Bypass routing: An on-demand local recovery protocol for ad hoc networks

Cigdem Sengul, Robin Kravets

Department of Computer Science, University of Illinois Urbana-Champaign, USA

Ad Hoc Networks, Volume 4, Issue 3, May 2006

Outline

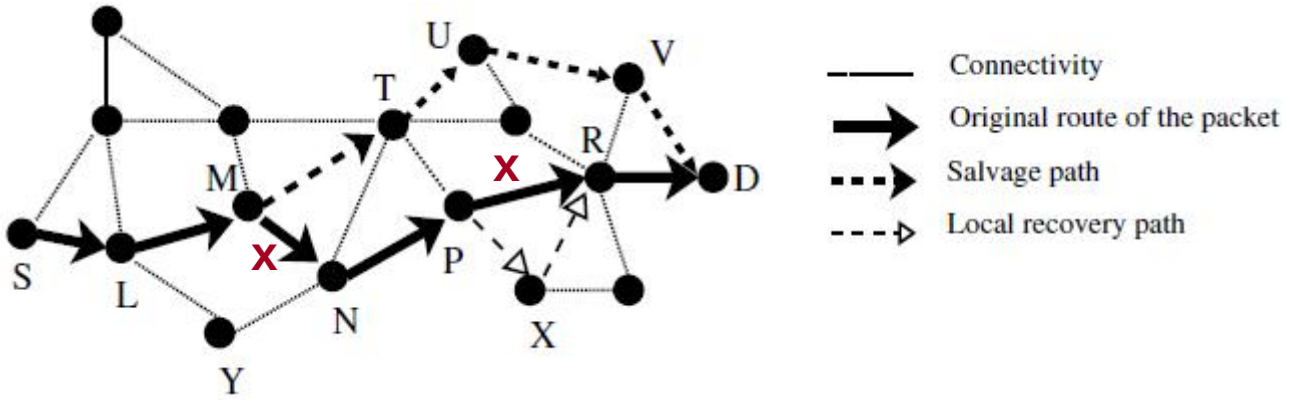
- Introduction
- Bypass routing
- Source routing with local recovery
- Performance evaluation
- Conclusions

Introduction

- Routing protocols for ad hoc networks can be categorized into three classes
  - Proactive
  - Reactive (on-demand)
  - Hybrid
- Route discovery in on-demand routing is typically performed via flooding

Bypass routing (1/8)

- It performs on-demand route recovery utilizing both route caches and local error recovery
  - A node first salvages a route by searching its route cache
  - If the node is not able to repair the route from its route cache
    - Bypass recovery



Error recovery example

When the link M-N breaks, Node M salvages the packet from its cache

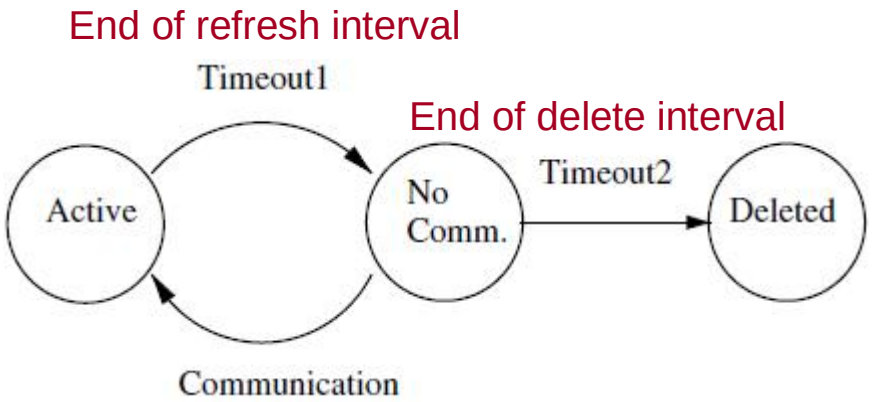
When the link P-R breaks, node P does not have alternate route in its cache

Bypass routing (2/8)

- It uses three mechanisms that work together to allow efficient recovery from route failures
  - MAC cache
  - Route cache
  - Error recovery
    - Route salvaging and Bypass recovery

Neighbor ID	Last Update	Link Status
Neighbor 1	Time 1	No Comm.
Neighbor 2	Time 2	Active

MAC cache



Operation of MAC cache

Bypass routing (3/8)

