

CMPE 257: Wireless Networking

SET 4:

Unicast Routing in MANETs

Spring 2005

UCSC

CMPE257

1

Ad-Hoc Routing Requirements

- Dissemination of routing information
 - Multi-hop paths
 - Loop free all the time, or almost loop-free
 - Limited signaling overhead.
- Self configuring, and adaptive to dynamic topology.
- Low consumption of communication bandwidth, energy.
 - Scalable with number of nodes.
 - Localized effect of topology or flow change.

Spring 2005

UCSC

CMPE257

2

MANET Unicast Routing

- Many protocols have been proposed.
- Many have been invented specifically for MANETs.
- Many are adapted from protocols for wired networks.
- Can any one protocol work well in all MANET environments?

Spring 2005

UCSC

CMPE257

3

Conventional Wisdom: DV or LS?

- DV protocols:
 - May form loops: wasteful in wireless environment
 - Bandwidth and power.
 - Loop avoidance may be complex
- LS protocols:
 - Higher storage and communication overhead.

Spring 2005

UCSC

CMPE257

4

Reality

- It is all in the distributed algorithms used!

Spring 2005

UCSC

CMPE257

5

Taxonomy

- Proactive protocols:
 - Determine routes independently of traffic patterns.
 - Traditional link-state and distance-vector routing protocols are proactive.
- Reactive protocols:
 - Maintain routes only if needed.
- Hybrid protocols.

Spring 2005

UCSC

CMPE257

6

Why Loop-Free Routing?

- Efficient use of resources
 - Bandwidth, processing, storage, and energy (for un-tethered nodes)
- Application deadlines
- Some degree of confidence that information reaches its destination in a dynamic topology.

Spring 2005

UCSC

CMPE257

7

Trade-Offs

- Latency of route discovery.
 - Proactive protocols may have lower latency since routes are maintained at all times.
 - Reactive protocols may have higher latency because a route from X to Y will be found only when X attempts to send to Y.
- Overhead of route discovery/maintenance.
 - Reactive protocols may have lower overhead because routes are determined only if needed.
 - Proactive protocols may result in higher overhead due to continuous route updating (depends on rate of changes).
- Which approach achieves a better trade-off depends on the traffic and mobility patterns.

Spring 2005

UCSC

CMPE257

8

On-Demand Approaches

Spring 2005

UCSC

CMPE257

9

Well-known On-demand Routing Techniques

■ Nodal Synchronization

- TORA and others.
 - Require synchronization over multiple hops.
 - Based on "link-reversal" algorithms first proposed by Gafni and Bertsekas.
 - TORA requires synchronized clocks.

■ Source Routing

- Dynamic Source Routing (DSR) protocol and others.
 - Complete routes need to be specified in every data packet.

■ Destination sequence numbers

- Ad hoc On-demand Distance Vector (AODV) and others.
 - Require a sequence number reset from the destination.

Spring 2005

UCSC

CMPE257

10

Assumptions in On-demand Routing?

Spring 2005

UCSC

CMPE257

11

Components of On-demand Routing?

Spring 2005

UCSC

CMPE257

12

Limitations of Coordination over Many Hops

- Consider DUAL (what Cisco uses in EIGRP):
 - Excessive coordination among nodes to keep distances ordered.



Spring 2005

UCSC

CMPE257

13

Dynamic Source Routing (DSR) [Johnson96]

- When node S wants to send a packet to D, and does not have a route to D, node S initiates a **route discovery**.
- S floods **Route Request (RREQ)**.
- Each node **appends own identifier** when forwarding RREQ.

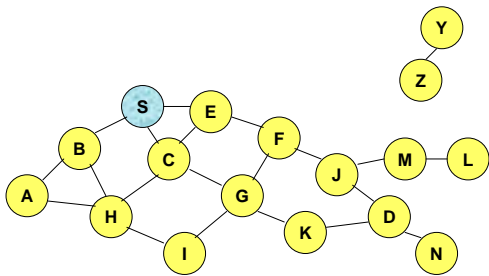
Spring 2005

UCSC

CMPE257

14

Route Discovery in DSR



Represents a node that has received RREQ for D from S

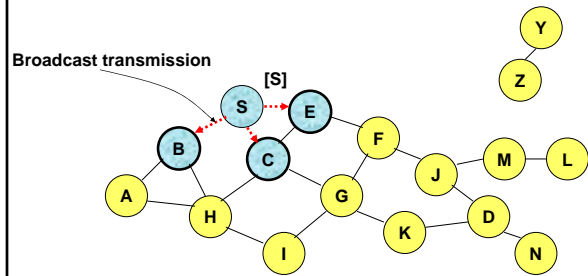
Spring 2005

UCSC

CMPE257

15

Route Discovery in DSR



.....→ Represents transmission of RREQ

[X,Y] Represents list of identifiers appended to RREQ

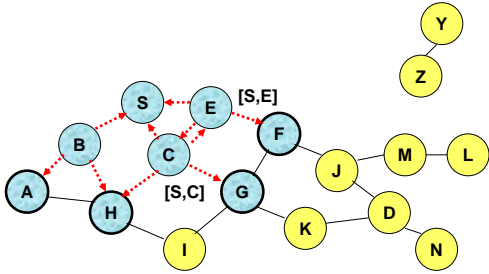
Spring 2005

UCSC

CMPE257

16

Route Discovery in DSR



- Node H receives packet RREQ from two neighbors: **potential for collision**

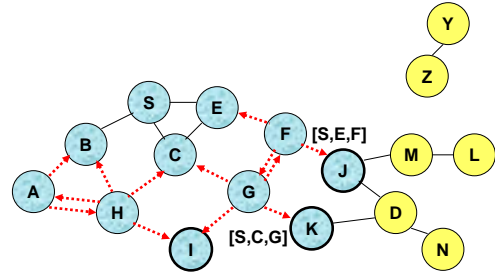
Spring 2005

UCSC

CMPE257

17

Route Discovery in DSR



- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ** once

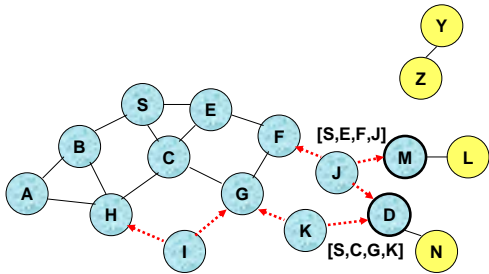
Spring 2005

UCSC

CMPE257

18

Route Discovery in DSR



- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are **hidden** from each other, their **transmissions may collide**

