Failures in routing
• Routing loops
• Counting to infinity
Unanswered questions
(about packet-switched networks)

How do nodes determine routes to every other node?
Nodes determine routes via either **link-state** or **distance-vector** routing

How do nodes route around link failures?

How do nodes **communicate reliably** given that the network is best-effort?
## Comparison of Routing Protocols
(no failures)

<table>
<thead>
<tr>
<th></th>
<th>Distance-vector</th>
<th>Link-state</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node X’s advertisement format</strong></td>
<td>list of all nodes X knows about and the costs to those nodes</td>
<td>list of all X’s neighbors and the link costs to those nodes</td>
</tr>
<tr>
<td><strong>Who receives X’s advertisement</strong></td>
<td>X’s neighbors</td>
<td>all nodes (via flooding)</td>
</tr>
<tr>
<td><strong>Integration</strong></td>
<td>Bellman-Ford</td>
<td>Dijkstra’s Algorithm</td>
</tr>
<tr>
<td><strong>Convergence time</strong></td>
<td>Proportional to the number of hops in the longest min-cost path</td>
<td>Proportional to flooding time + complexity of Dijkstra’s</td>
</tr>
</tbody>
</table>
## Comparison of Routing Protocols
(no failures)

<table>
<thead>
<tr>
<th></th>
<th>Distance-vector</th>
<th>Link-state</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Integration</strong></td>
<td>Bellman-Ford</td>
<td>Dijkstra’s Algorithm</td>
</tr>
<tr>
<td><strong>Convergence time</strong></td>
<td>Proportional to the number of hops in the longest min-cost path</td>
<td>Proportional to flooding time + complexity of Dijkstra’s</td>
</tr>
<tr>
<td><strong>Amount of data consumed by advertisements</strong></td>
<td>$O(L)$</td>
<td>$O(L^2)$</td>
</tr>
</tbody>
</table>

*better for large networks?*
Convergence

**route validity:** if node N’s routing table contains D, then there is a usable path in the network from N to D, and the routing table reflects a usable path.

**path visibility:** every router that has a usable path to a destination learns at least one valid route to that destination.
Eventual Convergence

Given:
- initial state at time 0
- time t after which no changes occur to the topology and no routing advertisements or HELLO packets are lost

If the routing protocol converges in some finite amount of time after t, we say the routing protocol has **eventually converged**.
goal: understand how link-state and distance-vector perform when links fail so that we can decide when to use which protocol
Routing Loops

C’s view

D’s view

lost advertisements cause incorrect network views and thus incorrect routes

A: C -> D
A: D -> C
Routing Loops

lost advertisements cause incorrect network views and thus incorrect routes
Routing Loops

lost advertisements cause incorrect network views and thus incorrect routes

C’s view

D’s view

A: D->C

dst: A
Routing Loops

Lost advertisements cause incorrect network views and thus incorrect routes.
Routing Loops

lost advertisements cause incorrect network views and thus incorrect routes

C’s view

D’s view

A: C -> D

B

C

A: D -> C

E

D

dst: A
Routing Loops

lost advertisements cause incorrect network views and thus incorrect routes
### Comparison of Routing Protocols

(failures)

<table>
<thead>
<tr>
<th></th>
<th>Distance-vector</th>
<th>Link-state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount of data</td>
<td>Small (O(L))</td>
<td>Large (O(L^2))</td>
</tr>
<tr>
<td>consumed by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>advertisements</td>
<td><strong>better for large networks?</strong></td>
<td></td>
</tr>
<tr>
<td>Convergence</td>
<td></td>
<td>Generally fast,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>routing loops are rare</td>
</tr>
</tbody>
</table>
A sends advertisements at $t=0, 10, 20, \ldots$; B sends advertisements at $t=5, 15, 25, \ldots$

$t=9$: $B \leftarrow C$ fails

$t=10$: $B$ receives the following advertisement from $A$:

$[(A, 0), (B, 1), (C, 2)]$

$t=15$: $A$ receives the following advertisement from $B$:

$[(A, 0), (B, 1), (C, 3)]$

$t=20$: $B$ receives the following advertisement from $B$:

$[(A, 0), (B, 1), (C, 4)]$

continues until both costs to $C$ are $\text{INFINITY}$
**problem:** distance-vector protocols can count to infinity, which increases the convergence time. can we solve the count-to-infinity problem?
Split-horizon

Don’t send advertisements about a route to the node providing the route

continues until all costs to C are INFINITY
## Comparison of Routing Protocols

### (failures)

<table>
<thead>
<tr>
<th></th>
<th>Distance-vector</th>
<th>Link-state</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Amount of data</strong></td>
<td>Small ((O(L)))</td>
<td>Large ((O(L^2)))</td>
</tr>
<tr>
<td><strong>consumed by</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>advertisements</strong></td>
<td>better for large networks?</td>
<td></td>
</tr>
<tr>
<td><strong>Convergence</strong></td>
<td>Can depend on value of INFINITY; the larger INFINITY is, the slower convergence is</td>
<td>Generally fast, routing loops are rare</td>
</tr>
<tr>
<td></td>
<td>so.. not good for large networks?</td>
<td></td>
</tr>
</tbody>
</table>
# Comparison of Routing Protocols  
(failures)

<table>
<thead>
<tr>
<th></th>
<th>Distance-vector</th>
<th>Link-state</th>
<th>Path-vector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Amount of data</strong></td>
<td>Small (O(L))</td>
<td>Large (O(L^2))</td>
<td>Fairly small in practice (but larger than DV)</td>
</tr>
<tr>
<td><strong>consumed by</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>advertisements</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Convergence</strong></td>
<td>Can depend on value of INFINITY</td>
<td>Generally fast, routing loops are rare</td>
<td>Not as fast as LS, but does not depend on INFINITY</td>
</tr>
<tr>
<td><strong>good for</strong></td>
<td>good for very small networks where we can make guarantees about (a lack of) routing loops</td>
<td>good for small (university-sized) networks where the overhead of the advertisements doesn’t overwhelm</td>
<td>good for large networks (the Internet!)</td>
</tr>
</tbody>
</table>
• **Distance-vector Routing**
  Low overhead, but slow convergence (count-to-infinity)

• **Link-state Routing**
  High overhead, but faster convergence (routing loops can happen, but are rare)

• **Path-vector Routing**
  An improvement on distance-vector routing that avoids counting to infinity

• **Which protocol to use depends on the environment, particularly on the size of the network**