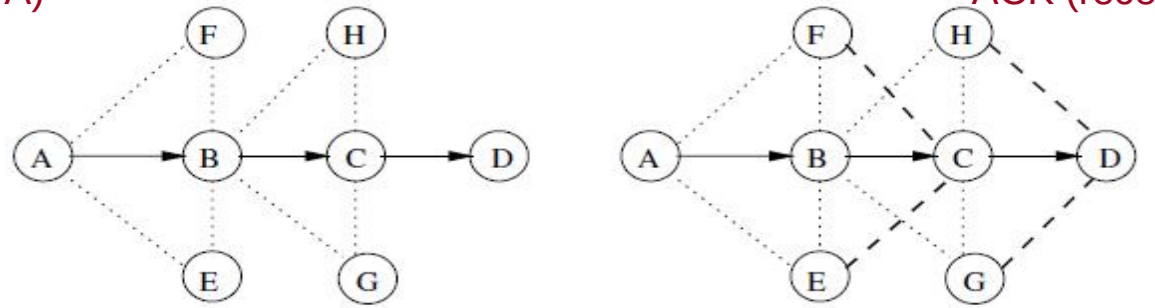
- IEEE 802.11 uses RTS (Request-to-send) and CTS (Clear-to-send) to provide a form of channel reservation
- A node listens for RTS-CTS-DATA-ACK messages and updates its MAC cache accordingly
  - RTS and DATA : sender and receiver information
  - CTS and ACK : receiver information
- A node that is not the sender of the RTS(DATA) can not determine the originator of the CTS(ACK)
  - A node can cache the sender of RTS(DATA)

Bypass routing (4/8)

- When the node overhears the CTS(ACK), it checks the recorded sender of RTS(DATA)
  - If the sender of the RTS(DATA) matches the receiver of the CTS(ACK)
    - The receiver of RTS(DATA) is a neighbor

F → RTS (sender : A, receiver : B)
CTS (receiver : A)

H → DATA (sender : C, receiver : D)
ACK (receiver : C)



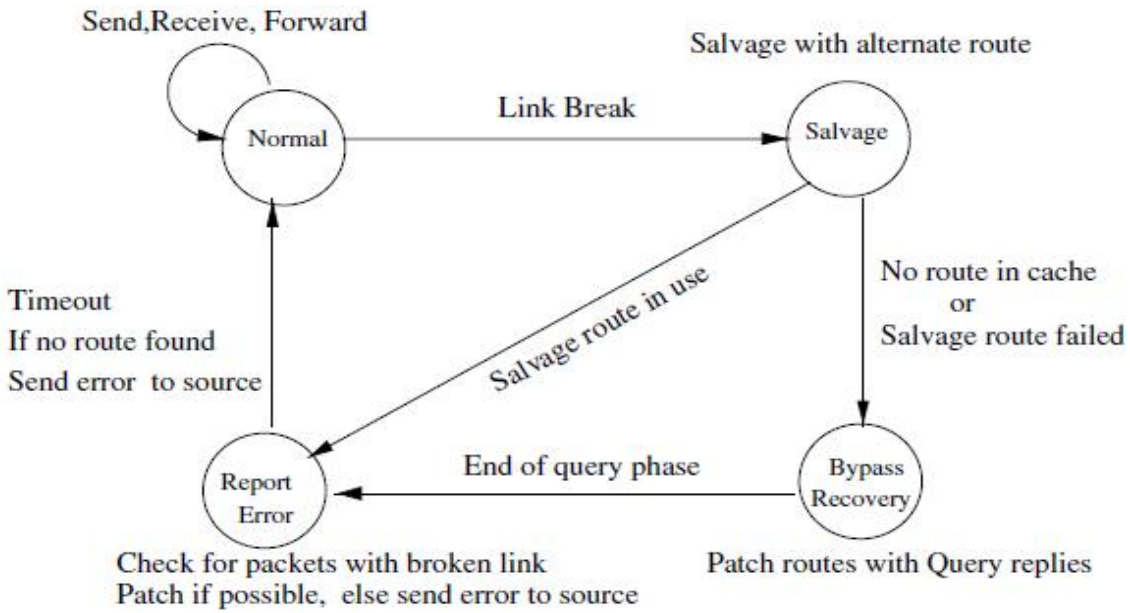
E → RTS (sender : A, receiver : B)
CTS (receiver : A)

G → DATA (sender : C, receiver : C)
ACK (receiver : C)

Path traversed by data
Links learned via snooping RTS/DATA
Links learned via snooping CTS/ACK

Bypass routing (5/8)

- Error recovery proceeds in three stages
  - Salvaging using route caches
  - Bypass recovery
  - Error reporting



Protocol state diagram

Bypass routing (6/8)

Salvaging using route caches

- A node searches its cache
- To record the recovery attempt in a fail-record table
- If the salvaged packet arrives at its destination
  - To send an enhanced route error message to the source

Bypass routing (7/8)

Bypass recovery

- The node buffers the failing packet and all packets that use the broken link in a fail-packet buffer
- A list of nodes existing on such soon-to-fail routes is broadcast to one-hop neighbors
- When the querying node receives a reply
  - It checks its fail-packet buffer to repair packets with new link-state information

