DSDV & AODV

2/27/06

Last class

- Basic classification of ad hoc routing
  - Proactive
  - Reactive, on-demand
  - Geographical routing
  - Hierarchical routing
  - …
- DSR: dynamic source routing
  - Reactive protocol
  - Route discovery phase + maintenance phase.
  - Packet contains the path information.

This class

- DSDV: Destination-Sequenced Distance-Vector
  - Proactive
- AODV: Ad hoc on-demand distance vector routing
  - Reactive
  - Based on DSDV

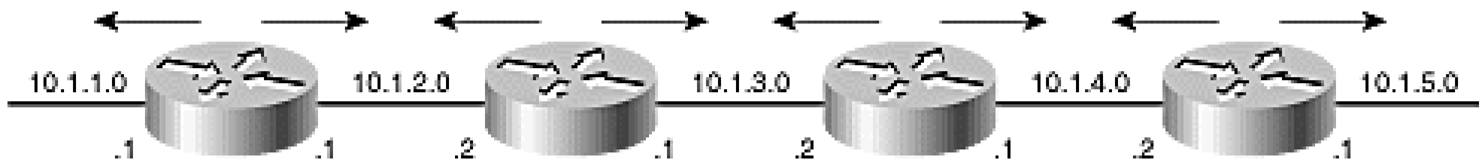
Distance vector routing

- Routing protocol in wired networks.
- Distributed Bellman-Ford algorithm.
  - Each node maintains a hop count for each destination.
  - Nodes periodically send their routing tables to neighbors.
  - Nodes re-calculate shortest path upon the receipt of a routing table update.
- Proactive protocol.
- Shortest path routing.

Distance vector routing

- Routing protocol in wired networks.
  - Continuously update the "reachability" information at all the network nodes
  - Low route request latency and high overhead
- Problems in dynamic environment
  - Changes propagate slowly, slow convergence
  - Create loops
  - Count to infinity

Convergence of distance vector



Router A

	NET	VIA	HOPS
t_0	10.1.1.0	--	0
	10.1.2.0	--	0
t_1	10.1.1.0	--	0
	10.1.2.0	--	0
	10.1.3.0	10.1.2.2	1
t_2	10.1.1.0	--	0
	10.1.2.0	--	0
	10.1.3.0	10.1.2.2	1
	10.1.4.0	10.1.2.2	2
t_3	10.1.1.0	--	0
	10.1.2.0	--	0
	10.1.3.0	10.1.2.2	1
	10.1.4.0	10.1.2.2	2
	10.1.5.0	10.1.2.2	3

Router B

	NET	VIA	HOPS
t_0	10.1.2.0	--	0
	10.1.3.0	--	0
t_1	10.1.2.0	--	0
	10.1.3.0	--	0
	10.1.1.0	10.1.2.1	1
	10.1.4.0	10.1.3.2	1
t_2	10.1.2.0	--	0
	10.1.3.0	--	0
	10.1.1.0	10.1.2.1	1
	10.1.4.0	10.1.3.2	1
	10.1.5.0	10.1.3.2	2
t_3	10.1.2.0	--	0
	10.1.3.0	--	0
	10.1.1.0	10.1.2.1	1
	10.1.4.0	10.1.3.2	1
	10.1.5.0	10.1.3.2	2

