ContikiRPL and TinyRPL: Happy Together

JeongGil Ko
Joakim Eriksson
Nicolas Tsiftes
Stephen Dawson-Haggerty
Andreas Terzis
Adam Dunkels
David Culler

IP+SN 2011

Overview

- WSN Interoperability
- Goal/Contributions
- IPv6 in Contiki and TinyOS
- Evaluation
- Lessons Learned
- Future Work & Conclusions

Interoperability of WSN Systems

- For widespread commercial adoption of WSN systems (e.g., smart grid), achieving interoperability between different platforms is essential
- The ***performance*** of interoperable systems are equally important

WSNs of Today

- WSN systems are mostly single-purpose, homogeneous systems running the same software platform
- Incompatible protocols and network architectures

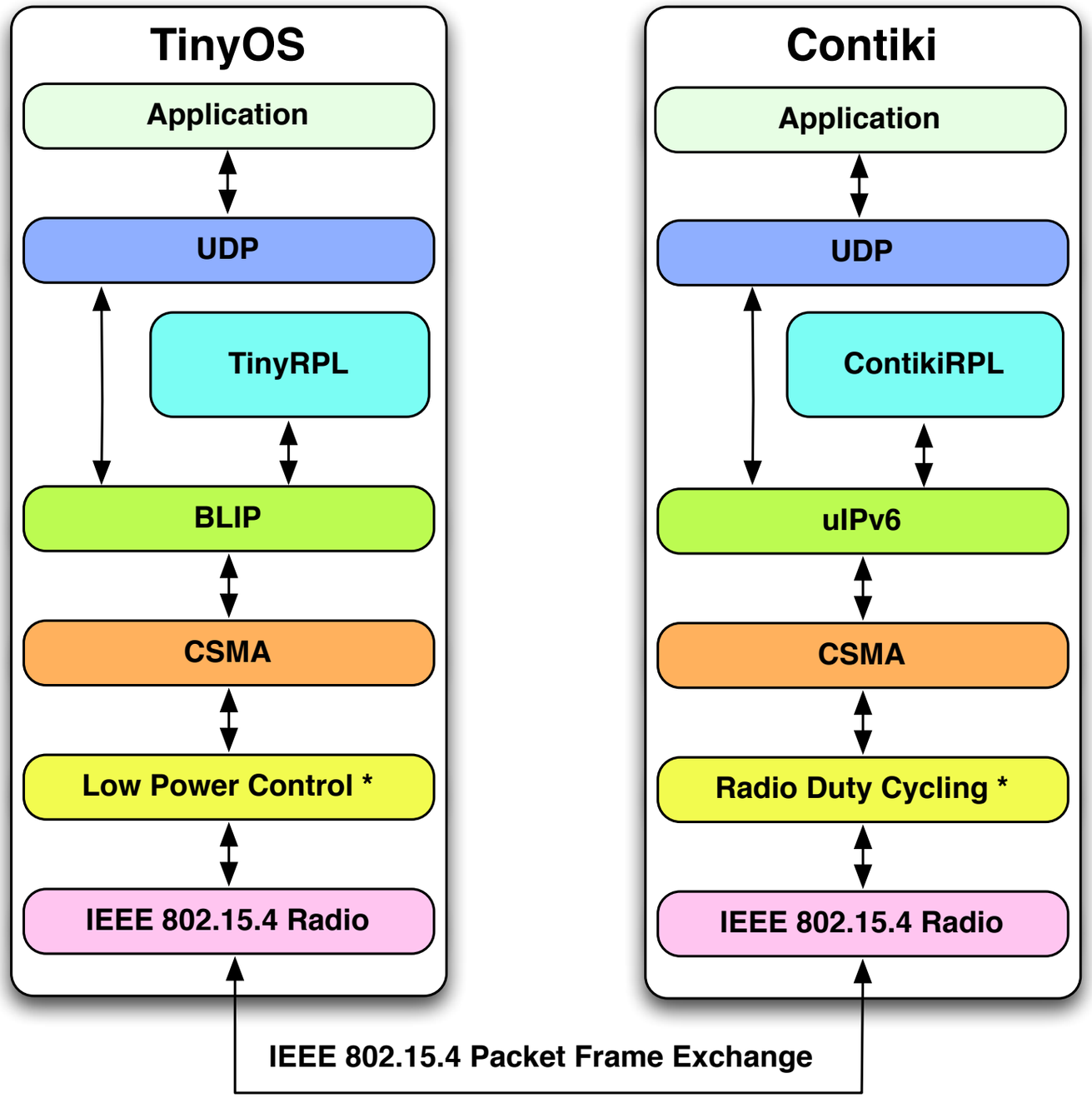
IP-based WSNs

- The IP architecture has successfully integrated various network technologies into the Internet
- With IP-based architectures WSNs can also be fully integrated to the Internet
- Theoretically, the IP architecture should allow WSNs to interoperate

Contributions

- Test IPv6 interoperability between IPv6 stacks of TinyOS and Contiki
  - IETF 6LoWPAN, IETF RPL, IEEE 802.15.4
- Evaluate the performance of the heterogeneous network
  - Although the two systems may interoperate, variations in implementation choices and system components can affect the end-system's performance

Contiki/TinyOS Interoperability