Bypass routing (8/8)

Error reporting

- Salvaging using route caches
  - If no ACK is received from the destination
- Bypass recovery
  - If a repairing node does not receive query reply before a timeout occurs
- A route error message is sent back to the source

Source routing with local recovery (1/2)

- SLR (Source Routing with Local Recovery) uses DSR as the underlying protocol
  - Route selection and error recovery based on bypass routing
- SLR utilizes optimizations
  - Increased spreading of route errors
  - Snooping
  - Route salvaging

Source routing with local recovery (2/2)

- SLR uses a simple path cache known as Path-Gen-34
  - Primary cache
  - Secondary cache
- Path-Gen-34 does not employ a timeout mechanism
  - Bypass routing uses the MAC cache to determine the validity of route

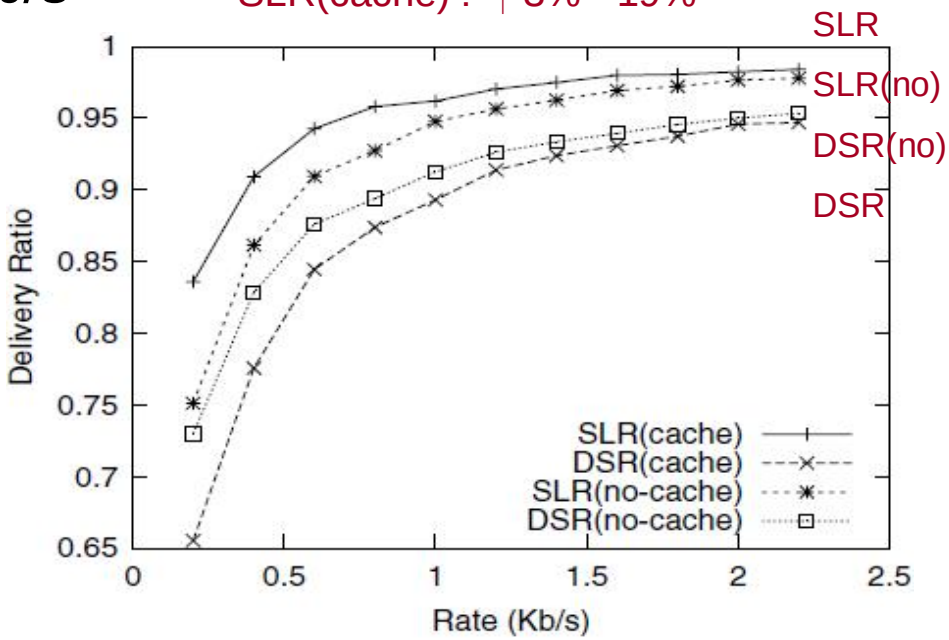
Performance evaluation (1/8)

- 1500m x 500m region, 60 nodes, 20 CBR connections
- All data packets are 128 bytes
- Speed : 0 - 20m/s, Pause time : 60s
- Transmission rate : 0.2 - 2.2Kb/s
- Each simulation runs for 600s

Fail-record: Table size	34
Fail-record: Timeout (s)	1.0
Fail-buffer: Packet timeout (s)	0.02
MAC Cache: Refresh interval (s)	0.05
MAC Cache: Delete interval (s)	3.0

Parameters used in SLR simulation

SLR(cache) : ↑ 3% - 19%

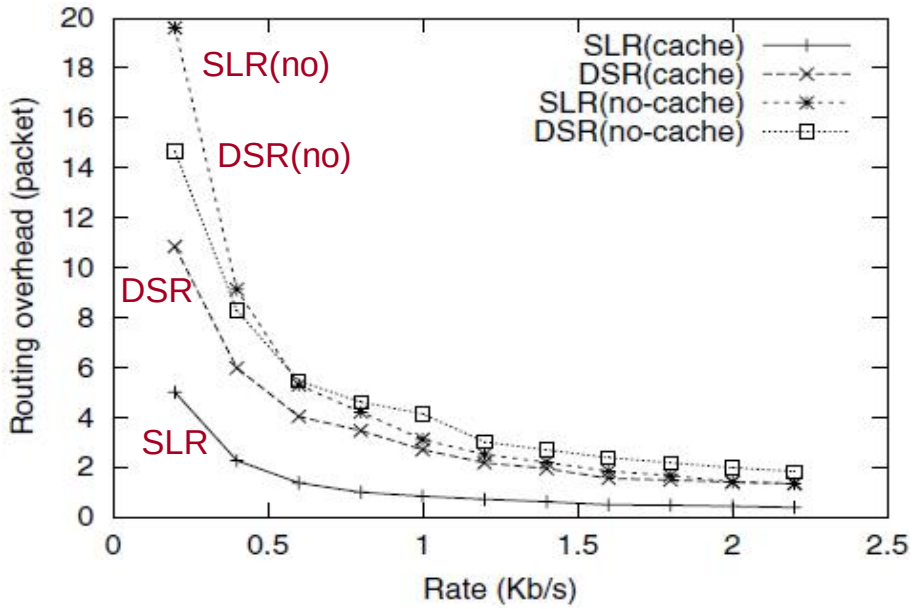


Delivery ratio vs. traffic load

Performance evaluation (2/8)

