TCP/IP Protocols:
Outline

Yang Richard Yang

9/12/2001

Review
### The Goals of the DARPA Internet

0. **Connect different networks**

1. Survivability in the face of failure
2. Support multiple types of service
3. Accommodate a variety of networks
4. Permit distributed management of resources
5. Be cost effective
6. Permit host attachment with a low level of effort
7. Be accountable

---

### The End-to-End Arguments

The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication systems. Therefore, providing that questioned function as a feature of the communications systems itself is not possible.

J. Saltzer, D. Reed, and D. Clark, 1984
### Internet Architecture

![Diagram showing IP, TCP, UDP, Telnet, Email, FTP, WWW, Ethernet, Wireless, FDDI]

### Comparisons Between Datagram and Virtual Circuit

<table>
<thead>
<tr>
<th></th>
<th>Datagram</th>
<th>Virtual Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup</strong></td>
<td>No</td>
<td>Required</td>
</tr>
<tr>
<td><strong>Addressing</strong></td>
<td>Full address</td>
<td>A short VC number</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>No</td>
<td>Each established VC needs table space</td>
</tr>
<tr>
<td><strong>Routing</strong></td>
<td>Routing for each packet is independent</td>
<td>Routing chosen at setup; all packets same route</td>
</tr>
<tr>
<td><strong>Failure</strong></td>
<td>None; except during failure</td>
<td>All VC passing through the node</td>
</tr>
<tr>
<td><strong>Congestion control</strong></td>
<td>End-to-end</td>
<td>Router implementation</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>Transport: end host</td>
<td>Network</td>
</tr>
<tr>
<td><strong>Transport service</strong></td>
<td>Connection-oriented and connectionless</td>
<td>Connection-oriented</td>
</tr>
</tbody>
</table>
TCP/IP: Initial → Current → NG

Network Layer

- Reminder: service, interface, and protocol
- What services does the network layer (IP) provide?
- How to design a protocol to implement the services?
  - data plane
  - control plane
Addressing

Cerf & Kahn TCP/IP addressing (CK74, 1974)

- Length: 24bits
- Organization: hierarchical

<table>
<thead>
<tr>
<th></th>
<th>16</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>TCP Identifier</td>
<td></td>
</tr>
</tbody>
</table>
IPv4 Addressing (standardized in 1981)

- Length: 32 bits (RFC 791)
- Organization: hierarchical

```
<table>
<thead>
<tr>
<th>Class</th>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>C</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>D</td>
<td>111</td>
<td>28</td>
</tr>
<tr>
<td>E</td>
<td>1111</td>
<td>27</td>
</tr>
</tbody>
</table>
```

Revisions to IPv4 Addressing

- Add subnet mask in 1984
- Classless Inter Domain Routing (CIDR)
IPv6 Addressing (standardized in 1995)

- Length: 128 bits (RFC 1752)
- Organization: hierarchical

<table>
<thead>
<tr>
<th></th>
<th>010</th>
<th>n</th>
<th>m</th>
<th>α</th>
<th>p</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider-based unicast</td>
<td>10</td>
<td>n</td>
<td>118-n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link-local use unicast</td>
<td>10</td>
<td>n</td>
<td>m</td>
<td>118-n-m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site-local use unicast</td>
<td>11111111010</td>
<td>0</td>
<td>Interface ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Embedded IPv4 unicast</td>
<td>80</td>
<td>16</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>112</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other Addressing Schemes

- ISO/OSI NSAP (Network Service Access Point)
  - variable length (up to 20 bytes)
  - multiple class

<table>
<thead>
<tr>
<th>AFI (Authority and Format Id)</th>
<th>IDI (Initial Domain Id)</th>
<th>DSP (Domain Specific Part)</th>
</tr>
</thead>
</table>

- What are the assumptions of IPv4 and IPv6 addressing scheme?
- Can you think of a new scheme?
Network Layer Protocol

Original TCP/IP Segment/Packet Format

- **Packet format**
  - 24: Source
  - 24: Destination
  - 16: Seq. no.
  - 16: Byte Count
  - 16: Flags
  - 8n: Text
  - 16: Checksum

- **Segment format**
  - 32: Source port
  - 32: Destination port
  - 16: Window
  - 16: ACK
  - 8n: Text
**IPv4 Header**

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>HLen</td>
<td>TOS</td>
<td>Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>Flags</td>
<td>Fragment offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>Protocol</td>
<td>Header checksum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source address</td>
<td>Destination address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **HLen** - header length in 32-bit words (5 ≤ HLen ≤ 15)
- **TOS (Type of Service)**: now split into
  - differentiated Service Field (6 bits)
  - remaining two bits used by ECN (Early Congestion Notification)
- **Length** - the length of the entire datagram/segment; header + data
- **Flags**: Don’t Fragment (DF) and More Fragments (MF)
- **Fragment offset** - all fragments excepting last one contain multiples of 8 bytes
- **Header checksum** - uses 1’s complement

**IPv4 Options**

- Every router must process
- Length limited to ≤ 40
- Example:
  - Source routing

```
1 1 1 4 4 4
```

- code: 0x83 loose source routing; 0x89 strict
**IPv6 Header (1998)**

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>12</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Priority</td>
<td>Flow label</td>
<td></td>
</tr>
<tr>
<td>Payload length</td>
<td>Next header</td>
<td>Hop limit</td>
<td></td>
</tr>
</tbody>
</table>

Source address

Destination address

Options (variable)

---

**The Transition from IPv4 → IPv6:**

*An example of multiple network protocols*

- What need to be changed when we change from IPv4 to IPv6?
- Discussion: How to introduce IPv6 into the current IPv4 Internet?
- Why is the deployment of IPv6 slow? What may drive the deployment?
Transport Layer

- What services does TCP provide?
- How to design a protocol to implement the services?

Original TCP/IP Segment/Packet Format

Packet format

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Seq. no.</th>
<th>Byte Count</th>
<th>Flags</th>
<th>Text</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8n</td>
<td>16</td>
</tr>
</tbody>
</table>

flags:
- SYN: synchronizing to packet seq. no.
- REL: release of Process/Port
- ES: end of segment
- EM: end of message

Segment format

<table>
<thead>
<tr>
<th>Source port</th>
<th>Destination port</th>
<th>Window</th>
<th>ACK</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>8n</td>
</tr>
</tbody>
</table>
TCP Header

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>10</th>
<th>16</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source port</td>
<td>Destination port</td>
<td>Sequence number</td>
<td>Acknowledgement</td>
<td>HdrLen</td>
</tr>
<tr>
<td>Flags</td>
<td>Advertised window</td>
<td>Checksum</td>
<td>Urgent pointer</td>
<td>Options (variable)</td>
</tr>
</tbody>
</table>

- Flags:
  - SYN, FIN - establishing/terminating a TCP connection
  - ACK - set when Acknowledgement field is valid
  - URG - urgent data; Urgent Pointer says where non-urgent data starts
  - PUSH - don't wait to fill segment
  - RESET - abort connection

TCP Header (Cont')

- Checksum - 1's complement and is computed over
  - TCP header
  - TCP data
  - Pseudo-header (from IP header)
    - Note: breaks the layering!

<table>
<thead>
<tr>
<th>Source address</th>
<th>Destination address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Protocol (TCP) TCP Segment length</td>
<td></td>
</tr>
</tbody>
</table>
What Were Missing?

- **Connection setup**
  - goal: agree on a set of parameters: the start sequence number for each side

- **Connection teardown**
  - goal: agree on a set of parameters: the start sequence number for each side

- **Congestion control**

What Would the Segment of A Connectionless Protocol (UDP) Look Like?