Routing

Goal: determine "good" path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:
- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

"good" path:
- typically means minimum cost path
- other def's possible

Routing Algorithm classification

Global or decentralized information?
Global:
- all routers have complete topology, link cost info
- "link state" algorithms
Decentralized:
- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?
Static:
- routes change slowly over time
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
  - gives routing table for that node
  - iterative: after k iterations, know least cost path to k dest.'s

Dynamic:
- routes change more quickly
  - periodic update
  - in response to link cost changes

A Link-State Routing Algorithm

Dijkstra's algorithm
- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
  - gives routing table for that node
- iterative: after k iterations, know least cost path to k dest.'s

Notation:
- $c(i,j)$: link cost from node $i$ to $j$. Cost infinite if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. $v$
- $p(v)$: predecessor node along path from source to $v$, that is next $v$
- $N$: set of nodes whose least cost path definitively known

Dijsktra’s Algorithm

Initialization:
1. $N = \{A\}$
2. for all nodes $v$
3. if $v$ adjacent to $A$
4. then $D(v) = c(A,v)$
5. else $D(v) = \infty$
6. end if
7. end for
8. Loop
9. find $w$ not in $N$ such that $D(w)$ is a minimum
10. add $w$ to $N$
11. update $D(v)$ for all $v$ adjacent to $w$ and not in $N$
12. $D(v) = \min(D(v), D(w) + c(w,v))$
13. if new cost to $v$ is either old cost to $v$ or known
14. shortest path cost to $w$ plus cost from $w$ to $v$ /$
15. until all nodes in $N$

Dijkstra’s algorithm: example

<table>
<thead>
<tr>
<th>Step</th>
<th>start N</th>
<th>D(B),p(B)</th>
<th>D(C),p(C)</th>
<th>D(D),p(D)</th>
<th>D(E),p(E)</th>
<th>D(F),p(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A</td>
<td>2,A</td>
<td>5,A</td>
<td>1,A</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>1</td>
<td>AD</td>
<td>2,A</td>
<td>4,D</td>
<td>2,B</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>2</td>
<td>ADE</td>
<td>2,A</td>
<td>3,E</td>
<td>4,E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ADEB</td>
<td>3,E</td>
<td>4,E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ADEBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ADEBCF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dijkstra’s algorithm, discussion

Algorithm computational complexity: n nodes
- each iteration: need to check all nodes, w, not in N
- n*(n+1)/2 comparisons: O(n²)
- more efficient implementations possible:
  - O(nlogn): Use a heap (sorted) to maintain interim table

Oscillations possible:
- e.g., link cost = amount of carried traffic

Distance Vector Routing Algorithm

iterative:
- continues until no nodes exchange info.
- self-terminating: no "signal" to stop
asynchronous:
- nodes need not exchange info/iterate in lock step!
distributed:
- each node communicates only with directly-attached neighbors

Distance Table: example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D(\cdot,C))</td>
<td>7</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>(D(\cdot,D))</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(D(\cdot,A))</td>
<td>6</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>(D(\cdot,B))</td>
<td>4</td>
<td>11</td>
<td>2</td>
</tr>
</tbody>
</table>

Distance table gives routing table

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

Distance Vector Routing: overview

Iterative, asynchronous:
- each local iteration caused by:
  - local link cost change
  - message from neighbor: its least cost path change from neighbor
- distributed:
  - each node notifies neighbors only when its least cost path to any destination changes
  - neighbors then notify their neighbors if necessary

Each node:

- wait for (change in local link cost of msg from neighbor)
- compute distance table
- if least cost path to any dest has changed, notify neighbors

Distance Vector: link cost changes

Link cost changes:
- node detects local link cost change
- updates distance table (line 15)
- if change in least cost path, notify neighbors (lines 23, 24)

"good news travels fast"
**Distance Vector: link cost changes**

- **Link cost changes:**
  - good news travels fast
  - bad news travels slow

![Diagram showing link cost changes](image)

**Distance Vector: poisoned reverse**

If Z routes through Y to get to X:
- Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- Does not work on larger loops

![Diagram showing poisoned reverse](image)

**Comparison of LS and DV algorithms**

**Message complexity**
- **LS:** with n nodes, with an average of l links/node, each node sends O(nl). Total messages O(n^2l)
- **DV:** exchange between neighbors only
  - convergence time varies
  - may be routing loops
  - count-to-infinity problem

**Robustness:** what happens if router malfunctions?
- **LS:**
  - node can advertise incorrect link cost
  - each node computes only its own table
- **DV:**
  - DV node can advertise incorrect path cost
  - each node’s table used by others
  - error propagate thru network

**Hierarchical Routing**

Our routing study thus far -idealization
- all routers identical
- network “flat”
  ... not true in practice

**Scale:** with 50 million destinations:
- can’t store all dest’s in routing tables!
- routing table exchange would swamp links!

