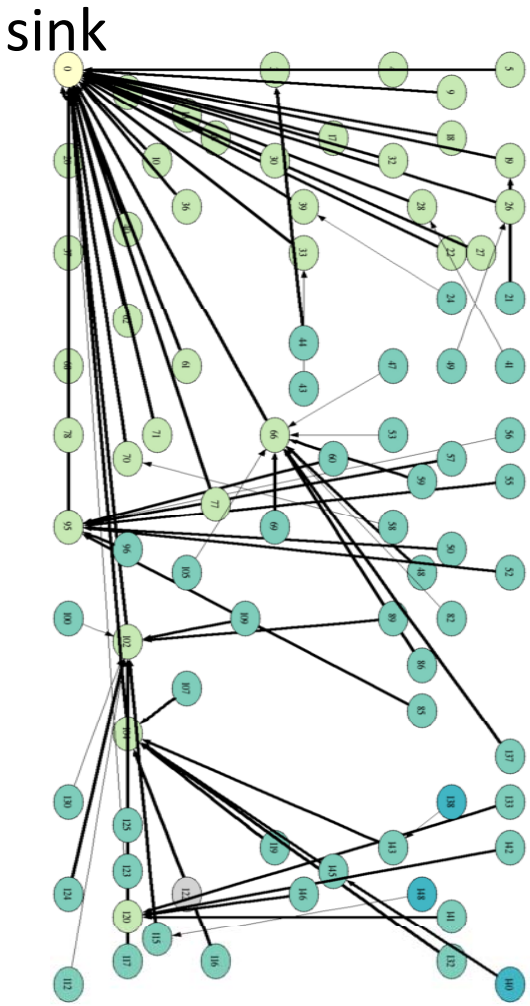
Collection Tree Protocol

Omprakash Gnawali (Stanford University)
with
Rodrigo Fonseca (Brown University)
Kyle Jamieson (University College London)
David Moss (People Power Company)
Philip Levis (Stanford University)

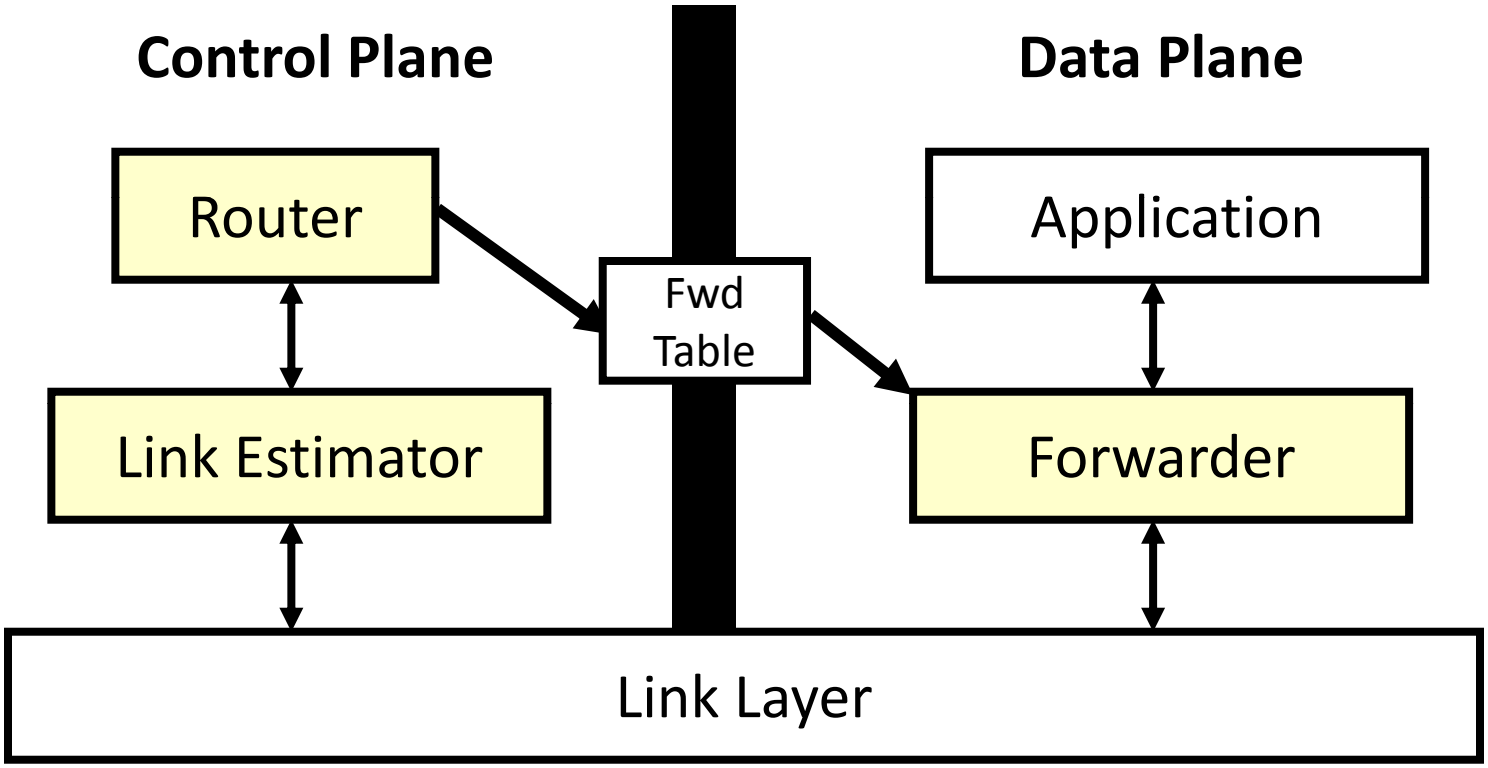
ACM SenSys
November 4, 2009

Collection

- Anycast route to the sink(s)
  - Used to collect data from the network to a small number of sinks (roots, base stations)
  - Network primitive for other protocols
- A distance vector protocol



Common Architecture



Prior Work

Control Plane

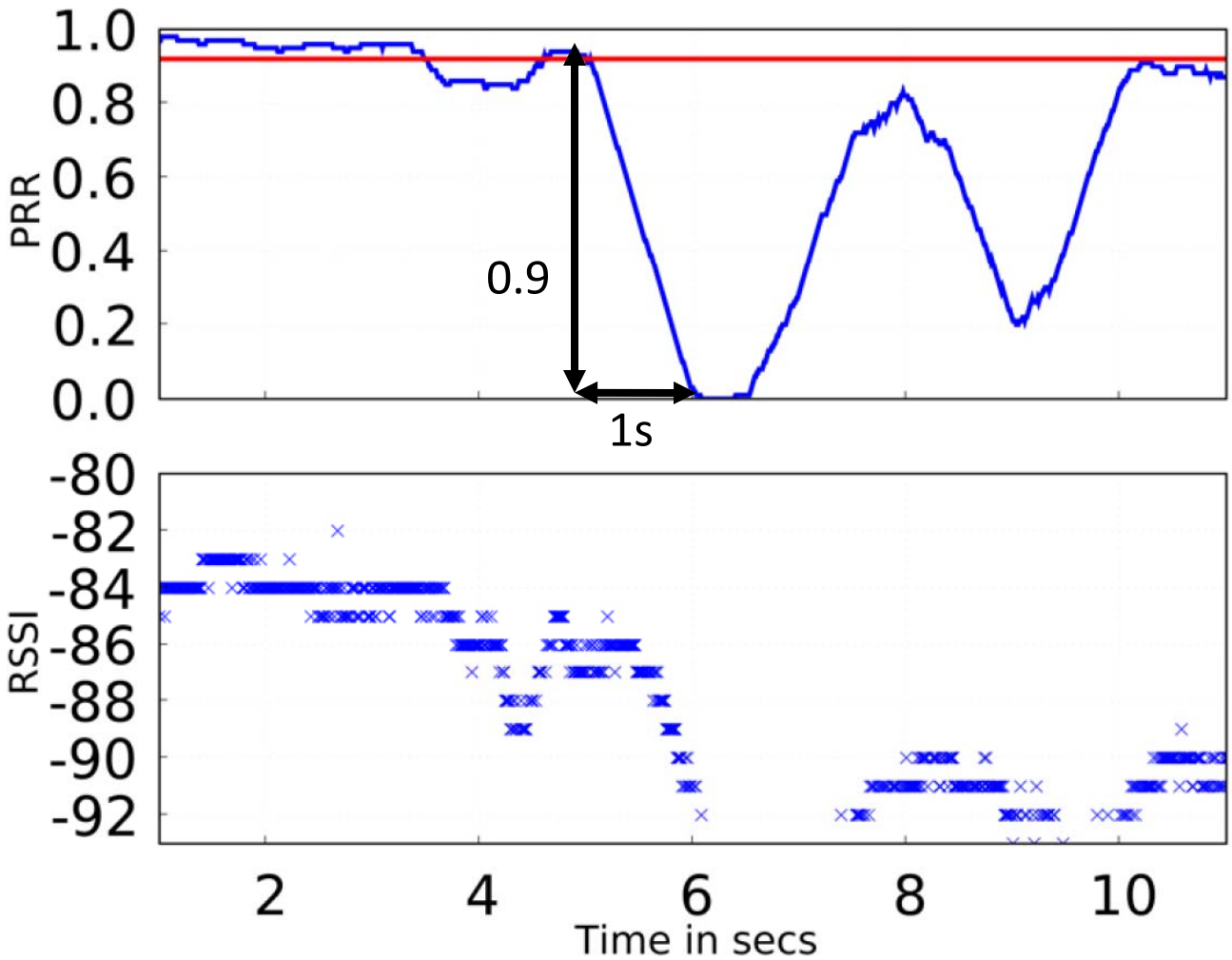
ETX, MT, MultiHopLQI, EAR, LOF, AODV, DSR, BGP, RIP, OSPF, Babel