* Both software stacks have the capability of supporting a low power MAC. However, they are disabled for our evaluations presented in this work.

Contiki IPv6 Stack

- **uIPv6 + ContikiRPL**
- **uIPv6**
  - IPv6 Stack + 6LoWPAN HC/ND/Fragmentation
  - Packet forwarding control
- **ContikiRPL**
  - IETF RPL implementations in Contiki
  - Connects with uIPv6
  - Modular design
  - OF0, MRHOF
  - Controls routing decisions

TinyOS IPv6 Stack

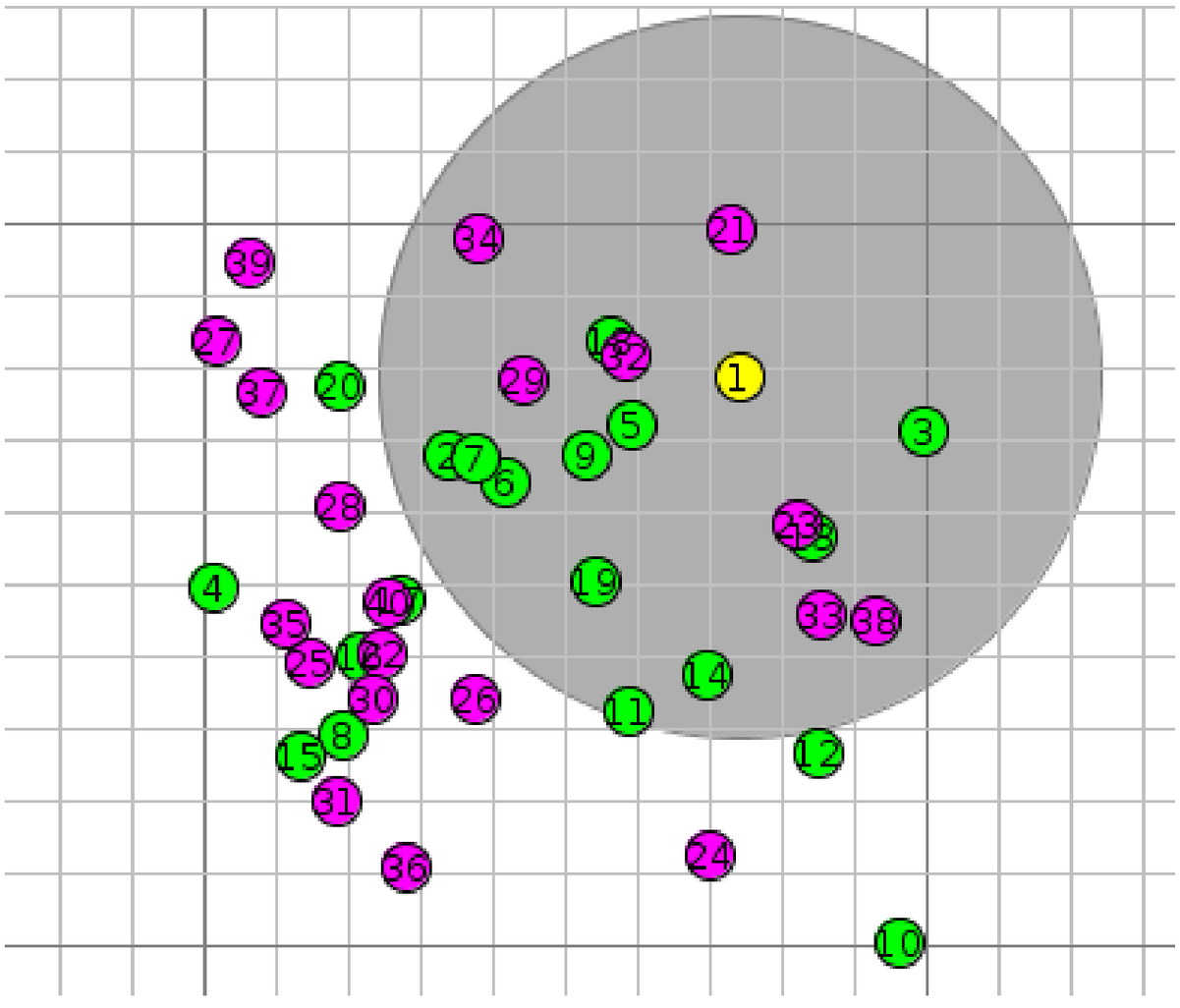
- **BLIP + TinyRPL**
- **BLIP**
  - 6LoWPAN HC/Fragmentation
  - PPP connection support
  - Packet forwarding management
- **TinyRPL**
  - IETF RPL implementations in TinyOS
  - Connects heavily with BLIP
  - OF0, MRHOF
  - Routing path control

