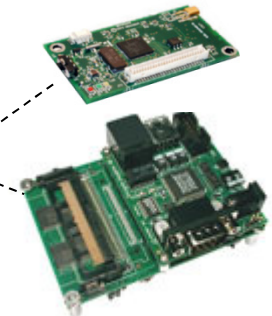
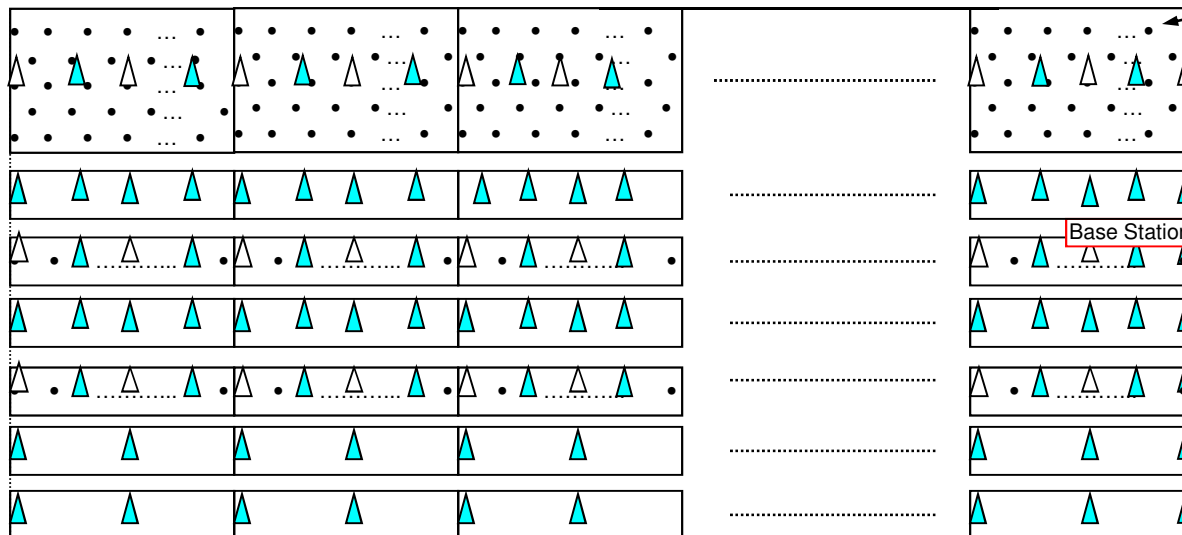
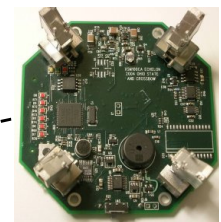
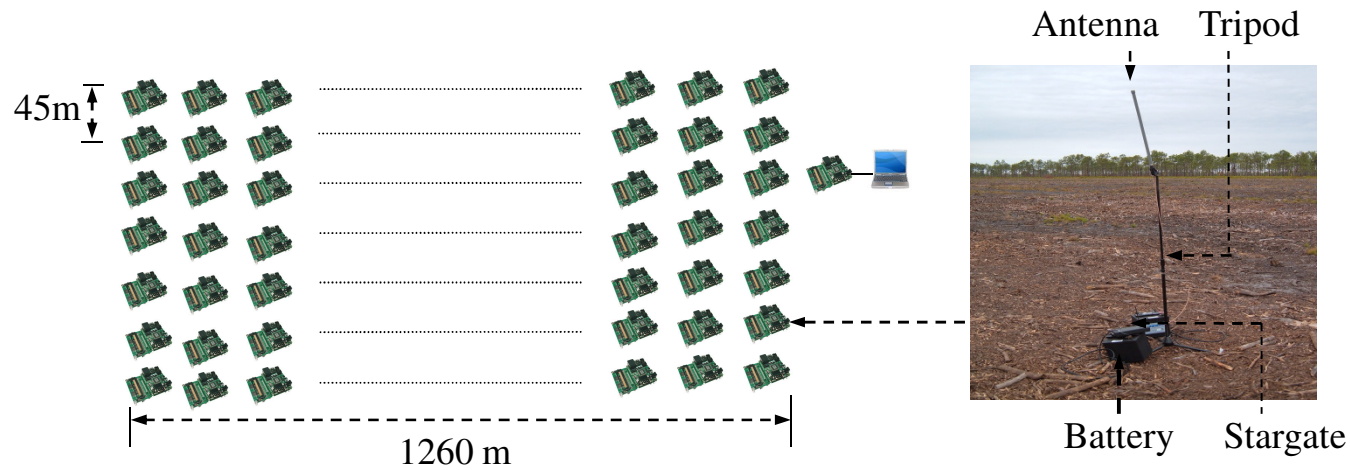


