DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks

Anthony D. Wood, John A. Stankovic, Gang Zhou

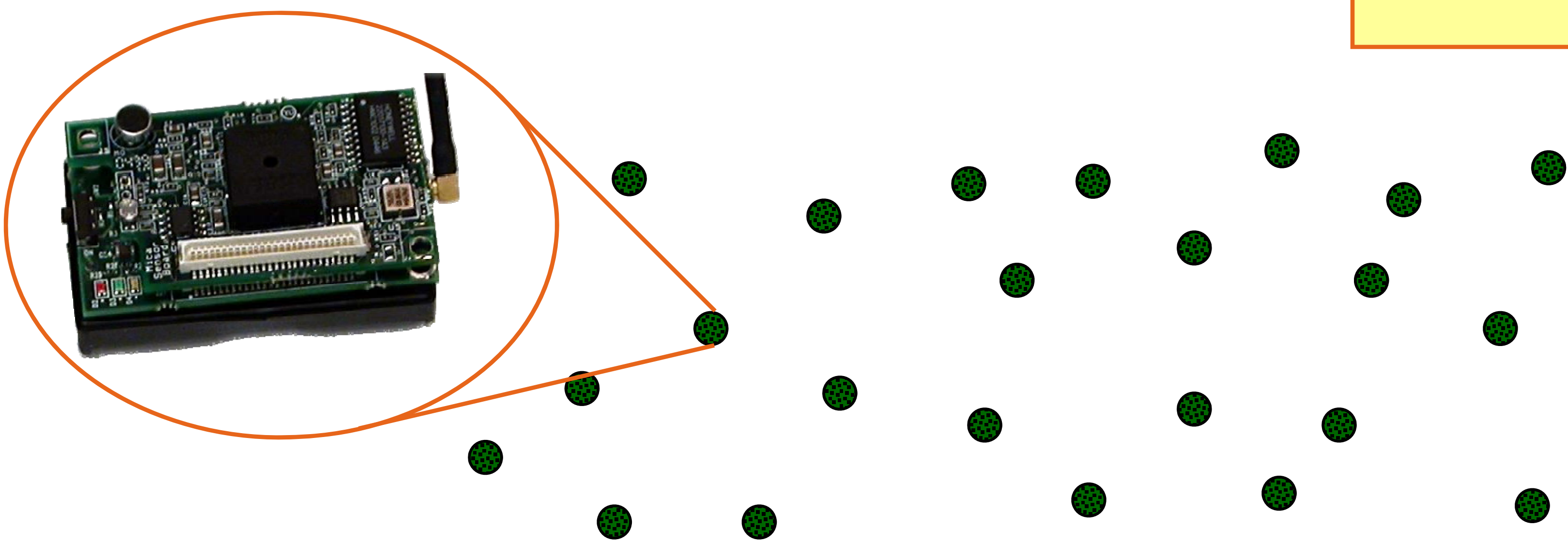
Department of Computer Science
University of Virginia

Wireless Sensor Networks

- Embedded in physical environment
- Devices with limited resources
- Large scale static deployment
- Diverse applications: military, volcano monitoring, zebra tracking, healthcare, emergency response ...

MICAz mote:

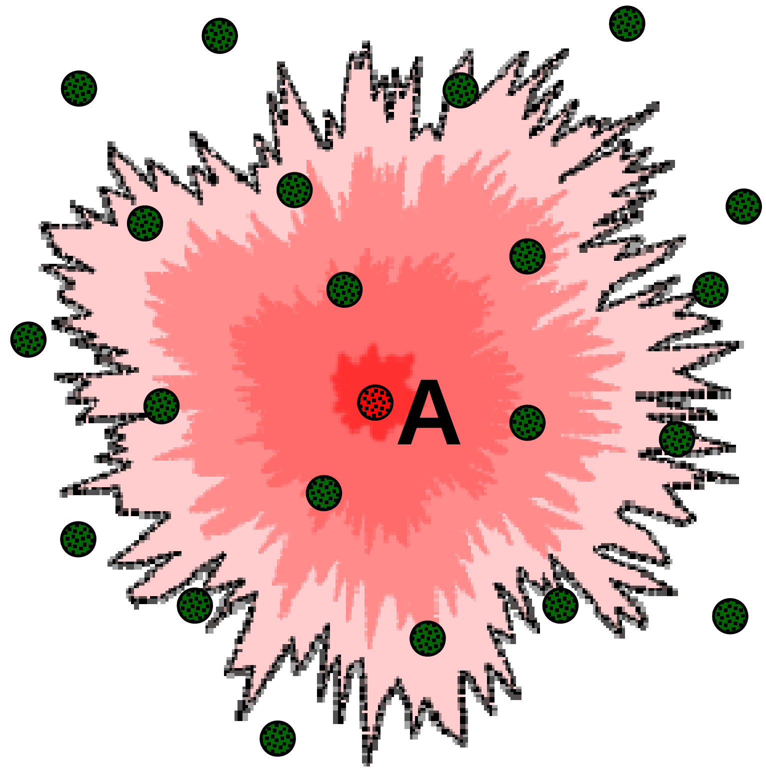
8 MHz 8-bit uP
128 MB code
4 KB data mem
250 Kbps radio



