CSE331: Introduction to Networks and Security

Lecture 9
Fall 2002
Recap

- IPv4 Adressing
  - Hierarchical names
  - Subnetting

- Today:
  - DNS
  - IP Routing
Domain Name System

- System for mapping mnemonic names for computers into IP addresses.
  
  zeta.cis.upenn.edu → 158.130.12.244

- Domain Hierarchy
- Name Servers
- Name Resolution
Domain Name Hierarchy

cornell ... upenn

cisco ... yahoo

cis

com

wharton ...

seoas

gov

org

mil

net

edu

arpa

nsf

nasa

CSE331 Fall 2002
Hierarchy of Name Servers

- Root Name Server
  - Cornell Name Server
  - Upenn Name Server
    - CIS Name Server
    - SEAS Name Server
    - Wharton Name Server
Records on Name Servers

• < Name, Value, Type, Class >

• Types
  – A  Host to address mappings
  – NS  Name server address mappings
  – CNAME  Aliases
  – MX  Mail server mappings

• Class IN for IP addresses
Name resolution

client

Local Name server

Root Name server

Upenn Name server

CIS Name server

zeta.cis.upenn.edu

198.168.0.100

zeta.cis.upenn.edu

128.196.128.233

zeta.cis.upenn.edu

198.168.0.1

zeta.cis.upenn.edu

198.168.0.100
IP Routing

• Begin by partitioning problem:
  • Intradomain Routing
    – Inside *administrative domains* (AD’s)
  • Interdomain Routing
    – Between administrative domains (e.g., companies)
    – Exterior Gateway Protocol (EGP)
    – Border Gateway Protocol (BGP) [Replaced EGP]
Intradomain Routing

• RIP - Routing Information Protocol
  – Uses distance vector algorithm
  – Limited to small nets; <15 hops

• OPSF - Open Shortest Path First
  – Augmented version of link-state
  – Augmentation includes authentication, load-balancing, and defined areas
Distance Vector Algorithm (RIP)

- Similar to the Spanning Tree Algorithm
  - Except that information about distance to ALL nodes is forwarded (not just info. about root.)
  - Sometimes called Bellman-Ford algorithm

- Each node constructs a *Distance Vector*
  - Contains distances (costs) to reach all other nodes
  - Initially:
    - Distance to neighbors = 1
    - Distance to others = $\infty$
  - Routing table reflects node’s beliefs
Example Network Graph

A’s initial information

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Cost</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>∞</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>∞</td>
<td>-</td>
</tr>
</tbody>
</table>
Iteration Steps

- Each host sends its DV to its neighbors
- Neighbors can update their distance vectors and routing information accordingly.
  - As in spanning tree, the nodes ignore worse information
  - Update any better routes
- If host changed its tables, send new DV to neighbors
- After a few iterations, routing information converges
Example Iteration Steps

F sends A its DV.
- A discovers that G can be reached in two hops through F

C sends A its DV.
- A discovers that D can be reached in two hops through C

A’s initial information

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Cost</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>F</td>
</tr>
</tbody>
</table>
Details

• Note: No single host has all routing information.

• When to send update vectors?
  – When your routing table changes (triggered)
  – Periodically (“I’m alive!”)

• Detecting link/node failure
  – (1) Periodically exchange “I’m alive!” messages.
  – (2) Timeout mechanism
Stability problem

- Loops may form and stability cannot occur without “counting to infinity”.
- Timing of events may cause cycles of updates.
(Partial) Solutions for Stability

- Pick small value for “infinity”
  - If “infinity” is 16 then at most 15 hops in network
  - Distance of 16 considered unreachable

- Disallow cyclic updates
  - Called *split horizon* algorithm
  - Don’t send updates learned from a neighbor back to that neighbor
  - Only works for small (e.g. 2-hop) cycles
Open Shortest Path First (OSPF)

• Each node sends a *reliable flood* of information to all other nodes

• These Link-State Packets (LSPs) contain
  – ID of the node that created the packet
  – List of (neighbor, cost) pairs associated with the source node
  – Sequence Number (64bits—no wrapping)
  – Time To Live (ensure old info is eventually removed)
Reliable Flooding

- Transmission between adjacent routers is made reliable through ACKs & retransmission
- Source sends to all neighbors
- Each recipient
  - Sends to all neighbors except the one it got the message from
  - Ignores duplicates
OSPF continued

- Once all of the link-state info has been flooded each node has complete network topology
- Compute routing information using Dijkstra’s shortest-path algorithm
- Periodic updates and failure detection are like RIP.
OSPF Features

• Authentication of routing messages
  – Misconfigured or malicious host could advertise bad route info (i.e. reach anywhere in 0 hops)
  – (Eventually added to RIP too.)

• Additional Hierarchy
  – Partitions domains into areas
  – Reduces transmission & storage overhead

• Load Balancing
  – Multiple routes with same cost
  – Traffic evenly distributed