# Low-Power System Design

227-0781-00L

Fall Semester 2019

Jan Beutel

# Plan for Today

- MAC Layer Techniques
  - Contention-based Rendezvous – Bootstrapping: LPL/LPP
  - Dedicated acknowledgements, multiple channels: A-MAC
  - Arbitration using controlled collisions: StrawMan
  - Distributed scheduling: DOZER
  - Constructive interference: A-MAC
  - Network Flooding: GLOSSY

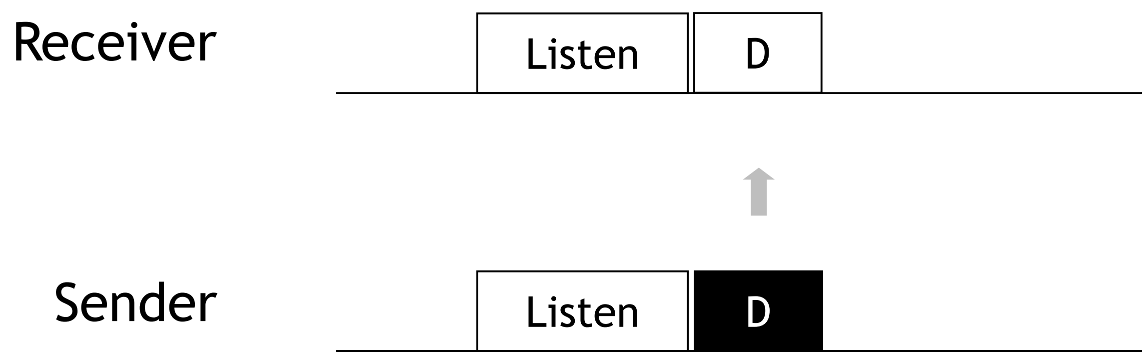
- Present metrics used for performance analysis

Networked Embedded Systems

# MAC LAYER TECHNIQUES – CONTENTION-BASED RENDEZVOUS

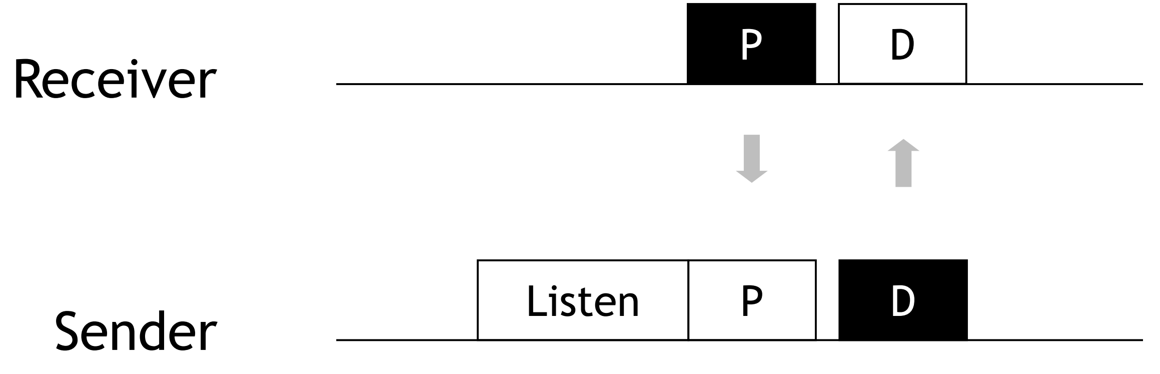
# Simple Sender-Initiated MAC

- Sender triggers communications by transmitting data
- Receiver is listening



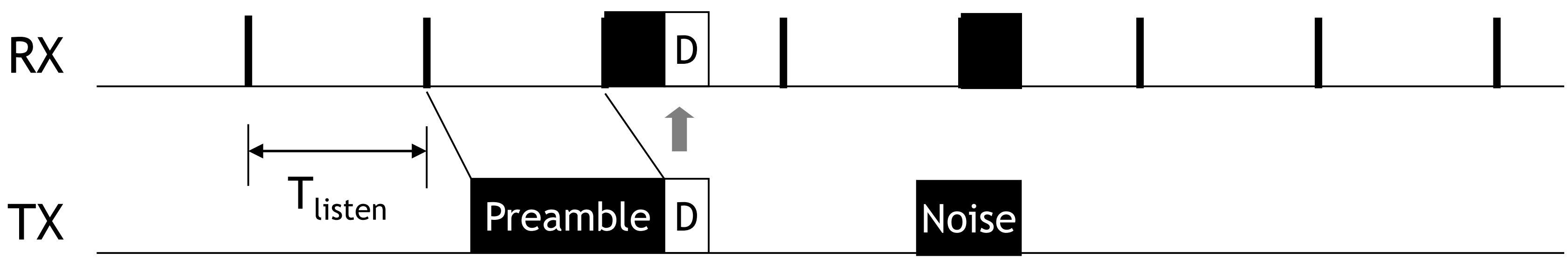
# Simple Receiver-Initiated MAC

- Receiver triggers exchange by transmitting a probe
- Sender receives probe and sends data
- Low-power probing (LPP)

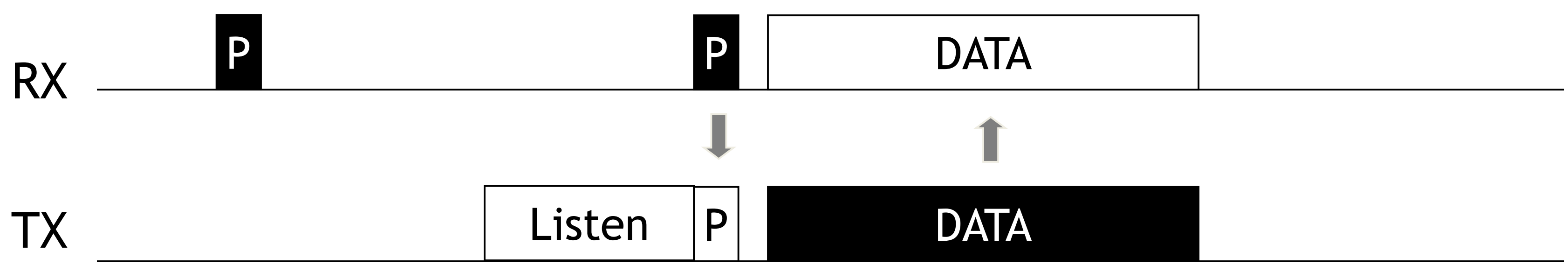


# MAC Layer Decision: Stay awake or go to sleep?

## Sender-Initiated: Channel Sampling

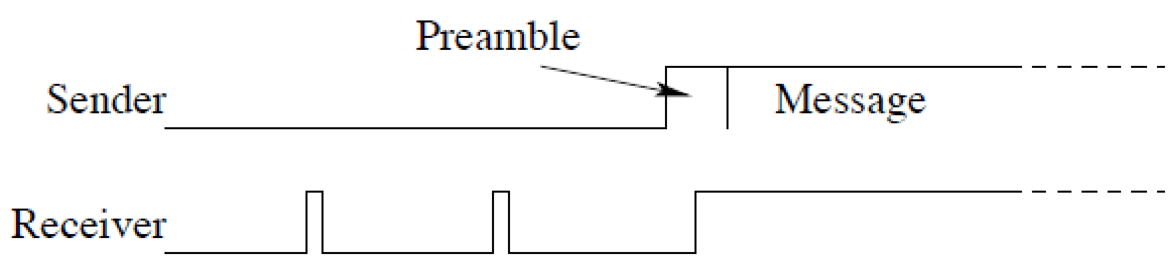



## Receiver-Initiated: Channel Probing



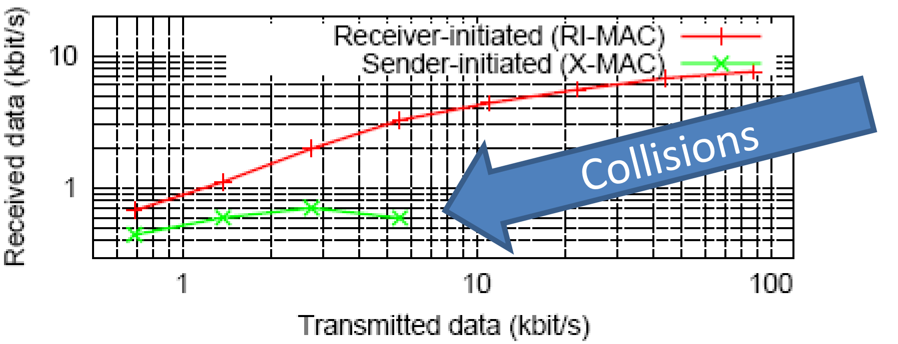
# Optimizing LPL: Shorter Preamble Sampling

- Bookkeeping to avoid sending out long preambles
  - Maintaining the phase offset (clock) to selected neighbors
  - Start transmitting a message just before receiver wakes
  - Synchronized transmit/receive
  - Piggybacking of local phase offset on ACKs of the underlying CSMA protocol
- Benefits
  - WiseMAC is able to squeeze out up to 80% (20 out of 25 ms) of TX cost and up to 67% (10 out of 15ms) of RX costs
  - Shortening the preambles also reduces overhearing by nodes other than the sender/receiver pair



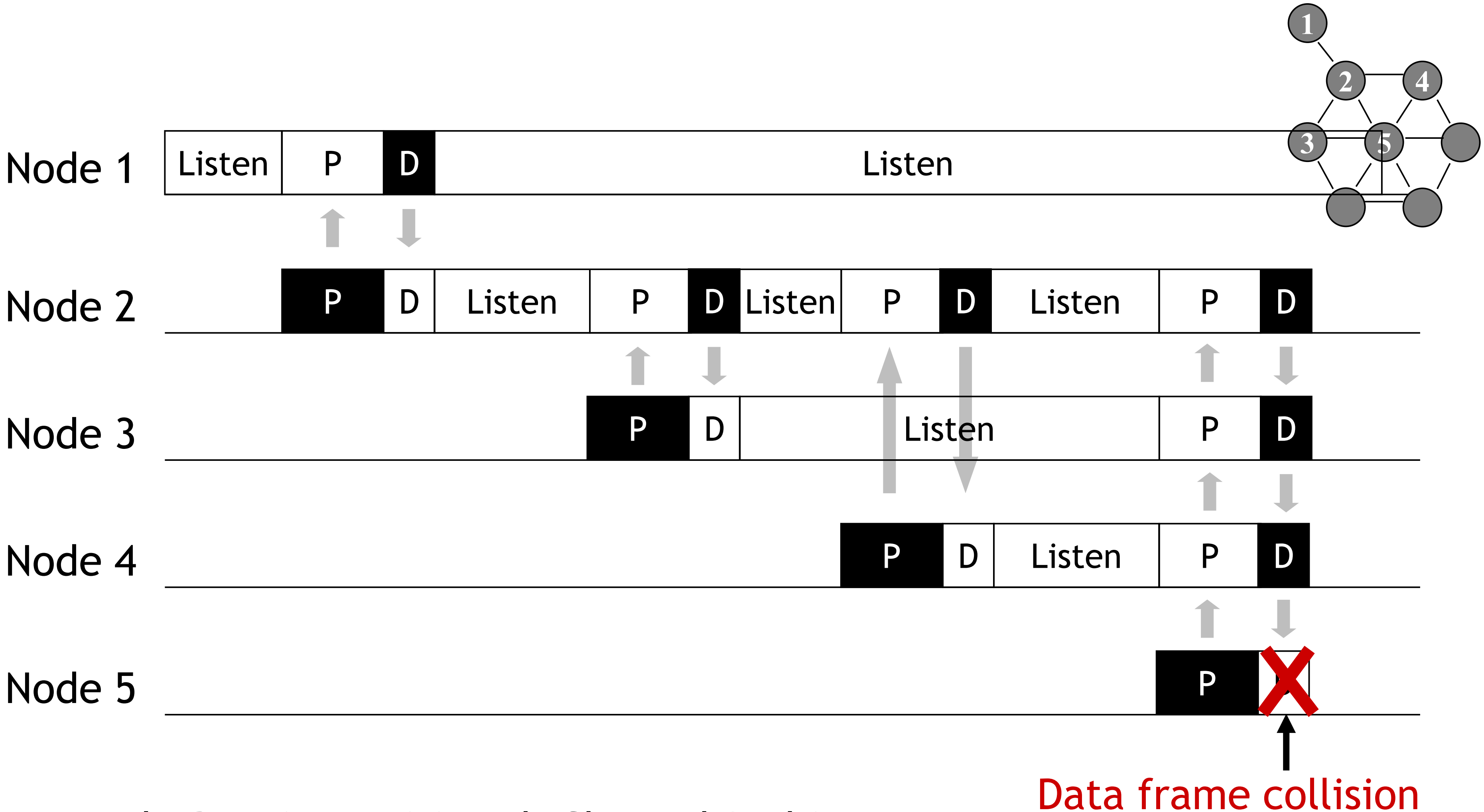
# Receiver vs. Sender-initiated Tradeoff

- Receiver-initiated Pro's
  - Handle hidden terminals better than sender-initiated ones
  - Support asynchronous communication w/o long-preambles
  - Support extremely low duty cycles or high data rates




- Receiver-initiated Con's
  - Probe (LPP) is more expensive than channel sample (LPL)
    - Baseline power is higher
  - Frequent probe transmissions
    - Could congest channel & increase latency
    - Could disrupt ongoing communications
    - Channel usage scales with node density rather than traffic

# Scaling to Larger Networks



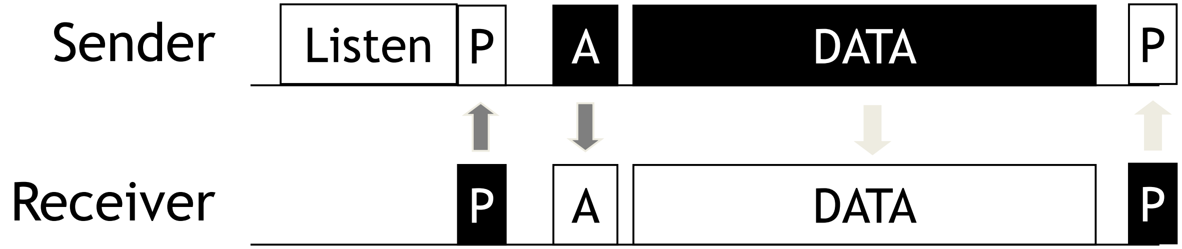

Example Receiver-Initiated: Channel Probing

Networked Embedded Systems

# MAC LAYER TECHNIQUES – DEDICATED ACKNOWLEDGEMENTS, MULTIPLE CHANNELS

# 802.15.4 Receiver-Initiated Link Layer

*Is it possible to design a general-purpose, yet efficient, receiver-initiated link layer?*