Data Plane

Flush, RMST, CODA, Fusion, IFRC, RCRT

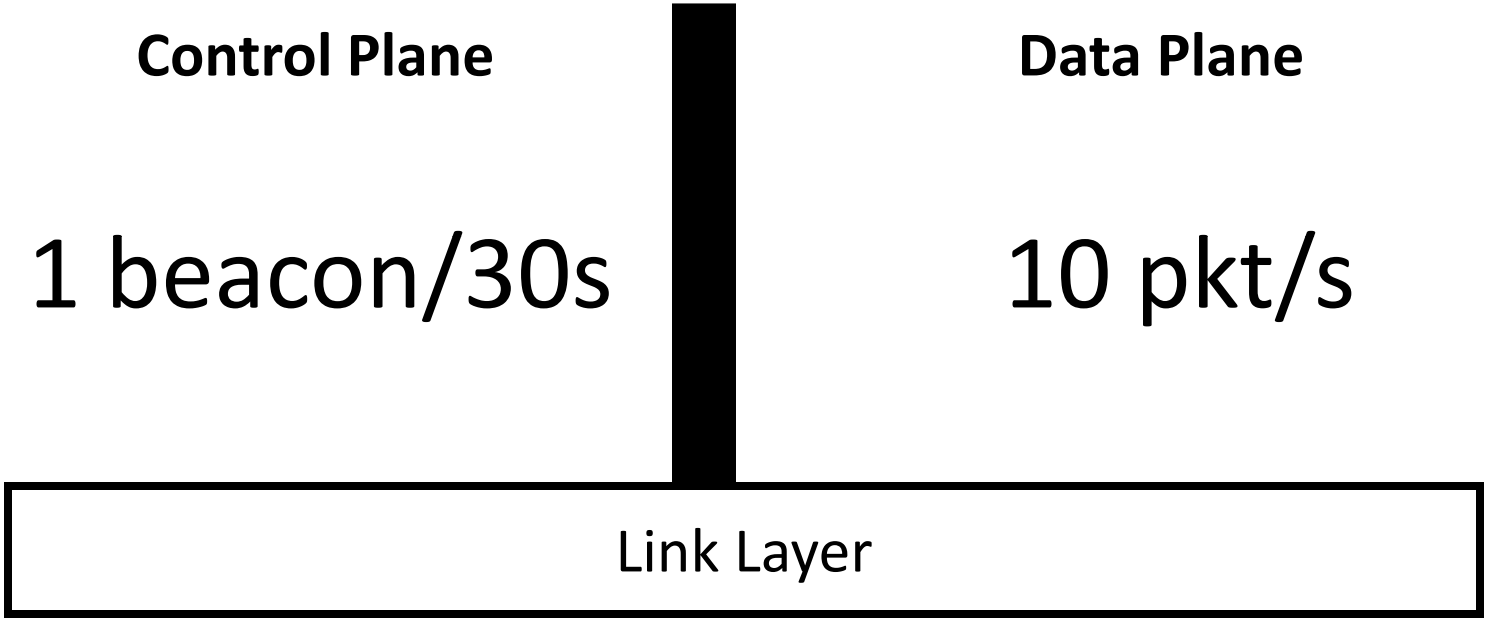
Link Layer

Wireless Link Dynamics



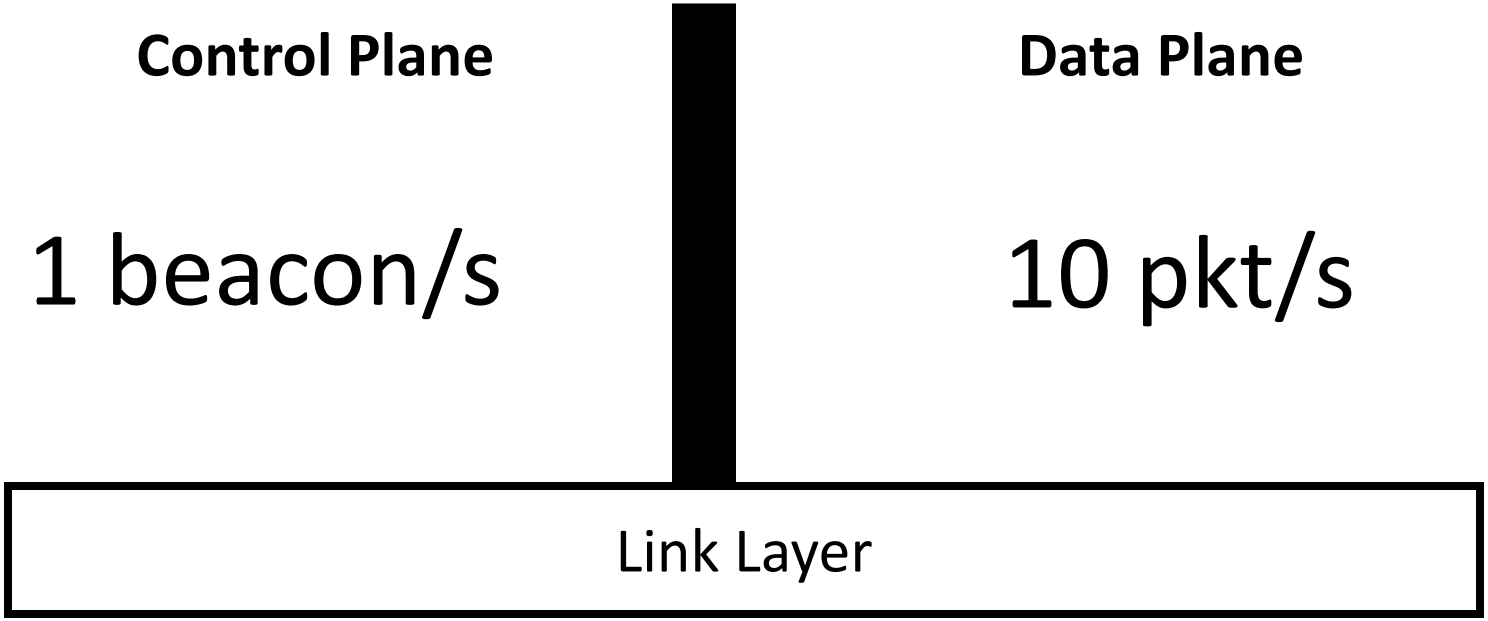
Control and Data Rate Mismatch

- Can lead to poor performance

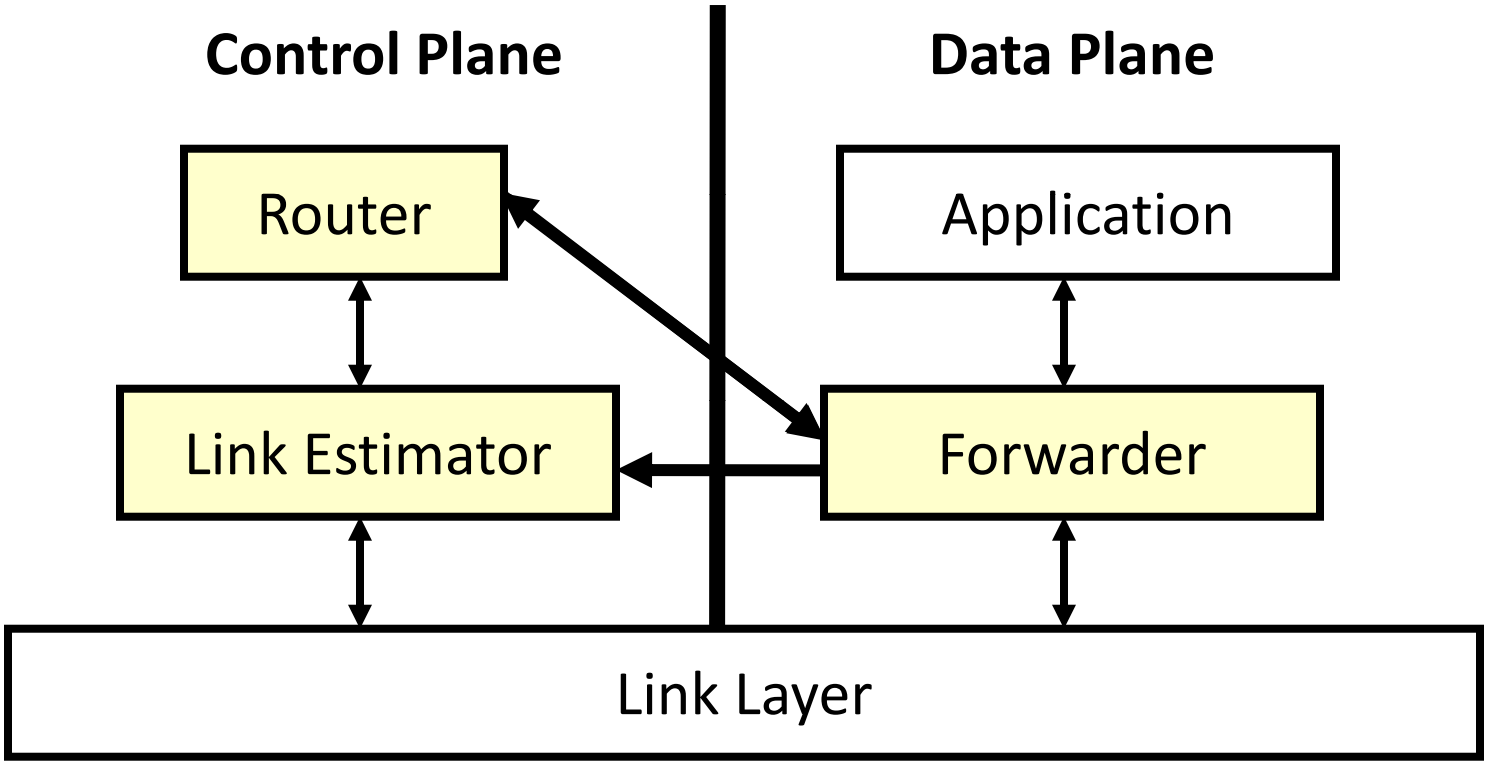


Control and Data Rate Mismatch

- Can lead to poor performance

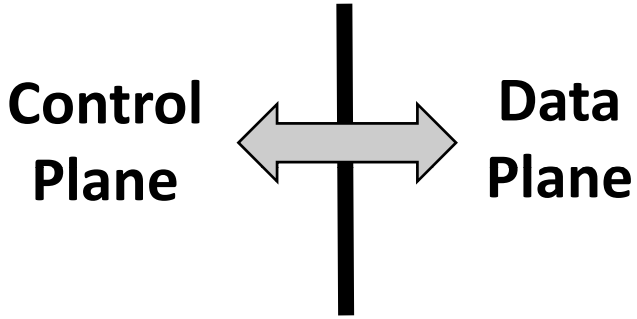


CTP Noe


Control Plane
Data Plane
Router
Application
Link Estimator
Forwarder
Link Layer

CTP Noe's Approach

- Enable control and data plane interaction
- Two mechanisms for efficient and agile topology maintenance
  - Datapath validation
  - Adaptive beaconing



