CMPE 257: Wireless Networking

SET 5:

Unicast Routing in MANETs

Proactive Approaches

Optimized Link State Routing (OLSR) [Jacquet00ietf]

- Overhead of flooding link state information reduced by having fewer nodes forward the information.
- Broadcast from X only forwarded by its *multipoint relays (MPRs).*

Proactive Approaches

Schemes:

- Topology broadcast (OLSR)
- Partial topology information or path information
- Distance vectors with some constraint (typically a sequence number)

Techniques:

- Same three types of termination detection

OLSR

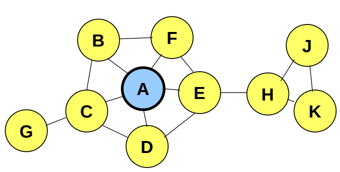
- OLSR is proactive.
- It floods information through MPRs.
- Flooded information contains links connecting nodes to respective MPRs.
  - I.e., node sends info on nodes that selected it as their MPR.
  - Periodic HELLO messages inform nodes which other nodes selected it as their MPR.
- Routes used by OLSR only include multipoint relays as intermediate nodes.

MPRs

- Multipoint relays of node X are its neighbors such that each two-hop neighbor of X is a one-hop neighbor of at least one multipoint relay of X.
  - Each node transmits its neighbor list in periodic beacons, so that all nodes know their 2-hop neighbors.
- MPRs of X are 1-hop neighbors of X covering X's 2-hop neighbors.

Optimized Link State Routing (OLSR)

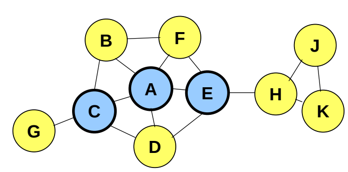
- C and E are multipoint relays of A.



Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

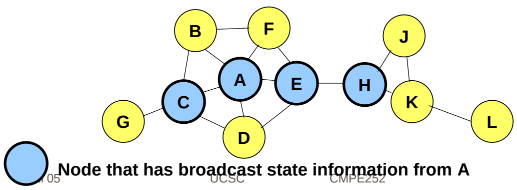
- Nodes C and E forward information received from A.



Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

- E and K are multipoint relays for H.
- K forwards information received from H.
  - E has already forwarded the same information once.



Termination Detection over Cycles

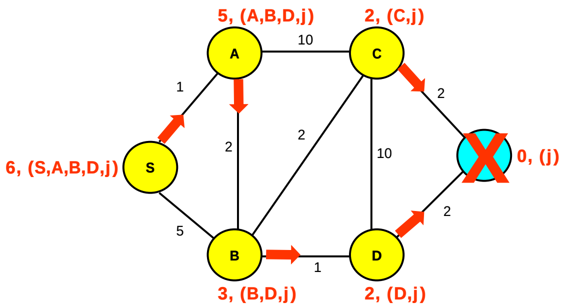
Motivation

- Topology broadcast and distance flooding incur too many updates.
- Diffusing computations may incur too many update messages when nodes need to synchronize over many hops.
- Objective: Have the same information regarding paths available with topology-broadcast algorithms, without the communication overhead.

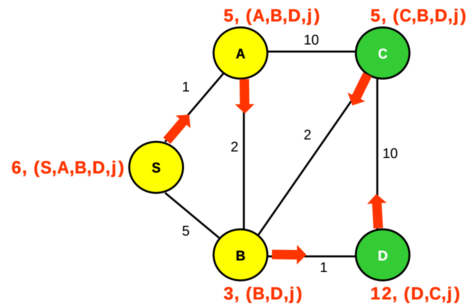
Basic Approach

- Each router maintains the entire path to each destination.
- Update for destination j report the length and node constituency of path to j.
- Complete path information is used to **detect** loops.
- A router can always adopt a path to a destination that does not already include the router itself and has the shortest length among all valid paths.
- Example: BGP

