Introduction to Networks and the Internet

CMPE 80N
Spring 2003
Week 7

Announcements

• Project 2.
  -- Reference page.
• Library presentation.
• Internet History video.

Today

• Internetworking (cont’d).
  -- Fragmentation.
  -- IP.
  -- IP addressing.

Maximum Transfer Unit

• Each data link layer technology specifies the maximum size of a frame.
  -- Called the Maximum Transfer Unit (MTU).
    • Ethernet: 1,500 bytes.
    • Token Ring: 2048 or 4096 bytes.
• What happens when large packet wants to travel through network with smaller MTU?
  • Maximum payloads (data portion of datagram) range from 48 bytes (ATM cells) to 64Kbytes (IP packets).
**MTU (cont’d)**

- A possible solution:
  - The sender may limit the size of the datagrams to the MTU of the network
  - What if there are other networks in the path to destination with smaller MTU?

**Fragmentation**

- Another solution (used by IP): **fragmentation**.
  - Gateways break packets into fragments to fit the network’s MTU; each sent as separate datagram.
  - Gateway on the other side have to **reassemble** fragments into original datagram.

---

**Keeping Track of Fragments**

- Fragments must be numbered so that original data stream can be reconstructed.
- Define elementary fragment size that can pass through every network.
- When packet fragmented, all pieces equal to elementary fragment size, except last one (may be smaller).
- Datagram may contain several fragments.

**Fragmentation - Example**
**Fragmentation – Example (cont’d)**

- Header contains packet number, number of first fragment in packet, and last-fragment bit.

(a) Original packet with 10 data bytes.

(b) Fragments after passing through network with MTU = 8 bytes.

**Announcements**

- Quiz 2.
- Christy Hightower’s presentation on 05.22.
- Project 2.

**The IP Protocol**

**The Internet’s Network Layer**

- The Internet as a collection of networks or autonomous systems (ASs).
- Hierarchical structure.
The Internet Protocol: IP

- Glues Internet together.
- Common network-layer protocol spoken by all Internet participating networks.
- Best effort datagram service:
  - No reliability guarantees.
  - No ordering guarantees.

IP (cont’d)

- IP is responsible for datagram routing.
- Important: each datagram is routed independently!
  - Two different datagrams from same source to same destination can take different routes!
  - Why?
  - Implications?

IP (cont’d)

- IP provides a **best effort** delivery mechanism
  - Does not guarantee to prevent duplicate datagrams, delayed and out-of-order delivery, corruption of data or datagram loss
- **Reliable delivery** is provided by the transport layer, not the network layer (IP)
- Network layer (IP) can detect and report errors without actually **fixing** them

IP

- Transport layer breaks data streams into datagrams which are transmitted over Internet, possibly being fragmented.
- When all datagram fragments arrive at destination, reassembled by network layer and delivered to transport layer at destination host.
IP Datagram Format

- IP datagram consists of header and data (or payload).
- Header:
  - 20-byte fixed (mandatory) part.
  - Variable length optional part.

IP Data Gram Format

 Datagram Header Format

IP Versions

  - Current, predominant version.
  - 32-bit long addresses.
  - Longer addresses (16-byte long).

Addressing

- Single, uniform addresses.
- Sending hosts puts destination internetworking address in the packet.
- Destination addresses can be interpreted by any intermediate router.
- Router examines address and forwards packet on to the destination.
- Router at destination delivers packet to appropriate host.
**IP Addresses**

- Each computer (host) on the Internet has a unique **IP address**.
- IP address are 32 bits long.
  - 4 billion ($2^{32}$) addresses.
  - IP addresses are represented in “dotted decimal”, such as 128.114.144.4
    - Each group of numbers corresponds to 8 bits.
    - Can go from 0 to 255.

**IP Addresses (cont’d)**

- IP addresses are hierarchical.
- Divided into a **prefix** and a **suffix**
  - Prefix identifies network to which computers are attached.
  - Suffix identifies computers within that network.
- Remember that two computers in a network can communicate “directly”.

**Networks and Host Numbers**

- Every network in an Internet is assigned a unique **network number**.
- Each host on a specific network is assigned a **host address** that is unique within that network.
- Host’s IP address is the combination of the network number (prefix) and host address (suffix).
- Assignment of network numbers must be coordinated globally; assignment of host addresses can be managed locally.