Summary of Results

- 90-99.9% delivery ratio
  - Testbeds, configurations, link layers
- Compared to MultihopLQI
  - 29% lower data delivery cost
  - 73% fewer routing beacons
  - 99.8% lower loop detection latency
- Robust against disruption
- Cause for packet loss vary across testbeds

Outline

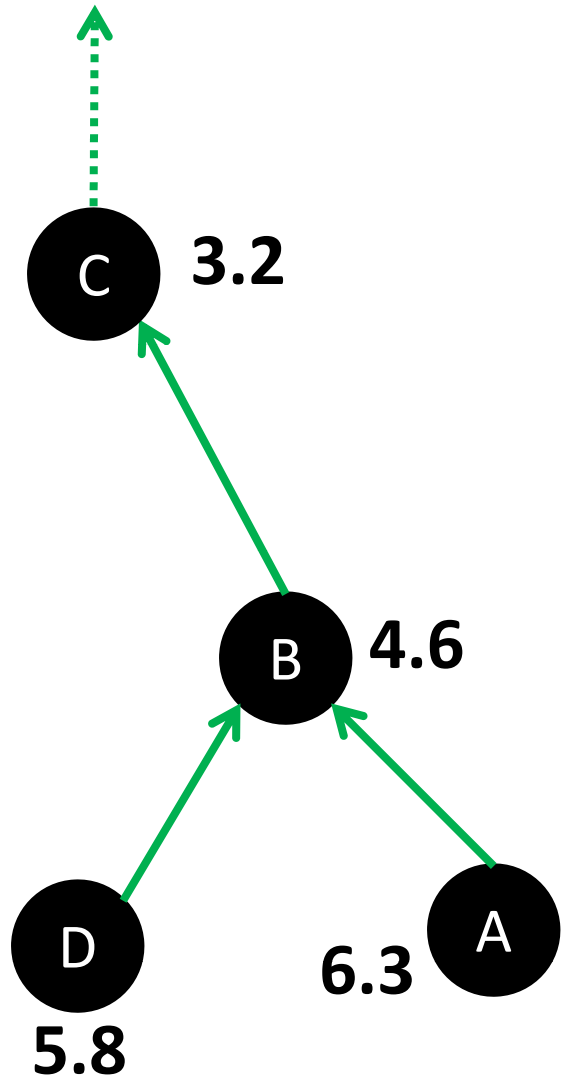
- Collection
- Datapath validation
- Adaptive beacons
- Evaluation
- Conclusion

Datapath validation

- Use data packets to validate the topology
  - Inconsistencies
  - Loops
- Receiver checks for consistency on each hop
  - Transmitter's cost is in the header
- Same time-scale as data packets
  - Validate only when necessary