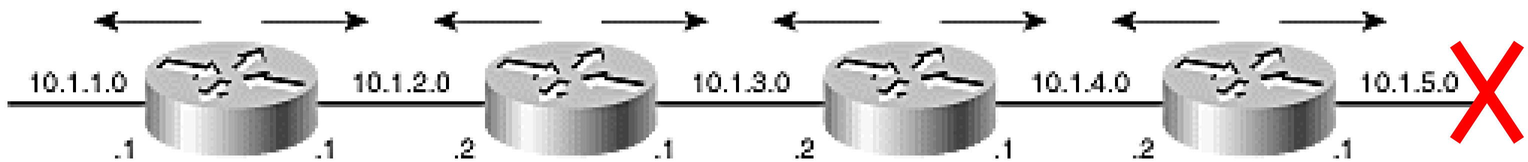
Router C

	NET	VIA	HOPS
t_0	10.1.3.0	--	0
	10.1.4.0	--	0
t_1	10.1.3.0	--	0
	10.1.4.0	--	0
	10.1.2.0	10.1.3.1	1
	10.1.5.0	10.1.4.2	1
t_2	10.1.3.0	--	0
	10.1.4.0	--	0
	10.1.2.0	10.1.3.1	1
	10.1.5.0	10.1.4.2	1
	10.1.1.0	10.1.3.1	2
t_3	10.1.3.0	--	0
	10.1.4.0	--	0
	10.1.2.0	10.1.3.1	1
	10.1.5.0	10.1.4.2	1
	10.1.1.0	10.1.3.1	2

Router D

	NET	VIA	HOPS
t_0	10.1.4.0	--	0
	10.1.5.0	--	0
t_1	10.1.4.0	--	0
	10.1.5.0	--	0
	10.1.3.0	10.1.4.1	1
t_2	10.1.4.0	--	0
	10.1.5.0	--	0
	10.1.3.0	10.1.4.1	1
	10.1.2.0	10.1.4.1	2
t_3	10.1.4.0	--	0
	10.1.5.0	--	0
	10.1.3.0	10.1.4.1	1
	10.1.2.0	10.1.4.1	2
	10.1.1.0	10.1.4.1	3

