

Anti-Jamming for Embedded Wireless Networks

Miroslav Pajic and Rahul Mangharam
Department of Electrical and Systems Engineering
University of Pennsylvania



Jamming

- Jamming – radiation of electromagnetic energy
 - loss of link reliability,
 - increased energy consumption,
 - extended packet delays,
 - disruption of end-to-end routes
- Malicious and non-malicious
 - The hard part: distinguishing and concealing
- Anti-jamming
 - Resilience to jamming and its avoidance
 - Must be native to the communication protocol

Outline

- **Jammers and trade-offs with jamming**
- **Anti-jamming:**
 - Coordinated Spatio-temporal Randomization**
 - WisperNet-Time
 - WisperNet-Space
- **Performance analysis**
- **Implementation Experiences**

What is so special about Wireless Sensor Networks?

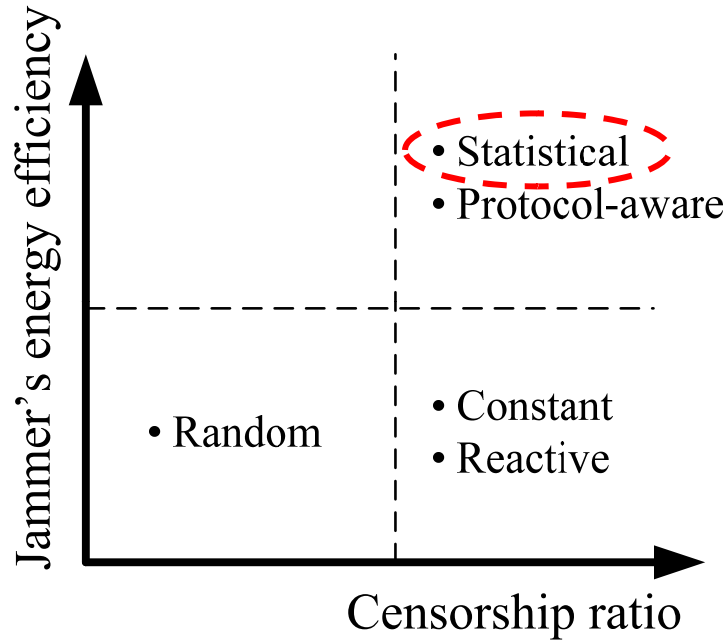
- Nodes scheduled to maximize the common sleep duration and coordinate communication
 - **Temporal patterns** in communication
 - **Predictable intervals** of transmission activity

- Efficient to scan and jam only during activity
 - Jamming Competitive Ratio
-
- 32 Scheduled Slots in every frame

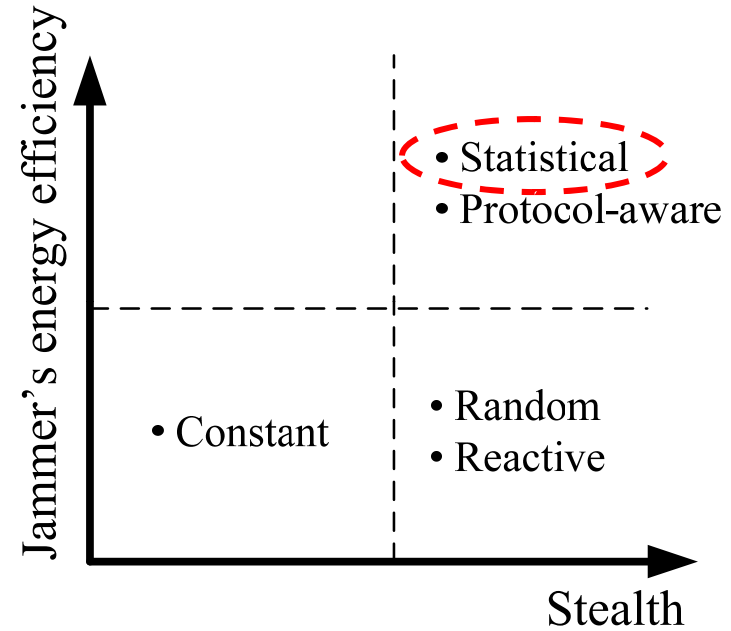
Properties of anti-jamming protocol

1. Non-predictable schedules
2. Non-predictable slot sizes
3. Coordinated and scheduled transmission
4. Coordinated changes of slot sizes
5. Collision-free transmission