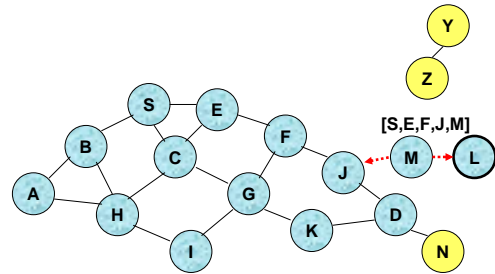
Spring 2005

UCSC

CMPE257

19

Route Discovery in DSR



- Node D **does not forward RREQ**, because node D is the **intended target** of the route discovery

Spring 2005

UCSC

CMPE257

20

Route Discovery in DSR

- Destination D on receiving the first RREQ, sends a **Route Reply (RREP)**.
- RREP is sent on a route obtained by **reversing** the route appended to received RREQ.
- RREP **includes the route** from S to D on which RREQ was received by node D.

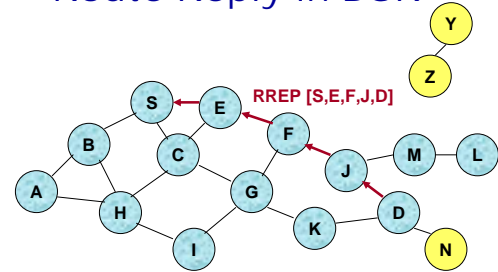
Spring 2005

UCSC

CMPE257

21

Route Reply in DSR



← Represents RREP control message

Spring 2005

UCSC

CMPE257

22

Route Reply in DSR

- RREP can be sent by reversing the route in RREQ only if links are guaranteed to be bi-directional
- If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from D.
 - Unless D already knows a route to S.
 - If a route discovery is initiated by D for a route to S, then the RREP is piggybacked on D's RREQ.

Spring 2005

UCSC

CMPE257

23

Processing RREP

- Node S on receiving RREP, caches the route.
- When node S sends a data packet to D, the entire route is included in the packet header
 - Hence the name **source routing**.
- Intermediate nodes use the **source route** included in a packet to determine to whom a packet should be forwarded.

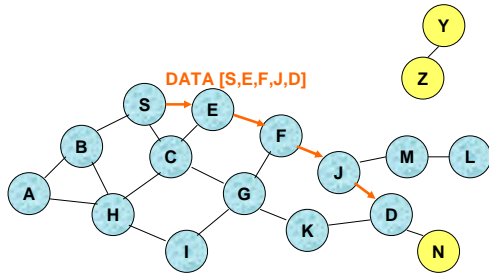
Spring 2005

UCSC

CMPE257

24

Data Delivery in DSR



Packet header size grows with route length

Spring 2005

UCSC

CMPE257

25

DSR Optimization: Route Caching

- Each node caches a new route it learns by *any means*.
 - When node S finds route [S,E,F,J,D] to node D, node S also learns route [S,E,F] to node F.
 - When node K receives Route Request [S,C,G], K learns route [K,G,C,S] to S.
 - When node F forwards Route Reply RREP [S,E,F,J,D], F learns route [F,J,D] to D.
 - When node E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to node D
 - Nodes may also learn route when it overhears data.

Spring 2005

UCSC

CMPE257

26

Use of Route Caching

- When S learns that a route to D is broken, it uses another route from its local cache, if such a route to D exists in its cache; otherwise, S initiates route discovery.
- Node X on receiving a RREQ for some node D can send a RREP if X knows a route to D.
- Use of route cache
 - Can speed up route discovery.
 - Can reduce propagation of route requests.

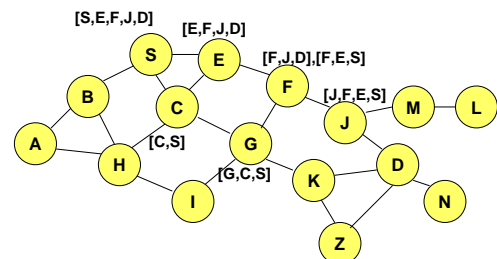
Spring 2005

UCSC

CMPE257

27

Use of Route Caching



[P,Q,R]: Represents cached route at a node

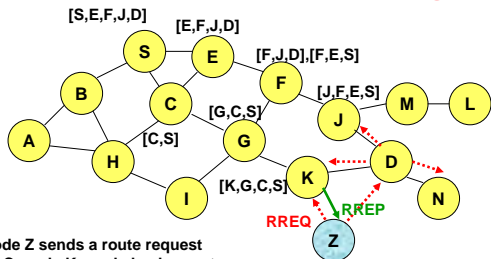
Spring 2005

UCSC

CMPE257

28

Route Caching: Speed up Route Discovery, Reduce RREQ Flooding



When node Z sends a route request for node C, node K sends back a route reply [Z,K,G,C] to node Z using a locally cached route

Route caches at K and J limit the flooding of Z's RREQ.

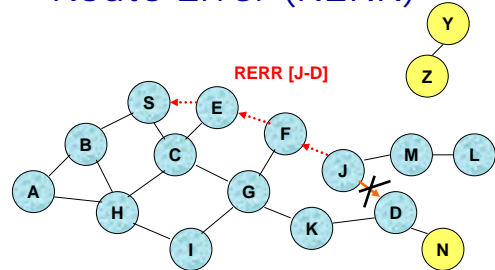
Spring 2005

UCSC

CMPE257

29

Route Error (RERR)



J sends a route error to S along route J-F-E-S when its attempt to forward the data packet S (with route SEFJD) on J-D fails. Nodes hearing RERR update their route cache to remove link J-D

Spring 2005

UCSC

CMPE257

30

Route Caching: Beware!