Convergence of distance vector



t_0

Router A

NET	VIA	HOPS
10.1.1.0	- -	0
10.1.2.0	- -	0

Router B

NET	VIA	HOPS
10.1.2.0	- -	0
10.1.3.0	- -	0

Router C

NET	VIA	HOPS
10.1.3.0	- -	0
10.1.4.0	- -	0

Router D

NET	VIA	HOPS
10.1.4.0	- -	0
10.1.5.0	- -	0

t_1

Router A

NET	VIA	HOPS
10.1.1.0	- -	0
10.1.2.0	- -	0
10.1.3.0	10.1.2.2	1

Router B

NET	VIA	HOPS
10.1.2.0	- -	0
10.1.3.0	- -	0
10.1.1.0	10.1.2.1	1
10.1.4.0	10.1.3.2	1

Router C

NET	VIA	HOPS
10.1.3.0	- -	0
10.1.4.0	- -	0
10.1.2.0	10.1.3.1	1
10.1.5.0	10.1.4.2	1

Router D

NET	VIA	HOPS
10.1.4.0	- -	0
10.1.5.0	- -	0
10.1.3.0	10.1.4.1	1

t_2

Router A

NET	VIA	HOPS
10.1.1.0	- -	0
10.1.2.0	- -	0
10.1.3.0	10.1.2.2	1
10.1.4.0	10.1.2.2	2

Router B

NET	VIA	HOPS
10.1.2.0	- -	0
10.1.3.0	- -	0
10.1.1.0	10.1.2.1	1
10.1.4.0	10.1.3.2	1
10.1.5.0	10.1.3.2	2

Router C

NET	VIA	HOPS
10.1.3.0	- -	0
10.1.4.0	- -	0
10.1.2.0	10.1.3.1	1
10.1.5.0	10.1.4.2	1
10.1.1.0	10.1.3.1	2

Router D

NET	VIA	HOPS
10.1.4.0	- -	0
10.1.5.0	- -	0
10.1.3.0	10.1.4.1	1
10.1.2.0	10.1.4.1	2

t_3

Router A

NET	VIA	HOPS
10.1.1.0	- -	0
10.1.2.0	- -	0
10.1.3.0	10.1.2.2	1
10.1.4.0	10.1.2.2	2
10.1.5.0	10.1.2.2	3

Router B

NET	VIA	HOPS
10.1.2.0	- -	0
10.1.3.0	- -	0
10.1.1.0	10.1.2.1	1
10.1.4.0	10.1.3.2	1
10.1.5.0	10.1.3.2	2

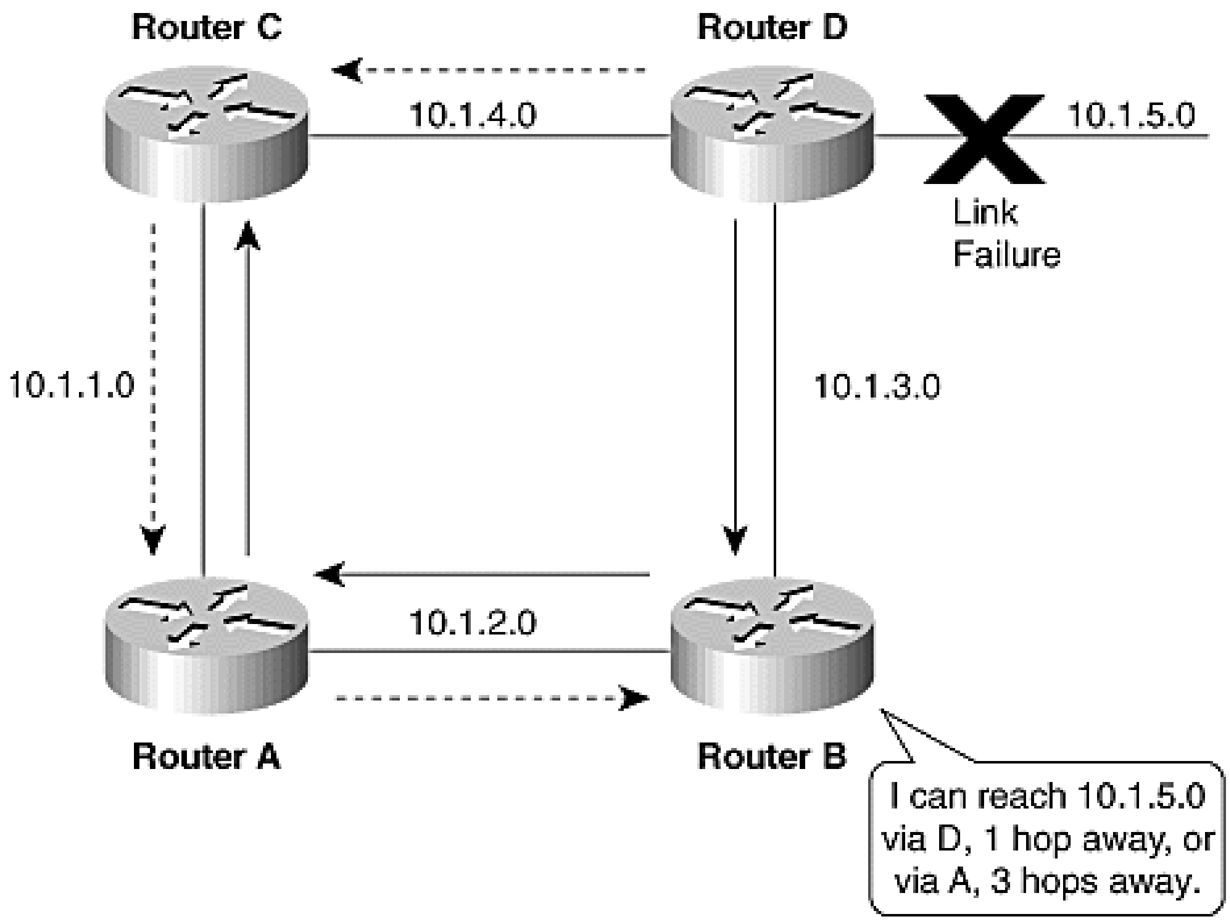
Router C

NET	VIA	HOPS
10.1.3.0	- -	0
10.1.4.0	- -	0
