Data Communication & Computer Networks

9. Routing Protocols

Routing protocols

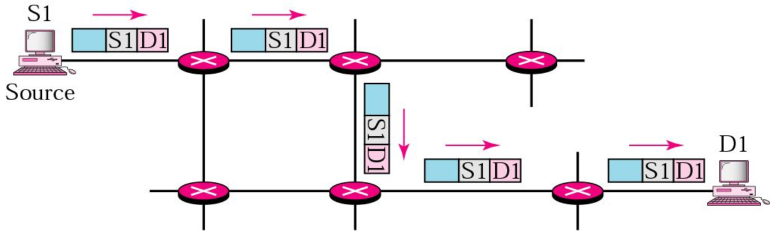
- An internet needs dynamic routing tables
- Routing protocols have been created in response to this demand
- A routing protocols is a combination of rules and procedures that let routers in the internet inform each other of changes
- They also include procedures for combining received routing information

Routing protocols

- routers receive and forward packets
- make decisions based on knowledge of topology and traffic/delay conditions
- use dynamic routing algorithm
- distinguish between:
  - routing information - about topology & delays
  - routing algorithm - that makes routing decisions based on information

Unicast communication

- 1 source and 1 destination
  - One-to-one relationship btw src and dest
- Both SA and DA in IP datagram are unicast addresses



Unicast routing

- In unicast routing, the router forwards the received packet through only one of its interfaces.
  - The interface is selected according to the optimum path defined in routing table
- Router may discard the packet if it can’t find the destination in its routing table

Metric

- A metric is a cost assigned for passing through a network
- Router chooses the route with smallest metric
- Metric assignment depends on the routing protocol
  - RIP: cost of all networks is same (1 hop)
  - OSPF: Admin can assign costs

Autonomous Systems (AS)

- AS is a group of routers and networks managed by single organization

- It consists of a group of routers exchanging information via a common routing protocol

Why need the concept of AS?

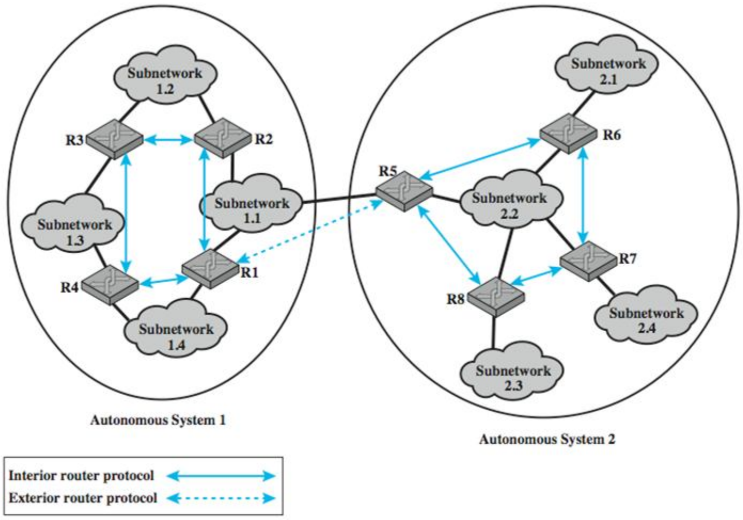
- Routing algorithms are not efficient enough to deal with the size of the entire Internet
- Different organizations may want different internal routing policies
- Allow organizations to hide their internal network configurations from outside
- Allow organizations to choose how to route across multiple organizations (BGP)
- Basically, easier to compute routes, more flexibility, more autonomy/independence

Interior & Exterior Routing

- interior router protocol (IRP)
  - passes routing information between routers within AS
  - can be tailored to specific applications
  - needs detailed model of network to function
- may have more than one AS in internet
  - routing algorithms & tables may differ between them
- routers need info on networks outside own AS
- use an exterior router protocol (ERP) for this
  - supports summary information on AS reachability

RQ 9

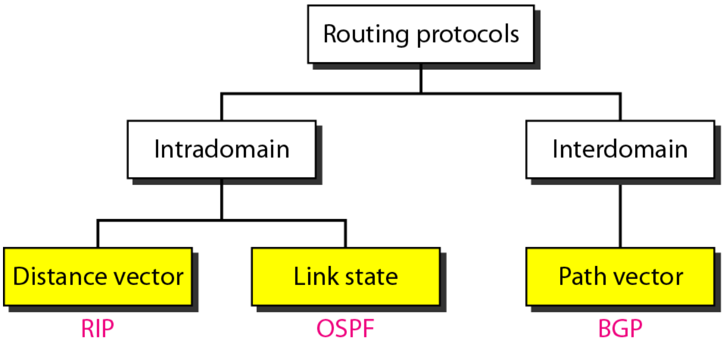
Application of IRP and ERP



10

Popular routing protocols

- Interior routing = Intra-domain routing
- Exterior routing = Inter-domain routing



