Network Routing

Packet Routing, Routing Algorithms, Routers, Router Architecture
Routing

Routing protocol

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
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

Static or dynamic?

Static:
- routes change slowly over time

Dynamic:
- routes change more quickly
  - periodic update
  - in response to link cost changes
What is the problem?

How should packets from A reach B?
Approaches
- Flooding
- Source routing
- Forwarding table
- Spanning tree

Metrics

Shortest path spanning trees

Other types of routing
- Multipath
- Multicast
Flooding

- Inefficient link usage.
- Packets can loop forever.
- Used when we don’t know (or can’t trust) the topology.
Source Routing

- “End-to-end” solution – no support needed from network.
- Packet carries a variable (and maybe long) list of addresses.
- End host must know the topology and choose the route.
- Used when end user wants to control the route.
Forwarding Table

<table>
<thead>
<tr>
<th>Address</th>
<th>Next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>S₂</td>
</tr>
<tr>
<td>C</td>
<td>S₃</td>
</tr>
<tr>
<td>D</td>
<td>S₃</td>
</tr>
</tbody>
</table>

- An optimization: Network handles hop-by-hop routing.
- Requires population of forwarding tables.
- Per-destination state, not (necessarily) per-flow state.
Spanning Tree

**Spanning tree**
- Spanning: It reaches all leaves
- Tree: It has no loops
Example of a no-loop spanning tree
Spanning Trees in Practice

1. Periodically, all switches broadcast a “Bridge Protocol Data Unit” (BPDU) (ID of sender, ID of root, distance from sender to root).

2. Initially, every switch claims to be Root: sets distance field to 0.
3. Every switch broadcasts until it hears a “better” message:
   - A root with a smaller ID
   - A root with equal ID, but with shorter distance
   - Ties broken by smaller ID of sender.
4. If a switch hears a better message, retransmit message (add 1 to distance).

Root port: The port on a switch that is closest to the Root.
Designated port: The port neighbors agree to use to reach the Root.
All other ports are blocked from forwarding (but still send/receive BPDUs).

Eventually:
- Only the root originates configuration messages (others retransmit them).
- Locally, switch only forwards on ports.
Routing Metrics

Choices
- Min distance
- Min hop-count
- Min delay
- Max throughput
- Least-loaded path
- Most reliable path
- Lowest cost path
- Most secure path
- ...

Diagram showing network with nodes A, B, C, D, R1, R2, R3, R4, R5, R6, R7, R8, X.
Simple Topology

Minimum cost spanning tree.
In this case, simple.
How about this network...!? 

The Opte Project
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
Hierarchical Routing

• aggregate routers into regions, “autonomous systems” (AS)
• routers in same AS run same routing protocol
  • “intra-AS” routing protocol
  • routers in different AS can run different intra-AS routing protocol
  • also responsible for routing to destinations outside AS
  • run inter-AS routing protocol with other 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
Intra-AS and Inter-AS routing

- Intra-AS routing within AS A
- Intra-AS routing within AS B
- Inter-AS routing between A and B

- Host h1
- Host h2

We’ll examine specific inter-AS and intra-AS Internet routing protocols shortly
Multipath
Multicast
Reverse Path Broadcast (RPB) + Pruning

1. Packets delivered loop-free to every end host.
2. Routers with no interested hosts send prune messages towards source.
3. Resulting tree is the minimum cost spanning tree from source to the set of interested hosts.
One Tree vs. Multiple Trees

[Diagram showing network topology with nodes R1, R2, R3, R4, R5, R6, R7, R8, A, B, C, D, E, X connected by arrows and lines.]
Multicast in Practice

Multicast used less than originally expected
- Most communication is individualized (e.g. time shifting)
- Early implementations were inefficient
- Today, used for some IP TV and fast dissemination
- Some application-layer multicast routing used

Some interesting questions
- How to make multicast reliable?
- How to implement flow-control?
- How to support different rates for different end users?
- How to secure a multicast conversation?
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
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,D</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>2</td>
<td>ADE</td>
<td>2,A</td>
<td>3,E</td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ADEB</td>
<td></td>
<td>3,E</td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ADEBC</td>
<td></td>
<td></td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ADEBCF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Graph representation](image)
Dijkstra’s algorithm, discussion

