



Radio Stack Iteration

How to improve the CC1000

Joe Polastre

January 15, 2004

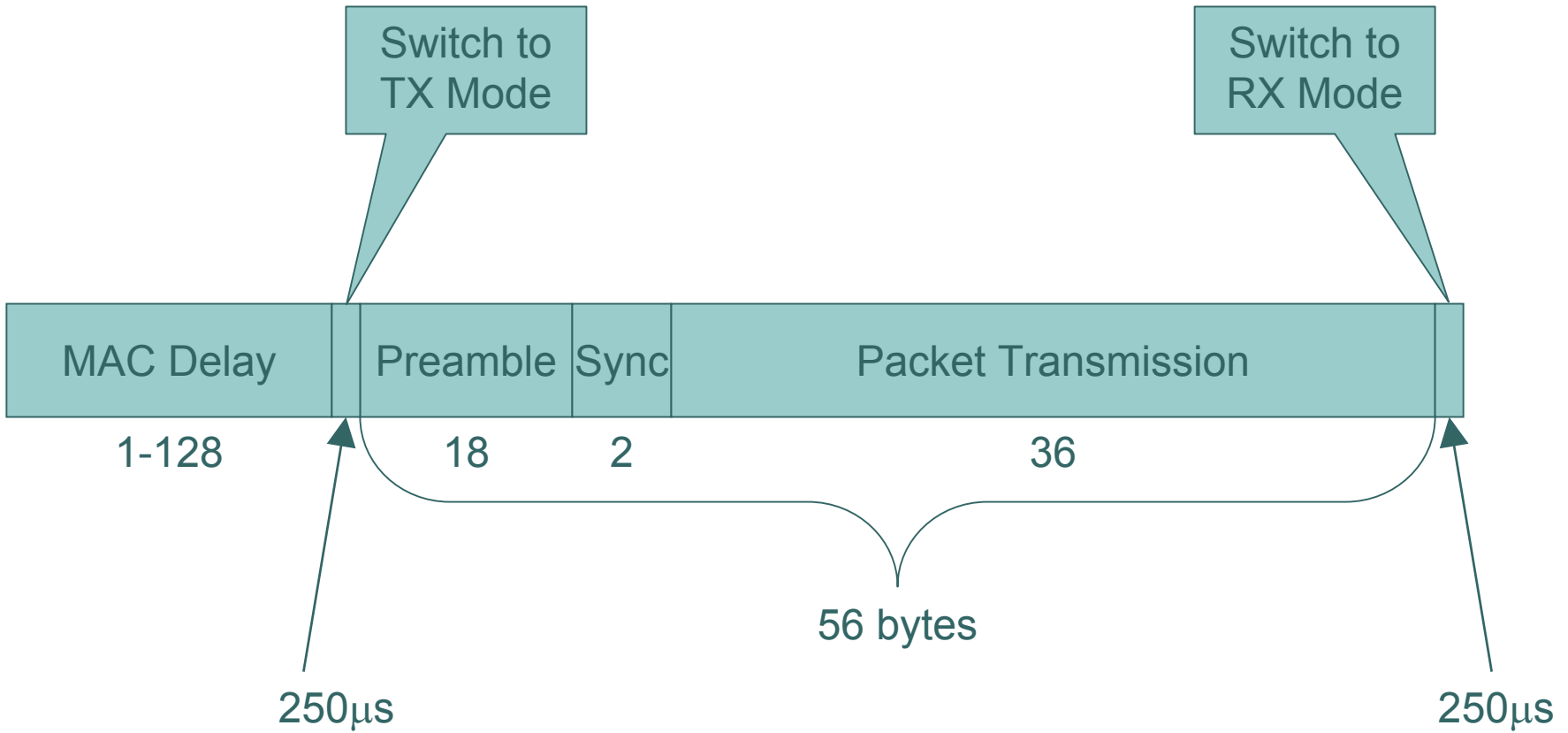
NEST Retreat



Outline

- Mica2 TinyOS 1.1 MAC
- New Methods for Clear Channel Assessment
- Improvements to the MAC
- How to develop TinyOS core services across many groups
(without breaking everyone else's applications)