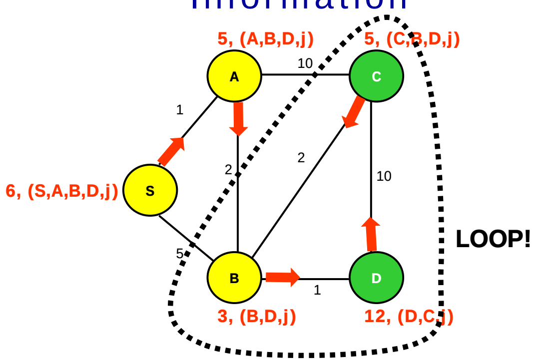
Detecting Loops Using Path Information



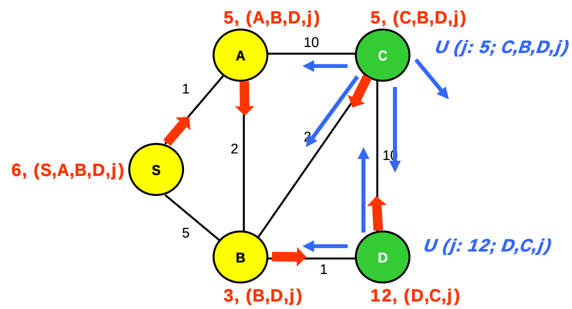
Detecting Loops Using Path Information



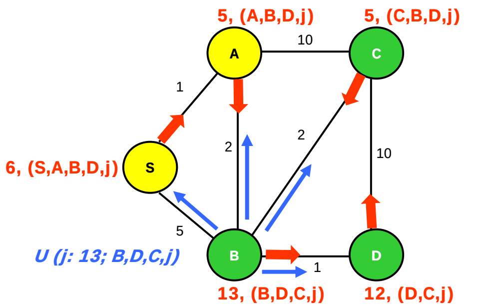
Detecting Loops Using Path Information



Detecting Loops Using Path Information

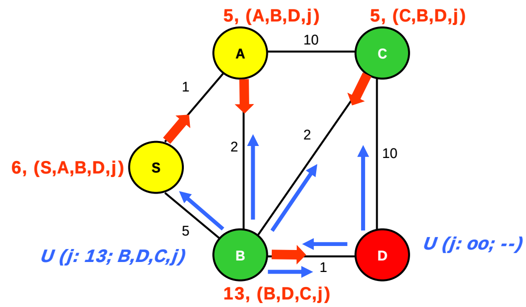


Detecting Loops Using Path Information



Update from D forces B to update its path to j

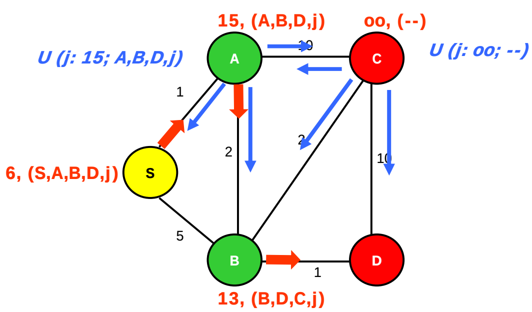
Detecting Loops Using Path Information



Update from C makes D detect and break loop

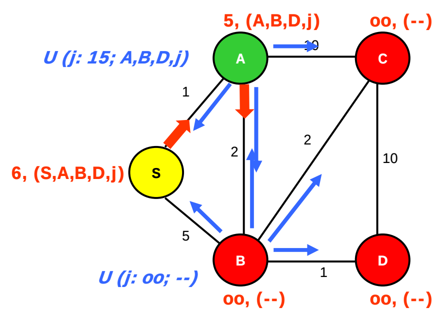
All paths reported to D include D, and it must set j as unreachable