- Stale caches can adversely affect performance.
 - With time and host mobility, cached routes may become invalid.
 - A sender host may try several stale routes (obtained from local cache, or replied from cache by other nodes), before finding a good route.

Spring 2005

UCSC

CMPE257

31

DSR: Advantages

- Routes maintained only between nodes who need to communicate.
 - Reduces overhead of route maintenance.
- Route caching can further reduce route discovery overhead.
- Single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches.

Spring 2005

UCSC

CMPE257

32

DSR: Disadvantages

- Packet header size grows with route length.
- Flood of route requests may potentially reach all nodes in the network.
 - Care must be taken to avoid collisions between route requests propagated by neighboring nodes.
 - Insertion of random delays before forwarding RREQ.

Spring 2005

UCSC

CMPE257

33

DSR: Disadvantages

- Increased contention if too many route replies come back due to nodes replying using their local cache.
 - "RREP" storm problem.
 - Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route.

Spring 2005

UCSC

CMPE257

34

DSR: Disadvantages

- An intermediate node may send RREP using a stale cached route, thus polluting other caches.
 - This problem can be eased if some mechanism to purge (potentially) invalid cached routes is incorporated.
 - Static timeouts.
 - Adaptive timeouts based on link stability.

Spring 2005

UCSC

CMPE257

35

DSR: key Disadvantages

- Source routes are too brittle in large nets with dynamic topologies!
- Repairing source routes is difficult to do at intermediate nodes w/o causing loops.

Spring 2005

UCSC

CMPE257

36

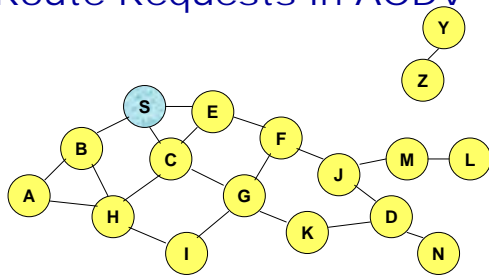
Routing Using Destination Sequence Numbers

- Goal is to avoid inter-nodal synchronization and source-routed data packets.
- Approach derives from the flooding of link-state information.
- First instance was DSDV (proactive), which amounts to flooding of distances with higher sequence numbers.

AODV

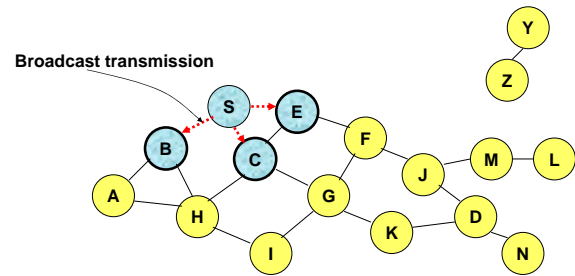
- Route Requests (RREQ) are forwarded similarly to DSR.
 - When a node re-broadcasts a RREQ, it sets up a reverse path pointing towards the source.
 - AODV assumes symmetric (bi-directional) links.
- When the intended destination receives a RREQ, it replies by sending a RREP.
- RREPs travel along the reverse path set-up when RREQ is forwarded.

Route Requests in AODV



Represents a node that has received RREQ for D from S

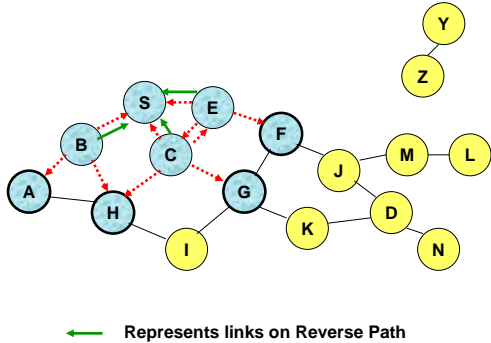
Route Requests in AODV



Broadcast transmission

.....> Represents transmission of RREQ

Route Requests in AODV



Spring 2005

UCSC

CMPE257

41

AODV Route Discovery: Observations

- RREQ contains source and destination IP address, **current destination seq. number** (incremented as a result of loss of prior route), and **broadcast id** (incremented for every RREQ).
 - Source IP + bcast id uniquely identifies RREQ: nodes do not forward RREQs they have forwarded recently.
 - RREQ processing: node creates *reverse* route table entry for RREQ source with TTL.
 - If node has “unexpired” route to destination in its table with **sequence number \geq RREQ's**, it replies to RREQ with **Route Reply (RREP)** back to source.
 - Otherwise, broadcast RREQ onward.

Spring 2005

UCSC

CMPE257

42

Destination Sequence Number

- When node D receives route request with destination sequence number N, D sets its sequence number to N, unless it is already larger than N.
- Node's own sequence number is monotonically increasing.
 - Sequence number is incremented after neighborhood topology change.

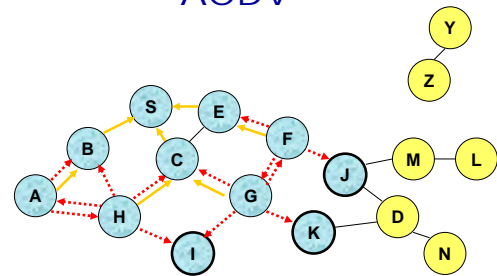
Spring 2005

UCSC

CMPE257

43

Reverse Path Setup in AODV



- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ once**

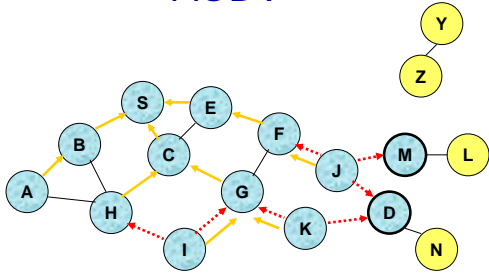
Spring 2005

UCSC

CMPE257

44

Reverse Path Setup in AODV



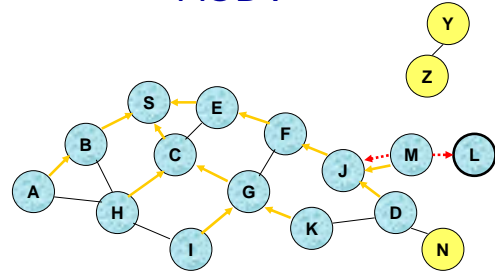
Spring 2005

UCSC

CMPE257

45

Reverse Path Setup in AODV



- Node D does not forward RREQ, because node D is the intended target of the RREQ

