Chapter 8: Packet Switching

Raj Jain

Overview

- Circuit, datagram, virtual circuit switching
- Routing algorithms
- ARPAnet routing
**Datagram vs Virtual Circuit**

Connectionless vs connection-oriented

**Circuit vs Datagram vs VC**

<table>
<thead>
<tr>
<th>Circuit Switching</th>
<th>Datagram</th>
<th>Virtual Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dedicated transmission path</td>
<td>No dedicated path</td>
<td>No dedicated path, Shared path</td>
</tr>
<tr>
<td>Continuous transmission of data</td>
<td>Bursty</td>
<td>Bursty</td>
</tr>
<tr>
<td>No buffering required</td>
<td>Buffers required</td>
<td>Buffers required</td>
</tr>
<tr>
<td>Path fixed at connection setup</td>
<td>Different packets may take different paths</td>
<td>Path fixed at connection setup</td>
</tr>
<tr>
<td>Call setup delay</td>
<td>Queueing delays</td>
<td>Call setup + queueing delays</td>
</tr>
<tr>
<td>Overload blocks new calls</td>
<td>Overload increases queueing delays</td>
<td>Overload may block new calls. May increase queueing delays</td>
</tr>
<tr>
<td>Source and destination have the same speed</td>
<td>Source and destination may have different speed</td>
<td>Source and destination may have different speed</td>
</tr>
<tr>
<td>Bandwidth is reserved. Unused</td>
<td>Bandwidth is dynamically shared among users</td>
<td>Bandwidth is reserved as well as dynamically shared</td>
</tr>
<tr>
<td>bandwidth is wasted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No overhead bits after call setup</td>
<td>Overhead bits in each packet</td>
<td>Less overhead bits in each packet</td>
</tr>
<tr>
<td>Switches keep state</td>
<td>Switches do not keep state</td>
<td>Switches keep state</td>
</tr>
<tr>
<td>No or negligible loss</td>
<td>Loss possible</td>
<td>Loss possible</td>
</tr>
</tbody>
</table>

On link failure Connection continues VC broken

Table 8.1
External vs Internal VC

- External: End-to-end (Transport layer)
- Internal: On the path (Network layer)

<table>
<thead>
<tr>
<th>End-to-end (External)</th>
<th>No connection (Datagram)</th>
<th>Connection (VC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Datagram</td>
<td>Virtual Ckt</td>
</tr>
<tr>
<td></td>
<td>UDP/IP</td>
<td>TCP/IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYMNET</td>
</tr>
</tbody>
</table>

Routing

Fig 8.8
Rooting or Routing

- **Rooting** is what fans do at football games, what pics do for truffles under oak trees in the Vaucluse, and what nursery workers intent on propagation do to cuttings from plants.

- **Routing** is how one creates a beveled edge on a table top or sends a corps of infantrymen into full scale, disorganized retreat

Ref: Piscitello and Chapin, p413

---

Routeing or Routing

- Routeing: British
- Routing: American

Since Oxford English Dictionary is much heavier than any other dictionary of American English, British English generally prevails in the documents produced by ISO and CCITT; wherefore, most of the international standards for routing standards use the routeing spelling.

Ref: Piscitello and Chapin, p413
Routing Techniques Elements

- **Performance criterion**: Hops, Distance, Speed, Delay, Cost
- **Decision time**: Packet, session
- **Decision place**: Distributed, centralized, Source
- **Network information source**: None, local, adjacent nodes, nodes along route, all nodes
- **Routing strategy**: Fixed, adaptive, random, flooding
- **Adaptive routing update time**: Continuous, periodic, topology change, major load change

Dijkstra’s Algorithm

- **Goal**: Find the least cost paths from a given node to all other nodes in the network
- **Notation**:
  - $d_{ij}$ = Link cost from i to j if i and j are connected
  - $D_n$ = Total path cost from s to n
  - $M$ = Set of nodes so far for which the least cost path is known
- **Method**:
  - Initialize: $M=\{s\}$, $D_n = d_{sn}$
  - Find node $w \not\in M$, whose $D_n$ is minimum
  - Update $D_n$
Example

Example (Cont)

<table>
<thead>
<tr>
<th>M</th>
<th>D2</th>
<th>Path</th>
<th>D3</th>
<th>Path</th>
<th>D4</th>
<th>Path</th>
<th>D5</th>
<th>Path</th>
<th>D6</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 8.4a
Dijkstra's routing algorithm

Apply to the following network and compute paths from node 1.

