase */
/* Congestion avoidance phase */
/* means one per RTT or when ack of all win segments are received linear increase*/

if (timeout or duplicate)
\[
\begin{align*}
&\text{ssthresh} = 1/2 \times \text{sendWin} \\
&\text{cwnd} = 1 \text{ mss;}
\end{align*}
\]
/* multiplicative expon bakeoff */
/* back to slow start */

\[\text{See Fig 21.8} \quad \text{Dr. Ehab Al-Shaer}\]

Congestion Avoidance

\[\text{to slow down (shorten) “Slow Start”}\]
\[\text{If cwnd > ssthresh then} \quad /\!* \text{each time a segment is acknowledged increment cwnd by} 1/\text{cwnd}*/\]
\[
\text{cwnd} += 1/\text{cwnd}.
\]
\[\text{So cwnd is increased by one only if all segments have been acknowledged.}\]
\[\text{(more about ssthresh latter)}\]
Assume that $cwnd = 1$

Assume that $ssthresh = 8$

The Big Picture
TCP Timers

There are four timers in TCP:
- Re-transmission timer: for ack
- Persist timer: for requesting receiver win
- Keepalive timer: for checking the connection
- 2 MSL timer: for TIME_WAIT state

Comments on Timers Implementation