Spring 2005

UCSC

CMPE257

46

Route Reply in AODV

- An intermediate node has current route to destination, responds to RREQ with RREP.
- RREP contains source and destination IP, current sequence number, number of hops to destination.
 - If destination, then destination seq. #.
 - Else, node's current record of destination's seq. #.
- Node receiving RREP sets up *forward* path to destination.
- If multiple RREPs received, node forwards first one. Later RREPs discarded unless greater seq. # or smaller # of hops.

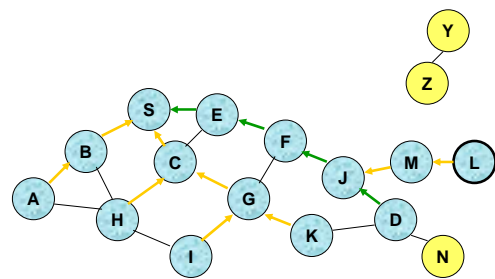
Spring 2005

UCSC

CMPE257

47

Route Reply Example



← Represents links on path taken by RREP

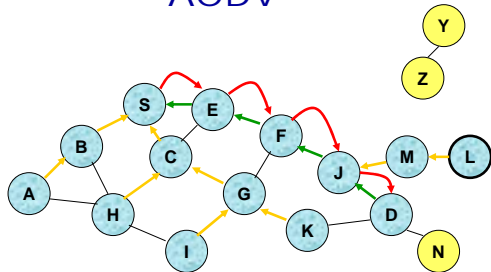
Spring 2005

UCSC

CMPE257

48

Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

Represents a link on the forward path

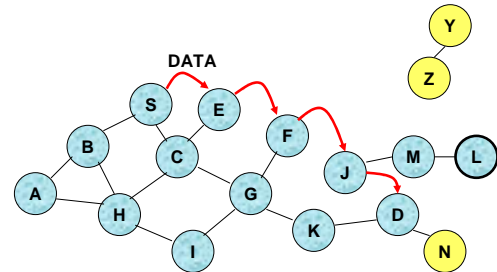
Spring 2005

UCSC

CMPE257

49

Data Delivery in AODV



Routing table entries used to forward data packet.
Route is *not* included in packet header.

Spring 2005

UCSC

CMPE257

50

Timeouts

- A routing table entry maintaining a **reverse path** is purged after a timeout interval.
 - Timeout should be long enough to allow RREP to come back.
- Routing table entry maintaining a **forward path** is purged if *not used* for **active_route_timeout** interval.
 - If no data being sent using a particular routing table entry, that entry will be deleted from the routing table (even if the route may actually still be valid).

Spring 2005

UCSC

CMPE257

51

Link Failure Reporting

- Link failures are propagated by means of Route Error messages, which also update destination sequence numbers.
 - RERR lists destinations now unreachable.
- If upstream node has neighbors as precursors for the affected destinations, it broadcasts RERR.
- Nodes receiving the RERR update cost to destination to infinity and forward RERR if needed.
- Upon receiving RERR, source will initiate route discovery if still needs route.

Spring 2005

UCSC

CMPE257

52

Route Error

- When node X is unable to forward packet P (from node S to node D) on link (X,Y), it generates a RERR message.
- Node X increments the destination sequence number for D cached at node X.
- The incremented sequence number N is included in the RERR.
- When node S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as N .

Spring 2005

UCSC

CMPE257

53

Link Failure Detection

- *Hello* messages: neighbor nodes periodically exchange hello messages.
- Absence of hello message is used as an indication of link failure.
- Alternatively, failure to receive several MAC-level ACKs may be used as an indication of link failure.

Spring 2005

UCSC

CMPE257

54

Sequence Numbers in AODV

- To avoid using old/broken routes.
 - To determine which route is newer.

Spring 2005

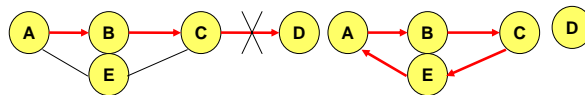
UCSC

CMPE257

55

Sequence Numbers in AODV

- To prevent formation of loops



- Assume that A does not know about failure of link C-D because RERR sent by C is lost
- Now C performs a route discovery for D. A receives the RREQ (say, via path C-E-A)
- A will reply since it knows a route to D via B.
- Results in a loop (for instance, C-E-A-B-C)

Spring 2005

UCSC

CMPE257

56

Optimization: Expanding Ring Search

- RREQs are initially sent with small Time-to-Live (TTL) field, to limit their propagation.
 - DSR also includes a similar optimization.
- If no RREP is received, then larger TTL tried.

Spring 2005

UCSC

CMPE257

57

Does The Sequence Numbering Work?

- To some extent:
 - Sequence numbering scheme is not very efficient
 - Scheme requires that any given node A either never forgets a destination sequence number it learns, or is able to wait "long enough" so that it cannot possibly attempt to reach a destination D through a path involving a node B that uses A to reach D.

Spring 2005

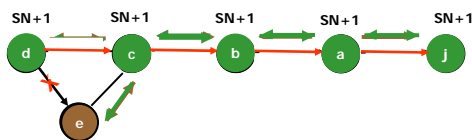
UCSC

CMPE257

58

Problems with AODV's Sequence Numbering

- For valid routes, distances are non-increasing moving away from destination, but a path may have equal SN's along it.
- Invalidating upstream nodes also invalidates downstream nodes.
- This results in too many route requests answered by the destinations!