10.1.2.0	10.1.3.1	1
10.1.5.0	10.1.4.2	1
10.1.1.0	10.1.3.1	2

Router D

NET	VIA	HOPS
10.1.4.0	- -	0
10.1.5.0	- -	0
10.1.3.0	10.1.4.1	1
10.1.2.0	10.1.4.1	2
10.1.1.0	10.1.4.1	3

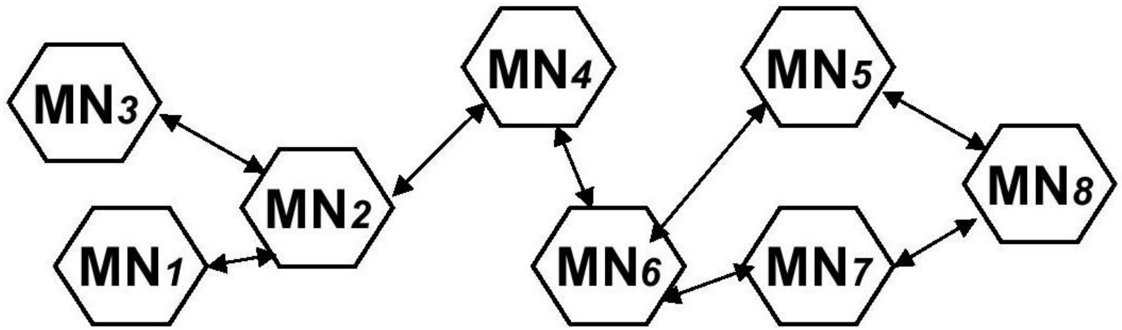
Distance vector, count to infty



DSDV

- DSDV: Destination-Sequenced Distance-Vector
- Adds two things to distance-vector routing
  - Sequence number; avoid loops
  - Damping; hold advertisements for changes of short duration.

Sequence number



Dest	Nexthop	Metric	DestSequence	InstallTime
MN1	MN2	2	406	
MN2	MN2	1	128	
MN3	MN2	2	564	
MN4	MN4	0	710	
MN5	MN6	2	392	
MN6	MN6	1	076	
MN7	MN6	2	128	
MN8	MN6	3	050	

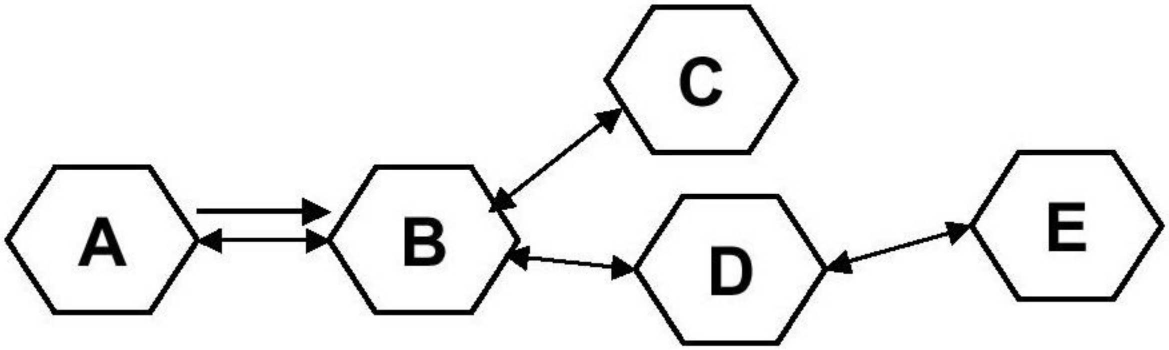
DSDV routing updates

- Each node periodically transmits updates
  - Includes its own sequences number, routing table updates
- Nodes also send routing table updates for important link changes
- When two routes to a destination received from two different neighbors
  - Choose the one with greatest destination sequence number
  - If equal, choose the smaller metric (hop count)

DSDV full dump

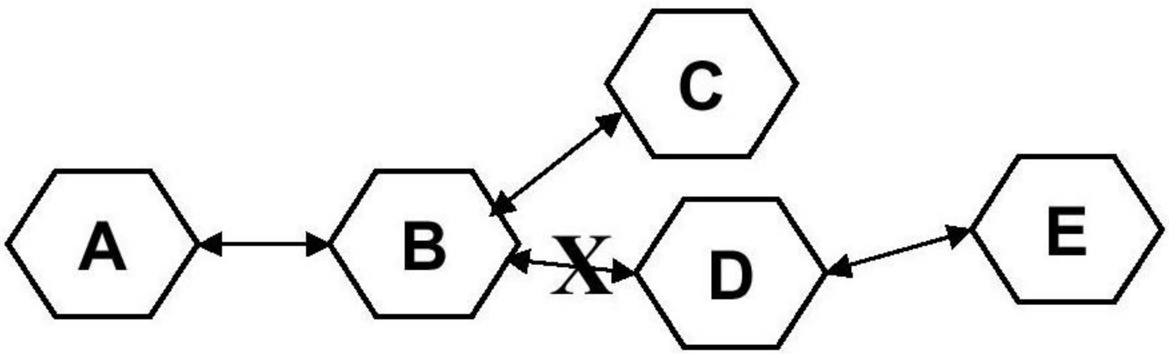
- Full Dumps
  - Carry all routing table information
  - Transmitted relatively infrequently
- Incremental updates
  - Carry only information changed since last full dump
  - Fits within one network protocol data unit
  - If can’t, send full dump