Distance vector routing

- Each router periodically shares it knowledge with its neighbours
  - Shares knowledge about entire AS
  - Shares only with neighbours
  - Shares at regular intervals

Distance vector routing

- Examples of Distance Vector routing protocols:
  - Routing Information Protocol (RIP)
  - Interior Gateway Routing Protocol (IGRP)
  - Enhanced Interior Gateway Routing Protocol (EIGRP)

Distance vector routing

- Distance Vector Technology - the Meaning of Distance Vector
  - A router using distance vector routing protocols knows 2 things:
    - Distance to final destination
    - Vector, or direction, traffic should be directed

Distance vector routing table

- One entry for each dest. network known
- Other info = subnet mask, time etc.

Destination	Hop Count	Next Router	Other information
163.5.0.0	7	172.6.23.4	
197.5.13.0	5	176.3.6.17	
189.45.0.0	4	200.5.1.6	
115.0.0.0	6	131.4.7.19	

RIP

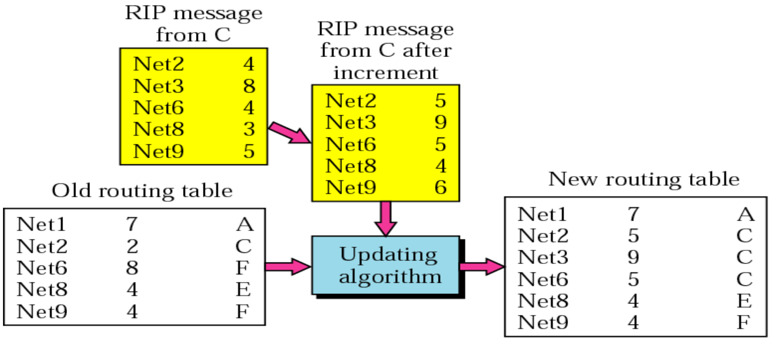
- Routing Information Protocol
- Interior routing protocol
- Uses distance vector routing
  - Uses Bellman-Ford algorithm for calculating the routing table

RIP updating algorithm

Receive: a response RIP message

1. Add one hop to the hop count for each advertised destination.
2. Repeat the following steps for each advertised destination:
   1. If (destination not in the routing table)
      Add the advertised information to the table.
   2. Else
      1. If (next-hop field is the same)
         Replace entry in the table with the advertised one.
      2. Else
         1. If (advertised hop count smaller than one in the table)
            Replace entry in the routing table.
3. Return.

Updating a routing table

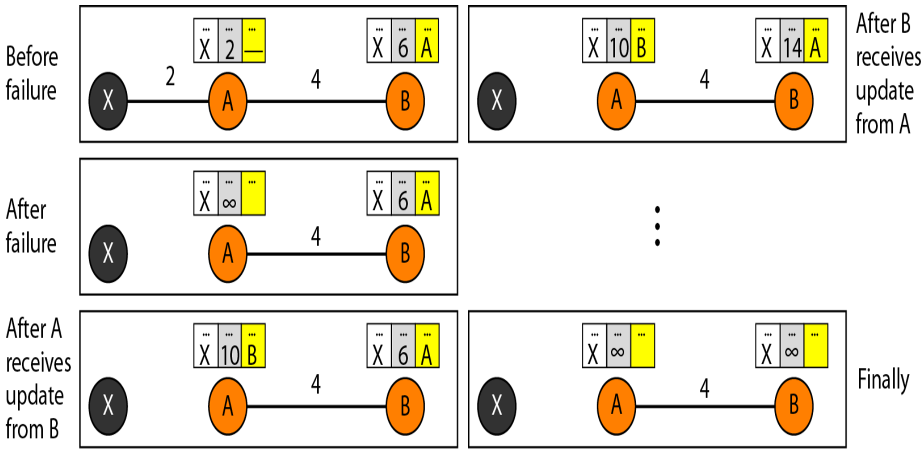


Net1: No news, do not change
Net2: Same next hop, replace
Net3: A new router, add
Net6: Different next hop, new hop count smaller, replace
Net8: Different next hop, new hop count the same, do not change
Net9: Different next hop, new hop count larger, do not change

When to share routing tables?

