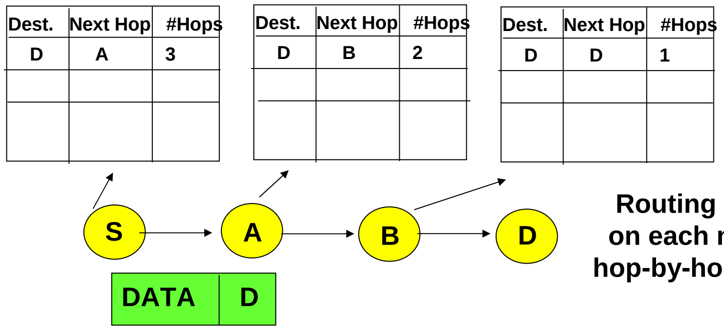
Hop-by-Hop Routing



Dest.	Next Hop	#Hops
D	A	3

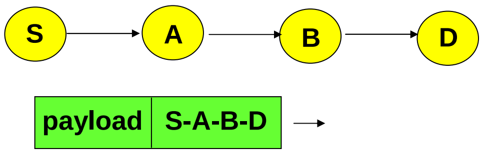
Dest.	Next Hop	#Hops
D	B	2

Dest.	Next Hop	#Hops
D	D	1

Routing Tables
on each node for
hop-by-hop routing

- Routing table on each node contains the next hop node and a cost metric for each destination.
- Data packet only has the destination address.

Source Routing



- In source routing, the data packet has the complete route (called source route) in the header.
- Typically, the source node builds the whole route
- The data packet routes itself.
- *Loose source routing:* Only a subset of nodes on the route included.

Static vs. Dynamic Routing

- **Static routing has fixed routes, set up by network administrators, for example.**
- **Dynamic routing is network state-dependent. Routes may change dynamically depending on the "state" of the network.**
- **State = link costs. Switch traffic from highly loaded links to less loaded links.**

Distributed, Dynamic Routing Protocols

- **Distributed because in a dynamic network, no single, centralized node "knows" the whole "state" of the network.**
- **Dynamic because routing must respond to "state" changes in the network for efficiency.**
- **Two class of protocols: Link State and Distance Vector.**

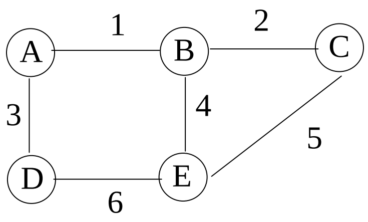
Link State Protocol

- **Each node "floods" the network with link state packets (LSP) describing the cost of its own (outgoing) links.**
  - Link cost metric = typically delay for traversing the link.
  - Every other node in the network gets the LSPs via the flooding mechanism.
- **Each node maintains a LSP database of all LSPs it received.**
  - Only the recent most LSP is maintained for a link.
  - The LSP database describes this node's view of the "state" of the network.

Flooding Mechanism

- **The originator generates LSPs periodically, or when some link costs changes significantly.**
  - The originator transmits LSP on all its interfaces.
- **Upon receiving an LSP, a node**
  - Inserts the LSP in its database if not already there, otherwise drops the LSP.
  - If not dropped, the LSP is forwarded on all interfaces except the one on which it was received.

Shortest Path Routing



LSP database on a node

A - B 1

A - D 3

B - E 4

…..

……

- **LSP database describes a node's view of the state of the network.**
- **Compute shortest path from this node to every other node using Dijkstra's shortest path algorithm.**

Link State Routing

- **Advantages:**
  - Each node can use its own routing "policy."
  - Reasonable convergence speed.
  - Has flavor of centralized routing. Loop-free, unless the network is very dynamic, when transient loops may form.
- **Disadvantage:**
  - High routing overhead as LSP packets flood the entire network.
  - Large LSP database size.