**IP Address Format**

- IP (v4) address are 32 bits long.
- There are different classes of addresses, corresponding to different subdivisions of the 32 bits into prefix and suffix.
  - Some address classes have large prefix, small suffix.
    - Many networks, few hosts per network.
  - Other address classes have small prefix, large suffix.
    - Few networks, many hosts per network.
IP Address Format (cont’d)

- How can we recognize to which class an IP address belongs to?
  - Look at the first 4 bits!

<table>
<thead>
<tr>
<th>Class</th>
<th>Net</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>0</td>
<td>Host</td>
</tr>
<tr>
<td>Class B</td>
<td>10</td>
<td>Host</td>
</tr>
<tr>
<td>Class C</td>
<td>110</td>
<td>Host</td>
</tr>
<tr>
<td>Class D</td>
<td>1110</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>

- Class A, B and C are **primary classes**
  - Used for ordinary addressing.
- Class D is used for **multicast**, which is a limited form of **broadcast**.
  - Internet hosts join a multicast group.
  - Packets are delivered to all members of the group.
  - Routers manage delivery of single packets from source to all members of multicast group.

IP Address Formats 1

- 4 different classes:

<table>
<thead>
<tr>
<th>Class</th>
<th>Range of Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 through 127</td>
</tr>
<tr>
<td>B</td>
<td>128 through 191</td>
</tr>
<tr>
<td>C</td>
<td>192 through 233</td>
</tr>
<tr>
<td>D</td>
<td>224 through 239</td>
</tr>
<tr>
<td>E</td>
<td>240 through 255</td>
</tr>
</tbody>
</table>

IP Addresses (cont’d)

- Another way to determine the address class is by looking at the first group of numbers in the dotted decimal notation.
Networks and Hosts in Each Class

<table>
<thead>
<tr>
<th>Address Class</th>
<th>Bits in Prefix</th>
<th>Maximum Number of Networks</th>
<th>Bits in Suffix</th>
<th>Maximum Number of Hosts Per Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7</td>
<td>128</td>
<td>24</td>
<td>1024</td>
</tr>
<tr>
<td>B</td>
<td>14</td>
<td>16384</td>
<td>16</td>
<td>65536</td>
</tr>
<tr>
<td>C</td>
<td>21</td>
<td>2097152</td>
<td>8</td>
<td>256</td>
</tr>
</tbody>
</table>

IP addresses: how to get one?

- The network IP numbers are assigned by the Network Information Center.
- How does host get its IP address in the network?
  - Hard-coded by system administrator in a file.
  - DHCP: "Dynamic Host Configuration Protocol"
    * Dynamically get address: "plug-and-play"

DHCP

- DHCP allows a computer to join a new network and automatically obtain an IP address. The network administrator establishes a pool of addresses for DHCP to assign.
- When a computer boots, it broadcasts a DHCP request to which a server sends a DHCP reply.

Multi-Addresses

- A router usually has more than one IP address.
- Multi-homed host: host with multiple network interfaces each of which has different IP address.
The Transport Layer

- End-to-end.
- Communication from source to destination host.
- Only hosts run transport-level protocols.
- Under user’s control as opposed to network layer which is controlled/owned by network provider.

Types of Transport Services

- Provided to the application layer.
- Connection-less versus connection-oriented.
- Connection-less service: no logical connections, no flow or error control.
- Connection-oriented:
  - Based on logical connections: connection setup, data transfer, connection teardown.
  - Flow and error control.
  - Reliability and in-order delivery.
**TPDU**
- Transport protocol data unit.
- Messages sent between transport entities.
- TPDU's contained in network-layer packets, which in turn are contained in DLL frames.

**Transport Protocol Addressing**
- Address of the transport-level entity.
- Several transport-level entities may be running on single machine.
- Source-destination address pair not enough to uniquely identify transport entity.
- Port number: uniquely identifies transport entity.

**The Internet Transport Protocols: TCP and UDP**
- UDP: user datagram protocol (RFC 768).
  - Connection-less protocol.
- TCP: transmission control protocol (RFCs 793, 1122, 1323).
  - Connection-oriented protocol.