Context: ExScal (<http://www.cse.ohio-state.edu/exscal>)

- Field project to study scalability of middleware and applications in sensor networks
- Deployed in an area of $\sim 1,300\text{m} \times 300\text{m}$ (Dec. 2004)
- 2-tier architecture
 - Lower tier: $\sim 1,000$ XSM, ~ 210 MICA2 sensor nodes (TinyOS)
 - Higher tier: ~ 210 IEEE 802.11b Stargates (Linux)



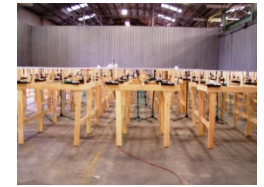
A core challenge of ExScal: Routing in the 802.11b backbone network



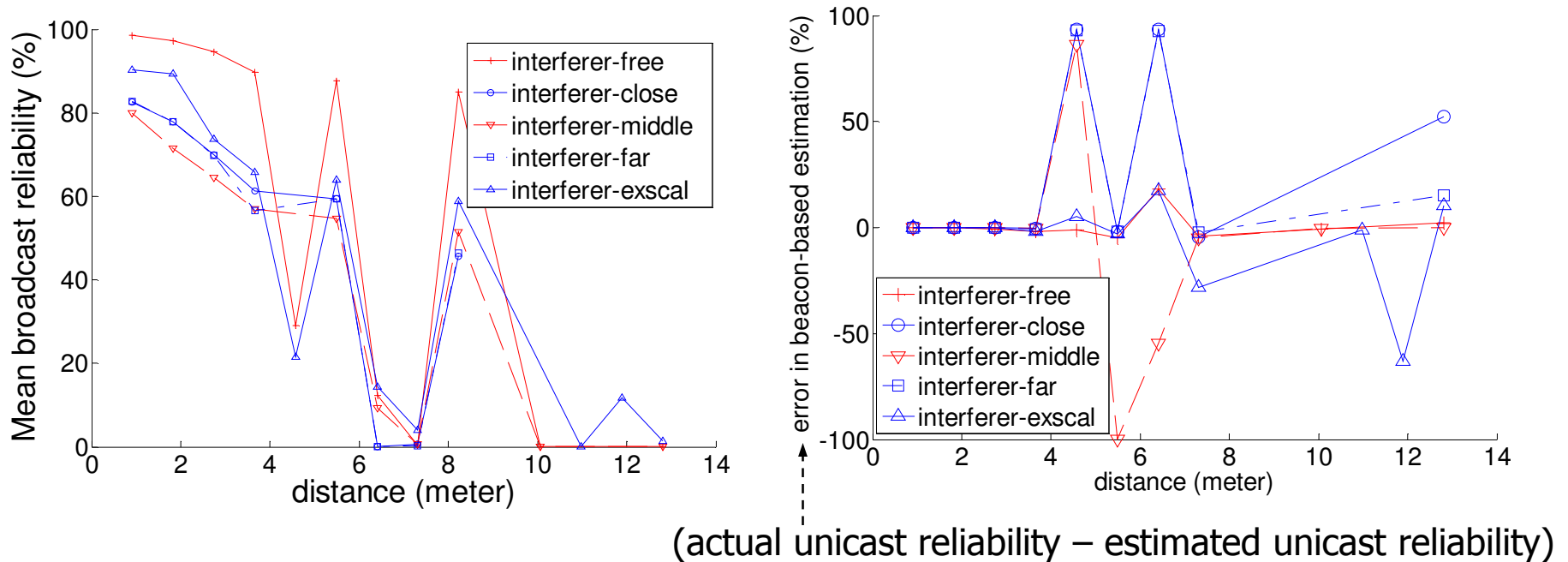
- Requirements on high reliability, low latency, energy efficiency
- Bursty high-volume data traffic
 - Existing schemes did not work well for us in *A Line in the Sand* (e.g., using unreliable links)
- Large scale
 - Long routes pose more challenges

Why did not existing routing protocols work for us?