Evaluation

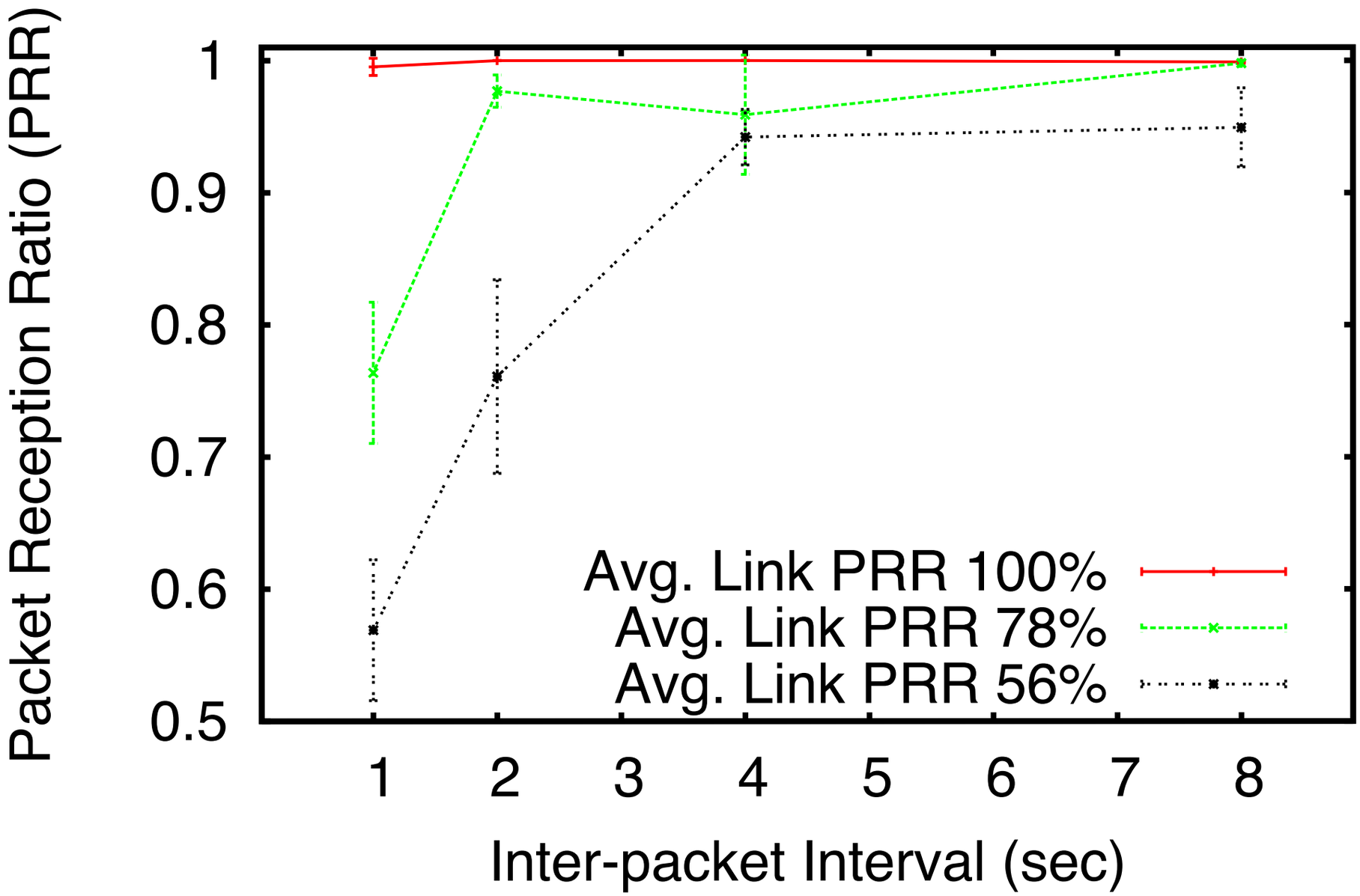
- Using the two implementations, we test the *performance* of interoperability
- Contiki Simulation Environment
  - MSPsim node level emulator + Cooja Network Simulator
  - Bit-level accurate simulations for Tmote Sky
  - Benefits: Accurate simulations for multiple binaries in a single network setting -- essential for interoperability tests