- IEEE 802.15.4 radios: MICAz, Telos/Tmote/Tmini, iMote2, XYZ

Physical-Layer DoS

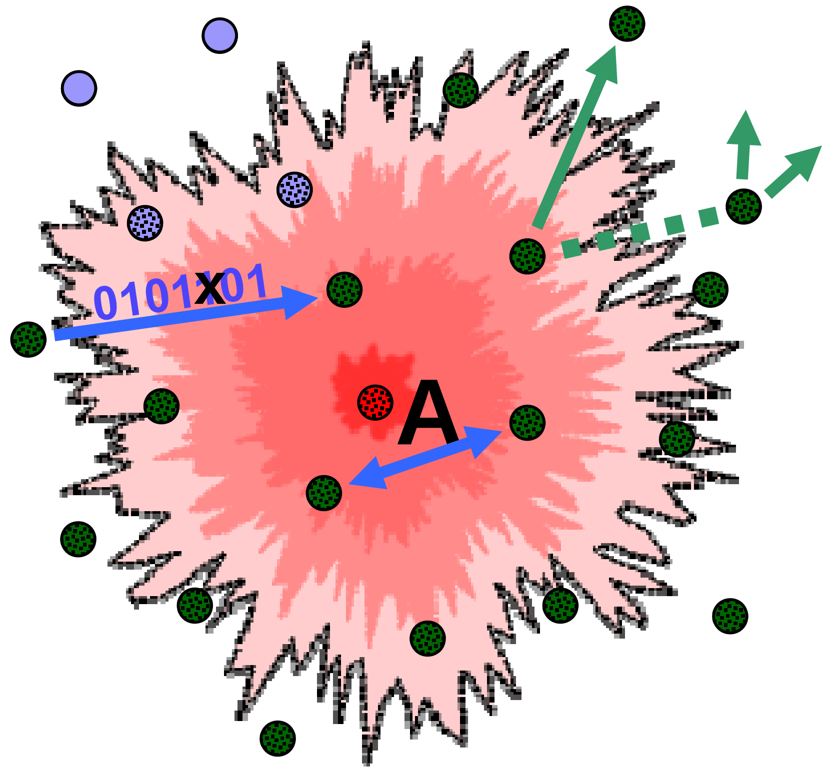
- Threats and Vulnerabilities:
  - WSNs becoming ubiquitous, connected to IP networks
  - Devices are easy to compromise
  - Jamming is easy to do in software
  - DoS attacks will spread to WSNs



► Attacker's goal: disrupt communication as steathily and energy-efficiently as possible

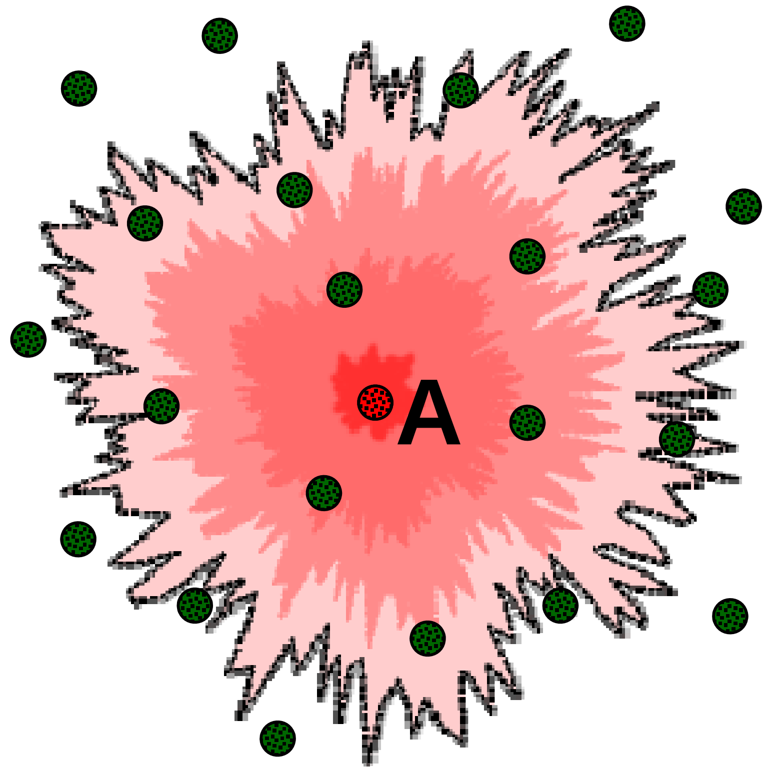
Physical-Layer DoS

- State of the Art:
  - Military hardware
  - Detection of jamming, evasion by physically moving, channel surfing (Xu et al.)
  - Data blurting, schedule switching (Law et al.)
  - Multi-frequency protocols:
    - Bluetooth, Tang et al., Zhou et al.
  - Wormholes to exfiltrate data (Cagalj et al.)
  - Low-density parity codes (Noubir)



Physical-Layer DoS

- Our approach:
  - Hide messages from the jammer
  - Evade the jammer's search
  - Reduce impact of corrupted messages
  - Raise the bar for jamming DoS attackers



► DEEJAM: defeating jamming at the MAC-layer

Contributions

- Define, implement, and show efficacy of four jamming attack classes:
  - interrupt jamming, activity jamming, scan jamming, pulse jamming
- Propose four complementary solutions that together greatly improve communication:
  - frame masking, channel hopping, packet fragmentation, redundant encoding
- Evaluate integrated protocol on MICAz platform to show suitability for popular embedded hardware.
- Empirically show continued communication despite an ongoing attack

Assumptions

- Static wide-area deployment, no mobility
- Lightweight cryptographic primitives available
- Key distribution, time synchronization available
- *Each pair* of neighbors shares K_N, used to generate other keys and pseudo-random sequences.