Sequence Numbers

- **LSP must have a sequence number.**
  - Otherwise, it is hard to tell whether an arriving LSP is newer than the one in the database.
  - Asynchronous flooding does not guarantee that LSPs will arrive in order in all nodes.
- **Sequence number uses a finite no. of bits**
  - Typically, 32 bits.
  - Can wrap around during the operational lifetime of the network. Thus, smaller does not always mean older.
  - Use heuristic: seq no. a is older than seq no. b, if $a < b$, and $|b-a| < N/2$, or $a > b$ and $|b-a| > N/2$, where seq. no.s are from 0 to $N-1$.

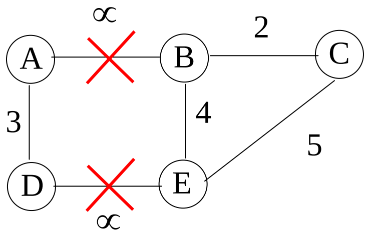
Aging

- **Idea:** **Remove very old LSP records from data base. Old records may present stale information.**
- **How:** **New LSP packets have an age (MAX_AGE).**
  - Age is decremented periodically.
  - When age becomes 0, the node floods the network with this LSP.
  - A zero-age LSP is always accepted in the database, resulting in actual removal of that LSP.
- **End result is that all nodes remove that LSP in a synchronized fashion.**
- **Thus, LSPs must be reissued at regular intervals, even without any change in link cost.**

Loss of Sequence Number on Router Failure/Reboot

- **New sequence number after reboot will be typically zero.**
  - Will be regarded as older, if the last seq no. before failure < N/2.
- **Solution: Use a unique sequence no. to be used only after reboot.**
  - Any neighbor receiving LSP packets with this seq. no. updates the rebooted node with the seq. no. used before failure.
  - The rebooted node now uses one plus this sequence no.
  - Read about "Lollipop seq. no." in Keshav's book.

Recover from network partition



- LSP databases are updated independently on the nodes in each partition.
- On a join, need to merge the databases to make them consistent.

- For example, assume the E-C link breaks after partition. D does not know about it.
- D may still try to route to C via E after D-E link comes up.

Recovering from Partition

- **Nodes on either side of a newly restored link cooperate to merge the respective LSP databases.**
  - Keep only the “freshest” information.
  - Seq no.s in the LSP database records are useful to determine the freshest.
  - If there are stale LSP records (that are now updated), such stale records may be present elsewhere in the network.
  - Originators of such LSP records are requested for new LSP updates to be flooded.

Choice of Routing Metric

- **Static metric:**
  - Link is up –> cost is one.
  - Link is down –> cost is infinity.
  - Some “popular” links may get really congested.
- **Dynamic metric:**
  - Original ARPAnet metric: use the average queue length at the interface queue over a small time interval.
  - Too much fluctuation in metric -> rapid routing fluctuations/oscillations.
  - Large queue length -> high cost -> routes avoid this link -> small queue length -> low cost -> routes prefer this link -> large queue length -> high cost ….

Modified ARPAnet Routing Metric

- **Idea: Use link delay = queuing delay at interface queue + transmission time + propagation time.**
  - Queuing delay dominates at high load.
  - Transmission and propagation times dominate at low load.
    - Transmission time dominates for large packets.
    - Propagation time dominates for small packets.
- **Provides some balance. Less fluctuation.**

Modified ARPAnet Routing Metric (more ideas)

- **Use exponential moving average, rather than just an average over a measurement interval.**
  - Factor in the averages in a few previous intervals, albeit with progressively lower weights for earlier intervals.
- **Reduce dynamic range by providing some artificial limits.**
  - Also do not allow too fast change in link costs.
- **The actual metric uses a “well-behaved” function of link utilization and type of link (bw, delay properties).**

Multiple Routing Metrics Possible in Link State Protocol

- **An LSP advertisement can carry multiple definitions of link costs.**
- **OSPF (Open Shortest Path Protocol) example:**
  - Throughput metric
  - Delay metric
  - Reliability metric
  - Cost ($$) metric
- **Routers (nodes) can use any one metric to chose a route.**
  - Use of different metrics on different routers possible, but must be careful about looping.

