Att göra ...

• Frågor på Routerprojektet ställs till William
  – Kolla Projekt på kursens hemsida
Circuit switched routing
Routing in Packet Switching Networks

• Key design issue for (packet) switched networks
• Select route across network between end nodes
• Characteristics required:
  – Correctness
  – Simplicity
  – Robustness
  – Stability
  – Fairness
  – Optimality
  – Efficiency
Routing Strategies - Flooding

• Packet sent by node to every neighbor
• Eventually multiple copies arrive at destination
• No network information required
• Each packet is uniquely numbered so duplicates can be discarded
• Need to limit incessant retransmission of packets
  – Nodes can remember identity of packets retransmitted
  – Can include a hop count in packets
Packet-switched Routing

Choosing an optimal path

• According to a cost metric
• Decentralised: each router has full/necessary information
Router

- Internetworking device
  - Passes data packets between networks
  - Checks *Network Layer* addresses
  - Uses Routing/forwarding tables

Two functions:
1. **Routing**
2. **Forwarding**
Router Architecture Overview
Input Port

Decentralized switching:
- Given destination, lookup output port using routing table in input port memory
- Goal: complete input port processing at ‘line speed’

Physical layer: bit-level reception

Data link layer: e.g., Ethernet
Input Port Queuing

• Fabric slower that sum of input ports $\rightarrow$ **queuing**
• **Delay and loss** due to input buffer overflow
• **Head-of-the-Line (HOL) blocking**: Datagram at front of queue prevents others in queue from proceeding

![Diagram showing input port queuing and HOL blocking]
Output Port

Priority Scheduling:

• Scheduling discipline may choose among queued datagrams for transmission
Output Port Queuing

- Datagrams’ arrival rate through the switch exceeds the transmission rate of the output line → buffering
- Delay and loss due to output port buffer overflow
Switching Fabrics

Crossbar

Switching fabric

Routing processor

Memory

Bus
Routing Tables and Forwarding Table

- RIP Process
  - RIP Routing tables
- BGP Process
  - BGP Routing tables
- OSPF Process
  - OSPF Routing tables
- Forwarding Table Manager
  - Forwarding Table
- Static routing table
- OS kernel

Relates to:
- RIP Domain
- OSPF Domain
- BGP
Performance Criteria

• Used for selection of route
• Simplest is to choose "minimum hop"
• Can be generalized as "least cost" routing
• Because "least cost" is more flexible it is more common than "minimum hop"
Table 19.1
Elements of Routing Techniques for Packet-Switching Networks

<table>
<thead>
<tr>
<th>Performance Criteria</th>
<th>Network Information Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of hops</td>
<td>None</td>
</tr>
<tr>
<td>Cost</td>
<td>Local</td>
</tr>
<tr>
<td>Delay</td>
<td>Adjacent node</td>
</tr>
<tr>
<td>Throughput</td>
<td>Nodes along route</td>
</tr>
<tr>
<td></td>
<td>All nodes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision Time</th>
<th>Network Information Update Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet (datagram)</td>
<td>Continuous</td>
</tr>
<tr>
<td>Session (virtual circuit)</td>
<td>Periodic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision Place</th>
<th>Major load change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each node (distributed)</td>
<td>Topology change</td>
</tr>
<tr>
<td>Central node (centralized)</td>
<td></td>
</tr>
<tr>
<td>Originating node (source)</td>
<td></td>
</tr>
</tbody>
</table>
Figure 19.1 Example Network Configuration
Best Path: Decision Time and Place

Decision time (when?)
- Packet or virtual circuit basis
- Fixed or dynamically changing

Decision place (where?)
- Distributed - made by each node
  - More complex, but more robust
- Centralized – made by a designated node
- Source – made by source station
Network Information Source and Update Timing

• Routing decisions usually based on knowledge of network, traffic load, and link cost
  – Distributed routing
    • Using local knowledge, information from adjacent nodes, information from all nodes on a potential route
  – Central routing
    • Collect information from all nodes

Issue of update timing

• Depends on routing strategy
• Fixed - never updated
• Adaptive - regular updates
Routing Strategies - Fixed Routing

- Use a **single permanent** route for each source to destination pair of nodes
- Determined using a least cost algorithm
- **Route is fixed**
  - Until a change in network topology
  - Based on expected traffic or capacity
- Advantage is **simplicity**
- Disadvantage is **lack of flexibility**
  - Does not react to network failure or congestion
### CENTRAL ROUTING DIRECTORY

<table>
<thead>
<tr>
<th>From Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>---</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>---</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>---</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>---</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>---</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>---</td>
</tr>
</tbody>
</table>

**Figure 19.2** Fixed Routing (using Figure 19.1)
Routing Strategies - Adaptive Routing

• Used by almost all packet switching networks
• **Routing decisions change as conditions on the network change due to failure or congestion**
• Requires information about network