- Periodic update
  - A node sends its routing table, normally every 30s. (depends on protocol)
- Triggered update
  - A node sends its routing table to its neighbors anytime there is a change in its routing table.

Problem: Routing Loop
Two-node instability



Solutions

1. Defining Infinity
2. Split Horizon
3. Split Horizon and Poison Reverse

RQ 21

1. Defining Infinity

- Redefine infinity to a smaller number
  - system will become stable in fewer updates
  - Most distance-vector protocols define the distance between each node to be 1 and define 16 as infinity.
- Smaller infinity value limits he size of the network as well

RQ 22

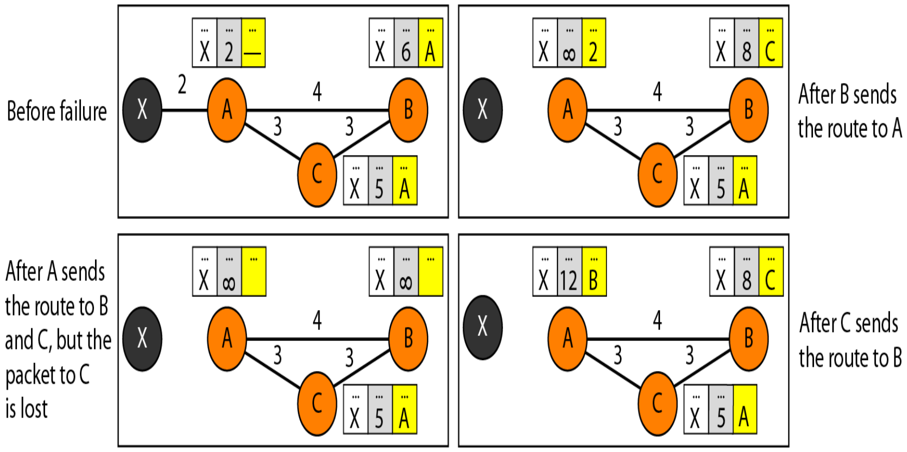
2. Split Horizon

- each node sends only part of its table through each interface
- routing information is not sent back in the direction from which it was received
- If B has learnt a piece of information from A, it does not need to advertise this information to A (A already knows)

3. Split Horizon with Poison Reverse

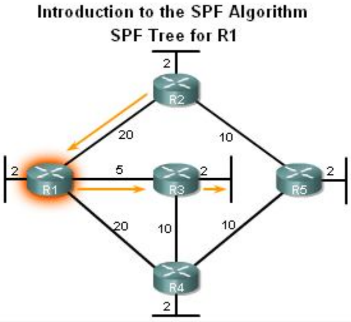
- E.g. *X* goes down
- *A* initiates route poisoning by advertising *X* to be unreachable
- *B* receives route poisoning from router *A*, it sends an update, called a *poison reverse,* back to router *A*
- Poison reverse with split horizon create a much more resilient and dependable distance-vector network

Problem: Routing Loop
Three-node instability



Link-State Routing

- The shortest path to a destination is not necessarily the path with the least number of hops



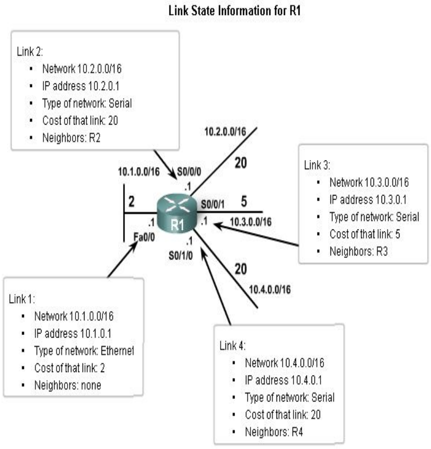
Destination	Shortest Path	Cost
R2 LAN	R1 to R2	22
R3 LAN	R1 to R3	7
R4 LAN	R1 to R3 to R4	17
R5 LAN	R1 to R3 to R4 to R5	27

Link-State Routing Process

- How routers using Link State Routing Protocols reach convergence
  - Each router learns about its own directly connected networks
  - Link state routers exchange hello packet to “meet” other directly connected link state routers
  - Each router builds its own Link State Packet (LSP) which includes information about neighbors such as neighbor ID, link type, & bandwidth
  - After the LSP is created the router floods it to all neighbors who then store the information and then forward it until all routers have the same information
  - Once all the routers have received all the LSPs, the routers then construct a topological map of the network which is used to determine the best routes to a destination