Simulation Environment

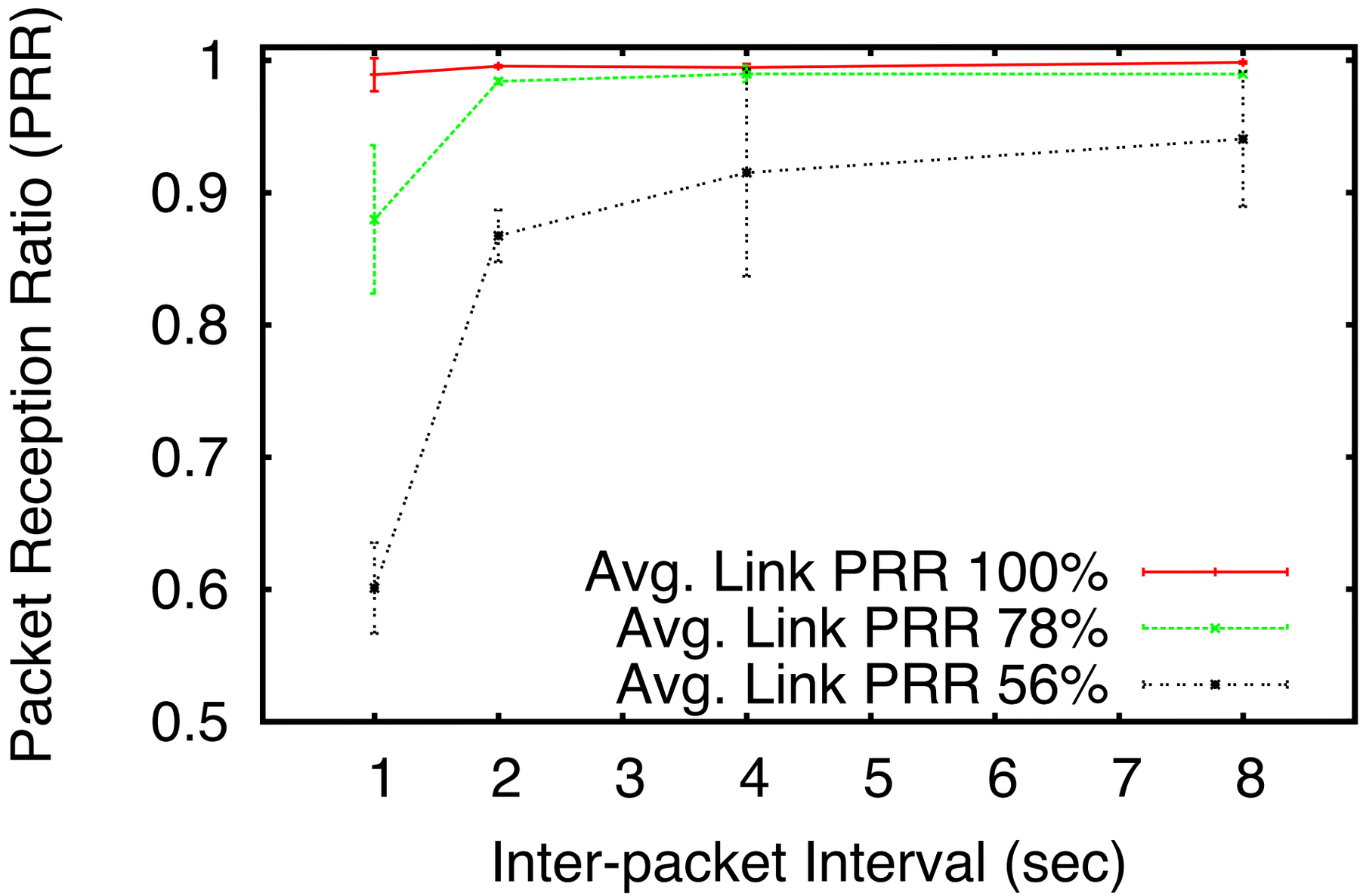
- 40 node topology
- IPI = 8 seconds (unless specified)
- Unit disk graph model with Bernoulli loss model
  - 0% loss (100% link PRR), 50% loss (avg 78% link PRR), 100% loss (avg. 56% link PRR) at edge of disk
- This channel environment differs from real wireless channels but fits our need to create network dynamism to compare system performance



