Chapter 4: Network Layer

Chapter goals:

- understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - routing (path selection)
  - broadcast, multicast
- instantiation, implementation in the Internet

Slide from Kurose & Ross, 6th Ed
Chapter 4: Outline

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

- 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
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

Interplay between routing and forwarding

- Routing algorithm determines end-end-path through network
- Forwarding table determines local forwarding at this router

Value in arriving packet’s header:

- **0111**

- **0100**
- **0101**
- **0111**
- **1001**

Header value | Output link
--- | ---
0100 | 3
0101 | 2
0111 | 2
1001 | 1

Slide from Kurose & Ross, 6th Ed
**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

---

**Network service model**

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
## Network layer service models:

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Guarantees?</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bandwidth</td>
<td>Loss</td>
</tr>
<tr>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>constant</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>VBR</td>
<td>guaranteed</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR</td>
<td>guaranteed</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>UBR</td>
<td>minimum</td>
<td>no</td>
</tr>
</tbody>
</table>

CBR = Constant Bit Rate, VBR = Variable Bit Rate, ABR = Available Bit Rate, UBR = Unspecified Bit Rate

---

## Chapter 4: Outline

1. **introduction**
2. **virtual circuit and datagram networks**
3. what’ s inside a router
4. **IP: Internet Protocol**
   - datagram format
   - IPv4 addressing
   - ICMP
   - IPv6
5. **routing algorithms**
   - link state
   - distance vector
   - hierarchical routing
6. **routing in the Internet**
   - RIP
   - OSPF
   - BGP
7. **broadcast and multicast routing**
Connection, connection-less service

- **datagram** network provides network-layer connectionless service
- **virtual-circuit** network provides network-layer connection service
- analogous to TCP/UDP connection-oriented / connectionless 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. A 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.
     - new VC number comes from forwarding table

---

VC 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>

VC routers maintain connection state information!
Virtual circuits: signaling protocols

- used to setup, maintain, teardown VC
- used in ATM, frame-relay, X.25
- not used in today’s Internet

Virtual Circuit Switching

- Explicit connection setup (and tear-down) phase
- Subsequence packets follow same circuit
- Sometimes called *connection-oriented* model

- Analogy: phone call
- Each switch maintains a VC table

Step 1  Step 2

http://www.cs.princeton.edu/courses/archive/spr03/cs461/
Virtual Circuit Switching

- Explicit connection setup (and tear-down) phase
- Subsequence packets follow same circuit
- Sometimes called connection-oriented model
- Analogy: phone call
- Each switch maintains a VC table

Switch 1

<table>
<thead>
<tr>
<th>Incoming Interface</th>
<th>Incoming VCI</th>
<th>Outgoing Interface</th>
<th>Outgoing VCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Switch 2

<table>
<thead>
<tr>
<th>Incoming Interface</th>
<th>Incoming VCI</th>
<th>Outgoing Interface</th>
<th>Outgoing VCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>11</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Switch 3

<table>
<thead>
<tr>
<th>Incoming Interface</th>
<th>Incoming VCI</th>
<th>Outgoing Interface</th>
<th>Outgoing VCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Step 3

Step 4
Virtual Circuit Switching

- Explicit connection setup (and tear-down) phase
- Subsequence packets follow same circuit
- Sometimes called *connection-oriented* model

- Analogy: phone call
- Each switch maintains a VC table

**Step 5**

**Step 6**
Virtual Circuit Switching

• Explicit connection setup (and tear-down) phase
• Subsequence packets follow same circuit
• Sometimes called connection-oriented model

• Analogy: phone call

• Each switch maintains a VC table

**Step 7**

<table>
<thead>
<tr>
<th>Switch 2</th>
<th>(\text{Incomming Interface} )</th>
<th>(\text{Incomming VCI} )</th>
<th>(\text{Outgoing Interface} )</th>
<th>(\text{Outgoing VCI} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch 2</td>
<td>3</td>
<td>11</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch 1</th>
<th>(\text{Incomming Interface} )</th>
<th>(\text{Incomming VCI} )</th>
<th>(\text{Outgoing Interface} )</th>
<th>(\text{Outgoing VCI} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch 1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch 3</th>
<th>(\text{Incomming Interface} )</th>
<th>(\text{Incomming VCI} )</th>
<th>(\text{Outgoing Interface} )</th>
<th>(\text{Outgoing VCI} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch 3</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**Step 8**
Virtual Circuit Switching

- Explicit connection setup (and tear-down) phase
- Subsequence packets follow same circuit
- Sometimes called connection-oriented model
- Analogy: phone call
- Each switch maintains a VC table

<table>
<thead>
<tr>
<th>Switch 2</th>
<th>Incoming Interface</th>
<th>Incoming VCI</th>
<th>Outgoing Interface</th>
<th>Outgoing VCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>11</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch 1</th>
<th>Incoming Interface</th>
<th>Incoming VCI</th>
<th>Outgoing Interface</th>
<th>Outgoing VCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch 3</th>
<th>Incoming Interface</th>
<th>Incoming VCI</th>
<th>Outgoing Interface</th>
<th>Outgoing VCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Virtual Circuit Model

