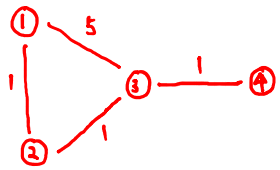
2. Link State Routing

- Each node maintains three lists:
  1. A list of next hops to each destination node along the shortest path
     - routing table
  2. A list of distances to each destination node
     - distance vector
  3. A list of link costs to each neighbor node
     - ∞ link cost to each non-neighbor node

E.g.



At node 1:

dest j	H_{1j}	D_{1j}	C_{1j}
1	1	0	0
2	2	1	1
3	2	2	5
4	2	3	∞

∞ → Link state (local info)

routing table

→ Distance Vector (global info)

- Each node broadcasts its local link state information to all other nodes in the network, often via flooding
  - Each node has complete information about all links in the network

local info exchanged globally

Dijkstra's Algorithm

- Each node will calculate its shortest paths to all other nodes upon receiving new link state information

- Dijkstra's Shortest-Path Routing Algorithm
  - An iterative algorithm to find the shortest paths from a source node to all other nodes in the network

algorithm executed at

Dijkstra's Algorithm

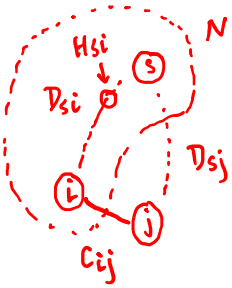
- Notations:
  - s: source node
  - N: the set of nodes whose shortest paths have already been found

- Initialization Step
  - $N = \{s\}$, $D_{sj} = C_{sj}$ and $H_{sj} = j$ for all j
- Step A: (Finding the next closest node i)
  - Find node $i \notin N$ such that $D_{si} = \min D_{sj}$ for $j \notin N$
  - Add i to N
  - If N contains all the nodes, Stop

Dijkstra's Algorithm

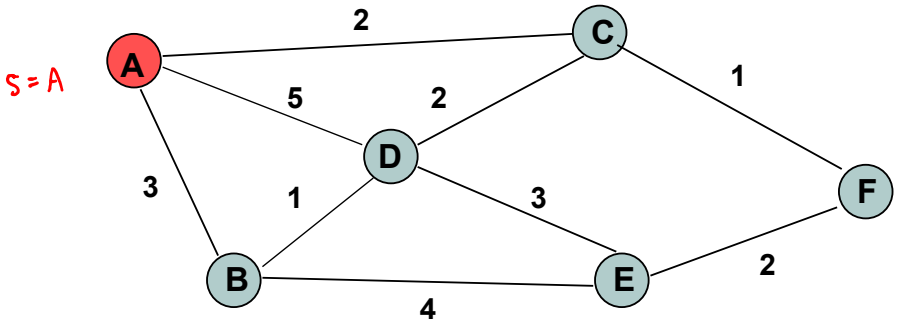
- Notations:
  - s: source node
  - N: the set of nodes whose shortest paths have already been found

- Step B: (Updating minimum costs after node i is added to N)
  - For each node j ∉ N
    - if $(D_{si} + C_{ij}) < D_{sj}$, then $D_{sj} = D_{si} + C_{ij}$ and $H_{sj} = H_{si}$
  - Go to Step A



if $D_{si} + C_{ij} < D_{sj}$
then $D_{sj} = D_{si} + C_{ij}$
$H_{sj} = H_{si}$?

Example: From Source Node A



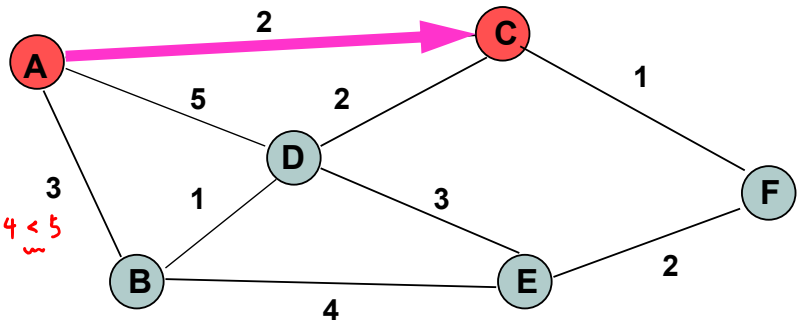
local table

(H; D) pair

Iteration	N	H_{AB}; D_{AB}	H_{AC}; D_{AC}	H_{AD}; D_{AD}	H_{AE}; D_{AE}	H_{AF}; D_{AF}
Initial	{A}	B; 3	C; 2	D; 5	E; ∞	F; ∞