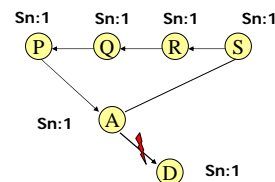
Spring 2005

SN+1 UCSC

CMPE257

59

Counting-to-Infinity in AODV



- t1 – Link failure
- t2 – RERR and route invalidation
- t3 – RREQ (SQ:2)
- t4 – SN deleted
- t5 – RREQ(null)
- t6 – RREP(1)

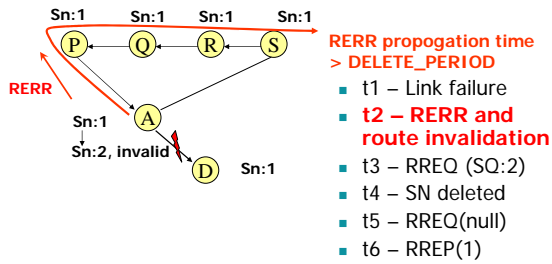
Spring 2005

UCSC

CMPE257

60

Counting-to-Infinity in AODV



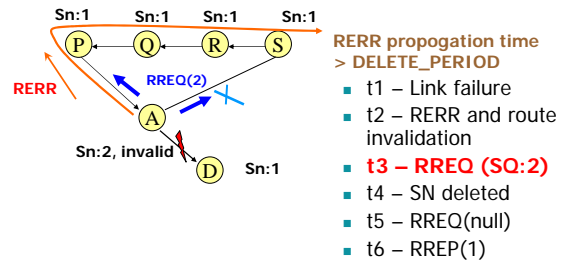
Spring 2005

UCSC

CMPE257

61

Counting-to-Infinity in AODV



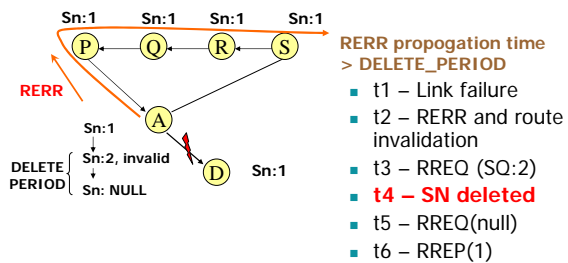
Spring 2005

UCSC

CMPE257

62

Counting-to-Infinity in AODV



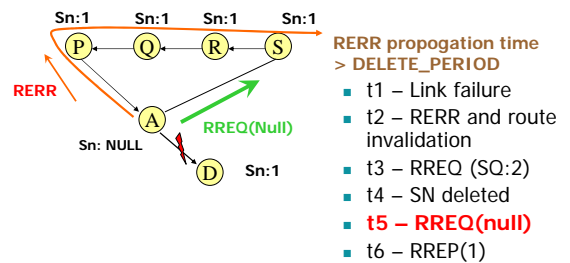
Spring 2005

UCSC

CMPE257

63

Counting-to-Infinity in AODV



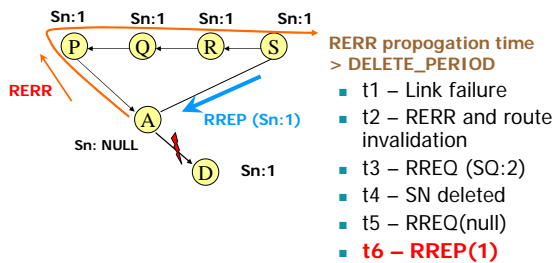
Spring 2005

UCSC

CMPE257

64

Counting-to-Infinity in AODV



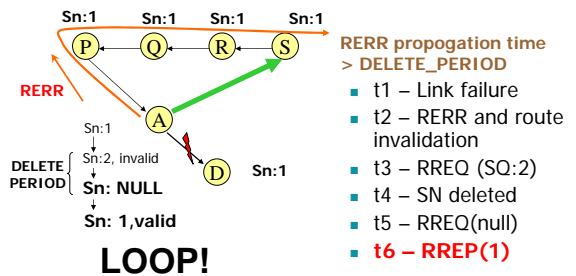
Spring 2005

UCSC

CMPE257

65

Counting-to-Infinity in AODV



Spring 2005

UCSC

CMPE257

66

Problems with Current Use of Path Information

Consider DSR:

- Path information is used for source routed data packets.
- To recover from route failures:
 - Relay node drops packet and sends a route error.
 - Route error propagates to source.
 - Source must re-initiate route discovery.
- Localized route recovery (packet salvaging)
 - Can form loops (source routes erased).

Spring 2005

UCSC

CMPE257

67

Motivation for Using Path Information

- Avoids the problems associated with sequence number wrap-around.
- Provides rich information about the network topology.
- No on-demand protocol based on path information features hop-by-hop loop-free routing of data packets based on their destination address.

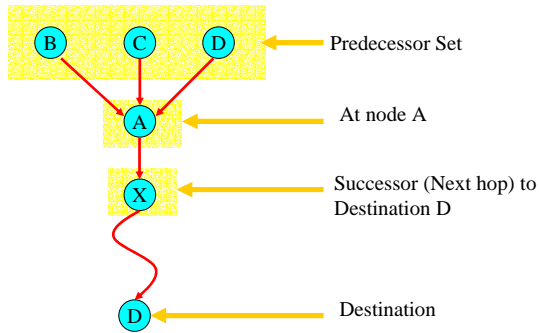
Spring 2005

UCSC

CMPE257

68

Terminology



Spring 2005

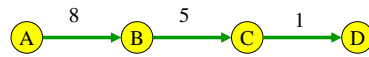
UCSC

CMPE257

69

Terminology for Labels

- Label: An ordered sequence of unique elements of size n . $L = \{e_1(L), e_2(L), \dots, e_n(L)\}$.
- Each element $e(L)$ consists of a pair $\{ID, cost\}$.
 - ID is a valid identifier for a node in the network.
 - Cost is the associated link cost.
- Derived from path information.