- Attacker compromises mote or uses mote-class hardware
  - Can use all resources available to regular node

IEEE 802.15.4 Transceivers

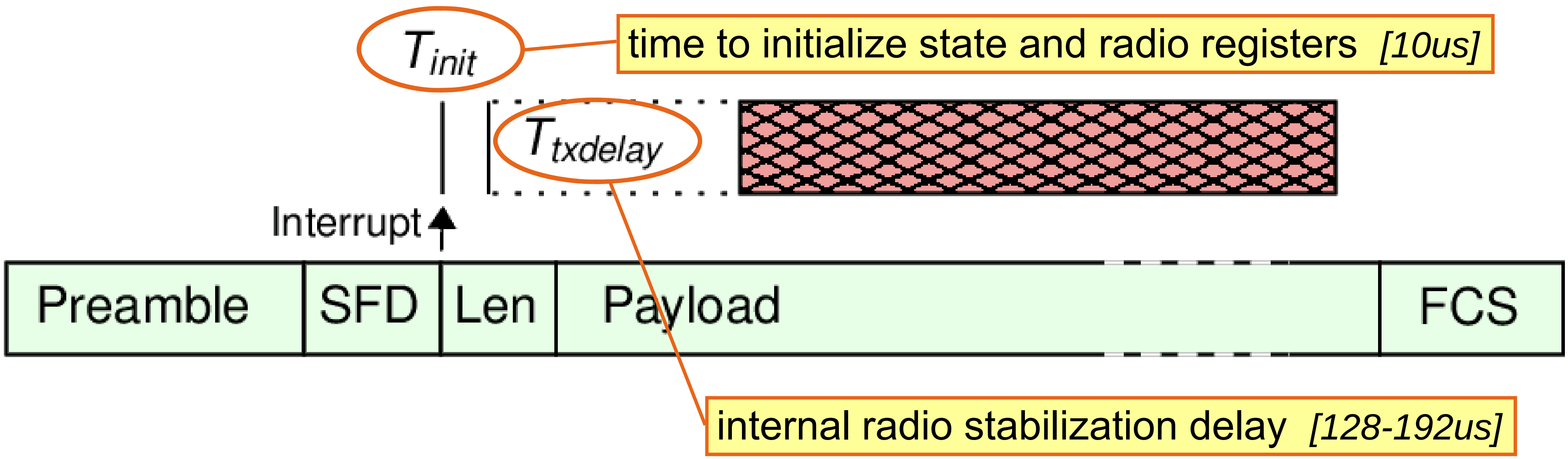
- 802.15.4 defines: 250 Kbps, 16 channels, DSSS, 4-bit symbols, 32 chips/symbol
- Transmit path:
  - micro fills TXFIFO, issues transmit command
  - after small delay, radio chip transmits frame

Preamble	SFD	Len	Payload	FCS

- Receive path:
  - search for DSSS coding
  - sync 4-bit symbols on preamble
  - sync bytes on Start of Frame Delimeter (SFD)
  - buffer frame, signal micro
  - micro reads RXFIFO, parses packet

A1: Interrupt Jamming

- *Attack goal: only jam when message on air*
- Configure radio to generate interrupt on SFD
- In SFD interrupt vector, issue transmit command



- Only need to invalidate Frame Check Sequence

D1: Frame Masking

- *Defense goal: prevent interrupt upon message header reception*
- Neighbors use secret SFD sequence:

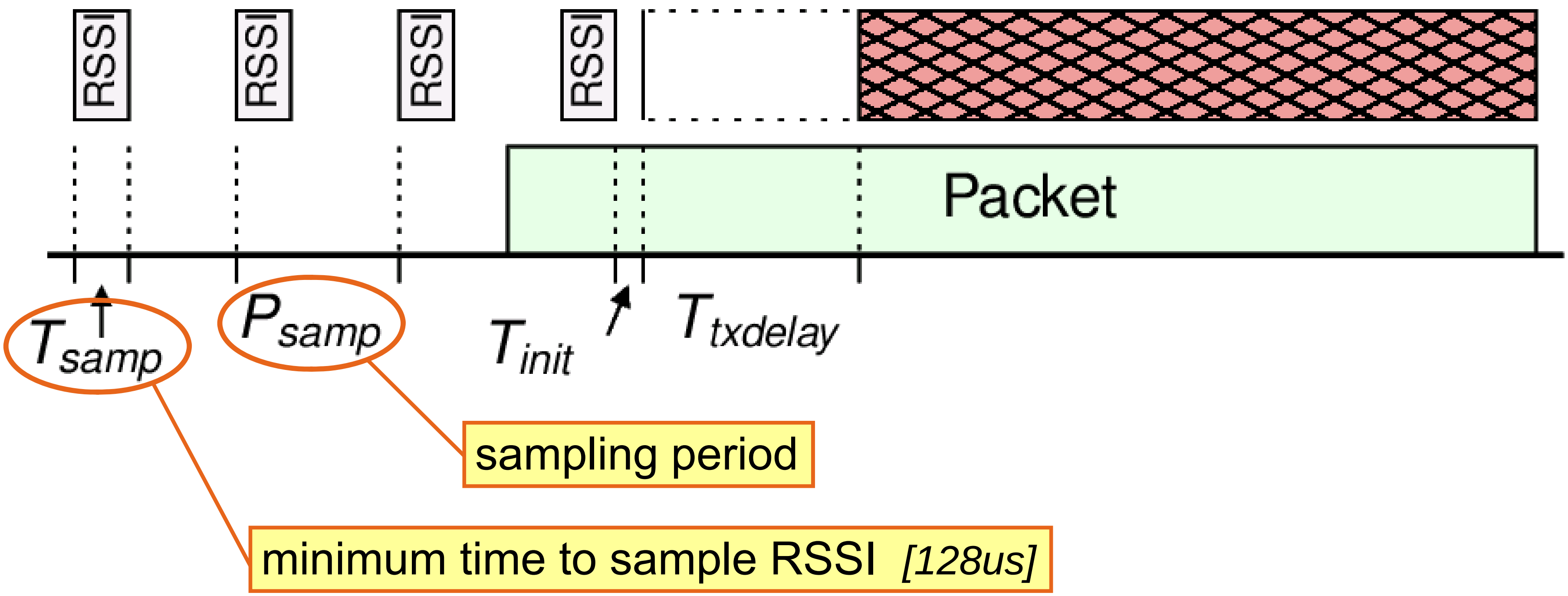
  $K_S = E_{Kn}(0)$

  $SS = \{ E_{Ks}(i) \bmod 2^q \}$, q is length of SFD *[1 or 2B]*

- Without knowing SS, attacker's radio:
  - synchronizes on DSSS encoding in preamble
  - searches for its configured SFD (not SS_i)
  - does not capture message or generate interrupt

A2: Activity Jamming

- *Attack goal: poll channel energy to find message*
- Attacker's micro polls RSSI / CCA output of radio
- When activity is detected, initiate jamming



- Less reliable detection (false positives), more latency

UNIVERSITY of VIRGINIA

D2: Channel Hopping

- *Defense goal: evade activity check*
- Neighbors channel hop according to secret shared sequence:

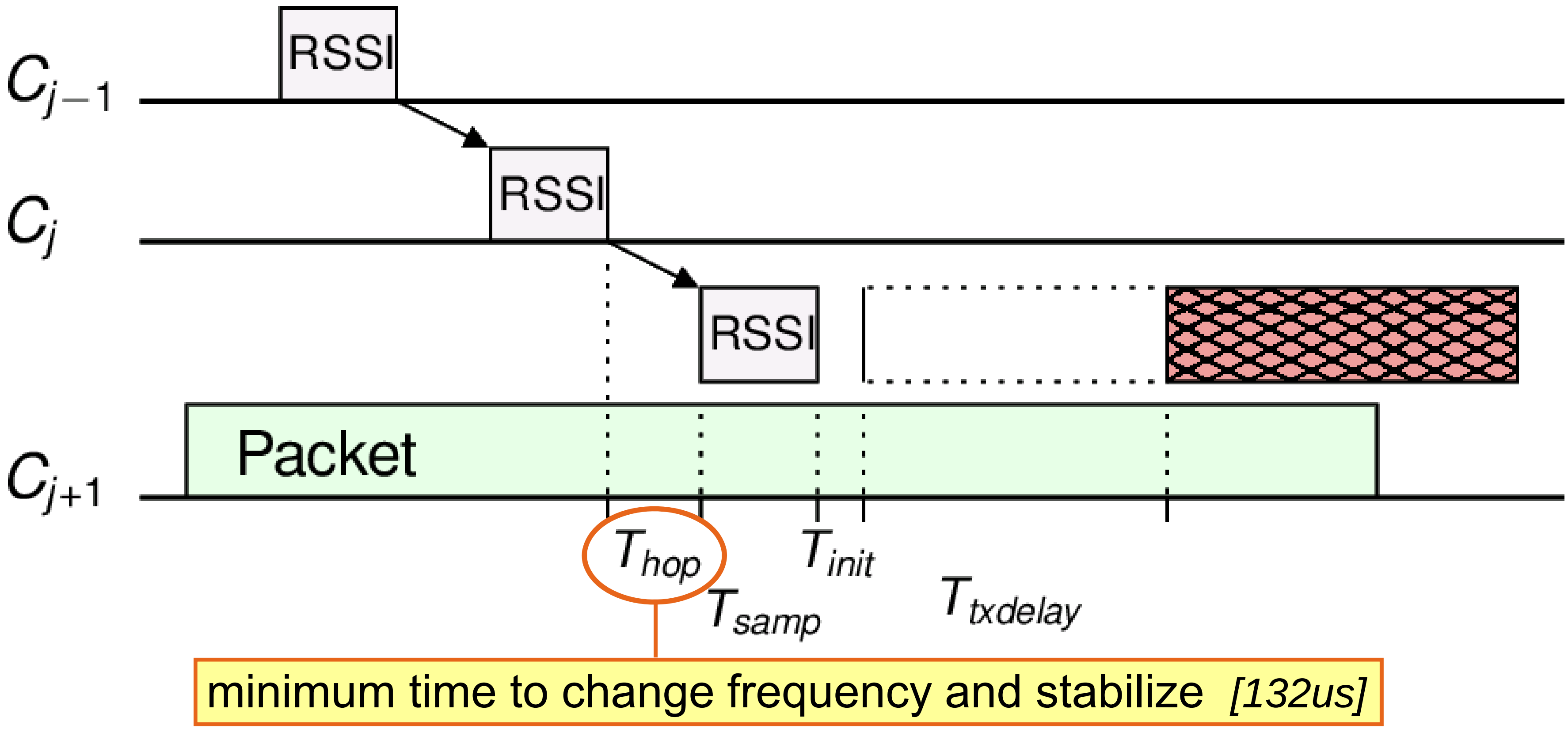
  $K_C = E_{Kn}(1)$

  $CS = \{ E_{Kc}(i) \bmod C \}$, C is number of channels *[16]*

- Attacker has $^1/_C$ chance of sampling correct channel, $^U/_C$ chance of detecting a message for channel utilization U

A3: Scan Jamming

- *Attack goal: find messages and jam*
- Attacker scans channels, checking for activity and jamming if detected



UNIVERSITY of VIRGINIA

A3: Scan Jamming

- For C channels, attacker can always jam if:

$$\frac{T_{pkt} - (T_{init} + T_{txdelay})}{T_{scan}} > C$$

- Since channel is chosen randomly, probability of successful scan jamming is at most:

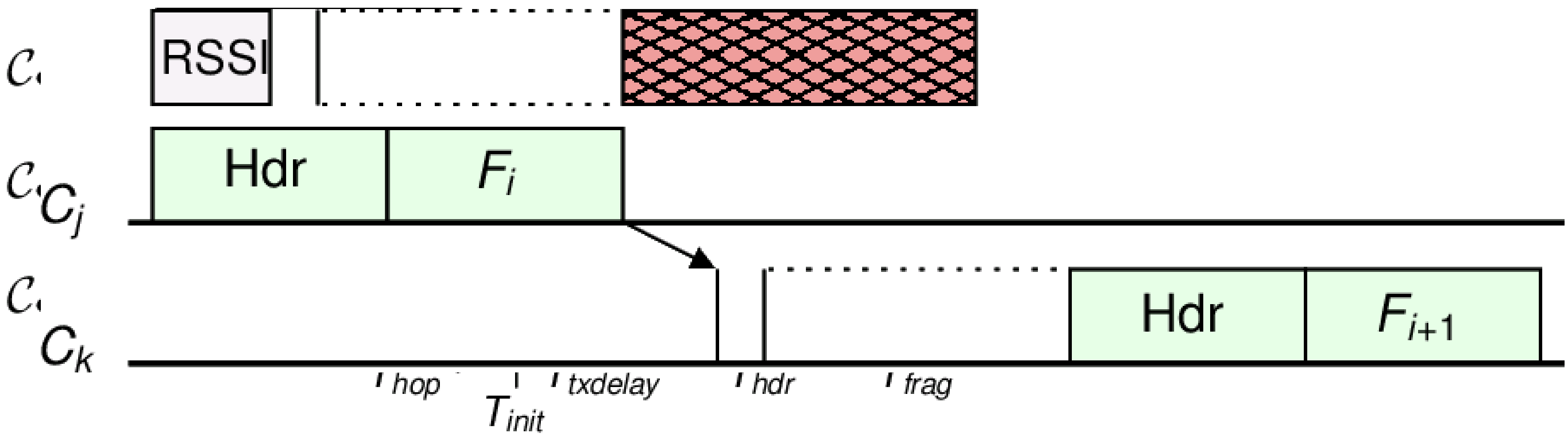
$$\mathcal{P} = \min\left(\frac{T_{pkt} - (T_{init} + T_{txdelay})}{C \cdot T_{scan}},\ 1\right)$$

► Defender wants to increase C and/or decrease T_{pkt}

D3: Packet Fragmentation

- *Defense goal: hop away before jammer reacts*

- Fragment packets based on minimum reactive jam time
- Reassemble sequence of fragments at receiver



UNIVERSITY of VIRGINIA

A4: Pulse Jamming

- *Attack goal: blindly disrupt fragments*
- Transmit with duty cycle sufficient to corrupt any fragments present on a chosen channel:

$$T_{hdr} / (2T_{hdr} + T_{frag}) \qquad [< 50\%]$$

- Disadvantages:
  - Not reactive, not stealthy
  - Cannot selectively jam by inspecting header