Comparison of Jamming Models



Jammer's Energy Efficiency vs. Censorship Ratio

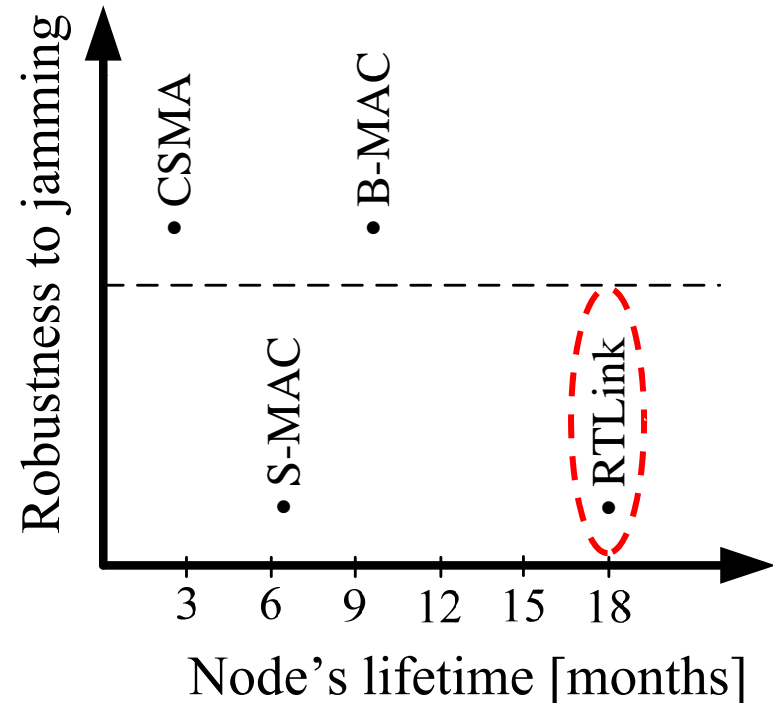


Energy Efficiency vs. Stealth

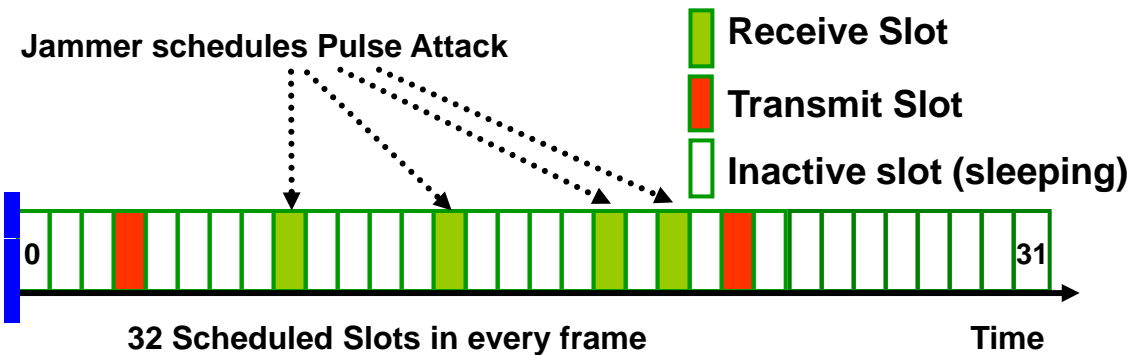
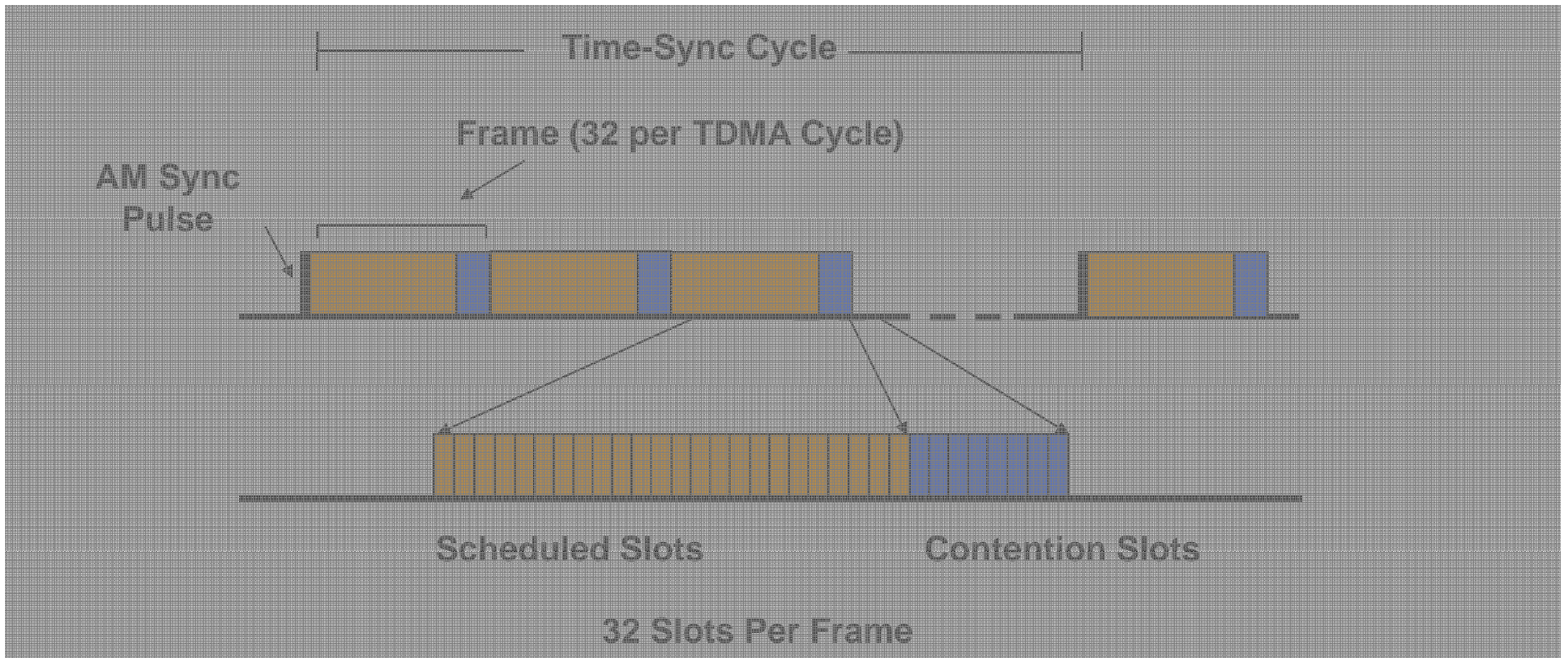
Energy-Efficient MAC Protocols

Energy-Efficient MAC protocols

- Asynchronous protocols (B-MAC)
- Loosely-synchronous (S-MAC)
- Synchronous protocols (RT-Link)