- Existing routing protocols use **beacon-based link estimation**
 - Neighbors periodically exchange broadcast beacons
 - Estimate unicast properties via those of broadcast
 - Note: application data is transmitted using unicast
- Drawbacks of beacon-based estimation
 - Network condition experienced by beacons may not apply to data
 - Traffic pattern may change quickly (especially in event-detection applications), and Traffic pattern affects link properties due to interference
 - It is hard to precisely estimate unicast properties via those of broadcast beacons
 - Temporal link properties (e.g., correlation, variance): non-trivial to model, and not considered in well-known approaches such as ETX



An example: Impact of traffic pattern and temporal properties



- Network condition varies significantly across different interference scenarios (e.g., up to 39.26%); variations change with distance
- There is significant estimation error, especially in the transitional region; error changes with distance and interference pattern

Implications

- Beacon-based link estimation tends to be imprecise (and degenerates network performance)
- Should we explicitly model the impact of traffic patterns and temporal link properties?
Non-trivial
- Our answer: data-driven link estimation and routing
 - Estimate unicast link properties via data transmission itself
 - Circumvent the drawbacks and complexity of beacon-based estimation

Data-driven link estimation and routing

- Data-driven link estimation
 - How to evaluate link properties via data transmission itself?
- Data-driven routing

Link estimation via MAC feedback

application
transport
network
link: MAC
physical


- MAC (medium access control) feedback carries information on
 - Success or failure
 - MAC latency
 - time spent in transmitting a packet (including retries)
- **MAC latency** as the basis of link evaluation
 - MAC latency reflects link reliability, since link reliability determines the number retries at MAC layer
 - Routes of lower MAC latency tend to be more reliable
 - Reducing end-to-end MAC latency also improves network throughput

Data-driven link estimation and routing

- Data-driven link estimation

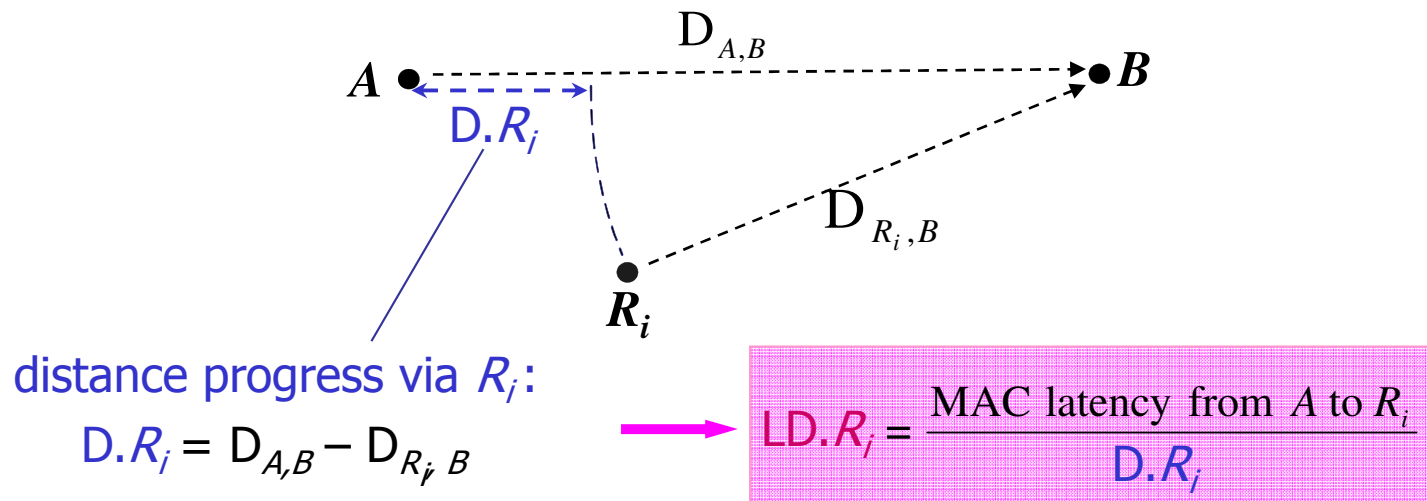
- **Data-driven routing**
 - Routing metric
 - Routing protocol

Routing metric: ELD


- Objective of routing: choose routes that
 - minimize the end-to-end MAC latency to destination
 - minimize MAC latency per unit-distance to destination  Geography