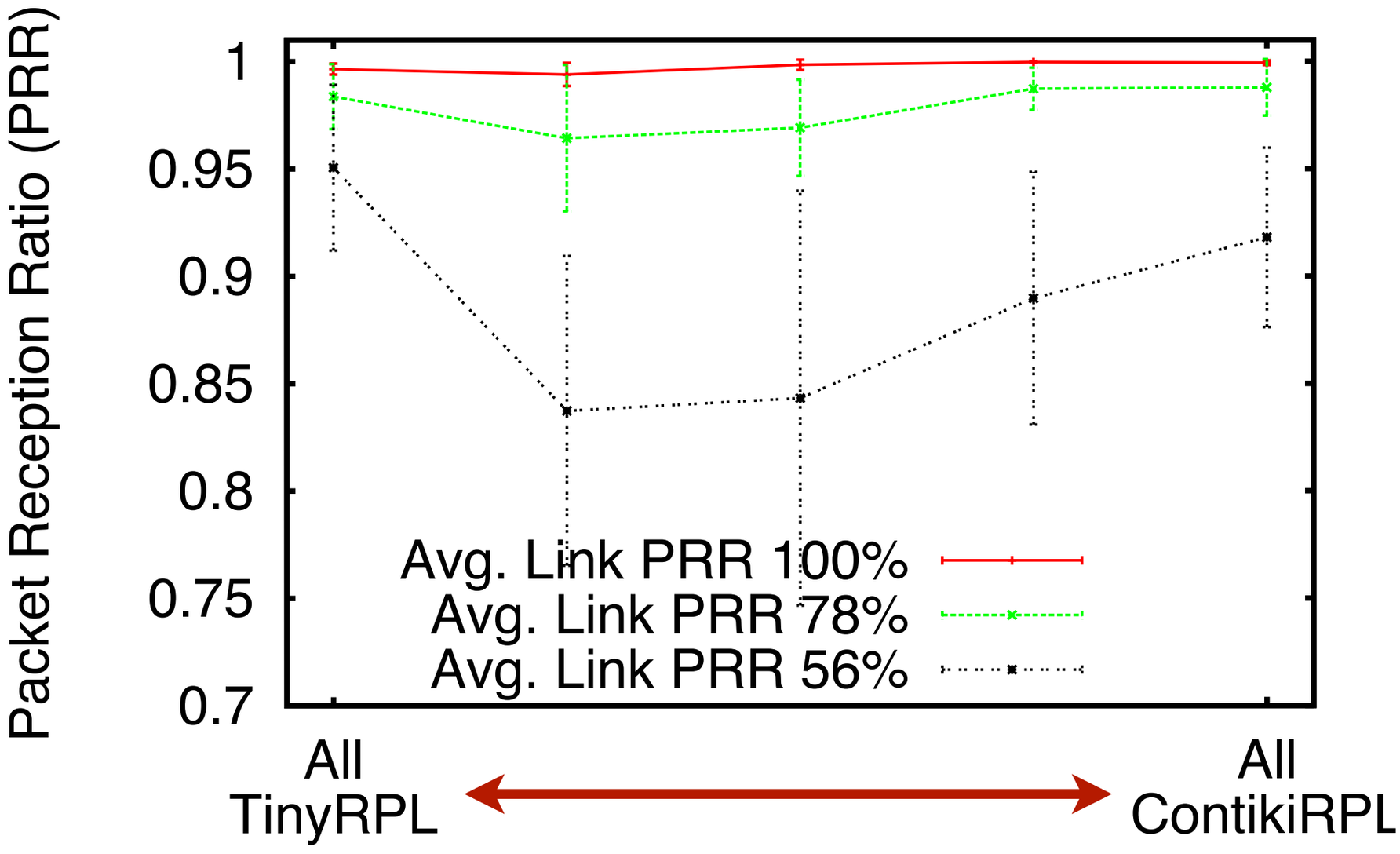
Contiki Evaluation



TinyOS Evaluation



When TinyOS and Contiki First Met



TinyOS and Contiki with their original settings

Why?

