Data Dissemination with Ring-Based Index for Wireless Sensor Networks

IEEE TRANSACTIONS ON MOBILE COMPUTING

Presented by Chia-Yi Lien

May 15, 2008

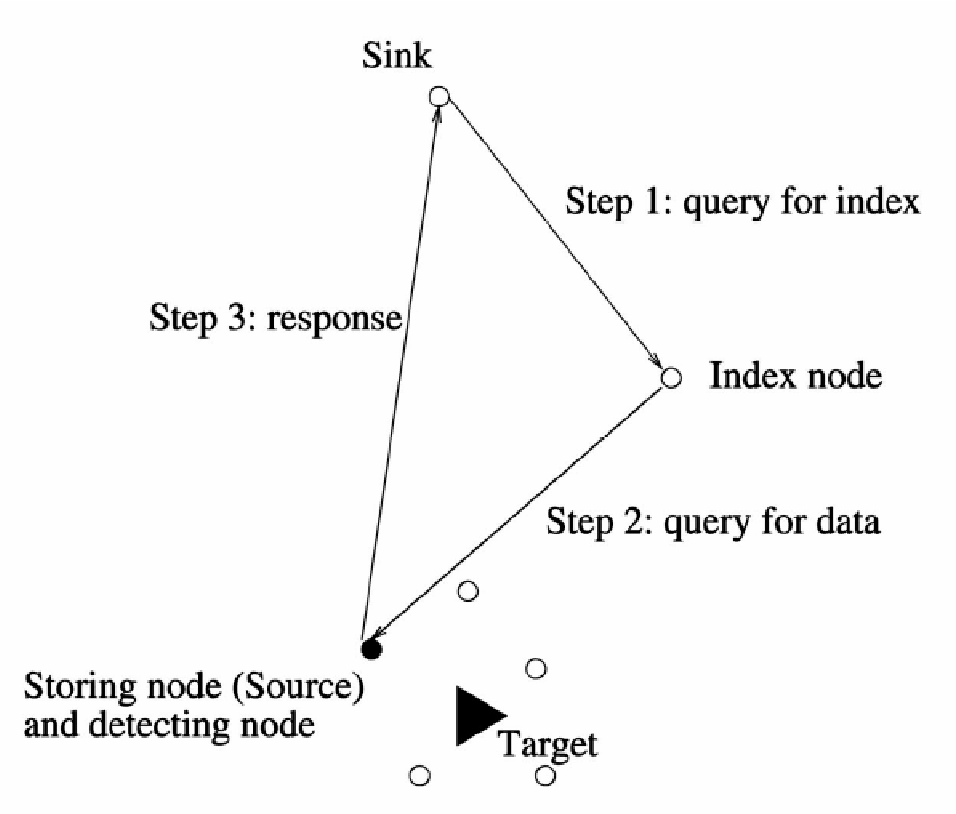
Outline

- Introduction
- The Index-based Data Dissemination
- An Adaptive Ring-based Index (ARI) Scheme
- Enhancements
- Performance Evaluations
- Conclusions

Introduction

- Scenario
  - A large amount of sensing data are generated, but only a small portion of them will be queried by users