D4: Redundant Encoding

- *Defense goal: recover from damaged fragments*

- Redundantly encode fragments with configurable rate R
- (Some) fragments corrupted on a pulse jammed channel are recoverable
- Requirement for CS: $C_i \neq C_{i+1}$

DEEJAM MAC Protocol Summary

- Compute FCS for entire packet
- Divide into small fragments
- Encode redundantly with rate ***R***
- Assign SFD from *receiver's* current ***SS***
- Transmit on channel in *receiver's* current ***CS***

► Channel hopping by itself is not sufficient

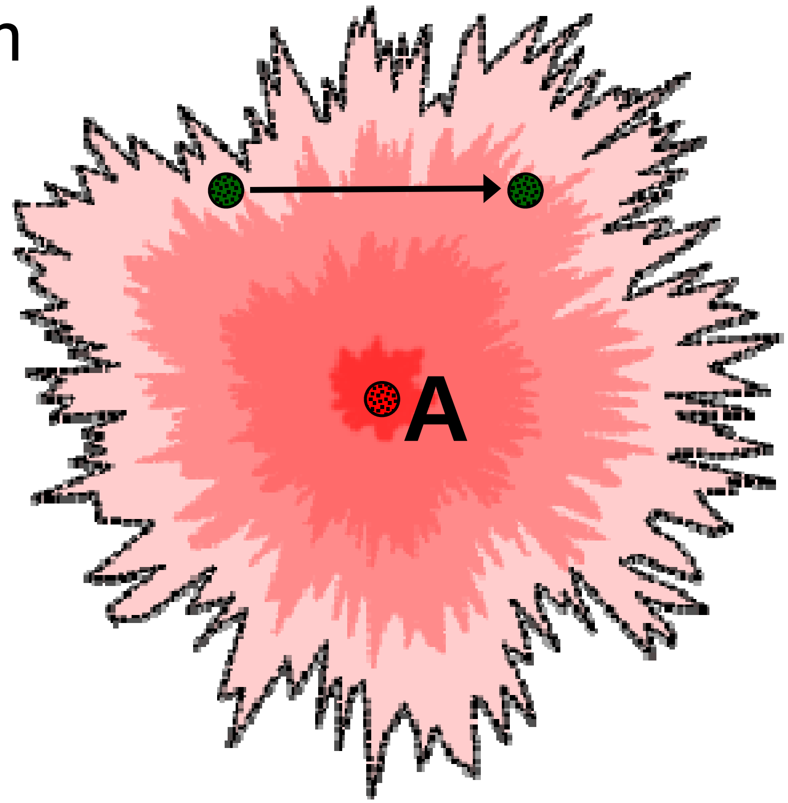
► Cannot assume a priori that attacker pulse jams

Implementation

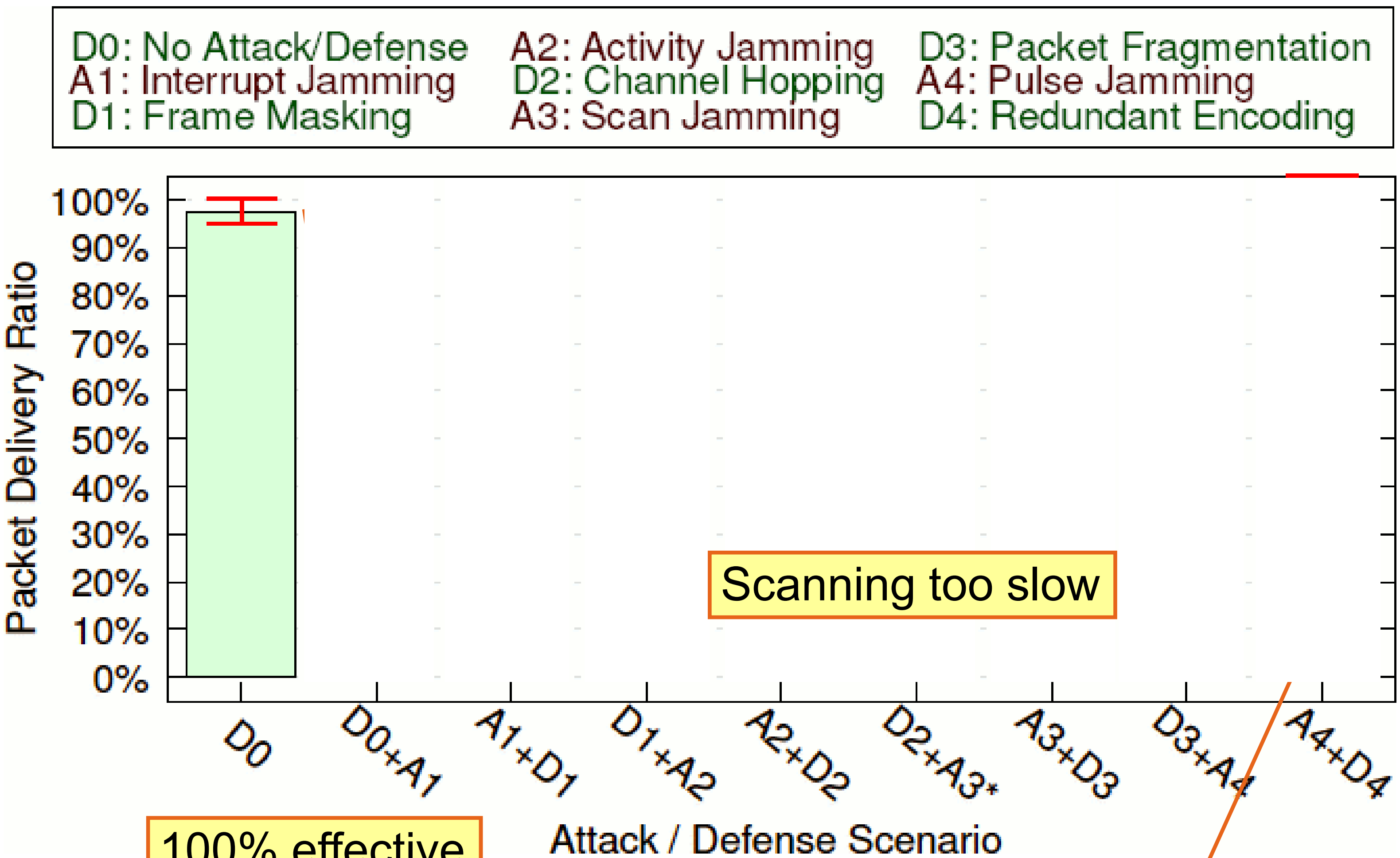
- Prototype implementation in nesC for TinyOS, using MICAz's TI Chipcon CC2420
- To minimize fragment length:
  - shortened $T_{txdelay}$ to 4B
  - shortened preamble to 1B
  - removed unused IEEE 802.15.4 MAC fields
- Interrupt jamming: byte-serial receive mode + FIFOP interrupt with threshold zero

