Lecture 8 Overview

- Bridging & switching
  - Spanning Tree

- Internetworking
  - Routering
  - Internet Protocol
Spanning Tree Example

- Sample messages to and from B3:
  1. B3 sends (B3, 0, B3) to B2 and B5
  2. B3 receives (B2, 0, B2) and (B5, 0, B5) and accepts B2 as root
  3. B3 sends (B2, 1, B3) to B5
  4. B3 receives (B1, 1, B2) and (B1, 1, B5) and accepts B1 as root
  5. B3 wants to send (B1, 2, B3) but doesn’t as its nowhere “best”
  6. B3 receives (B1, 1, B2) and (B1, 1, B5) again and again...

Data forwarding is turned off for LAN A

What messages is B5 receiving?

A. (B1, 2, B3)
B. (B1, 0, B1)
C. (B1, 1, B7)
D. All of the above
Important Details

- What if root bridge fails?
  - Age configuration info
    » If not refreshed for MaxAge seconds then delete root and recalculate spanning tree
    » If config message is received with a more recent age, then recalculate spanning tree
  - Applies to all bridges (not just root)

- Temporary loops
  - When topology changes, takes a bit for new configuration messages to spread through the system
  - Don’t start forwarding packets immediately -> wait some time for convergence
Switched Ethernet

- Hosts directly connected to a bridge
  - learning + spanning tree protocol

- Switch supports parallel forwarding
  - A-to-B and A'-to-B’ simultaneously
  - Generally full duplex as well

- **Switch** backplane capacity varies
  - Ideally, non-blocking
  - I.e., can run at full line rate on all ports

- No longer any shared bus
  - Each link is its own collision domain
Layer-2 Forwarding

- Create spanning tree across LANs
  - Learn which ports to use to reach which addresses

Benefits
- Higher link bandwidth (point-to-point links)
- Higher aggregate throughput (parallel communication)
- Improved fault tolerance (redundant paths)

Limitations
- Requires homogeneous link layer (e.g. all Ethernet)
- Harder to control forwarding topology

What if we want to connect different link layers?
Combing Networks

- Main challenge is heterogeneity of link layers:
  - Addressing
    » Each network media has a different addressing scheme
  - Bandwidth
    » Modems to terabits
  - Latency
    » Seconds to nanoseconds
  - Frame size
    » Dozens to thousands of bytes
  - Loss rates
    » Differ by many orders of magnitude
  - Service guarantees
    » “Send and pray” vs reserved bandwidth
internetworking

- Cerf & Kahn 74, “A Protocol for Packet Network Intercommunication”
  - Foundation for the modern Internet

- Routers forward packets from source to destination
  - May cross many separate networks along the way

- All packets use a common Internet Protocol
  - *Any* underlying data link protocol
  - *Any* higher layer transport protocol
Defining the Internet

CSE 123 – Lecture 8: Networks to Internetworks
IP Networking

CSE 123 – Lecture 8: Networks to Internetworks
Routers

- A router is a store-and-forward device
  - Routers are connected to multiple networks
  - On each network, looks just like another host
  - A lot like a switch, but supports multiple datalink layers and makes decisions at the network layer

- Must be explicitly addressed by incoming frames (L2)
  - Not at all like a bridge or switch, which are transparent
  - Removes link-layer header, parses IP header (L3)

- Looks up next hop, forwards on appropriate network
  - Each router need only get one step closer to destination
IP Philosophy

- Impose few demands on network
  - Make few assumptions about what network can do
  - No QoS, no reliability, no ordering, no large packets
  - No persistent state about communications; no connections

- Manage heterogeneity at hosts (not in network)
  - Adapt to underlying network heterogeneity
  - Re-order packets, detect errors, retransmit lost messages…
  - Persistent network state only kept in hosts (fate-sharing)

- Service model: best effort, a.k.a. send and pray
## IP Packet Header

The IP (Internet Protocol) packet header consists of several fields that are used to control the transmission and routing of data packets. The header is structured as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version (ver)</td>
<td>4</td>
</tr>
<tr>
<td>Header Length (HL)</td>
<td>4</td>
</tr>
<tr>
<td>Type of Service (TOS)</td>
<td>8</td>
</tr>
<tr>
<td>Length (length)</td>
<td>16</td>
</tr>
<tr>
<td>Identification</td>
<td>16</td>
</tr>
<tr>
<td>Flags (F)</td>
<td>3</td>
</tr>
<tr>
<td>Offset (offset)</td>
<td>12</td>
</tr>
<tr>
<td>Time to Live (TTL)</td>
<td>8</td>
</tr>
<tr>
<td>Protocol (protocol)</td>
<td>8</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16</td>
</tr>
<tr>
<td>Source Address</td>
<td>32</td>
</tr>
<tr>
<td>Destination Address</td>
<td>32</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>variable</td>
</tr>
<tr>
<td>Data (if any)</td>
<td>variable</td>
</tr>
</tbody>
</table>

The diagram above illustrates the layout of the IP packet header, showing the placement and size of each field. The header occupies 20 bytes, and fields such as identification, flags, and offset are used to manage packet processing and error handling. The source and destination addresses are used to identify the origin and destination of the packet, respectively.
Version field

- Which version of IP is this?
  - Plan for change
  - Very important!

What versions of IP are in use today?
A. IPv4
B. IPv6
C. Both
D. More than two
Header length

- How big is IP header?
  - Counted in 32-bit words
  - Variable length
    » Options
  - Engineering consequences of variable length…

- Most IP packet headers are 20 bytes long

What is the content of the HL field for most IP packets?

A. 20
B. 5
C. Something else
Type-of-Service

- How should this packet be treated?
  - Care/don’t care for delay, throughput, reliability, cost
  - How to interpret, how to apply on underlying net?
  - Largely unused until 2000 (hijacked for new purposes, ECN & Diffserv)
TTL (Time-to-Live)

- How many more routers can this packet pass through?
  - Designed to limit packet from looping forever
- Each router decrements TTL field
- If TTL is 0 then router discards packet
Protocol

- Which transport protocol is the data using?
  - i.e. how should a host interpret the data

- TCP = 6
- UDP = 17
**IP Checksum**

- Header contains simple checksum
  - Validates content of header *only*

- Recalculated at each hop
  - Routers need to update TTL
  - Hence straightforward to modify

- Ensures *correct* destination receives packet
For Next Time

- Read 3.3
- Project 1a DUE MONDAY night
- Project 1b assigned Monday; introduced in discussion