Detecting Loops Using Path Information



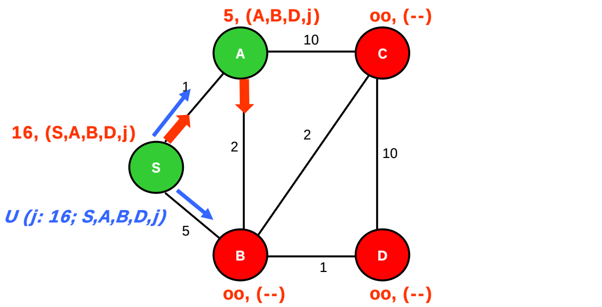
Update from B makes A update its path to j; update from D makes all reported paths include C

Detecting Loops Using Path Information



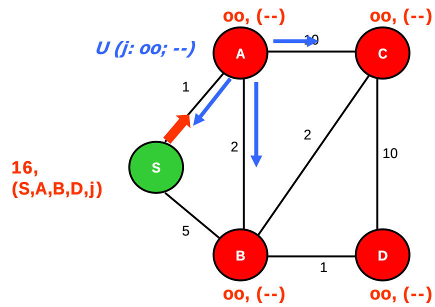
After update from D, all finite-length paths reported to B include B, and it must set j as unreachable

Detecting Loops Using Path Information



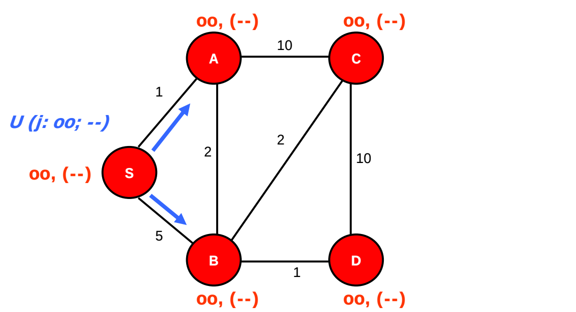
Update from A makes S change its path to j

Detecting Loops Using Path Information



After update from B, all finite-length paths reported to A include A, and it must set j as unreachable

Detecting Loops Using Path Information



No counting to infinity, but temporary loops exist!

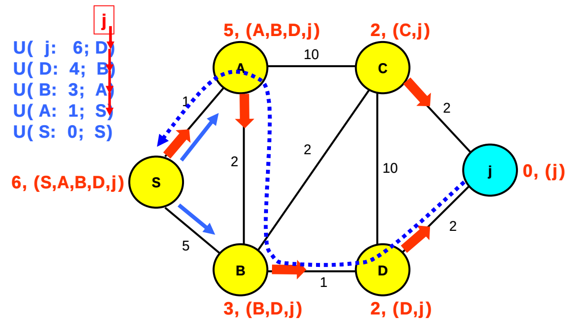
Limitations of Basic Approach

- Temporary loops occur.
- Complete path information is not necessary in updates!
- Loop detection based on reported paths exclusively is not very efficient!
  - A router can believe a neighbor whose reported path includes a neighbor that has just reported an updated path that is invalid!
- Fixing these limitations leads to the Loop-Free Path Finding Algorithm (LPA).

Path Finding

- A tree rooted at s can be represented by specifying the nodes in the path traversed from s to each other node in the tree.
- This has too much redundancy!
- The same tree can be represented by specifying the second-to-last hop of the path traversed from s to each other node in the tree.
- The second-to-last hop in the path to destination d is called the predecessor of d.