Label for path ABCD = $\{(A,8), (B,5), (C,1), (D,0)\}$

Weight of label = $\{8 + 5 + 1 + 0\} = 14$,
Number of elements = 4.

Spring 2005

UCSC

CMPE257

70

Labels – Mathematical Relations

- Two labels (L_1 and L_2) can be compared)
 - $L_1 = L_2$ if
 - Weights are equal, and
 - Elements are the same.
 - $L_1 > L_2$ if
 - Weight (L_1) > Weight(L_2), or
 - Weights equal, and number of elements of $L_1 > L_2$, or
 - Weights equal, number of elements equal, and L_1 is lexicographically greater than L_2 .
 - $L_2 < L_1$ is defined similarly.

Spring 2005

UCSC

CMPE257

71

Loop-Free Conditions Using Labels

- Node A can make node B its successor for destination D after processing an input event if
 - Feasible Label Condition (FLC) : $L_{DB}^A < L_D^{*A}$
 - Extended Label Condition (ELC) : $(A, c_B^A) \oplus L_{DB}^A < L_D^{*A}$
 - Next-hop Label Condition (NLC) : $L_{DB}^A < L_{DS}^{*A}$

L_{DB}^A – Label reported by node B for destination D to node A.

L_D^{*A}, L_{DS}^{*A} – Minimum values attained by L_D^A, L_{DS}^A .

L_D^A – Label stored at node A for destination D.

Spring 2005

UCSC

CMPE257

72

Feasible Label Condition

- FLC extrapolates the properties of feasible distances used in Source Node Condition (SNC) from DUAL.
- FLC allows nodes to switch to neighbors that advertise smaller labels.
 - A state is reached where no neighbors satisfy FLC, although physical paths exist to destination.
 - A mechanism for re-labeling without creating loops is required. Ex. Diffusing Update Algorithm [DUAL], Link Reversal used by TORA.

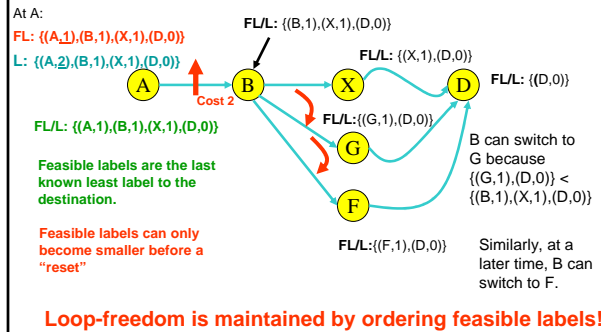
Spring 2005

UCSC

CMPE257

73

Feasible Labels



Spring 2005

UCSC

CMPE257

74

Feasible Label Routing (FLR) protocol

- FLR is an on-demand routing protocol.
 - Uses the usual RREQ, RREP, RERR control messaging framework.
- Loop-freedom is attained by ordering labels so that they become progressively "smaller" moving towards the destination.
- Can ensure instantaneous loop-freedom if reliable route error updates can be sent.
- Uses packet filtering to break temporary loops if unreliable route errors are used.

Spring 2005

UCSC

CMPE257

75

Stored Information

- Routing table entry at node A for destination D contains
 - Successor (Next hop): s_D^A
 - Predecessor set: PS_D^A
 - Current path label: L_D^A
 - Feasible label: FL_D^A
- Nodes maintain no additional information about route discovery except the RREQ (src, reqid) to process unique requests.

Spring 2005

UCSC

CMPE257

76

Control Messages

- Route request (RREQ):

$\{dst, src, reqid, MFL_{dst}^{req}, path_{dst}^{req}\}$

MFL_{dst}^{req} – Minimum feasible label along the set of nodes which relayed the RREQ.

- Route reply (RREP):

$\{dst, src, L_{dst}^{rep}, ttl, path_{dst}^{rep}\}$

L_{dst}^{rep} – Current label at the node transmitting the RREP.

ttl – Lifetime of the route at the node which relayed the RREP.

- Route error (RERR):

$\{orig, reset\}$

orig – The source of the route error.

reset – The list of destinations which are no longer reachable.

$path_{dst}^{req}, path_{dst}^{rep}$ – The path (list of nodes) traversed by the RREQ and RREP, respectively.

src – Source initiating the route request.

dst – Destination to which route is requested.

Spring 2005

UCSC

CMPE257

77

FLR Operation

Based on four rules:

- Processing RREPs:** Accept Label Condition (ALC).
 - When can nodes safely update their routing tables by processing a reply and not cause any loop?
- Relaying RREQs and RREPs:** Minimum Label Condition (MLC).
 - How to find a feasible path extending across multiple hops?
- Responding to a RREQ:** Start Label Condition (SLC).
 - Which nodes can respond to a route request and create loop-free paths to the destination?
- Increasing feasible labels:** Reset Label Condition (RLC).
 - When can nodes safely increase their feasible label?

Spring 2005

UCSC

CMPE257

78

Minimum Label Condition (RREQ)

Node A relays a RREQ only if

- Node A has not processed the RREQ (src, reqid) pair.
- Node A is not in the path traversed by the RREQ.
- Node A additionally sets MFL in the RREQ to the minimum of its feasible label and the received label.

$$MFL_D^{req} = \min(MFL_D^{req}, FL_A)$$

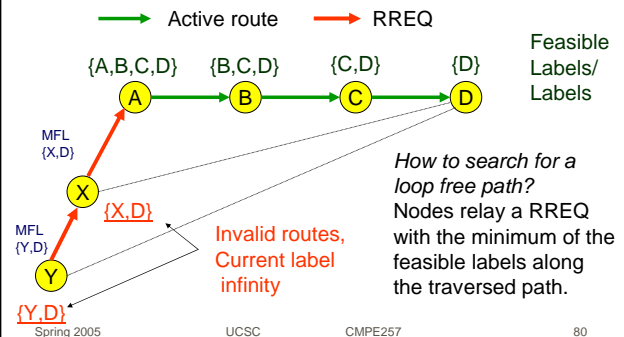
Spring 2005

UCSC

CMPE257

79

Minimum Label Condition (RREQ) - Illustration



Spring 2005

UCSC

CMPE257

80

Start Label Condition (SLC)