Evaluation

- Sender to receiver, attacker jamming
- Five 60s runs, 32 msg/s, 39B total length
- Total of 9595 messages per datum
- Use 16 channels
- Transmit power -7 dBm
- Measure:
  - Packet Delivery Ratio with attacks
  - Jamming effort
  - PDR with no attacks



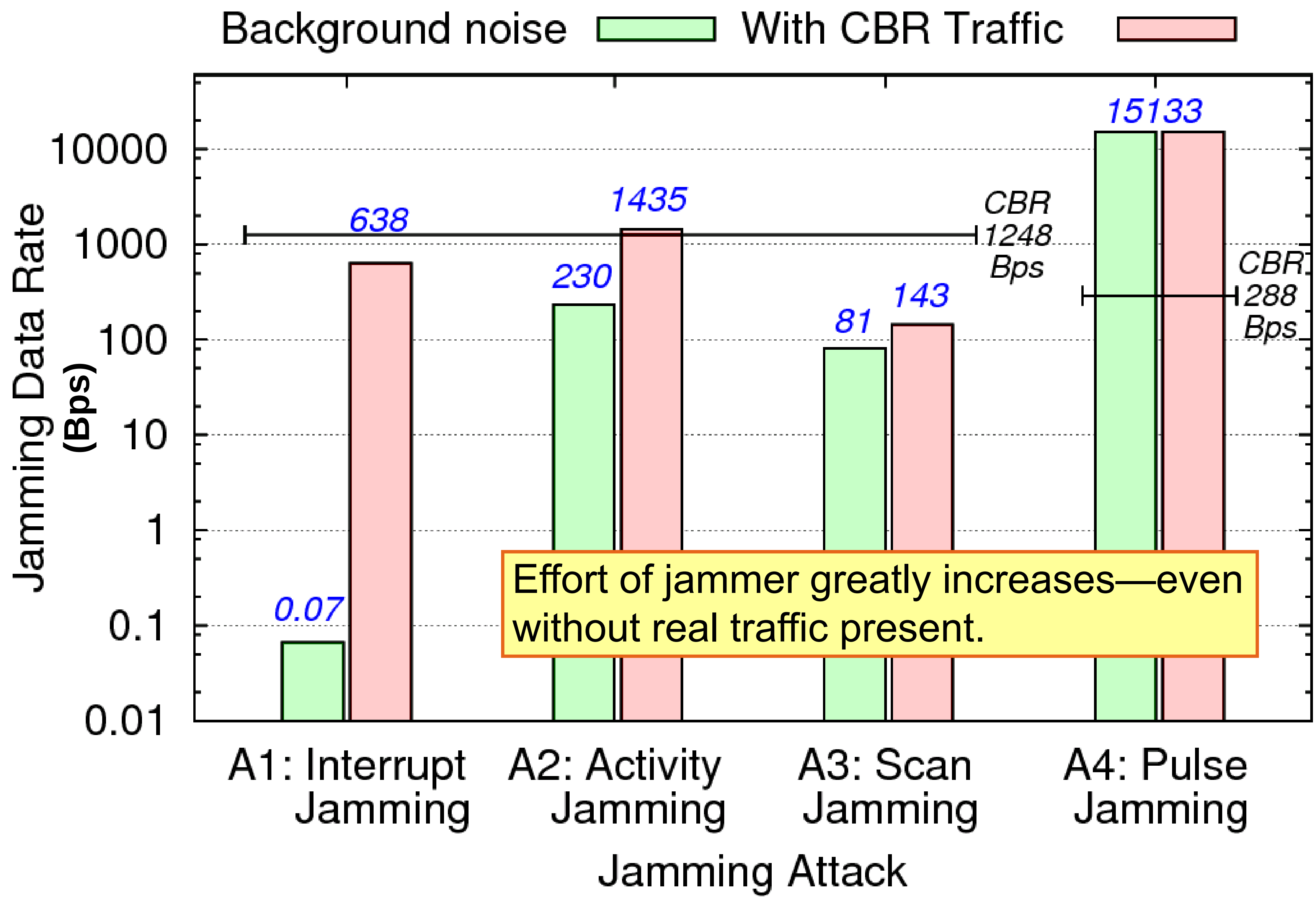
Performance (with attacks)