**TCP**
- Reliable end-to-end communication.
- TCP transport entity:
  - Interfaces to the IP layer.
  - Manages TCP streams.
    - Accepts user data, breaks it down and sends it as separate IP datagrams.
    - At receiver, reconstructs original byte stream from IP datagrams.
**Features of TCP**

- **Connection oriented**: An application requests a "connection" to destination and uses connection to transfer data
  - IP does not use "connections" - each datagram is sent independently!
- **Point-to-point**: A TCP connection has two endpoints (no broadcast/multicast)
- **Reliability**: TCP guarantees that data will be delivered without loss, duplication or transmission errors

**Features of TCP (cont’d)**

- **Full duplex**: Endpoints can exchange data in both directions simultaneously
- **Reliable connection startup**: TCP guarantees reliable, synchronized startup between endpoints (using “three-way handshake”)
- **Gracefully connection shutdown**: TCP guarantees delivery of all data after endpoint shutdown

**Delivering TCP Segments**

- TCP segments travel in IP datagrams.
- Internet routers only look at IP header to forward datagrams.

**Delivering TCP**

- TCP at destination interprets TCP messages
TCP Connection Setup

- 3-way handshake.

TCP Connection Release

- Graceful release:
  - Each side of the connection released independently.
  - Either side send TCP segment with FIN=1.
  - When FIN acknowledged, that direction is shut down for data.
  - Connection released when both sides shut down.
  - 4 segments: 1 FIN and 1 ACK for each direction;
    1st. ACK+2nd. FIN combined.

TCP and Reliable Delivery

- TCP provides reliable delivery, recovering from:
  - Lost packets
  - Duplicate packets
  - Delayed packets
  - Corrupted data
  - Source/destination mismatches
  - Congestion

TCP Reliability

- Reliable delivery.
  - Acknowledgements..
  - Timeouts and retransmissions.
- Ordered delivery.
  - Sequence numbers.
Lost Packets

- Recipient sends acknowledgment control message (ACK) to sender to verify successful receipt of data
  - ACKs usually are carried onboard other TCP packets.
  - However, even if an application has nothing to transmit, it must transmit acknowledgment packets for each packet it receives.
- Thus, for each packet sent, a host expects to receive an acknowledgment, which ensures that the packet did not get lost.
  - What if the packet or the acknowledgment get lost?

Lost Packets (cont’d)

- Retransmission timer
  - When a data segment is sent, a timer is started
  - If the segment is acknowledged before the timer expires, the timer is stopped and reset
  - Otherwise, the segment is retransmitted (and the timer is reset and started again)
- The choice of the timeout is critical!
  - If timeout is too long: overall throughput may be reduced (always waiting for acknowledgments)
  - If timeout is too short: too many packets get retransmitted (may increase network congestion)

IMPORTANT: packet retransmission (especially if it has to be carried out on an end-to-end basis) significantly increases latency (delay)

- For real-time video or audio transmission, delay is a more important performance issue than error rate
- Thus, in many cases it is preferable to forget the error and simply work with the received data stream

Lost Packets - Example
TCP Transmission

- Sender process initiates connection.
- Once connection established, TCP can start sending data.
- Sender writes bytes to TCP stream.
- TCP sender breaks byte stream into segments.
  - Each byte assigned sequence number.
  - Segment sent and timer started.

TCP Transmission (cont’d)

- If timer expires, retransmit segment.
  - After retransmitting segment for maximum number of times, assumes connection is dead and closes it.
- If user aborts connection, sending TCP flushes its buffers and sends RESET segment.
- Receiving TCP decides when to pass received data to upper layer.

Flow Control

- Flow control is necessary so that source doesn’t transmit too fast for given receiver.
  - E.g., a fast server trying to send 1Gb/s data to a small PC.
  - Without some form of control, some data will get lost.

TCP Flow Control

- Sliding window.
  - Receiver’s advertised window.
  - Size of advertised window related to receiver’s buffer space.
  - Sender can send data up to receiver’s advertised window.
**TCP Sliding Window**

- TCP sliding window is a mechanism used in the Transmission Control Protocol (TCP) to control the flow of data between two endpoints.
- A sliding window is a sequence of bytes that can be sent or received.
- The window size determines how much data can be sent before an acknowledgment is received.

**TCP Flow Control: Example**

- Illustrates the interaction between the application layer and the transport layer.
- Shows how the window size affects the flow of data.
- Demonstrates the use of acknowledgment (ACK) to control the window size.