SLR(cache) vs. DSR(nocache) : ↓ 3 - 5 times

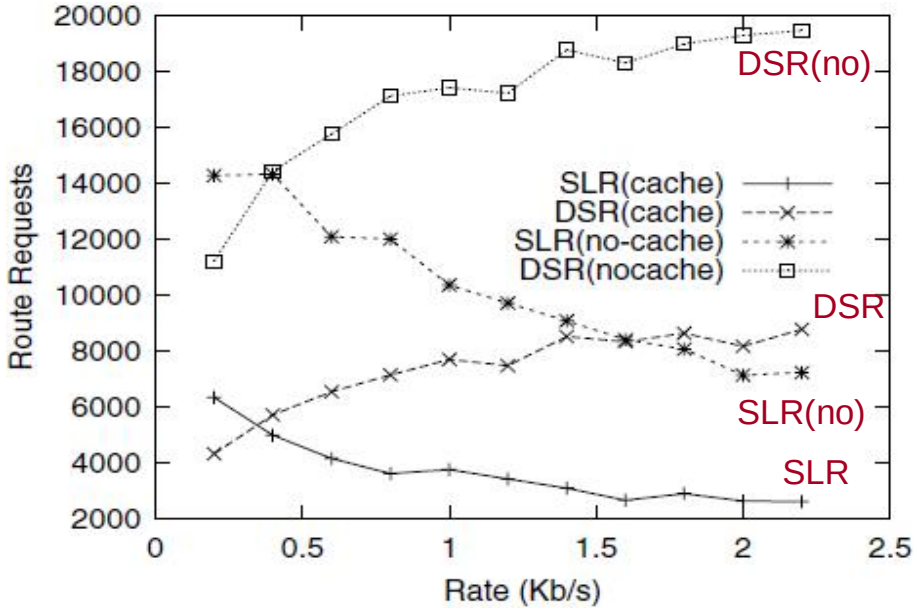
SLR(cache) vs. DSR(cache) : ↓ 2 - 4 times



