Computer Networks
Delhi Technological university
Network Layer: Routing
Instructor: Divya Sethia

Objective

- Routing Algorithms Classification
- Shortest Path First
- Flooding
- Distance Vector Routing (RIP)
- Link State Routing (OSPF)

Routing

•The **routing algorithm is that part of the network layer software responsible for deciding** which output line an incoming packet should be transmitted on
- Process is responsible for filling in and updating the routing tables.

Forwarding is the process in which router handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables

Goals

The main goals of routing are:
1. Correctness: routing should be done properly so that packets may reach proper destination.
2. Simplicity: routing should be done in a simple manner so that overhead is as low as possible. With increasing complexity of the routing algorithms the overhead also increases.
3. Robustness: algorithms designed for routing should be robust enough to handle hardware and software failures and should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted and the network rebooted every time some router goes down.
4.Stability: routing algorithms should always be stable

Goals…

5. Fairness: Every node connected to network should get fair chance of transmitting their packets. This is generally done on a first come first serve basis.

6.Optimality: The routing algorithms should be optimal in terms of throughput and minimizing mean packet delays. Here there is a trade-off and one has to choose depending on his suitability.

Classification of Routing Algorithms

1.Non-Adaptive Routing Algorithm:

- Do not base their routing decisions on measurements and estimates of the current traffic and topology.
- Route to be taken is computed in advance, off-line, and downloaded to the routers when the network is booted.
-Known as static routing.

2. Adaptive Routing Algorithm:

- Change their routing decisions to reflect changes in topology and in traffic as well.
- Get their routing information from adjacent routers or from all routers.
- Optimization parameters are distance, number of hops and estimated transit time.

Internet Routing

- IP implements datagram forwarding
- Both hosts and routers
  - Have an IP module
  - Forward datagrams
- IP forwarding is table-driven
- Table known as *routing table*

Linux commands

$ netstat -rn

Kernel IP routing table

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

$ ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:B0:D0:DF:09:5D

inet addr:153.18.17.11 Bcast:153.18.31.255 Mask:255.255.240.0

. . .

Shortest Path Routing

How to determine the optimal path for routing

- Network is represented as graph, with its terminals as nodes and the links as edges.
- A 'length' is associated with each edge, which represents the cost of using the link for transmission.
Lower the cost, more suitable is the link. The cost is determined depending upon the criteria to be optimized.

Some of the important ways of determining the cost are:
-**Minimum number of hops**
-**Transmission and Propagation Delays**
-Queuing Delays

Shortest Path Routing…

- Shortest paths are calculated using suitable algorithms on the graph representations of the networks.
- Network is represented by graph G (V, E) and let the number of nodes be 'N'.
- Assumption: costs associated with the links are assumed to be positive.
A node has zero cost w.r.t itself.
All the links are assumed to be symmetric, i.e. if $d_{i,j}$ = cost of link from node i to node j, then $d_{i,j} = d_{j,i}$.
- The graph is assumed to be complete.
- If there exists no edge between two nodes, then a link of infinite cost is assumed.
- The algorithm finds costs of the paths from all nodes to a particular node; the problem is equivalent to finding the cost of paths from a source to all destinations.

Shortest Path Routing…

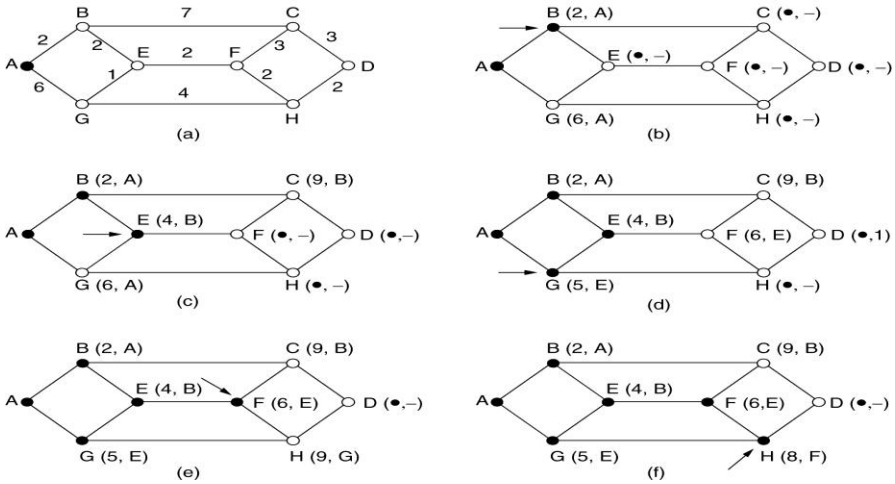
•labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors.

