Contents

- Physical layer issues
  - Communication frequency
  - Signal propagation
  - Modulation and Demodulation
- Channel access issues
  - Multiple access / Random access / Asynchronous
  - 802.11 / Bluetooth
  - Capacity / Energy / Fairness / Directional
- System issues
  - Embedded processor
  - Low power design
- Network issues
  - Location management
  - Mobile IP / Cellular IP
  - MANET routing / Clustering
  - Multicast
  - Interoperability
  - Network reliability (TCP)
  - Quality of service (QoS)
The Internet
1969, ARPANet

- After WWII and during Cold war, US government was interested in science and technology research to improve radar signals and communications.

- ARPANet (Advanced Research Projects Agency network) in 1969
  - They documented the Internet protocols
  - Email was developed
  - Networked 4 computers together
  - Government also funded universities for research

1989-1990
Transfer of Internet from Government

- At the beginning of 1989 over 80,000 host computers were connected to what was now called the Internet.
- The US Government officially transferred the governance of the Internet to the National Science Foundation (NSF).
- NSF took control of managing the backbone of the internet and was then called the “NSFNet.”
- In 1995, the NSF turned control of the Internet to a consortium.
Who Controls the Internet?

- Not one person, company or government owns the Internet
- It's truly collaborative, collective enterprise
- There are organizations that have influence and together form a collective body to guide the Internet and the web:
  - **The World Wide Web Consortium (W3C):** sets specification for HTML and the web
  - **The Internet Engineering Task Force (IETF):** focuses on the evolution of the Internet and making sure it runs smoothly
  - **The Internet Architecture Board (IAB):** responsible for defining the backbone of the Internet
  - **The Internet Society (ISOC):** made up of organizations, governments, non profit, communities, Academics, professionals. The group comments on Internet polices, politics, and oversee other boards such as IETF
  - **The Internet Assigned Authority (IANA) and the Internet Network Information Center (InterNIC):** This group is responsible for IP and domain name addressing

Who Controls the Backbone of the Internet?

- Regional and long-distance phone companies, backbone ISP's, cable and satellite companies, and U.S government contribute in significant ways to the telecommunication infrastructure that supports the Internet
- Companies like Sprint, MCI and AT&T make lots of money by leasing access to the Internet
Internet Protocol (IP)

- Packets may be delivered out-of-order
- Packets may be lost
- Packets may be duplicated

Transmission Control Protocol (TCP)

- Reliable ordered delivery
- Implements congestion avoidance and control
- Reliability achieved by “positive acknowledgement with retransmissions”

- End-to-end semantics
  - Acknowledgements sent to TCP sender confirm delivery of data received by TCP receiver
  - Ack for data sent only after data has reached receiver
Reliable, Ordered Delivery:
Positive Acknowledgement

Send packet 1
Receive packet 1
Send ACK 1
Receive ACK 1
Send packet 2
Receive packet 2
Send ACK 2
Receive ACK 2

Reliable, Ordered Delivery:
Timeout & Retransmission

Send packet 1
Start timer
Packet lost
Packet should arrive
ACK should be sent
ACK normally arrive now
Timer expires
Retransmit packet 1
Start timer
Receive packet 1
Send ACK 1
Receive ACK 1
Cancel timer
Issues in TCP

1. Fast packet loss detection
   - Retransmission timeout (RTO)
   - Duplicated acknowledgements (Dupacks)

2. Efficient delivery
   - Sliding window with cumulative acknowledgement

3. Flow problems
   - End-to-end flow problem
   - Flow problem in intermediate systems (congestion)

1. Reliable, Ordered Delivery:
   Fast Packet Loss Detection

- Two methods
  
  A. Retransmission timeout (RTO)

  A. Duplicate acknowledgements (Dupacks)
A. Fast Packet Loss Detection Using Retransmission Timeout (RTO)

- At any time, TCP sender sets retransmission timer for only one packet
- If acknowledgement for the timed packet is not received before timer goes off, the packet is assumed to be lost
- RTO (timeout value) dynamically calculated

Retransmission Timeout (RTO) Calculation

Q1: Which value will you use?
Q2: How to set the value in dynamic Internet?
Q3: If a retransmission occurs, is an ack for the original or for the retransmission (important because RTT is calculated)?

