Internet Protocol: Routing Algorithms

Srinidhi Varadarajan

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
- 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. 4. if $v$ adjacent to $A$
5. then $D(v) = c(A,v)$
6. else $D(v) = \infty$
7. Loop
8. 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. *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, discussion

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

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

Link State: Reliable Flooding

- Link State routers exchange information using Link State Packets (LSP).
- LSP contains:
  - id of the node that created the LSP
  - cost of the link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet
- Reliable flooding:
  - store most recent LSP from each node
  - forward LSP to all nodes but one that sent it
  - generate new LSP periodically
    - increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP
    - discard when TTL=0

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!

**Distance Table data structure**
- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node

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

Distance table gives routing table

<table>
<thead>
<tr>
<th>Distance table</th>
<th>Routing table</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost to destination via</td>
<td>cost to destination via</td>
</tr>
<tr>
<td>( D^E() )</td>
<td>( D^E() )</td>
</tr>
<tr>
<td>A,1</td>
<td>A,14,5</td>
</tr>
<tr>
<td>B,7,8,5</td>
<td>B,5</td>
</tr>
<tr>
<td>C,6,9,4</td>
<td>C,4</td>
</tr>
<tr>
<td>D,4,11,2</td>
<td>D,4</td>
</tr>
</tbody>
</table>
**Distance Vector: link cost changes**

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

"good news travels fast"

algorithm terminates

**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

algorithm terminates

**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

**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

**gateway routers**
- special routers in AS
- run intra-AS routing protocol with all other routers in AS
- also responsible for routing to destinations outside AS
  - run inter-AS routing protocol with other gateway routers
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

- 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

Routing in the Internet

- 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 Table example (continued)

**Routing table in D**

<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>
</tbody>
</table>

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)

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 (eg, 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 nodes 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 Algorithm. (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:
    \[
    \text{Path (X,Z)} = X,Y_1,Y_2,Y_3,\ldots,Z
    \]
**Internet inter-AS routing: BGP**

**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:
  - Path \((W, Z) = w, \text{Path (X, Z)}\)
- Note: X can control incoming traffic by controlling it route advertisements to peers:
  - e.g., don’t want to route traffic to Z -> don’t advertise any routes to Z

**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 links succeeds a slow link?

**Reading**
- **Recommended**
  - *Persistent Route Oscillations in Inter-Domain Routing*, K. Varadhan, R. Govindan, D. Estrin,