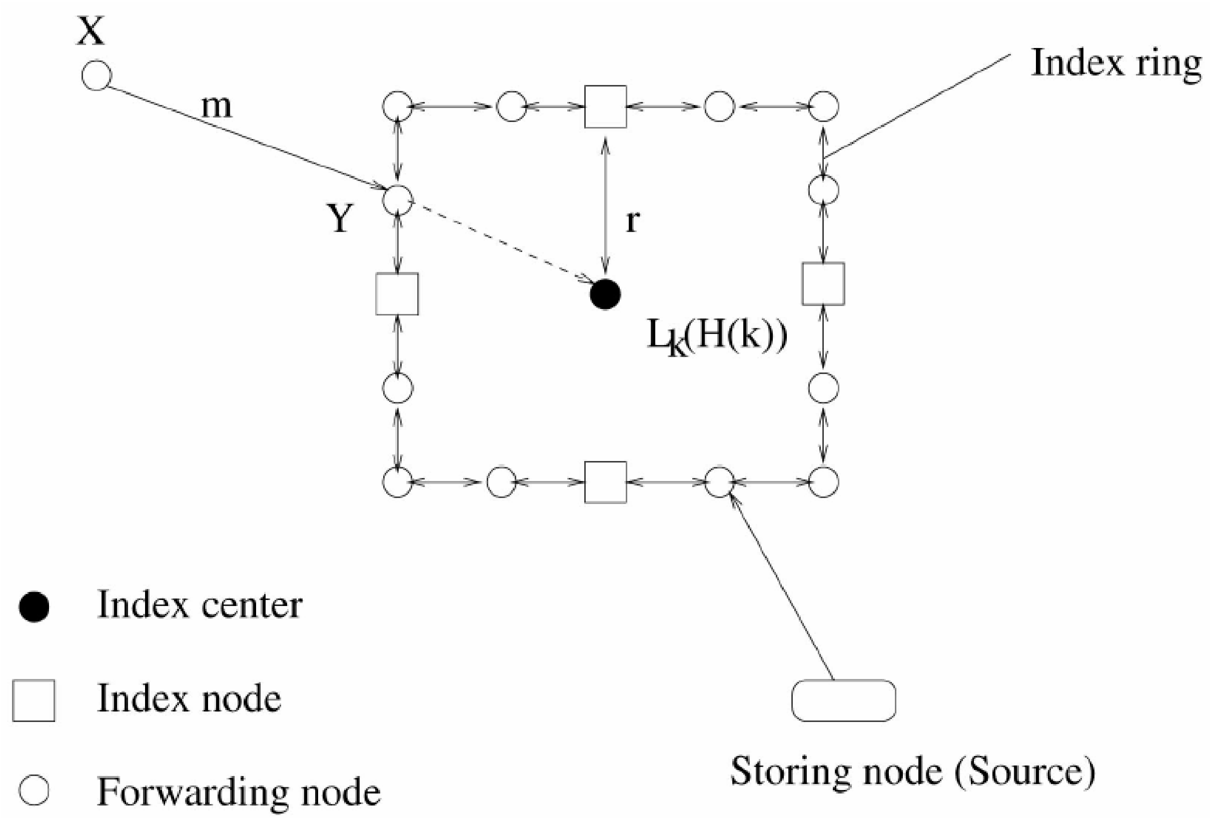
The Index-based Data Dissemination



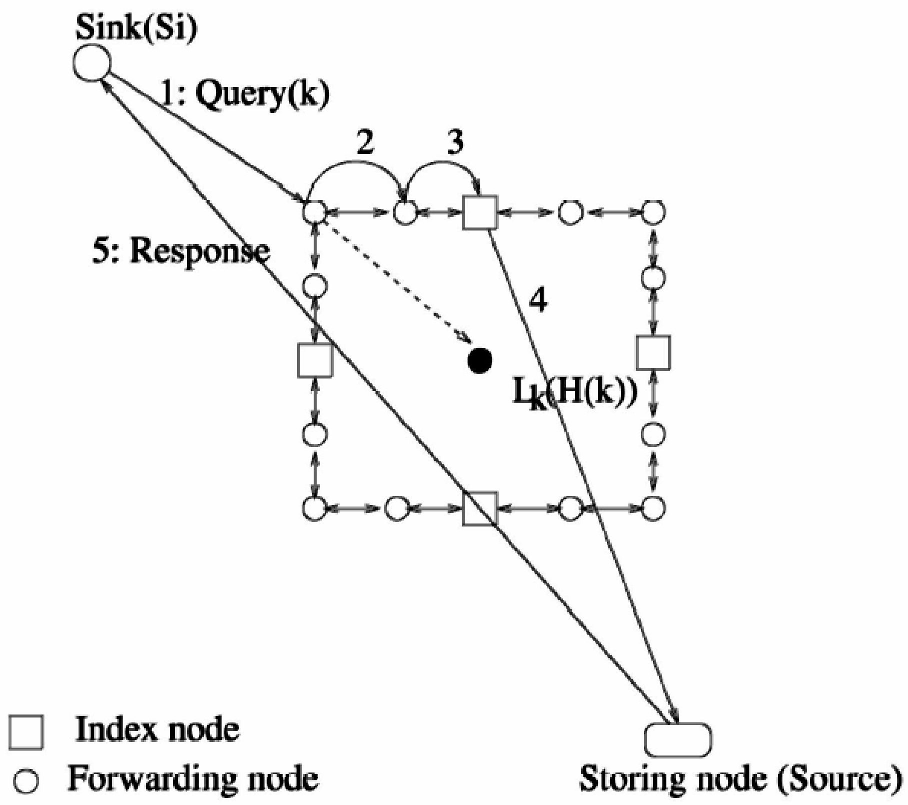
An Adaptive Ring-based Index (ARI) Scheme