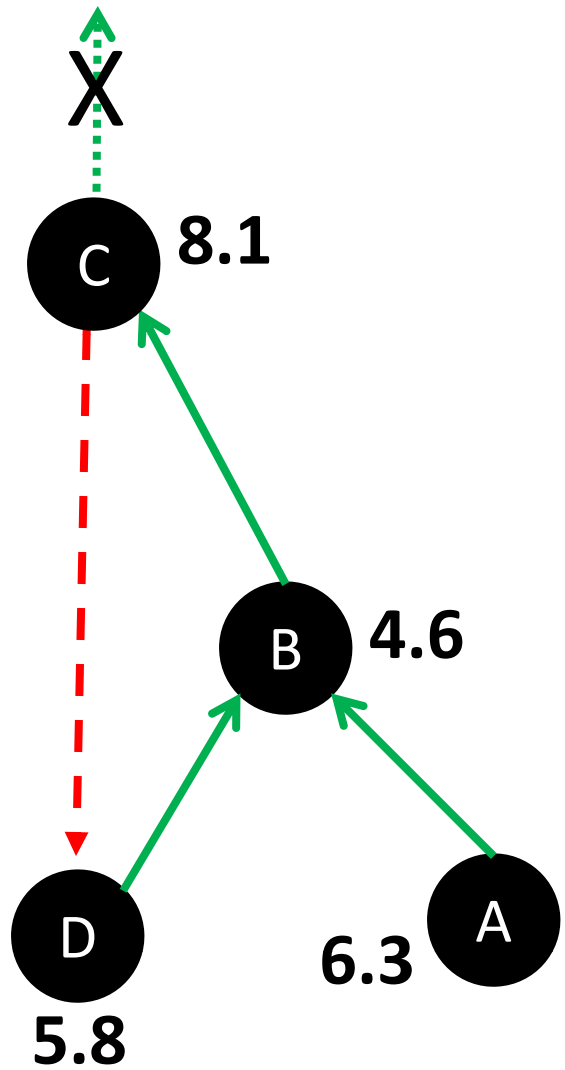
Routing Loops

– Cost does not decrease



Routing Loops

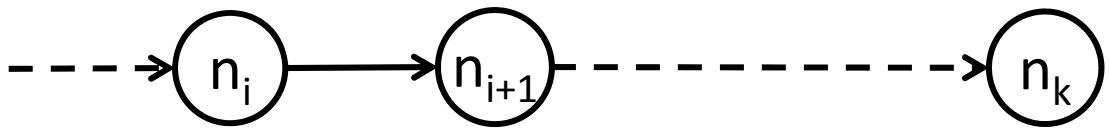
– Cost does not decrease



Routing Consistency

- *Next hop* should be closer to the destination
- Maintain this consistency criteria on a path

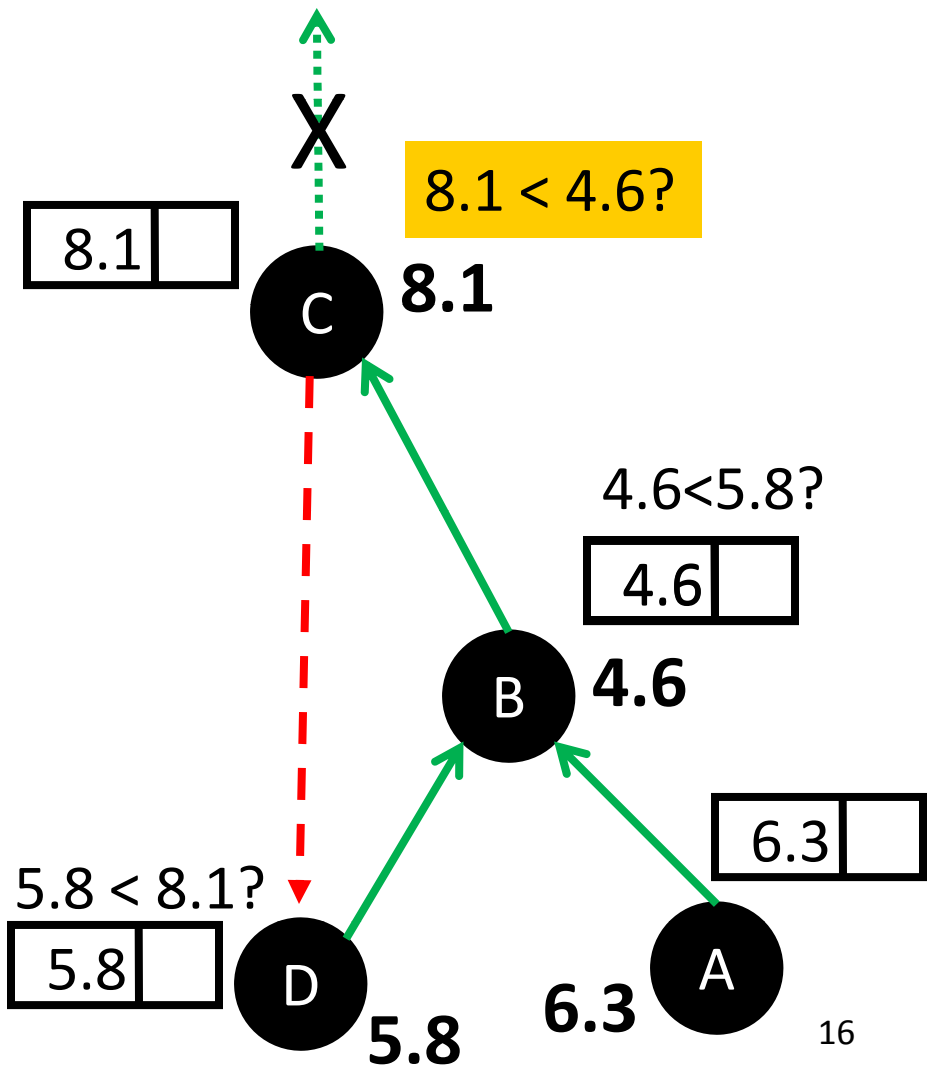
$$\forall i \in \{0, k-1\},\ ETX(n_i) > ETX(n_{i+1})$$



- Inconsistency due to stale state

Detecting Routing Loops

- Datapath validation
  - Cost in the packet
  - Receiver checks
- Inconsistency
  - Larger cost than on the packet
- On Inconsistency
  - Don't drop the packets
  - Signal the control plane



Outline

- Collection
- Datapath validation
- Adaptive beacons
- Evaluations
- Conclusion

How Fast to Send Beacons?

- Using a fixed rate beacon interval
  - Can be too fast
  - Can be too slow
  - Agility-efficiency tradeoff
- Agile+Efficient possible?

Routing as Consistency

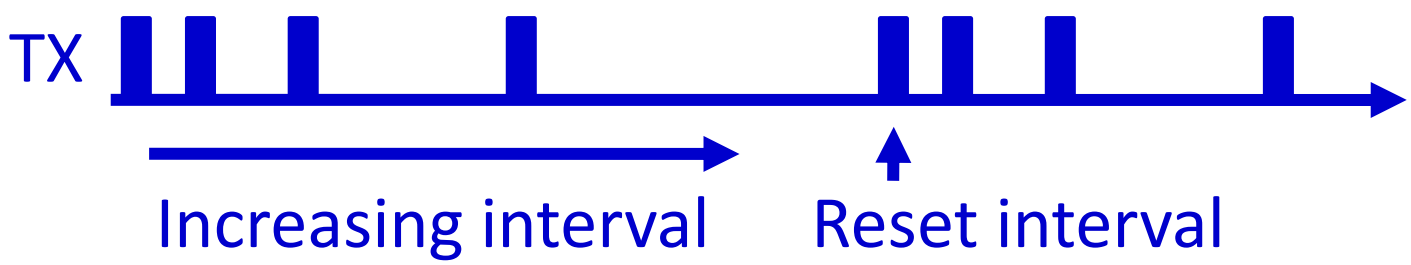
- Routing as a consistency problem
  - Costs along a path must be consistent
- Use consistency protocol in routing
  - Leverage research on consistency protocols
  - Trickle