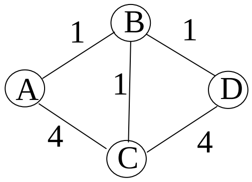
Type of Service (TOS) Routing in IP

- **IP packets carry a 5-bit TOS field denoting the type of routing service preferred**
  - E.g., minimize delay, throughput, $$ cost etc.
- **Related to Quality of Service (QoS).**
- **However, all routers may not be TOS capable.**

Distance Vector Routing

- **Distance Vector: Shortest distance (in hops, e.g.) for every destination.**
  - Routing table minus next-hop information.
- **Propagate own distance vector to neighbors only.**
- **Each neighbor determines whether a new distance vector received on a link imply any change in any component of its own distance vector by factoring in the link costs.**

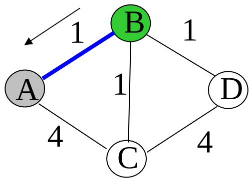
DV Routing: Example



Destinations

Node	A	B	C	D
A	0	1	4	inf
B	1	0	1	1
C	4	1	0	2
D	inf	1	2	0

B sends own DV to A

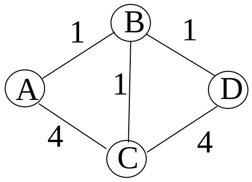


Destinations

		A	B	C	D
Node	A	0	1	4	inf
	B	1	0	1	1
	C	4	1	0	2
	D	inf	1	2	0

Add cost of link A-B to get distance via B.

DV Routing: Example



Destinations

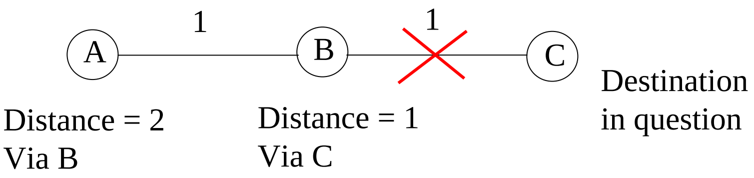
		A	B	C	D
Node	A	0	1	2	2
	B	1	0	1	1
	C	4	1	0	2
	D	inf	1	2	0

Note that now routing table entries for C and D will point to B as next hop.

Distance Vector: Properties

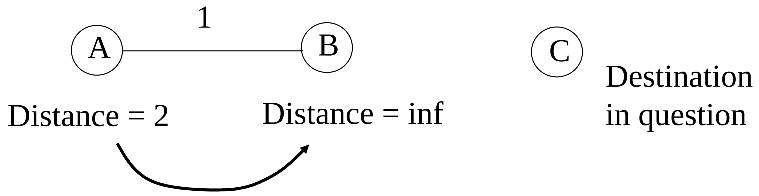
- **Distributed variation of Bellman-Ford algorithm.**
- **Converges even if the nodes asynchronously updates their distance vectors.**
- **Asynchronous updates and/or lost updates may cause temporary routing loops.**
- **Updates are typically periodic, possibly augmented by triggered updates on link/node failures.**
  - Example, RIP (Routing Information Protocol) on the Internet (30 sec period).

Counting to Infinity Problem



- **B-C link goes down (distance 1 -> inf).**
- **B may now switch its route through A, as A offers a lower cost route via its own distance vector.**

Counting to Infinity Problem


1

A

B

C

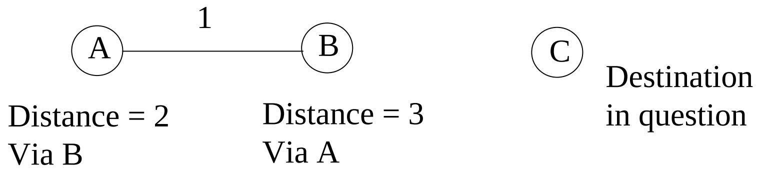
Destination in question

Distance = 2

Distance = inf

- **B-C link goes down (distance 1 -> inf).**
- **B may now switch its route through A, as A offers a lower cost route via its own distance vector.**