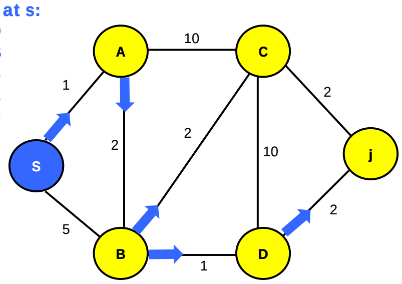
Path Finding



If A is the next hop to j, it must also be the next hop to each node in the path to j

Path Finding

Source Tree at s:
j: 6; D
D: 4; B
C: 5; B
B: 3; A
A: 1; S
S: 0; S



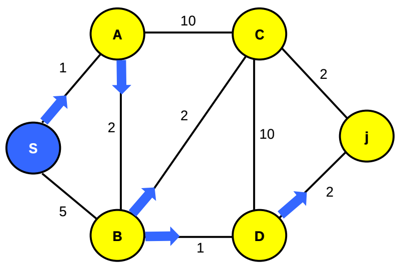
Each router conveys to its neighbors its shortest-path routing tree.
This is the same information obtained from having complete topology maps!

Path Finding with Link-State Information

- The same solution can be applied to routing using partial link-state information.
- Each router maintains and reports the state of those links along the shortest paths to each destination.
- Each node keeps the shortest-path tree reported by each neighbor, and runs local path selection algorithm over the aggregated graph.
- Examples:
  - Link Vector Algorithm (Jochen Behrens and JJ, SIGCOMM 94 and IEEE JSAC 94)
  - Source Tree Adaptive Routing (Marcelo Spohn and JJ, MONET Journal 2001)

Path Finding with Link States

Source Tree at s:
(D,j): 2
(B,D): 1
(B,C): 2
(A,B): 2
(S,A): 1
(S,S): 0



Each router conveys to its neighbors its shortest-path routing tree.
Tricky part is how to reduce communication overhead.

Location-Aided Routing (LAR) [Ko98Mobicom]

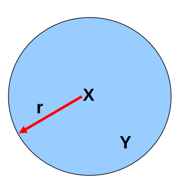
- Exploits location information to limit scope of route request flood.
  - Location information may be obtained using GPS.
- *Expected Zone* : region expected to hold the current location of the destination.
  - Expected region based on old location information, and knowledge of destination's speed.
- Route requests limited to a *Request Zone* that contains Expected Zone and location of sender node.

Expected Zone in LAR

X = last known location of node D, at time t0

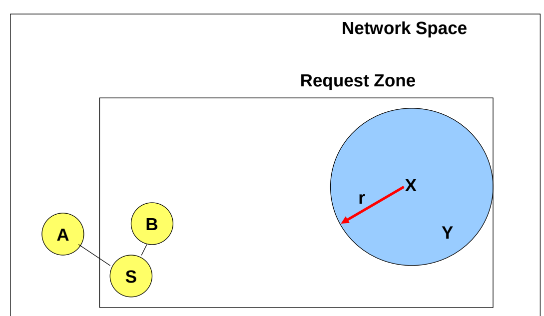
Y = location of node D at current time t1, unknown to node S

r = (t1 - t0) * estimate of D's speed



Expected Zone

Request Zone in LAR



LAR

- Only nodes within request zone forward RREQs.
  - Node A does not forward RREQ, but node B does.
- Request zone explicitly specified in the RREQ.
- Each node must know its physical location to determine whether it is within the request zone.

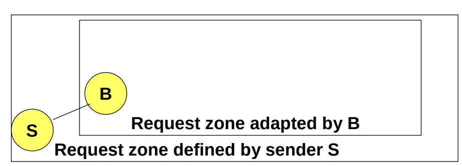
LAR

- If route discovery using smaller request zone fails to find a route, sender initiates another route discovery (after a timeout) using larger request zone.
  - Larger request zone may be the entire network.
- Rest of route discovery protocol similar to DSR.

LAR Variations: Adaptive Request Zone

- Each node may modify the request zone included in the forwarded request
- Modified request zone may be determined using more recent/accurate information, and may be smaller than the original request zone

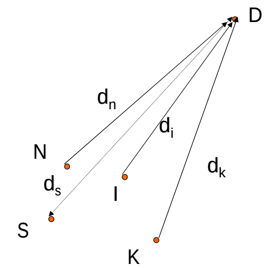
Adaptive Request Zones



LAR Variations: Implicit Request Zone

- In the previous scheme, RREQ explicitly specified request zone.
- Alternate approach: node X forwards RREQ received from Y if X is deemed to be closer to expected zone as compared to Y.
- The motivation is to attempt to bring the RREQ physically closer to the destination node after each forwarding.