Trickle

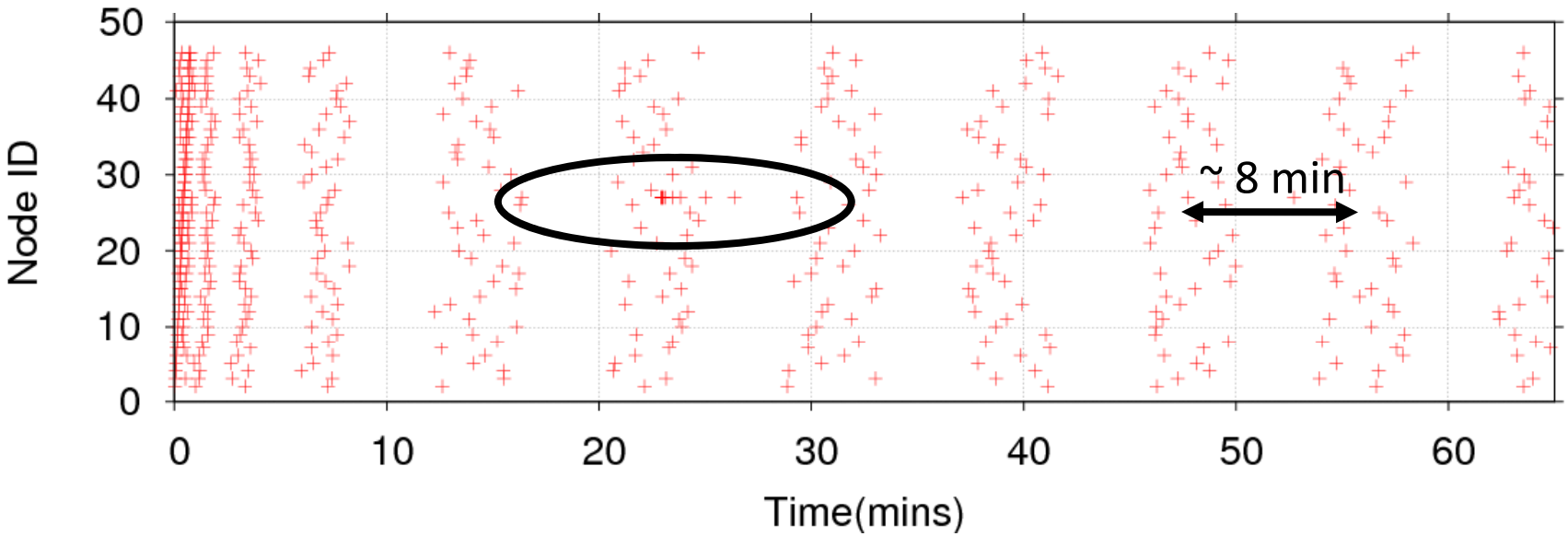
- Detecting inconsistency
  - Code propagation: Version number mismatch
  - Does not work for routing: use path consistency
- Control propagation rate
  - Start with a small interval
  - Double the interval up to some max
  - Reset to the small interval when inconsistent

Control Traffic Timing

- Extend Trickle to time routing beacons
- Reset the interval
  - ETX(receiver) >= ETX(sender)
  - Significant decrease in gradient
  - “Pull” bit



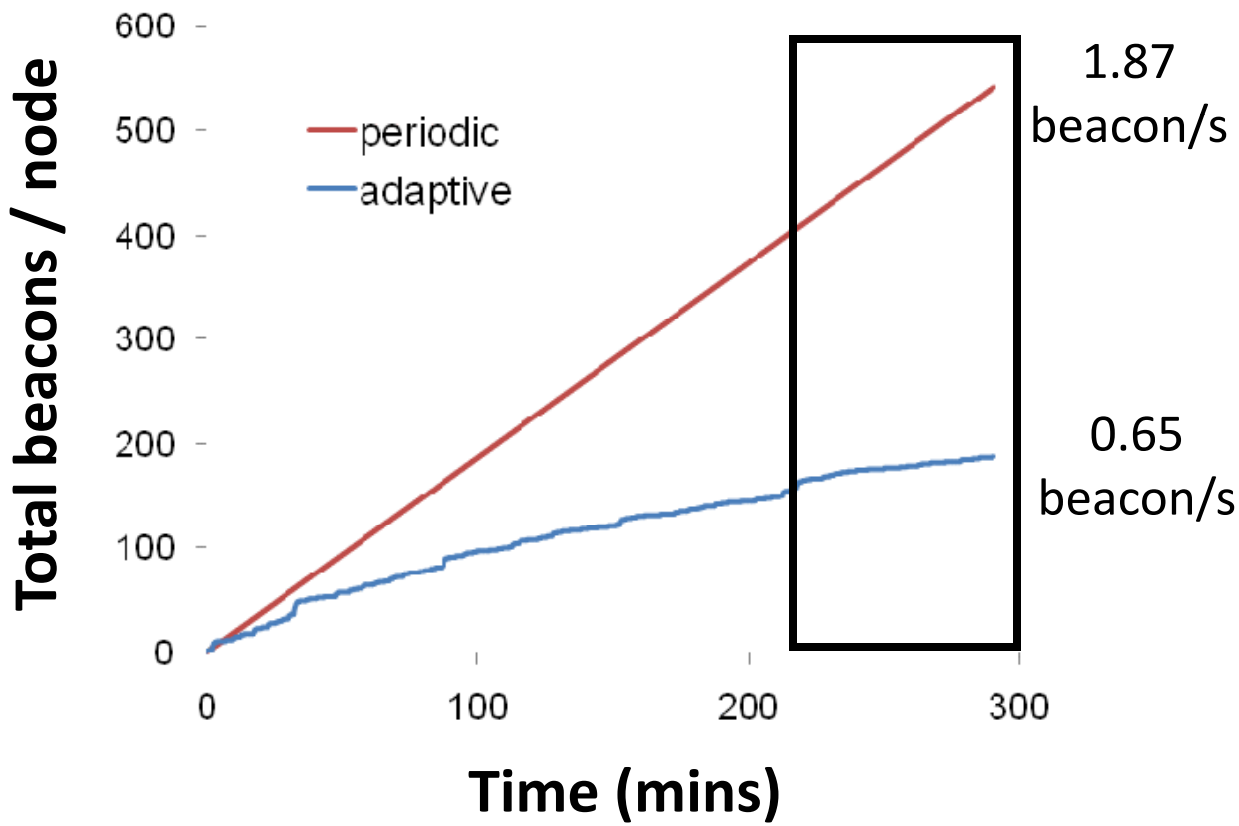
Adaptive Beacon Timing



Tutornet

Infrequent beacons in the long run

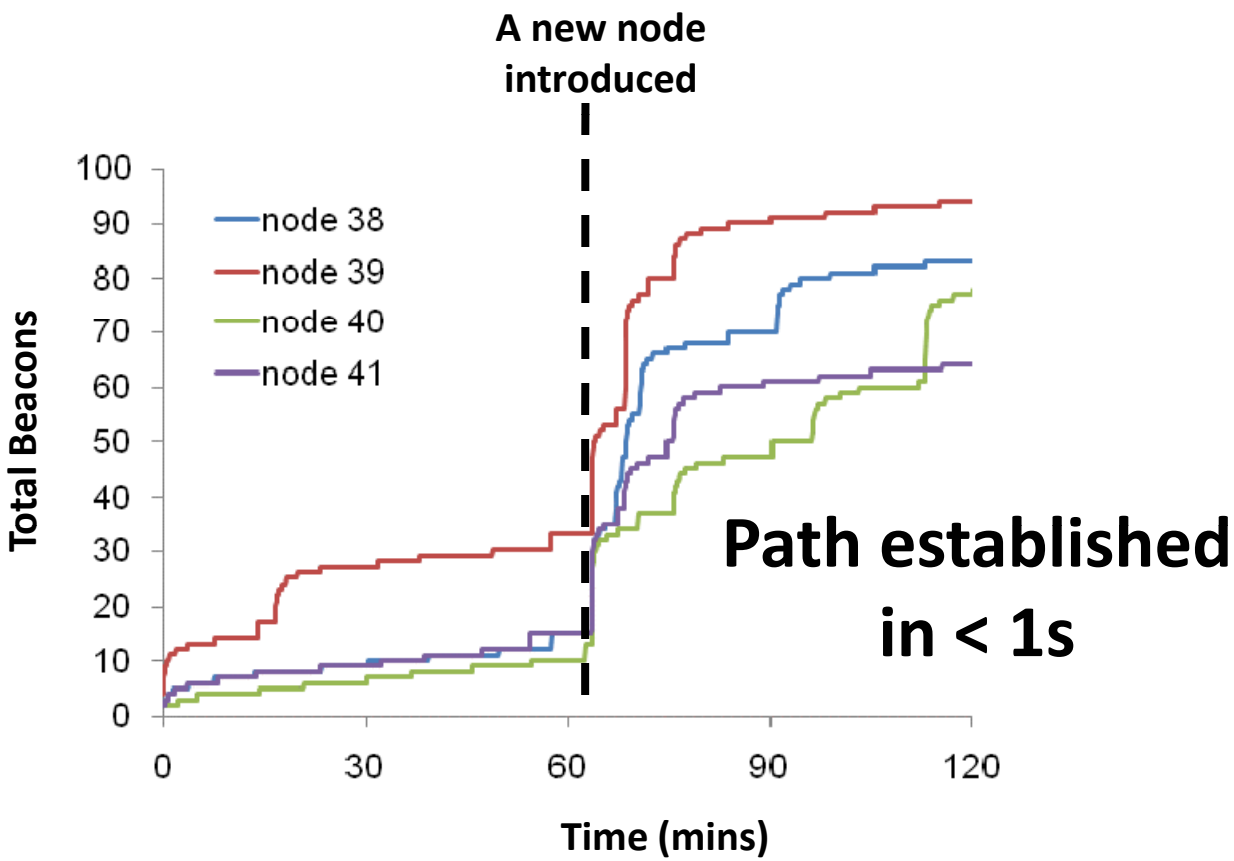
Adaptive vs Periodic Beacons



Tutornet

Less overhead compared to 30s-periodic

Node Discovery



Tutornet

Efficient and agile at the same time

Outline

- Collection
- Datapath validation
- Adaptive beacons
- Evaluation
- Conclusion

Experiments

- 12 testbeds
- 20-310 nodes
- 7 hardware platforms
- 4 radio technologies
- 6 link layers

