CS419: Computer Networks

Lecture 1 (part 2): Jan 26, 2004
Intro to Computer Networking
Remember this picture?

- How did the switch know to forward some packets to B and some to D?
From the address in the packet header…

- A packet has a header and a body
  - and, sometimes, a trailer
- The header says:
  - Where the packet is going (*address*)
  - How big the packet is (*length*)
  - Some other stuff
Packets

Body  Header  Body  Header

Tells where the next packet is

Length  Address

Tells where to route the packet
Like an envelope?

- The address field is somewhat analogous to the address on an envelope
  - And the contents of the envelope would then be like the packet body
- But this analogy doesn’t work for the length field!
Forwarding Table

- Routers (or switches) have a forwarding table
  - Router is a forwarding box that operates on IP packets
- This table is indexed by the address in the header, and tells which next hop to send the packet to
- Addresses can be hierarchical (like phone numbers)
Forwarding Table

A's Forwarding Table

<table>
<thead>
<tr>
<th>Addr</th>
<th>NH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>100</td>
<td>E</td>
</tr>
</tbody>
</table>
Forwarding tables and routing algorithms

- How did the forwarding table get there?
- Typically a routing algorithm is run among the routers, and this algorithm establishes the contents of the forwarding table.
- In this class, we’ll look in detail at address structures and routing algorithms.
Two kinds of links

- Routers and hosts in the Internet are typically connected by two types of links.
- We’ve been looking at pictures of point-to-point links.
- The other common kind is the broadcast link.
  - Usually Ethernet.
Point-to-point and broadcast links

---

**Point-to-point**

**Broadcast**
Point-to-point and broadcast links

One packet sent

One packet sent
Point-to-point and broadcast links

One packet received

\[ \circlearrowleft \rightarrow \circlearrowright \]

\[ \text{One packet received} \]

N packets received

\[ \uparrow \downarrow \downarrow \downarrow \]

\[ \text{N packets received} \]
Broadcast link (Ethernet)

- Well, N packets are “seen”, not really received
- The Ethernet hardware filters out packets that are not for “self”
  - By examining the Ethernet address
- The operating system (OS) never sees the packet (no packet interrupt)
- Though Ethernet does have multicast and broadcast address
Ethernet addresses and IP addresses???

- As you all know, the Internet is a network of networks
  - That’s why it’s called the *Internet*
- This introduces the concepts of:
  - Interface
  - Encapsulation
Next hop and interface (and logical interface!)

- Next hop is the next router on the path to the destination host
  - Or may be the destination host itself
- Interface is the input/output port over which the next hop can be reached
  - May be physical (an actual wire)
  - Or logical (multiple interfaces on an actual wire)
Next hop and interface (and logical interface!)

R has more next hops than interfaces
So the router has another table (neighbor table)
Router packet forwarding procedure:

- Look up dest IP address in received packet
  - Obtain Next Hop router (its IP address)
- Look up Next Hop router in the Neighbor Table
  - (with a pointer from the forwarding table entry)
  - Obtain iface (interface) and “link” address of Next Hop router
- *Encapsulate* IP packet in link packet and send over iface
Router packet forwarding procedure:

1. The IP packet is received at the router.
2. The router consults its routing table to determine the next hop (NH).
3. The routing table entry with the matching address is used to forward the packet.
4. The packet is sent out the appropriate interface (I).

Routing Table:

<table>
<thead>
<tr>
<th>addr</th>
<th>NH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>100</td>
<td>B</td>
</tr>
</tbody>
</table>

Routing Table:

<table>
<thead>
<tr>
<th>NH</th>
<th>If</th>
<th>addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>I1</td>
<td>42</td>
</tr>
<tr>
<td>B</td>
<td>I1</td>
<td>96</td>
</tr>
<tr>
<td>C</td>
<td>I2</td>
<td>103</td>
</tr>
<tr>
<td>F</td>
<td>I3</td>
<td>77</td>
</tr>
</tbody>
</table>

Example:

- IP = SA=5 DA=1
  - Forwarded on link
- IP = SA=5 DA=1
  - Forwarded on link
- IP = SA=52 DA=103
  - Forwarded on link
Protocol Layers

End Host

App

TCP

IP

Link

Router

App

TCP

IP

Link

Physical

End Host

Physical
Packets (revisited)

- Body
- Header
- Body
- Header

- Tells where the next packet is
- Length
- Address

- Tells where to route the packet
But, what is “where”?