Routing overhead vs. traffic load

DSR(nocache) : 10,000 - 20,000 packets

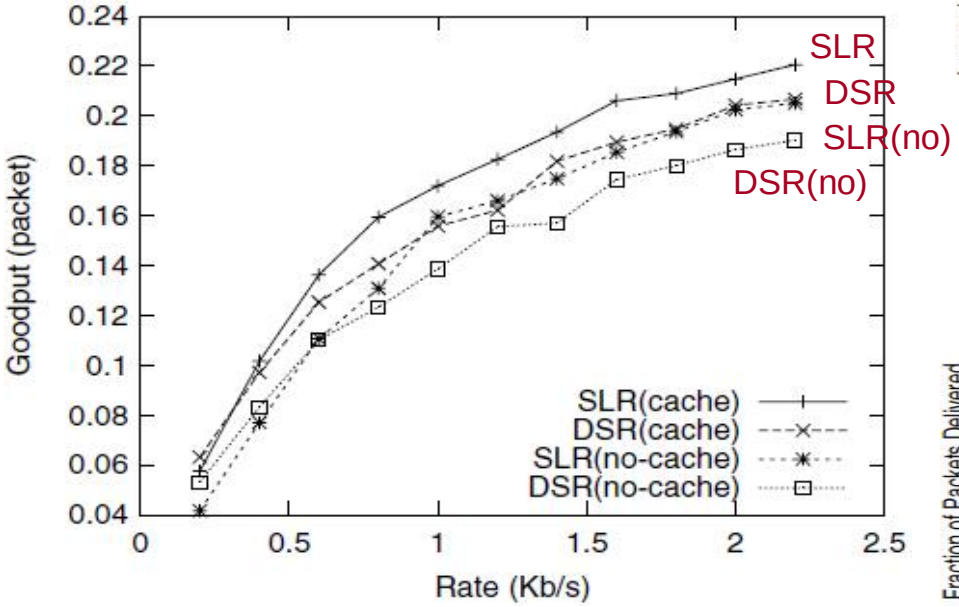
SLR(cache) : 2,000 - 6,000 packets



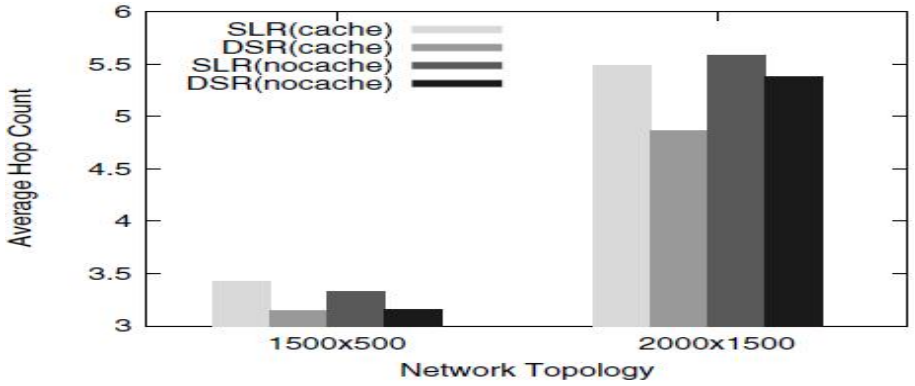
Route requests vs. traffic load

20 CBR connections, 60 nodes, 1500m x 500m region, speed 0-20m/s

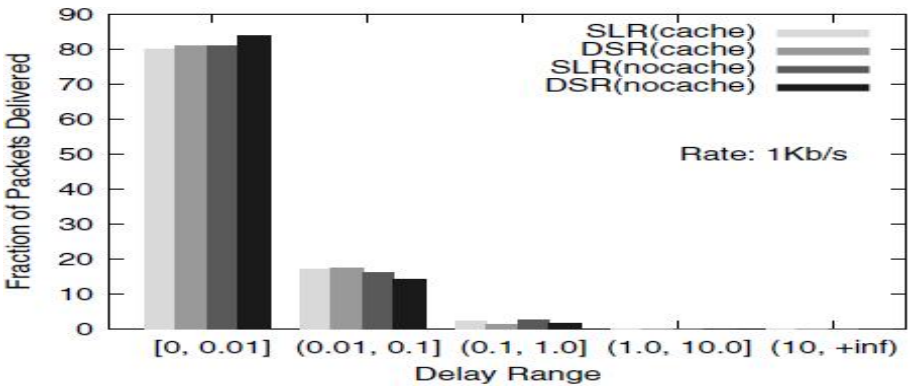
Performance evaluation (3/8)



Goodput vs. traffic load



Average hop count with different network topologies



Goodput vs. traffic load

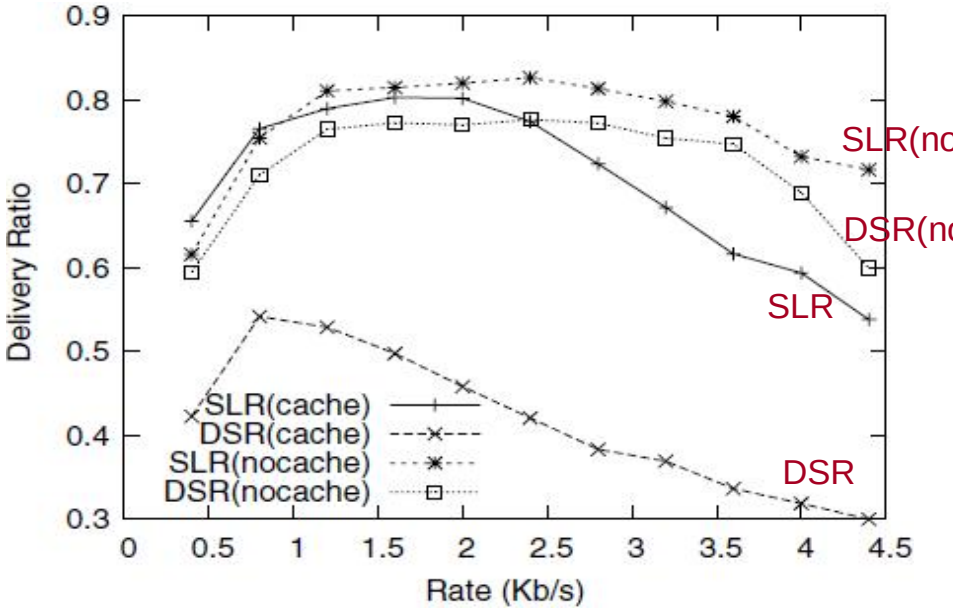
20 CBR connections, 60 nodes, 1500m x 500m region, speed 0-20m/s

Performance evaluation (4/8)

- 2000m x 1500m region, 150 nodes, 40 CBR connections
- All data packets are 256 bytes
- Speed : 0 - 20m/s, Pause time : 60s
- Transmission rate : 0.4 - 4.4Kb/s
- Each simulation runs for 900s