<table>
<thead>
<tr>
<th>Disadvantages:</th>
<th>Decisions more complex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tradeoff between quality of network information and overhead</td>
</tr>
<tr>
<td></td>
<td>Reacting too quickly can cause oscillation</td>
</tr>
<tr>
<td></td>
<td>Reacting too slowly means information may be irrelevant</td>
</tr>
</tbody>
</table>
Classification of Adaptive Routing Strategies

• A convenient way to classify is on the basis of information source

| Local (isolated) | Route to outgoing link with shortest queue  
                      | Can include bias for each destination  
                      | Rarely used - does not make use of available information |
|------------------|-------------------------------------------------|
| Adjacent nodes   | Takes advantage of delay and outage information  
                      | Distributed or centralized  |
| All nodes        | Like adjacent  |
ARPANET Routing Strategies
1st Generation

Distance Vector Routing

• 1969
• Distributed adaptive using estimated delay
  – Queue length used as estimate of delay
• Version of Bellman-Ford algorithm
• Node exchanges delay vector with neighbors
• Update routing table based on incoming information
• Doesn't consider line speed, just queue length and responds slowly to congestion
A note: Presumed knowledge

• Distance Vector
• Bellman-Ford
ARPANET Routing Strategies
2nd Generation

Link-State Routing

• **1979**
• Distributed adaptive using *delay* criterion
  – Using timestamps of arrival, departure and ACK times
• Re-computes average delays every 10 seconds
• Any changes are flooded to all other nodes
• Re-computes routing using **Dijkstra’s algorithm**
• Good under light and medium loads
• Under heavy loads, little correlation between reported delays and those experienced
ARPANET Routing Strategies
3rd Generation

• **1987**
• Link cost calculation changed
  – Damp routing oscillations
  – Reduce routing overhead
• Measure average delay over last 10 seconds and transform into link utilization estimate
• Normalize this based on current value and previous results
• **Set link cost as function of average utilization**
Autonomous Systems (AS)

• Exhibits the following characteristics:
  – Is a set of routers and networks managed by a single organization
  – Consists of a group of routers exchanging information via a common routing protocol
  – Except in times of failure, is connected (in a graph-theoretic sense); there is a path between any pair of nodes
Interior Router Protocol (IRP)

Interior Gateway Protocol (IGP)

- A shared routing protocol which passes routing information between routers within an AS
- Custom tailored to specific applications and requirements

Examples
- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)
Figure 19.9 Application of Exterior and Interior Routing Protocols
Exterior Router Protocol (ERP)
Exterior Gateway Protocol (EGP)

• Protocol used to pass routing information between routers in different ASs
• Will need to pass less information than an IRP for the following reason:
  – If a datagram is to be transferred from a host in one AS to a host in another AS, a router in the first system need only determine the target AS and devise a route to get into that target system
  – Once the datagram enters the target AS, the routers within that system can cooperate to deliver the datagram
  – The ERP is not concerned with, and does not know about, the details of the route
Graphical representation of a net

a. An internet

b. The weighted graph

Legend:
- Router
- LAN
- Node
- WAN
- Edge
- 2, 3, ... Costs
What is an end node?

Problem: The LANs are our destinations/end nodes, not the routers
A more realistic representation

• Solution: Nets and routers are all nodes in the tree.

• Routers hold tables how to reach nets and what is the *next hop* for to get there
Approaches to Routing

- Internet routing protocols employ one of three approaches to gathering and using routing information:
  - Distance-vector routing
  - Link-state routing
  - Path-vector routing (next lecture)
Routing Algorithms and Protocols

Routing protocols

- Intradomain
  - Distance vector: RIP
  - Link state: OSPF

- Interdomain
  - Path vector: BGP
Distance-Vector Routing

• Requires that each node exchange information with its neighboring nodes
  – Two nodes are said to be neighbors if they are both directly connected to the same network
• Used in the first-generation routing algorithm for ARPANET
• Each node maintains a vector of link costs for each directly attached network and distance and next-hop vectors for each destination
• Routing Information Protocol (RIP) uses this approach
RIP (Routing Information Protocol)

• Included in BSD-UNIX Distribution in 1982
• Distance metric:
  – # of hops (max 15) to destination network
• Distance vectors:
  – exchanged among neighbours every 30” via Response Message (advertisement)
• Implementation:
  – Application layer protocol, uses UDP/IP
# A RIP Forwarding/Routing Table

<table>
<thead>
<tr>
<th>Destination=net</th>
<th>Cost</th>
<th>Next hop=router</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>D</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>
RIP update message

• Contains the whole forwarding table
• Action on reception:
  – Add 1 to cost in received message
  – Change next hop to sending router
  – Apply RIP updating algorithm

• IMPORTANT! Received update msgs identify neighbours!
RIP Updating Algorithm (Bellman-Ford)

if (advertised destination not in table)
{
    add new entry // rule #1
}
else if (adv. next hop = next hop in table)
{
    update cost // rule #2
}
else if (adv. cost < cost in table)
{
    replace old entry // rule #3
}
RIP Example
Two node instability/Count to infinity

a. Before failure
b. After link failure
c. After A is updated by B
d. After B is updated by A
e. Finally
Split Horizon breaks Count to infinity