- Node I can issue a RREP responding to a RREQ for D if
 - Node I has an active route, and
 - Node I's current label is smaller than the minimum feasible label carried in the RREQ.

$$L_D^I < MFL_D^{req}$$

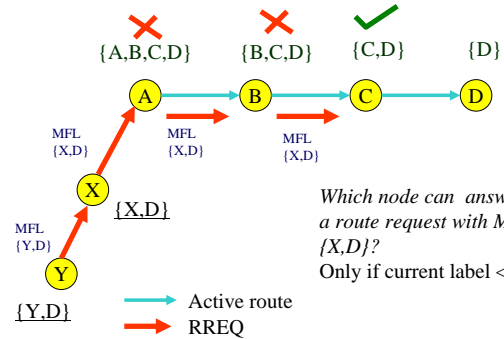
Spring 2005

UCSC

CMPE257

81

Start Label Condition (SLC)



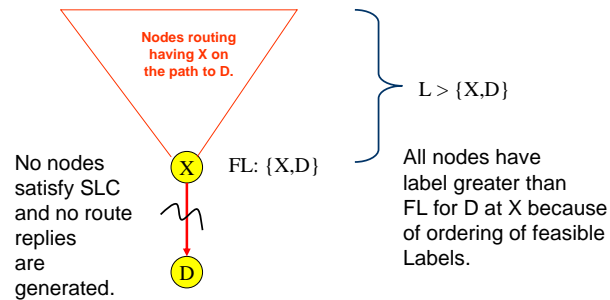
Spring 2005

UCSC

CMPE257

82

SLC - Avoids Replies from Upstream Nodes



Spring 2005

UCSC

CMPE257

83

Minimum Label Condition (RREP)

- Relaying RREPs
 - Node A relays a RREP for destination D, it sets the label in the RREP to its current label.

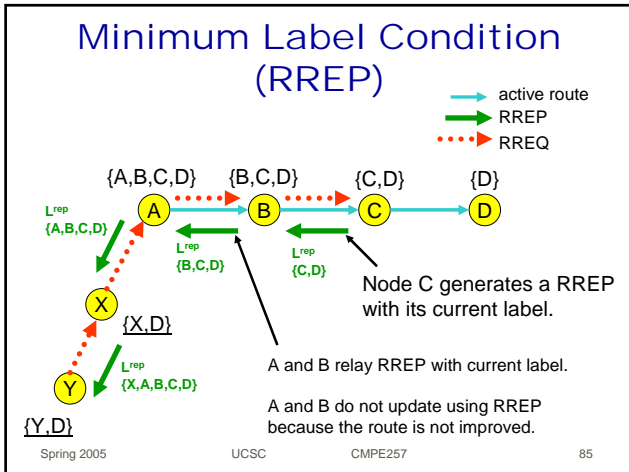
$$L_D^A = L_D^{rep}$$

Spring 2005

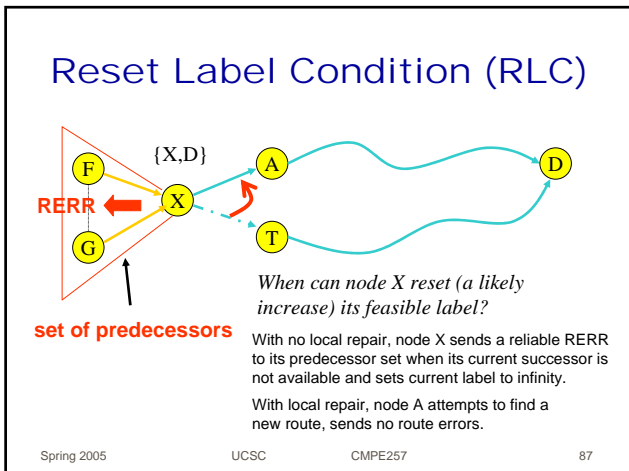
UCSC

CMPE257

84



- ### Reset Label Condition (RLC)
- When a node must change its successor, it sets its current label to invalid.
 - When not local repairing, sends a RERR reliably to all predecessors.
 - When local repairing, sends a new RREQ with its current feasible label.
- Spring 2005 UCSC CMPE257 86



- ### Accept Label Condition (ALC)
- When node A receives a RREP from node B for destination D:
- **With no local repair**, switch to B if $L_D^{rep} < FL_D^A$
 - Condition 1 : Current label is invalid.
 - Condition 2 : Current label is greater than the new path through B and feasible label at node A is greater than the label carried in the RREP.
 - **With local repair**, switch to B if $L_D^{rep} \geq FL_D^A$
 - Condition 2, in addition, sends a reliable route error to the set of predecessor nodes if label in the reply is greater than or equal feasible label at node A.
- Spring 2005 UCSC CMPE257 88

Performance

- Simulation scenarios
 - 50-node on 1500x300m topology
 - 100-node on 2200x600m topology
 - Qualnet 3.5.2, 802.11, random-waypoint
 - 512-byte CBR, 10-flows at 10pps, 30-flows at 4pps
 - Flow length mean 100s, exp. variate length
 - FLR against DSR,AODV, and OLSR.