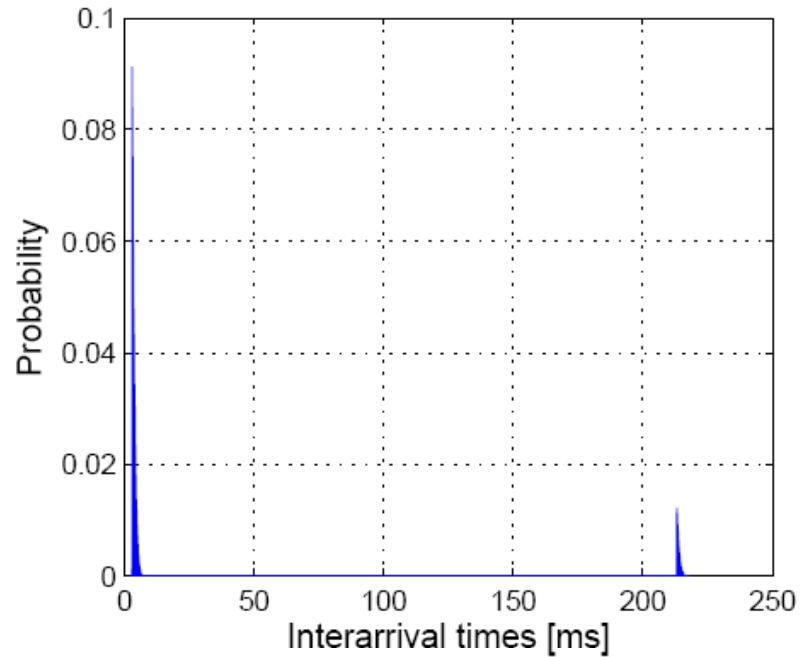


RT-Link: Real-Time Link Protocol

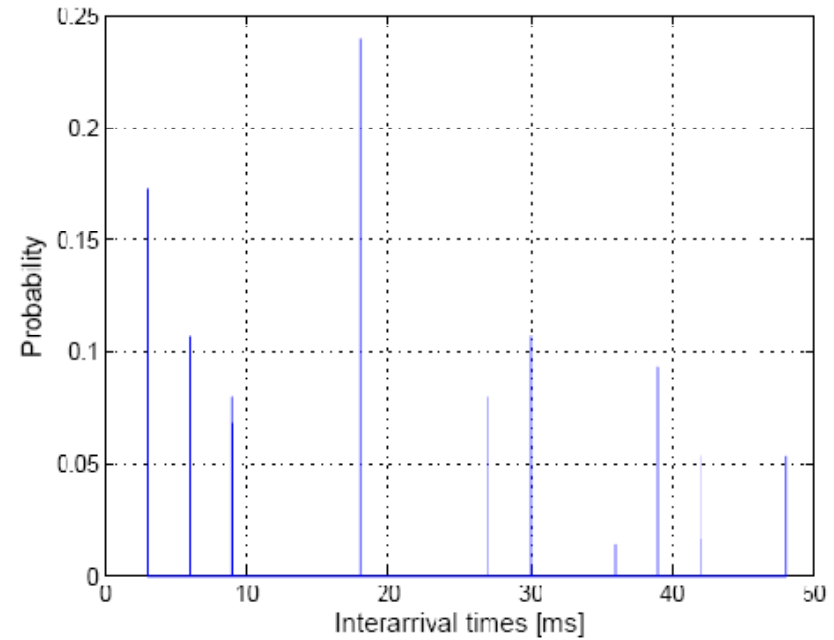


Statistical jamming

Histogram: S-MAC



Histogram: RT-Link



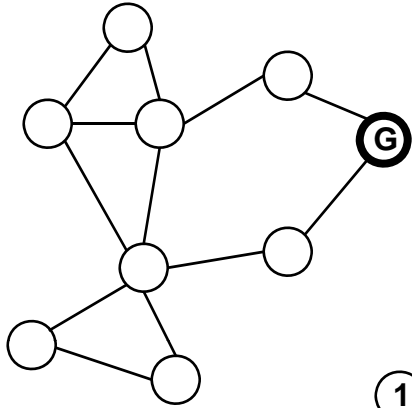
10 nodes in network, average transmission 3ms

Assumptions

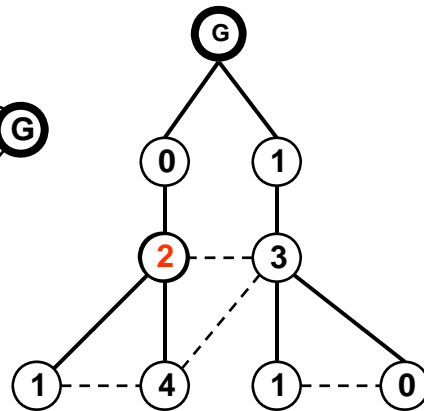
- Jammer is as energy constrained as a legitimate node
- Jammer is not protocol aware
- Both malicious and non-malicious are considered
- Transmission power 0dBm
- Interference range is equal to transmission range (1 hop)

Schedule Randomization

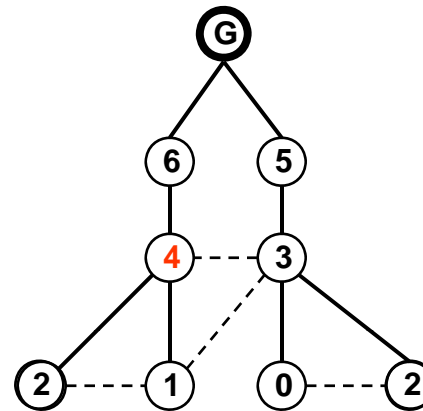
Physical Topology



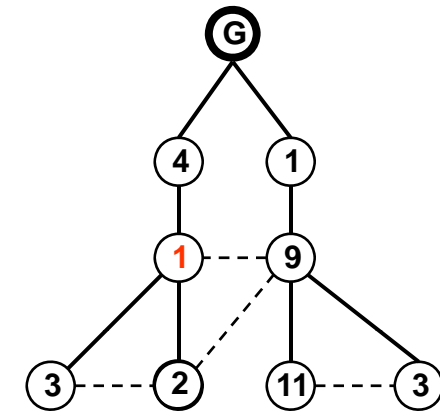
Logical Pruning:



Scheduled Frame i



Frame $i + 1$

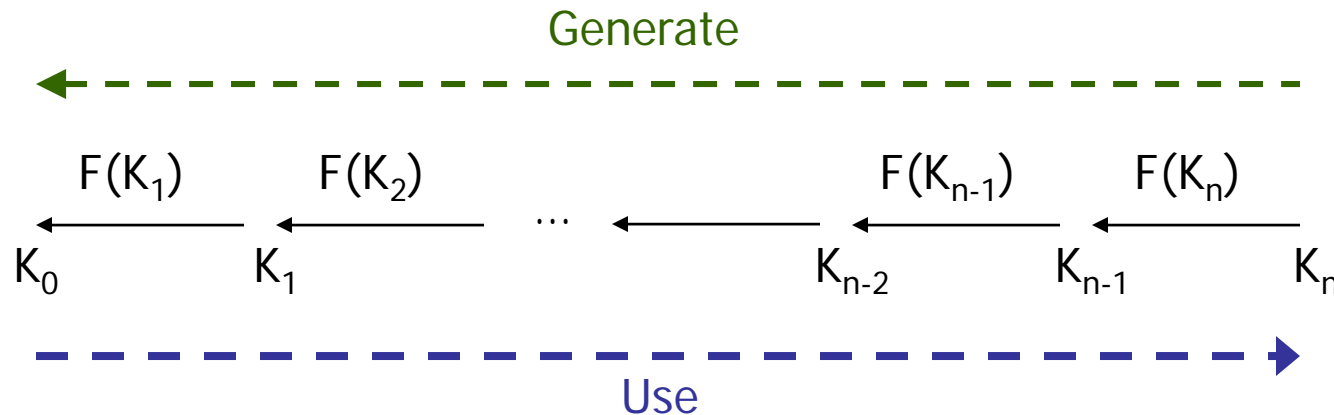
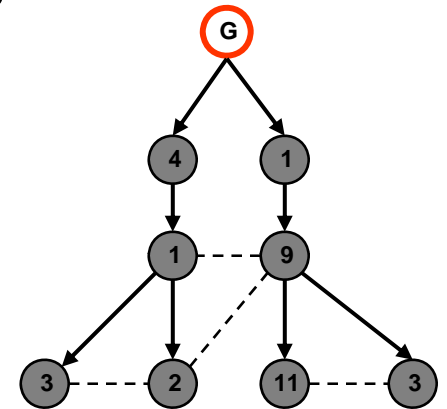


Frame $i + 2...$

- Gateway distributes new keys periodically to each node (~5secs)
- Slot schedules based on $\text{SHA1_HMAC}(\text{key}, \text{nodeID})$
- Non-repeating schedules
- Tight time synchronization between nodes
- Schedules must not cause packet collisions

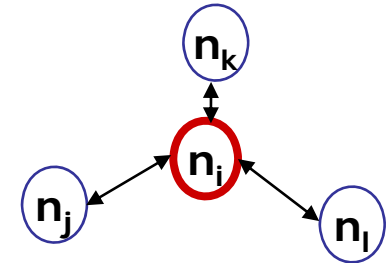
HMAC Schedule Randomization

- Gateway generates One-way key chain using SHA1
 - One key per cycle (~3s)

