Session 7.
Packet Switching Networks

Dongsoo S. Kim (dskim@iupui.edu)
Electrical and Computer Engineering
Indiana U. Purdue U. Indianapolis

Overview
- General Layer-3 (Network) Protocols
- Packet network topology
- IP network
- ATM network
- Datagram and virtual circuits
- Routing
- Shortest path algorithms
- Optimal path algorithms
- Traffic control
- Congestion control
Network Layer Functions

- Transport packet from sending to receiving hosts
- Network layer protocols in every host, router
- Three important functions:
  - Path determination: route taken by packets from source to destination
    - Routing algorithms
  - Switching: move packets from router’s input to appropriate router output
    - Switch architecture
      - Shared memory architecture/shared bus architecture
      - Multistage switch architectures
      - Input buffer vs. output buffer
  - Call setup: some network architectures require router call setup along path before data flows
    - Signaling
    - Admission control
Network service model

Question: What service model of the channel for transporting packets from sender to receiver?

- Service model
  - Guaranteed bandwidth?
  - Preservation of inter-packet timing (no jitter)?
  - Loss-free delivery?
  - In-order delivery?
  - Congestion feedback to sender?
  - Cost?

Virtual Circuits

- Circuit
  A source-to-destination path behaves much like telephone circuit
  - Performance-wise
  - Network actions along source-to-destination path
  - call setup, teardown for each call before and after data flows

- Virtual
  - No link is dedicated to the connection not as pure circuit switching
  - A channel may not dedicate to the connection not as TDM

- Not a Datagram
  - Each packet carries VC identifier (not destination host OD)
  - Every router on source-destination paths maintain state for each passing connection
  - Link, router resources (bandwidth, buffers) may be allocated to VC to get circuit-like performance
Virtual Circuits: Signaling Protocols

- Used to setup, maintain, and teardown VC
- Used in ATM, frame-relay, X.25

Datagram Networks: the Internet model

- No call setup at network layer
- Routers: no state about end-to-end connections
  - No network-level concept of connection
- Packets typically routed using destination host ID
  - Packets between same source-destination pair may take different paths
Network Layer Service Models

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Guarantees?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bandwidth</td>
</tr>
<tr>
<td>Internet</td>
<td>Best Effort</td>
<td>None</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>Constant Rate</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>Guaranteed Rate</td>
</tr>
<tr>
<td></td>
<td>ABR</td>
<td>Guaranteed minimum</td>
</tr>
<tr>
<td></td>
<td>UBR</td>
<td>None</td>
</tr>
</tbody>
</table>

Datagram or VC network: why?

**Internet**
- Data exchange among computers
  - elastic service
  - no strict timing requirement
- Intelligent end-systems (computers)
  - Can adapt, perform control, error recovery
  - Simple inside network, complexity at edge
- Many link types
  - Different characteristics
  - Uniform service difficult

**ATM**
- Designed for integration (B-ISDN)
  - Telephony
    - strict timing, reliability requirements
    - need for guaranteed service
  - Stream Service
    - Delay is OK
    - Variance of delay is critical
  - Data communication
    - Bursty
    - Need bandwidth as much as possible, but not always
- Complication of network device
What Are They?

Application or below
Transport
Network
Data Link
Physical

IP wares
AAL
ATM
Ether
DSL

Network Topology

Internet
Gateway or Border Router
High speed backbone
Autonomous System
router
hub
Department server
Distributive servers
switches
server
**IP Router**

- General computer system with two or more IP interfaces
- Each IP packet contains the destination address
- The IP router decides the next router with the destination address
  - No previous knowledge on the destination address
  - No touch on the IP packet (with a small exception?)
- Routing table
  - List of destination address and port number
  - Table lookup for each packet, and forward to the port number