- To an Ethernet, “where” is an Ethernet port
  - Ethernet address
- To the Internet, “where” is a host computer on a network
  - IP address
- To a host computer, “where” is a process
  - TCP or UDP port
- To a process, “where” may be a file
  - HTTP URL
A stack of headers

- To deal with all these “wheres”, a packet in fact contains a stack of headers:
A stacked header requires one more field: "next header"

Sometimes combined
Header stack as protocol services

- Except for the physical layer protocol, protocol peers communicate with each other by talking to a lower layer.
  - HTTP peers use TCP, TCP peers use IP, etc.
- We say that each protocol provides a service to the layer above it.
  - Often there is a service interface that defines the service.
Protocol services

- **App**: Provides reliability and ordered byte stream
- **TCP**: Provides unreliable packet delivery
  - across the Internet
  - across a link
- **IP**: Service Boundaries
- **Ethernet**:
Services as a protocol graph

HTTP  FTP  RADIUS  RTP

TCP  UDP

IP

Link1  Link2  ...  LinkN

The famous IP hourglass!
Services as a protocol graph

TCP
SCTP
UDP
IPv4
IPv6
Link1
Link2
…”

This is closer to reality!!!
Example Microsoft VPN stack

- Application
  - TCP
  - IP
  - PPP
  - L2TP
  - UDP
  - IPsec
  - IP
  - PPP
  - PPPoE
  - Ethernet

- The actual end-to-end network and transport layers
- A network abstraction that Microsoft finds convenient
- A security layer
- A tunnel
- A logical link layer
- The link layer
Example Microsoft VPN stack

TCP: Transport Control Protocol
IP: Internet Protocol
PPP: Point-to-Point Protocol
L2TP: Layer 2 Tunneling Protocol
UDP: User Datagram Protocol
IPsec: Secure IP
PPPoE: PPP over Ethernet

TCP: Transport Control Protocol
IP: Internet Protocol
PPP: Point-to-Point Protocol
L2TP: Layer 2 Tunneling Protocol
UDP: User Datagram Protocol
IPsec: Secure IP
PPPoE: PPP over Ethernet
Protocol layers revisited

TCP addressing gets packets across this boundary

OS boundary

Network Interface hardware boundary
Summary of lecture 1

- *Packet networks* are more flexible than *circuit networks*
  - But have “QoS” issues of *delay (latency)*, *dropping*, and *jitter* (versus *blocking* for circuit networks)
- Fancy *queuing* can help, but ultimately traffic sources have to slow down to avoid *congestion*
Summary of lecture 1

- Delay has three components, queuing, propagation, and transmit.
- Large $\text{Delay} \times \text{Bandwidth Product}$ pipes are becoming more common.
- Packets have headers that tell where the packet is going, and how long it is (among other things).
Summary of lecture 1

- **Routers** have *forwarding tables* that select the *next hop* in a *path* to an *address*
  - And *neighbor tables* that tell which *interface* and link address to use to get to the next hop
- *Encapsulation* is used to get the IP packet from one router to another over a link
Protocols are *layered*, with each layer providing a communications *service* to the layer above.

The layering is complex, with *tunnels* that allow protocols to be layered over themselves.

IP is a special layer at the waist of the Internet hourglass.
Next Lecture: IP

- Because of IP’s special position in the Internet, it seems reasonable to start with IP, then work down and up...