I have a route to X, but I got it from A so I won’t tell A about it!
RIP: Link Failure and Recovery

- If no advertisement heard after 180”
  - Neighbour/link declared dead
  - Routes via neighbour invalidated
    (infinite distance = 16 hops)
  - New advertisements sent to neighbours
    (triggering a chain reaction if tables changed)
  - “Poison reverse” used to prevent count to infinity loops
  - “Good news travel fast, bad news travel slow”
Routing Algorithms and Protocols

Routing protocols

Intradomain

Distance vector

RIP

Link state

OSPF

Interdomain

Path vector

BGP
Link-State Routing

• Designed to overcome the drawbacks of distance-vector routing
• When a router is initialized, it determines the link cost on each of its network interfaces
• The router then advertises this set of link costs to all other routers in the internet topology, not just neighboring routers
• From then on, the router monitors its link costs
• Whenever there is a significant change the router again advertises its set of link costs to all other routers in the configuration
• The OSPF protocol is an example
• The second-generation routing algorithm for ARPANET also uses this approach
Open Shortest Path First (OSPF) Protocol

• RFC 2328
• Used as the interior router protocol in TCP/IP networks
• Computes a route through the internet that incurs the least cost based on a user-configurable metric of cost
• Is able to equalize loads over multiple equal-cost paths
OSPF (Open Shortest Path First)

• Divides domain into areas
  – Limits flooding for efficiency
  – One ”backbone” area connects all

• Distance metric:
  – Cost to destination network
Areas, Router and Link Types

Autonomous system

Area 0 (backbone)

Area border router

Area 1

Area border router

Area 2

Backbone router

AS boundary router

To other ASs
Point-to-Point Link

- Connects two routers
- No need for addresses
Transient Link

a. Transient network

b. Unrealistic representation

c. Realistic representation

Designated router
Stub Link

Autonomous system

Area 1
Area border router

Area 2
Area border router

Area 0 (backbone)

Backbone router
AS boundary router

Ethernet

a. Stub network

b. Representation

Designated router
Link State Advertisements

- What to advertise?
  - Different entities as nodes
  - Different link types as connections
  - Different types of cost
Router Link Advertisement
Network Link Advertisement

- Network is a passive entity
  - It cannot advertise itself
Summary Link to Network

• Done by area border routers
  – Goes through the backbone
Summary Link to AS Boundary Router

• Links to other domains
  "autonomous systems"
External Link Advertisement

• Link to a single network outside the domain
Hello message

• Find neighbours
• Keep contact with neighbours: I am still alive!
• Sent out periodically (typically every 10th second)
• If no hellos received during holdtime (typically 30 seconds), neighbour declared dead.

• Compare RIP update messages
Dijkstra’s Algorithm

• Finds shortest paths from given source node to all other nodes
• Develop paths in order of increasing path length
• Algorithm runs in stages
  – Each time adding node with next shortest path
• Algorithm terminates when all nodes have been added to \( T \)
<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>R3</td>
<td>10</td>
</tr>
<tr>
<td>N2</td>
<td>R3</td>
<td>10</td>
</tr>
<tr>
<td>N3</td>
<td>R3</td>
<td>7</td>
</tr>
<tr>
<td>N4</td>
<td>R3</td>
<td>8</td>
</tr>
<tr>
<td>N6</td>
<td>R10</td>
<td>8</td>
</tr>
<tr>
<td>N7</td>
<td>R10</td>
<td>12</td>
</tr>
<tr>
<td>N8</td>
<td>R10</td>
<td>10</td>
</tr>
<tr>
<td>N9</td>
<td>R10</td>
<td>11</td>
</tr>
<tr>
<td>N10</td>
<td>R10</td>
<td>13</td>
</tr>
<tr>
<td>N11</td>
<td>R10</td>
<td>14</td>
</tr>
<tr>
<td>H1</td>
<td>R10</td>
<td>21</td>
</tr>
<tr>
<td>R5</td>
<td>R5</td>
<td>6</td>
</tr>
<tr>
<td>R7</td>
<td>R10</td>
<td>8</td>
</tr>
<tr>
<td>N12</td>
<td>R10</td>
<td>10</td>
</tr>
<tr>
<td>N13</td>
<td>R5</td>
<td>14</td>
</tr>
<tr>
<td>N14</td>
<td>R5</td>
<td>14</td>
</tr>
<tr>
<td>N15</td>
<td>R10</td>
<td>17</td>
</tr>
</tbody>
</table>

**Table 19.3**

Routing Table for R6
Comparison

- Bellman-Ford
  - Calculation for node \( n \) needs link cost to neighboring nodes plus total cost to each neighbor from \( s \)
  - Each node can maintain set of costs and paths for every other node
  - Can exchange information with direct neighbors
  - Can update costs and paths based on information from neighbors and knowledge of link costs

- Dijkstra
  - Each node needs complete topology
  - Must know link costs of all links in network
  - Must exchange information with all other nodes