Link-State Routing

- Directly Connected Networks
- Link
  - This is an interface on a router
- Link state
  - This is the information about the state of the links

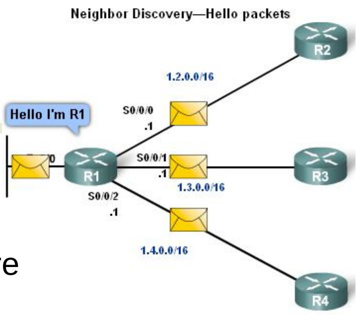


Link-State Routing

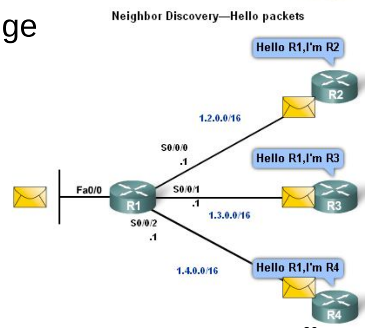
- Sending Hello Packets to Neighbors
  - Link state routing protocols use a hello protocol
  - Purpose of a hello protocol:
    - To discover neighbors (that use the same link state routing protocol) on its link

RQ 29

Link-State Routing



- Sending Hello Packets to Neighbors
  - Connected interfaces that are using the same link state routing protocols will exchange hello packets
  - Once routers learn it has neighbors they form an adjacency
    - 2 adjacent neighbors will exchange hello packets



RQ 30

Link-State Routing

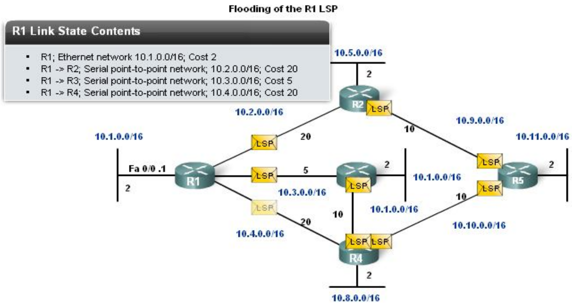
- Building the Link State Packet
  - Each router builds its own Link State Packet (LSP)
  - Contents of LSP:
    - State of each directly connected link
    - Includes information about neighbors such as neighbor ID, link type, & bandwidth

Link-State Routing

- Flooding LSPs to Neighbors
  - Once LSPs are created they are forwarded out to neighbors
  - After receiving the LSP the neighbor continues to forward it throughout routing area

Link-State Routing

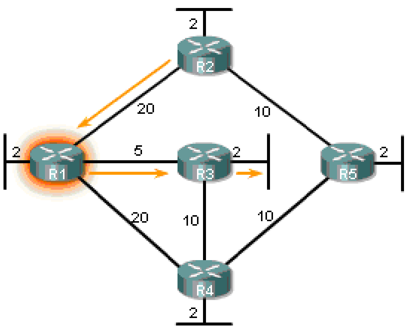
- LSPs are sent out under the following conditions:
  - Initial router start up or routing process
  - When there is a change in topology



Link-State Routing

- Constructing a link state data base
  - Routers use a database to construct a topology map of the network

Link-State Routing



R1 Link-State Database

R1s Link-State DatabaseLSPs from R2:
- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

LSPs from R3:
- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

LSPs from R4:
- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

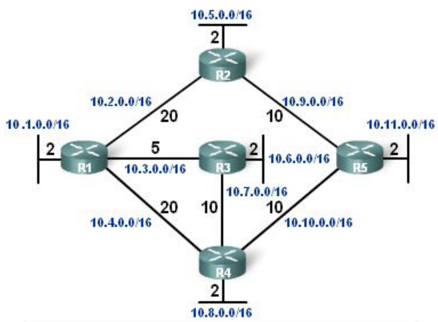
LSPs from R5:
- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

R1 Link-states:
- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20

Link-State Routing

- Determining the shortest path
  - The shortest path to a destination determined by adding the costs & finding the lowest cost

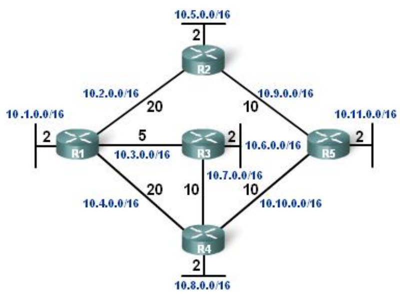
SPF Tree for R1



Destination	Shortest Path	Cost
R2 LAN	R1 to R2	22
R3 LAN	R1 to R3	7
R4 LAN	R1 to R3 to R4	17
R5 LAN	R1 to R3 to R4 to R5	27