<table>
<thead>
<tr>
<th>M</th>
<th>D2 Path</th>
<th>D3 Path</th>
<th>D4 Path</th>
<th>D5 Path</th>
<th>D6 Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>1</td>
<td>1-2</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>{1,2}</td>
<td>1</td>
<td>1-2</td>
<td>4</td>
<td>1-2-3-5</td>
</tr>
<tr>
<td>3</td>
<td>{1,2,3}</td>
<td>1</td>
<td>1-2</td>
<td>4</td>
<td>1-2-3-5</td>
</tr>
<tr>
<td>4</td>
<td>{1,2,3,5}</td>
<td>1</td>
<td>1-2</td>
<td>4</td>
<td>1-2-3-5</td>
</tr>
<tr>
<td>5</td>
<td>{1,2,3,4,5}</td>
<td>1</td>
<td>1-2</td>
<td>4</td>
<td>1-2-3-5</td>
</tr>
<tr>
<td>6</td>
<td>{1,2,3,4,5,6}</td>
<td>1</td>
<td>1-2</td>
<td>4</td>
<td>1-2-3-5</td>
</tr>
</tbody>
</table>
**Bellman-Ford Algorithm**

- **Notation:**
  - $h =$ Number of hops being considered
  - $D^{(h)}_n =$ Cost of $h$-hop path from $s$ to $n$

- **Method:**
  - Find all nodes 1 hop away
  - Find all nodes 2 hops away
  - Find all nodes 3 hops away

- **Initialize:**
  - $D^{(h)}_n = \infty$ for all $n \neq s$; $D^{(h)}_n = 0$ for all $h$

- **Find jth node for which $h+1$ hops cost is minimum**
  - $D^{(h+1)}_n = \min_j \{D^{(h)}_j + D_{jn}\}$

---

**Example**

![Example Diagram](Fig 8.9b)
### Example (Cont)

<table>
<thead>
<tr>
<th>h</th>
<th>( D_{2}^{(h)} ) Path</th>
<th>( D_{3}^{(h)} ) Path</th>
<th>( D_{4}^{(h)} ) Path</th>
<th>( D_{5}^{(h)} ) Path</th>
<th>( D_{6}^{(h)} ) Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \infty ) -</td>
<td>( \infty ) -</td>
<td>( \infty ) -</td>
<td>( \infty ) -</td>
<td>( \infty ) -</td>
</tr>
<tr>
<td>1</td>
<td>2 1-2</td>
<td>5 1-3</td>
<td>1 1-4</td>
<td>( \infty ) -</td>
<td>( \infty ) -</td>
</tr>
<tr>
<td>2</td>
<td>2 1-2</td>
<td>4 1-4-3</td>
<td>1 1-4</td>
<td>2 1-4-5</td>
<td>10 1-3-6</td>
</tr>
<tr>
<td>3</td>
<td>2 1-2</td>
<td>3 1-5-4-3</td>
<td>1 1-4</td>
<td>2 1-4-5</td>
<td>4 1-4-5-6</td>
</tr>
<tr>
<td>4</td>
<td>2 1-2</td>
<td>3 1-5-4-3</td>
<td>1 1-4</td>
<td>2 1-4-5</td>
<td>4 1-4-5-6</td>
</tr>
</tbody>
</table>

**Table 8.4b**

---

### Flooding

- Uses all possible paths
- Uses minimum hop path Used for source routing

**Fig 8.11b**
ARPAnet Routing (1969-78)

- Features: Cost=Queue length,
- Each node sends a vector of costs (to all nodes) to neighbors. Distance vector
- Each node computes new cost vectors based on the new info using Bellman-Ford algorithm


ARPAnet Routing (1979-86)

- Problem with earlier algorithm: Thrashing (packets went to areas of low queue length rather than the destination), Speed not considered
- Solution: Cost=Measured delay over 10 seconds
- Each node floods a vector of cost to neighbors. Link-state. Converges faster after topology changes.
- Each node computes new cost vectors based on the new info using Dijkstra’s algorithm

![Diagram of ARPAnet Routing (1979-86)](https://example.com/diagram-arpanet-routing-1979-86)
ARPAnet Routing (1987+)

- Problem with 2nd Method: Correlation between delays reported and those experienced later: High in light loads, low during heavy loads ⇒ Oscillations under heavy loads ⇒ Unused capacity at some links, over-utilization of others, More variance in delay more frequent updates More overhead

![Fig 8.15](image1)

Routing Algorithm

- Delay is averanged over 10 s
- Link utilization = \( r = \frac{2(s-t)}{(s-2t)} \)
  where \( t = \) measured delay,
  \( s = \) service time per packet (600 bit times)
- Exponentially weighted average utilization
  \[ U(n+1) = \alpha U(n) + (1-\alpha)r(n+1) \]
  \[ = 0.5 U(n) + 0.5 r(n+1) \] with \( \alpha = 0.5 \)
- Link cost = \( f_n(U) \)

![Fig 8.16](image2)
Summary

- Connection-oriented and Connectionless
- Routing: Least-cost, Flooding, Adaptive
- Dijkstra’s and Bellman-Ford algorithms
- ARPAnet

Homework

- Exercise 8.4 (in b assume a unidirectional single loop), 8.10, 8.15
- Due: Next class