DSDV link addition



- When A joins network
  - Node A transmits routing table: <A, 101, 0>
  - Node B receives transmission, inserts <A, 101, A, 1>
  - Node B propagates new route to neighbors <A, 101, 1>
  - Neighbors update their routing tables: <A, 101, B, 2> and continue propagation of information

DSDV link breaks



- Link between B and D breaks
  - Node B notices break
    - Update hop count for D and E to be infinity
    - Increments sequence number for D and E
  - Node B sends updates with new route information
    - <D, 203, infinite>
    - <E, 156, infinite>

DSDV routing updates

- Each node periodically transmits updates
  - Includes its own sequences number, routing table updates
- Nodes also send routing table updates for important link changes
- When two routes to a destination received from two different neighbors
  - Choose the one with greatest destination sequence number
  - If equal, choose the smaller metric (hop count)

DSDV summary

- Routes maintained through periodic and event triggered routing table exchanges
- Incremental dumps and settling time used to reduce control overhead
- Lower route request latency, but higher overhead
- Perform best in network with low to moderate mobility, few nodes and many data sessions
- Problems:
  - Not efficient for large ad-hoc networks
  - Nodes need to maintain a complete list of routes.

DSDV, DSR

- DSDV performs well under low node mobility
  - High delivery rate
  - Fails to converge for increased mobility
- DSR performs well at all mobility rates
  - Increased overhead of routing tables and control packets
  - Scalability for dense networks

AODV

- DSR includes source routes in packet headers
- Resulting large headers can degrade performance
  - When data content is small
- AODV improves on DSR by maintaining routing tables (reverse paths) at nodes, instead of in data packets.
- AODV retains the desirable feature of DSR that routes are only maintained between communicating nodes.

AODV

- The Ad-hoc On-Demand Distance Vector Algorithm
- Descendant of DSDV
- Reactive
- Route discovery cycle used for route finding
- Maintenance of active routing
- Sequence number used for loop prevention and route freshness criteria
- Provides unicast and multicast communication

Goal of AODV

- Quick adaptation under dynamic link conditions
- Lower transmission latency
- Consume less network bandwidth (less broadcast)
- Loop-free property
- Scalable to large network

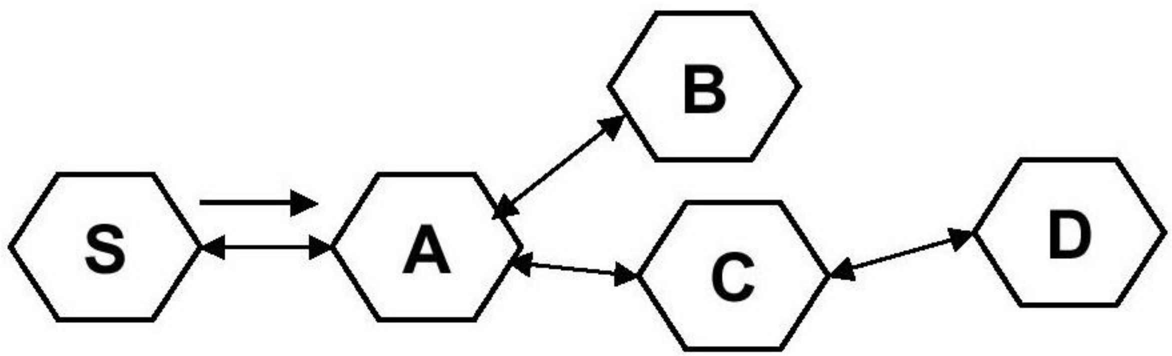
AODV – unicast route discovery

- RREQ (route request) is broadcast
  - Sequence Number:
    - Source SN: freshness on reverse route to source
    - Destination SN: freshness on route to destination
  - RREQ message
    - <bcast_id, dest_ip, dest_seqno, src_seqno, hop_count>
- RREP (route reply) is unicast back
  - From destination if necessary
  - From intermediate node if that node has a recent route