SLR(cache) vs. SLR, DSR(nocache) : ↓ 2 - 13%

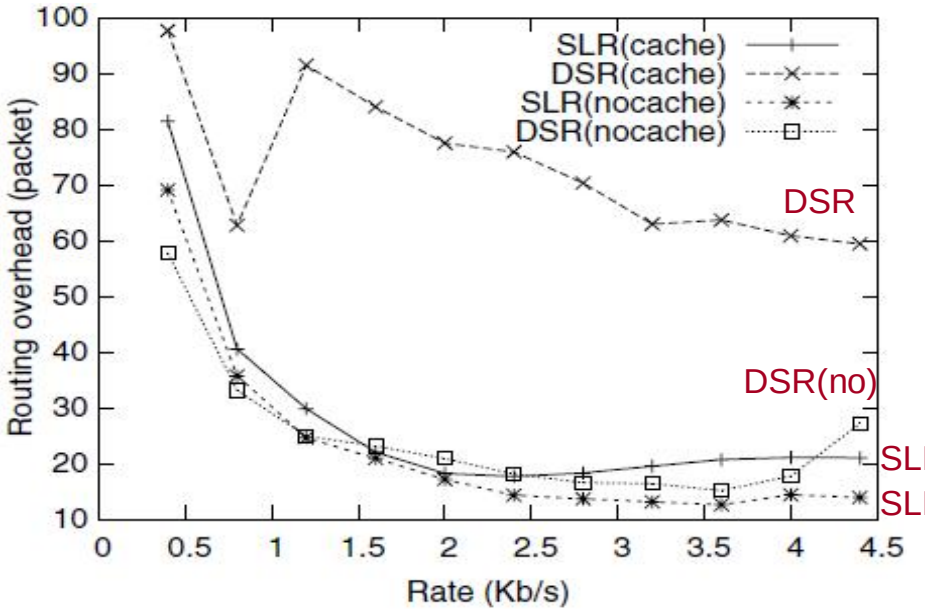
SLR(cache) vs. DSR(cache) : ↑ 22 - 35%



Delivery ratio vs. traffic load

Performance evaluation (5/8)

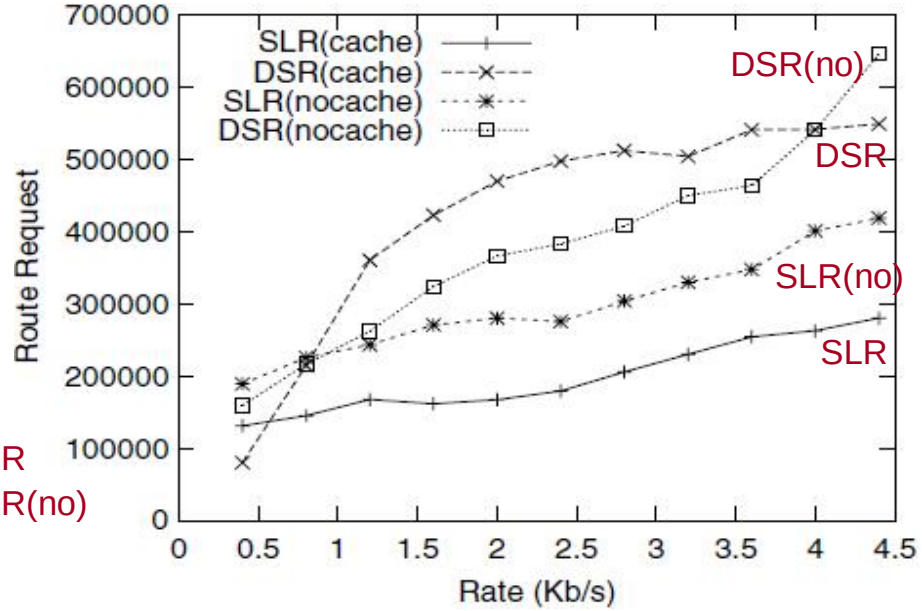
DSR(cache) vs. DSR(nocache), SLR : ↑ 5 times