A1: RTT (round trip time)
A2: adaptive retransmission algorithm (changing RTT and thus, changing time-out values)
A3: if ACK arrives, is it for the original packet or for the retransmitted packet (acknowledge ambiguity)?
A4: timer backoff strategy
Retransmission Timeout (RTO) Calculation

- Large variations in the RTT necessitates a larger RTO (why???)

- **RTO** = mean + 4 mean deviation
  - Standard deviation $\sigma^2$ $\sigma = \text{average of (sample} - \text{mean)}$
  - Mean deviation $\delta = \text{average of } |\text{sample} - \text{mean}|$
  - Mean deviation easier to calculate than standard deviation
  - Mean deviation is more conservative: $\delta \geq \sigma$ (right??)

- Karn’s algorithm – no ack ambiguity

Retransmission: Exponential Backoff

- Double RTO on each timeout (why???)

![Diagram of retransmission exponential backoff](image)
History of TCP

- **Fathers**
  - Vinton Cerf and Robert Kahn’s paper in 1973 introduces TCP.
  - Became standard in 1983.
  - They received Turing Award in 2005.
  - They are with MCI and NRI now.

- **TCP has been improved over the years**
  - Original (83), Tahoe (88, Fast retransmit, by Van Jacobson, LBL)
  - More robust estimates of round-trip time
  - Faster recovery from packet loss
  - Congestion avoidance and improvements

- **TCP Reno**
  - Developed by Van Jacobsen in 1990
  - Added fast recovery and fast retransmit

- **TCP Vegas**
  - Developed by Brakmo and Peterson (Univ. Arizona) in 1995
  - New congestion avoidance algorithm
B. Fast Packet Loss Detection Using Dupacks

- Timeouts can take too long
  - how to detect packet loss sooner and initiate retransmission sooner?

- If the receiver receives out-of-order packet
  - What does the receiver do ??? do nothing
  - “Duplicated acks (dupacks)”
  - What are the benefits - Fast Retransmission

Dupacks

- Dupacks may be generated due to
  - packet loss, or
  - out-of-order packet delivery

- TCP sender assumes that a packet loss has occurred if it receives three dupacks consecutively (why not one/two???)

  12 11 10 9 8 7

  - Packet Loss Detection: Receipt of packets 9, 10 and 11 will each generate a dupack from the receiver.
  - Fast Retransmission: The sender, on getting these dupacks, will retransmit packet 8.
Issues in TCP

1. Fast packet loss detection
   - Retransmission timeout (RTO)
   - Duplicated acknowledgements (Dupacks)

2. Efficient delivery
   - Sliding window with cumulative acknowledgement

3. Flow problems
   - End-to-end flow problem
   - Flow problem in intermediate systems (congestion)

2. Efficient Delivery

Send packet 1
Receive packet 1
Send ACK 1
Receive ACK 1
Send packet 2
Receive packet 2
Send ACK 2
Receive ACK 2

Sending it one after the other takes too long!
Efficient Delivery: Sliding Window

- A simple positive acknowledgement protocol wastes a substantial amount of network bandwidth
- Sliding windows protocol allows the sender to transmit multiple packets before waiting for an acknowledgement

![Diagram showing sliding window protocol]

Some packets are sent and ack’ed

Some packets are sent but not ack’ed yet

Some packets are not sent yet but can be sent without delay because they are within the window

Sliding Window Protocol

What if the sender receives ACK for packet 3?

What does it mean if the window size is 1?