Testbed	Platform	Nodes	Physical size m^2 or m^3
Tutornet	Tmote	91	$50 \times 25 \times 10$
Wymanpark	Tmote	47	80×10
Motelab	Tmote	131	$40 \times 20 \times 15$
Kansei	TelosB	310	40×20
Mirage	Mica2dot	35	50×20
NetEye	Tmote	125	6×4
Mirage	MicaZ	86	50×20
Quanto	Epic-Quanto	49	35×30
Twist	Tmote	100	$30 \times 13 \times 17$
Twist	eyesIFXv2	102	$30 \times 13 \times 17$
Vinelab	Tmote	48	60×30
Blaze	Blaze	20	30×30

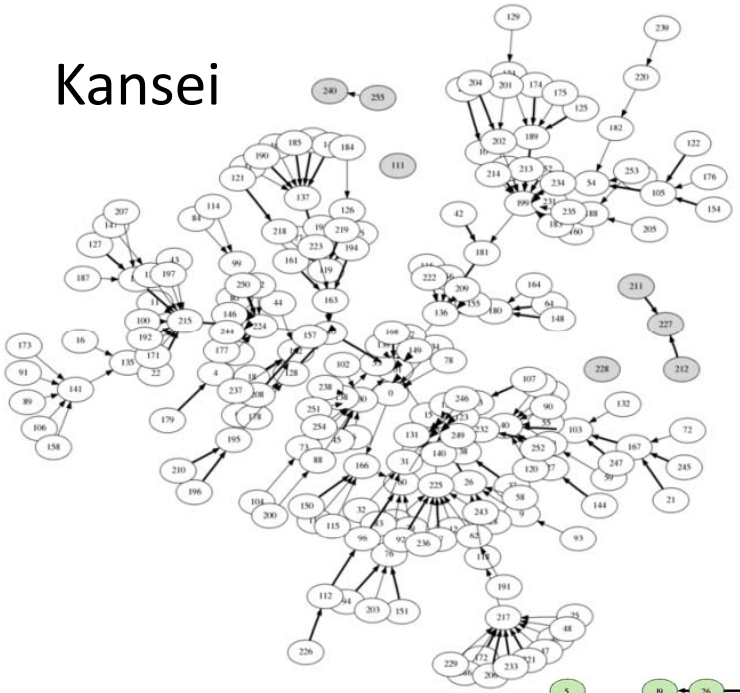
Variations in hardware, software, RF environment, and topology

Evaluation Goals

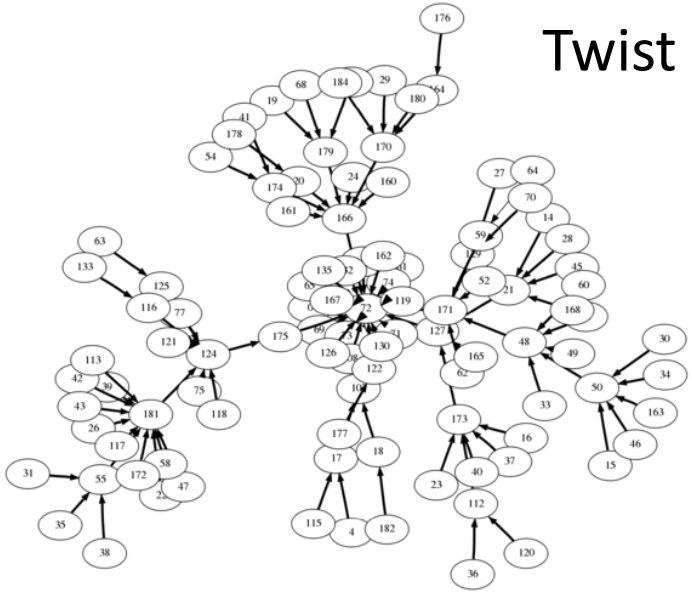
- Reliable?
  - Packets delivered to the sink
- Efficient?
  - TX required per packet delivery
- Robust?
  - Performance with disruption

CTP Noe Trees

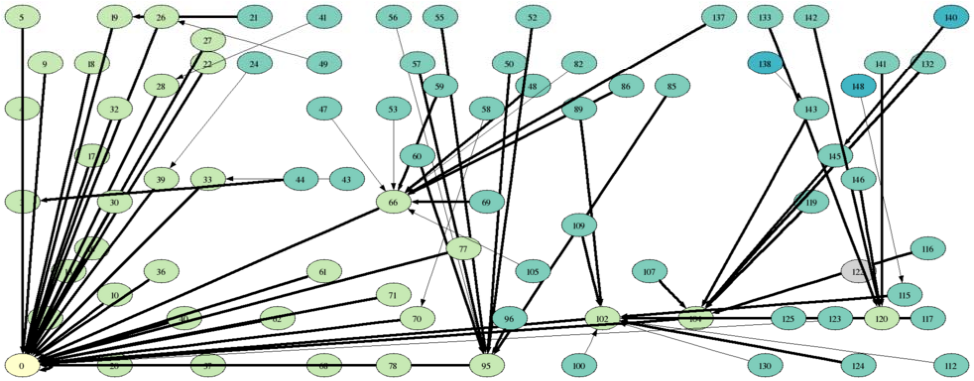
Kansei



Twist



Mirage



Reliable, Efficient, and Robust

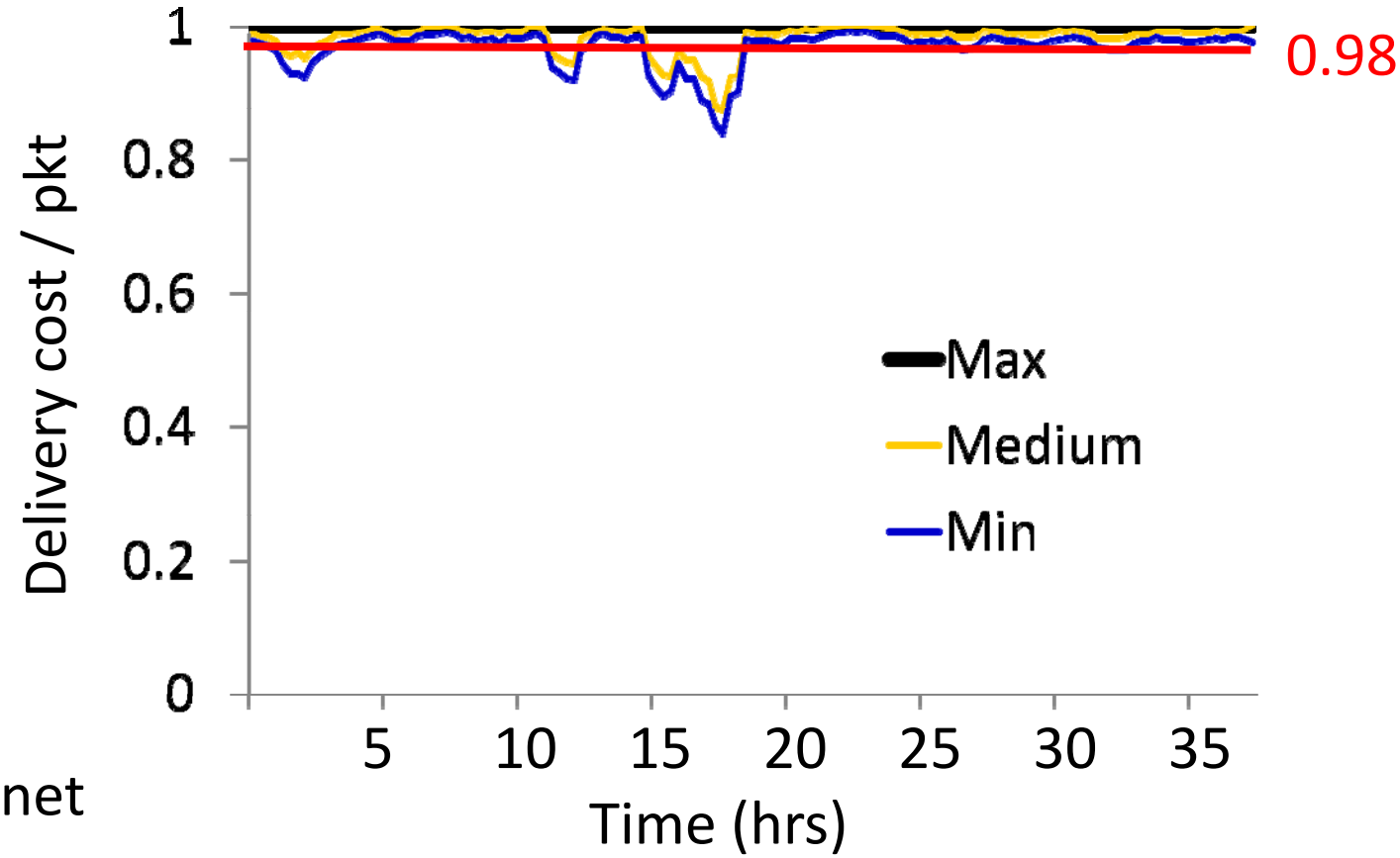
Testbed	Delivery Ratio
Wymanpark	0.9999
Vinelab	0.9999
Tutornet	0.9999
NetEye	0.9999
Kansei	0.9998
Mirage-MicaZ	0.9998
Quanto	0.9995
Blaze	0.9990
Twist-Tmote	0.9929
Mirage-Mica2dot	0.9895
Twist-eyesIFXv2	0.9836
Motelab	0.9607