Routing overhead vs. traffic load

DSR(cache) : 427,000 packets

SLR(cache) :199,000 packets

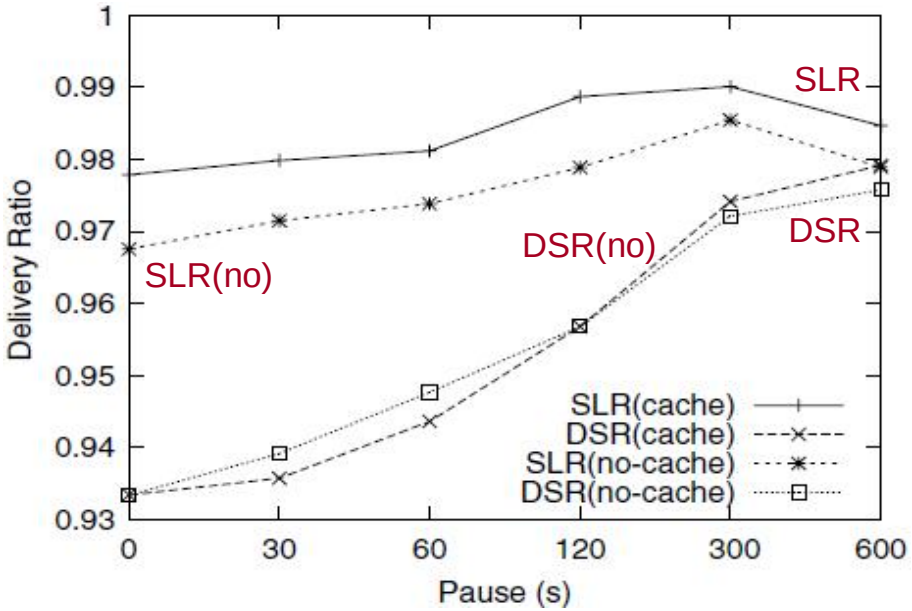


Route requests vs. traffic load

40 CBR connections, 150 nodes, 2000m x 1500m region, speed 0-20m/s

Performance evaluation (6/8)

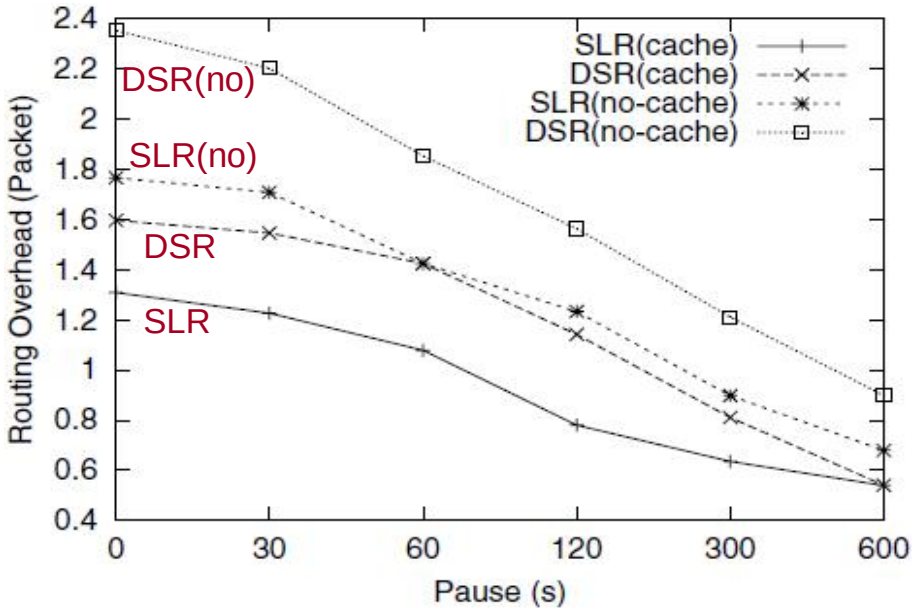
- 1500m x 500m region, 60 nodes, 20 CBR connections
- All data packets are 256 bytes
- Speed : 0 - 20m/s, Pause time : 0, 30, 60, 120, 300 and 900s
- Transmission rate : 4Kb/s
- Each simulation runs for 600s



Delivery ratio vs. mobility

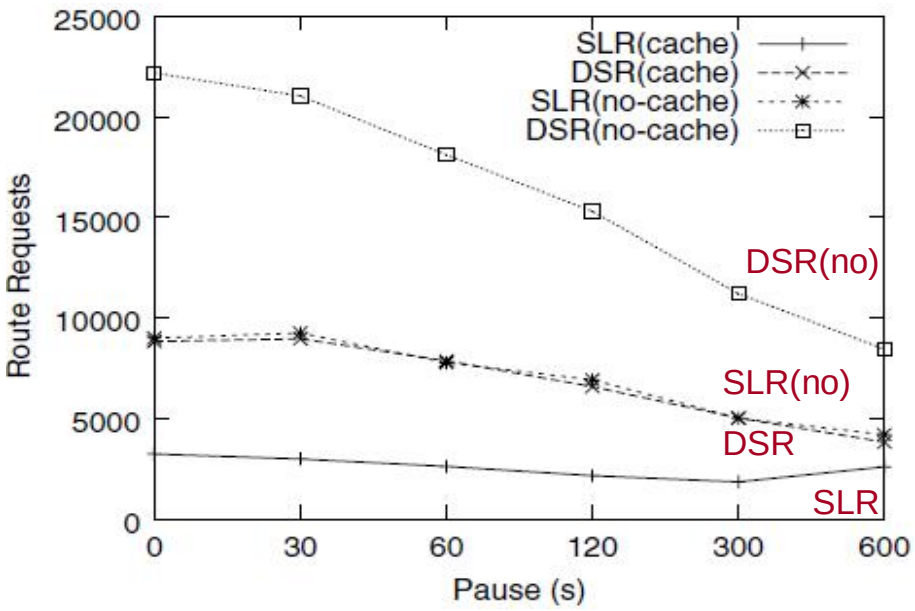
Performance evaluation (7/8)