•The algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria

Shortest Path Routing…

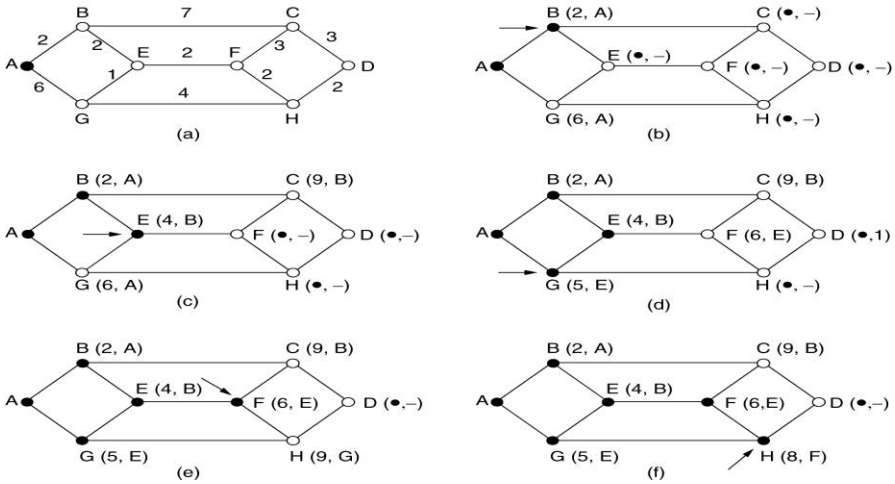
• Consider the shortest path algorithm suggested by :
- Each node is labeled with its distance from the source node along the best known path.
- Initially, no paths are known, so all nodes are labeled with infinity.
- As the algorithm proceeds and paths are found, the labels may change, reflecting better paths
- A label may be either tentative or permanent.
- Initially, all labels are tentative.
- When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

Shortest Path Routing…



The first 5 steps used in computing the shortest path from A to D.
The arrows indicate the working node.

Shortest Path Routing..



The first 5 steps used in computing the shortest path from A to D.
The arrows indicate the working node.

Flooding

•Every incoming packet is sent out on every outgoing line except the one it arrived on

• Generates vast numbers of duplicate packets which can be prevented by:

i) keeping a hop count
- hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero
- Hop count = of the path from source to destination OR full diameter of the subnet if distance is unkown

ii) keep track of which packets have been flooded, to avoid sending them out a second time
- source router put a sequence number in each packet it receives from its hosts
- Each other router maintains list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.

15

Divyashikha Sethia (DTU)

Flooding..

Uses:

-Military : tremendous robustness of flooding is highly desirable when large numbers of routers may be blown to bits

-distributed database applications update all the databases concurrently, in which case flooding can be useful.

- wireless networks, all messages transmitted by a station can be received by all other stations within its radio range, which is, in fact, flooding

- Used as a metric for comparing other algorithms since Flooding always chooses the shortest path because it chooses every possible path in parallel.

16

Building Routing Tables

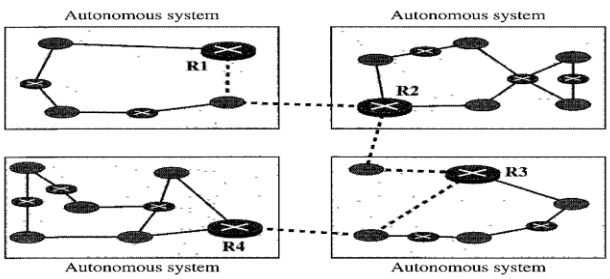
•Depends on size / complexity of internet
•Static routing
– Fixes routes at boot time
– Useful only for simplest cases
•Dynamic routing
– Table initialized at boot time
– Values inserted / updated by protocols that propagate route information
– Necessary in large internets

•Hosts tend to freeze the routing table after initialization
•Routers use protocols to learn new information and update their routing table dynamically

17

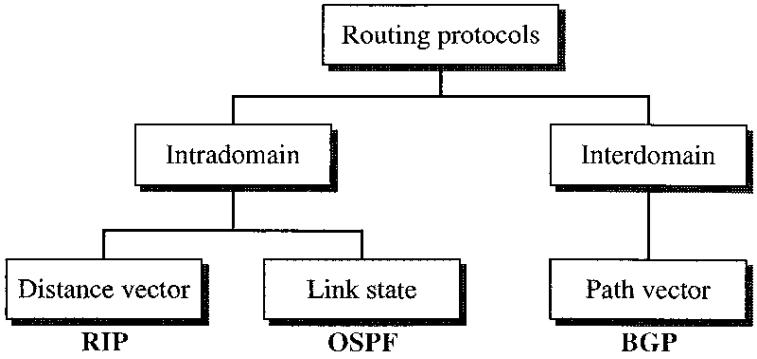
Intra- and Interdomain Routing

• Internet is large hence one routing protocol cannot handle updating all routing tables
•Internet divided into autonomous systems (AS) which is group of networks and routers under single administrative authority
•Intra Domain routing – routing inside AS
•Inter Domain routing - Routing between AS



18

Popular Routing Protocols



19

Automatic Route Propagation

- Two basic algorithms used by routing update protocols
  - Distance-vector
  - Link-state

20

Distance-Vector Algorithm

•The distance vector routing algorithm is sometimes called by other names, most commonly the distributed **Bellman-Ford routing algorithm**

•Initialize routing table with one entry for each directly connected network

•Periodically (30 sec) run a distance-vector update to exchange information with routers that are reachable over directly connected networks