Mica2



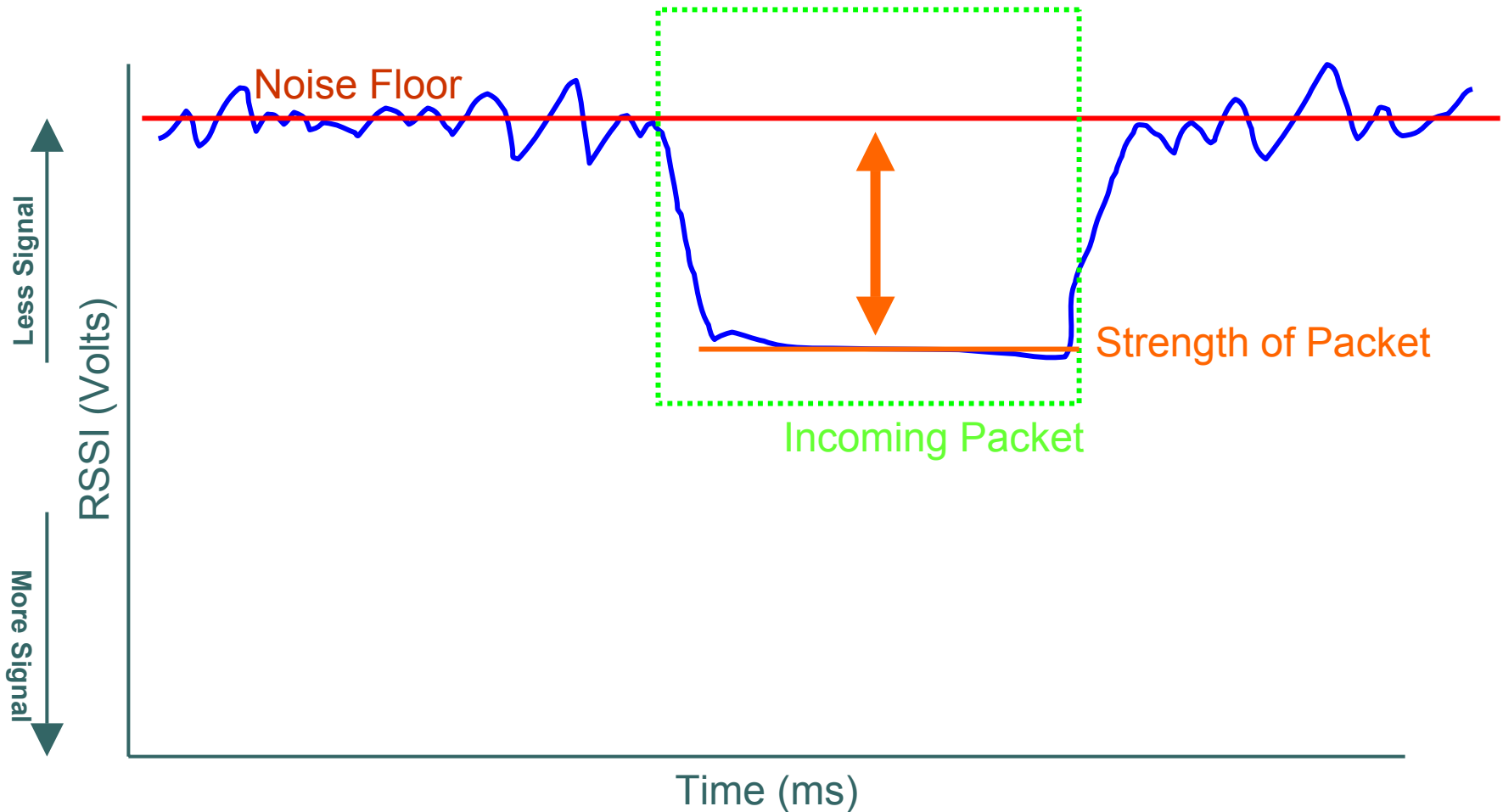


Mica2 Release CSMA

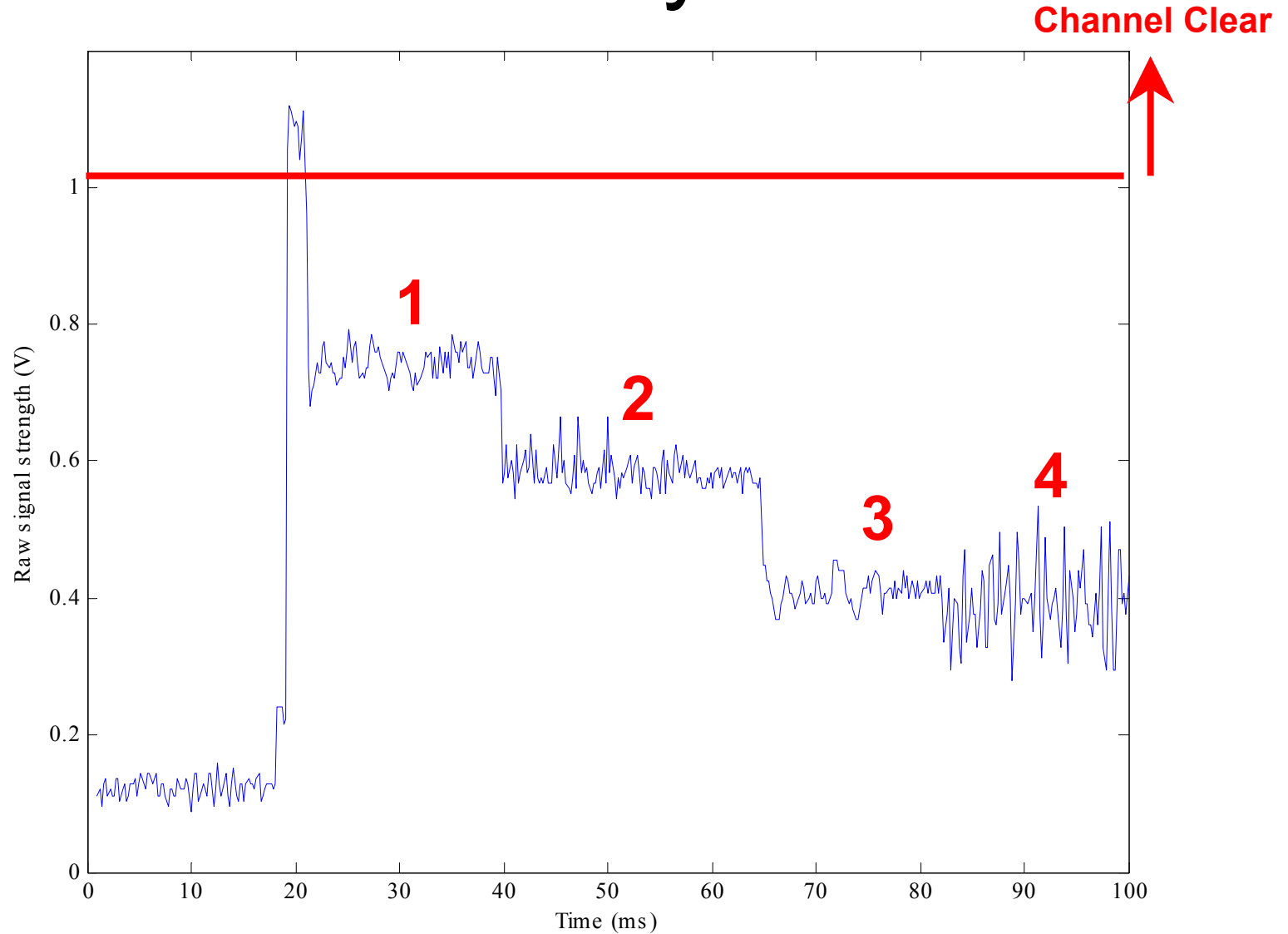
- Set a fixed noise floor threshold
- On transmission:
 - Take RSSI reading
 - If reading $<$ floor, transmit on next byte boundary from radio
- With a few months in the lab, what happens?