- Goal
  - Fault tolerance
  - Load balance
  - Efficiency

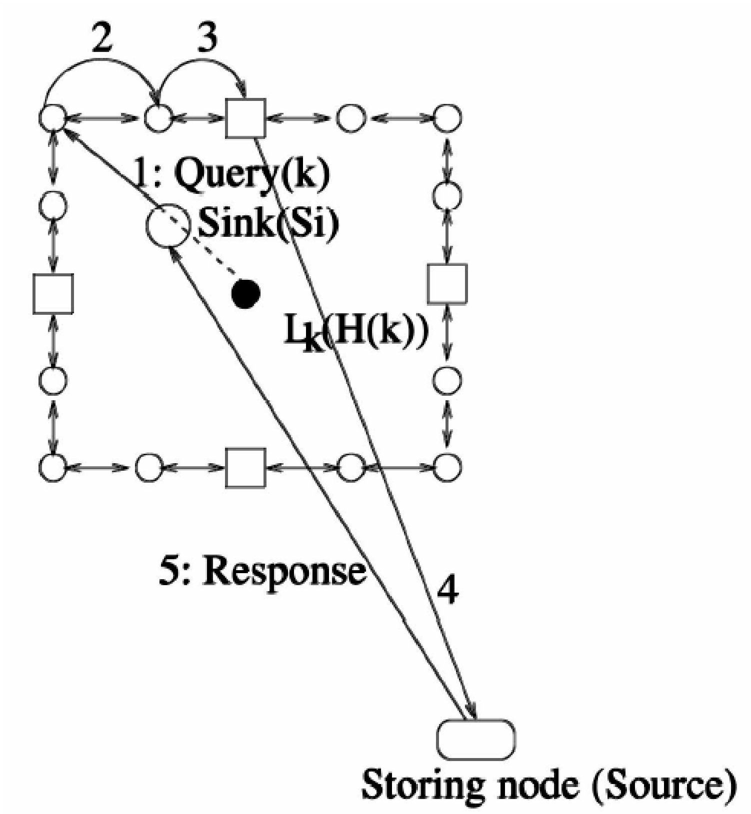
Initializing an index ring



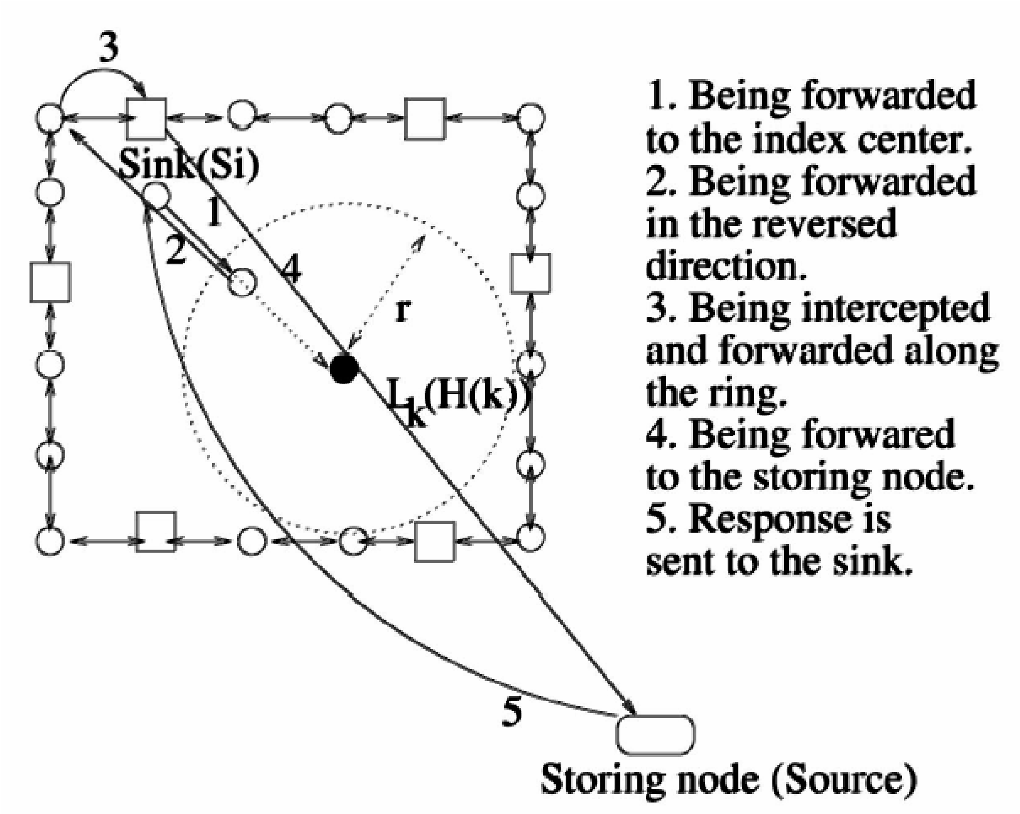
Query an index (1/3)



Query an index (2/3)



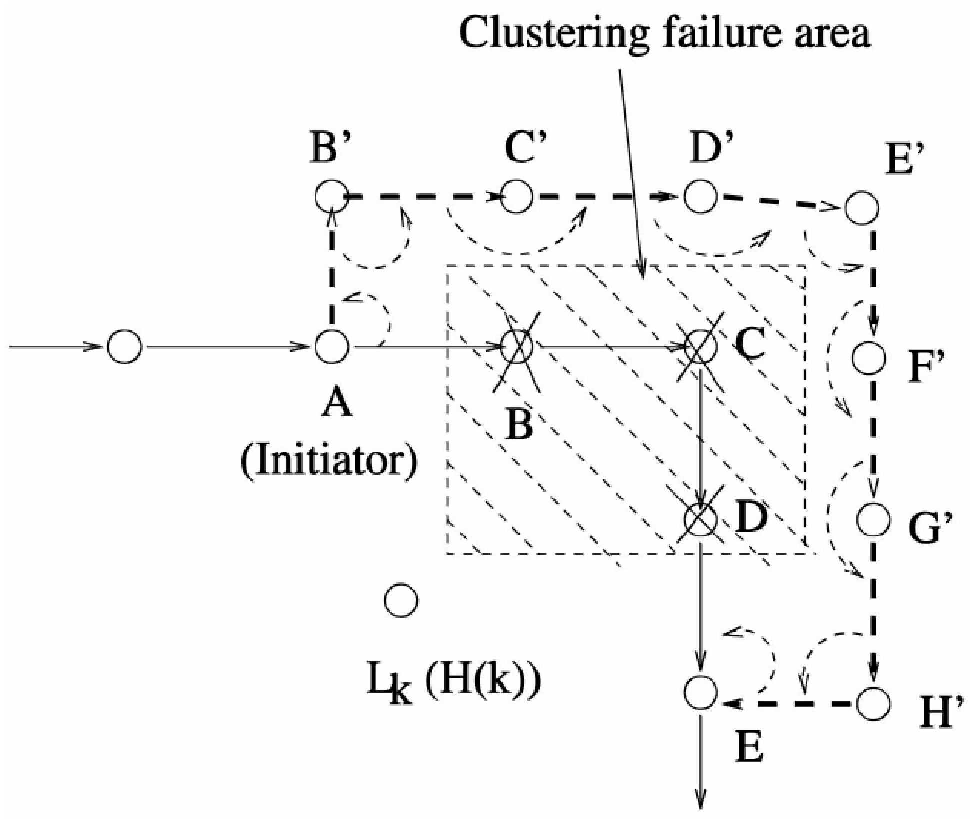
Query an index (3/3)



Updating an index

- Similar to query an index
- When the message arrives at an index node on the ring, the node updates its index and forwards the message along the circle in the clockwise direction
- The message is dropped when it is forwarded back to a node that has already received it