Dynamic Update With Distance-Vector

•One router sends list of its routes to another

•List contains pairs of destination network and distance

•Receiver replaces entries in its table by routes to the sender

•if routing through the sender is less expensive than the current route

•Receiver propagates new routes next time it sends out an update

•Algorithm has well-known shortcomings (we will see an example later)

Example Of Distance-Vector Update

Destination	Distance	Route
Net 1	0	direct
Net 2	0	direct
Net 4	8	Router L
Net 17	5	Router M
Net 24	6	Router J
Net 30	2	Router Q
Net 42	2	Router J

(a)

Destination	Distance
Net 1	2
→ Net 4	3
Net 17	6
→ Net 21	4
Net 24	5
Net 30	10
→ Net 42	3

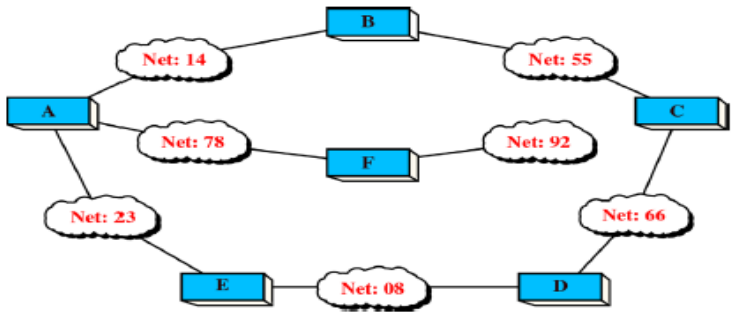
(b)

(a) is existing routing table
(b) incoming update (marked items cause change)

23

Distance-Vector Update

- cost = 1 for every link ⇒ finds *minimum-hop* routes

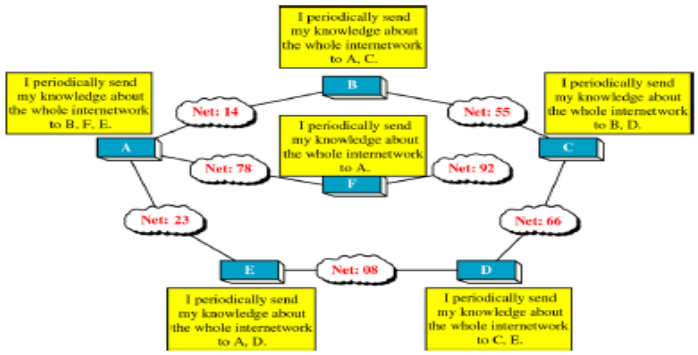


- "clouds" represent LANs; number in cloud represents network ID
- A, B, C, D, E, F are routers (or gateways)

24

Distance-Vector Update

- each router sends its information about the entire network only to its neighbours



- how do non-neighbouring routers learn about each other and share information?
- a router sends its information to its neighbours
- each neighbour router adds this information to its own, and sends the updated information to *its* neighbours; the first router learns about its neighbours' neighbours

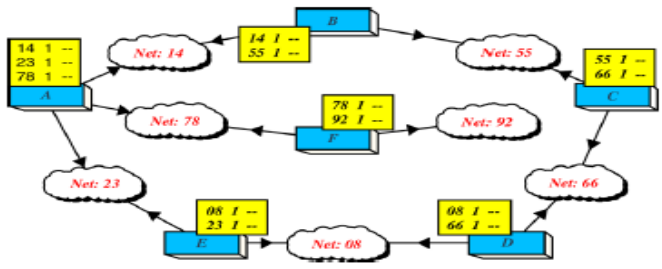
Distance-Vector Update

- each router stores information about the network in its *routing table*

Network ID	Cost	Next Hop

Network ID = final destination of Packet
Cost = number of hops from this router to final destination
Next Hop = neighbouring router to which Packet should be sent

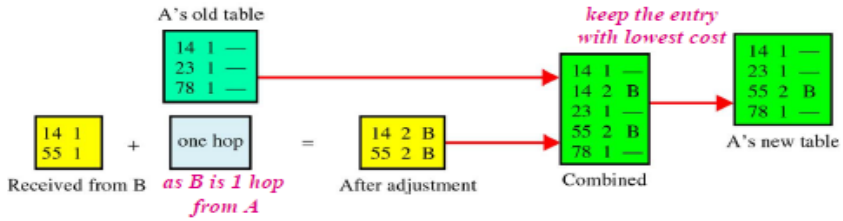
- initially, all a router knows is the network IDs of the networks to which it is directly connected



initial routing table exchanges (no multi-hop routes yet)

Distance-Vector Update

- how is a router's routing table updated when new information is received?