- Routing metric

- MAC latency per unit-distance to destination (LD)



Routing metric: ELD

- Objective of routing: choose routes that
 - minimize the end-to-end MAC latency to destination
 - minimize MAC latency per unit-distance to destination  geography
- Routing metric
 - MAC latency per unit-distance to destination (**LD**)
- Routing decision (greedy)
 - Select a neighbor with the lowest **E[LD]**
 - Break ties by preferring stabler links (i.e., links with smaller variance)

Learn on the Fly (LOF): A data-driven routing protocol

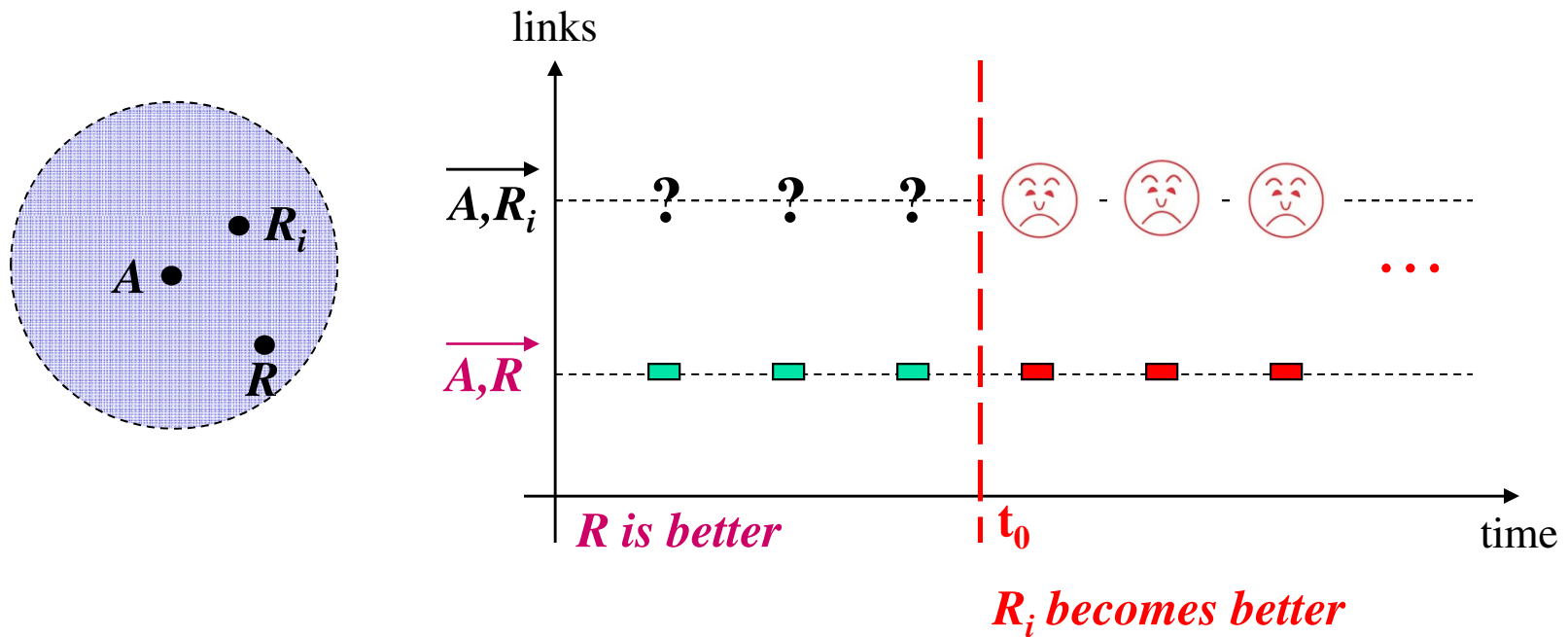
- Objective of routing protocol
 - Estimate ELD for each neighbor via data transmissions
 - Maintain the next-hop forwarder
- Protocol LOF
 - Upon booting up: neighborhood discovery, initial sampling
 - When forwarding data packets
 - Adapt estimation of ELDs via MAC feedback, and maintain forwarder according to dynamically estimated ELDs