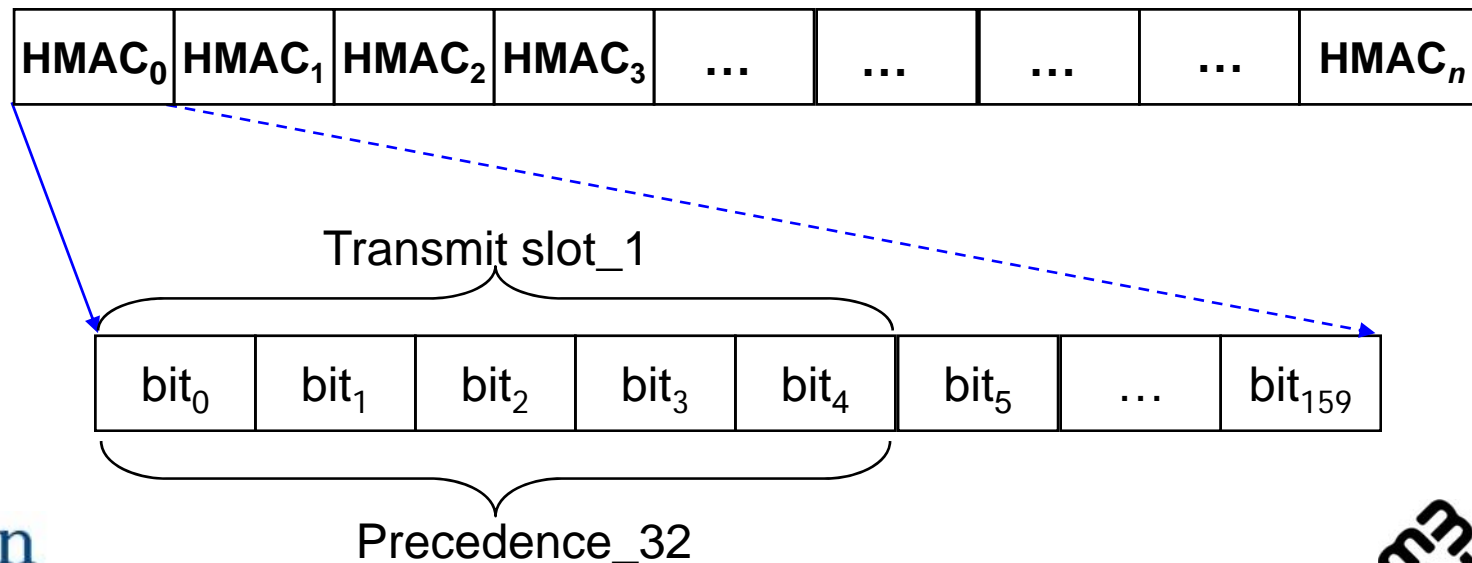


Schedule based on SHA1_HMAC

- Each node computes (using SHA1_HMAC):
 - its **own transmit slots** and their **precedence**
 - its neighbors' transmit slots and their precedence
 - collision-free operation

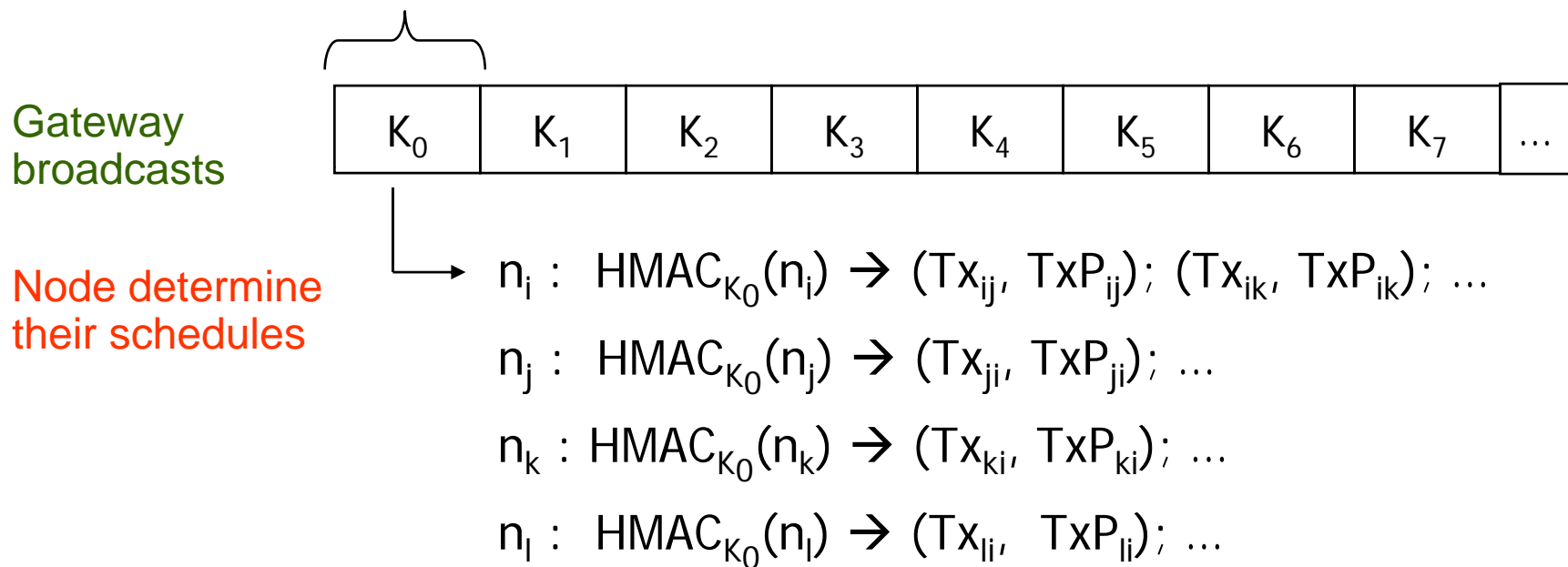


SHA1_HMAC output:



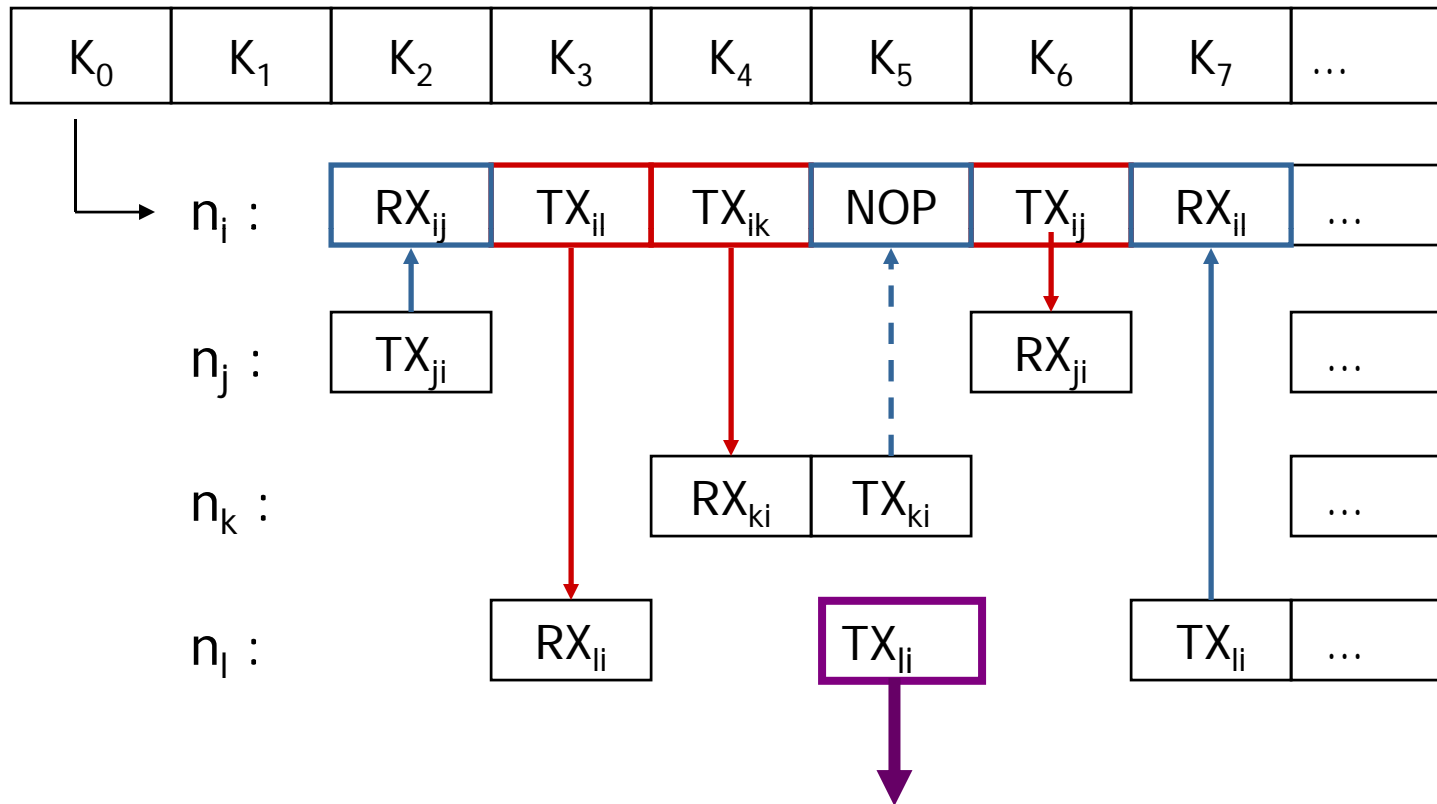
Implicit Schedule Conflict Resolution (1)

Valid for ~3 secs



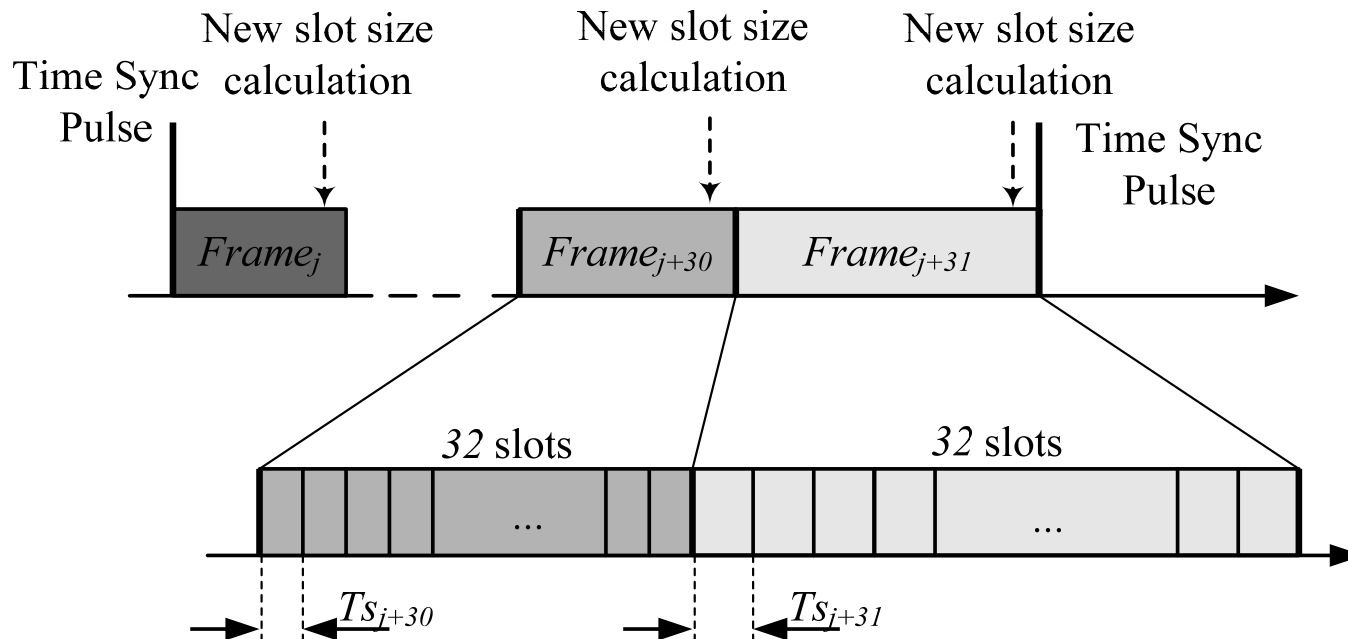
Implemented SHA1 and HMAC_SHA1 in 8-bit fixed-point

Implicit Schedule Conflict Resolution (2)