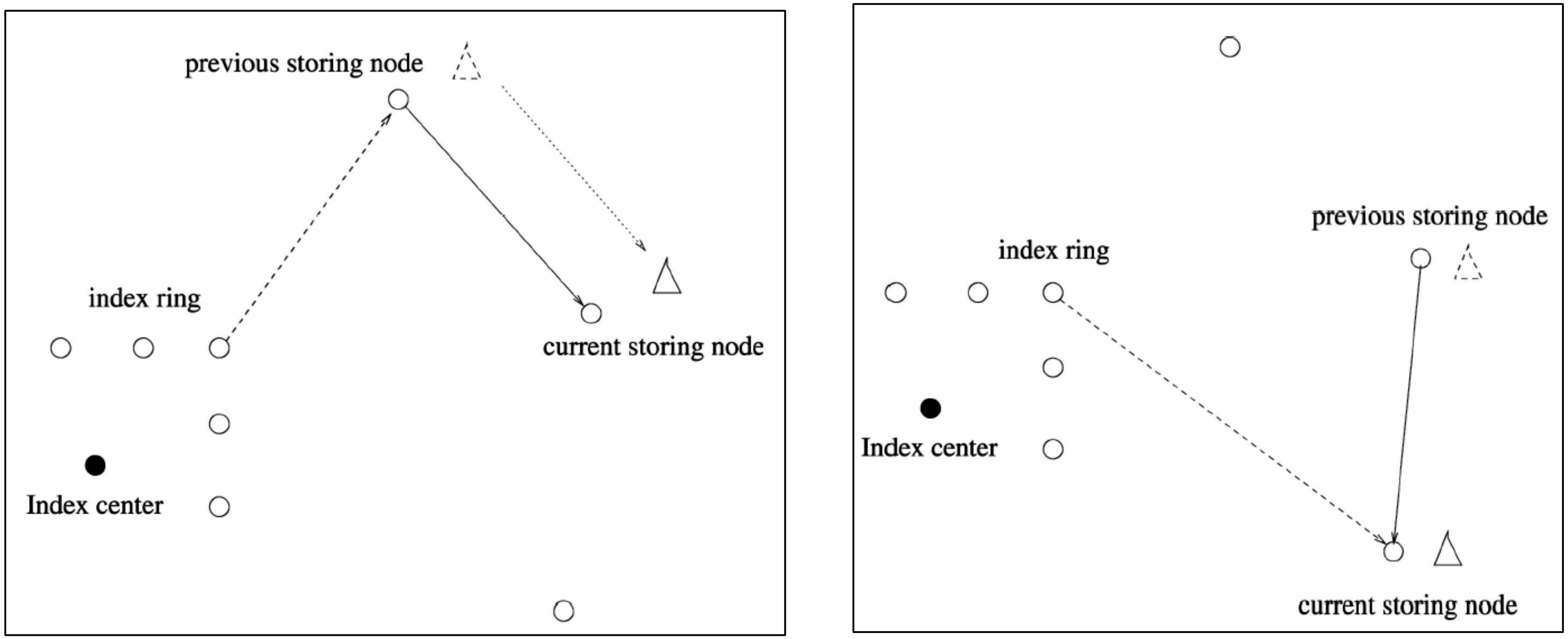
Dealing with clustering failures



Enhancements

- Lazy index updating (LIU)
- Lazy index query (LIQ)

Lazy index updating (LIU)



Lazy index query (LIQ)

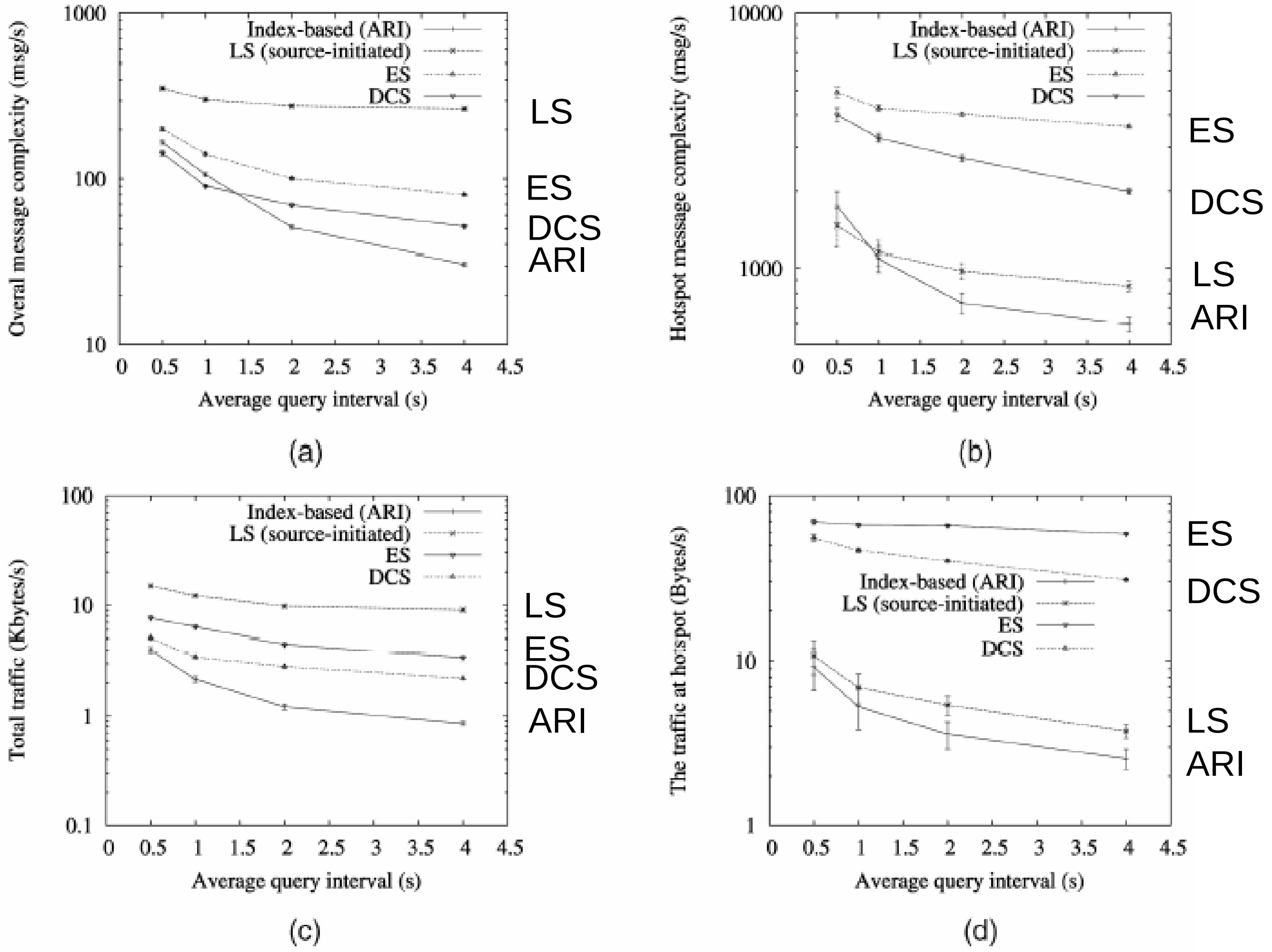
- An old storing node keeps a pointer to the next storing node for at least $\max(\beta, \theta)$, where θ is system parameter, and β represents the time period that an old storing node should keep an pointer to the next storing node
- When a source replies a data message to a sink, it attaches its location to the message. On receiving the message, the sink caches the location.
- When a sink wants to query the source of a target, it first checks if it has cached the location of the source. If the location is cached and the caching time is less than θ, it will send a query directly to the source. Otherwise, the query is sent to the index nodes.