Estimator: exponentially weighted moving average (EWMA)

$$\text{ELD} \leftarrow \alpha^{\beta(R)} \text{ELD} + (1 - \alpha^{\beta(R)}) \text{ELD}'$$
, $\beta(R)$ is an age factor
 - Blacklisting: identify extremely unreliable links/neighbors as *dead*

Uneven link sampling in data-driven estimation

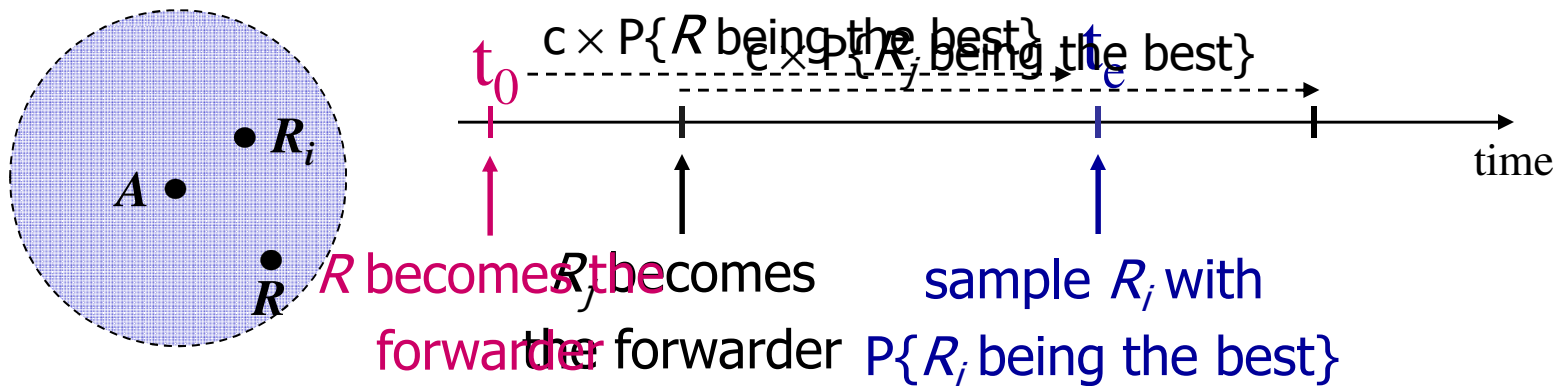
- A link will be sampled only if it is used as the forwarder



Better routes may not be used.

Countermeasure: exploratory sampling

- Proactively sample alternative neighbors, by using them as forwarders
 - Control the frequency to reflect the goodness of the current forwarder
 - Sample a neighbor with the probability of it being the best forwarder



Feasibility: sample size analysis

- We find that MAC latency and LD are log-normally distributed (Kolmogorov-Smirnov test)
 - That is, $\text{Log}(\text{LD})$ is normally distributed
- Sample size calculation (for comparing the means of two LDs):