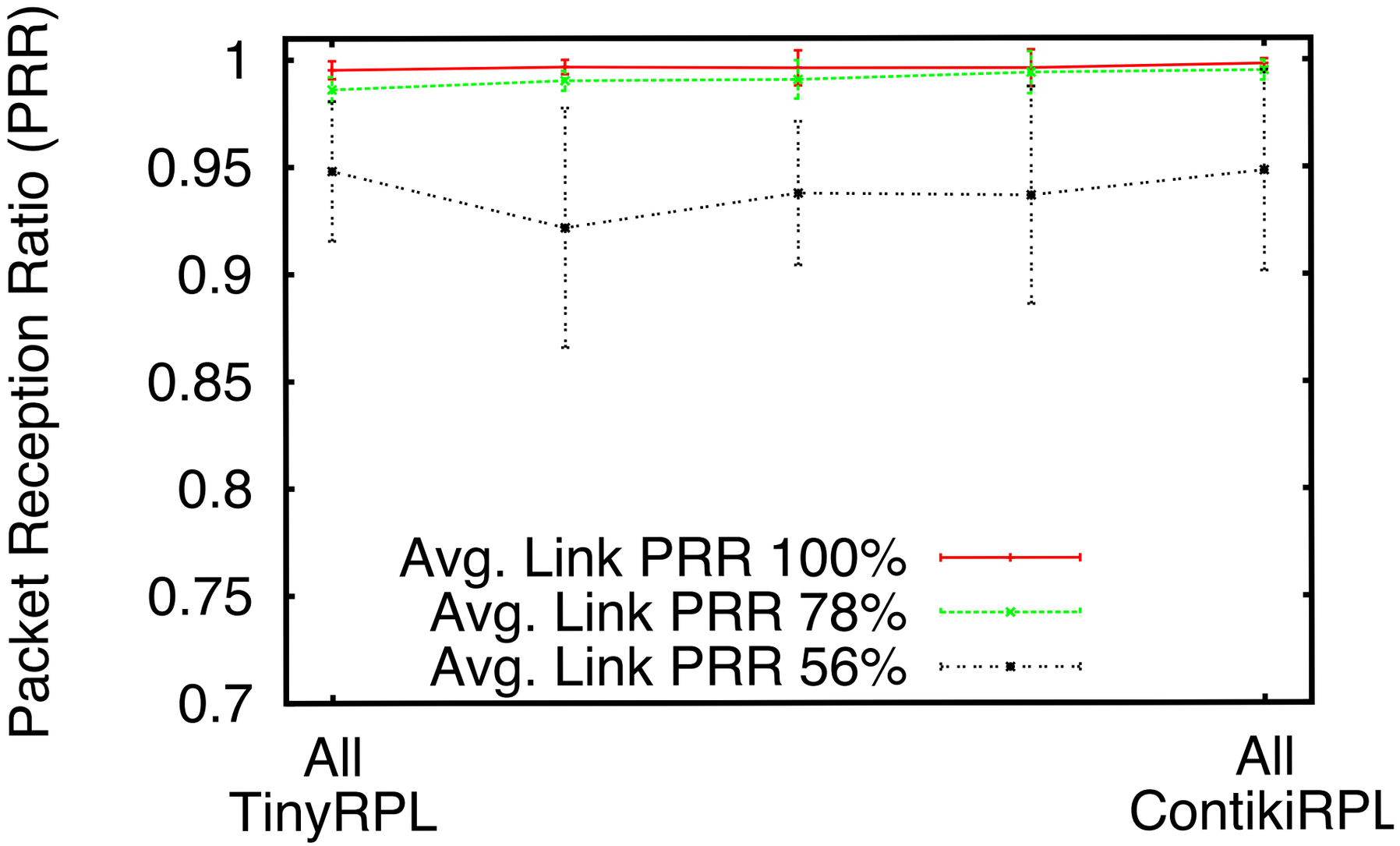
- Same RPL-related parameters -- distributed using root-initiated DIOs
  - RPL's operations does not vary over different implementations
- PRR decrease is not the effect of RPL itself!