SLR(cache) vs. DSR(nocache) : ↓ half



Routing overhead vs. mobility

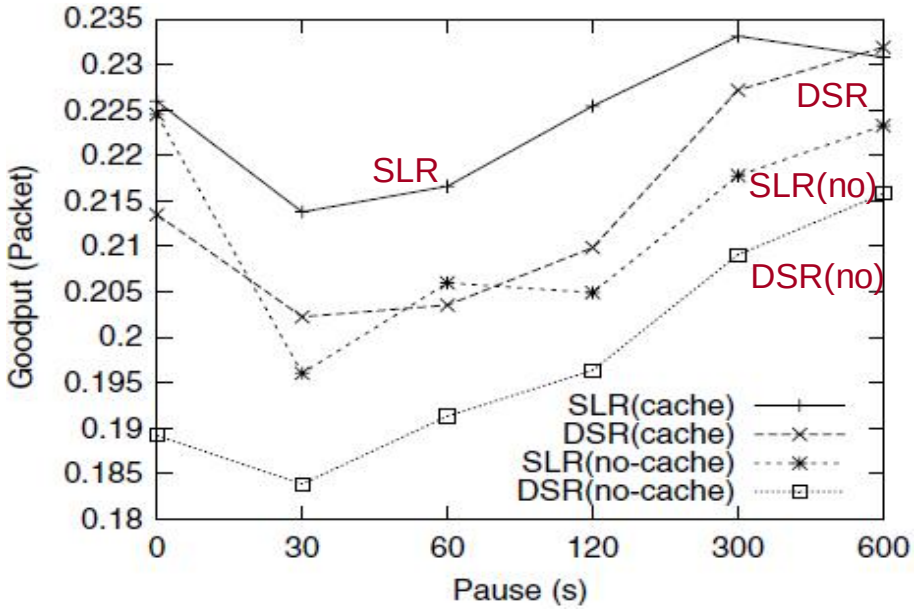
SLR(cache) : 3000 packets



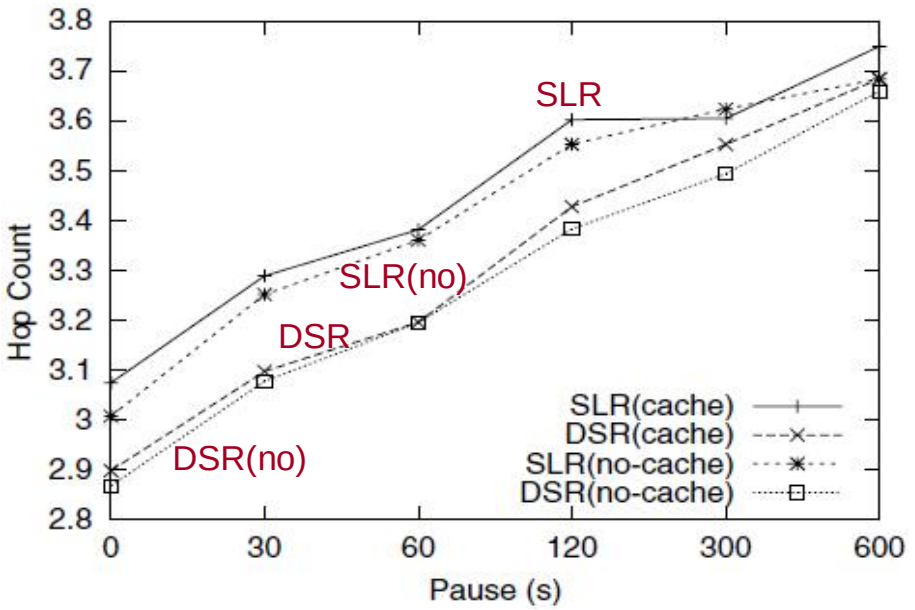
Route requests vs. mobility

20 CBR connections, 60 nodes, 1500m x 500m region, speed 0-20m/s

Performance evaluation (8/8)



Goodput vs. mobility



Average number of hops vs. mobility

20 CBR connections, 60 nodes, 1500m x 500m region, speed 0-20m/s

Conclusions

- Bypass routing reduces the need to perform route discovery for broken routes
  - Bypass recovery
  - A novel cache invalidation mechanism
- To provide robustness to route failures and maintain high delivery ratio and low overhead