- Typically wait full RTT for connection setup before sending first data packet.
- While the connection request contains the full address for destination, each data packet contains only a small identifier, making the per-packet header overhead small.
- If a switch or a link in a connection fails, the connection is broken and a new one needs to be established.
- Connection setup provides an opportunity to reserve resources.

http://www.cs.princeton.edu/courses/archive/spr03/cs461/
Datagram networks

- no 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

Datagram forwarding table

- 4 billion IP addresses, so rather than list individual destination address list range of addresses (aggregate table entries)
Datagram forwarding table

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

Q: but what happens if ranges don’t divide up so nicely?

Longest prefix matching

longest prefix matching

destination address, use longest address prefix that matches destination address.

<table>
<thead>
<tr>
<th>Destination Address Range</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>

eamples:

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.
- many link types
  - different characteristics
  - uniform service difficult
- “smart” end systems (computers)
  - can adapt, perform control, error recovery
  - *simple inside network, complexity at “edge”*

**ATM (VC)**
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- “dumb” end systems
  - telephones
  - *complexity inside network*

---

Chapter 4: outline

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
Switches

Juniper T series
- T4000 (2010)
  - 4 Tbps in ½ 19" rack
  - 208 ports @ 10 Gbps
  - 4 Gpps
  - 11.5 kW

Cisco CRS-3 (2010)
- 1.28 Tbps in ½ 19" rack
- 16 ports @ 40 Gbps
- 13.5 kW

Router Physical Layout

Modified from http://www.cs.princeton.edu/courses/archive/spring10/cos461/docs/lec10-linkstate.ppt
Performance Challenge ...

Example
64Byte packets @10Gbps = 51.2 ns/packet
512 packets/s @10Gbps = 0.1 ns/packet!!!
Cycle time for a 3 GHz CPU = 0.33ns!!
High-Performance Switches are High-Performance Parallel Computer Systems
with low latency high-bandwidth internal networks

Router architecture overview

two key router functions:
- run routing algorithms/protocol (RIP, OSPF, BGP)
- forwarding datagrams from incoming to outgoing link

Slide from Kurose & Ross, 6th Ed
Routing vs. Forwarding

• Routing: control plane
  – Computing paths the packets will follow
  – Routers talking amongst themselves
  – Individual router creating a forwarding table

• Forwarding: data plane
  – Directing a data packet to an outgoing link
  – Individual router using a forwarding table

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable

- three types of switching fabrics
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)

Slide from Kurose & Ross, 6th Ed

http://www.stanford.edu/class/ee384x/EE384X//handouts/handout2.ppt
Switching via Memory

- DDR3 1333 MHz, 64-bit wide, 10.66 GB/s
- 64-byte packets, write and read = 2 accesses, max rate $10.66 \times 10^9/(2 \times 64) = 83.33 \times 10^6$ pps
- SRAM comparable bandwidth, but ~5 nsec latency (e.g. Cypress QDR-II+ Xtreme SRAM) compared to DDR3’s latency of ~10 – 30 nsec

Compare e.g.
- Juniper T4000 router: 4 Gpps in single chassis (½ rack) configuration

Early days: Modified Computer

Must run at rate $N \times R$

Bottlenecks

http://www.stanford.edu/class/e384x/EE384X/handouts/handout2.ppt
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

---

2nd Generation Router

http://www.stanford.edu/class/ee384x/EE384X/handouts/handout2.ppt
Switching via bus

• I/O bus PCI Express Gen 2.0 x16 peak 8 GB/sec
• Two trips across the I/O bus for each packet
• Max packet rate for 64-byte packets:
  \[8 \times 10^9 / (2 \times 64) = 64 \times 10^6 \text{ pps}\]
• PCI Express Gen 3.0, Nov 2010, 16 GB/s ....

Switching via interconnection network

- overcome bus bandwidth limitations
- banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network
Multistage Networks

- Centralized Switched (Indirect) Networks
  
  - Multistage interconnection networks (MINs)
    
    - Crossbar split into several stages consisting of smaller crossbars
    - Complexity grows as $O(N \times \log N)$, where $N$ is # of end nodes
    - Inter-stage connections represented by a set of permutation functions

Omega topology, perfect-shuffle exchange

3rd Generation Router: Switch

http://www.stanford.edu/class/ee384x/EE384X//handouts/handout2.ppt
4th Generation Router

Multirack; optics inside

Optical links

100s of metres

Linecards

Switch
Input port functions

- **Physical layer**: bit-level reception
- **Data link layer**: e.g., Ethernet
  - see chapter 5

**Decentralized switching:**
- given datagram dest., lookup output port using forwarding table in input port memory ("match plus action")
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

---

Line Cards (Interface Cards, Adaptors)