Performance Evaluations

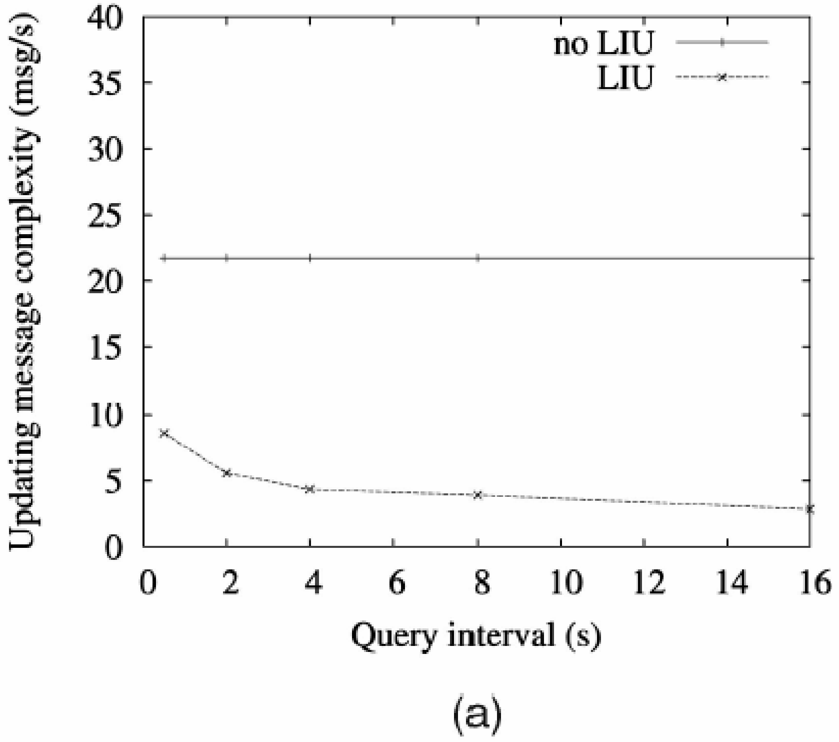
TABLE 2
Simulation Parameters

Parameter	Value
field size (m^2)	850×850
number of nodes	2500
communication range (m)	40.0
grid side (m)	17.0
number of target types: N_t	10
data update rate: r_d (per target per second)	0.25
number of index centers: N_i	4
the migration threshold for a source (m):	34.0
initial radius of an index ring: r (m)	34.0
initial number of index nodes on a ring: m	4
simulation time for each experiment (s)	1000.0
average velocity of a mobile target: v (m/s)	1.0-6.0
size of an update message ($byte$)	10
size of a query message ($byte$)	10
size of a data message ($byte$)	50