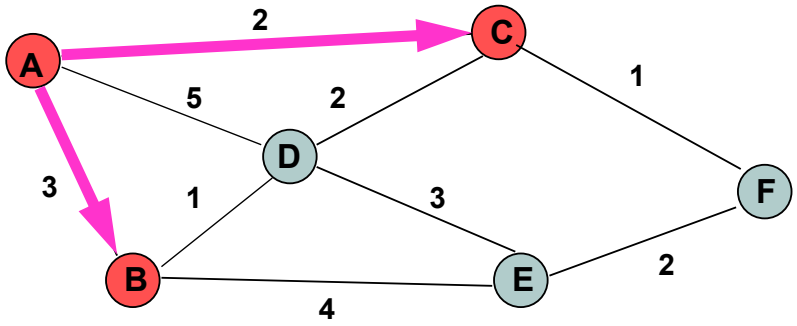
Example: From Source Node A

s = A
i = C
j = D
$D_{AD} = 5$
$D_{AC} + C_{CD} = 2 + 2 = 4 < 5$
$\Rightarrow D_{AD} = 4$
$H_{AD} = H_{AC} = C$



Iteration	N	$H_{AB}; D_{AB}$	$H_{AC}; D_{AC}$	$H_{AD}; D_{AD}$	$H_{AE}; D_{AE}$	$H_{AF}; D_{AF}$
Initial	{A}	B; 3	C; 2	D; 5	E; ∞	F; ∞
1	{A,C}	B; 3	C; 2	C; 4	E; ∞	C; 3

Example: From Source Node A

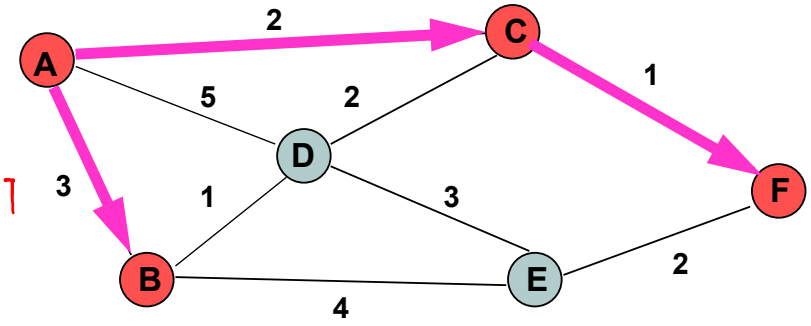


Iteration	N	$H_{AB}; D_{AB}$	$H_{AC}; D_{AC}$	$H_{AD}; D_{AD}$	$H_{AE}; D_{AE}$	$H_{AF}; D_{AF}$
Initial	{A}	B; 3	C; 2	D; 5	E; ∞	F; ∞
1	{A,C}	B; 3	C; 2	C; 4	E; ∞	C; 3
2	{A,B,C}	B; 3	C; 2	C; 4	B; 7	C; 3

Example: From Source Node A

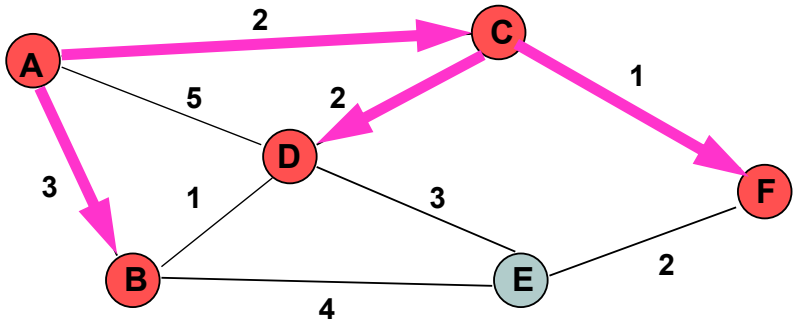
s = A
i = F
j = E

$D_{AE} = 7$
$D_{AF} + C_{FE} = 3 + 2 = 5 < 7$
$\Rightarrow D_{AE} = 5$
$H_{AE} = H_{AF} = C$



Iteration	N	$H_{AB}; D_{AB}$	$H_{AC}; D_{AC}$	$H_{AD}; D_{AD}$	$H_{AE}; D_{AE}$	$H_{AF}; D_{AF}$
Initial	{A}	B; 3	C; 2	D; 5	E; ∞	F; ∞
1	{A,C}	B; 3	C; 2	C; 4	E; ∞	C; 3
2	{A,B,C}	B; 3	C; 2	C; 4	B; 7	C; 3
3	{A,B,C,F}	B; 3	C; 2	C; 4	C; 5	C; 3