Given

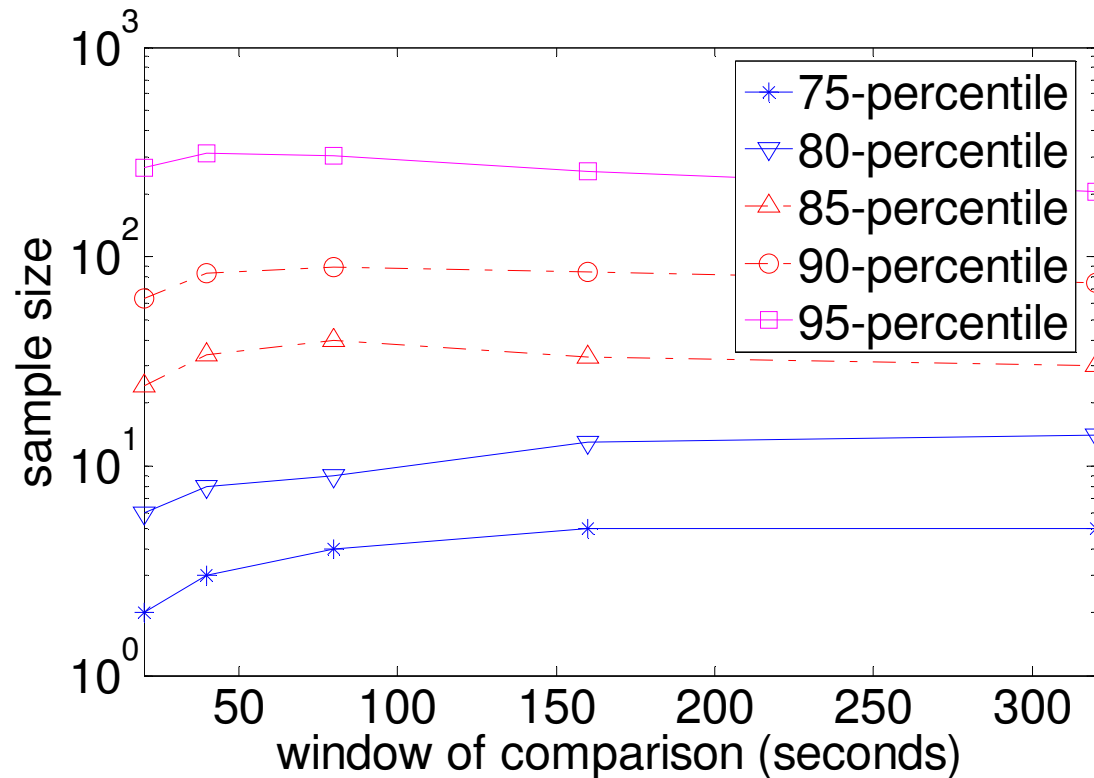
- normal variates: $\log(\text{LD}_1) \sim N(\mu_1, \delta_1^2)$, $\log(\text{LD}_2) \sim N(\mu_2, \delta_2^2)$
- confidence level of comparison: $100(1 - \alpha)\%$, $0 \leq \alpha \leq 1$

Sample size required is

$$\left(\frac{Z_\alpha (\delta_1 + \delta_2)}{\mu_1 - \mu_2} \right)^2$$

Z_α : α -quantile of a unit normal variate

Find out the best forwarder at 95% confidence level



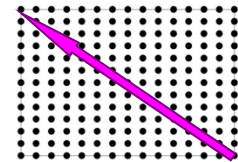
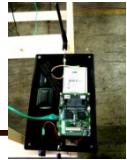
- Sample size requirement is small most of the time
 - 75-percentile: ~2
 - 80-percentile: ~6
- The above fact implies
 - quick routing convergence
 - the usefulness of initial link sampling at boot-up

Experiment design: protocols studied

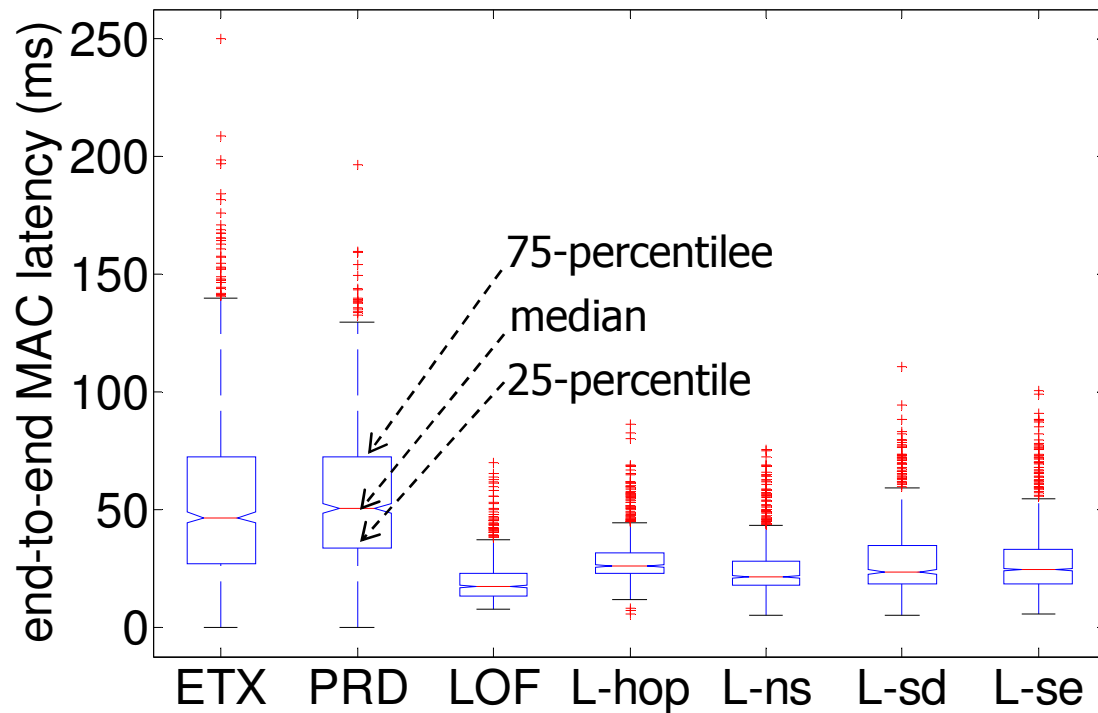
- Beacon-based routing
 - ETX: expected transmission count; geography unaware
(Alec Woo et al. 2003, Douglas Couto et al. 2003)
 - PRD: product of link reliability and distance progress; geography based
(Karim Seada et al., 2004)
- Other versions of LOF
 - L-ns: no exploratory sampling
 - L-sd: consider every neighbor in exploratory sampling (i.e., including dead neighbor)
 - L-se: try exploratory sampling after every packet transmission
 - L-hop: assume geographic-uniformity