Compare the performance of data dissemination schemes



The index updating message complexity



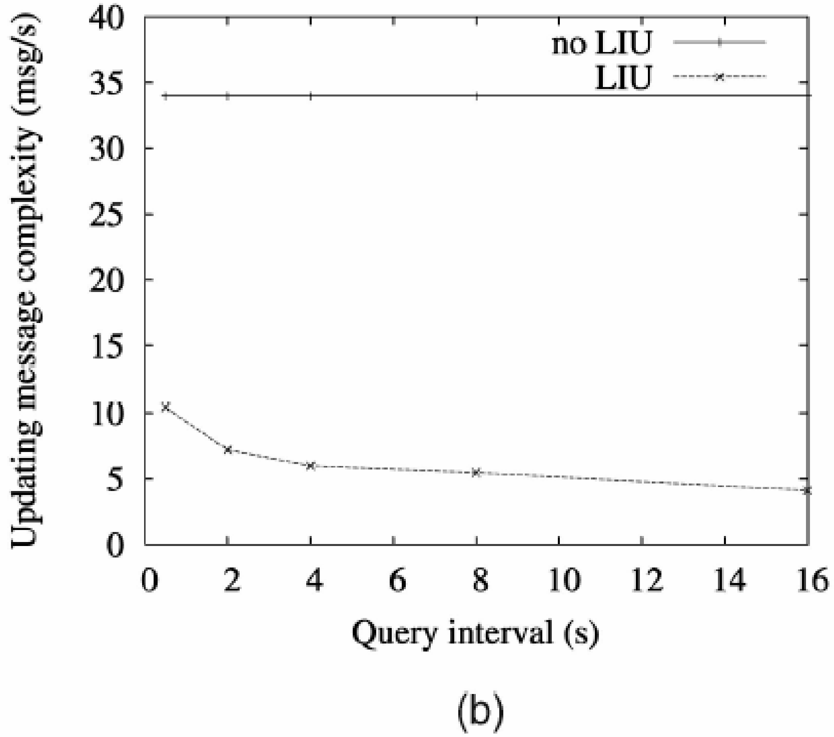


Fig. 13. The index updating message complexity with/without LIU ($r = 34m * 2, m = 8$). (a) v = 3.0 m/s. (b) v = 6.0 m/s.

The query message complexity

No LIU : $sink \rightarrow index\_node \rightarrow current\_source$

LIU : $sink \rightarrow index\_node \rightarrow old\_source_1 \rightarrow \cdots old\_source_m \rightarrow current\_source.$

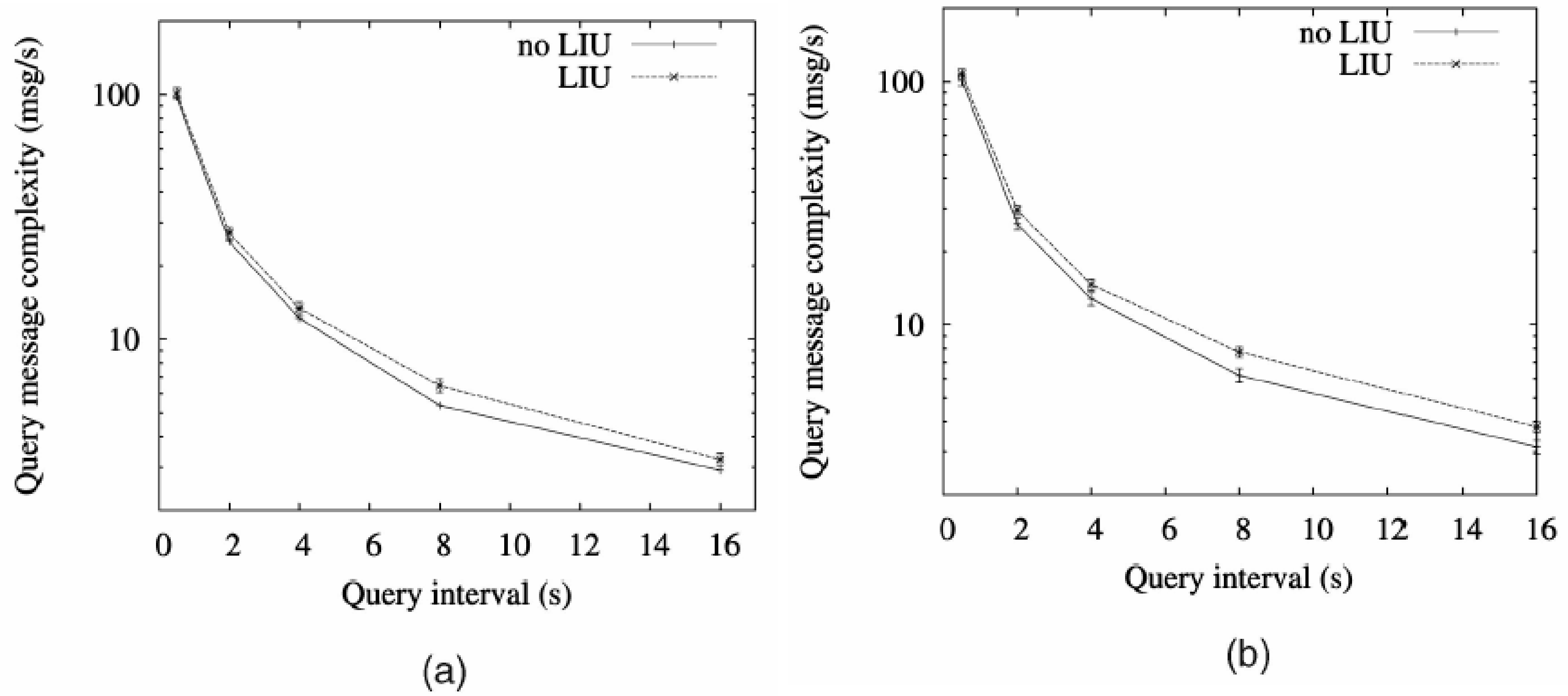


Fig. 14. The query message complexity with/without LIU ($r = 34m * 2, m = 8$). (a) v = 3.0 m/s. (b) v = 6.0 m/s.