AODV multicast route discovery

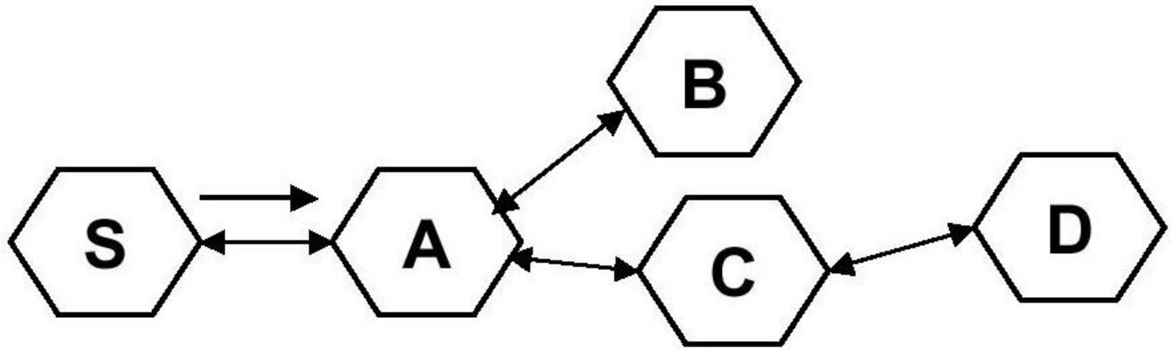
- Message types
  - RREQ, with new flags:
    - Join and Repair
  - RREP
  - MACT (Multicast activation message)
- Multicast routes have destination sequence number and multiple next hops
  - Multicast group leader extension for RREQ and RREP

AODV route discovery



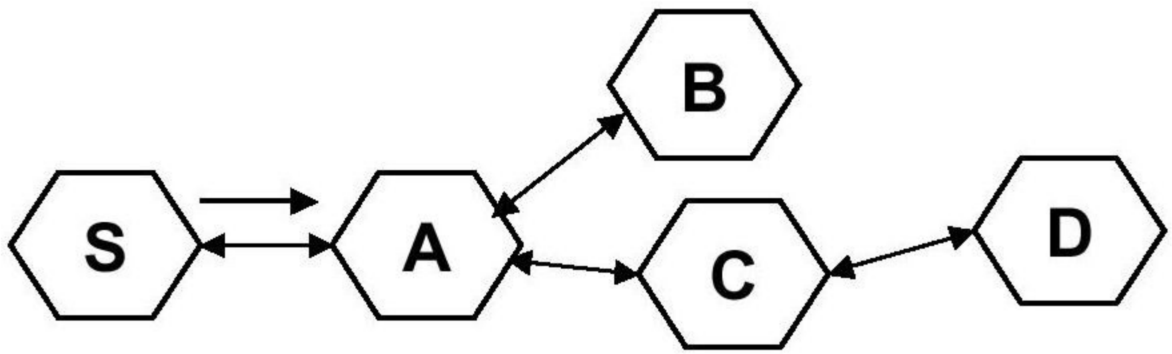
1. Node S needs a route to D
2. Create a route request (RREQ)
   - Enters D's IP address, sequence number, S's IP address, sequence number
   - Broadcasts RREQ to neighbors

AODV route discovery, cont.