Implicit Request Zone



RREQ includes position of D and distance of current node to D.

More on LAR...

- Basic LAR assumes that, *initially,* location information for X becomes known to Y only during route discovery.
- This location information is used for future route discovery. Why?

Variations

- Location information can also be piggybacked on any message from Y to X.
- Y may also proactively distribute its location.
  - Similar to other protocols (e.g., DREAM, GLS).

LAR Summary

- Advantages:
  - Reduces scope of route request flood.
  - Reduces overhead of route discovery.
- Disadvantages:
  - Nodes need to know their physical locations.
  - Choice of request zone.

Hybrid Protocols

ZRP [Haas98]

Zone Routing Protocol combines:

- Proactive protocol: which pro-actively updates network state and maintains routes regardless of whether any data traffic exists or not.
- Reactive protocol: which only determines route to a destination if there is some data to be sent to the destination.

ZRP "Hybridness"

- Limits scope of proactive procedure to a node's *local neighborhood*.
- Limits scope of topology changes to local neighborhood.
- Reactive protocol executed for routes to destination far-away.

Routing Zone

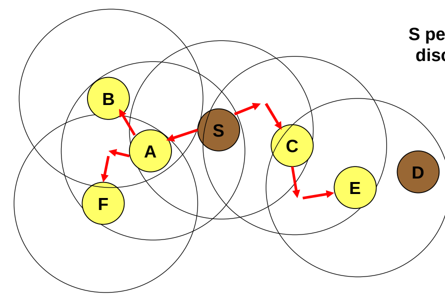
- All nodes within hop distance of at most *d* from node X are said to be in the routing zone (RZ) of X.
- All nodes at hop distance exactly *d* are said to be peripheral nodes of X's routing zone.
- Each node maintains its own RZ.

ZRP

- Intra-zone routing: Pro-actively maintain state information for links within a short distance from any given node.
  - Routes to nodes within short distance are thus maintained proactively (using, say, link state or distance vector protocol).
- Inter-zone routing: Uses reactive protocol for determining routes to far away nodes. Route discovery is similar to DSR with the exception that route requests are propagated via peripheral nodes.