- **Interfacing**
  - Physical link
  - Switching fabric
- **Packet handling**
  - Packet forwarding
  - Decrement time-to-live
  - Buffer management
  - Link scheduling
  - Packet filtering
  - Rate limiting
  - Packet marking
  - Measurement

---

http://www.cs.princeton.edu/courses/archive/spring10/cos461/docs/lec10-linkstate.ppt
**Output port queuing**

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

**Input port queuing**

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

---

*Slide from Kurose & Ross, 6th Ed*
Queue Management - Scheduling

- First-in-First-Out (FIFO)
- Priority Scheduling/Queueing (PQ)
- Fair Scheduling/Queueing (FQ)
- Weighted Fair Scheduling/Queueing (WFQ)
- Class Based Queueing (CBQ)
- Random

Queue Discipline | Advantages | Disadvantages
--- | --- | ---
FIFO | Extremely simple | No service class supported
No packet reordering | Queueing delay increases as congestion increases
Maximum delay easy to predict | PQ | Supports different service classes | Bandwidth starvation problem
Relatively simple | No fairness | FQ | Supports different service classes | Packets must have the same size for efficient scheduling
Bursty flows do not affect other flows | No way to provide real-time services
Allows equal sharing of bandwidth among many flows | No way to serve flows with different bandwidth requirements | WFQ | Gives a minimum amount of resources to each service class | High complexity
Bursty flows do not affect other flows | CBQ | Gives a minimum amount of resources to each service class | Packets must have same size for fair scheduling
Allows equal sharing of bandwidth among many flows | Limited complexity

Queue Management – Congestion Control

• Drop Tail
• Active Queue Management (AQM)
  – Random Early Detection (RED)
    • Based on queue length and rate mismatch between demand and capacity
    • Many variants: Adaptive RED (ARED), Flow RED (FRED), Robust RED (RRED), Stabilized RED (SRED), RED Preferential Dropping (RED-PD), ...
  – Random Exponential Marking (REM)
    • Based on “price” that is based on actual and target queue lengths and rate mismatch between demand and capacity
  – BLUE, Stochastic Fair BLUE (SFB), Resilient SFB (RSFB), ...
    (adaptive drop/mark probability)
    • Explicit Congestion Notification (ECN)

Drop-Tail Queueing

• Drop-tail queuing leads to bursty loss
  – When a link becomes congested...
  – ... many arriving packets encounter a full queue
  – And, as a result, many flows divide sending rate in half
  – ... and, many individual flows lose multiple packets

Source: http://www.cs.princeton.edu/courses/archive/spr10/cos461/docs/lec06-congestion.ppt
Drop Tail Queueing

- Slow Feedback
  - comes when buffer is completely full
  - even though the buffer has been filling for a while
- The filling buffer is increasing RTT
  - ... and the variance in the RTT
- Might be better to give early feedback
  - Get 1-2 connections to slow down, not all of them
  - Get these connections to slow down before it is too late

Random Early Detection (RED)

- Basic idea of RED
  - Router notices that the queue is getting backlogged
  - ... and randomly drops packets to signal congestion
- Packet drop probability
  - Drop probability increases as queue length increases
  - If buffer is below some level, don’t drop anything
  - ... otherwise, set drop probability as function of queue
Properties of RED

- Drops packets before queue is full
  - In the hope of reducing the rates of some flows
- Drops packet in proportion to each flow’s rate
  - High-rate flows have more packets
  - ... and, hence, a higher chance of being selected
- Drops are spaced out in time
  - Which should help desynchronize the TCP senders
- Tolerant of burstiness in the traffic
  - By basing the decisions on average queue length

Problems With RED

- Hard to get the tunable parameters just right
  - How early to start dropping packets?
  - What slope for the increase in drop probability?
  - What time scale for averaging the queue length?
- Sometimes RED helps but sometimes not
  - If the parameters aren’t set right, RED doesn’t help
  - And it is hard to know how to set the parameters
- RED is implemented in practice
  - But, often not used due to the challenges of tuning right
- Many variations in the research community
  - With cute names like “Blue” and “FRED”... 😊
Explicit Congestion Notification

- Early dropping of packets
  - Good: gives early feedback
  - Bad: has to drop the packet to give the feedback

- Explicit Congestion Notification
  - Router marks the packet with an ECN bit
  - ... and sending host interprets as a sign of congestion

- Surmounting the challenges
  - Must be supported by the end hosts and the routers
  - Requires 2 bits in the IP header for detection (forward dir)
    - One for ECN mark; one to indicate ECN capability
    - Solution: borrow 2 of Type-Of-Service bits in IPv4 header
  - Also 2 bits in TCP header for signaling sender (reverse dir)

Source: http://www.cs.princeton.edu/courses/archive/spr10/cos461/docs/lec06-congestion.ppt
How much buffering?

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

• recent recommendation: with $N$ flows, buffering equal to $\frac{RTT \cdot C}{\sqrt{N}}$