CS2105 Lecture 7
Routing
3 March, 2014
After this class, you are expected to understand

- how longest prefix forwarding in a router works
- the purpose of routing protocols on the Internet
- the differences between inter-domain and intra-domain routing.
- the workings of link-state and distance vector routing
- the principle of Bellman-Ford equation
- how RIP works
``Millions of Routers Work in Concert
to Route Packets on the Internet,
Based on This One Simple Equation.
Amazing."
Transport

IP

ICMP

Routing Protocols

Link
## Forwarding Table

<table>
<thead>
<tr>
<th>destinations</th>
<th>outgoing interface</th>
</tr>
</thead>
</table>

Longest Prefix Matching

<table>
<thead>
<tr>
<th>address prefix</th>
<th>outgoing interface</th>
</tr>
</thead>
</table>
The Internet is a "network-of-networks", organized into autonomous systems (AS), each is owned by an organization.
Inter-AS routing
Intra-AS routing
Abstract view of intra-domain routing
Routing: finding the least cost path in a graph.
Link-State Routing
1. Broadcast link cost to each other
2. Compute least cost path locally
1. Broadcast link cost to each other
2. Compute least cost path locally

We will not cover Step 2 in CS2105. The algorithm will be covered in CS2010.
Distance Vector Routing
1. Swap local view with neighbors
2. Update own's local view
3. Repeat
decentralized
self-terminating
iterative
asynchronous
$d_x(y)$: the total cost to reach $y$ from $x$'s view
$c(x, y)$: the link cost between $x$ and $y$. 
\[ d_x(y) = \min_i \{ c(x, i) + d_i(y) \} \]
\[ d_x(y) = \min_i \{ c(x, i) + d_i(y) \} \]

<table>
<thead>
<tr>
<th>dest</th>
<th>next</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>z</td>
<td>10</td>
</tr>
<tr>
<td>w</td>
<td>z</td>
<td>23</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>11</td>
</tr>
<tr>
<td>y</td>
<td>x</td>
<td>17</td>
</tr>
</tbody>
</table>
distance vector is the `local view" exchanged between neighbors.

<table>
<thead>
<tr>
<th>dest</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>10</td>
</tr>
<tr>
<td>w</td>
<td>23</td>
</tr>
<tr>
<td>x</td>
<td>11</td>
</tr>
<tr>
<td>y</td>
<td>17</td>
</tr>
</tbody>
</table>
The diagram shows a network with nodes labeled z, y, and x. There are connections between these nodes with labels 1, 7, and 2.

A table is also present with columns labeled "to x", "to y", and "to z". The rows in the table are labeled @x, @y, and @z.
\[ d_x(y) = \min_i \{ c(x, i) + d_i(y) \} \]
decentralized
self-terminating
iterative
asynchronous
\[ d_x(y) = \min_i \{ c(x, i) + d_i(y) \} \]
``count-to-infinity''
``poisoned reverse''
Without poisoned reverse

\[
\text{for all neighbor } n \text{ do} \\
\quad \text{distance vector } v_n \leftarrow [ \ ] \\
\quad \text{for all entry } e \text{ in local table do} \\
\quad \quad \text{add } (e.\text{dest}, e.\text{cost}) \text{ to } v_n \\
\quad \text{end for} \\
\text{send } v_n \text{ to } n \\
\text{end for}
\]
With poisoned reverse

\[\textbf{for all} \text{ neighbor } n \text{ do}\]
\[\text{distance vector } v_n \leftarrow [\ ]\]
\[\textbf{for all} \text{ entry } e \text{ in local table do}\]
\[\text{if } e.\text{next} \text{ is } n \text{ then}\]
\[\text{add } (e.\text{dest}, \infty) \text{ to } v_n\]
\[\text{else}\]
\[\text{add } (e.\text{dest}, e.\text{cost}) \text{ to } v_n\]
\[\text{end if}\]
\[\textbf{end for}\]
\[\text{send } v_n \text{ to } n\]
\[\textbf{end for}\]
link state vs. distance vector
message overhead
convergence speed
robustness (error)
RIP: Routing Information Protocol
<table>
<thead>
<tr>
<th>dest</th>
<th>next hop</th>
<th>hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>z</td>
<td>3</td>
</tr>
<tr>
<td>w</td>
<td>z</td>
<td>7</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>y</td>
<td>z</td>
<td>5</td>
</tr>
</tbody>
</table>
Route is static and load insensitive.
Exchange routing table every 30 seconds over UDP port 520
``Self-repair'": if no update from a neighbor for 3 minutes, assume neighbor has failed.
Inter-AS routing
Intra-AS routing
BGP: Border Gateway Protocol
Hot Potato Routing: route to AS whose gateway has the least cost.