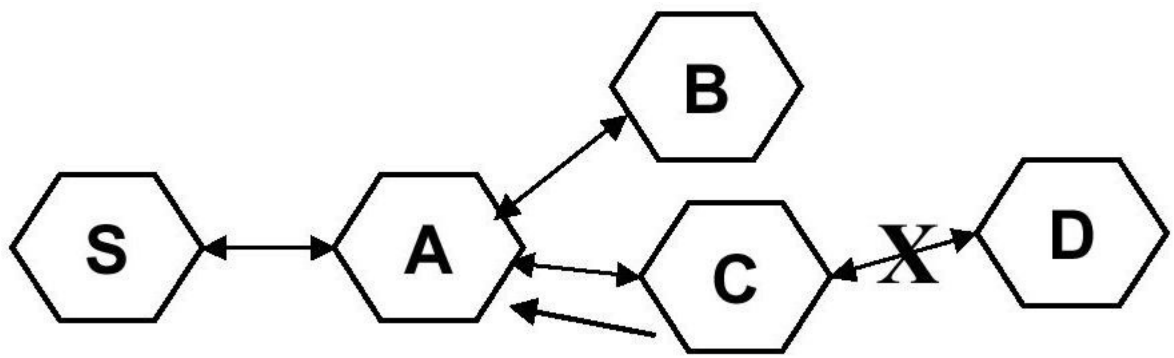
3. Node A receives RREQ
   - Makes reverse route entry for S
     - Dest = S, nexthop = S, hopcount = 1
   - It has no route to D, so it broadcasts RREQ
4. Node C receives RREQ
   - Makes reverse route entry for S
     - Dest = S, nexthop = A, hopcount = 2
   - It has route to D && seq# for route D > seq# in RREQ
     - Creates a route reply (RREP)
       - Enters D's IP address, sequence number, S's IP address, hopcount
     - Unicasts RREP to A

AODV route discovery, cont.



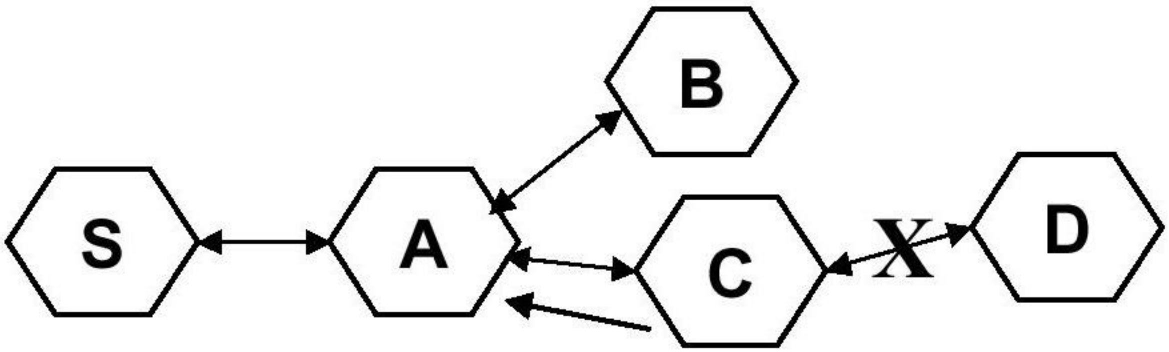
5. Node A receives RREP
   - Unicasts RREP to S
   - Makes forward route entry to D
     - Dest = D, nexthop = C hopcount = 2
6. Node S receives RREP
   - Makes forward route entry to D
     - Dest = D, nexthop = A hopcount = 3
   - Sends data packets on route to D

AODV --- route maintenance (1)



- Link between C and D breaks
  - Node C invalidates route to D in routing table
  - Node C creates route error (RERR) message
    - Lists all destinations with are now unreachable
    - Sends to upstream neighbors
  - Node A receives RERR
    - Checks whether C is its next hop on route to D
    - Deletes route to D, and forwards RERR to S

AODV --- route maintenance (2)



– Node S receives RERR
  - Checks whether A is its next hop on route to D
  - Deletes route to D
  - Rediscovers route if still needed

AODV --- Optimizations

- Expanding ring search
  - Prevents flooding of network during route discovery
  - Control Time to Live of RREQ
- Local repair
  - Repair breaks in active routes locally instead of notifying source
  - Use small TTL because destination probably has not moved far
  - If first repair attempt is unsuccessful, send RERR to source

AODV --- Summary

- Reactive / On-demand
- Sequence numbers used for route freshness and loop prevention
- Route discovery cycle
- Maintains only active routes
- Optimization can be used to reduce overhead and increase scalability