Internet Protocols

Transport Layer

2013, Lecture 05
Kaan Bür, Stefan Höst
Previously on EITF25

Network Layer

- Internet architecture
  - End-to-end principle, routing
- Internet Protocol (IPv4, IPv6)
  - Addressing, forwarding
  - Datagram format
  - Fragmentation
- Address Resolution Protocol

[Forouzan ed.5 ch.9.2]
Address Resolution Protocol (ARP)

- Mapping of IP addresses to MAC addresses
- Internet
  - Network of networks connected by routers
- Routers/hosts need information
  - Logical (IP) → physical (MAC)
ARP packet

<table>
<thead>
<tr>
<th></th>
<th>Hardware Type</th>
<th>Protocol Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hardware length</td>
<td>Protocol length</td>
</tr>
</tbody>
</table>

- **Sender hardware address**
  (For example, 6 bytes for Ethernet)

- **Sender protocol address**
  (For example, 4 bytes for IP)

- **Target hardware address**
  (For example, 6 bytes for Ethernet)
  (It is not filled in a request)

- **Target protocol address**
  (For example, 4 bytes for IP)

**ARP request or reply packet**

- **Type:** 0x0806
- **Preamble and SFD:** 8 bytes
- **Destination address:** 6 bytes
- **Source address:** 6 bytes
- **Type:** 2 bytes
- **Data:**
- **CRC:** 4 bytes
ARP operation

- ARP query broadcast every time a host/router needs a MAC address
- Intended host answers with an ARP response
- ARP cache (table) used to store MAC/IP pairs

- Some IP addresses known from start
  - Default gateway (router) →  "rest of Internet"
  - DNS server
ARP request and reply

Looking for physical address of a node with IP address 141.23.56.23

a. ARP request is broadcast

The node physical address is A4:6E:F4:59:83:AB

b. ARP reply is unicast
ARP example

ARP request:
- Multicast frame
  - M: Broadcast address
  - Destination: L1, N1, All 0s, N2
  - Source: A

ARP reply:
- Unicast frame
  - Data
  - Destination: A
  - Source: B

From A to All:
1. ARP request
2. ARP reply

From B to A:
1. ARP request
2. ARP reply
Four use cases for ARP

Case 1. A host has a packet to send to another host on the same network.

Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.

Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

Case 4. A router receives a packet to be sent to a host on the same network.
Today: Internet Protocols

Transport Layer

- Introduction
  [Forouzan ed.5 ch.23.1]
- Transmission Control Protocol
  [Forouzan ed.5 ch.24.3]
- User Datagram Protocol
  [Forouzan ed.5 ch.24.2]

*[Kihl & Andersson: 10]*
TCP/IP model and data units

- Application layer: Processes
- Transport layer: SCTP, TCP, UDP
- Network layer: IP and other protocols
- Data link layer: Underlying physical networks
- Physical layer

Data units:
- SEGMENT
- DATAGRAM
- PACKET
- FRAME
Network Layer

• L3

• L4?
Transport Layer

• Communication between applications
• Process-to-process delivery
• Client/server concept
  – Local host
  – Remote host
• Transport Protocol
  – Even more end-to-end
Transport Layer

Diagram showing the layers of the transport protocol, from physical to application layer, with intermediate nodes and peer-to-peer protocol at each layer.
Logical end-to-end connection

- Transport protocol creates a logical (virtual) connection between source and destination.
Transport protocol

- Encapsulates application data and ensures that it is sent to the correct receiving application to be decapsulated and used
Process-to-Process Delivery

• Multiple applications even on the same host
Multiplexing and demultiplexing

- **Socket addresses** allow multiplexing and demultiplexing multiple applications’ data
Socket addresses

- Combination of IP address & port number
  - Unique for each process on the host
Port numbers

Internet Cooperation for Assigned Names and Numbers Authority (ICANN) defines 3 types

- **Well-known** ports, assigned and controlled by Internet Assigned Numbers Authority (IANA)
- **Registered** ports, to be registered with IANA to prevent duplication
- **Dynamic** (a.k.a. ephemeral) ports, neither controlled nor registered, to be used by any application
Port number ranges