ZRP: Example with Zone Radius *d* = 2

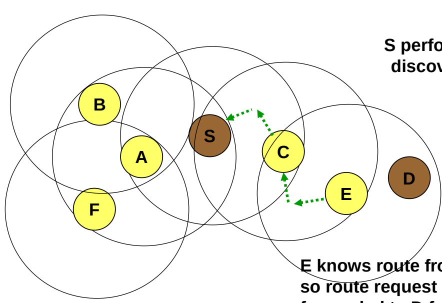
S performs route discovery for D



Denotes route request

ZRP: Example with *d* = 2

S performs route discovery for D

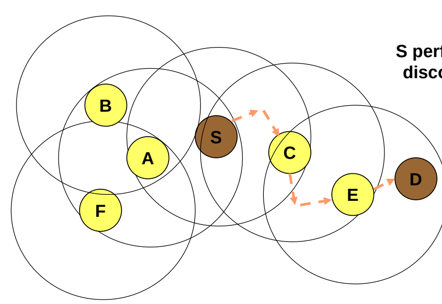


E knows route from E to D, so route request need not be forwarded to D from E

Denotes route reply

ZRP: Example with *d* = 2

S performs route discovery for D

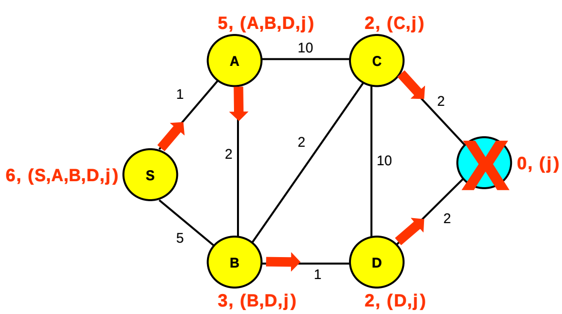


Denotes route taken by Data

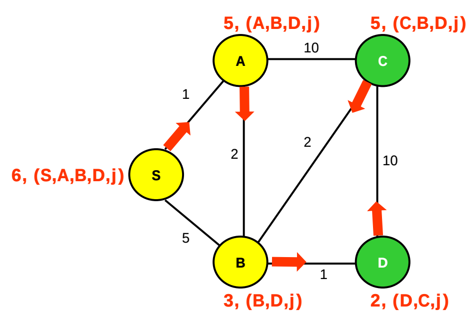
Path Finding: Improved Path Updating

- If neighbor k reports a new path to destination j:
  - Traverse the paths for j reported by the other neighbors.
  - If neighbor n's path to j includes node k, substitute the new path to j reported by k as the subpath from k to j reported by n.