RSSI on the Chipcon

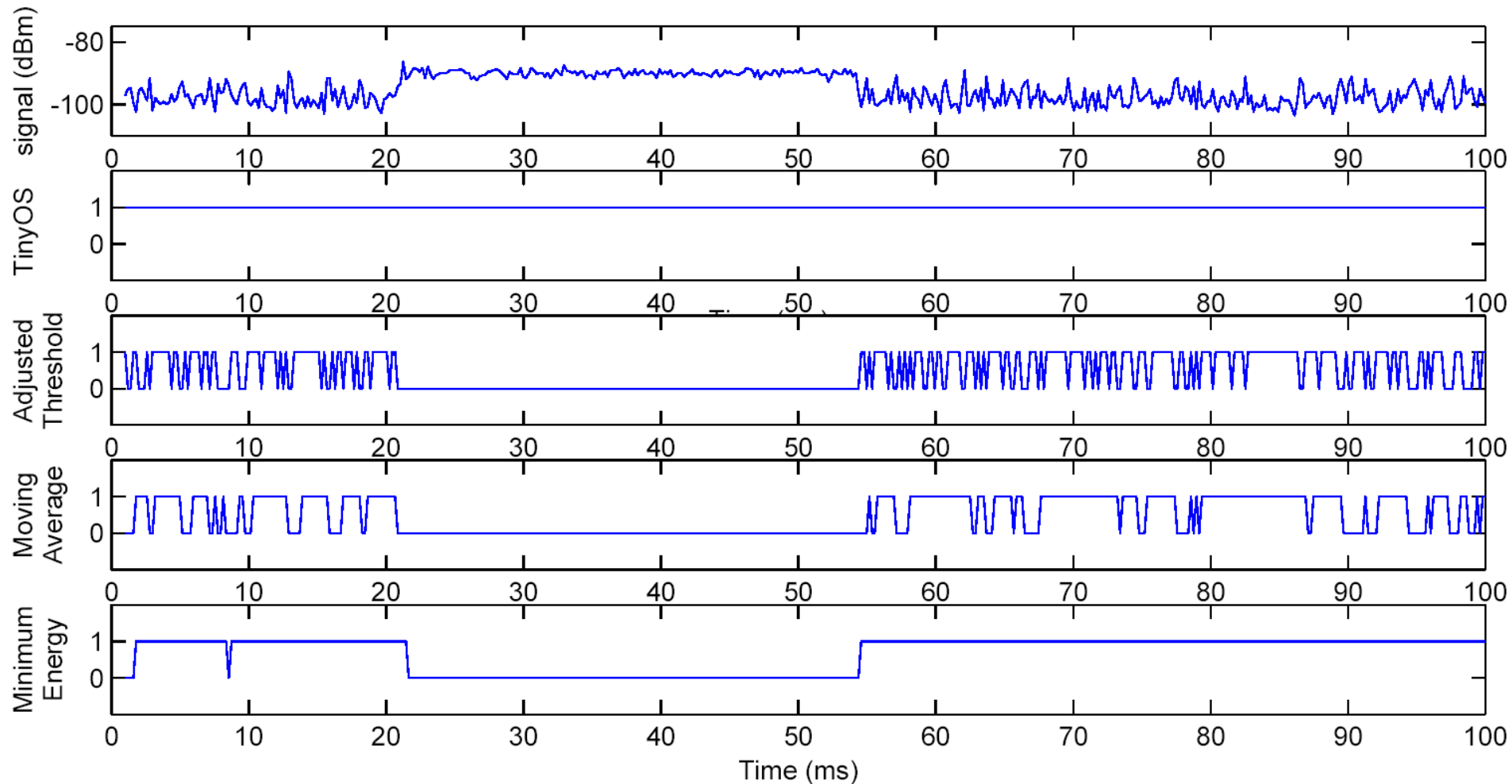
signal strength inversely proportional to RSSI



Collisions Oh My!