What does it mean if the window size is $\infty$?
Sliding Window (Example): the maximum range of data sent but not acknowledged

An Example: Sliding Window Size = 4 bytes

```plaintext
[0 1 2 3] 4 5 6 7 8 9
          0 1 2 3
ACK 0
0 [1 2 3 4] 5 6 7 8 9
        4
Timeout
0 [1 2 3 4] 5 6 7 8 9
        1 2 3 4
```

**Sliding Window Basics**

- **Cumulative acknowledgements**
  - Good: Easy to generate and unambiguous
  - Bad: the sender does not receive information about all successful transmissions, but only about a single position in the stream that has been received

- An acknowledgement ack's all contiguously received data

- TCP assigns byte sequence numbers
Cumulative Acknowledgements

- A new cumulative acknowledgement is generated only on receipt of a **new in-sequence** packet.

```
S: 40 39 38 37
33 34 35 36

R: 41 40 39 38
34 35 36 37
```

One More Optimization

- If the sender receives ACK_{36}
  - It means ???
  - Do we need to send acks for all packets?
  - “Delayed acks” (e.g., ACK2, ACK4, ACK6, ...)

```
S: 40 39 38 37
34 35 36 37

R: 40 39 38 37
34 35 36
```
Delayed ACKs

Issues in TCP

1. Fast packet loss detection
   - Retransmission timeout (RTO)
   - Duplicated acknowledgements (Dupacks)

2. Efficient delivery
   - Sliding window with cumulative acknowledgement

3. Flow problems
   - End-to-end flow problem
   - Flow problem in intermediate systems (congestion)
3. Flow Problems

A. End-to-end flow problem
- Control a source that sends more traffic than the receiver can tolerate
- Between source and destination
  => Easier to solve: The receiver informs its status to the sender
  ("advertised window size")

B. Flow problem in the intermediate systems
- Control a source that sends more traffic than the intermediate systems can tolerate
- Called "congestion"

Window-based Flow Control

- Congestion simply means increased delay
  - Respond by retransmissions
  - Incurs more congestion
  - Thus, we need to reduce the window size

- Window size is determined as the minimum of
  - receiver’s advertised window - determined by available buffer space at the receiver <= "flow control"
  - congestion window (cwnd) - determined by the sender, based on feedback from the network <= "congestion control"

- On detecting a packet loss, TCP sender
  - assumes that network congestion has occurred
  - drastically reduces the congestion window
TCP Flow Control

- TCP inherently supports flow control to prevent buffer overflow at the receiver
  - Useful for fast sender transmitting to slower receiver
- Receiver advertises a window \((wnd)\) in acknowledgements returned to the sender
- Sender cannot send more than \(wnd\) unacknowledged bytes to the receiver

TCP Congestion Avoidance

- Congestion avoidance (control) was added to TCP in an attempt to reduce congestion inside the network
- A much harder problem ...
  - Requires the cooperation of multiple senders
  - Must rely on indirect measures of congestion
- Implemented at sender
TCP Operation

- Flow control (already discussed)
- Congestion control
  - **Key Assumption**: Packet loss is due to congestion
  - Reduces the number of segments sent when detects losses
  - Introduce a congestion window ($cwnd$), in addition to flow control window ($wnd$)
  - Need to manage size of congestion window

- Three stages
  - “Multiplicative decrease” (when congestion is detected)
  - “Slow Start” (when congestion ends)
  - “Congestion avoidance” (when $cwnd$ reaches a reasonable value)

Determining Window Size for Congestion Control

- Congestion control used in TCP ($cwnd$ changes)
  - Multiplicative decrease
  - Slow start
    - Starts with a slow transmission rate at the start of a new connection or after a congestion condition
    - When congestion ends, it can increase the window size multiplicatively. But it may result in oscillation.
    - “Additive” recovery (increase one for each ACK arrived)
    - But in fact, $cwnd$ grows *exponentially* up to threshold (half the window size before packet loss)
  - Congestion avoidance
    - $cwnd$ increases “linearly” with time (increase one if all ACKs arrive)
    - Rate of increase could be lower if sender does not always have data to send
Format of a TCP Segment