Experiment design: evaluation method

- 802.11b testbed of Kansei
 - 15×13 grid
- Traffic flow
 - from the right-bottom corner to the upper-left corner
 - *ExSca*/traffic trace
 - 50 runs for each protocol ($50 \times 19 = 950$ packets)
- Evaluation criteria
 - End-to-end MAC latency
 - Energy efficiency
 - Links used in routing

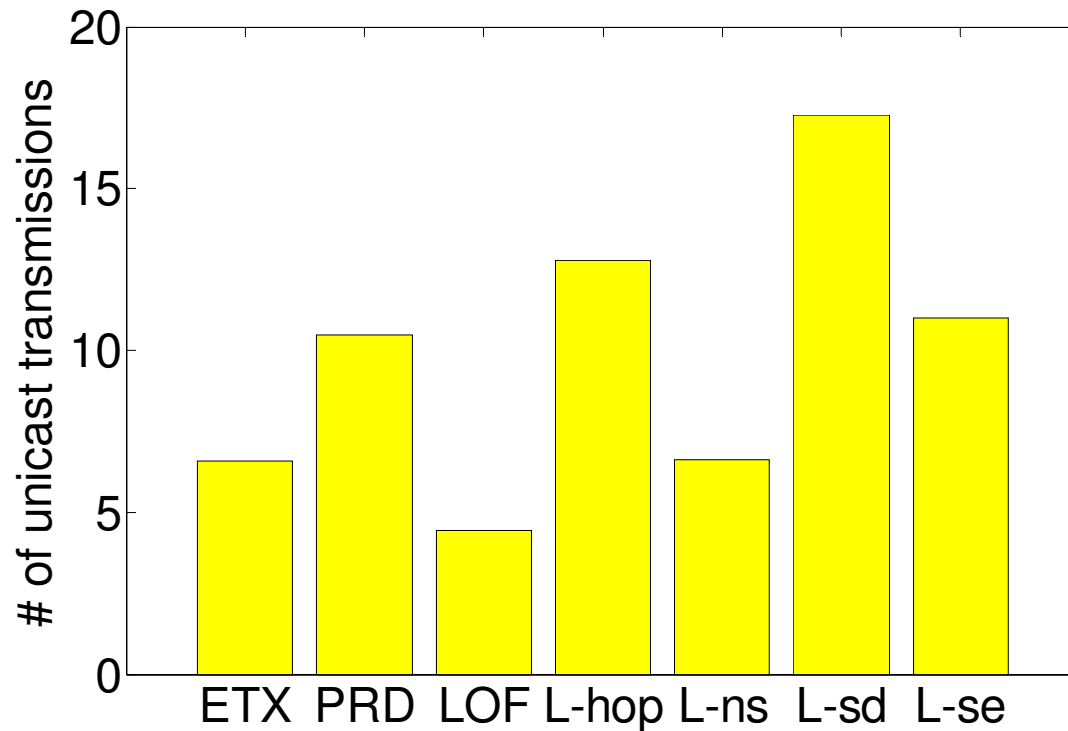


End-to-end MAC latency



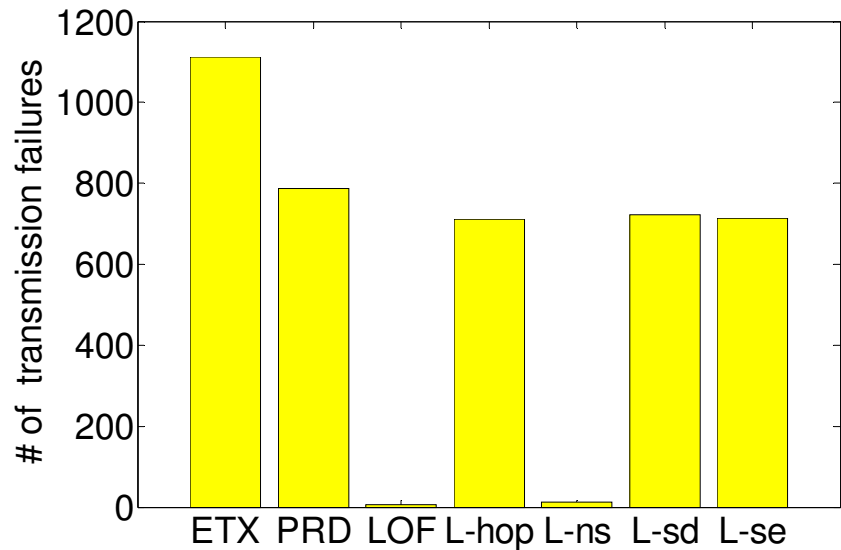
- Compared with ETX and PRD, LOF reduces MAC latency by a factor of 3
- LOF has the smallest MAC latency compared with L-*, showing the importance of
 - proper exploratory sampling
 - not assuming geographic uniformity

Average number of unicast transmissions per packet received

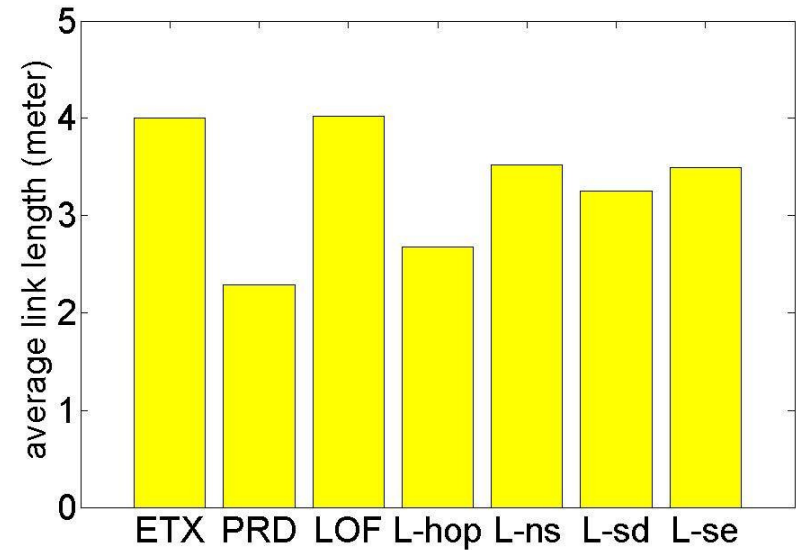


- Compared with ETX and PRD, LOF improves efficiency by a factor of 1.49 and 2.37 respectively
- LOF is more efficient than L-*

Links used: reliability and length



of transmission failures



average link length

- LOF uses reliable links
 - 1112 and 786 failures in ETX and PRD respectively; only 5 failures in LOF
- L-ns uses reliable but shorter links than LOF does

Other experiments

- Multiple event-traffic sources, periodic traffic
- Data-driven link estimation + distance-vector routing

Deployments



- 200 nodes 802.11b network of ExScal (2004)
- 802.11 backbone network of the 10,000 nodes sensor network being deployed by *Northrop Grumman Corporation* (since 2005)
- Kansei (a testbed for large scale wireless networking and sensing experimentation)

Summary: Learn-on-the-Fly (LOF) routing

- Beacon-based link estimation is imprecise
- Data-driven link estimation and routing
 - Circumvent the drawbacks and difficulty of beacon-based link estimation
 - Use reliable and long links, and improve network performance

Hongwei Zhang, Anish Arora, Prasun Sinha, [Link Estimation and Routing in Sensor Network Backbones: Beacon-based or Data-driven?](#), *IEEE Transactions on Mobile Computing (TMC)*, 8(5), May 2009