0  1023  1024  49152  65535

Well known

Registered

TCP

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Echo</td>
<td>Echoes a received datagram back to the sender</td>
</tr>
<tr>
<td>9</td>
<td>Discard</td>
<td>Discards any datagram that is received</td>
</tr>
<tr>
<td>11</td>
<td>Users</td>
<td>Active users</td>
</tr>
<tr>
<td>13</td>
<td>Daytime</td>
<td>Returns the date and the time</td>
</tr>
<tr>
<td>17</td>
<td>Quote</td>
<td>Returns a quote of the day</td>
</tr>
<tr>
<td>19</td>
<td>Chargen</td>
<td>Returns a string of characters</td>
</tr>
<tr>
<td>20</td>
<td>FTP, Data</td>
<td>File Transfer Protocol (data connection)</td>
</tr>
<tr>
<td>21</td>
<td>FTP, Control</td>
<td>File Transfer Protocol (control connection)</td>
</tr>
<tr>
<td>23</td>
<td>TELNET</td>
<td>Terminal Network</td>
</tr>
<tr>
<td>25</td>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>53</td>
<td>DNS</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>67</td>
<td>BOOTP</td>
<td>Bootstrap Protocol</td>
</tr>
<tr>
<td>79</td>
<td>Finger</td>
<td>Finger</td>
</tr>
<tr>
<td>80</td>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>111</td>
<td>RPC</td>
<td>Remote Procedure Call</td>
</tr>
</tbody>
</table>

UDP

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Echo</td>
<td>Echoes a received datagram back to the sender</td>
</tr>
<tr>
<td>9</td>
<td>Discard</td>
<td>Discards any datagram that is received</td>
</tr>
<tr>
<td>11</td>
<td>Users</td>
<td>Active users</td>
</tr>
<tr>
<td>13</td>
<td>Daytime</td>
<td>Returns the date and the time</td>
</tr>
<tr>
<td>17</td>
<td>Quote</td>
<td>Returns a quote of the day</td>
</tr>
<tr>
<td>19</td>
<td>Chargen</td>
<td>Returns a string of characters</td>
</tr>
<tr>
<td>53</td>
<td>Nameserver</td>
<td>Domain Name Service</td>
</tr>
<tr>
<td>67</td>
<td>BOOTPs</td>
<td>Server port to download bootstrap information</td>
</tr>
<tr>
<td>68</td>
<td>BOOTPc</td>
<td>Client port to download bootstrap information</td>
</tr>
<tr>
<td>69</td>
<td>TFTP</td>
<td>Trivial File Transfer Protocol</td>
</tr>
<tr>
<td>111</td>
<td>RPC</td>
<td>Remote Procedure Call</td>
</tr>
<tr>
<td>123</td>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>161</td>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>162</td>
<td>SNMP</td>
<td>Simple Network Management Protocol (trap)</td>
</tr>
</tbody>
</table>
Addressing the processes

• Port numbers
  – Organised by IANA
IP addresses and port numbers
Logical and port addresses

Diagram showing the flow of data through different layers of the internet protocol stack, from logical and port addresses to the internet.
Addressing in TCP/IP

Application layer
- Processes

Transport layer
- SCTP
- TCP
- UDP

Network layer
- IP and other protocols

Data link layer
- Underlying physical networks

Physical layer
- Physical addresses

Port addresses

Logical addresses

Physical addresses
See you in 15’ :) 

- After the break
  - TCP
  - QoS
  - UDP
Transmission Control Protocol (TCP)

- Connection-oriented
  - Sessions
  - Byte stream service
    - Sequence numbers
- Reliable
  - Flow control
  - Error control
    - Retransmissions
  - Congestion control
Connection-oriented service
## TCP header format

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source port address</td>
<td>16 bits</td>
<td>Source address of the sender</td>
</tr>
<tr>
<td>Destination port address</td>
<td>16 bits</td>
<td>Destination address of the receiver</td>
</tr>
<tr>
<td>Sequence number</td>
<td>32 bits</td>
<td>Sequence number for the packet</td>
</tr>
<tr>
<td>Acknowledgment number</td>
<td>32 bits</td>
<td>Acknowledgment number for the packet</td>
</tr>
<tr>
<td>HLEN</td>
<td>4 bits</td>
<td>Header length extension</td>
</tr>
<tr>
<td>Reserved</td>
<td>6 bits</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>URG, ACK,PSH,RST,SYN,FIN</td>
<td>6 bits</td>
<td>Flags and controls</td>
</tr>
<tr>
<td>Window size</td>
<td>16 bits</td>
<td>Window size for the receiver</td>
</tr>
<tr>
<td>Checksum</td>
<td>16 bits</td>
<td>Checksum for data integrity</td>
</tr>
<tr>
<td>Urgent pointer</td>
<td>16 bits</td>
<td>Urgent pointer for data delivery</td>
</tr>
<tr>
<td>Options and Padding</td>
<td></td>
<td>Options and padding</td>
</tr>
</tbody>
</table>
Exercise: Fill in the header.