**Administrative autonomy**
- internet = network of networks
- each network admin may want to control routing in its own network

**Why different Intra- and Inter-AS routing?**

**Policy:**
- Inter-AS: admin wants control over how its traffic is routed and who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

**Scale:**
- hierarchical routing saves table size, reduced update traffic

**Performance:**
- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance
Intra-AS and Inter-AS routing

- Gateways: perform inter-AS routing amongst themselves, perform intra-AS routers with other routers in their AS

Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  - Stub AS: small corporation
  - Multihomed AS: large corporation (no transit)
  - Transit AS: provider

- Two-level routing:
  - Intra-AS: administrator is responsible for choice
  - Inter-AS: unique standard

Internet AS Hierarchy

- Intra-AS border (exterior gateway) routers
- Inter-AS interior (gateway) routers

Intra-AS Routing

- Also known as Interior Gateway Protocols (IGP)
- Most common IGPs:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary.)

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)  
  - Can you guess why?
- Distance vectors: exchanged every 30 sec via Response Message (also called advertisement)
- Each advertisement: routes for up to 25 destination nets
RIP (Routing Information Protocol)

<table>
<thead>
<tr>
<th>Destination Network</th>
<th>Next Router</th>
<th>Num. of hops to dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td>7</td>
</tr>
<tr>
<td>x</td>
<td>--</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Routing table in D

RIP: Link Failure and Recovery

If no advertisement heard after 180 sec → neighbor/link declared dead
- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly propagates to entire net
- poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

RIP Table processing

- RIP routing tables managed by application-level process called routed (daemon)
- advertisements sent in UDP packets, periodically repeated

RIP Table example (continued)

Router: giroflee.eurocom.fr

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Flags</th>
<th>Ref</th>
<th>Use</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>U</td>
<td>0</td>
<td>26492</td>
<td>lo0</td>
</tr>
<tr>
<td>192.168.2.1</td>
<td>192.168.2.5</td>
<td>U</td>
<td>2</td>
<td>53</td>
<td>fa0</td>
</tr>
<tr>
<td>192.168.4.4</td>
<td>193.55.114.4</td>
<td>U</td>
<td>3</td>
<td>58503</td>
<td>le0</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td>193.55.114.4</td>
<td>U</td>
<td>3</td>
<td>0</td>
<td>le0</td>
</tr>
<tr>
<td>default</td>
<td>193.55.114.129</td>
<td>G</td>
<td>143454</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OSPF (Open Shortest Path First)

- “open”: publicly available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra’s algorithm
- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)

OSPF “advanced” features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion); TCP connections used
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different TOS (e.g., satellite link cost set “low” for best effort; high for real time)
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- Hierarchical OSPF in large domains.
Hierarchical OSPF

- **Two-level hierarchy**: local area, backbone.
  - Link-state advertisements only in area
  - Each node has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers**: "summarize" distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers**: run OSPF routing limited to backbone.
- **Boundary routers**: connect to other ASs.

IGRP (Interior Gateway Routing Protocol)

- Cisco proprietary; successor of RIP (mid 80s)
- Distance Vector, like RIP
  - Hold time
  - Split Horizon
  - Poison Reverse
- Several cost metrics (delay, bandwidth, reliability, load etc)
- Uses TCP to exchange routing updates
- EIGRP (Garcia-Luna): Loop-free routing via Distributed Updating Alg. (DUAL) based on diffused computation
  - Uses a mix of link-state and distance vector

Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): the de facto standard
  - Path Vector protocol:
    - Similar to Distance Vector protocol
    - Each Border Gateway broadcast to neighbors (peers) entire path (i.e., sequence of ASs) to destination
    - E.g., Gateway X may send its path to dest. Z:
      \[ \text{Path} (X, Z) = X, Y_1, Y_2, Y_3, \ldots, Z \]

Suppose: gateway X send its path to peer gateway W

- W may or may not select path offered by X
  - Cost, policy (don’t route via competitors AS), loop prevention reasons.
- If W selects path advertised by X, then:
  \[ \text{Path} (W, Z) = w, \text{Path} (X, Z) \]
- Note: X can control incoming traffic by controlling its route advertisements to peers:
  - E.g., don’t want to route traffic to Z -> don’t advertise any routes to Z

Inter-AS routing

- Note: X can control incoming traffic by controlling its route advertisements to peers:
  - E.g., don’t want to route traffic to Z -> don’t advertise any routes to Z

Internet inter-AS routing: BGP

Path (X, Z) = X, Y1, Y2, Y3, ..., Z
Internet inter-AS routing: BGP

- BGP messages exchanged using TCP.
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

Other Routing Techniques

- Hot-Potato Routing a.k.a Deflection Routing
  - Use the first available link irrespective of whether it leads to the destination or not.
- Cut Through routing
  - Non-store and forward: Routes before entire packet is received at the router.
  - Outgoing link is reserved. What happens if a fast link succeeds a slow link?

Reading

- Required
  - End-To-End Routing Behavior in the Internet, V. Paxson, SIGCOMM 1996.
    - Due: 3/26/01
  - Persistent Route Oscillations in Inter-Domain Routing, K. Varadhan, R. Govindan, D. Estrin,
    - Due: 3/28/01

- Recommended
  - http://netresearch.ics.uci.edu/agentos/related/routing:
    Contains information on CISCO Routing Protocols