Counting to Infinity Problem


1

A

B

C

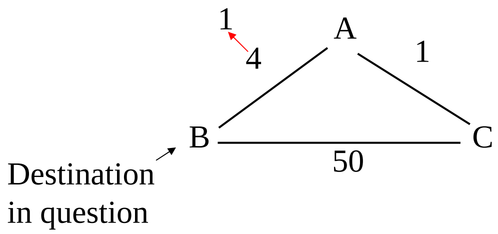
Destination in question

Distance = 2
Via B

Distance = 3
Via A

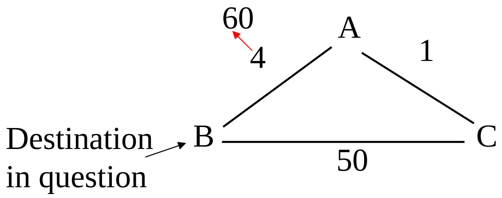
- **A will now refine its estimate to 4 after getting update from B.**
- **This will go on until the distance reaches a large enough value that is deemed as infinity.**
- **Takes too long to converge. Temporary looping.**

Note 1: Good news travels fast



- **A advertises lower distance to B.**
- **C updates its own distance to B (via A).**

Note 2: Bad news travels slow



- A should advertise higher distance to B.
- Before it does, A hears C advertising smaller distance (=5) to B. So it changes its route via C. This goes on.
- Solution: Get A to recognize that the route C advertises is actually via itself.
- How? Need to know whether A is on route from C->B.
  - Possible. Need to include/maintain additional information in DV.

Popular Solutions to Counting to Infinity

- **Make infinity small. Can’t take too long to converge.**
  - RIP uses #hops as distance metric and a value of 16 as infinity. Can’t recognize more than 16 hops in a network.
- **Split horizon based solutions**
  - Don’t send DV update to a neighbor for a destination, where that neighbor is the next hop for that destination.
  - **Poisoned reverse:** Send such DV updates but with infinite distance metric. Used in RIP.
- **The above split horizon based solutions can’t prevent looping involving more than 2 nodes.**
  - Try to construct examples.
  - Allow for lost update messages, if that makes examples easier.

Distance Vector (RIP) vs. Link State (OSPF)

- **Speed of Convergence:**
  - Counting to Infinity problem in RIP. RIP has only incomplete solutions.
  - Route oscillations in OSPF. Needed routing metric stabilization.
- **Routing Overhead:**
  - Network wide flood in OSPF for each link cost change.
  - Broadcast only to neighbors for each link cost change. Neighbors will broadcast to their own neighbors only of change in DV.

Scalability

- **Routing protocols like OSPF (link-state) and RIP (distance vector) typically routes between networks (i.e., LANs).**
- **There are a large number of networks on the Internet (~ millions).**
- **None of these protocols are scalable to that extent.**
  - LS (DV) updates (may) have to propagate throughout the entire network. Too much overhead. Too long to converge.
  - LS database or DV message size will be too large.

Hierarchical Routing

- **Typical solution to address scalability is to introduce hierarchy.**
  - Let protocols work independently in different tiers of the hierarchy.
- **Routers are organized into autonomous systems (AS).**
  - Each AS typically belongs to a single administrative domain (e.g., university campus).
  - Each AS runs its own DV or LS routing protocol.
- **Gateway router**
  - Router that connects two ASs, or an AS to a backbone router.

Intra- and Inter-AS Routing

- **Routing within an AS and routing between multiple ASs.**
  - Note sometimes an AS is called a domain.
  - So, intra- and inter-domain routing.
  - Inter-domain routing coming later.
- **Stub and transit AS**
  - Stub: AS does not carry traffic between other ASs.
  - Transit: AS does carry traffic between other ASs.

Hosts, Routers and Networks

- **Difference between host and router**
  - Host: connected to a single network. One interface.
  - Router: connected to multiple networks. More than one interface. Capable of directing traffic from one interface to the another.
- **Network here usually means just a LAN**
  - An interconnection of hosts that do not need a router to communicate (e.g., Ethernet).
  - Network may also mean an interconnection of LANs using routers. (Inclusive definition).