Effect of Lower Layers

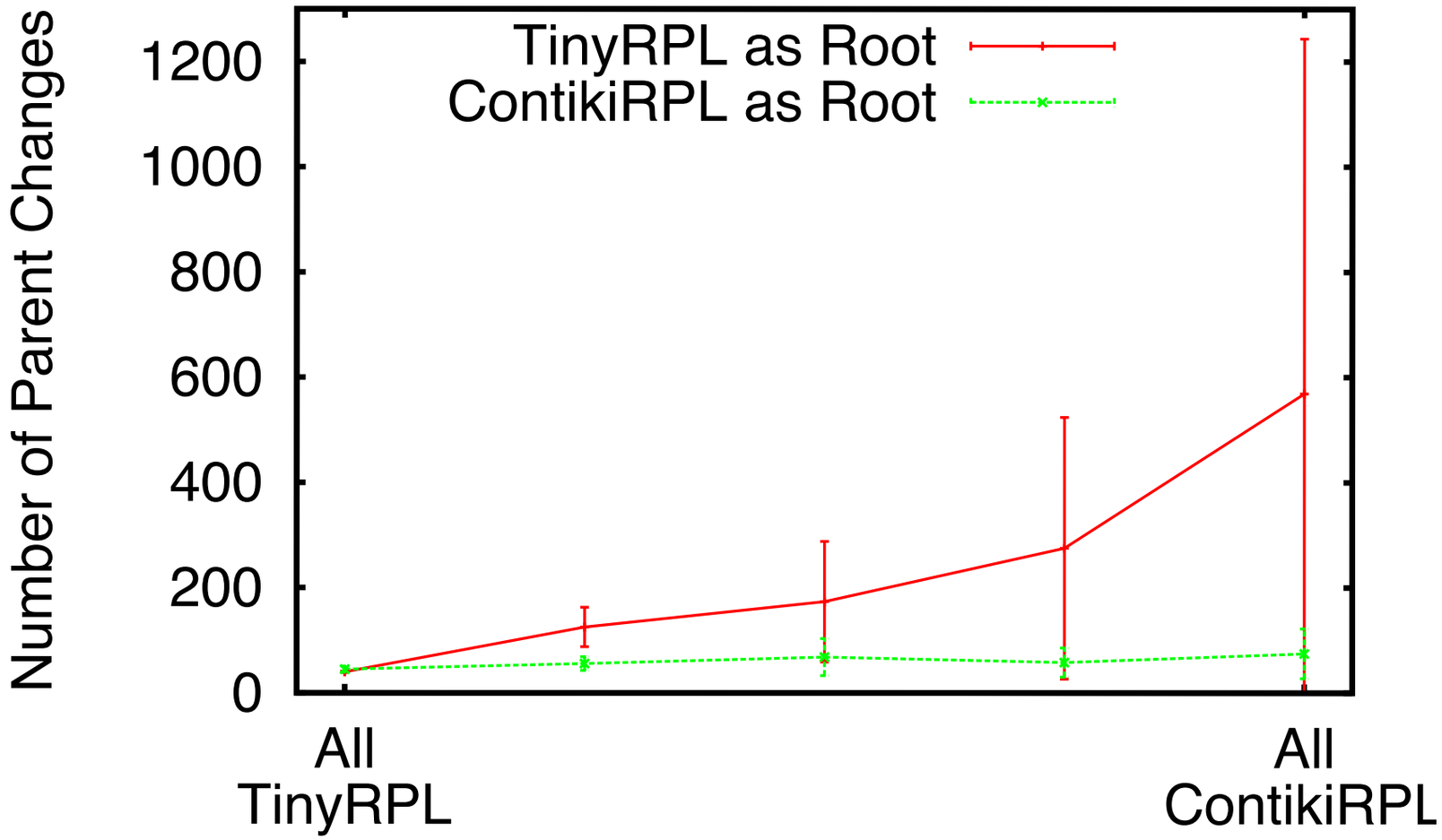
- **Network layer:**
  - Message buffer sizes vary
- **MAC-layer**
  - Retransmission Timers/Limits vary

- **Need to align parameter values across all layers of the software stack**

TinyOS/Contiki Interoperability



Need for deeper investigation!



The unexpected can happen!

Even if PRR performance is high, we should pay careful attention to other metrics as well.

Lessons Learned (1)

- Maintain the lowest common denominator
  - Implementation specific optimizations can *interfere* with the interoperability process
- Leverage simulations
  - Testbed experiments can show more realistic results but have *temporal and topological limitations*
  - Visibility and control of the environments helps to understand the 'functionality'
  - E.g., Average of 300 runs per PRR graph

Lessons Learned (2)

- Examine the performance at all layers
  - All layers of the protocol stack will affect the overall performance of interoperating systems

Future Work

- Point-to-Multipoint, Point-to-Point Traffic
- Other objective functions
- Effect of radio duty cycling

Conclusion

- IPv6/RPL interoperability and the performance of the heterogeneous network between TinyOS and Contiki look promising
- Interoperability testing is not enough and we should consider the performance of the overall system which is affected by multiple-layers in the protocol stack

Questions

- JeongGil Ko
  - jgko@cs.jhu.edu