INTRODUCTION TO USER DATAGRAM PROTOCOL, A SIMPLE PACKET TRANSPORT SERVICE IN THE INTERNET PROTOCOL SUITE

PETER R. EGLI
INDIGOO.COM
UDP - User Datagram Protocol

Contents
1. UDP (RFC768) characteristics
2. UDP Service
3. Typical UDP applications
4. UDP Checksum
5. UDP versus TCP
1. UDP (RFC768) characteristics

- No connection establishment/teardown; data is just sent right away.
- No flow control / congestion control, sender can overrun receiver's buffer:
  - UDP is not suited for bulk data transfer.
  - For data transfer with UDP a lock-step protocol is required (to be implemented by the application).
- No error control; corrupted data is not retransmitted (even though UDP header has a checksum to detect errors and report these to the application).

Packet 4 is dropped (buffer overrun).
Packet 4 is missing. The packet loss needs to be handled by the application (detection, retransmission).
UDP - User Datagram Protocol

2. UDP service

- UDP is basically a simple extension of the IP datagram service.
- UDP adds multiplexing (on port number) to IP datagram service.
- Application writes are mapped 1:1 to UDP datagrams; UDP passes these 1:1 to the IP layer.

2500-byte packet is too big. IP layer fragments the UDP datagram.
3. Typical UDP applications

UDP is best suited for applications with short command-response type „transactions“ that do not justify the establishment / release prior to the data exchange.

- **SNMP**
  - Host 1: SNMP GET Request
  - Host 2: SNMP GET Response

- **DNS**
  - Host 1: DNS Query
  - Host 2: DNS Response

- **SNTP**
  - Host 1: Time request
  - Host 2: Time response

- **DHCP/Bootp**
  - Host 1: DHCP Request
  - Host 2: DHCP Response

- **RADIUS**
  - Host 1: RADIUS Access Request
  - Host 2: RADIUS Access Accept

- **RIP**
  - Host 1: RIP Route Update
  - Host 2: 

- **RPC**
  - Host 1: RPC Request
  - Host 2: RPC Reply
4. UDP checksum

- UDP has a checksum too that provides minimal protection against transmission errors.
- The checksum is optional; if it is not used it shall be set to 0.
- Because the IP addresses are used in the UDP checksum calculation, UDP is tightly bound to the IP layer. Therefore UDP can only run on top of IP.

## Pseudo header

The checksum is calculated over the pseudo header, UDP header and data (UDP payload).

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

<table>
<thead>
<tr>
<th>IP Source Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Destination Address</td>
</tr>
<tr>
<td>00000000</td>
</tr>
<tr>
<td>Pseudo header</td>
</tr>
<tr>
<td>UDP header</td>
</tr>
</tbody>
</table>

| Data |
5. UDP versus TCP

TCP

- Connection-oriented, point-to-point (unicast)
- Reliable end-to-end:
  - No bit errors due to checksum.
  - Packet ordering preserved.
  - No duplicates.
  - No packet loss.
- Stream-oriented (no message boundary preservation)
- Has flow control to maximise throughput
- Has congestion control to minimise packet loss
- Analogon: phone

Examples of application protocols using TCP:
HTTP, SMTP, FTP, TELNET

UDP

- Connection-less, best-effort
- Not reliable (no retransmissions)
- Message boundary preservation
- No flow control
- No congestion control
- Analogon: mail (snail mail)

Examples of application protocols using UDP:
SNMP, DNS, TFTP, RTP, DHCP, SNTP

N.B.: It is possible to run application protocols over both TCP and UDP. E.g. DNS is normally run on UDP, but for zone transfers (higher data volume) DNS uses TCP. Actually there is a shift towards using TCP instead UDP since TCP can better provide security (SSL/TLS, simpler filtering in firewalls etc.).