CSCI-1680
Network Layer:
Inter-domain Routing

Nick DeMarinis

Based partly on lecture notes by Rodrigo Fonseca, Rob Sherwood, David Mazières, Phil Levis, John Jannotti
Warmup

Suppose router R has the following table:

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Cost</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>S</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>S</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>U</td>
</tr>
</tbody>
</table>

What happens when it gets this update from router S?

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
</tr>
</tbody>
</table>
Administrivia

• You should have completed your IP milestone meeting
  – If not, contact us ASAP
• HW2: Due Friday
• IP
  – Due October 24
  – “Lab 2”: Important points and debugging setup, +5 on IP if you complete by October 15
Topics for today

• More on intra-domain routing
  – Challenges in RIP
  – Link-state routing

• Inter-domain routing: Intro to BGP
Dealing with Failures

• What happens when the D-A link fails?
RIP: WHAT HAPPENS WHEN A LINK FAILS?

\( (D,A_2) \) IN RIP

\( \infty = 16 \)

Diagram:

\[ D \rightarrow A \rightarrow B \]

\( (D, \infty) \)

\( \rightarrow C \)

\( (D, B, 3, 2) \)

\( \rightarrow B \)

\( \rightarrow C \)

OLD INFO OVERWRITES STATE!

ROUTING LOOP!

UPDATE OCCURS IN A LOOP WITH INCREASING COST UNTIL COST REACHES \( \infty \)

\( \rightarrow C \)

COUNT TO INFINITY

LONG CONVERGE TIME.
Dealing with Failures

- What happens when the D-A link fails?

=> “Count to Infinity” problem
How to avoid loops

• Does IP TTL help? DOESN'T SOLVE ROUTING PROBLEM.
• Simple approach: consider a small cost n (e.g., 16) to be infinity
  – After n rounds decide node is unavailable
  – But rounds can be long, this takes time

Problem: distance vector based only on local information
  NOT ENOUGH TO RESOLVE LOOPS, COUNT-TO-INFINITY (BUT THERE ARE TRICKS WE CAN DO)
One way: Split Horizon

- When sending updates to node A, don’t include routes you learned from A.
- Prevents B and C from sending cost 2 to A.

>$D \rightarrow A \rightarrow B$

>$\leftarrow\leftarrow$$\leftarrow\leftarrow$$\leftarrow\leftarrow$
**Split Horizon**

- If A uses N as NEXT HOP for D, DO NOT REPORT to N ABOUT D

  \[ \implies \text{PREVENTS "LINEAR" RESETTING LOOPS, BUT NOT OTHERS} \]

Where it would help:

1. D \rightarrow A \rightarrow B
2. (D, 1)
3. (D, 2)
4. (C, 1)
5. (D, 0)

\[ \text{① D-A link goes down} \]

\[ \text{② When B sends UPDATE to A, IT WOULD NOT TELL INCLUDE A} \]

\[ \text{③ A updates B w/ (D, 0)} \]
Split Horizon + Poison Reverse

- Rather than not advertising routes learned from A, explicitly include cost of $\infty$.
- Faster to break out of loops, but increases advertisement sizes.

\[ (D, D, 1) \quad (D, \infty) \]
\[ (D, A, 2) \]

**POISON REVERSE** is a common convention, might help time to converge in some cases, but hard to see effect vs. *Split Horizon*. 
BUT EVEN W/ SPLIT HORIZON
POISON REVERSE, CAN’T PREVENT
ALL LOOPS!

WHAT IF.

1. D-A FAILS
2. A UPDATES B (D,∞)
3. C SENDS (D,2) TO B;

\[ \text{\underline{\text{RACE CONDITION!}} \quad C \text{ MIGHT}
\\text{SEND OLD UPDATE TO B BEFORE}} \]
\[ C \text{ GETS UPDATE FROM A!} \]
4. B UPDATES A, OVERWRITES A’s ENTRY
5. ... COUNT TO INFINITY...

WHAT CAN WE DO?
Distance-vector updates

Even with split horizon + poison reverse, can still create loops with >2 nodes

What else can we do? Triggered updates:

- If A tells B and C immediately that its route changes, it can prevent this loop.
Distance-vector updates

Even with split horizon + poison reverse, can still create loops with >2 nodes

What else can we do?

- Triggered updates: send update as soon as link state changes
- Hold down: delay using new routes for certain time, affects convergence time
Practice

Routers A, B, C, D use RIP. When B sends a periodic update to A, what does it send...

- When using standard RIP?
- When using split horizon + poison reverse?
Routers A, B, C, D use RIP. When B sends a periodic update to A, what does it send... 
- When using standard RIP?
- When using split horizon + poison reverse?