Link-State Routing

- Once the SPF algorithm has determined the shortest path routes, these routes are placed in the routing table



R1 Routing Table

SPF Information

- Network 10.5.0.0/16 via R2 serial 0/0/0 at a cost of 22
- Network 10.6.0.0/16 via R3 serial 0/0/1 at a cost of 7
- Network 10.7.0.0/16 via R3 serial 0/0/1 at a cost of 15
- Network 10.8.0.0/16 via R3 serial 0/0/1 at a cost of 17
- Network 10.9.0.0/16 via R2 serial 0/0/0 at a cost of 30
- Network 10.10.0.0/16 via R3 serial 0/0/1 at a cost of 25
- Network 10.11.0.0/16 via R3 serial 0/0/1 at a cost of 27

R1 Routing Table

Directly Connected Networks
- 10.1.0.0/16 Directly Connected Network
- 10.2.0.0/16 Directly Connected Network
- 10.3.0.0/16 Directly Connected Network
- 10.4.0.0/16 Directly Connected Network

Remote Networks
- 10.5.0.0/16 via R2 serial 0/0/0, cost = 22
- 10.6.0.0/16 via R3 serial 0/0/1, cost = 7
- 10.7.0.0/16 via R3 serial 0/0/1, cost = 15
- 10.8.0.0/16 via R3 serial 0/0/1, cost = 17
- 10.9.0.0/16 via R2 serial 0/0/0, cost = 30
- 10.10.0.0/16 via R3 serial 0/0/1, cost = 25
- 10.11.0.0/16 via R3 serial 0/0/1, cost = 27

A Link State Routing Algorithm

Dijkstra's algorithm

- Net topology, link costs known to all nodes
  - Accomplished via "link state flooding"
  - All nodes have same info
- Compute least cost paths from one node ('source") to all other nodes
- Repeat for all sources

Notations

- 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 node *v*
- p(v): predecessor node along path from source to *v*, that is next to *v*
- P(v): path from source to *v*
- T: set of nodes whose least cost path definitively known

Dijkstra's Algorithm

- finds shortest paths from given source node *S* to all other nodes
- by developing paths in order of increasing path length
- algorithm runs in stages (next slide)
  - each time adding node with next shortest path
- algorithm terminates when all nodes processed by algorithm (in set T)

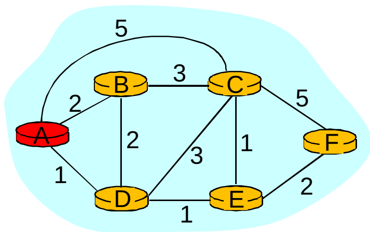
Dijsktra's Algorithm

```
Initialization:
  T = {S}; // S = source
  for all nodes v
    if v adjacent to S
      then D(v) = c(S,v);
      else D(v) = ∞;

 Loop
    find w not in T such that D(w) is a minimum;
    add w to T;
    update D(v) for all v adjacent to w and not in T:
      D(v) = min( D(v), D(w) + c(w,v) );
      // new cost to v is either old cost to v or known
      // shortest path cost to w plus cost from w to v
 until all nodes in T;
```

Example: Dijkstra’s Algorithm

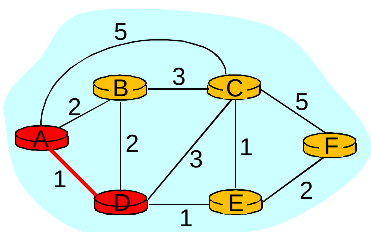
Step	Start T	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1						
2						
3						
4						
5						



Initialization:
T = {A};
for all nodes *v*
  if *v* adjacent to *A*
    then D(v) = c(A,v);
    else D(v) = ∞;
…

Example: Dijkstra’s Algorithm

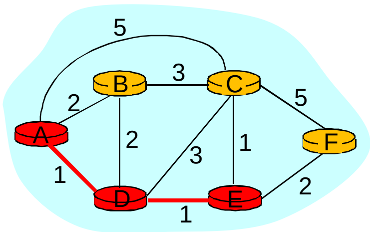
Step	Start T	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2						
3						
4						
5						



…
Loop
  find w not in T s.t. D(w) is a minimum;
  add w to T;
  update D(v) for all v adjacent to w and not in T:
    D(v) = min(D(v), D(w) + c(w,v));
until all nodes in T;