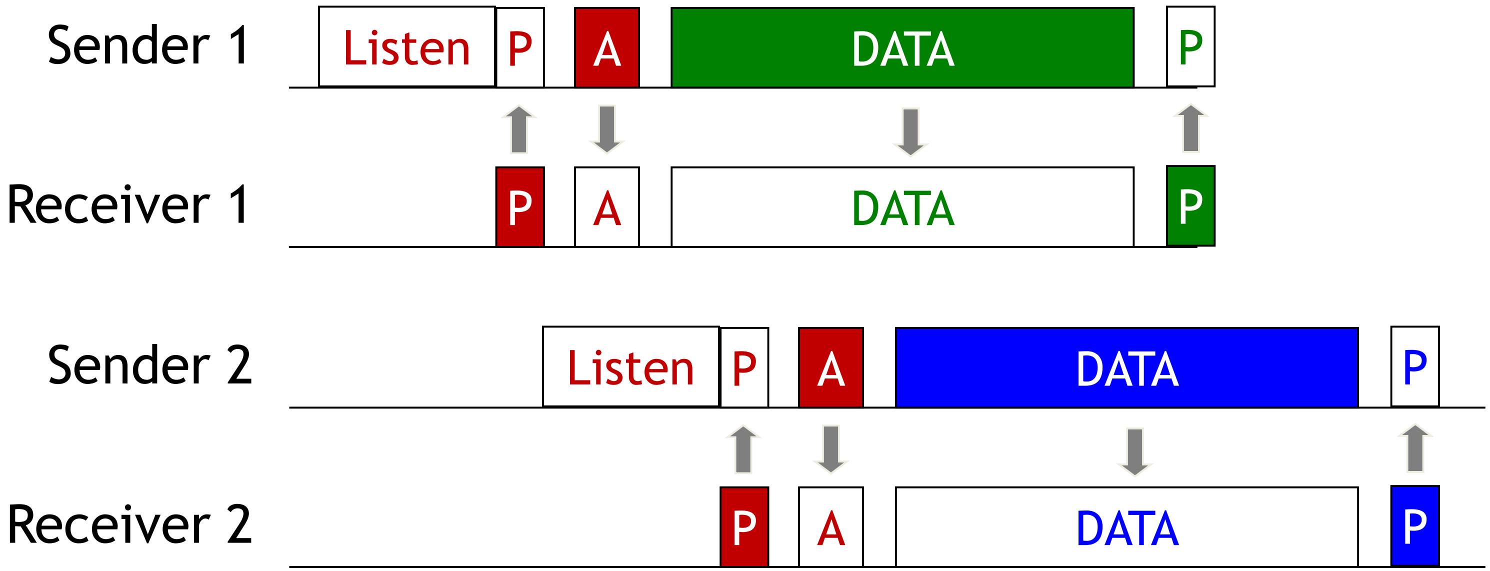



**A-MAC**

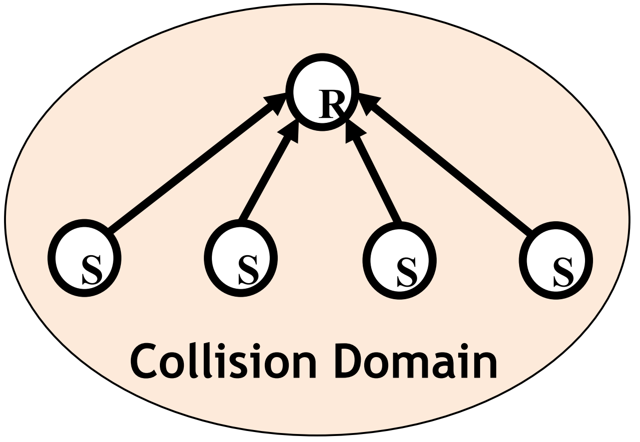
Dutta, P., Dawson-Haggerty, S., Chen, Y., Liang, C.-J. M., & Terzis, A.. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*.

# A-Mac Parallel Multichannel Data Transfers

- Arbitration using signaling in preambles (like Dozer!)
- Selective scheduling of data-senders
- Use control, data (1), and data (2) channels



# A-MAC Offers Modest Incast Performance



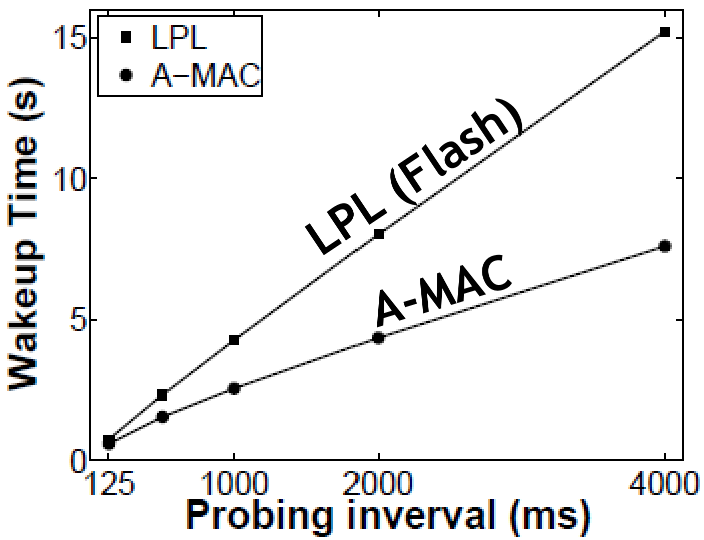

| MAC | No. of Senders | Packet Delivery Ratio | | |
|---|---|---|---|---|
| | | Avg | Min | Max |
| RI-MAC | 1 | 99.9% | — | — |
| | 2 | 97.5% | 97.3% | 97.7% |
| | 3 | 95.6% | 95.0% | 96.8% |
| | 4 | 90.7% | 90.3% | 90.9% |
| A-MAC | 1 | 99.9% | — | — |
| | 2 | 99.3% | 98.2% | 100% |
| | 3 | 99.3% | 98.3% | 99.5% |
| | 4 | 98.5% | 96.7% | 99.5% |

ETH zürich

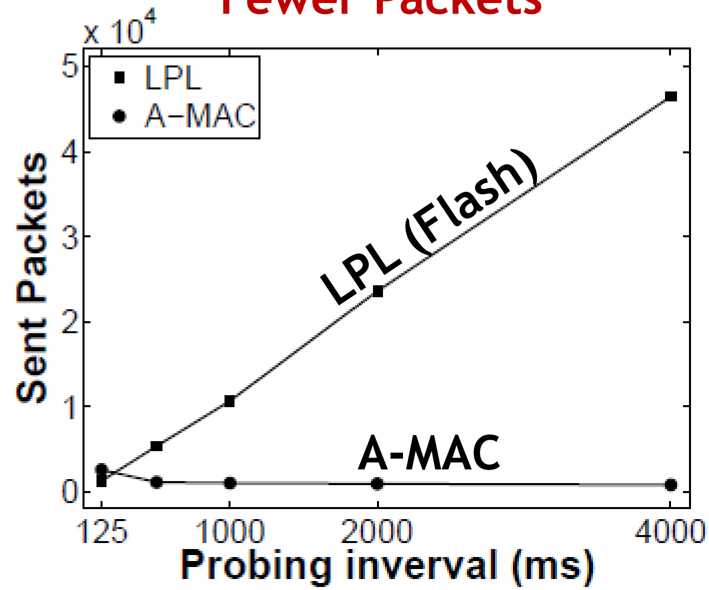
# A-MAC Network Wakeup

- Wakes up the network faster and more efficiently than LPL (Flash) flooding

**Faster Wakeup**

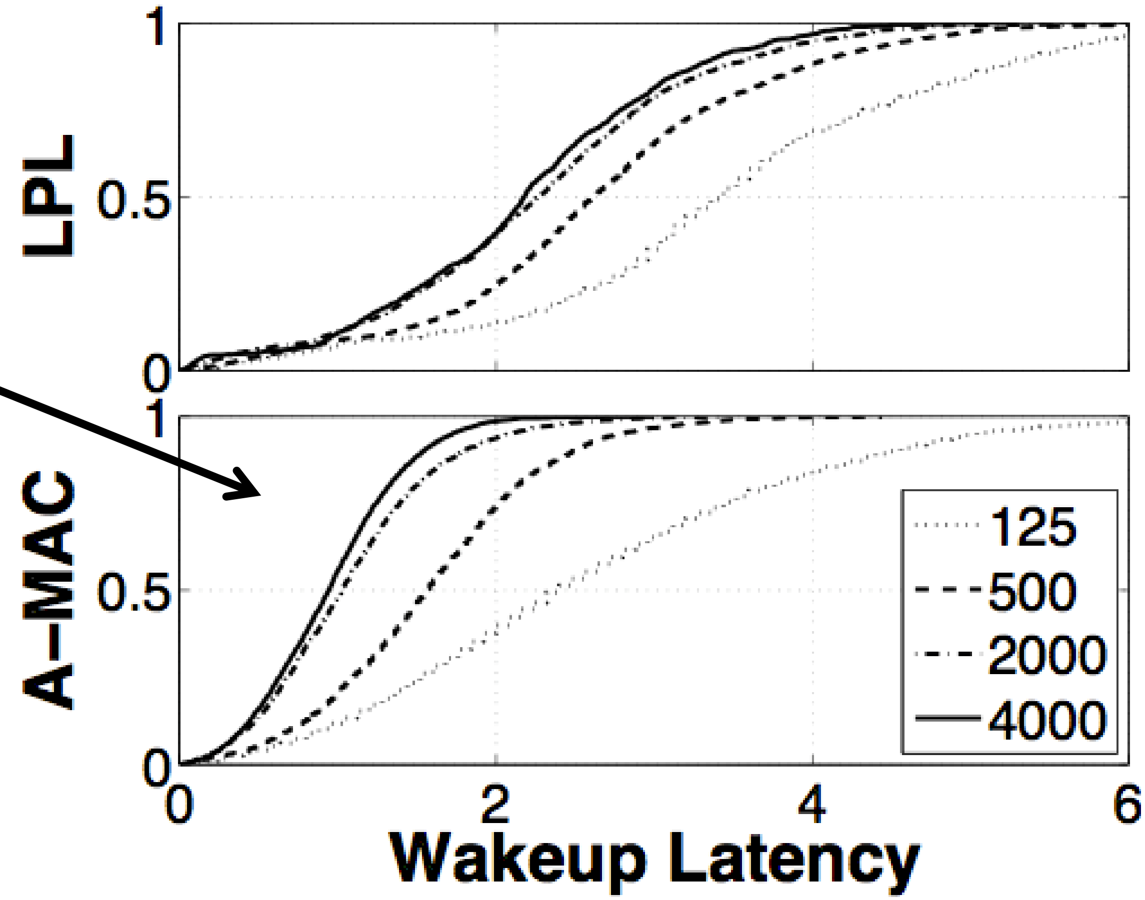



**Fewer Packets**



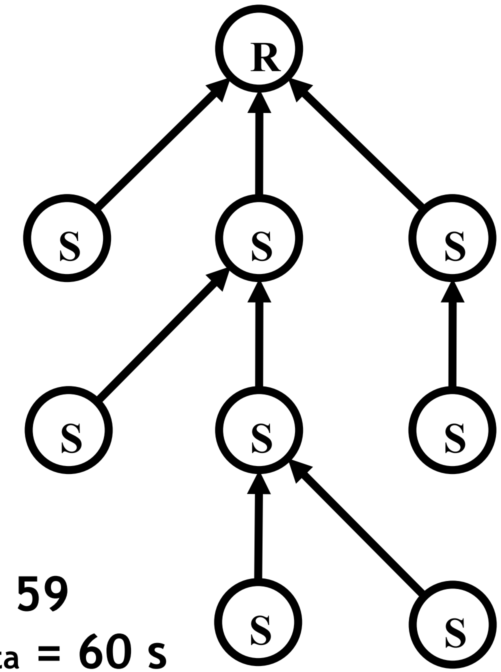
# A-MAC Works Well at Low Duty Cycles

$T_{probe}$ = 4,000 ms
$P_{avg}$ = 63 µW
$I_{avg}$ = 21 µA
N = 59

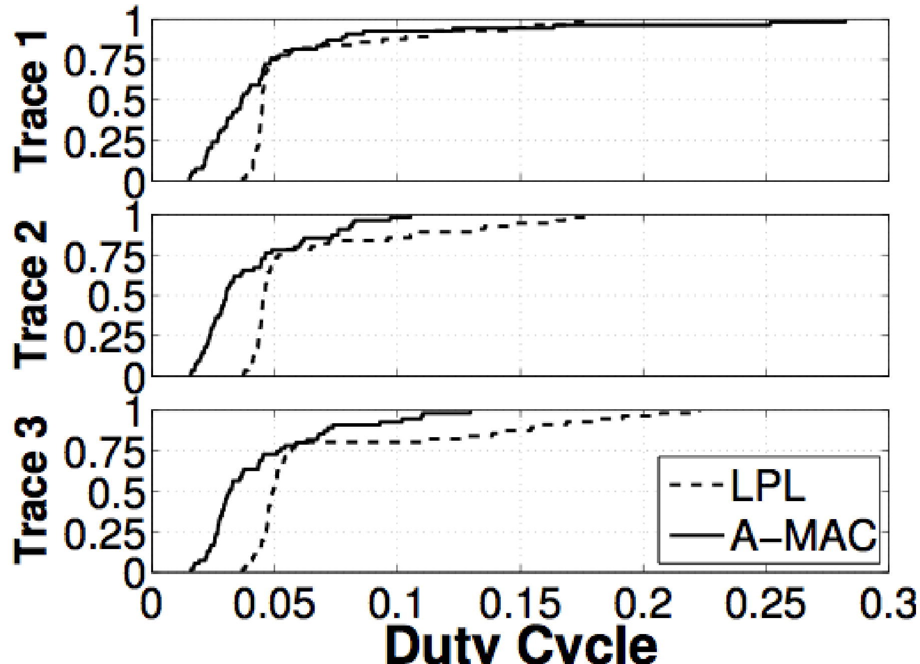



ETH zürich

# A-MAC Beats LPL/CTP Combinations




N = 59
$T_{data}$ = 60 s
$T_{probe}$ = 500 ms




| | LPL | A-MAC |
|---|---|---|
| Average Duty Cycle | 6.36% | 4.44% |
| Average Packet Delivery Ratio | 95.1% | 99.7% |
| Average Hop Count | 7.34 | 4.85 |
| Maximum Hop Count | 14 | 13 |