The average query delay

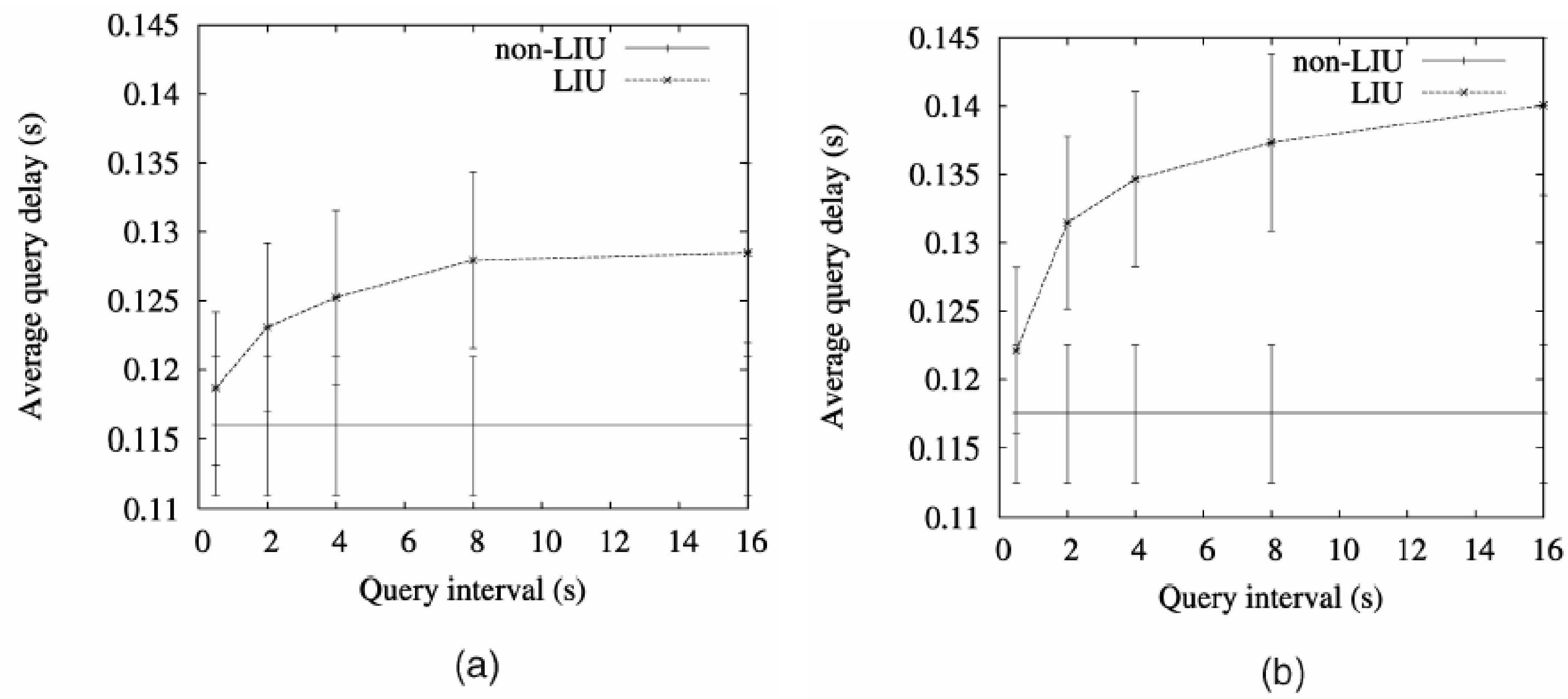


Fig. 15. The average query delay with/without LIU ($r = 34m * 2, m = 8$). (a) v = 3.0 m/s. (b) v = 6.0 m/s.