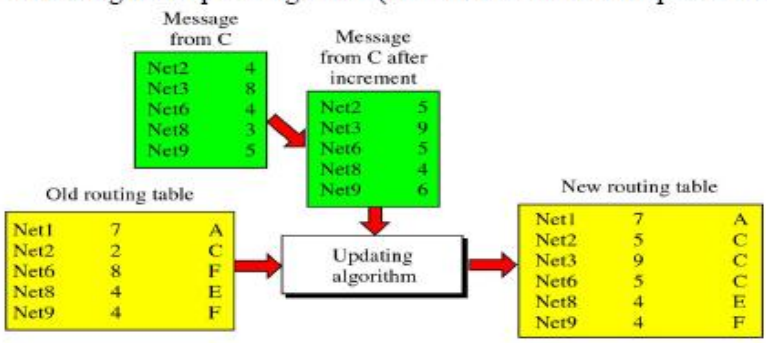


- routing table update algorithm *(distributed Bellman-Ford algorithm)*:
  - add 1 to cost of each incoming route (since each neighbour is 1 hop away)
  - if a new destination is learned, add its information to the routing table
  - if new information received on an existing destination:
    - if Next Hop field is the same, replace existing entry with the new information *even if the cost is greater* ("new information invalidates old")
    - if Next Hop field is not the same, *only* replace existing entry with the new information *if the cost is lower*

27

Distance-Vector Update

- example of routing table update algorithm (unrelated to earlier Example network):

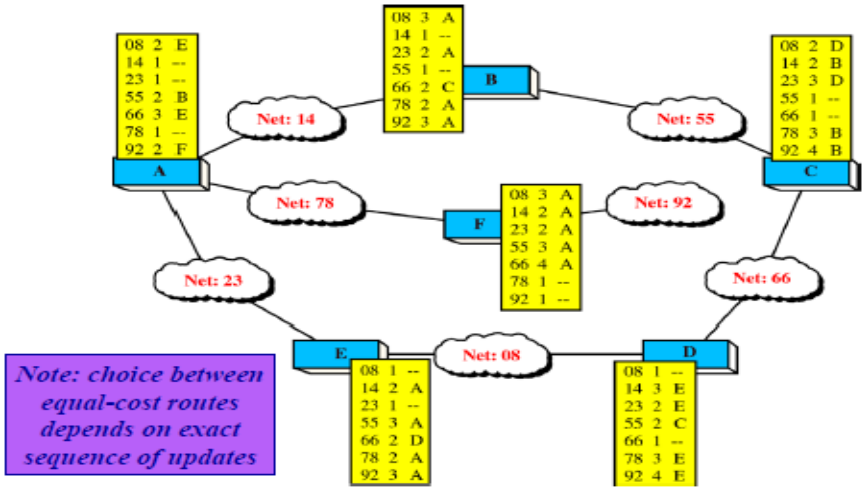


Rules
Net2: Replace **(Rule 2.a)**
Net3: Add **(Rule 1)**
Net6: Replace **(Rule 2.b.i)**
Net8: No change **(Rule 2.b.ii)**
Net9: No change **(Rule 2.b.ii)**

Note that there is no news about Net1 in the advertised message, so none of the rules apply to this entry.

Distance-Vector Update

• final (converged) routing tables for earlier Example network:



29

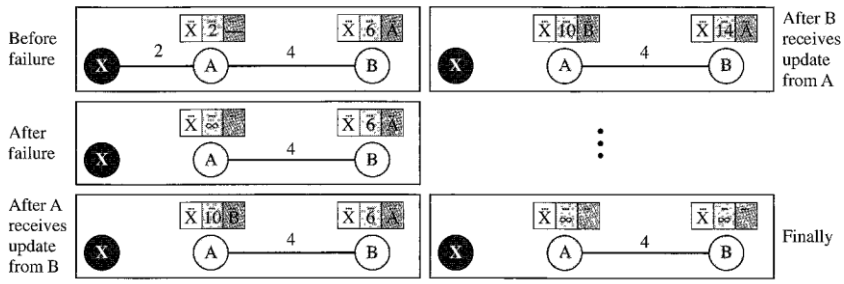
Sharing updates

• Periodic Update: Node sends routing table every 30 s

•Triggered Update – Node sends its routing table to its neighbors anytime there is a change in its routing table (received table from neighbor resulting in some change, or has observed failure in neighboring link)

30

Two-node Loop Instability



31

Two-node Loop Instability….

Solution:

-**Defining Infinity** – Distance 16 is infinity – implies Distance vector cannot use large systems with hops more than 15

-**Spilt Horizon :**

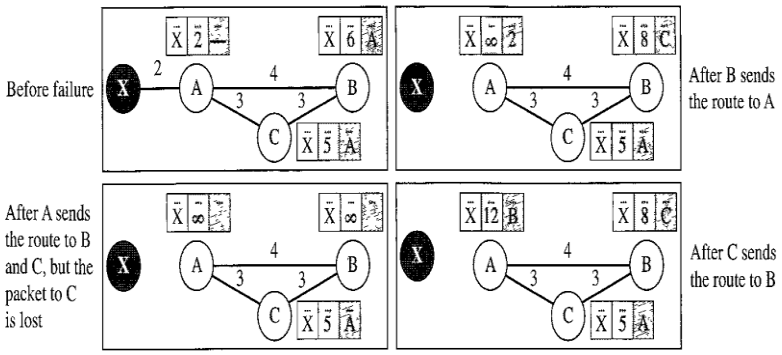
- Do not flood entire routing table information on all interfaces (only part of it) Router does not propagate information about route back over same interface from which the route arrived
- B reaches X via A hence it does not advertise this infromation of reaching X to A
- A finds that X is unreachable and keeps distance of X as infinity
- Later when A sends routing triggered update to B it corrects its routing table and systems becomes stable

32

Two-node Loop Instability….