Networked Embedded Systems

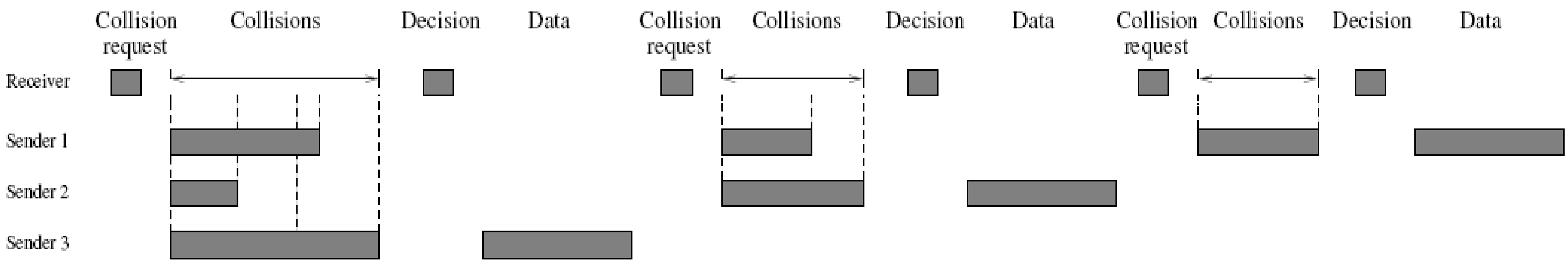
# MAC LAYER TECHNIQUES – ARBITRATION USING CONTROLLED COLLISIONS

# Collision Arbitration with StrawMan

Send Collision request

Reply longest length

Another request




Random length Packet
7 bytes granularity
(224us)

Winner send data

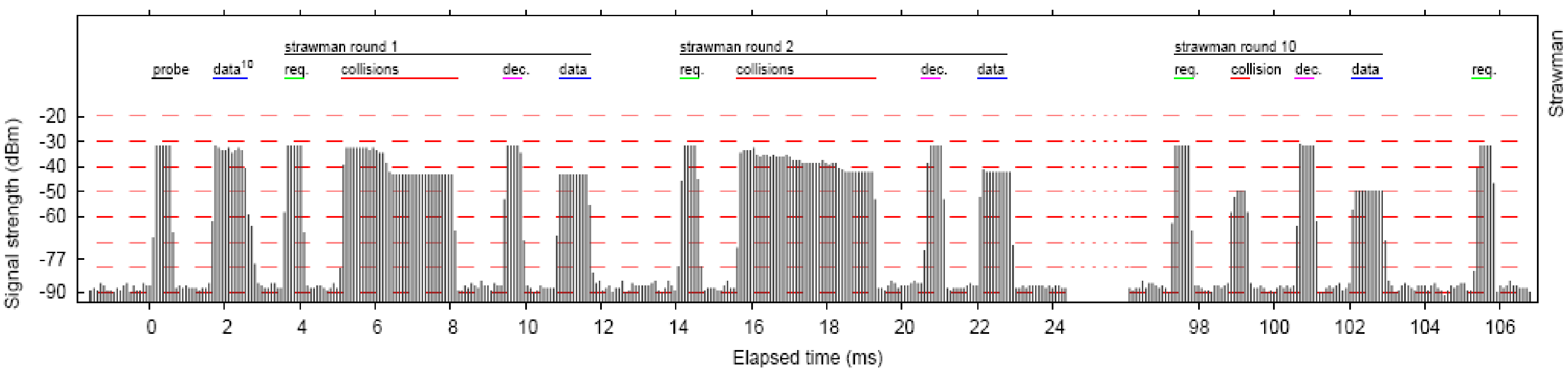
Until every sender sent its data

- Multi-channel operation
  - Initial probe at pre-determined channel
  - Rest of communication at the other channel

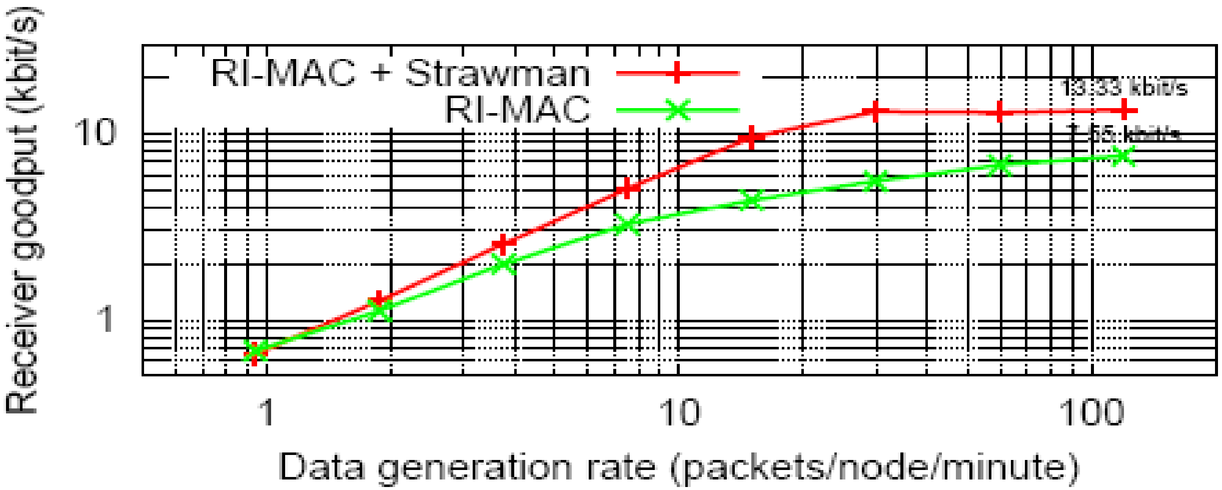
Österlind, F., Mottola, L., Voigt, T., Tsiftes, N., & Dunkels, A. (2012). Strawman: Resolving Collisions in Bursty Low-Power Wireless Networks. In *IPSN '12* (p. 161). New York, New York, USA: ACM Press.

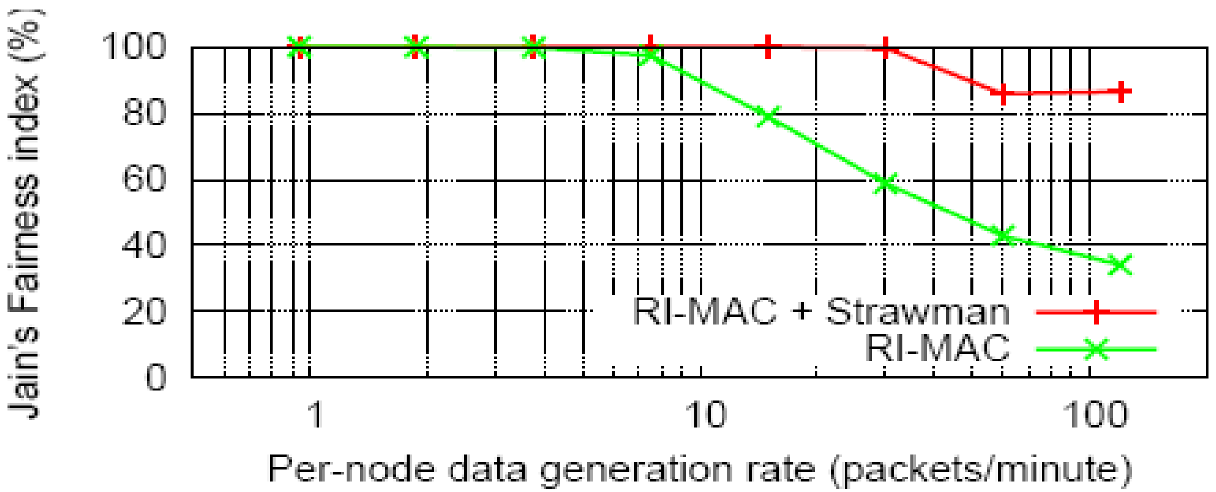
# StrawMan Performance

- Contiki + Tmote Sky
- RI-MAC
  - Version 1: Strawman + multi-channel operation
  - Version 2: random backoff (geometric distribution)
- Transmissions of COLLISION packets are synchronized
  - Receiver knows exactly when they occur
- Max COLLISION packets length is fixed



# StrawMan: Goodput and Fairness

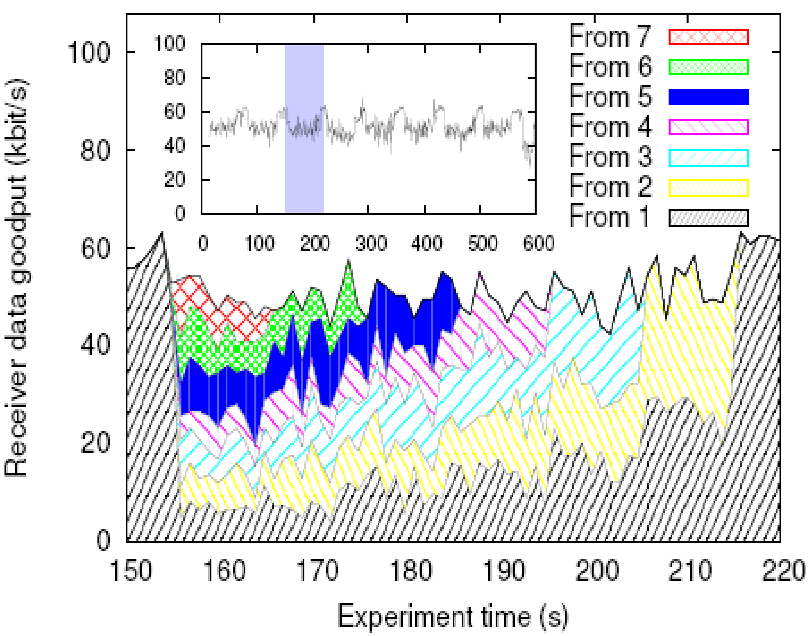


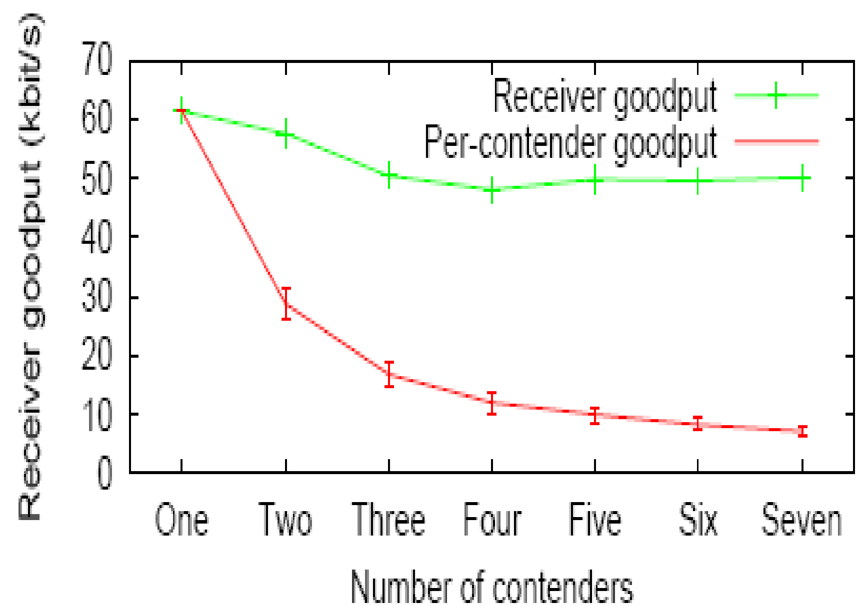




ETH zürich

# StrawMan: Reacting to Sudden Traffic Bursts

- 1-hop network with 8 nodes
  - Measuring the resulting goodput
  - Always contend
- Vary number of active contenders every 10s




ETH zürich

# StrawMan: Multi-hop Data Collection

- 82 nodes in the TWIST testbed
  - Multi-hop topologies (at least 4 hops)
  - Contiki Collect protocol
- Traffic patterns
  - No traffic (NT)
  - Periodic traffic (PT): 1 pkt every 5 minutes
  - Bursty traffic (BT):
    - Instantaneously generate 1 pkt on 8 randomly-selected nodes

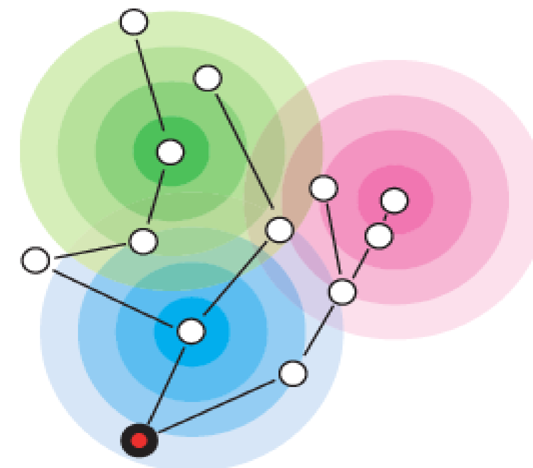
| | RI-MAC + Strawman | RI-MAC |
|---|---|---|
| NT radio duty cycle (%) | 0.34 | 0.40 |
| PT radio duty cycle (%) | 3.94 | 4.40 |
| BT radio on-time (sec) | *4.53* | *8.16* |