Clear Channel Estimation





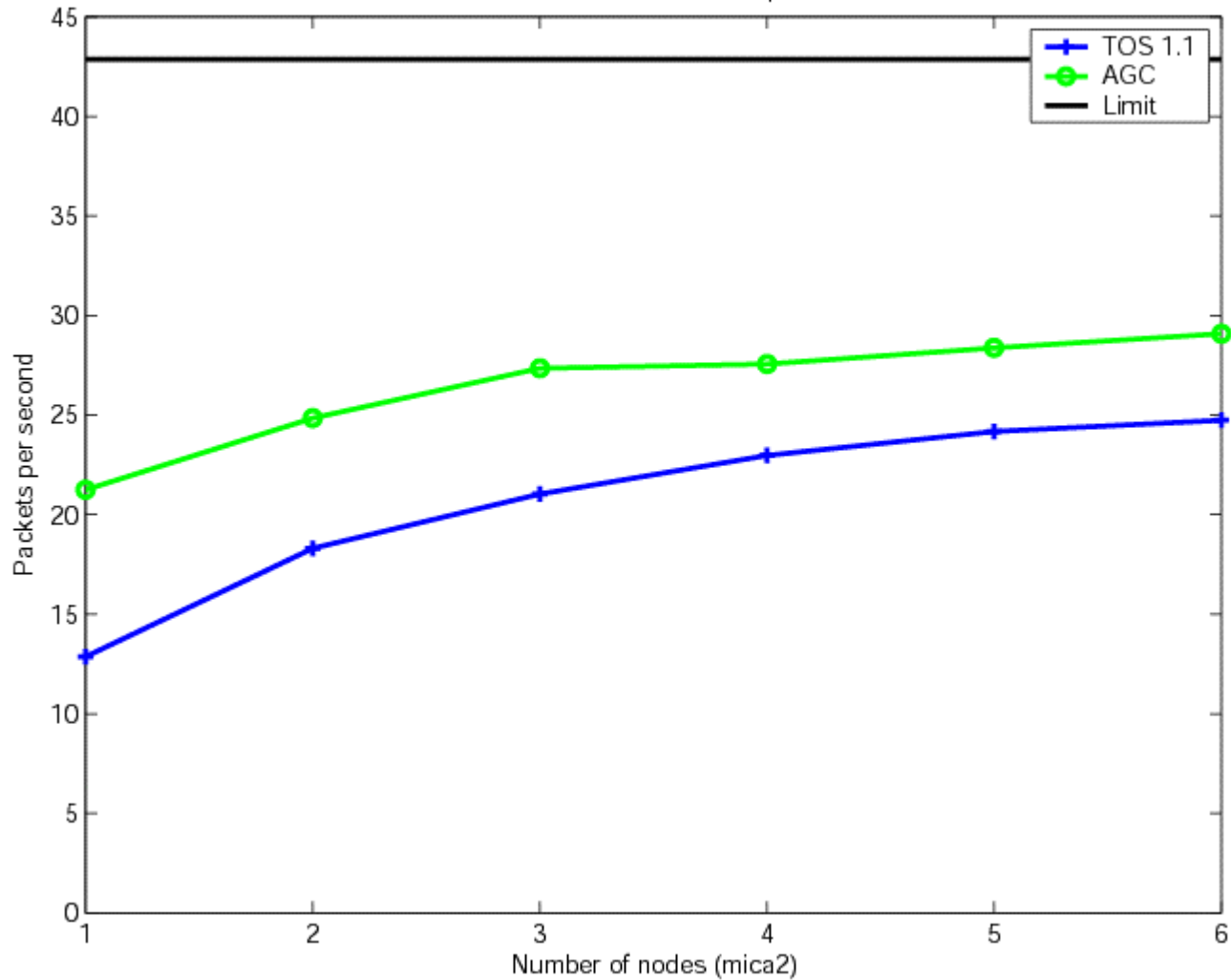
Adjusted Mica2 CSMA Layer

- Adjust noise floor (AGC):
 - After each packet transmission, sample RSSI and add to 10 sample FIFO queue
 - Select median(Q) and add to EWMA
- On transmit:
 - Take ≤ 5 RSSI samples
 - If one reading $<$ floor, transmit on next byte boundary from radio
- On initialization: timer takes radio samples to fill the FIFO queue

