Linux Networking: udp

David Morgan

© David Morgan 2003

UDP

- Stateless, unreliable transport protocol with no delivery guarantee
- performance over reliability
- well suited to broadcast and discovery type messaging
- RFC 768 and RFC 1122 (STD 6)

© David Morgan 2003
UDP Ports

- UDP multiplexes among various upper layer protocols by using PORTS
- 16 bit port numbers are assigned to specific applications by UDP
- Some UDP ports use "Well Known" numbers (53 DNS, 69 TFTP etc)

UDP header

<table>
<thead>
<tr>
<th>Ethernet</th>
<th>IP</th>
<th>UDP</th>
</tr>
</thead>
</table>

UDP = 8 bytes

<table>
<thead>
<tr>
<th>Srce Port</th>
<th>Dest Port</th>
<th>Length</th>
<th>Checksum</th>
</tr>
</thead>
</table>
UDP Header

- Source Port - originator of the data
  - Not required can legally be 0
- Destination Port - destination application
- Length - length of the UDP header and following data
- Checksum - over IP "pseudo header", UDP header and data
  - Optional - can be left 0xFFFF

UDP checksum

- To detect errors (sent-vs-received mismatch)
- Sender algorithm
  - sum all 16-bit words in packet
  - take binary 1’s-complement
  - place in checksum field
- Receiver algorithm
  - sum all 16-bit words in packet
  - add to that the checksum
  - result should be 1111111111111111
Algorithm example

data:
in ASCII-  A   B   C   D
in binary- 01000001 01000010 01000011 01000100

sum:  \[ \begin{align*}
01000001 & 01000010 \\
01000011 & 01000100 \\
10000100 & 10000110
\end{align*} \]

1’s-comp: \[ \begin{align*}
01111011 & 01111001
\end{align*} \]

(checksum) \[ \begin{align*}
01111011 & 01111001
\end{align*} \]

these add to \[ \begin{align*}
11111111 & 11111111
\end{align*} \]
so had better sum of received data plus received checksum

Who uses UDP instead of TCP?

- streaming applications
- discovery tools
- certain application protocols
  - DNS
  - TFTP
  - traceroute
UDP Trace (traceroute –m1 www.ucla.edu)

DNS using UDP

traceroute using UDP