Networked Embedded Systems

# MAC LAYER TECHNIQUES – DISTRIBUTED SCHEDULING

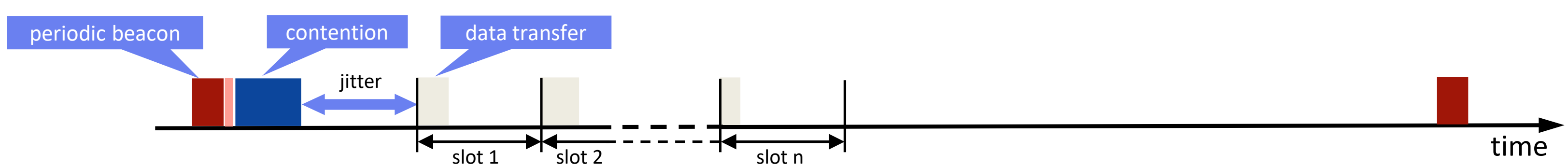
# Hybrid Protocol Schemes

- Dozer ultra low-power data gathering system
  - Beacon based, 1-hop synchronized TDMA
  - Tree-based routing towards a sink
  - Optimized for ultra-low duty cycles
  - **0.167%** duty-cycle, **0.032mA** (@ 30sec beacons)



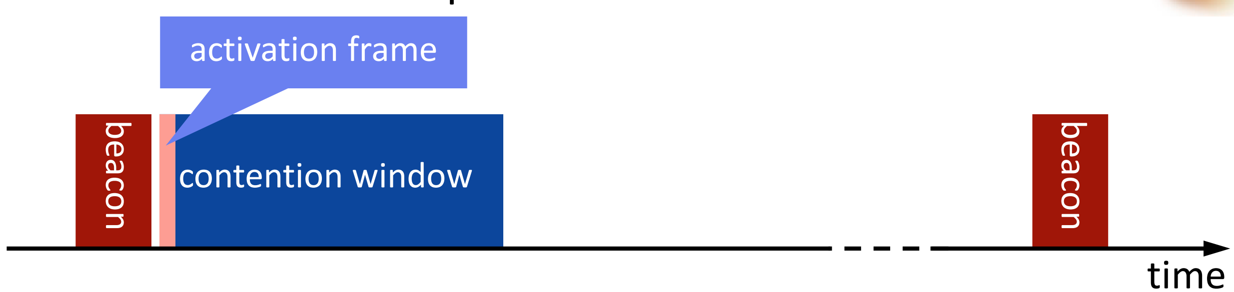
[Burri, N., von Rickenbach, P., & Wattenhofer, R. (2007). Dozer: Ultra-Low Power Data Gathering in Sensor Networks. 2007 6th International Symposium on Information Processing in Sensor Networks, 450–459.]

- Application is integrated with the protocol
  - Dynamic adaptation
  - Back-off randomization for diversity
  - Jitter adaptation over multiple hops
  - Adaptive duty-cycle accounts for long-term loss of connectivity



# Dozer System

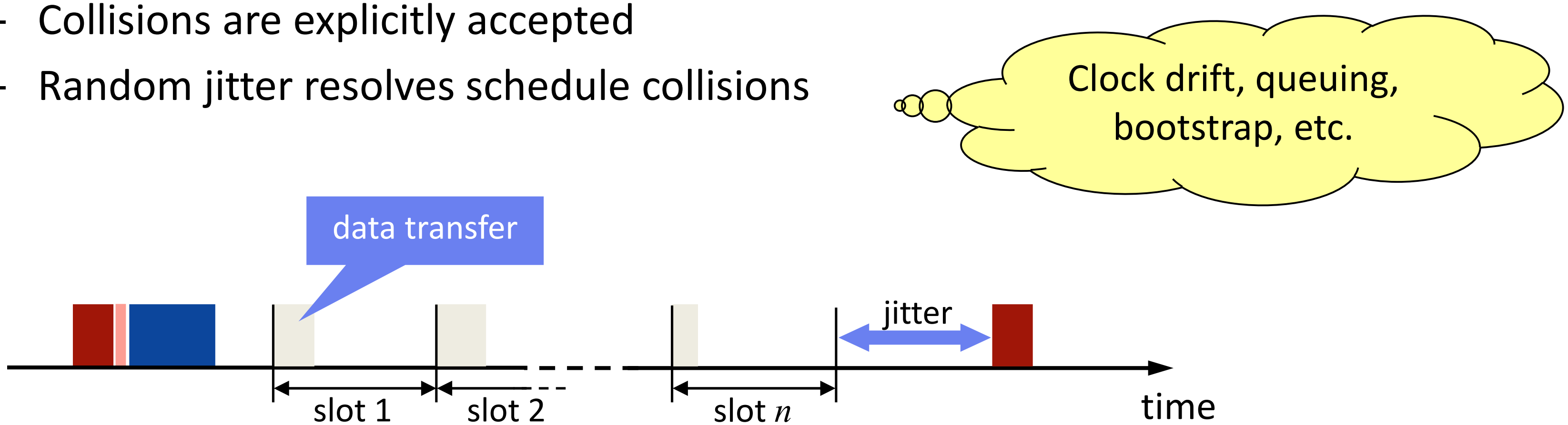
- Tree based routing towards data sink
  - No energy wastage due to multiple paths
  - Current strategy: SPT
- TDMA based link scheduling
  - Each node has two independent schedules
  - No global time synchronization
- The parent initiates each TDMA round with a beacon
  - Enables integration of disconnected nodes
  - Children tune in to their parent’s schedule

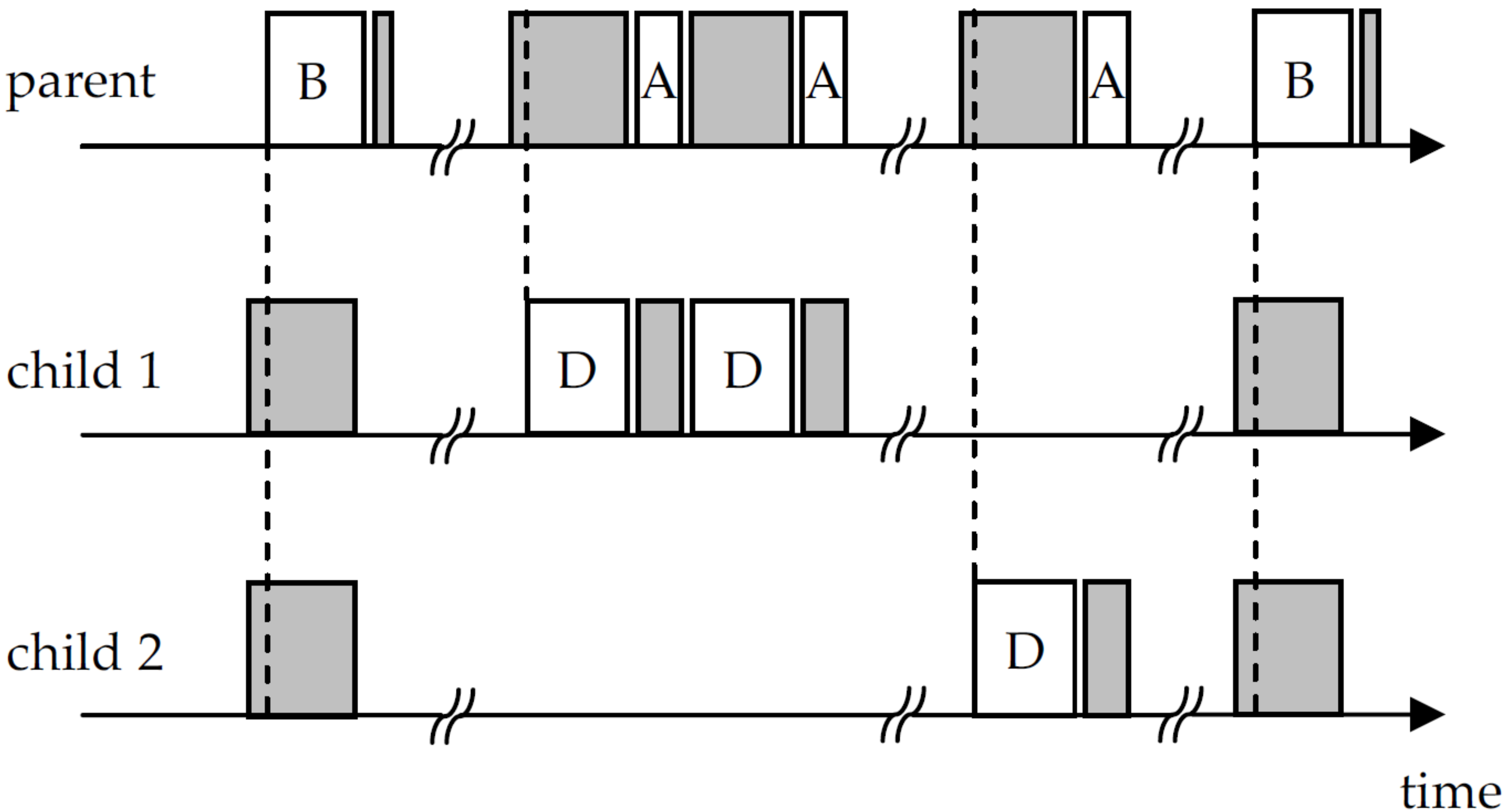
# Dozer System

- Parent decides on its children data upload times
  - Each interval is divided into upload slots of equal length
  - Upon connecting each child gets its own slot
  - Data transmissions are always ack'ed

- No traditional MAC layer
  - Transmissions happen at exactly predetermined point in time
  - Collisions are explicitly accepted
  - Random jitter resolves schedule collisions

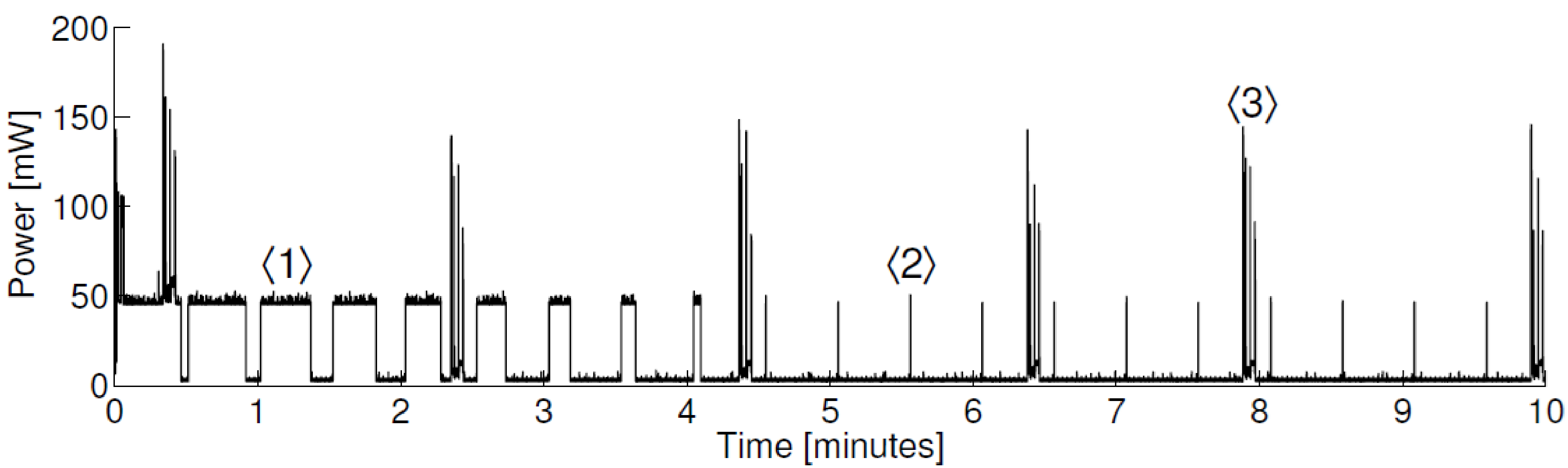


# Dozer Scheduled Data Transfers



# Graceful Degradation & Effective Retries

- Configurable beacon synchronization timeouts
  - Typically 3-5 retries
- Adaptive scanning activity
  - Reduction on intermittent loss of connectivity
  - Energy savings on bootstrapping and longer network failures