Figure 13.7 The format of a TCP segment with a TCP header followed by data. Segments are used to establish connections as well as to carry data and acknowledgements.
<table>
<thead>
<tr>
<th>Decimal</th>
<th>Keyword</th>
<th>UNIX Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>TCP Multiplexor</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TCPMUX</td>
<td>echo</td>
<td>Echo</td>
</tr>
<tr>
<td>9</td>
<td>DISCARD</td>
<td>discard</td>
<td>Discard</td>
</tr>
<tr>
<td>11</td>
<td>USERS</td>
<td>systat</td>
<td>Active Users</td>
</tr>
<tr>
<td>15</td>
<td>NETSTAT</td>
<td>netstat</td>
<td>Network Status Program</td>
</tr>
<tr>
<td>17</td>
<td>QUOTE</td>
<td>qotd</td>
<td>Quote of the Day</td>
</tr>
<tr>
<td>19</td>
<td>CHAP</td>
<td>chargen</td>
<td>Character Generator</td>
</tr>
<tr>
<td>20</td>
<td>FTP-DATA</td>
<td>ftp-data</td>
<td>File Transfer Protocol (data)</td>
</tr>
<tr>
<td>21</td>
<td>FTP</td>
<td>ftp</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>22</td>
<td>SSH</td>
<td>ssh</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>23</td>
<td>TELNET</td>
<td>telnet</td>
<td>Terminal Connection</td>
</tr>
<tr>
<td>25</td>
<td>SMTP</td>
<td>smtp</td>
<td>Simple Mail Transport Protocol</td>
</tr>
<tr>
<td>37</td>
<td>TIME</td>
<td>time</td>
<td>Time</td>
</tr>
<tr>
<td>53</td>
<td>DOMAIN</td>
<td>nameserver</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>67</td>
<td>BOOTPS</td>
<td>bootps</td>
<td>BOOTP or DHCP Server</td>
</tr>
<tr>
<td>77</td>
<td>rje</td>
<td>rje</td>
<td>Any private RJE service</td>
</tr>
<tr>
<td>79</td>
<td>FINGER</td>
<td>finger</td>
<td>Finger</td>
</tr>
<tr>
<td>80</td>
<td>WWW</td>
<td>www</td>
<td>World Wide Web Server</td>
</tr>
<tr>
<td>88</td>
<td>KERBEROS</td>
<td>kerberos</td>
<td>Kerberos Security Service</td>
</tr>
<tr>
<td>95</td>
<td>SUPDLP</td>
<td>supdup</td>
<td>SUPDLP Protocol</td>
</tr>
<tr>
<td>101</td>
<td>HOSTNAME</td>
<td>hostnames</td>
<td>NIC Host Name Server</td>
</tr>
<tr>
<td>102</td>
<td>ISO-TSAP</td>
<td>iso-tsap</td>
<td>ISO-TSAP</td>
</tr>
<tr>
<td>103</td>
<td>X400</td>
<td>x400</td>
<td>X.400 Mail Service</td>
</tr>
<tr>
<td>104</td>
<td>X400</td>
<td>x400</td>
<td>X.400 Mail Sending</td>
</tr>
<tr>
<td>110</td>
<td>POP3</td>
<td>pop3</td>
<td>Post Office Protocol Vers. 3</td>
</tr>
<tr>
<td>111</td>
<td>SUNRPC</td>
<td>sunrpc</td>
<td>SUN Remote Procedure Call</td>
</tr>
<tr>
<td>113</td>
<td>AUTH</td>
<td>auth</td>
<td>Authentication Service</td>
</tr>
<tr>
<td>117</td>
<td>UUCP-P</td>
<td>uucp-path</td>
<td>UUCP Path Service</td>
</tr>
<tr>
<td>118</td>
<td>NETBIOS-SSN</td>
<td></td>
<td>NETBIOS Session Service</td>
</tr>
<tr>
<td>161</td>
<td>SNMP</td>
<td>snmp</td>
<td>Simple Network Management Protocol</td>
</tr>
</tbody>
</table>

Figure 13.16 Examples of currently assigned TCP port numbers. To the extent possible, protocols like UDP use the same numbers.