**Congestion**

- Network with 1 Mb/s lines and 1000 computers, half of which are trying to transfer files at 100 Kb/s to the other half.
  - The total offered traffic exceeds what the network can handle (congestion).
- **Congestion collapse:**
  - When congestion occurs, packets get dropped.
  - Due to packet loss, packets get retransmitted.
  - Congestions gets worse and worse!

**Congestion Control**

- Why do it at the transport layer?
  - Real fix to congestion is to slow down sender.
- Use law of “conservation of packets”.
  - Keep number of packets in the network constant.
  - Don’t inject new packet until old one leaves.
- **Congestion indicator:** packet loss.
TCP and Congestion Control

- Interprets packet loss as an indicator of congestion
  - When it senses packet loss, it slows down the rate of packet transmission
  - When packets are received correctly, sends packets faster
    • Still within the limits of the sliding window

TCP Congestion Control

- Like, flow control, also window based.
  - Each sender keeps congestion window (cwin).
  - Number of bytes that may be sent is min(advertised window, cwin).

TCP Congestion Control (cont’d)

- Slow start [Jacobson 1988]:
  - Connection’s congestion window starts at 1 segment.
  - If segment ACKed before time out, cwin = cwin+1.
  - As ACKs come in, current cwin is increased by 1.
  - Exponential increase.

TCP Congestion Control (cont’d)

- Congestion Avoidance:
  - Third parameter: threshold.
  - Initially set to 64KB.
  - If timeout, threshold = cwin/2 and cwin = 1.
  - Re-enters slow-start until cwin = threshold.
  - Then, cwin grows linearly until it reaches receiver’s advertised window.
**TCP Retransmission Timer**

- When segment sent, retransmission timer starts.
  - If segment ACKed, timer stops.
  - If time out, segment retransmitted and timer starts again.

**How to set timer?**

- Based on round-trip time: time between a segment is sent and ACK comes back.
- If timer is too short, unnecessary retransmissions.
- If timer is too long, long retransmission delay.

---

**TCP Segment Header**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source port</td>
<td>Destination port</td>
</tr>
<tr>
<td>Sequence number</td>
<td>Acknowledgment number</td>
</tr>
<tr>
<td>Header length</td>
<td>Window size</td>
</tr>
<tr>
<td>Checksum</td>
<td>Urgent pointer</td>
</tr>
<tr>
<td>Options (0 or more 32-bit words)</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>

---

**TCP Header Fields**

- Source and destination ports identify connection end points.
- Sequence number.
- Acknowledgment number specifies next byte expected.
- TCP header length: how many 32-bit words are contained in header.
- 6-bit unused field.
**TCP Header Fields (cont’d)**

- **6 1-bit flags:**
  - URG: indicate urgent data present; urgent pointer gives byte offset from current sequence number where urgent data is.
  - ACK: indicates whether segment contains acknowledgment; if 0, acknowledgement number field ignored.
  - PUSH: indicates PUSHed data so receiver delivers it to application immediately.

**Flags (cont’d):**
- RST: used to reset connection, reject invalid segment, or refuse to open connection.
- SYN: used to establish connection; connection request, SYN=1, ACK=0.
- FIN: used to release connection.

- **Window size:** how many bytes can be sent starting at acknowledgment number.

**Checksum:** checksums the header+data+pseudo-header.

**Options:** provide way to add extra information.
- Examples:
  - Maximum payload host is willing to accept; can be advertised during connection setup.
  - Window scale factor that allows sender and receiver to negotiate larger window sizes.

**UDP**

- Provides connection-less, unreliable service.
  - No delivery guarantees.
  - No ordering guarantees.
  - No duplicate detection.

- Low overhead.
  - No connection establishment/teardown.

- Suitable for short-lived connections.
  - Example: client-server applications.
### UDP Segment Format

<table>
<thead>
<tr>
<th>Source port</th>
<th>Destination port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Checksum</td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>

Source and destination ports: identify the end points.
Length: 8-byte header + data.
Checksum: optional, if not used, set to zero.

### TCP and UDP

- TCP provides end-to-end communication. It takes care of reliable, error-free transfer of data, and in-sequence delivery.
- UDP has less overhead compared to TCP, but does not guarantee transfers.
  - TCP is preferred to transfer files
  - UDP is preferred to transfer audio/video streams
    - In real-time streaming, we cannot afford the delay consequent to packet retransmission
- Both protocols support multiplexing, i.e. they allow several distinct streams of data between two hosts.