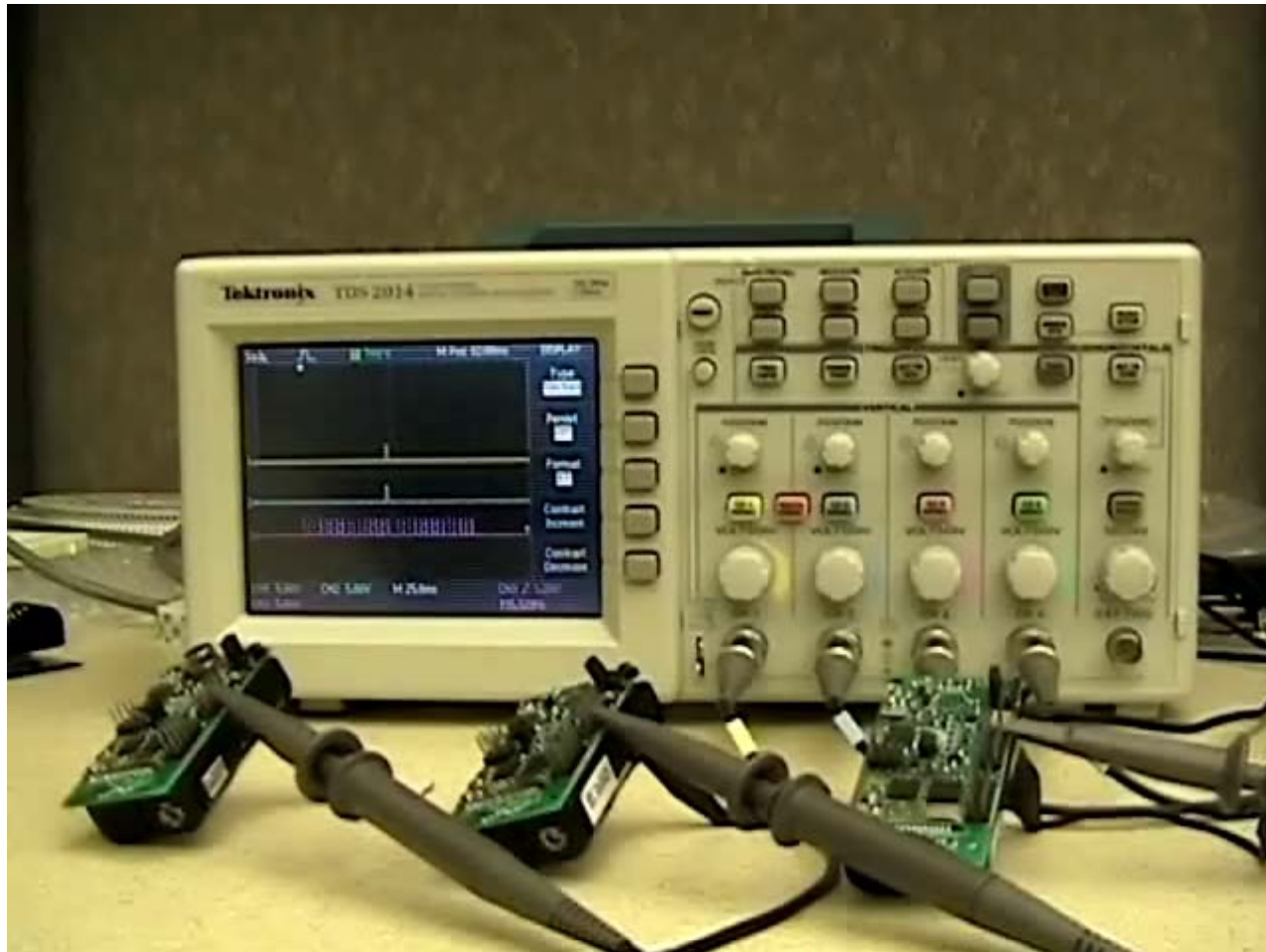


Slot size randomization

- Slot size as function of output (K_{slot} is a local key)
 $HMAC(cnt, K_{slot})$