---

**Virtual Circuit Switching**

- Establish a connection (fixed path) before communication
  - Set parameters in the switches along the path
    - Bandwidth
    - Loss rate
    - Delay and delay variants
- A physical link can be shared by multiple connections
  - Admission control - whether allow the establishment or not based on the available resources in the network
- All packets for a connection follow the same path
- Each packet contains a VCI but not a full destination address
  - The overhead caused by the full source and destination address
  - Table lookup with the VCI to find the next-hop port and VCI
  - The VCI in the packet is translated to the next VCI
Routing - Overview

- Find the best path. But, what is the best?
  - Shortest path - minimum number of hops
  - Minimum end-to-end delay
  - Maximum available bandwidth
- Classification
  - Static or Dynamic (Adaptive)
    - Static routing - pre-assigned path, manually configured routing table
      - Good for small and fixed network topology
      - Disadvantage for network failure
    - Dynamic (adaptive) routing - each router learns the network topology by exchanging information with its neighbors
      - Complication in each router
  - Centralized or Distributed routing
    - Centralized routing - a central computer calculates all possible paths and dictates routers in the network
      - Flood of routing information
    - Distributed routing - each router calculates paths with available information
      - Inconsistency, loop
Principle of Optimality

- Principle
  - If a node b is on the optimal path from node a to node c, then the optimal path from b to c also falls the same path.
- Expansion
  - Sink-tree

Hierarchical Routing

- Size of routing table
  - US telephone - 10 billion entries
  - IP - 32 bit address (4 billion entries)
  - IPv6 - 128 bit address (256x10^36 entries)
- Common prefix
  - Process only the prefix if they are not in my network

317 - 274 - 4493

- Area code
- Central office identifier
- Port # in CO
Distance Vector Routing

- Distributed Bellman-Ford algorithm
- Variations
  - Original ARPANET routing algorithm
  - Internet RIP
  - Novell's IPX
  - AppleTalk and Cisco routers
- Structure of Routing Table

<table>
<thead>
<tr>
<th>To</th>
<th>A, 8</th>
<th>C, 4</th>
<th>E, 3</th>
<th>Route Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>7</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>12</td>
<td>0</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>14</td>
<td>23</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>32</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

7-19

Distance Vector Routing - continued

- HOWTO
  - Metric - Number of hops, Time delay, or something else
  - Each router knows the metrics to the next router
  - Each router sends its table to each neighbor router periodically
  - Each router receives the table of each neighbor and updates its routing table

- Question: What if a link fails?
  - Hey baby, don’t worry.
  - Really?
  - Counting-to-infinity

- Question: Where is the bandwidth consideration?
Link State Routing

- Dijkstra’s algorithm for finding the shortest path
- Enhancing
  - Measure the delay or other cost as bandwidth
  - Propagate a new information quickly
- Example
  - OSPF

Flooding

- Forward an incoming packet to all ports except the one the packet was received from
- Application
  - When no routing table is available
  - For broadcasting
- Problems
  - Swamp the network. Packets are multiplicated exponentially
  - Circulating packets
- Solution
  - Use TTL fields
  - Each router adds an ID to each packet. If the router sees the packet again, then discard it.
  - Each router records an packet ID before flooding. If the router sees the packet again, then discard it.
Hot potato routing

- **Deflecting routing**
- **Howto**
  - In normal situation, a router forwards packets to a default port
  - If the default port is busy or congested, forward any other port
- **Problems**
  - Out-of order delivery
  - No path via the alternative port
- **Advantage**
  - Very simple.
  - No routing table is required
  - No buffer required

Source routing

- **Howto**
  - A source host knows the complete route to the destination
  - Each packet contains the route information in the header
  - Each route strips off the label indicating the node and forwards the packet to the next node
- **Characteristics**
  - In-order guaranteed
  - Overhead of packets
  - Overhead in the source host
Optimal routing

Based on the queuing model

Delay or cost on a link between node \( i \) and \( j \) = \( D_{ij} \)