Solution..
-Split Horizon and Poison Reverse
- Drawback of simple split horizon : If timer of route for X at B expires before it gets triggered update from A, it deletes route to X
- B will not be sending route to X in its ad to A
- A does not know that if this is due to split horizon or because B has not received any news about X recently
- **Split horizon** combined with **poison reverse**: B can still advertise X – but if it is sending info to source A, it replaces distance with infinity (indicating that B received info about X from A and it must not use it)

Three-Node instability



• Loop created and stops when each node reaches infinity

Routing Information Protocol (RIP)

Divyashikha Sethia (DTU)

•Implemented by UNIX program *routed*
•Uses hop count metric
•Distance-vector protocol
•Relies on broadcast
•Assumes low-delay local area network
•Uses split horizon and poison reverse techniques to solve inconsistencies

35

Divyashikha Sethia (DTU)

Two Forms Of RIP

•Active
– Form used by routers
– Broadcasts routing updates periodically
– Uses incoming messages to update routes
•Passive
– Form used by hosts
– Uses incoming messages to update routes
– Does not send updates

36

Divyashikha Sethia (DTU)

RIP Operation

•Each router sends update every 30 seconds
•Update contains pairs of (destination address, distance)
•Distance of 16 is *infinity (i.e., no route)*

37

Divyashikha Sethia (DTU)

Problem with RIP

• Slow Convergence
• Limited nw size (infinity is 16 hops)
•Routing Loops
A routing loop occurs when Router *A* has an entry telling it to send datagrams for Network *1* to Router *B*, and Router *B* has an entry saying that datagrams for Network *1* should be sent to Router *A*. Larger loops can also exist: Router *A* says to send to *B*, which says to send to *C*, which says to send to *A*.
While under normal circumstances these loops should not occur, they can happen in special situations. RIP does not include any specific mechanism to detect or prevent routing loops; the best it can do is try to avoid them.

38

Divyashikha Sethia (DTU)

Link-State Algorithm

•Alternative to distance-vector
The idea behind link state routing is simple and can be stated as five parts. Each router must do the following:
1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to **all** other routers.
5. Compute the shortest path to every other router. (Shortest Path Frist)

39

Divyashikha Sethia (DTU)

Link-State Update…

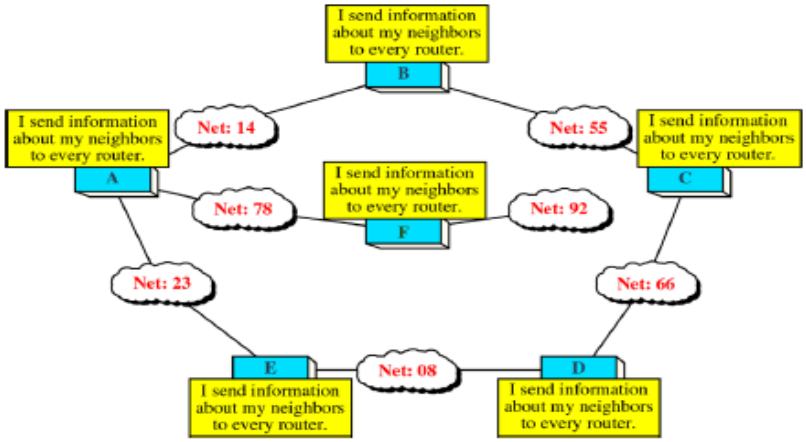
•Participating routers learn internet topology
•Think of routers as nodes in a graph, and networks connecting them as edges or links
•Pairs of directly-connected routers periodically
– Test link between them
– Propagate (broadcast) status of link
•All routers
– Receive link status messages
– Recompute routes from their local copy of information

40

Link-State Update…

Network Layer: Link-State routing

• each router sends information about its neighbourhood to every other router:



41

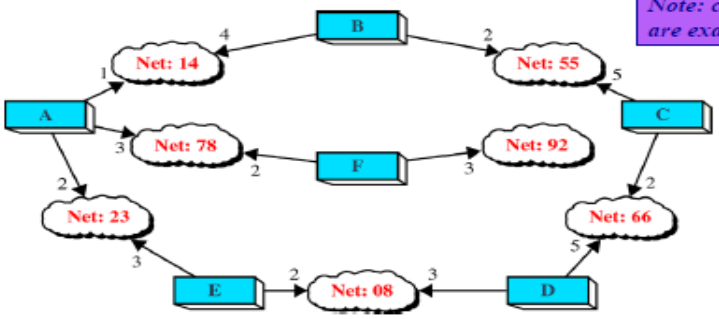
Link-State Update…

Network Layer: Link-State routing (cont.)

• link cost is usually a weighted sum of various factors
  • e.g. traffic level, security level, packet delay, …

• link cost is *from* a router *to* the network connecting it to another router
  • when a packet is in a LAN (which is typically a broadcast network), every node – including the router – can receive it ⇒ no cost assigned when going from a network to a router

Note: costs shown are examples only



42

Link-State Update…

Network Layer: Link-State routing (cont.)