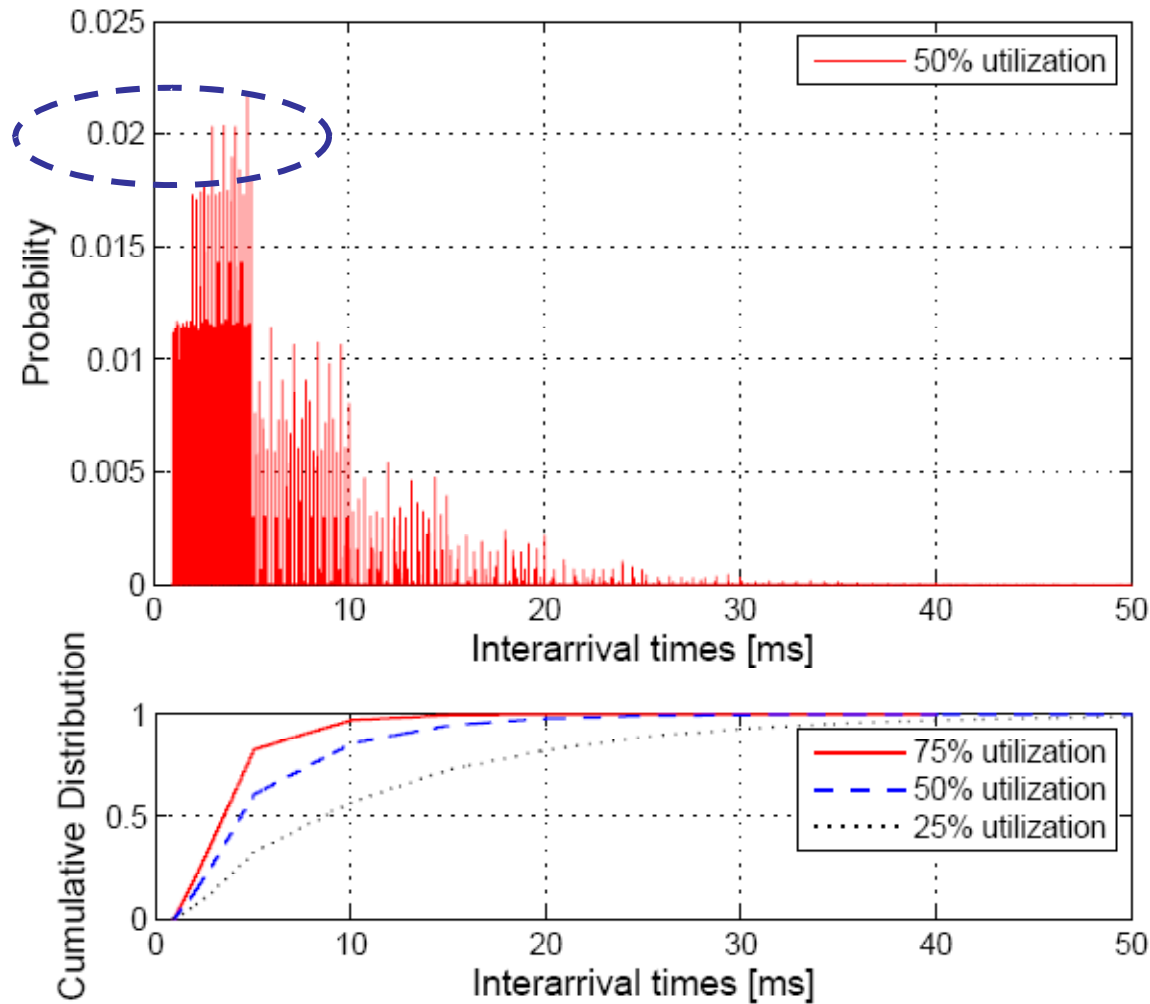


DEMO



WisperNet-Time: Performance Analysis

Effect of Channel Utilization

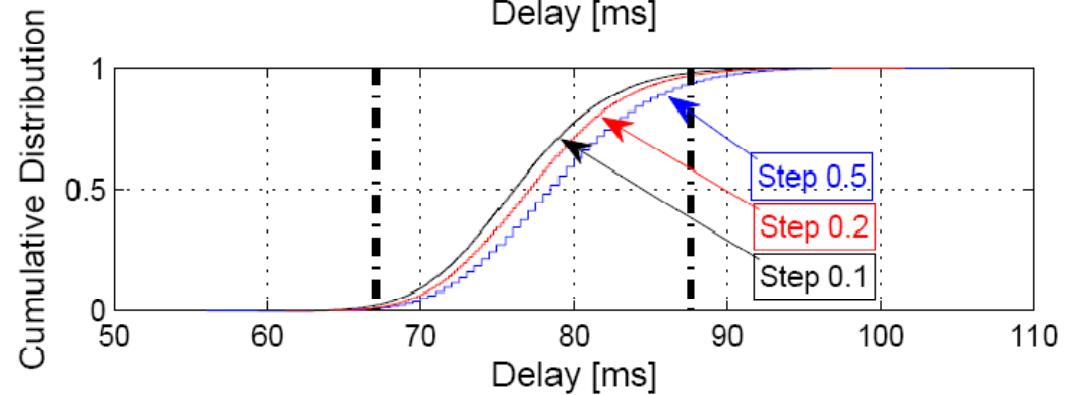
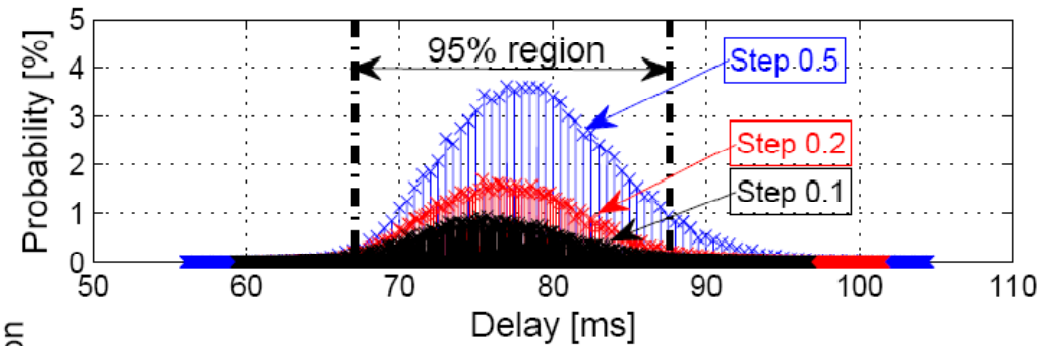


PDF

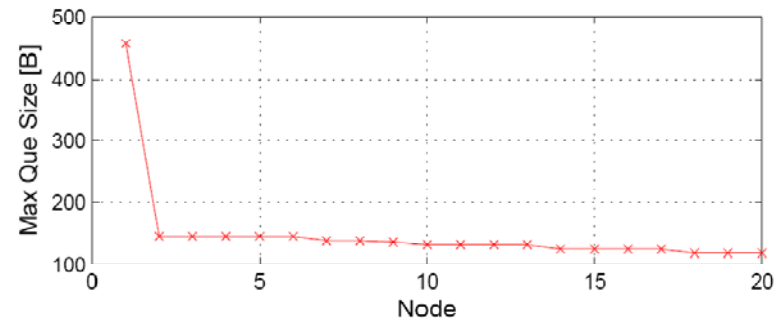
CDF

Effect of Slot size Randomization

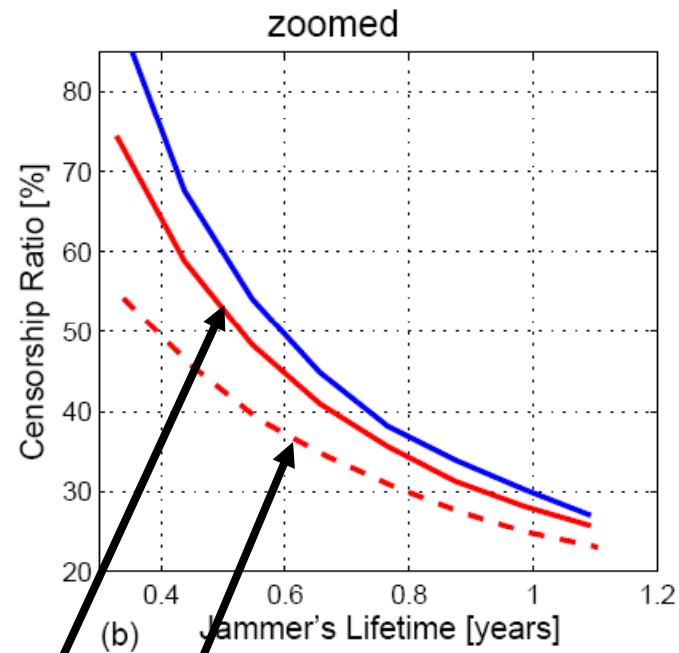
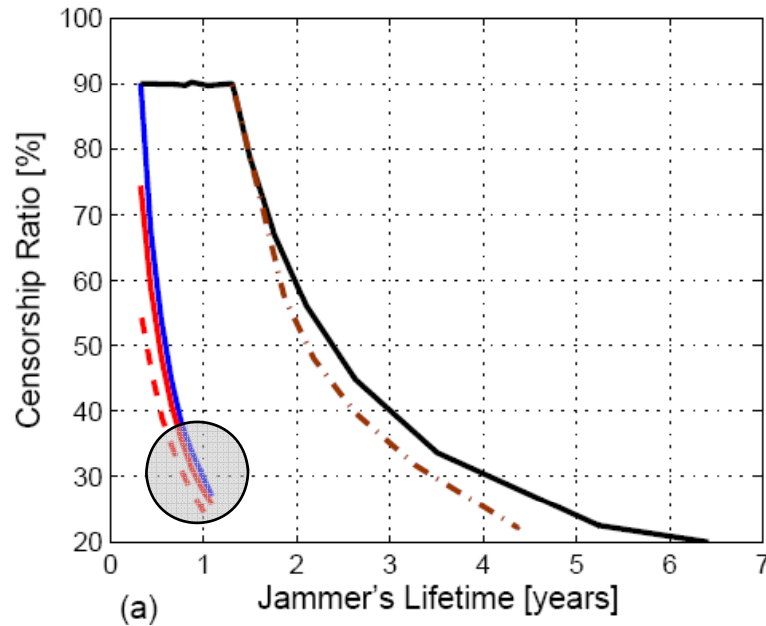
End-to-end delay



Memory requirements



Jammer's Lifetime and Censorship Ratio



- RSTDMA with Statistical Jammer
- WnT with Statistical Jammer
- WnT with Random Jammer
- RT Link with Statistical Jammer
- S-MAC with Statistical Jammer

