More on Network Routing and Internet Protocol

Outline:
- Routing principles
  - Link state routing (based on centralized information)
  - Distance vector routing (distributed)
- Internet protocol
  - Addressing
  - Format of IP datagrams
  - Internet routing

Distance Vector Routing

- routing table (at each host): the next hop for each destination in the network
- distance vector routing: the routing table is constructed from a distance vector at each node
- distance vectors can be maintained in a decentralized fashion

Distance Vector: An Example

Distance vector at node E

$D(E, x, y) = c(E, D) + \text{shortest}(D, C)$

- $D(E, D, C) = 2 + 2 = 4$
- $D(E, D, A) = 2 + 3 = 5$  \text{loop!}
- $D(E, B, A) = 8 + 6 = 14$  \text{loop!}

Distance Vector to Routing Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>cost to destination via D(E,y)</th>
<th>Outgoing link to use, cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 14 5</td>
<td>A A,1</td>
</tr>
<tr>
<td>B</td>
<td>7 8 5</td>
<td>B D,5</td>
</tr>
<tr>
<td>C</td>
<td>6 9 4</td>
<td>C D,4</td>
</tr>
<tr>
<td>D</td>
<td>4 11 2</td>
<td>D D,4</td>
</tr>
</tbody>
</table>

Outgoing link to use, cost
Distance Vector Routing: Constructing Distance Vector

Information:
- local link cost
- distance vector → routing table

Recompute caused by:
- local link cost change
- message from neighbor: its routing table has changed

Distributed:
- if my own routing table has changed, notify my neighbors

Distance vector can be built on neighbors' routing tables!

Each node:
- wait for (change in local link cost or msg from neighbor)
- recompute distance vector
- if the local routing table has changed, notify neighbors

Distance Vector Algorithm: An Example

Distance Vector: Link Cost Changes

Link cost changes:
- node detects local link cost change
- recompute - updates local distance vector
- if cost change in least cost path, notify neighbors

"good news travels fast"
Distance Vector: Link Cost Changes

Link cost changes:
- good news travels fast
- bad news travels slow - "count to infinity" problem!

algorithm continues on!

Distance Vector: Poisoned Reverse

If Z routes through Y to get to X:
- Z tells Y its (Z's) distance to X is infinity (so Y won't route to X via Z)
- Will this completely solve count to infinity problem?

Comparison of LS and DV Algorithms

Speed of convergence
- **LS**: $O(n \log n)$ algorithm
  - may have oscillations
- **DV**: convergence time varies
  - count-to-infinity problem

Robustness: what happens if router malfunctions?
- **LS**: node can advertise incorrect link cost
- **DV**: node can advertise incorrect routing table

Could there be routing loops?

Outline

- Routing principles
  - Link state routing (based on centralized information)
  - Distance vector routing (distributed)
- Internet protocol
  - Addressing
  - Format of IP packets
  - Internet routing
The Internet Network Layer

- Transport layer: TCP, UDP
- Routing protocols: RIP, OSPF, BGP
- The rest of the IP protocol
  - routing table
  - error reporting
  - packet format
  - packet handling conventions
- ICMP protocol

Link layer

Network layer

IP Addressing: Introduction

- IP address:
  - 32-bit identifier for each host, router interface
  - interface: connecting point into each data link
  - routers typically have multiple interfaces
  - hosts often have single interfaces

- IP address:
  - network part (high order bits)
  - host part (low order bits)
  - What's an IP network? (from IP address perspective)
    - can physically reach each other without intervening router (intervening switches?)
    - device interfaces with same network part of IP address

IP Addressing

- How to find the IP networks?
  - Detach each interface from router, host
  - create "islands of isolated networks"

- Interconnected system consisting of six networks
### IP Addresses

Given notion of "network", let's re-examine IP addresses:

"class"-based addressing:

<table>
<thead>
<tr>
<th>Class</th>
<th>Network</th>
<th>Host</th>
<th>Address Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td></td>
<td>10.0.0.0 to 127.255.255.255</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td></td>
<td>128.0.0.0 to 191.255.255.255</td>
</tr>
<tr>
<td>C</td>
<td>110</td>
<td></td>
<td>192.0.0.0 to 223.255.255.255</td>
</tr>
<tr>
<td>D</td>
<td>110</td>
<td></td>
<td>224.0.0.0 to 239.255.255.255</td>
</tr>
</tbody>
</table>

32 bits

---

### Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).