Example: Dijkstra’s Algorithm

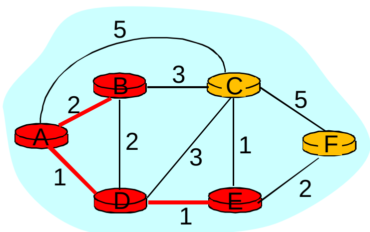
Step	Start T	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3						
4						
5						




```
…
Loop
  find w not in T s.t. D(w) is a minimum;
  add w to T;
 update D(v) for all v adjacent
  to w and not in T:
    D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in T;
```


Example: Dijkstra’s Algorithm

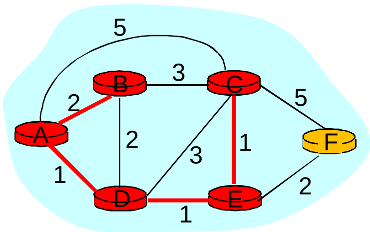
Step	Start T	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3	ADEB					
4						
5						




```
…
Loop
  find w not in T s.t. D(w) is a minimum;
  add w to T;
 update D(v) for all v adjacent
  to w and not in T:
    D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in T;
```

Example: Dijkstra's Algorithm

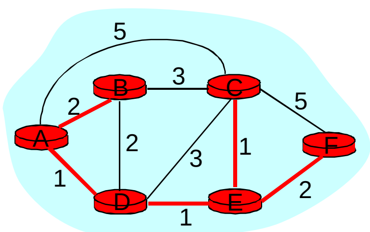
Step	Start T	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3	ADEB					
4	ADEBC					
5						




```
…
Loop
  find w not in T s.t. D(w) is a minimum;
  add w to T;
  update D(v) for all v adjacent
   to w and not in T:
     D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in T;
```

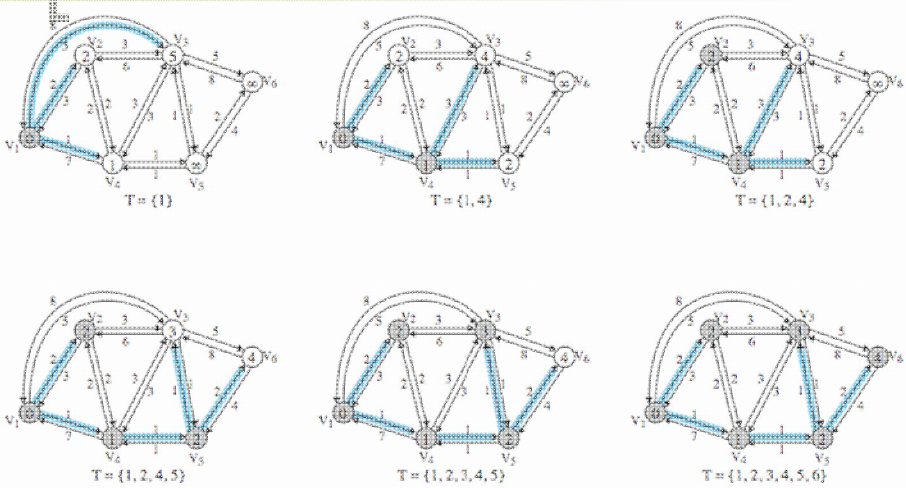

Example: Dijkstra's Algorithm

Step	Start T	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3	ADEB					
4	ADEBC					
5	ADEBCF					




```
…
Loop
  find w not in T s.t. D(w) is a minimum;
  add w to T;
  update D(v) for all v adjacent
   to w and not in T:
     D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in T;
```

Example 2: Dijkstra's Algorithm



Example 2: Dijkstra's Algorithm

Iter	T	D(2)	Path	D(3)	Path	D(4)	Path	D(5)	Path	D(6)	Path
0	{1}	2	1–2	5	1-3	1	1–4	∞	-	∞	-
1	{1,4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
2	{1, 2, 4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
3	{1, 2, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
4	{1, 2, 3, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
5	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4–5	4	1-4-5-6