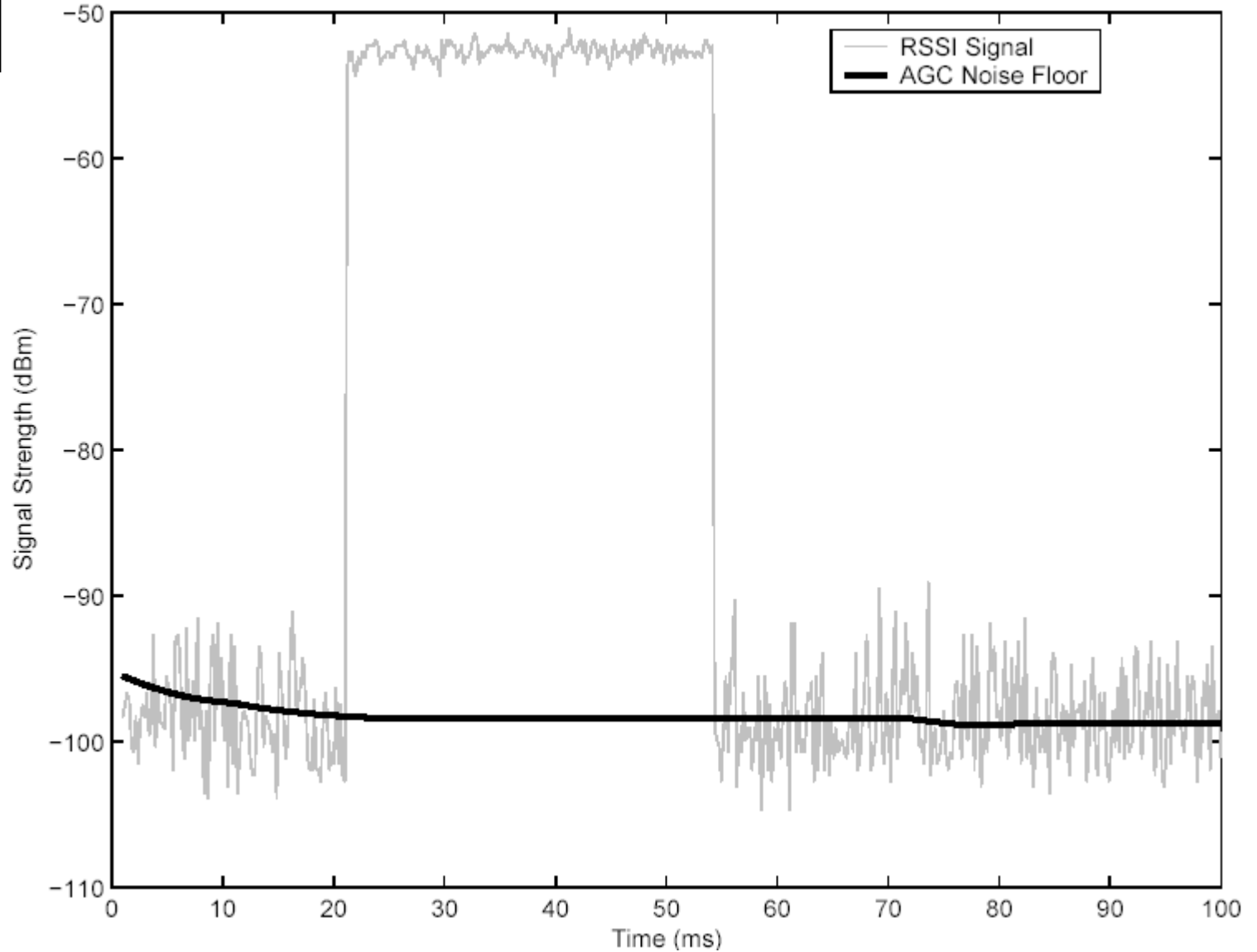
Results

Each node sends as fast as possible

Delivered bandwidth with Different Payloads
Each node sends as fast as possible



AGC Performance





New Mica2 CSMA Layer

*Let's call it **B-MAC** ...*

- How does it differ from AGC:
 - No initial backoff
 - Minimize preamble overhead:
 - Woo et al show that communication with nodes in the noise floor is futile
 - set preamble length
 - set radio settling
 - Now: radio hardware only detects nodes with strength greater than the noise floor