The total message complexity with LIU

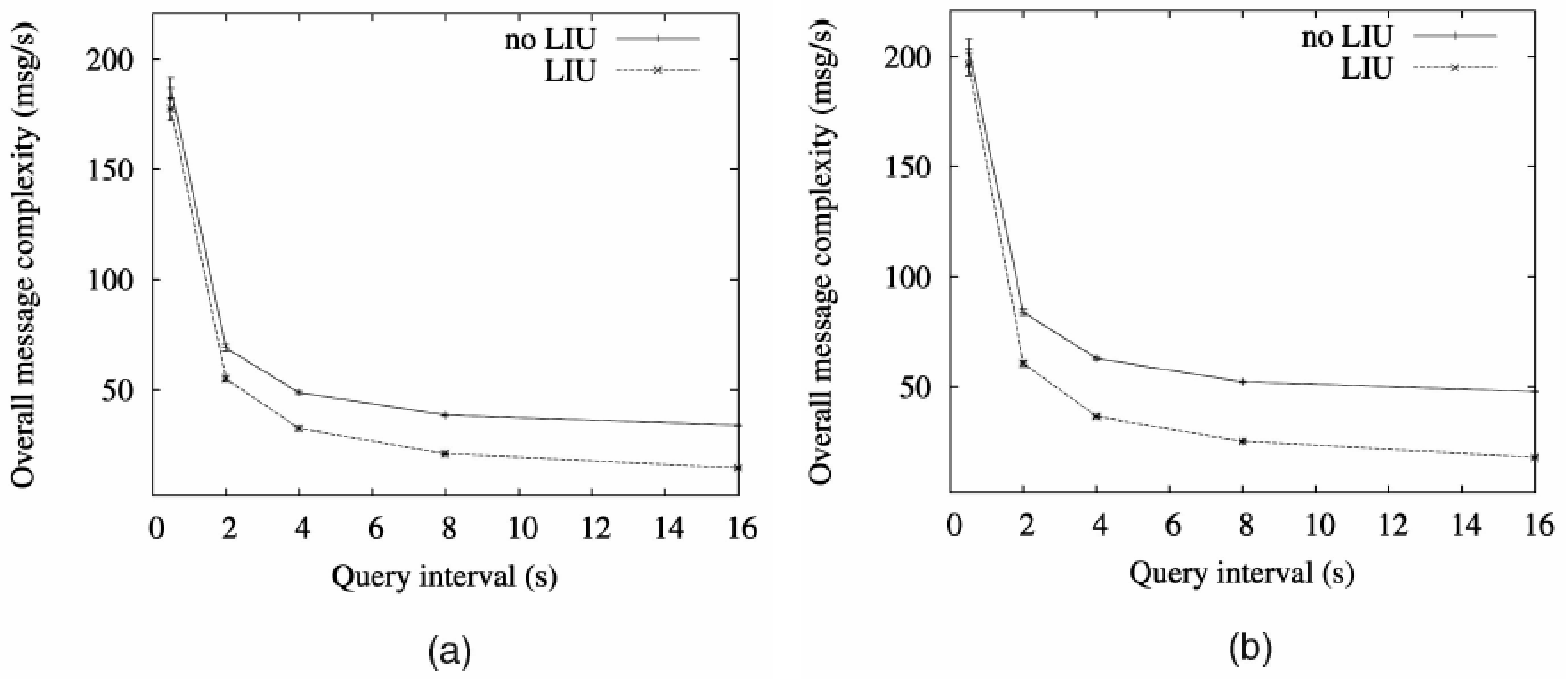


Fig. 16. The total message complexity with/without LIU ($r = 34m * 2, m = 8$). (a) v = 3.0 m/s. (b) v = 6.0 m/s.

The message complexity with LIQ

No LIQ : $sink \rightarrow index\_node \rightarrow old\_source_1 \cdots \rightarrow old\_source_m \rightarrow current\_source.$

LIQ : $sink \rightarrow old\_source_1 \cdots \rightarrow old\_source_m \rightarrow current\_source.$

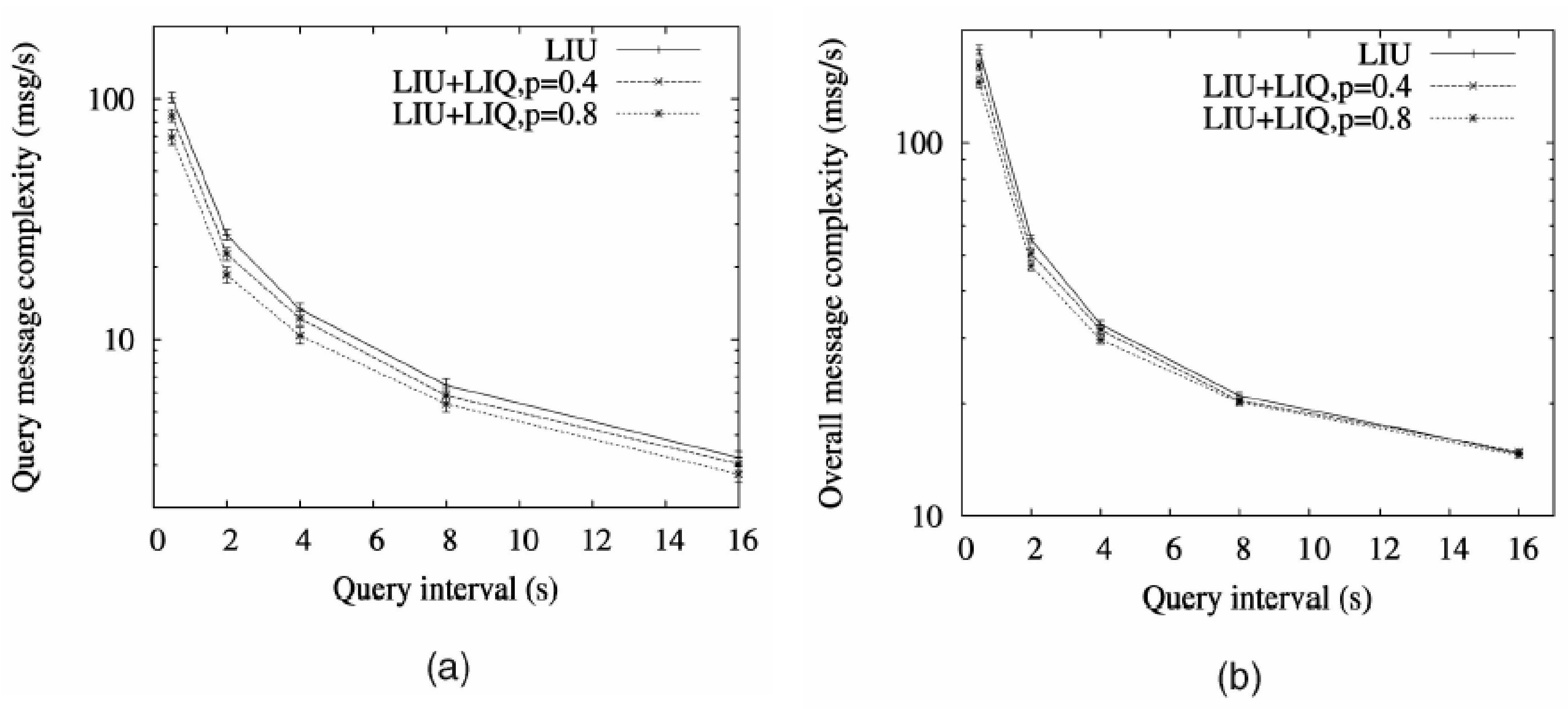


Fig. 17. The message complexity with/without LIQ ($r = 34m * 2, m = 8, v = 3.0$ m/s).

Conclusions

- Simulation results show that the index-based scheme outperforms the ES scheme, the DCS scheme, and LS scheme
- Authors also proposed several mechanisms to optimize the ARI scheme and the proposed optimization mechanisms can further improve the system performance