Today’s Lecture

I. Port Numbers
II. UDP function
III. UDP header

What Layer is UDP?

Interprocess Communication

- Distributed applications mean communication between services on different hosts
- Services vs. processes
  - processes are created and destroyed dynamically by the OS
  - a single process may provide multiple services; must be clear which service the requestor needs
  - binding of “service required” and “process that handles this service” should be determined by the server

Protocol Ports

- Port ≡ abstract source or destination service specifier
  - identified by an unsigned 16-bit integer
  - packets carry both source and destination ports
- OS provides an interface for processes to bind to or associate with ports
UDP Ports Example

Host A
- Application Client
  - Port 34891

Host B
- Application Server
  - Port 13

Data flow:
- A/34891 -> B/13
- B/13 -> A/34891

UDP Input Demultiplexing

Host B
- Daytime Server
  - Port 13
- Telnet Server
  - Port 23
- TFTP Server
  - Port 69

Demultiplexer

UDP / TCP Port Number Space

Port #: 0 1023 1024 49151 49152 65535

“Register-able”

“Dynamic (private)”

“Well-known (assigned by IANA)”

Examples of Well-Known UDP Ports

<table>
<thead>
<tr>
<th>PORT #</th>
<th>SERVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>ECHO</td>
</tr>
<tr>
<td>9</td>
<td>DISCARD</td>
</tr>
<tr>
<td>13</td>
<td>DAYTIME</td>
</tr>
<tr>
<td>17</td>
<td>Quote of the Day</td>
</tr>
<tr>
<td>23</td>
<td>Telnet</td>
</tr>
<tr>
<td>53</td>
<td>DNS</td>
</tr>
<tr>
<td>68</td>
<td>BOOTPC</td>
</tr>
<tr>
<td>69</td>
<td>TFTP</td>
</tr>
<tr>
<td>123</td>
<td>NTP</td>
</tr>
<tr>
<td>161</td>
<td>SNMP</td>
</tr>
</tbody>
</table>

- See [http://www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers) for more

User Datagram Protocol (UDP, RFC 768)

- Same service (or lack of) as IP
  - connectionless, no state maintained
  - unreliable, no notification of arrival (datagram may have been discarded, duplicated, or delivered out of order)
  - no flow control (source may transmit faster than destination is able to receive)

- UDP adds to IP
  - source and destination ports
  - payload checksum (optional) for data integrity
  - nothing else!
### When Should UDP Be Used?

1. When reliable delivery is unimportant (non-critical application), or the application layer provides reliability
2. When flow control is unimportant (there are only a few datagrams to send), or the application layer provides flow control
3. When TCP characteristics (retransmission delay, processing and packet overhead) are unacceptable

### UDP or TCP?

- File transfer
- Multicast and broadcast of announcements
- Streaming audio and video
- Routing protocols
- Web access (HTTP)
- Instant messaging (text chat)
- Examples of "UDP-able" applications?

### UDP Service Binding

1. Application programs negotiate with the OS to be listed as the service bound to a port
2. OS creates internal queue to hold arriving messages for the port

### UDP Service Binding (cont’d)

Upon receipt of a datagram, UDP checks if destination port is currently "active" (bound to a running process)

- **if** no process is receiving on this port
  - send ICMP "Destination Unreachable / Port Not Bound to Service" error message and discard datagram
- **else if** receiving queue is full
  - discards datagram (no error message sent back)
- **else**
  - enqueue datagram for receiving process

### UDP Datagram Format

- Source Port #, Destination Port #
- Total Length: of UDP header and data
  - why needed? what is max value?
- UDP Checksum

### UDP HEADER

<table>
<thead>
<tr>
<th>Source Port Number</th>
<th>Destination Port Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP Total Length</td>
<td>UDP Checksum</td>
</tr>
<tr>
<td>UDP payload (data)</td>
<td></td>
</tr>
</tbody>
</table>
(Remember…)

**UDP Checksum**
- Checksum: across *pseudo-header, UDP header, and data*
  - why needed?
  - includes IP layer???

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Source Address</td>
<td>IP Destination Address</td>
<td>IP Total Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--all 0's--</td>
<td>17</td>
<td>(protocol ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Port Number</td>
<td>Destination Port Number</td>
<td>UDP Total Length</td>
<td>UDP Checksum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDP payload (data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**UDP Checksum (cont’d)**
- Same algorithm as IP checksum
- If received checksum is incorrect, datagram is silently discarded (no error message sent back)

**Summary**
1. UDP provides
   1. checksum on the payload, for data integrity
   2. port numbers, to identify what process should receive the incoming data
2. That’s all!
3. Unreliable delivery, same as the underlying IP layer

**Next Lecture**
- Transmission Control Protocol (TCP)