False ack

Retransmit

High end-to-end delivery ratio
(but not on all the testbeds!)

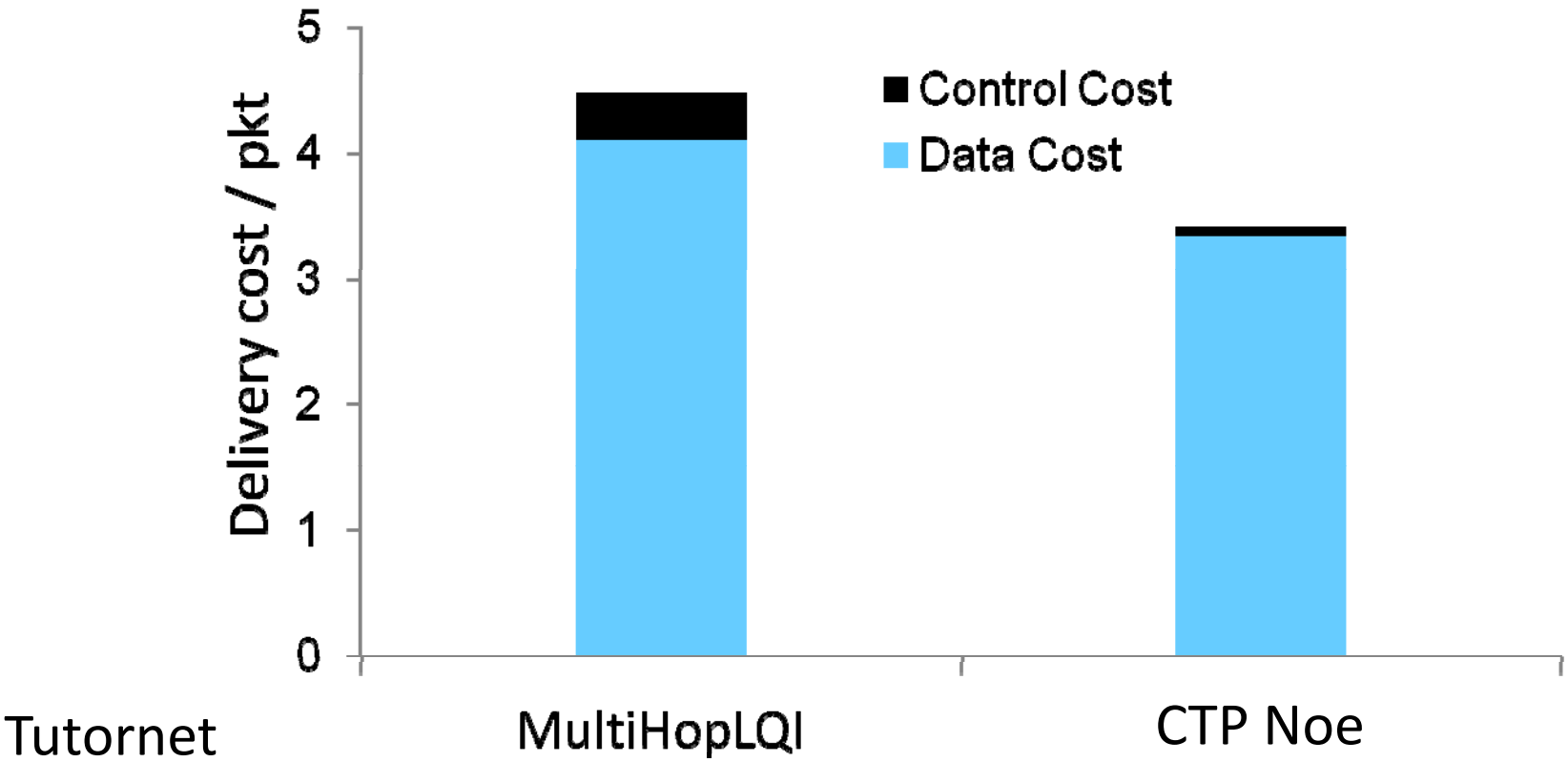
Reliable, Efficient, and Robust



Tutornet

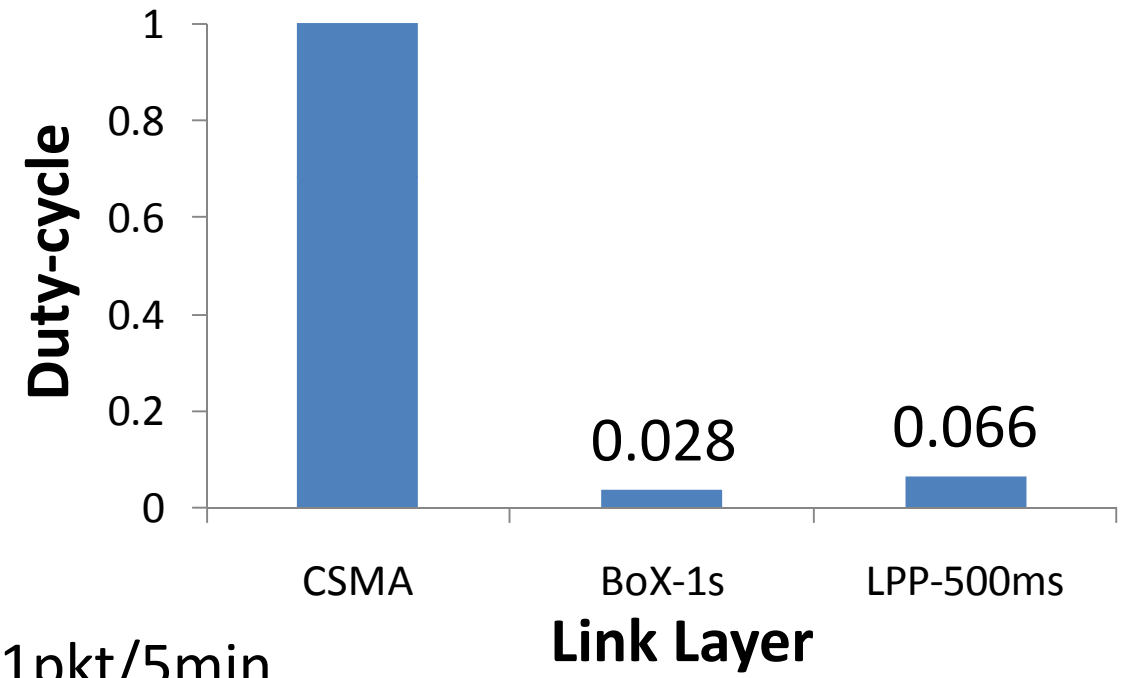
High delivery ratio across time
(short experiments can be misleading!)

