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

Control Plane
- Router
- Link Estimator

Data Plane
- Application
- Forwarder

Fwd Table

Link Layer
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

![Graph showing Wireless Link Dynamics](image)
Control and Data Rate Mismatch

- Can lead to poor performance

Control Plane

Data Plane

1 beacon/30s

10 pkt/s

Link Layer
Control and Data Rate Mismatch

- Can lead to poor performance

![Diagram showing control plane with 1 beacon/s and data plane with 10 pkt/s]
CTP Noe

Control Plane

Router

Link Estimator

Data Plane

Application

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
  - $\text{ETX(}\text{receiver}) \geq \text{ETX(}\text{sender})$
  - Significant decrease in gradient
  - “Pull” bit

![Diagram showing TX, Increasing interval, and Reset interval]
Adaptive Beacon Timing

Tutornet

Infrequent beacons in the long run
Adaptive vs Periodic Beacons

Less overhead compared to 30s-periodic

Tutornet
Node Discovery

A new node introduced

Path established in < 1s

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

<table>
<thead>
<tr>
<th>Testbed</th>
<th>Platform</th>
<th>Nodes</th>
<th>Physical size m² or m³</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutornet</td>
<td>Tmote</td>
<td>91</td>
<td>50×25×10</td>
</tr>
<tr>
<td>Wymanpark</td>
<td>Tmote</td>
<td>47</td>
<td>80×10</td>
</tr>
<tr>
<td>Motelab</td>
<td>Tmote</td>
<td>131</td>
<td>40×20×15</td>
</tr>
<tr>
<td>Kansei</td>
<td>TelosB</td>
<td>310</td>
<td>40×20</td>
</tr>
<tr>
<td>Mirage</td>
<td>Mica2dot</td>
<td>35</td>
<td>50×20</td>
</tr>
<tr>
<td>NetEye</td>
<td>Tmote</td>
<td>125</td>
<td>6×4</td>
</tr>
<tr>
<td>Mirage</td>
<td>MicaZ</td>
<td>86</td>
<td>50×20</td>
</tr>
<tr>
<td>Quanto</td>
<td>Epic-Quanto</td>
<td>49</td>
<td>35×30</td>
</tr>
<tr>
<td>Twist</td>
<td>Tmote</td>
<td>100</td>
<td>30×13×17</td>
</tr>
<tr>
<td>Twist</td>
<td>eyesIFXv2</td>
<td>102</td>
<td>30×13×17</td>
</tr>
<tr>
<td>Vinelab</td>
<td>Tmote</td>
<td>48</td>
<td>60×30</td>
</tr>
<tr>
<td>Blaze</td>
<td>Blaze</td>
<td>20</td>
<td>30×30</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Testbed</th>
<th>Delivery Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wymanpark</td>
<td>0.9999</td>
</tr>
<tr>
<td>Vinelab</td>
<td>0.9999</td>
</tr>
<tr>
<td>Tutornet</td>
<td>0.9999</td>
</tr>
<tr>
<td>NetEye</td>
<td>0.9999</td>
</tr>
<tr>
<td>Kansei</td>
<td>0.9998</td>
</tr>
<tr>
<td>Mirage-MicaZ</td>
<td>0.9998</td>
</tr>
<tr>
<td>Quanto</td>
<td>0.9995</td>
</tr>
<tr>
<td>Blaze</td>
<td>0.9990</td>
</tr>
<tr>
<td>Twist-Tmote</td>
<td>0.9929</td>
</tr>
<tr>
<td>Mirage-Mica2dot</td>
<td>0.9895</td>
</tr>
<tr>
<td>Twist-eyesIFXv2</td>
<td>0.9836</td>
</tr>
<tr>
<td>Motelab</td>
<td>0.9607</td>
</tr>
</tbody>
</table>

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

False ack
Retransmit
Reliable, Efficient, and Robust

High delivery ratio across time (short experiments can be misleading!)
Reliable, **Efficient**, and Robust

![Graph showing data and control cost comparison]

- **Tutornet**
  - Low data and control cost

- **MultiHopLQI**

- **CTP Noe**

**Legend:**
- Control Cost
- Data Cost
Reliable, **Efficient**, and Robust

Motelab, 1pkt/5min

**Low duty-cycle with low-power MACs**
Reliable, Efficient, and Robust

10 out of 56 nodes removed at t=60 mins

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 $\rightarrow$ 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/