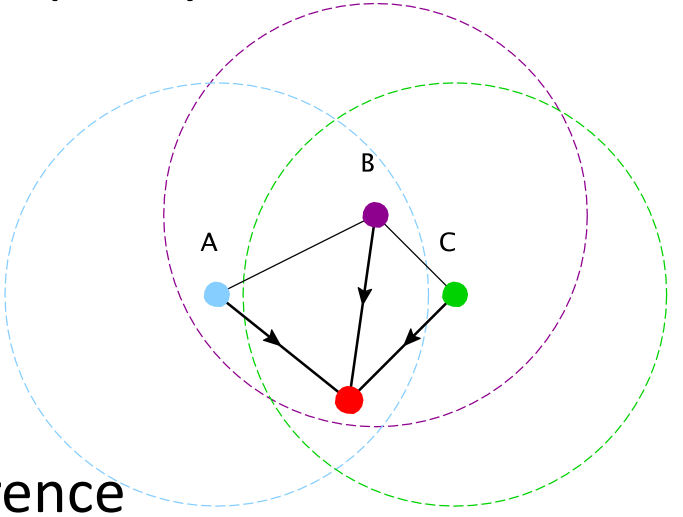


Low-Power System Design

# MAC LAYER TECHNIQUES – CONSTRUCTIVE INTERFERENCE

# Wireless Interference

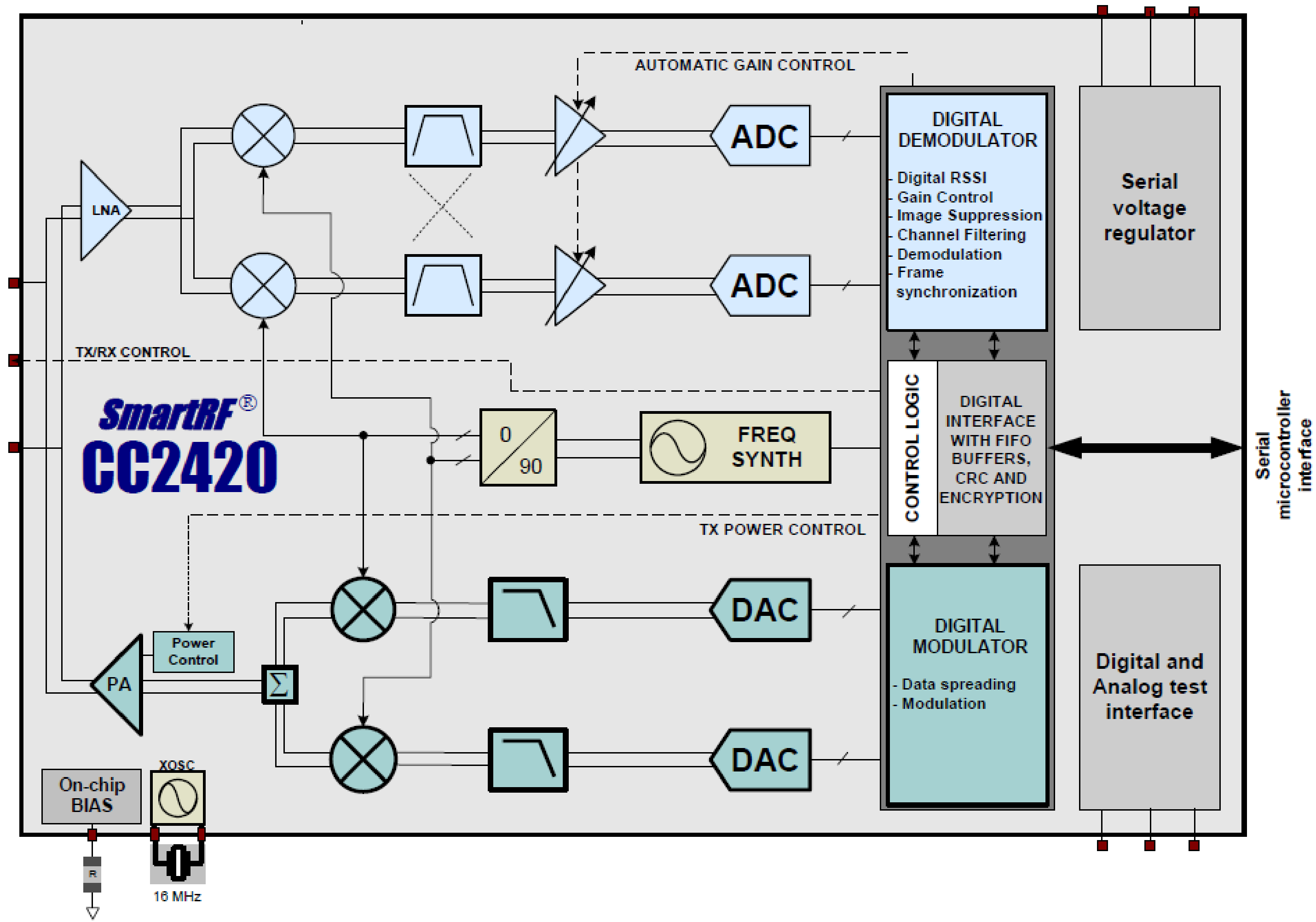
- Spatially close wireless stations transmit signals at the same time and with the same frequency




Stations A, B, and C transmit signals to a common receiver R

- Destructive interference
  - Interference generally reduces the probability that a receiver correctly detects the information
- Constructive interference
  - A receiver detects with high probability the superposition of the signals generated by multiple transmitters

# IEEE 802.15.4 Uses DSSS Modulation

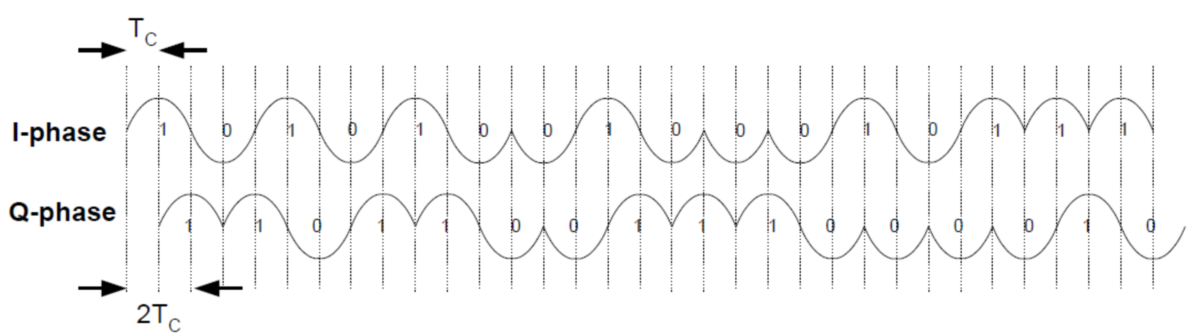



[Texas Instruments]

# IEEE 802.15.4 Modulation

- 1 Byte is divided into 2x 4-Bit Symbols
- Each Symbol is mapped to a pseudo-random noise (PN) sequence with 32 chips (2 MChips/sec)
- Offset-Quadrature Phase Shift Keying (O-QPSK) with half-sine chip shaping (equivalent to MSK modulation)

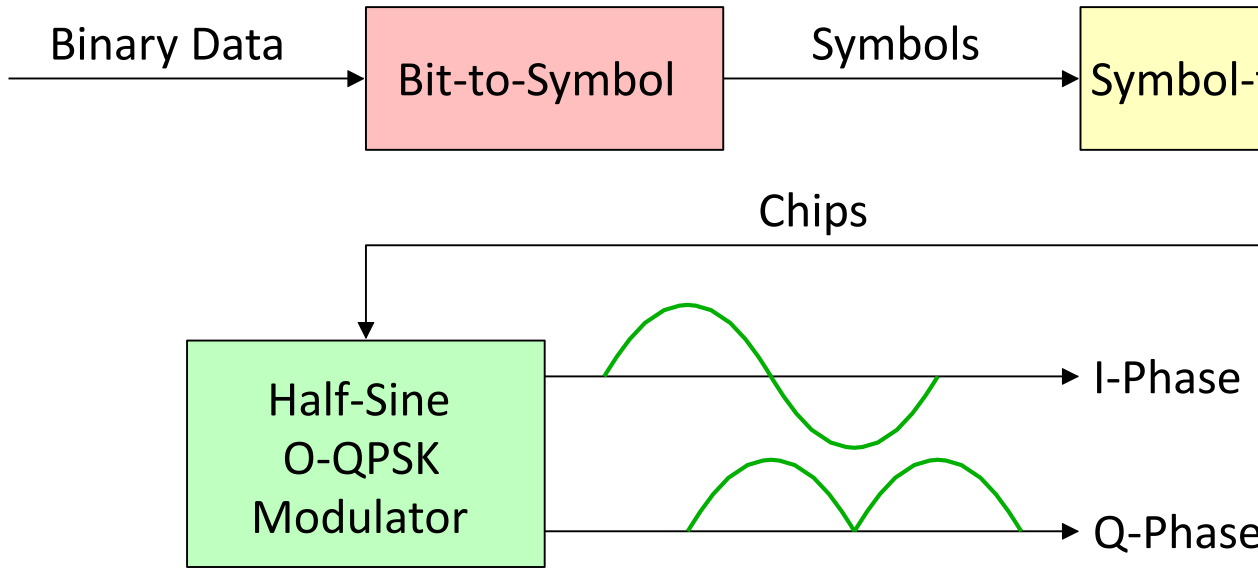
| Symbol | Chip sequence ($C_0, C_1, C_2, \ldots, C_{31}$) |
|---|---|
| 0 | 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 |
| 1 | 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 |
| 2 | 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 |
| 3 | 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 |
| 4 | 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 |
| 5 | 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 |
| 6 | 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 |
| 7 | 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 |
| 8 | 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 |
| 9 | 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 |
| 10 | 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 |
| 11 | 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 |
| 12 | 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 |
| 13 | 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 |
| 14 | 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 |
| 15 | 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 |




- PN sequences introduce randomization and redundancy

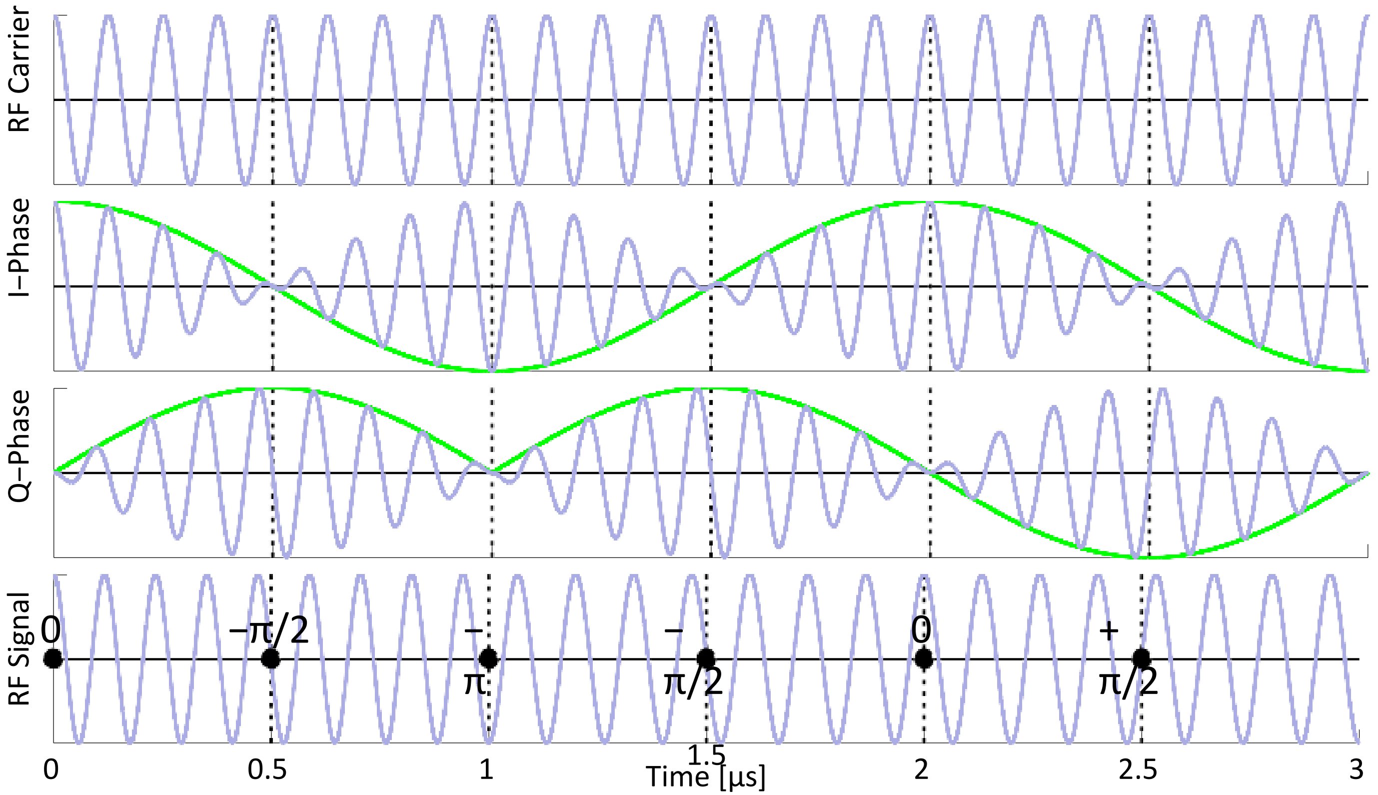
# IEEE 802.15.4 Modulation Scheme

- IEEE 802.15.4: standard for 2,450 MHz wireless radios
- A 3-step process converts binary data to a baseband signal




- In-phase and quadrature-phase components of the baseband signal determine the phase of the transmitted RF signal

# Half-Sine O-QPSK Modulation: Example

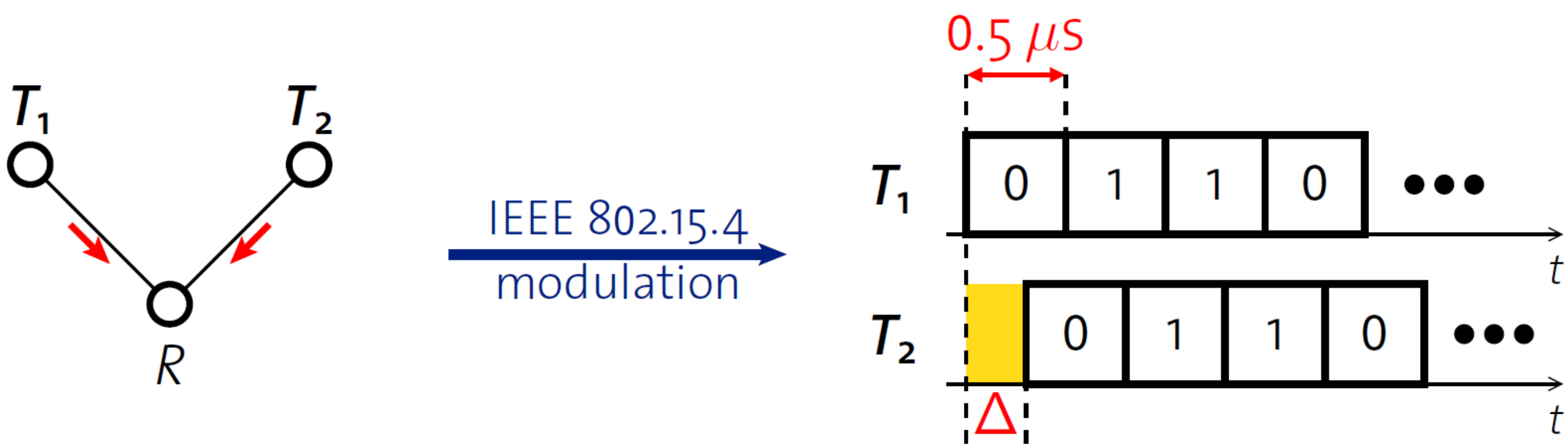



- Data rate: $1/T_C$ chip/s = 2 Mchip/s = 62.5 ksymbol/s = 250 kbps
- The information carried by each chip generates a complete phase change of the RF signal every 0.5 µs

# Synchronous Transmissions

- Multiple nodes transmit **same packet** at **same time**

Ferrari, F. and Zimmerling, M. and Thiele, L. and Saukh, O. (2011). Efficient Network Flooding and Time Synchronization with Glossy. In *10th International Conference on Information Processing in Sensor Networks (IPSN 2011)* (pp. 73–84).