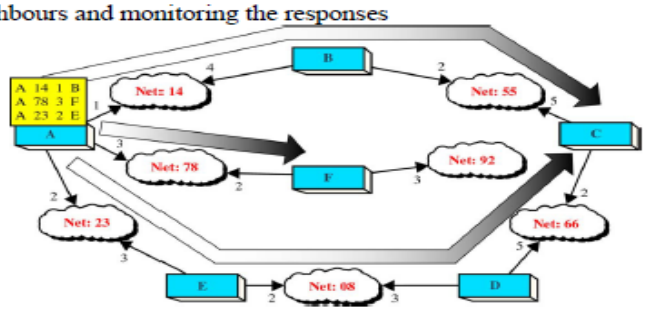
• routers share information by *advertising*, which means sending *link-state packets*

Advertiser	Network	Cost	Neighbor
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....

Advertiser: ID of sending router
Network: ID of destination network
Cost: link cost to neighbour
Neighbour: ID of neighbour router

• a router gets its information about its neighbourhood by sending short ECHO packets to its neighbours and monitoring the responses

the Figure shows how router A's link-state packet is flooded to all other routers



43

Link-State Update…

Network Layer: Link-State routing (cont.)

• every router builds a link-state packet and floods it through the network, so when all such packets have been received at a router, it can build its link-state database:

Advertiser	Network	Cost	Neighbor
A	14	1	B
A	78	3	F
A	23	2	E
B	**14**	**4**	**A**
B	**55**	**2**	**C**
C	55	5	B
C	66	2	D
D	**66**	**5**	**C**
D	**08**	**3**	**E**
E	23	3	A
E	08	2	D
F	**78**	**2**	**A**
F	**92**	**3**	**—**

Assuming that every router receives the same set of link-state packets (as if the routers were synchronised), every router builds the same link-state database.

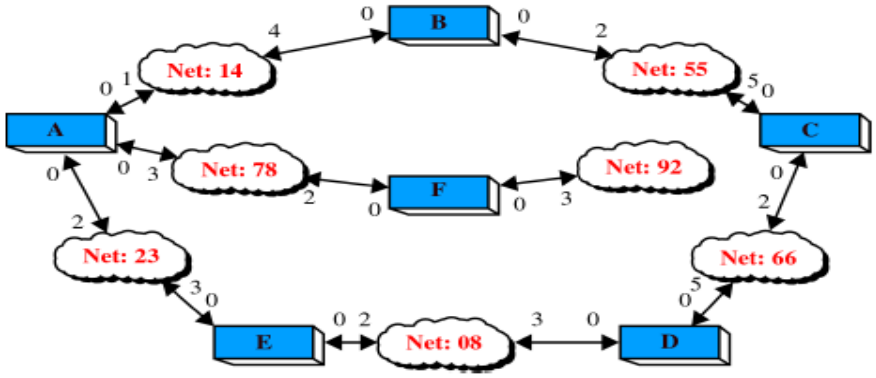
Using this database, each router can then calculate its routing table.

44

Link-State Update…

Network Layer: Link-State routing (cont.)

- to calculate its routing table, a router uses ***Dijkstra's Shortest-Path algorithm***
  - first, identify all link costs in the network: either from the link-state database, or using the fact that the cost of any link from a network to a router is 0



45

Link-State Update…

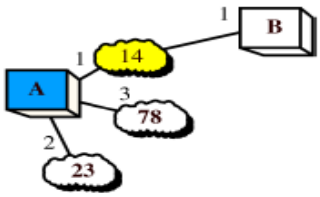
Network Layer: Link-State routing – Dijkstra's algorithm (cont.)

- as an Example, let's follow the steps of the algorithm run by router A

1.

Root is A, networks 14, 78, 23 added

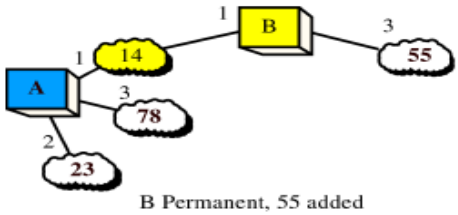
2.



14 permanent, B added

Note: arcs are marked with their cumulative cost from the root (not individual costs)

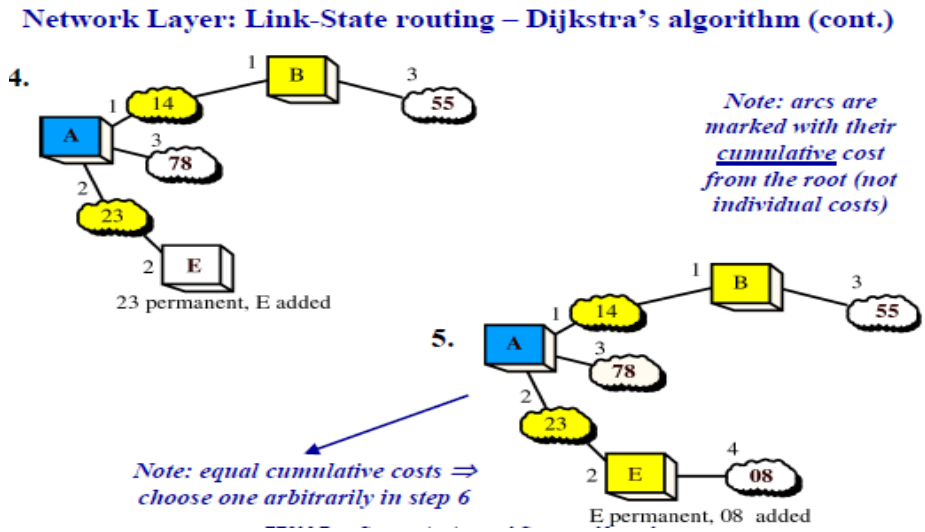
3.