Example: From Source Node A

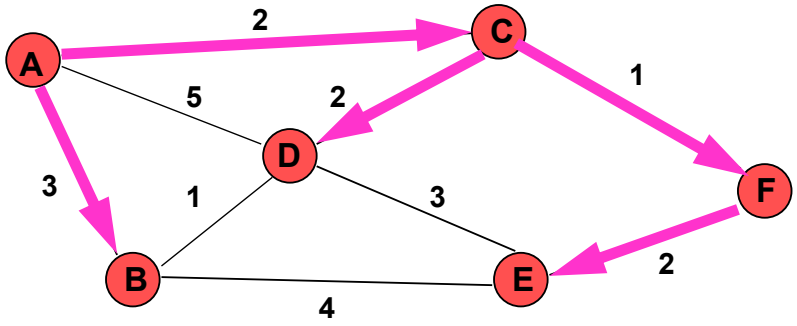


Iteration	N	$H_{AB}; D_{AB}$	$H_{AC}; D_{AC}$	$H_{AD}; D_{AD}$	$H_{AE}; D_{AE}$	$H_{AF}; D_{AF}$
Initial	{A}	B; 3	C; 2	D; 5	E; ∞	F; ∞
1	{A,C}	B; 3	C; 2	C; 4	E; ∞	C; 3
2	{A,B,C}	B; 3	C; 2	C; 4	B; 7	C; 3
3	{A,B,C,F}	B; 3	C; 2	C; 4	C; 5	C; 3
4	{A,B,C,D,F}	B; 3	C; 2	C; 4	C; 5	C; 3

Example: From Source Node A

At A:

j	H_{Aj}	D_{Aj}	C_{Aj}
A	A	0	0
B	B	3	3
C	C	2	2
D	C	4	5
E	C	5	8
F	C	3	8



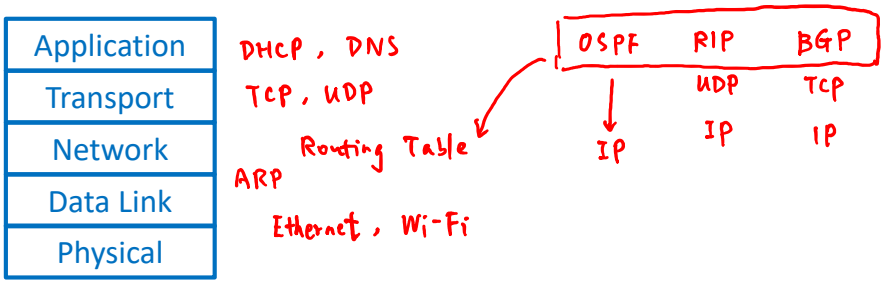
Iteration	N	H_{AB}; D_{AB}	H_{AC}; D_{AC}	H_{AD}; D_{AD}	H_{AE}; D_{AE}	H_{AF}; D_{AF}
Initial	{A}	B; 3	C; 2	D; 5	E; ∞	F; ∞
1	{A,C}	B; 3	C; 2	C; 4	E; ∞	C; 3
2	{A,B,C}	B; 3	C; 2	C; 4	B; 7	C; 3
3	{A,B,C,F}	B; 3	C; 2	C; 4	C; 5	C; 3
4	{A,B,C,D,F}	B; 3	C; 2	C; 4	C; 5	C; 3
5	{A,B,C,D,E,F}	B; 3	C; 2	C; 4	C; 5	C; 3

Reaction to Link Failure

- If a link is broken,
  - Affected nodes set link cost to infinity & flood the network with update packets
  - All nodes immediately update their link database & re-calculate their shortest paths
  - Recovery is very quick
- Link State Routing is NOT loop-free
  - Due to delay in link state information propagation

OSPF (Open Shortest Path First)

- Link State Routing Protocol
- OSPF runs directly over IP
  - Value in the protocol field of IP header: 89
- OSPF typically converges faster than RIP when there is a failure in the network



Autonomous Systems

- Autonomous System (AS) is defined as a set of routers or networks administered by a single organization
  - Stub AS: has only a single connection to the outside world
  - Multi-homed AS: has multiple connections to the outside world, but refuses to carry transit traffic
  - Transit AS: has multiple connections to the outside world, and can carry both transit and local traffic

Inter and Intra AS Routing

- IGP (Interior Gateway Protocol): routing within an AS
  - RIP, OSPF (possibly routing loops)
- EGP (Exterior Gateway Protocol): routing between ASs
  - BGPv4 (loop-free)