Algorithm complexity: n nodes
- each iteration: need to check all nodes, w, not in N
- \( n^2 \) comparisons: \( O(n^2) \)
- more efficient implementations possible: \( O(n \log n) \)

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

**distributed:**
- each node communicates *only* with directly-attached neighbors

**Distance Table data structure**
- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node $X$, for dest. $Y$ via neighbor $Z$:

$$D^{X}_{(Y,Z)} = \begin{cases} 
\text{distance from } X \text{ to } \ Y, \ via \ Z \text{ as next hop} \\
= c(X,Z) + \min_w \{D^Z_{(Y,w)}\}
\end{cases}$$
Distance Table: example

\[
D^E(C,D) = c(E,D) + \min_w \{D^D(C,w)\} \\
= 2+2 = 4
\]

\[
D^E(A,D) = c(E,D) + \min_w \{D^D(A,w)\} \\
= 2+3 = 5 \quad \text{loop!}
\]

\[
D^E(A,B) = c(E,B) + \min_w \{D^B(A,w)\} \\
= 8+6 = 14 \quad \text{loop!}
\]
Distance table gives routing table

<table>
<thead>
<tr>
<th>Destination</th>
<th>A</th>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<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>

Outgoing link to use, cost

<table>
<thead>
<tr>
<th>Destination</th>
<th>A</th>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>D,5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>D,4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>D,2</td>
<td></td>
<td></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)
- recompute distance table
- if least cost path to any dest has changed, notify neighbors
Comparison of LS and DV algorithms

Message complexity
- **LS**: with n nodes, E links, $O(nE)$ msgs sent each
- **DV**: exchange between neighbors only
  - convergence time varies

Speed of Convergence
- **LS**: $O(n^2)$ algorithm requires $O(nE)$ msgs
  - may have oscillations
- **DV**: 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
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 propr.)
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: route to up to 25 destination nets
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.
IGRP (Interior Gateway Routing Protocol)

- CISCO proprietary; successor of RIP (mid 80s)
- Distance Vector, like RIP
- several cost metrics (delay, bandwidth, reliability, load etc)
- uses TCP to exchange routing updates
- Loop-free routing via Distributed Updating Alg. (DUAL) based on diffused computation
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:
  \[ \text{Path (W,Z)} = w, \text{ Path (X,Z)} \]
- 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
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
Why different Intra- and Inter-AS routing?

Policy:
- Inter-AS: admin wants control over how its traffic routed, 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
Router Architecture Overview

Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *switching* datagrams from incoming to outgoing link
Input Port Queuing

- Fabric slower than input ports combined -> queueing may occur at input queues
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward
- Queueing delay and loss due to input buffer overflow!
Output Ports

- **Buffering** required when datagrams arrive from fabric faster than the transmission rate
- **Scheduling discipline** chooses among queued datagrams for transmission
Three types of switching fabrics

- Memory
  - Limited buffer

- Bus
  - Contention

- Crossbar
What about Routing in Wireless?

- Wireless transmission medium is not a wire
- Radiates over space
  - Signal weakens with distance: $r^2$ or faster
  - Intermediate links
- Uncontrolled medium
  - Signal strength changes over time
  - Interference from other transmitters
Closer look at Signal Strength

- Obstructions can further weaken signal
- Wireless signals can reflect
  - Multipath: can receive signal in multiple paths/reflections, with different delays (analogy: echoes in a canyon)
- There is no perfectly uniform antenna
- The world is continuously changing
Changing over time

Packet Reception Ratio

Received Signal Strength Indicator

Time in secs

PRR

RSSI
Interference
Example of a WiFi Channel
Summary

- Wireless networks are increasingly the last hop for personal communications
  - But generally don’t work as well
- Wireless behaves very differently from wired: many complex behaviors!
  - Signal weakens over distance
  - Signal affected by environment
  - Intermediate links
  - External interference
- Different behavior leads to different protocols and algorithms