0532 0017 00001234 00004321 5 002 07FF

<table>
<thead>
<tr>
<th>Source port address</th>
<th>0532</th>
<th>Destination port address</th>
<th>0017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acknowledgment number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Urgent pointer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options and Padding</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sending and receiving buffers...
... turned into Segments

Sequence numbers
TCP operation

• Connection establishment
  – Three-way handshake

• Data transfer
  – Flow control (→ congestion control)
  – Error control (→ go back N with selective repeat)

• Connection termination
  – Three-way handshake
  – Half-close
Connection establishment

A: ACK flag
S: SYN flag

seq: 8000
SYN

seq: 15000
ack: 8001
SYN + ACK

seq: 8000
ack: 15001
ACK
Data transfer
Connection termination

A: ACK flag
F: FIN flag

seq: x
ack: y
FIN

seq: y
ack: x + 1
FIN + ACK

seq: x
ack: y + 1
ACK

Active close

Passive close

Time
Half-close
Error control

- Reliable transport layer service: TCP
- Unreliable network layer service: IP
Error control in TCP

- Checksum
- Acknowledgement
  - ACK received data
- Retransmission
  - After time-out
  - After 3 duplicate ACK
Normal operation
Lost segment

Diagram showing the sequence of events for lost segments in a network protocol. The diagram illustrates the interaction between a sender and a receiver, highlighting the sequence numbers, acknowledgments, and the process of retransmission after a time-out.
Fast retransmission
Quality of Service (QoS)

• Maintaining a functioning network
  – Meeting applications’ demands
  – Dealing with flow characteristics

• Particularly important for real-time apps
  – Multimedia
Where to improve QoS?

- Admission control
  - INTSERV, DIFFSERV
- Resource reservation
  - RSVP

- Scheduling
- Traffic shaping

+ underlying WAN technologies:
  ATM

ANYWHERE YOU FIND QUEUES!
Congestion avoidance

• Congestion = data load > network capacity
  – Arrival rate > processing rate
  – Processing rate > departure rate

• Congestion control
### Summary and comparison: QoS

**Multimedia Performance Requirements**

- Sensitive to:
  - Delay
  - Jitter
- Not so sensitive to:
  - Packet loss
  - Corrupted packets

**vs. Characteristics of TCP**

- Sensitive to:
  - Lost or corrupted packets
- Not so sensitive to:
  - Delay
- No multicasting!

---

*So, what about UDP?*
User Datagram Protocol (UDP)

- **Connectionless**
  - Independent datagrams
  - No sessions

- **Unreliable**
  - No error control
  - No flow control

- **Process-to-process**
Connectionless service

![Diagram showing the flow of messages and packets in a connectionless service. Messages 0, 1, and 2 are sent from the client process to the server process. Packet 0, Packet 1, and Packet 2 are transmitted between them. Message 2 is delivered out of order.]

2013-11-15 EITF25 – Internet: Technology and Applications 47
UDP header format

- Checksum optional
- No numbering
  - No relation between datagrams
Real-time Transport Protocol

Application layer:
- PCM
- H.261
- MPEG audio
- MPEG1 video
- MPEG2 video
- ...
- Motion JPEG

Transport layer:
- RTP
- UDP

Network layer:
- IP

Data link layer:
- Underlying LAN or WAN technology

Physical layer:

2013-11-15
Summary: Internet Protocols (2)

• Mapping IP addresses to MAC addresses
• Addressing beyond IP
  – Ports, sockets
• Process-to-process delivery
• Transport layer protocols
  – TCP: connection-oriented, reliable
  – UDP: connectionless, unreliable
• Quality of Service