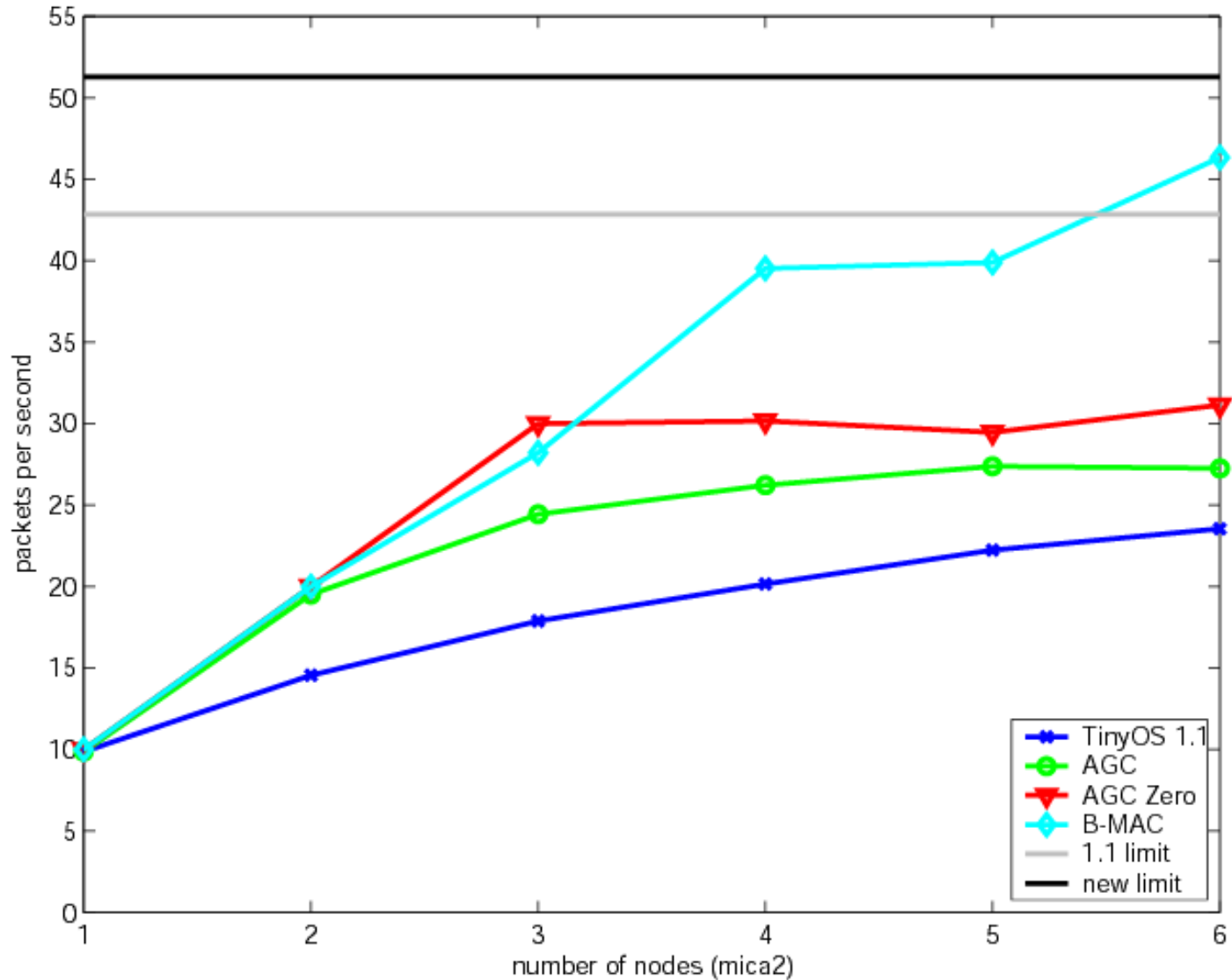


Comparison

Mote <i>Granularity</i>	Mica <i>bit</i>	Mica2 Rel <i>byte</i>	AGC <i>byte</i>	B-MAC <i>byte</i>
Max Init MAC Backoff	16 (5.3) 3.2ms	36+128 68.3ms	64 26.6ms	0 0ms
Max Congest Backoff	16 (5.3) 3.2ms	29*16 193ms	16 6.6ms	16 6.6ms
Noise Floor	AGC (Hardware)	Fixed	AGC (Software)	AGC (Software)