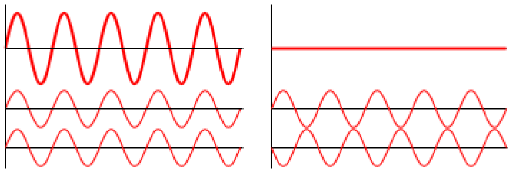


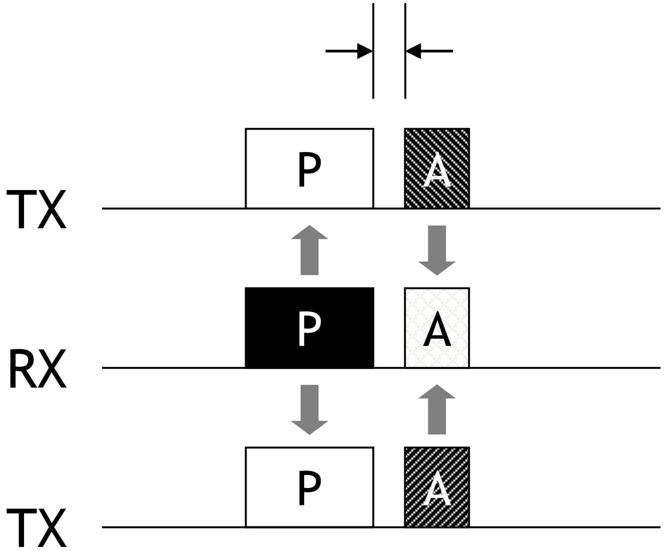

- R receives packet with high probability if $\Delta \leq 0.5$ μs

- Property exploited also in A-MAC [Dutta et al., SenSys '10]

# Synchronized Transmission with Backcast

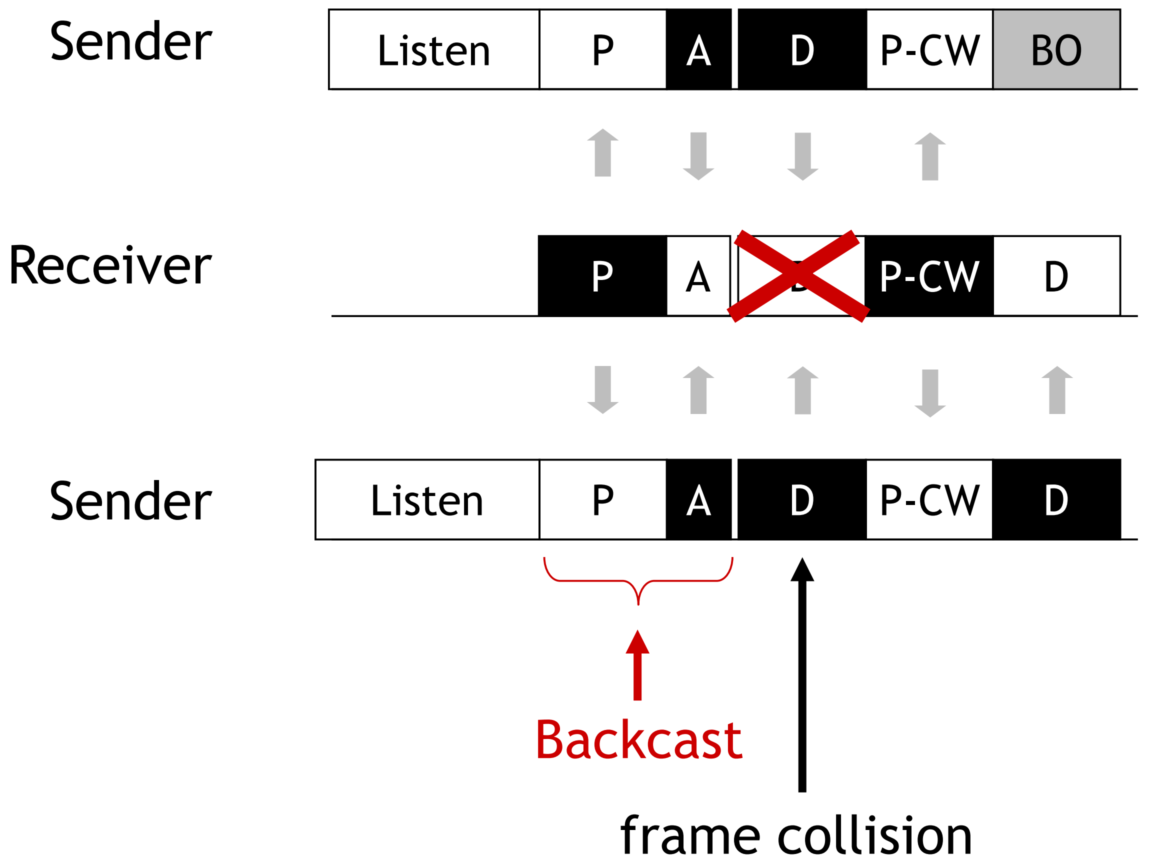
- A link-layer frame exchange in which:
  - A single radio PROBE frame transmission
  - Triggers zero or more *identical* ACK frames
  - Transmitted with tight timing tolerance
  - So there is minimum inter-symbol interference
  - And ACKs collide non-destructively at the receiver






P. Dutta, R. Musaloiu-E., I. Stoica, A. Terzis, "Wireless ACK Collisions Not Considered Harmful", HotNets-VII, October, 2008, Alberta, BC, Canada

# A-MAC's Contention Mechanism



Networked Embedded Systems

# ALL-TO-ALL NETWORK FLOODING: GLOSSY

# Increasing Reliability: Glossy Floods

- Main objectives
  - Fast and reliable flooding of messages
  - Accurate global time synchronization
  - Hide complexity of multi-hop networks

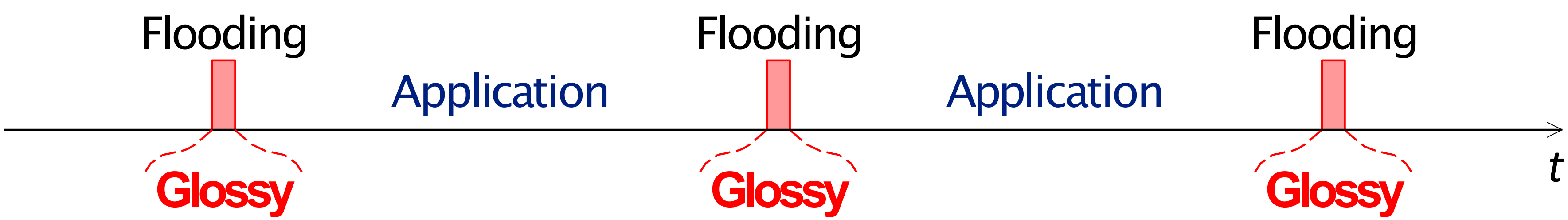
- Challenge in multi-hop wireless networks
  - Uncoordinated transmissions, packet loss, retransmission delays

- Glossy: Flooding architecture for wireless sensor networks
  - Fastest possible propagation, by design
  - Highly reliable (> 99.99 %)
  - Requires no network state information
  - Efficient also in dense networks
  - Time synchronization at no additional cost

Ferrari, F. and Zimmerling, M. and Thiele, L. and Saukh, O. (2011). Efficient Network Flooding and Time Synchronization with Glossy. In *10th International Conference on Information Processing in Sensor Networks (IPSN 2011)* (pp. 73–84).

# Glossy: Key Techniques

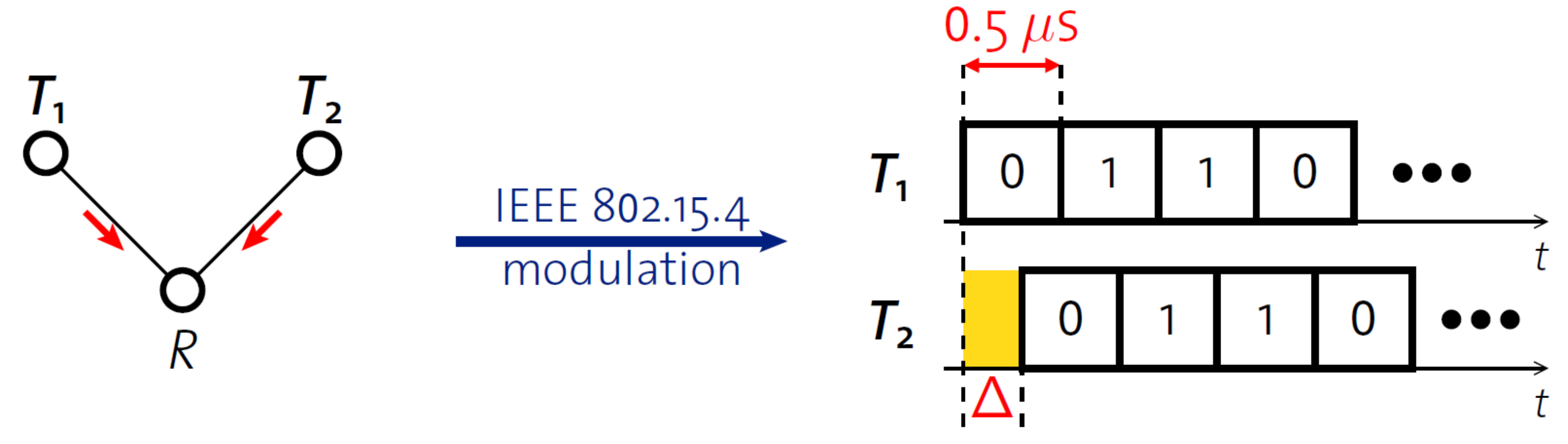
- **Temporally decouple** network flooding from application tasks




- Exploit **synchronous transmissions** for fast network flooding

# Synchronous Transmissions

- Multiple nodes transmit **same packet** at **same time**




- R receives packet with high probability if Δ ≤ 0.5 μs

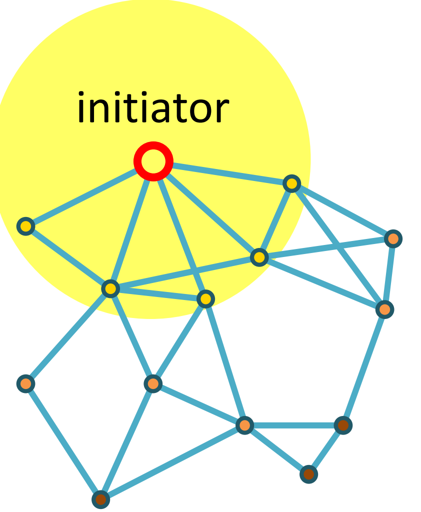
- Property exploited also in A-MAC [Dutta et al., SenSys '10]

# Challenges for Efficient Flooding

How to relay packets **efficiently** and **reliably**?

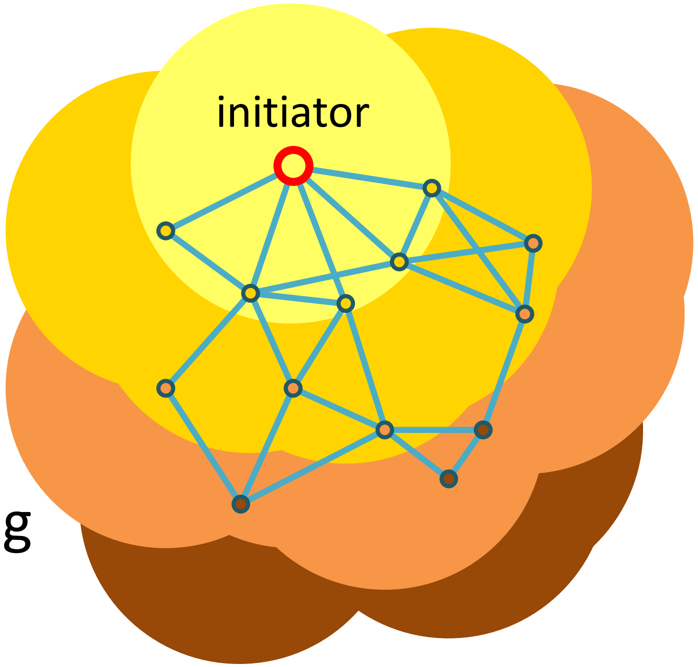
- Avoid aggressive, uncoordinated broadcasts

- Typical approach:
  Coordinate packet transmissions
  - CF [Zhu et al., NSDI 2010]
  - RBP [Stann et al., SenSys 2006]
  - Maintain topology-dependent state