Link State vs. Distance Vector

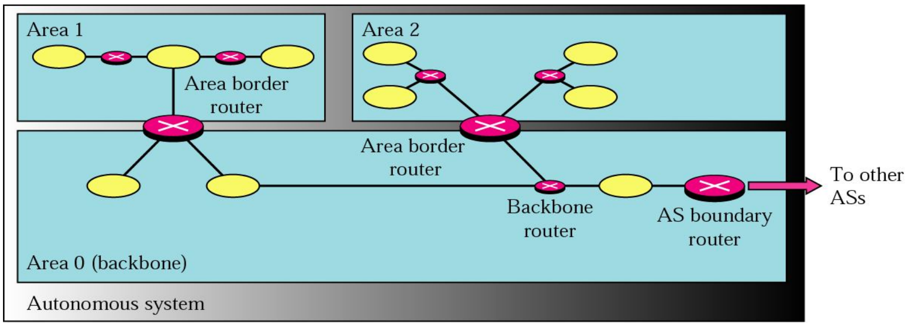
Routing protocol	Builds Topological map	Router can independently determine the shortest path to every network.	Convergence	A periodic/ event driven routing updates	Use of LSP
Distance vector	No	No	Slow	Generally No	No
Link State	Yes	Yes	Fast	Generally Yes	Yes

OSPF

- Open Shortest Path First
- Interior routing protocol
- Uses link state routing

OSPF areas

- For routing efficiency OSPF divides an autonomous system (AS) into areas



OSPF areas

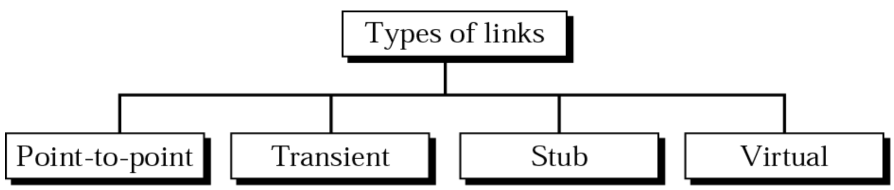
- Routers inside an area flood the area with routing information.
- At the border of an area, area border routers (ABR) summarize the information about the area and send it to other areas.
- All areas inside an AS must be connected to the backbone area (Area 0).

OSPF Metric

- The OSPF protocol allows the administrator to assign a cost, called the metric, to each route.
- The metric can be based on a type of service (min delay, max throughput etc)

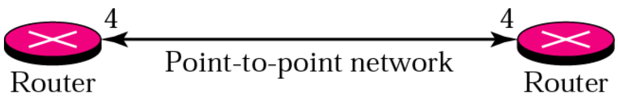
Link types

- In OSPF terminology, a connection is called a *link.*
- Four types of links have been defined:



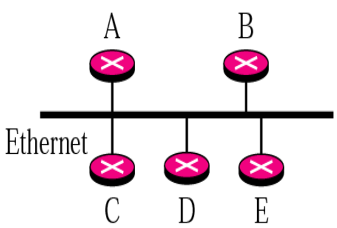
Point-to-point link

- A point-to-point link connects two routers without any other host or router in between.
- In other words, each router has only one neighbor at the other side of the link.



Transient link

- A transient link is a network with several routers attached to it.
- Each router has many neighbors.
- The data can enter through any of the routers and leave through any router.



a. Transient network

Transient link

- To reduce the amount of info exchange, a router is elected as designated router (DR), and another router as a backup designated router (BDR).
- Every router exchanges information with the DR and BDR.
- The DR relays the information to everybody else.

Stub link

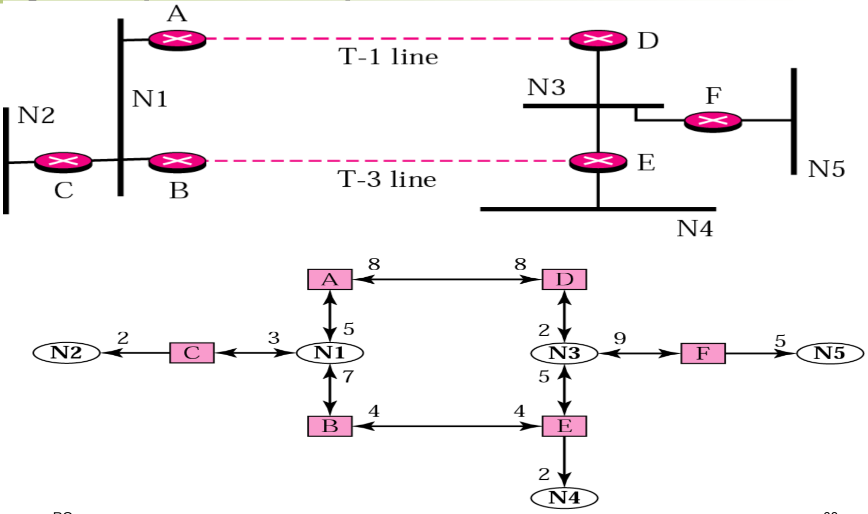
- A stub link is a network that is connected to only one router.
- The data packets enter the network through this single router and leave the network through this same router.