Spring 2005

UCSC

CMPE257

93

Performance Summary

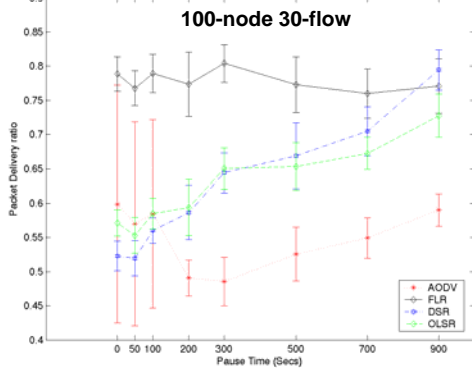
Table 2: Performance average over all pause times for 50 nodes network

Protocol	Flows	Delivery Ratio	Latency (sec)	Net Load	Data Hops
FLR	10	0.9701±0.0108	0.0858±0.0303	0.1872±0.0557	2.5371±0.1833
FLR (Opt)	10	0.9705±0.0108	0.0886±0.0323	0.1888±0.0549	2.5228±0.1822
AODV	10	0.9545±0.0193	0.1151±0.0497	0.5980±0.2286	2.5531±0.1994
DSR	10	0.7929±0.0654	1.4612±0.5874	0.5373±0.1982	1.7331±0.2140
OLSR	10	0.8543±0.0396	0.3479±0.1390	0.8048±0.0883	2.3968±0.1740
FLR	30	0.9078±0.0284	0.4523±0.1718	0.7012±0.2190	2.5604±0.2086
FLR (Opt)	30	0.9108±0.0276	0.4494±0.1820	0.6835±0.2062	2.5539±0.2067
AODV	30	0.7647±0.0558	1.9168±0.6460	3.8584±1.1390	2.9081±0.3422
DSR	30	0.7378±0.0654	5.6598±1.5812	0.6013±0.1858	1.6920±0.2149
OLSR	30	0.8204±0.0381	1.3499±0.4458	0.6958±0.0744	2.3850±0.1720

Table 3: Performance average over all pause times for 100 nodes network

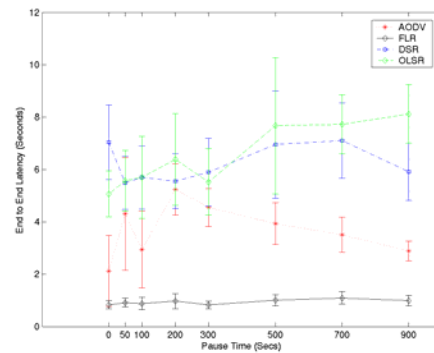
Protocol	Flows	Delivery Ratio	Latency (sec)	Net Load	Data Hops
FLR	10	0.8843±0.0282	0.2986±0.0942	1.0687±0.2895	3.9787±0.4724
FLR (Opt)	10	0.8863±0.0276	0.2996±0.0999	1.0639±0.2881	3.9506±0.4737
AODV	10	0.8359±0.0377	0.3536±0.0994	3.4476±0.9873	4.1541±0.5312
DSR	10	0.6774±0.0714	1.9072±0.4109	2.3397±0.8379	3.6690±0.5417
OLSR	10	0.6976±0.0547	2.1138±0.7194	5.7178±0.7198	3.9137±0.4811
FLR	30	0.7821±0.0363	0.9192±0.2126	2.9053±0.6039	4.0680±0.4520
FLR (Opt)	30	0.7863±0.0321	0.9044±0.1837	2.8382±0.5246	4.0883±0.4251
AODV	30	0.5619±0.0997	3.5194±1.3503	33.6577±35.7988	5.9871±0.8752
DSR	30	0.6266±0.0666	6.2102±1.3701	2.5235±0.7824	3.6898±0.4264
OLSR	30	0.6269±0.0461	6.4852±1.6194	5.3037±0.6408	4.0117±0.4518

Delivery Ratio



5

End-to-End Delay

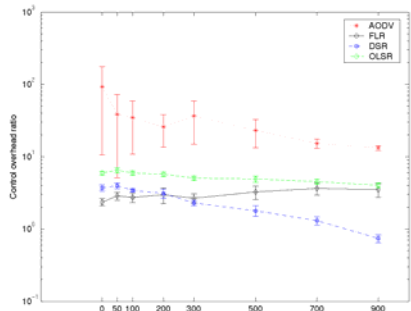


Sp

96

Control Overhead

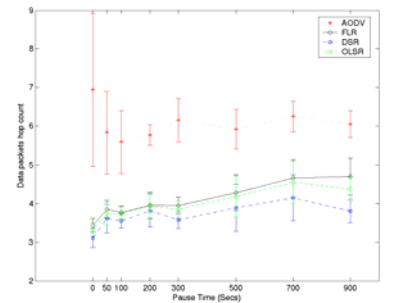
DSR's lower control overhead is due to route searches delayed because of incorrect source routes.



Spring 20

97

Data Hops



All protocols forward a equivalent number of data packets. Packet delivery ratio shows that FLR delivers more data packets to destinations.

Spring 2005

98

Conclusions

- Extended sufficient conditions for loop-freedom using distances to labels (path information).
- Demonstrated the feasibility of using path information to achieve hop-by-hop loop-free routing using FLR.
- Mechanisms to cope with the unreliability of the wireless medium.
- Simulation results show that FLR outperforms current state of the art MANET protocols.

Spring 2005

UCSC

CMPE257

99

Optimizations

- Inferring paths to relays.
 - Run a path selection algorithm (i.e., Dijkstra) over the current set of active labels.
 - Send route requests with complete path (source route) to destination and validate the path.
- Reverse routes.
 - Route requests when traversing set up labels for the source but the route is not activated.
 - Use source routed requests to activate the paths.

Spring 2005

UCSC

CMPE257

100

Optimizations (Cont.)

■ Multipath Routing

- A node A can store labels for destination D reported by neighbor B if $L_{DB}^A < FL_D^A$.
- Multiple paths can be used in one of the following ways
 - Use all available paths at the same time.
 - Use the next available path to achieve fail-over.

Spring 2005

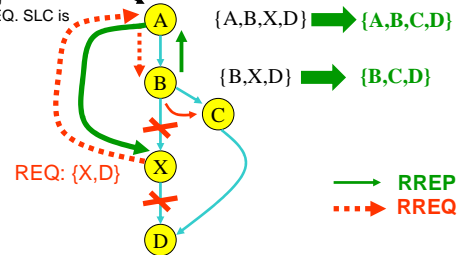
UCSC

CMPE257

101

Keeping Labels Unique

A cannot reply to X's request because X is in its current label {A,B,X,D}. Forwards RREQ. SLC is satisfied.



Now X can set a label {X,A,B,C,D}

Spring 2005

UCSC

CMPE257

102