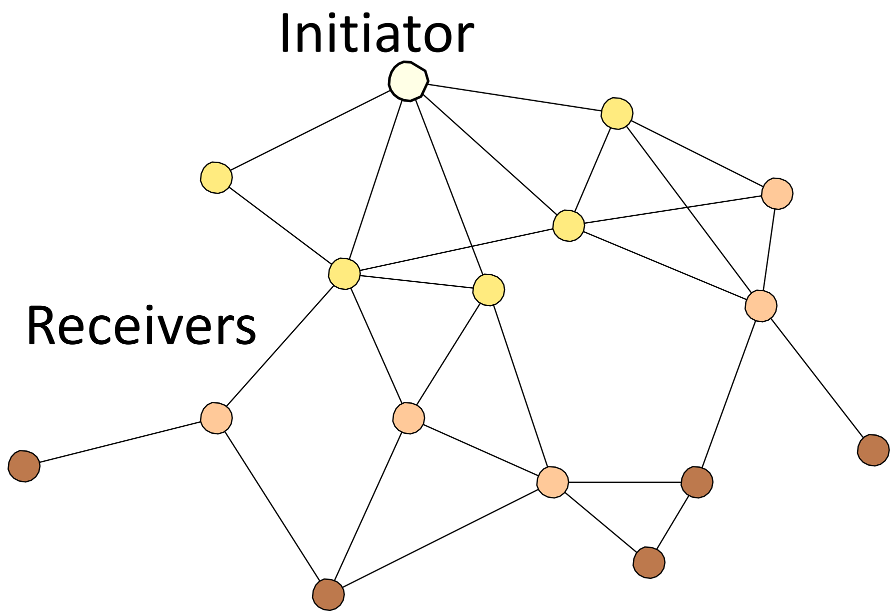
# Glossy Flooding Architecture

- **All** receiving nodes relay packets **synchronously**
  - Simple, but radically different solution
  - No explicit routing
  - No topology-dependent state

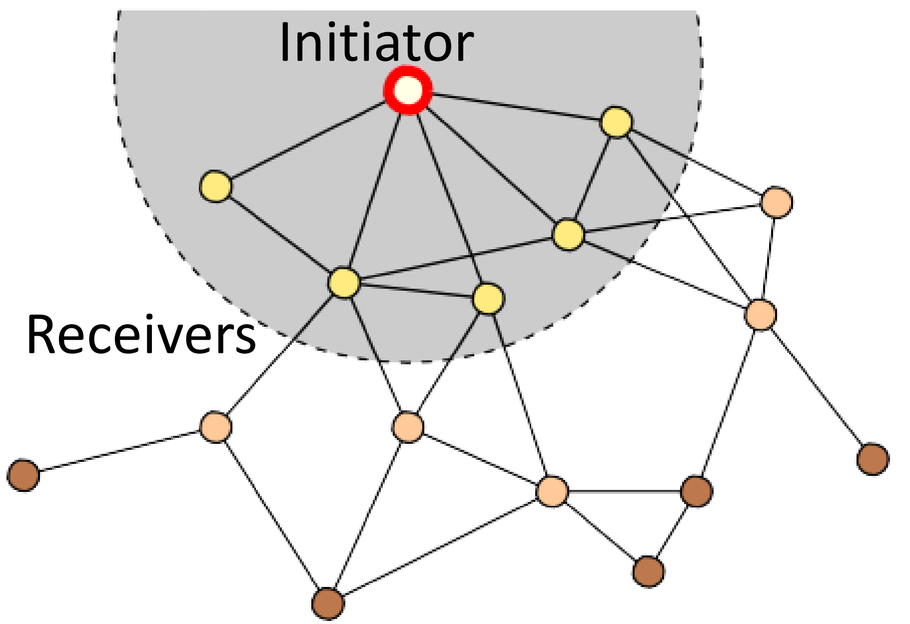



- Key Glossy mechanisms
  - Start execution at the same time
  - Compensate for hardware variations
  - Ensure deterministic execution timing

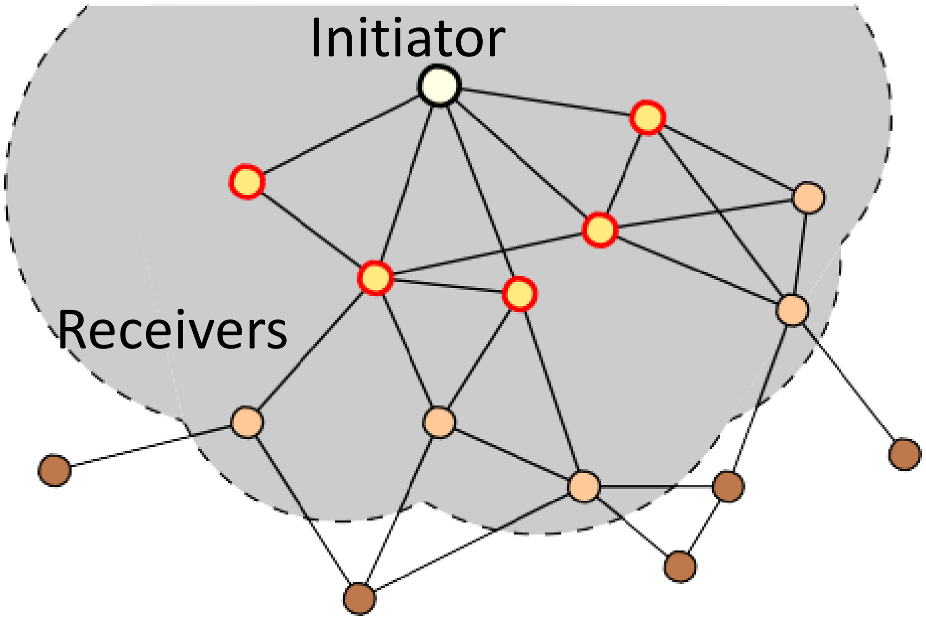
# Glossy Example Flood Propagation



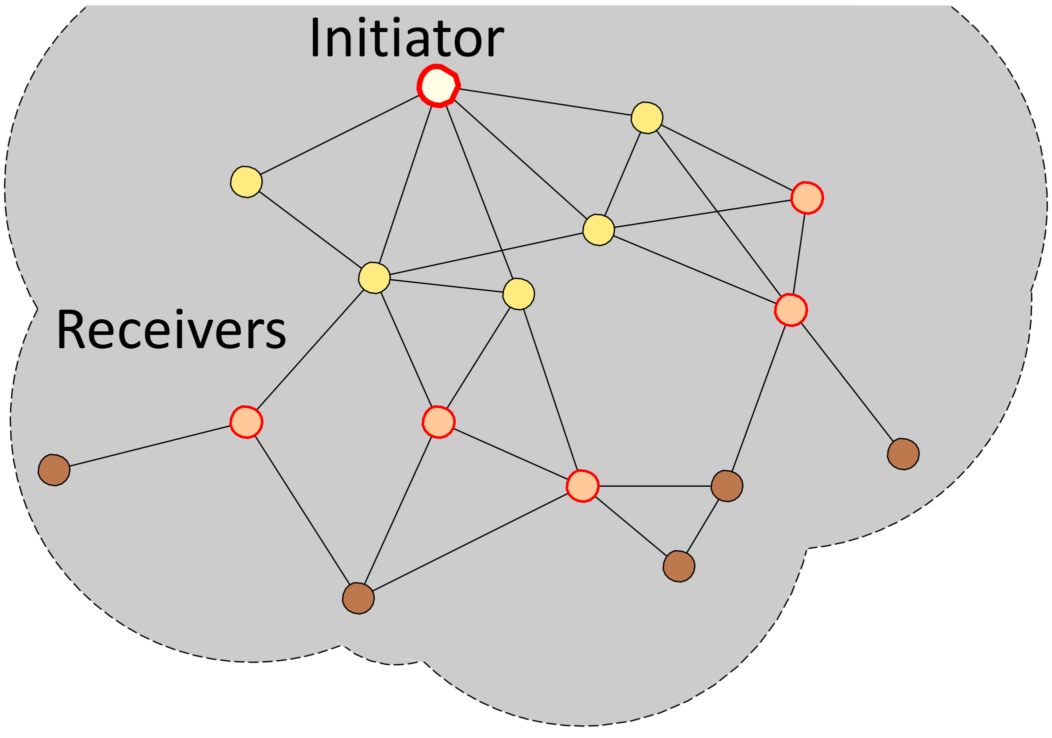
# Glossy Example Flood Propagation



# Glossy Example Flood Propagation



# Glossy Example Flood Propagation

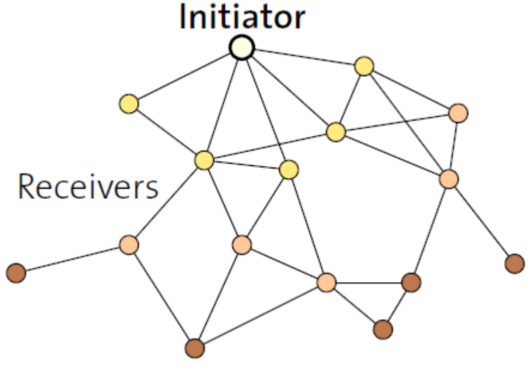


# Glossy Fast Packet Propagation Details

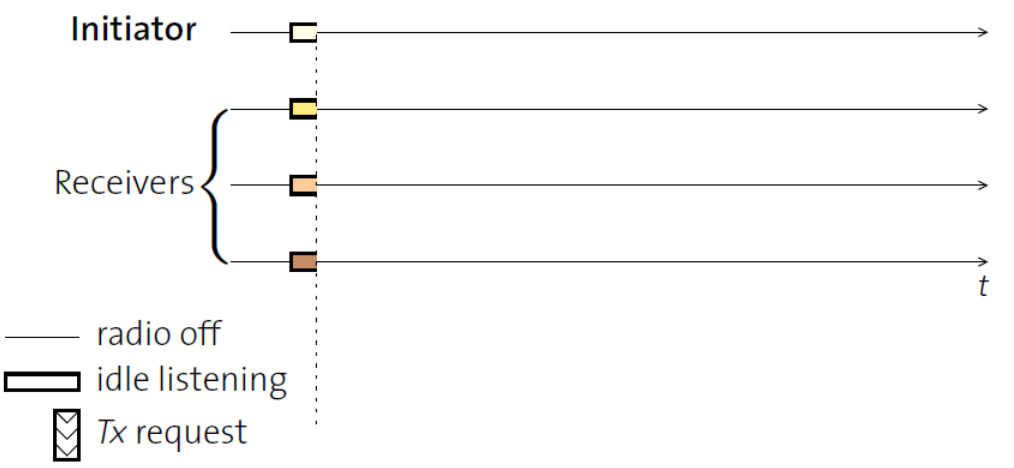
When Glossy starts:

- Turn on radio

## Example




## Timeline

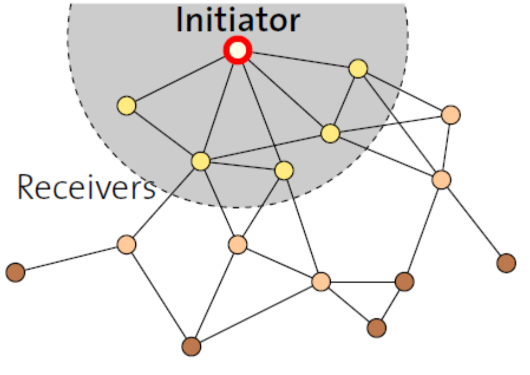


# Glossy Fast Packet Propagation Details

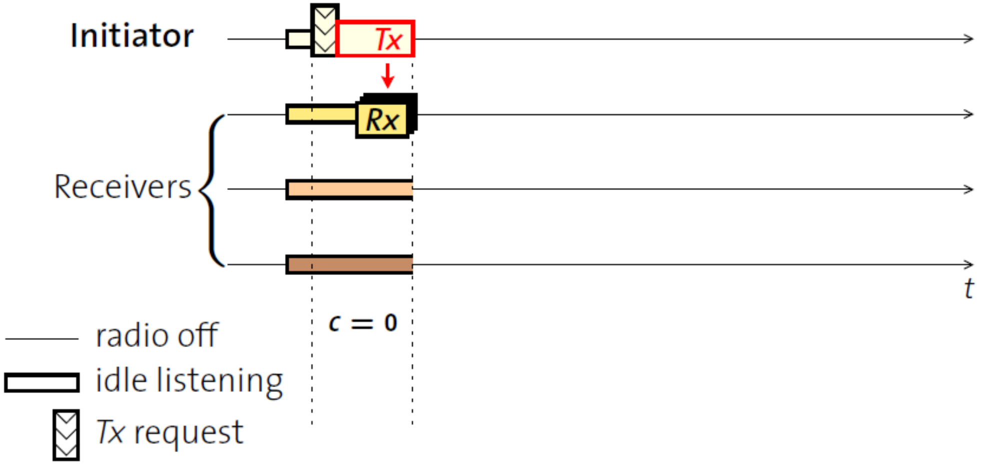
Initiator:

- Set relay counter $c = 0$
- Transmit packet

## Example




## Timeline

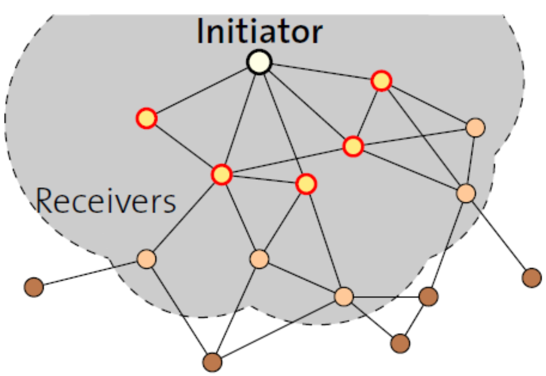


# Glossy Fast Packet Propagation Details

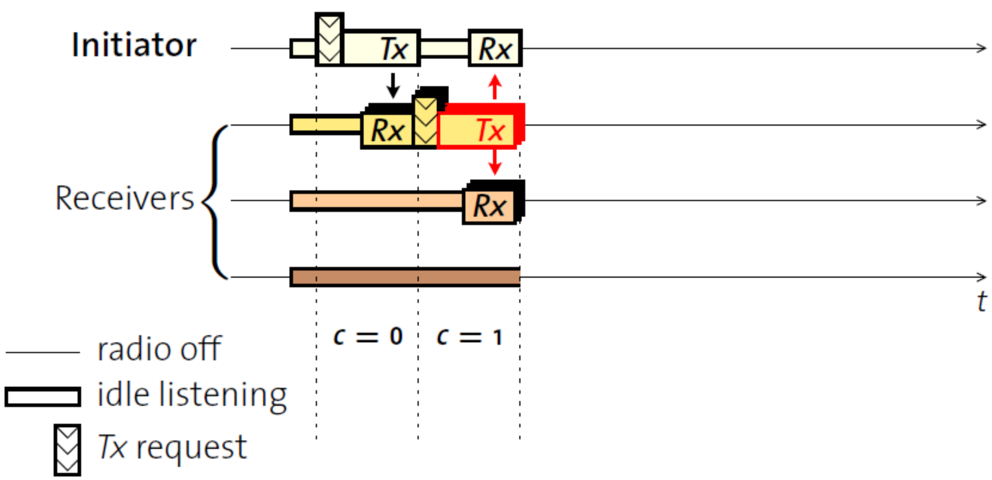
At packet reception:

