Basic Routing Concepts

Surasak Sanguanpong
nguan@ku.ac.th
http://www.cpe.ku.ac.th/~nguan

Last updated: July 30, 1999
Agenda

- Basic concepts
- Routing components
- Classes of routing protocol
What’s routing

- routing - path finding from one end to the other
  - routing occurs at layer 3
  - bridging occurs at layer 2
IP Routing

- IP performs:
  - search for a matching host address
  - search for a matching network address
  - search for a default entry

- Routing done by IP router, when it searches the routing table and decide which interface to end a packet out.
Routing Tables

- Routing is carried out in a router by consulting the routing table.
- No unique format for routing tables, typically:
  - address of a destination
  - IP address of next hop router
  - network interface to be used
  - subnet mask for the this interface
  - distance to the destination
Routing Component

- three important routing elements:
  - algorithm
  - database
  - protocol
- algorithm: can be differentiate based on several key characteristics
- database: table in routers or routing table
- protocol: the way information for routing to be gathered and distributed
Routing algorithm

- **design goals**
  - **optimality** - compute the best route
  - **simplicity/low overhead** - efficient with a minimum software and utilization overhead
  - **robustness/stability** - perform correctly in the face of unusual circumstances
  - **rapid convergence** - responds quickly when the network changes
  - **flexibility** - accurate adapt to a variety of network
Routing Protocols

- **Routing protocol** - protocol to exchange of information between routers about the current state of the network

- **Routing protocol jobs**
  - create routing table entries
  - keep routing table up-to-date
  - compute the best choice for the next hop router
Routing metrics

- How do we decide that one route is better than another?
- Solution: using a metric as a measurement to compare routes
- Metrics may be distance, throughput, delay, error rate, and cost.
- Today IP supports Delay, Throughput, Reliability and Cost (DTRC)
Hop Count

- A hop is defined as a passage through one router
- For some protocols, hop count means the number of links, rather than the number of routers
Routing algorithm types

- static V.S. dynamic
- source routing V.S. hop-by-hop
- centralize V.S. distributed
- distance vector V.S. link state
Routing algorithm: static route

- manually config routing table
- can’t react dynamically to network change such as router’s crash
- work well with small network or simple topology
- unix hosts use command `route` to add an entry
Routing algorithm: static technique

- **Flooding**
  - every incoming packet is sent out every outgoing
  - retransmit on all outgoing at each node
  - simple technique, require no network information
  - generate vast numbers of duplicate packet
Routing algorithm: Dynamic Route

- **Dynamic route**
  - network protocol adjusts automatically for topology or traffic changes
  - unix hosts run routing daemon *routed* or *gated*
**Routing algorithm:** Dynamic Route operation

- Routing protocol maintains and distributes routing information
Routing algorithm: source routing

- source routing
  - source will determine the entire route
  - routers only act as sore-forward devices
- hop-by-hop
  - routers determine the path based on theirs own calculation
Routing algorithm: distance vector

- **distance** means routing metric
- **vector** means destination
- flood routing table only to its neighbors
- RIP is an example
- also known as **Bellmann-Ford algorithm** or **Ford-Fulkerson algorithm**
Routing algorithm: link state

- flood routing information to all nodes
- each router finds who is up and flood this information to the entire routers
- use the link state to build a shortest path map to everybody
- OSPF is an example
- also known as Shortest Path First (SPF) algorithm
Distance vector algorithm

- using hop count as a metric
- each router periodically sends a copy of its routing table to neighbors
  - send <network X, hopcount Y>

```
W X Y Z
R1 R2 R3

W        0
X         0   Y         1   Z         2
routing table

W        1
X         0   Y         0   Z         1
routing table

W        2
X         1   Y         0   Z         0
routing table
```
Distance vector routing update

- step by step from router to router
- slow convergence

1. Reconstruct R3’s routing table
2. Reconstruct R3’s routing table
3. Reconstruct R2’s routing table
4. Reconstruct R2’s routing table
5. R2 sends out the updated table
6. Reconstruct R1’s routing table
Distance vector: broadcast (I)

- the first round

![Diagram showing distance vectors and hops between nodes I, J, K, L, M, N, O, and R1, R2, R3, R4, R5.]
Distance vector: broadcast (II)

- the second round
Distance vector: broadcast (III)

- the third round

- I, 1 hop
  - J, 1 hop
  - K, 2 hops
  - L, 2 hops
  - M, 2 hops
  - N, 2 hops
  - O, 3 hops

- J, 1 hop
  - K, 1 hop
  - M, 1 hop
  - N, 1 hop
  - I, 2 hops
  - L, 2 hops
  - O, 2 hops

- N, 1 hop
  - O, 1 hop
  - J, 2 hops
  - K, 2 hops
  - M, 1 hop
  - L, 2 hops
  - I, 3 hops

- L, 1 hop
  - M, 1 hop
  - O, 1 hop
  - I, 2 hops
  - K, 2 hops
  - J, 2 hops
  - N, 2 hops
Distance vector: crashed recovery

- **R3 crashed**
- new complete route of R1

R1 routing table

<table>
<thead>
<tr>
<th>net</th>
<th>hop</th>
<th>via</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>K</td>
<td>2</td>
<td>R2</td>
</tr>
<tr>
<td>L</td>
<td>2</td>
<td>R2</td>
</tr>
<tr>
<td>M</td>
<td>2</td>
<td>R3</td>
</tr>
<tr>
<td>N</td>
<td>2</td>
<td>R3</td>
</tr>
<tr>
<td>O</td>
<td>3</td>
<td>R5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>net</th>
<th>hop</th>
<th>via</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>K</td>
<td>2</td>
<td>R2</td>
</tr>
<tr>
<td>L</td>
<td>2</td>
<td>R2</td>
</tr>
<tr>
<td>M</td>
<td>3</td>
<td>R2</td>
</tr>
<tr>
<td>N</td>
<td>4</td>
<td>R2</td>
</tr>
<tr>
<td>O</td>
<td>3</td>
<td>R2</td>
</tr>
</tbody>
</table>
Count to infinity

- R2 does not hear any thing from R3
- R1 says: don’t worry, I can reach R3 in 2 hops, R2 update hop count to 3
- R1 sees R2’s update, then update itself to 4 and so on, ...

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>J</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td>initial</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1st round</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2nd round</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3rd round</td>
</tr>
</tbody>
</table>

hop count to R3

R3 crashed
Solving count to infinity

- solve by set distance “16” as infinity
- no destination can be more than 15 hops away from any other
- split-horizon: distance to X is not reported on the line that packet for X are sent
- Poison-reverse: send a route update that specifies that the distance is infinity

```
I   J   R3 crashed
R1    L    R2
```

to R3 = ∞ to R3 = ∞
Link State Overview

- using cost as a metric
- exchange its connection and cost to its neighbors
- each router compute the set of optimum path to all destination (Shortest Path First)
Link State concept

- each router initially begins with directly connected network
- determine full knowledge of distant routers and theirs connection
Link State routing update

- send information to other routers
- fast convergence
## Comparison

<table>
<thead>
<tr>
<th>Distance Vector</th>
<th>Link State</th>
</tr>
</thead>
<tbody>
<tr>
<td>pass a copy of whole routing table</td>
<td>pass links state update</td>
</tr>
<tr>
<td>add metric from router to router</td>
<td>calculate the shortest path to other routers</td>
</tr>
<tr>
<td>frequent periodic update: slow convergence</td>
<td>event updated: fast convergence</td>
</tr>
</tbody>
</table>