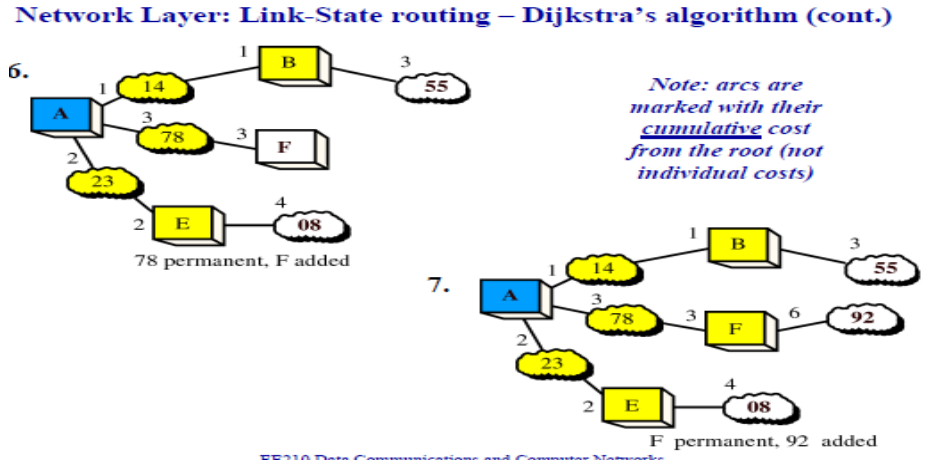
B Permanent, 55 added

46

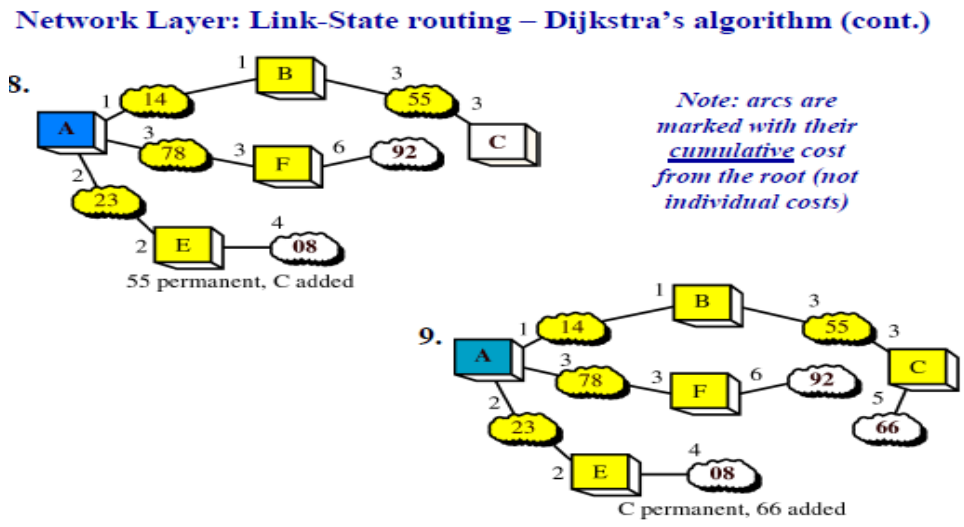
Link-State Update…



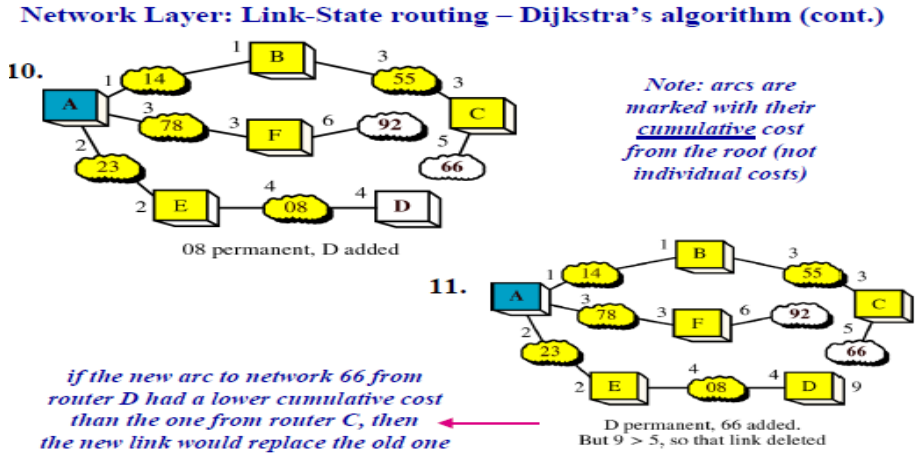
Link-State Update…



Link-State Update…



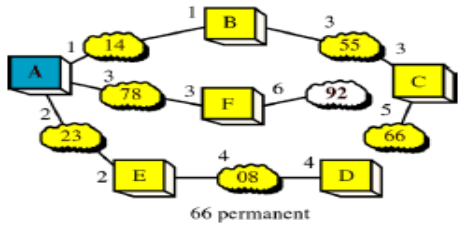
Link-State Update…



Link-State Update…

Network Layer: Link-State routing – Dijkstra's algorithm (cont.)

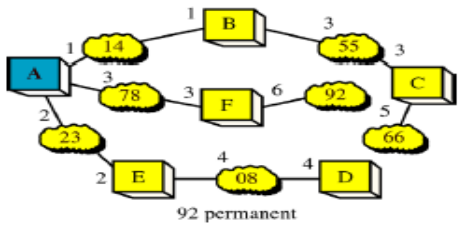
12.



Note: arcs are marked with their cumulative cost from the root (not individual costs)

13.

all nodes and arcs are Permanent ⇒ STOP: this router's shortest-path spanning tree has been found



51

Link-State Update…

Network Layer: Link-State routing – routing table

- once a router has found its shortest-path spanning tree, it can build its routing table
  - to complete the Example, here is router A's link-state routing table:

Net	Cost	Next router
08	4	E
14	1	--
23	2	--
55	3	B
66	5	B
78	3	--
92	6	F

Note: each router's routing table will (in general) be different

Networks 14, 23, and 78 don't have a "Next router" entry because they are directly connected to this router

- in large networks, the *memory required* to store the link-state database and the *computation time* to calculate the link-state routing table can be significant
- in practice, since the link-state packet receptions are not synchronised, routers may be using different link-state databases to build their routing tables: how inaccurate the results are depends on how different the routers' "views" of the network are

52

Link-State Update…

- link-state routing algorithms have several desirable properties, e.g. rapid convergence; small amount of traffic generated; rapid response to topology changes
- examples from the Internet are the *Open Shortest Path First* (OSPF) and *Intermediate System to Intermediate System* (IS-IS) routing protocols
  - link costs can be configured in OSPF. Possible link costs include:
    - 1 for each link
