CMSC 332
Computer Networks
Network Layer

Professor Szajda
Where in the Stack...

- Application
  - The Web, DNS, Bittorrent, etc
- Transport
- Everything Else!
  - Process to Process - Guarantees?
  - Everything Else? What’s that?
Chapter goals:

- understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - routing (path selection)
  - dealing with scale
  - advanced topics: IPv6, mobility
    - expect to hear more on mobility later!

- instantiation, implementation in the Internet
Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What’s inside a router
- 4.4 IP: Internet Protocol
  ‣ Datagram format
  ‣ IPv4 addressing
  ‣ ICMP
  ‣ IPv6
- 4.5 Routing algorithms
  ‣ Link state
  ‣ Distance Vector
  ‣ Hierarchical routing
- 4.6 Routing in the Internet
  ‣ RIP
  ‣ OSPF
  ‣ BGP
- 4.7 Broadcast/Multicast
Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in every host, router
- router examines header fields in all IP datagrams passing through it
- this is the first layer that everything between you and the receiver looks at!
Two Key Network-Layer Functions

- **forwarding**: move packets from router’s input to appropriate router output
- **routing**: determine route taken by packets from source to dest.
  - **routing algorithms**
  - analogy:

  - **routing**: process of planning trip from source to dest
  - **forwarding**: process of getting through single interchange

*Think of it this way* - this is the difference between figuring out the way to your destination using Google Maps and completing the next step in those directions!
Interplay between routing and forwarding

Routing algorithm

<table>
<thead>
<tr>
<th>header</th>
<th>output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>3</td>
</tr>
<tr>
<td>0101</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
</tbody>
</table>

Value in arriving packet's header

0111
Connection setup

- **3rd** important function in *some* network architectures:
  - ATM, frame relay, X.25

- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - routers get involved

- network vs transport layer connection service:
  - **network**: between two hosts (may also involve intervening routers in case of VCs)
  - **transport**: between two processes
Q: What service model for “channel” transporting datagrams from sender to receiver?

Example services for individual datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

Example services for a flow of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing
- security
## Network layer service models:

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Bandwidth</th>
<th>Loss</th>
<th>Order</th>
<th>Timing</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no (inferred via loss)</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>constant rate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no congestion</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR</td>
<td>guaranteed minimum</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Chapter 4: Network Layer

- **4.1 Introduction**
- **4.2 Virtual circuit and datagram networks**
- **4.3 What’s inside a router**
- **4.4 IP: Internet Protocol**
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- **4.5 Routing algorithms**
  - Link state
  - Distance Vector
  - Hierarchical routing
- **4.6 Routing in the Internet**
  - RIP
  - OSPF
  - BGP
- **4.7 Broadcast and multicast routing**
Network layer connection and connection-less service

• datagram network provides network-layer connectionless service

• VC network provides network-layer connection service

• analogous to the transport-layer services, but:
  ‣ service: host-to-host
  ‣ no choice: network provides one or the other
  ‣ implementation: in network core
Virtual circuits

“source-to-dest path behaves much like telephone circuit”

- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains “state” for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)
VC implementation

• A VC consists of:
  1. path from source to destination
  2. VC numbers, one number for each link along path
  3. entries in forwarding tables in routers along path

• packet belonging to VC carries VC number (rather than dest address)

• VC number can be changed on each link (why?)
  ‣ New VC number comes from forwarding table
Forwarding table

Forwarding table in northwest router:

<table>
<thead>
<tr>
<th>Incoming interface</th>
<th>Incoming VC #</th>
<th>Outgoing interface</th>
<th>Outgoing VC #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>97</td>
<td>3</td>
<td>87</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Routers maintain connection state information!
Virtual circuits: signaling protocols

- Used to setup, maintain, and teardown VC
- Used in ATM, frame-relay, X.25
- Not used in today’s Internet
  - Remember this for the next layer though!
Datagram Networks

- No individual call setup at network layer
- Routers: no state about end-to-end connections
  - no network-level concept of “connection”
- Packets forwarded using destination host address
  - packets between same source-dest pair may take different paths

1. Send data
2. Receive data
## Forwarding table

**Destination Address Range**  |  **Link Interface**
--- | ---
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111 | 0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011001 00000000 | 1
11001000 00010111 00011111 11111111 through 11001000 00010111 00011111 00000000 | 2
otherwise | 3

4 billion possible entries
Longest Prefix Matching

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

Examples

DA: 11001000 00010111 00010110 10100001  Which interface?

DA: 11001000 00010111 00011000 10101010  Which interface?
Datagram or VC network: why?

Internet (datagram)
- data exchange among computers
  - “elastic” service, no strict timing req.
- “smart” end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at “edge”
- many link types
  - different characteristics
  - uniform service difficult

ATM (VC)
- evolved from telephony (human conversation):
  - strict timing and reliability
  - need for guaranteed service
- “dumb” end systems
  - telephones
- complexity inside network
Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What’s inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing
Router Architecture Overview

Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link
Input Port Functions

Decentralized switching:
- given datagram dest., lookup output port using forwarding table in input port memory
- goal: complete input port processing at ‘line speed’
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Physical layer: bit-level reception
Data link layer: e.g., Ethernet
Three types of switching fabrics

- Memory
- Bus
- Crossbar
Switching Via Memory

First generation routers:

• traditional computers with switching under direct control of CPU
• packet copied to system’s memory
• speed limited by memory bandwidth (2 bus crossings per datagram)
Switching Via a Bus

- datagram from input port memory to output port memory via a shared bus

- **bus contention**: switching speed limited by bus bandwidth

- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers
Switching Via An Interconnection Network

- overcome bus bandwidth limitations
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network
Output Ports

- **Buffering** required when datagrams arrive from fabric faster than the transmission rate

- **Scheduling discipline** chooses among queued datagrams for transmission
Output port queueing

- buffering when arrival rate via switch exceeds output line speed
- queueing (delay) and loss due to output port buffer overflow!
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity $C$
  - e.g., $C = 10$ Gps link: 2.5 Gbit buffer

- Recent recommendation: with $N$ flows, buffering equal to

  $\frac{\text{RTT} \cdot C}{\sqrt{N}}$

- ISPs tend to far OVER provision buffer sizes.
  - Good or bad?
Input Port Queuing

- Fabric slower than input ports combined -> queueing may occur at input queues
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward
- queueing delay and loss due to input buffer overfill!
Until Next Time

• Project 2 - get started!
  ‣ Everyone should be paired up at this point.

• Next Class
  ‣ Read Section 4.4: IP