WisperNet-Time

Random Jammer

WisperNet-Space: Coordinated Spatial Adaptation

- Each link $e = e(u, v)$, associated with k weights, $w_j(u, v)$, $j = 1, 2, \dots, k$
- For a tree T , the aggregate weight is defined as

$$W_j(T) = \sum_{e \in T} w_j(e), \quad j = 1, 2, \dots, k$$

WisperNet-Space – Network's reliability

- Reliability of a path P : $\prod_{e \in P} PDR(e)$

- Maximizing reliability equivalent to maximizing

$$\ln\left(\prod_{e \in P} PDR(e)\right) = \sum_{e \in P} \ln(PDR(e)).$$

WisperNet-Space – Active topology update

- RW for some link $e = e(u, v)$ is defined as

$$w_r(u, v) = |\ln(PDR(u, v))|$$

- To defend against mobile jammers, use of leaky integrator

$$w_r(u, v) = \begin{cases} |\ln(PDR(u, v))|, & (u, v) \in T \\ \rho \cdot w_r(u, v), & (u, v) \notin T \end{cases}$$

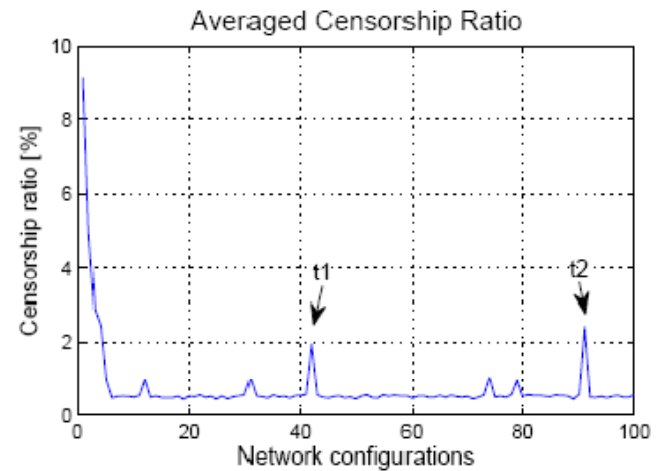
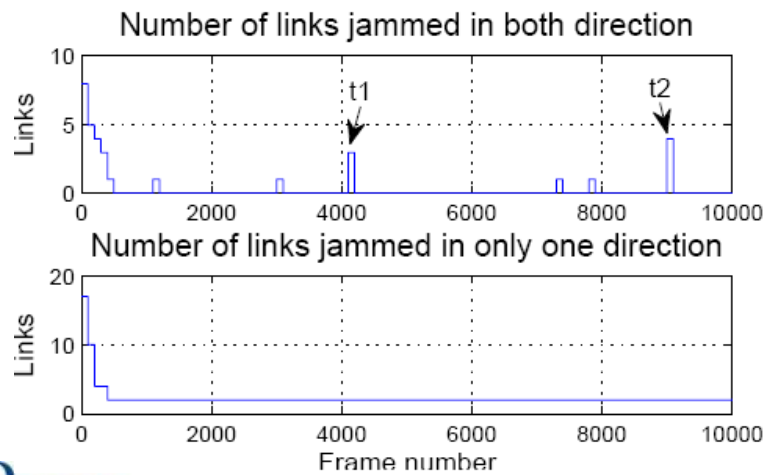
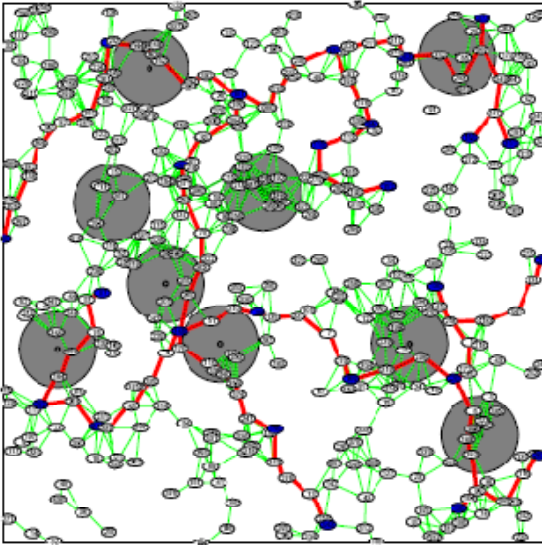
WisperNet-Space – Topology maintenance and updates

- A new network topology every 128 cycles – 6.4 minutes on average
- Problem: How to inform inactive nodes?
 - The gateway broadcasts a topology update
 - Topology configuration frame (8 slots after each synch pulse) is reserved for asynchronous communication
 - Topology maintenance accounts for a 0.78% overhead.

WisperNet-Space: Performance analysis

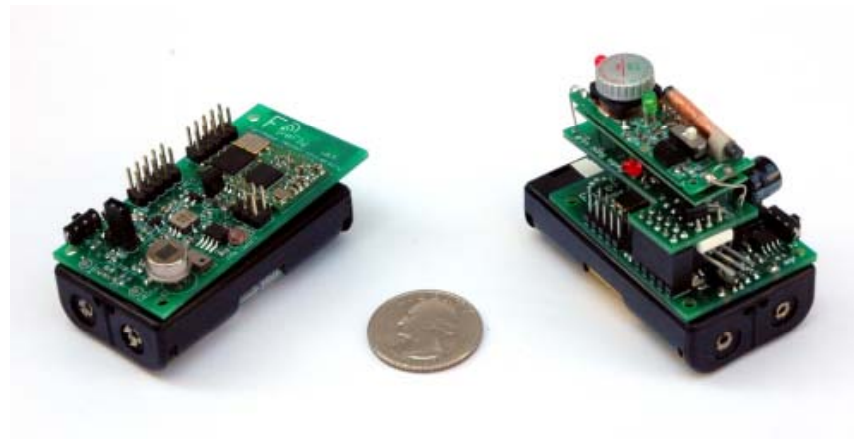
- 400 randomly distributed nodes in $4km \times 4km$ square
- 9 randomly distributed jamming nodes, with link 50% utilization
- Topology changes are performed once in 100 frames

WisperNet-Space: Performance analysis



Implementation: Platform

- Implemented on FireFly nodes




Implementation

- Implemented on
 - 8-bit microcontroller ATMEGA32L, running on 8MHz
 - 16-bit microcontroller MSP430F22x, running on 16MHz
- Schedule calculation – 276B of Flash, 400B of RAM
- SHA1-HMAC required only 3 additional 160-bit buffers
- SHA1-HMAC required 12.5ms on the TI MSP430F22x
- For networks with maximal degree $N=5$, less than 33% of CPU processing in the worst case (11% on average)

A thing to take away...

- Removal of any time-spatial patterns in communication for synchronized networks
- Achieved in fully coordinated and collision-free manner
- A light-weight anti-jamming scheme, native to used MAC protocol

Thank You

P R E  I S E

PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