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Cost</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>A</td>
</tr>
</tbody>
</table>

\[
\text{STANDARD: } (A, 1) (C, 1) (D, 2)
\]
\[
\text{SH+PR: } (A, \infty) (C, 1) (D, \infty)
\]
Link State Routing
Link State Routing

- **Strategy:** each router sends information about its neighbors to all nodes
  - (Not just our neighbors)
- **Link State Packet (LSP)**
  - ID of the node that created the LSP
  - Cost of link to each directly connected neighbor
  - Sequence number (SEQNO)
  - TTL
- Can use sequence number, TTL for versioning, and to avoid broadcasting updates in loops

Focus on consistent view state
Reliable Flooding

• Store most recent LSP from each node
  – Ignore earlier versions of the same LSP
• Forward LSP to all nodes but the one that sent it
• Generate new LSP periodically (increment SEQNO)
• Start at SEQNO=0 when reboot
  – If you hear your own packet with SEQNO=n, set your next SEQNO to n+1
• Decrement TTL of each stored LSP
  – Discard when TTL=0
Calculating best path

• Each node computes shortest paths from itself
• How? Dijkstra’s algorithm
  – Given: full graph of nodes
  – Find best next hop to each other node

• Computation: more expensive than DV
• Example: D: (D,0,-) (C,2,C) (B,5,C) (A,10,C)
Distance Vector vs. Link State

- # of messages (per node)
  - DV: $O(d)$, where $d$ is degree of node
  - LS: $O(nd)$ for $n$ nodes in system
Distance Vector vs. Link State

- # of messages (per node)
  - DV: $O(d)$, where $d$ is degree of node
  - LS: $O(nd)$ for $n$ nodes in system

- Computation
  - DV: convergence time varies (e.g., count-to-infinity)
  - LS: $O(n^2)$ with $O(nd)$ messages
Distance Vector vs. Link State

- **# of messages (per node)**
  - DV: $O(d)$, where $d$ is degree of node
  - LS: $O(nd)$ for $n$ nodes in system
- **Computation**
  - DV: convergence time varies (e.g., count-to-infinity)
  - LS: $O(n^2)$ with $O(nd)$ messages
- **Robustness: what happens with malfunctioning router?**
  - DV: Nodes can advertise incorrect path cost, which propagates through network
  - LS: Nodes can advertise incorrect link cost
Examples

- **RIPv2**
  - Fairly simple implementation of DV
  - RFC 2453 (38 pages)

- **OSPF (Open Shortest Path First)**
  - More complex link-state protocol
  - Adds notion of areas for scalability
  - RFC 2328 (244 pages)

- **ISIS (Intermediate System to Intermediate System)**
  - OSI standard (210 pages)
  - Link-state protocol (similar to OSPF)
  - Does not depend on IP
Why Inter vs. Intra
Why Inter vs. Intra

• Why not just use OSPF everywhere?
  – E.g., hierarchies of OSPF areas?
  – Hint: scaling is not the only limitation

**NEED A WAY TO CONTROL POLICY ON ROUTING — NOT ONE ADMINISTRATIVE DOMAIN**
Why Inter vs. Intra

• Why not just use OSPF everywhere?
  – E.g., hierarchies of OSPF areas?
  – Hint: scaling is not the only limitation

• BGP is a policy control and information hiding protocol
  – intra == trusted, inter == untrusted
  – Different policies by different ASs
  – Different costs by different ASs
Types of ASs

- **Local Traffic (Customers)**
  - Brown
  - Source or Destination for Traffic

- **Transit AS**
  - Traffic passes through to other networks
  - Charges customers
EXTRA MATERIAL

WE DIDN'T

COVER
OSPFv2

- Link state protocol
- Runs directly over IP (protocol 89)
  - Must provide its own reliability
- All exchanges are authenticated
- Adds notion of areas for scalability
OSPF Areas

• Area 0 is “backbone” area (includes all boundary routers)
• Traffic between two areas must always go through area 0
• Only need to know how to route exactly within area
• Otherwise, just route to the appropriate area
• Tradeoff: scalability versus optimal routes
RIPv2

- Runs on UDP port 520
  - (IP assignment: directly in IP, protocol 200)
- Link cost = 1
- Periodic updates every 30s, plus triggered updates
- Relies on count-to-infinity to resolve loops
  - Maximum diameter 15 ($\infty = 16$)
  - Supports split horizon, poison reverse
- Deletion
  - If you receive an entry with metric = 16 from parent OR
  - If a route times out