a. Stub network b. Representation

Virtual link

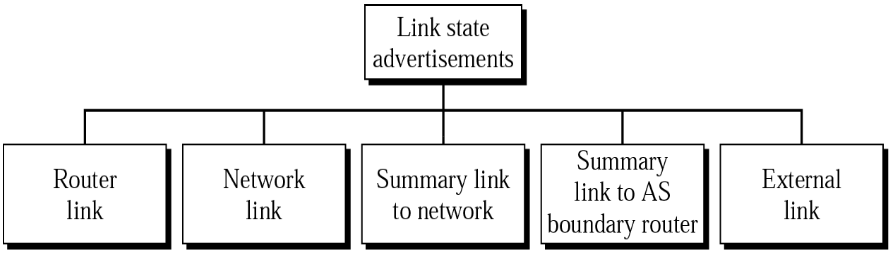
- An administrator can create a virtual link between two routers that may pass through several routers but appear like a single link

Graphical representation



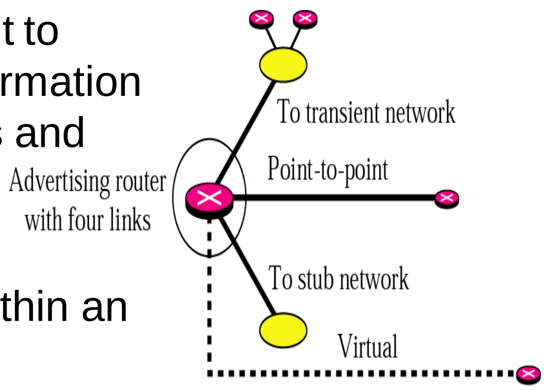
Link state advertisement (LSA)

- Routers share information about their neighbours by distributing LSAs
- Five types of LASs:



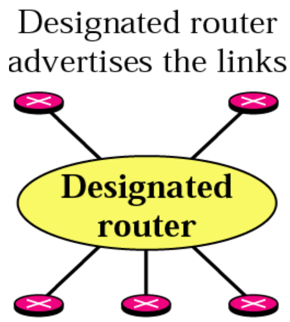
Router link advertisement

- A true router uses this advertisement to advertise information about its links and neighbours
- Distributed within an area



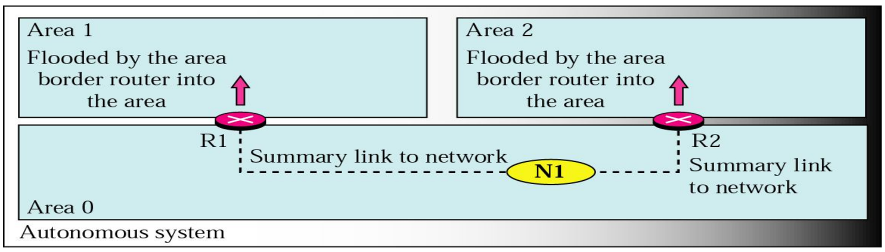
Network link advertisement

- It defines the links of a network
- A designated router distributes this LSA on behalf of the transient network
- Distributed within an area



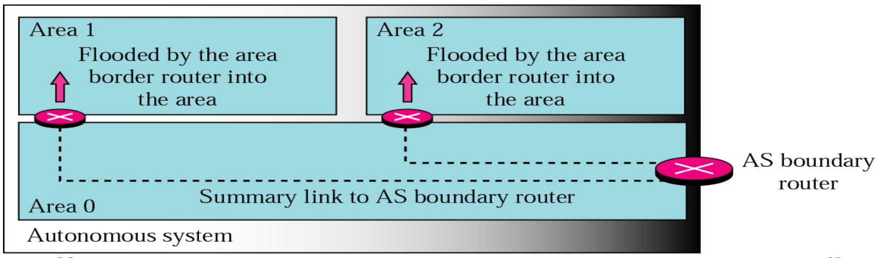
Summary link to network

- This type of LSA is used by area border routers to flood routing information of one area into another area



Summary link to AS Boundary Router

- If a router in an area wants to send packets outside its AS, it needs route to AS boundary router
- This type of LSA provides this info.



External link advertisement

- It advertises routing information about networks outside an AS
- Each advertisement carries routing info about only one external network