    - reliability: assigned by administrator, indicates how often the link fails
    - packet delay
    - link bandwidth
    - financial cost of the link
  - OSPF requires a lot of memory: each router holds its routing table & link-state database
  - Dijkstra's algorithm computations are processor-intensive
    - legacy routers may be unable to relay packets when these calculations are taking place – which *could* be every time a link-state packet is received
  - OSPF *can* consume a lot of bandwidth if the network topology changes often
  - link-state packets sent to all routers using **reliable flooding**
    - need sequence number and time-to-live (TTL) field in each packet…

53

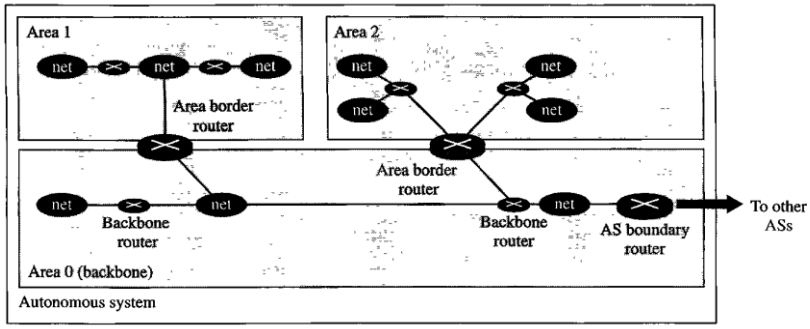
LSP (Link State Packet)

• LSP contents:
i) Node identity
ii) List of links
iii)Sequence number (To distinguish old and new LSPs)
iv)Age (prevents old LSPs from remaining in domain for long time)

Sharing of LSPs:
i) On change of topology
ii) On periodic basis (much longer time 60 min to 2 hr – to prevent traffic sue to flooding)

54

OSPF Areas



• Areas – to handle routing efficiently and in timely manner divides AS into areas which are interconnected
•Routers inside area flood routing info only within area
•**Area Border routers** are special routers at the border which summarize info about area and send it to other areas.
•Special area – Backbone area with routers known as **backbone routers**– all areas must be connected to the backbone

55

References

• Chapter 22 Network Layer: Delivery Forwarding and Routing, Forouzan (section 22.1 – 22.3)
•Chapter 5, Section 5.2 Routing Algorithms, Computer Networks by Tanenbaum
•Chapter 14, chapter 16: TCP/IP volume 1 by Douglas Comer, fourth edition

•http://www.networkcomputing.com/netdesign/ip101.html

http://www.tcpipguide.com/free/t_toc.htm
http://engweb.info/courses/itcn/dist-vector-routing-protocols/dv-routing-prots.html

56

THANKS