- Minimize the total cost

Minimize

\[
T = \sum D(C, F) = \sum \alpha \frac{F_y}{C_y - F_y} + \beta F_y
\]

subject to

\( F \) is a multic commodity flow satisfying the requirement, and

\( F \leq C \)

Useful for network planning

ATM Networks

Principles

- Connection-oriented packet switching (virtual circuit switching)
- Integration - voice, data, multimedia
- Statistical multiplexing
- Short fixed-length packets (cells): 53 bytes
  - 5 bytes ATM header and 48 bytes Payload
  - Why 48 bytes?
- Asynchronous – no global clock pulse or no synchronous frame
- Switched Virtual Circuit (SVC)
  - Connection-establishing on demand by ATM signaling
- Permanent Virtual Path, Circuit, or Channel (PVP, PVC)
  - Manually set up the connection for leased lines
- Traffic Descriptor and Quality-of-Service
  - Peak cell rate
  - Long-term average cell rate
  - Maximum length of bursty cells
  - Cell delay and cell loss rate
  - Cell delay jitter
Virtual Path and Virtual Channel

- Hierarchical switching to release the load in switches
  - A physical link contains 4096 (or 256) VPC's (VPI)
  - A VPC contains 65536 VCC's (VCI)
- VP segments

Traffic Management and QOS

- Provide quality-of-service to individual packet thru the series of switches or routers
- What to Provide
  - End-to-end delay
    - Sum of individual delays
  - Jitter experience
    - Variation of packet delay
    - Difference of minimum delay and maximum delay
  - Packet loss rate
    - Caused by faults in equipments, or traffic congestion
  - Bandwidth

The link doesn’t have a control, but the node does.
Queues

- **FIFO Queue**
  - A single common buffer
  - The order of arrival is the order of departure
  - Equal opportunity to every packet - no control of QOS
  - Fairness? - a low-volume traffic vs. a high-volume traffic

- **Priority Queue**
  - A separate buffer to each priority class
  - Transmit packets in a class if all higher queues are empty
  - Fairness in a priority class?

![Diagram of FIFO Queue](image)

- **Fair Queue**
  - To provide equal opportunity to access to transmission bandwidth
  - Each user has its own logical queue
  - Ideally, the network bandwidth is divided equally among the queues
  - Practically, it is impossible to divide the bandwidth equally

![Diagram of Fair Queue](image)
Approximation of Fair Queue

- **Difficulties**
  - Large frame size
  - Variable frame size

- **Approximation**
  - When a packet arrives at a user queue, the ideal completion time is derived and this number is used as a finish tag.
  - After completing a packet transmission, pick the next packet to be transmitted with the smallest finish tag.

Fair Queue, Generalization

The actual duration of a given round is *proportional* to the actual number of buffers ($n_A$).

Let $R(t)$ is the number of rounds at time $t$, then

$$\frac{dR(t)}{dt} = \frac{C}{n_A(t)}$$

where $C$ is the capacity.
Computing Finishing Tag

**Notation:**
- $F(i,k,t)$ = finish time of $k$th packet that arrives at time $t$ to flow $i$.
- $P(i,k,t)$ = size of $k$th packet that arrives at time $t$ to flow $i$.
- $R(t)$ = round number at time $t$

**Fair Queueing:**
- $F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)$

**Weighted Fair Queueing:**
- $F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)/w_i$

---

![Diagram](image)

**Buffer 1 at \( t=0 \):**
- Fluid-flow system: both packets served at rate 1/2

**Buffer 2 at \( t=0 \):**
- Packet from buffer 2 served at rate 1

**Packet from buffer 2 waiting:**
- Finish tag for buffer 2 is 2.

**Packet from buffer 1 served at rate 1:**
- Finish tag for buffer 1 is 1.

---

**Packet-by-packet fair queueing:**
- Buffer 2 served at rate 1
Weighted Fair Queueing

Buffer 1 at $t=0$, weight is 1
Buffer 2 at $t=0$, weight is 3

Packet from buffer 2 served at rate $3/4$
Packet from buffer 1 served at rate 1
Packet from buffer 1 waiting