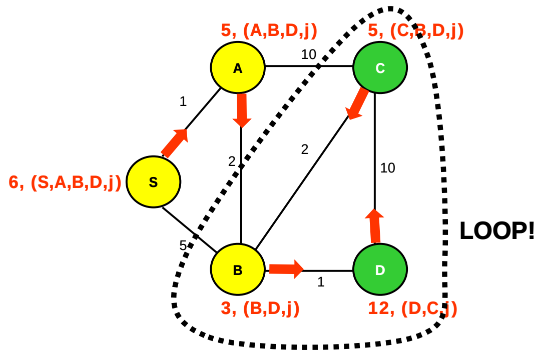
Improved Path Updating



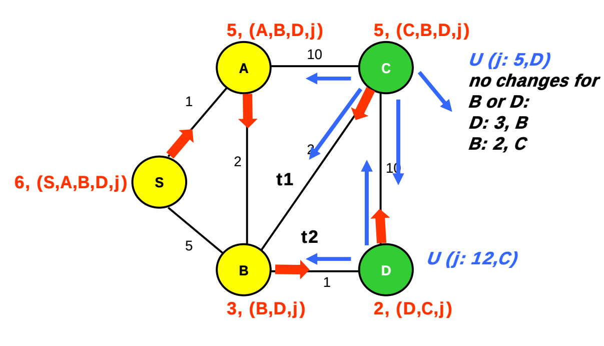
Improved Path Updating



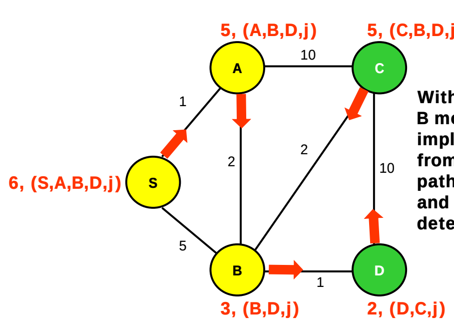
Improved Path Updating



Improved Path Updating



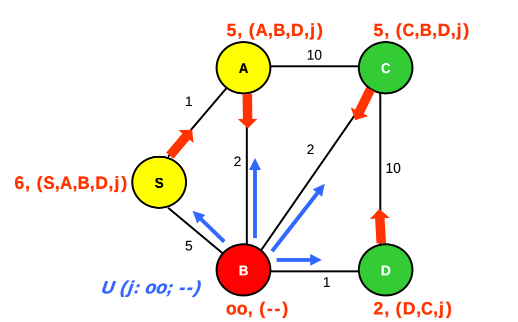
Improved Path Updating



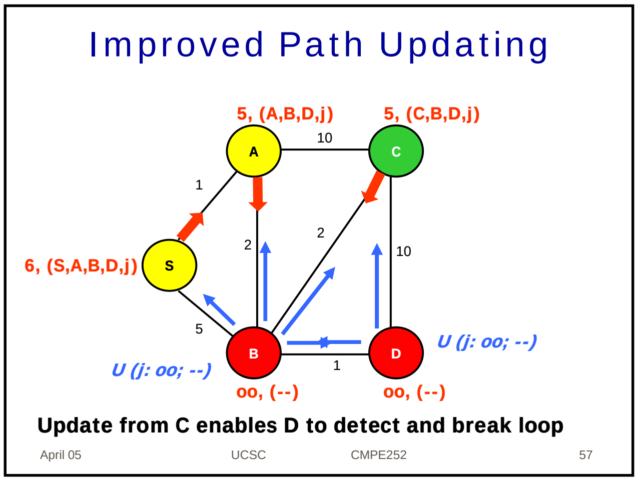
With update from C, B modifies the implicit subpath from C to j in the path from D to j, and a loop is detected!

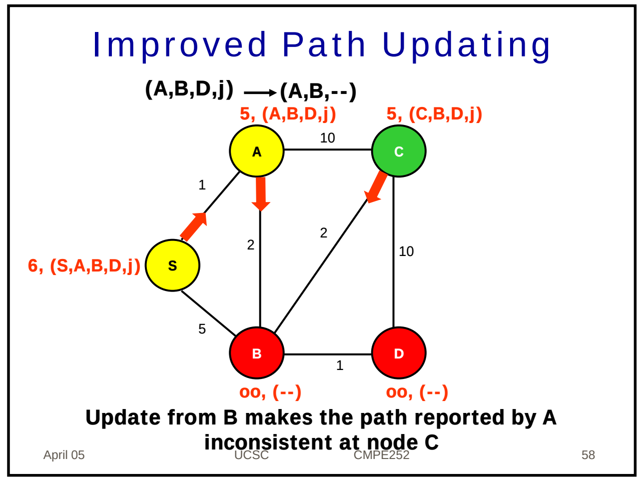
2, (D,C,j) ⟶ (D,C,B,D,j)