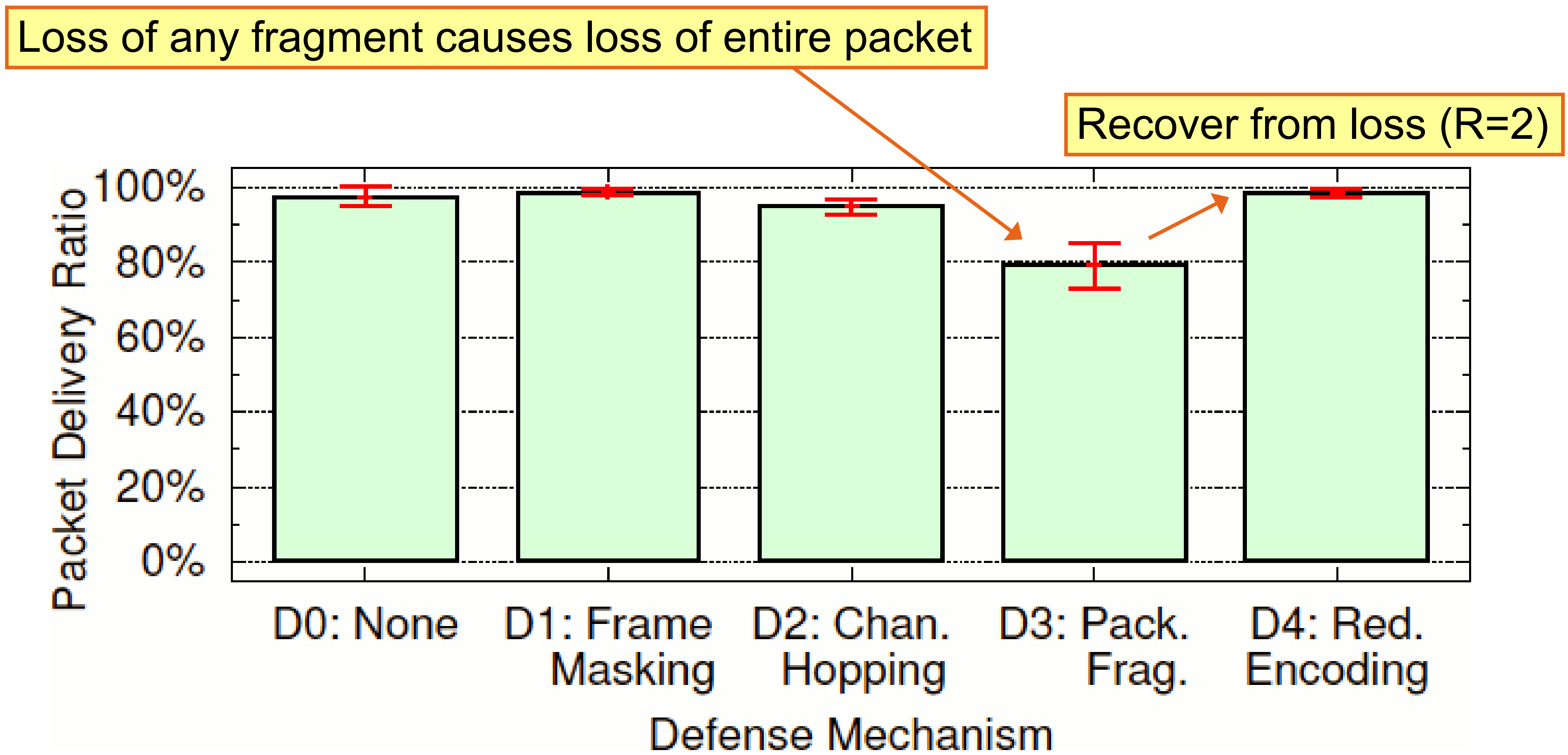
100% effective

89% PDR despite pulse jamming

Jamming Effort



Performance (no attacks)



- Impact of DEEJAM on PDR with no attacks is small

Conclusions

- With no defense, a stealthy interrupt jamming attack is 100% effective
- Adding defenses forces attacker to adapt
- Ultimately, despite an active pulse jamming attack, PDR drops by only 11%

► For many systems, recovery of performance during attack is worth the overhead

► More powerful jamming is possible—but without countermeasures it is not necessary

End