Reliable, Efficient, and Robust



Low data and control cost

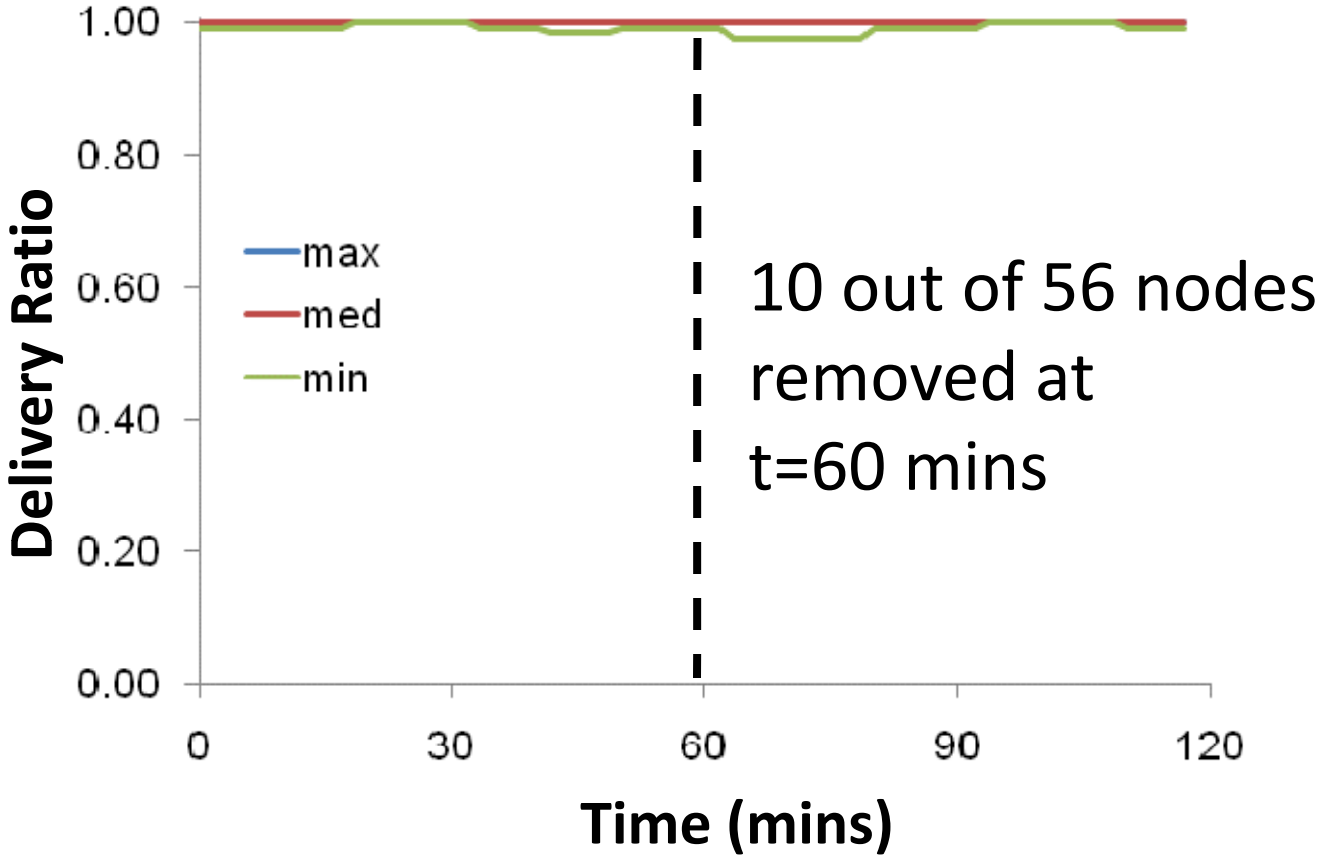
Reliable, Efficient, and Robust



Motelab, 1pkt/5min

Low duty-cycle with low-power MACs

Reliable, Efficient, and Robust



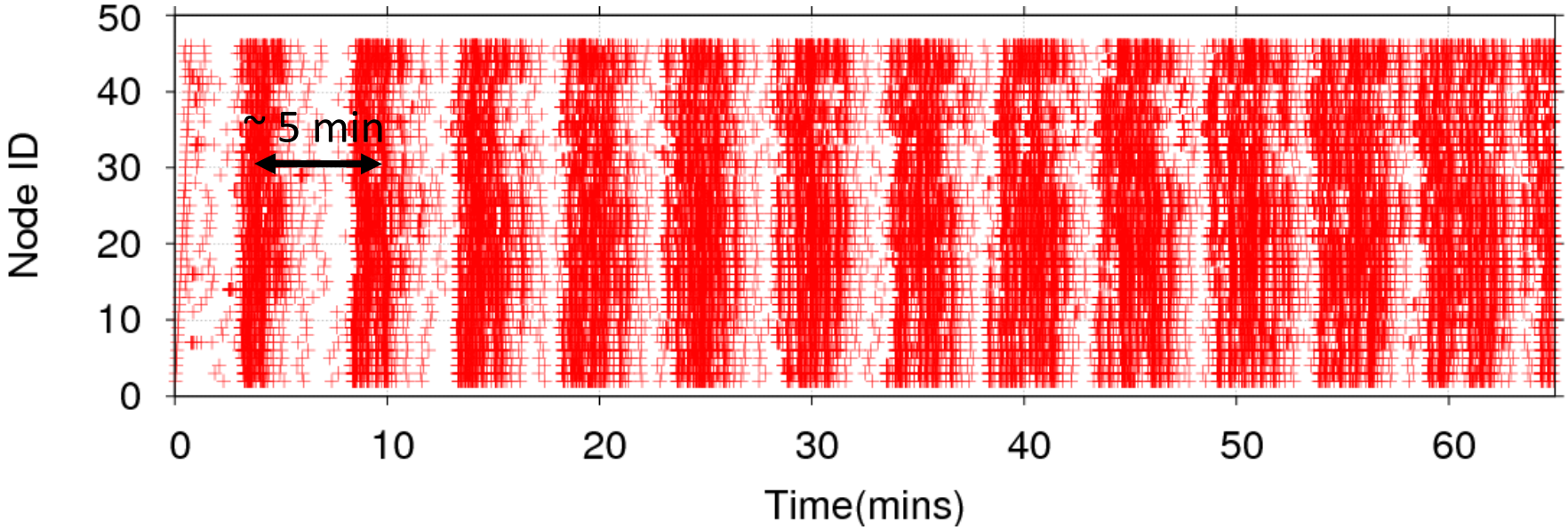
Tutornet

No disruption in packet delivery

Reliable, Efficient, and Robust

Nodes reboot every 5 mins

Routing Beacons



Tutornet

Delivery Ratio > 0.99

High delivery ratio despite serious network-wide disruption (most loss due to reboot while buffering packet)

CTP Noe Performance Summary

- Reliability
  - Delivery ratio > 90% in all cases
- Efficiency
  - Low cost and 5% duty cycle
- Robustness
  - Functional despite network disruptions

Acknowledgment

For testbed access and experiment help

- Anish Arora
- Geoffrey Werner Challen
- Prabal Dutta
- David Gay
- Stephen Dawson-Haggerty
- Timothy Hnat
- Ki-Young Jang
- Xi Ju
- Andreas Köpke
- Razvan Musaloiu-E.
- Vinayak Naik
- Rajiv Ramnath
- Mukundan Sridharan
- Matt Welsh
- Kamin Whitehouse
- Hongwei Zhang

For bug reports, fixes, and discussions

- Mehmet Akif Antepli
- Juan Batiz-Benet
- Jonathan Hui
- Scott Moeller
- Remi Ville
- Alec Woo
- and many others…

Thank You!

Conclusion

- "Hard" networks → good protocols
  - Tutornet & Motelab
- Wireless routing benefits from data and control plane interaction
- Lessons applicable to distance vector routing
  - Datapath validation & adaptive beaconing

Data trace from all the testbeds available at

http://sing.stanford.edu/gnawali/ctp/