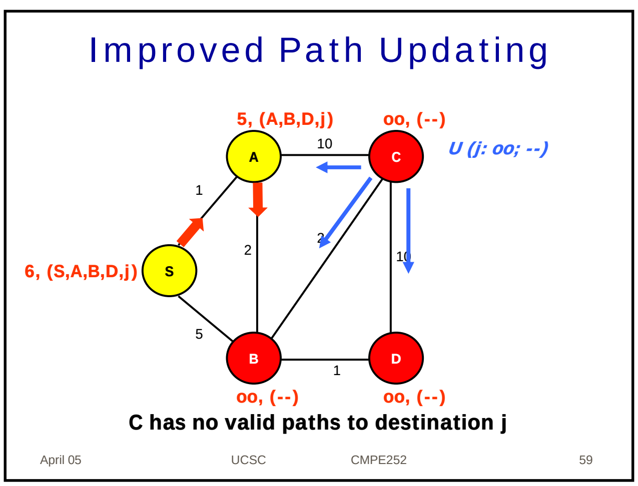
Improved Path Updating

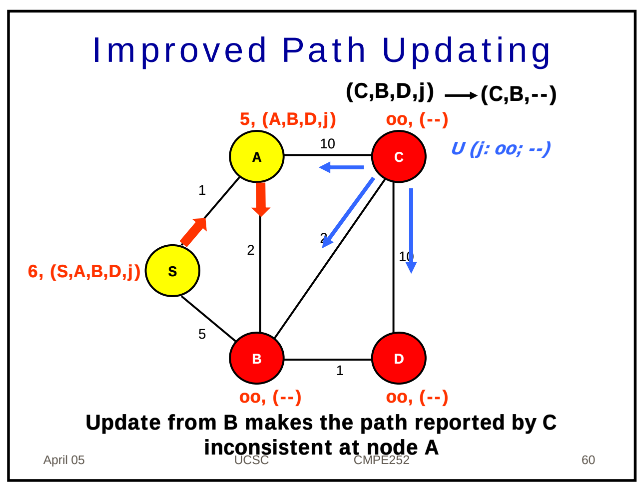


B detects and breaks loop

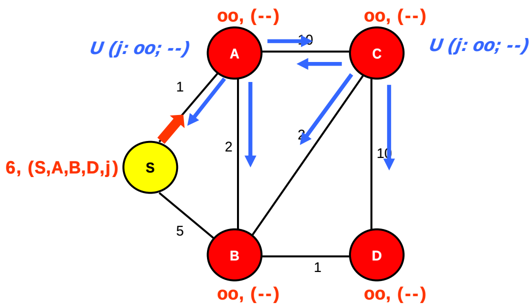

Improved Path Updating
5, (A,B,D,j)
5, (C,B,D,j)
6, (S,A,B,D,j)
A
C
S
B
D
10
1
2
2
10
5
1
U (j: oo; --)
U (j: oo; --)
oo, (--)
oo, (--)
Update from C enables D to detect and break loop
April 05
UCSC
CMPE252
57


Improved Path Updating
(A,B,D,j) → (A,B,--)
5, (A,B,D,j)
5, (C,B,D,j)
6, (S,A,B,D,j)
A
C
S
B
D
10
1
2
2
10
5
1
oo, (--)
oo, (--)
Update from B makes the path reported by A inconsistent at node C
April 05
UCSC
CMPE252
58


Improved Path Updating
5, (A,B,D,j)
oo, (--)
U (j: oo; --)
6, (S,A,B,D,j)
A
C
S
B
D
10
1
2
2
10
5
1
oo, (--)
oo, (--)
C has no valid paths to destination j
April 05
UCSC
CMPE252
59

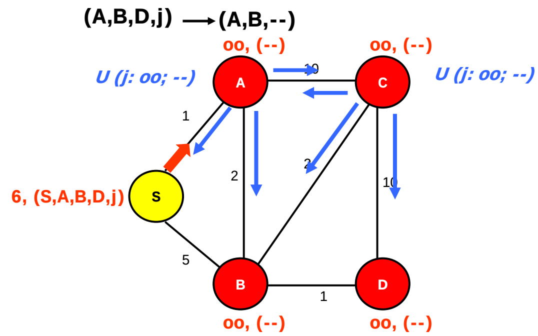

Improved Path Updating
(C,B,D,j) → (C,B,--)
5, (A,B,D,j)
oo, (--)
U (j: oo; --)
6, (S,A,B,D,j)
A
C
S
B
D
10
1
2
2
10
5
1
oo, (--)
oo, (--)
Update from B makes the path reported by C inconsistent at node A
April 05
UCSC
CMPE252
60

Improved Path Updating



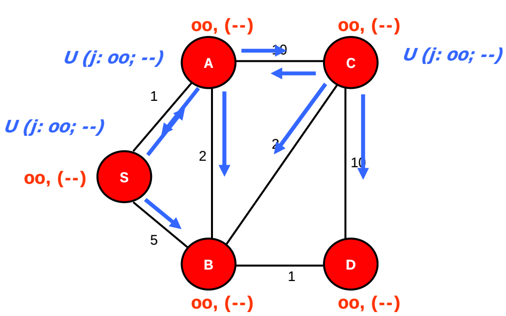
A has no valid paths to destination j

Improved Path Updating



Update from B makes the path reported by A inconsistent at node S

Improved Path Updating



A has no valid paths to destination j

Improved Path Updating

- Temporary loops are still possible.
- Convergence is much faster.
- Fewer update messages are needed, and fewer temporary loops can be expected.
- Sequence numbers can be used to validate paths more safely.