Fluid-flow system: packet from buffer 1 served at rate $1/4$;
Packet from buffer 1 served at rate 1
Packet-by-packet weighted fair queueing: buffer 2 served first at rate 1; then buffer 1 served at rate 1
Packet from buffer 1 served at rate 1

Congestion Control

Too many packets try to access the same link (buffer)
- The buffer builds up and eventually becomes full
- Packet loss
- Retransmission will make it worse

Strategies
- Open-loop control
  - Prevent congestion from occurring by making sure that the traffic flow generated by the source will not degrade the performance of the network below the specified QoS
  - Admission control
- Closed-loop control
  - React to congestion when it is already happening or is about to happen by regulating the traffic flow
  - Feedback information

7-36
Open-Loop Controls

- Regulate the traffic flow. Once it is accepted, its traffic flow will not overload the network

Strategies

- Admission control
  - Specify QoS parameters (or traffic descriptor) before establishing a connection as peak rate, average rate, maximum burst size, maximum delay, loss probability, and delay variance
  - Connection AC (CAC) systems should know the traffic flow of each source

- Policing
  - Once a connection is accepted, the system must monitors the traffic flow to check whether it violates the initial contract or not
  - Leaky bucket algorithm

- Traffic Shaping
  - Reform a traffic flow to another flow

Leaky Bucket

- Characteristics
  - Irregular incoming packet
  - Constant rate outgoing packet
  - The overflowed packet is discarded

- Capacity of the bucket
  - Depends on the traffic pattern
  - Smoothed traffic needs small bucket
  - Bursty traffic requires a large bucket

- Drain rate
  - Bandwidth allocation to the specific connection
Leaky Bucket Algorithm

Arrival of a packet at time $t_a$

$X' = X - (t_a - LCT)$

$X' < 0$?

Yes

Nonconforming packet

No

$X' > L$?

Yes

$X' = 0$

No

$X = X' + I$

$LCT = t_a$

$X$ = value of the leaky bucket counter

$L$ = nomial inter-arrival time

$L + I$ = size of the bucket

The bucket will drain at a continuous rate of 1 unit per packet time, if it is not empty.

Behavior of Leaky Bucket

Packet arrival

Nonconforming

$L + I$

Bucket content

$X$

Time
**Dual Leaky Bucket**

- Police the sustainable rate as well as the peak rate

Diagram:

- Incoming traffic
- Leaky bucket 1 (PCR and CDVT) → Tagged or dropped
- Untagged traffic
- Leaky bucket 2 (SCR and MBS) → Tagged or dropped
- Untagged traffic

PCR: Peak Cell Rate  
CDVT: Cell Delay Variation Tolerance  
SCR: Sustainable Cell Rate  
MBS: Medium Burst Size

---

**Traffic Shaping**

- For smoothing abrupt traffic flow
- Many possible ways
  - Eg) application periodically generate 10 Kbit data every second

Graphs:

- (a) 10 Kbps
- (b) 50 Kbps
- (c) 100 Kbps
Leaky Bucket Traffic Shaper

- Difference in implementation
  - Leaky bucket policing – counters
  - Leaky bucket traffic shaper - buffer

Token Bucket Traffic Shaper

- Too restricted leaky bucket traffic shaper (constant output rate)
- Flexibility for variable-rate traffic
Closed-Loop Control

- Use of feedback information
- TCP congestion control
  - Implicit feedback – timeout
- ATM congestion control
  - Explicit feedback – OAM message

TCP Congestion Control

- A sender maintains the “congestion window”
- The maximum amount of data that the sender can transmit is the minimum of an advertised window and the congestion window
- Dynamically adjust the congestion windows

![TCP Congestion Control Diagram](image)