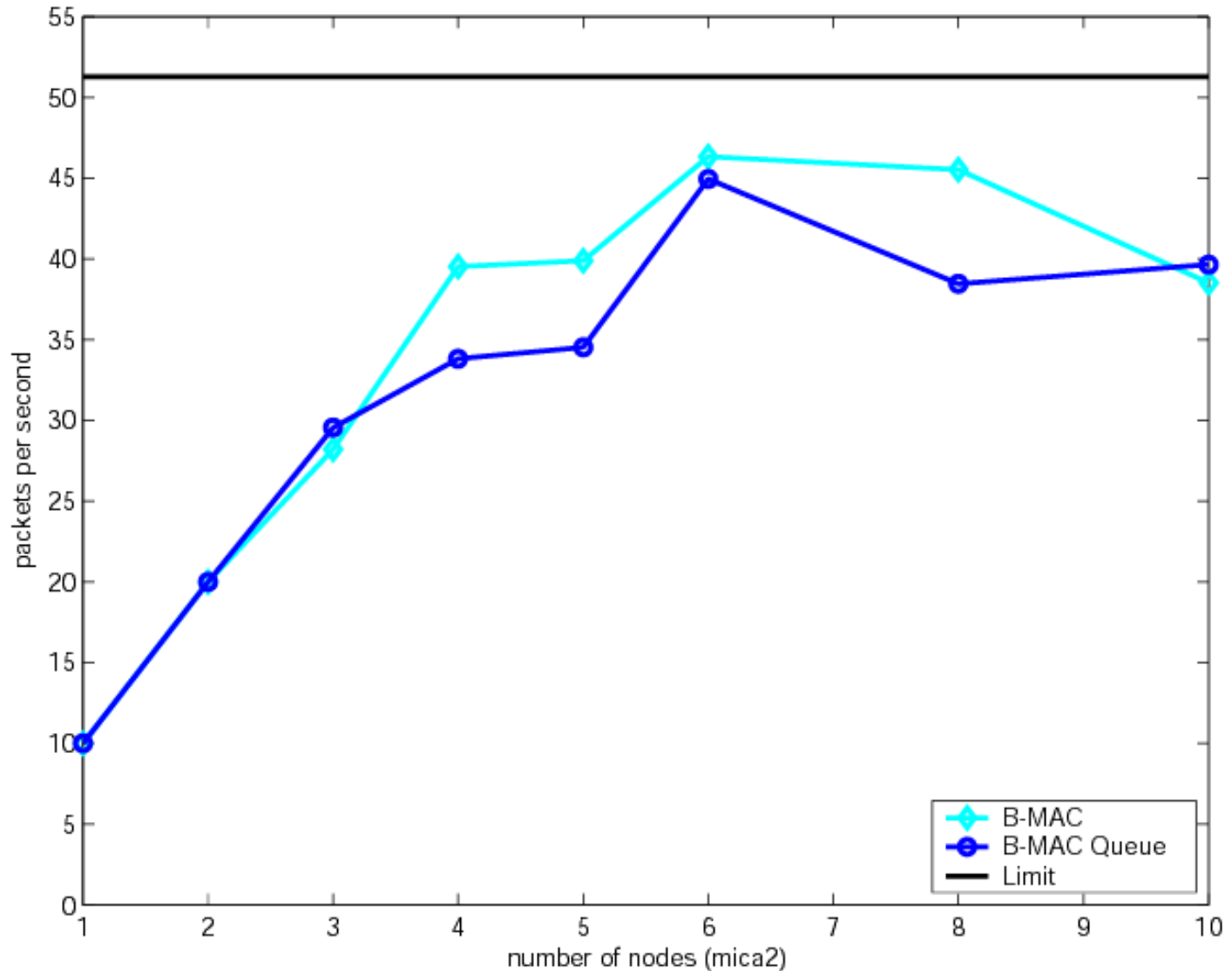
Results

Each node sends 10 packets/sec



Results

Each node sends 10 packets/sec





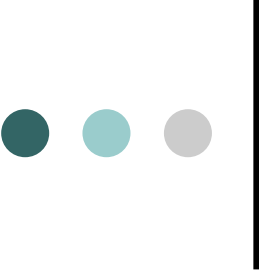
MAC Interfaces

- Application controlled parameters!
 - For example: initial CSMA backoff:
 - No backoff for bulk data transfer
 - Random backoff for broadcasts
 - 2 packet backoff for multihop routing (tree)
- Expose MAC parameters enabling cross layer control



Other Improvements

- Link Layer ACKs
 - Small ACK packet immediately following data packet
 - Requested by many services including multihop routing
- Low Power Listening
 - Periodically wake up, sample the channel for activity, and return to sleep
 - Reduced the time to sample the channel



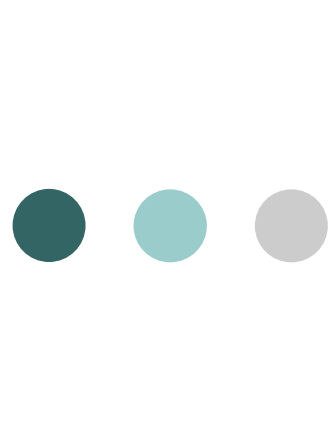
What is the process for getting this into TinyOS?

- “beta” structure
- Defined project with a lifetime
- Expected that the beta project will become part of the main branch
- Enables multiple groups to work on the same evolving code without breaking the main branch
 - eg: radio stack was work between UCLA, USC, UCB, IRB, and JHill
- Moved to the main branch one month before its release



Conclusion

- Systematic design of MAC layer
 - Provides application cross layer control
 - Different services require different MAC parameters
 - Evolving functionality and improvements
 - AGC, Low Power Listening, ACKs, Interfaces
- Beta system now in place for collaborative work on new TinyOS services



webs.cs.berkeley.edu

TinyOS 1.1.3 Release