- Increment relay counter *c*
- Transmit synchronously

## Example




## Timeline

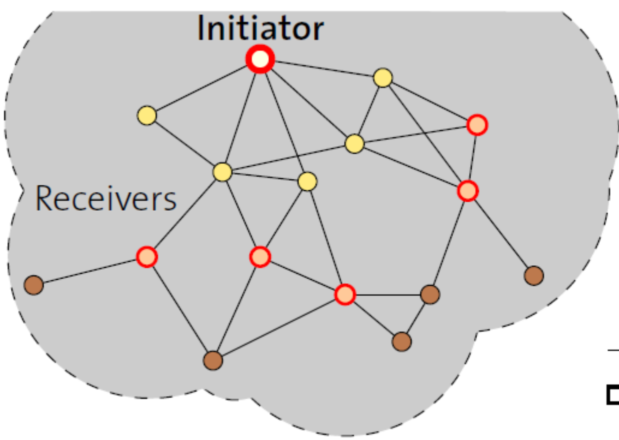


# Glossy Fast Packet Propagation Details

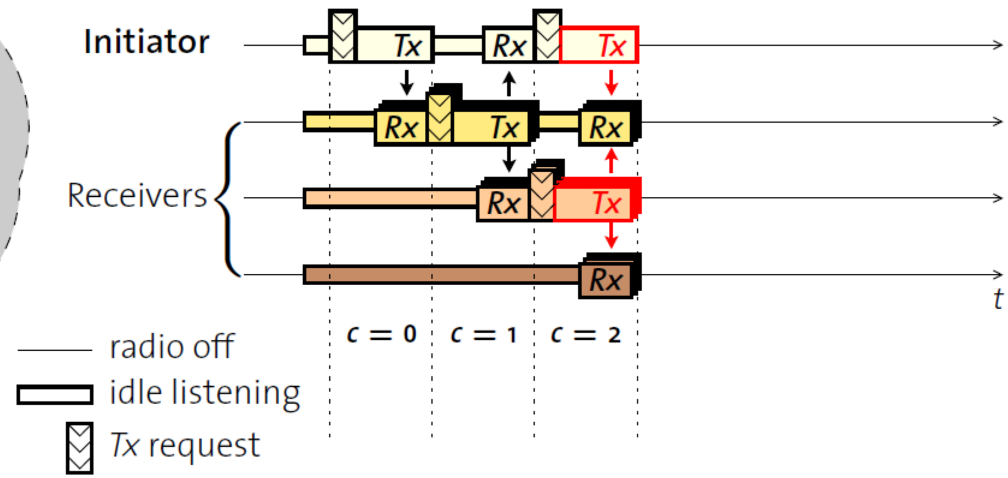
At packet reception:

- Increment relay counter $c$
- Transmit synchronously

## Example




## Timeline

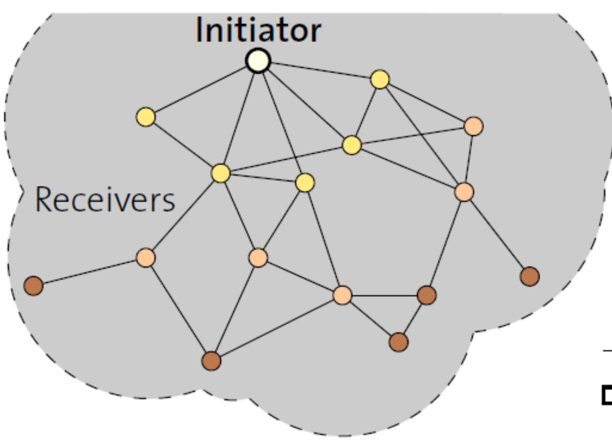


# Glossy Fast Packet Propagation Details

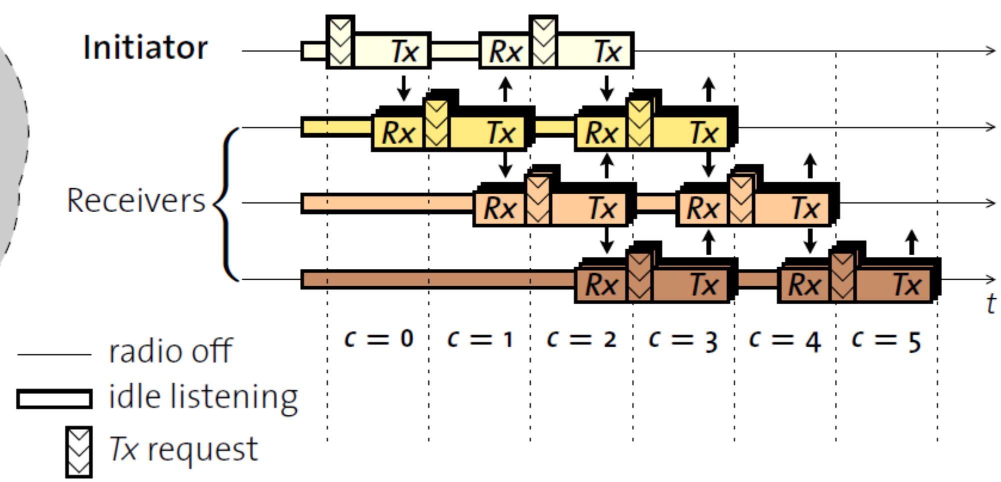
Stop and turn off radio when:

- Already transmitted *N* times

## Example



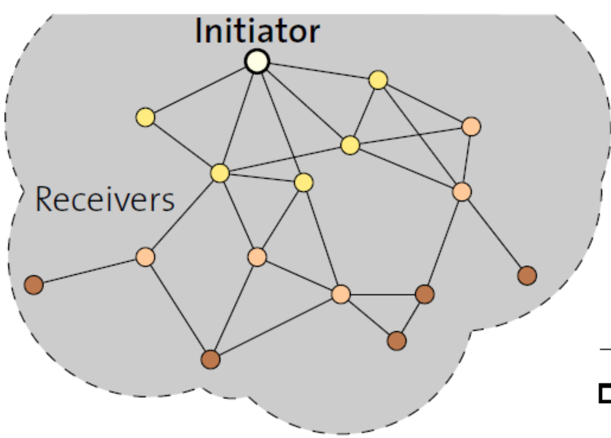

## Timeline ($N = 2$)



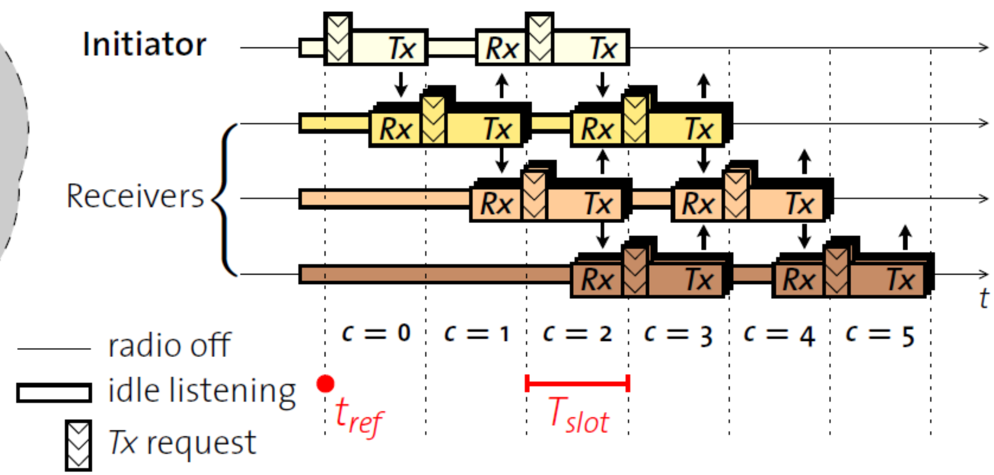
# Glossy Fast Packet Propagation Details

- $T_{slot}$ is constant by design
- Local estimates of $T_{slot}$ / Received relay counter $c$ } Reference time $t_{ref}$
- $t_{ref}$ provides synchronized time

**Example**

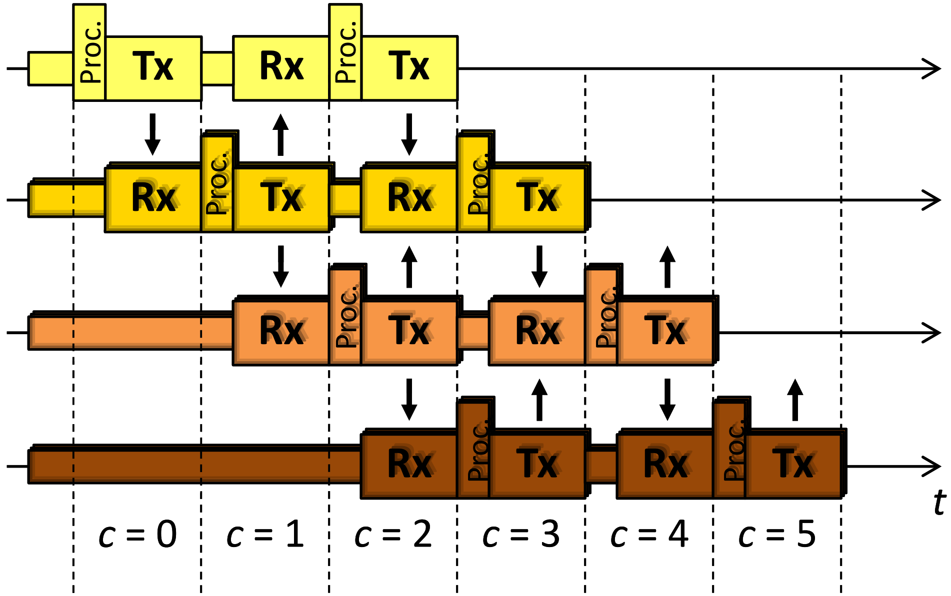



**Timeline** ($N = 2$)

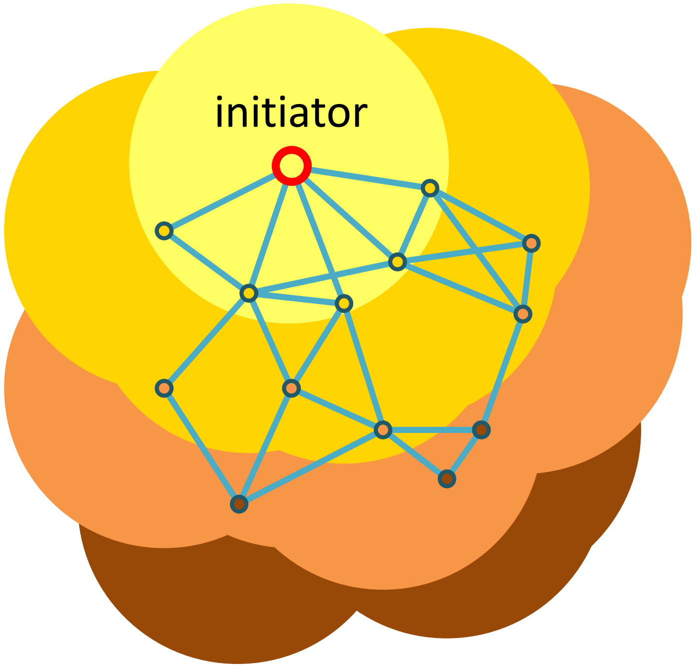


# Propagation in Glossy

- A **relay counter** *c* is set to 0 at the first transmission
- A node increments *c* before relaying the packet

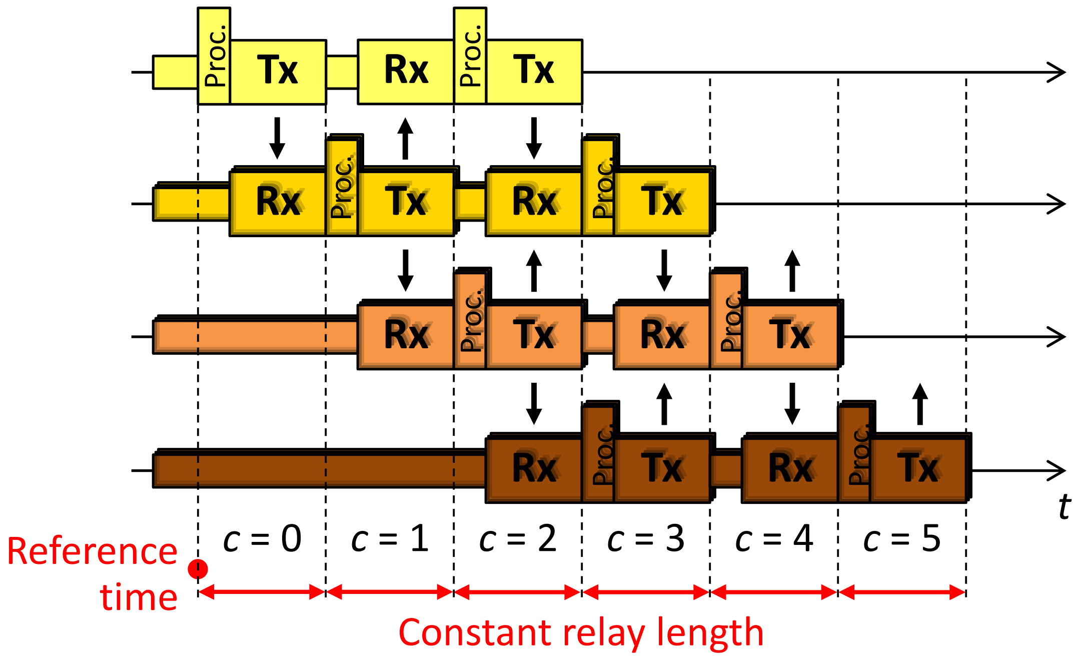


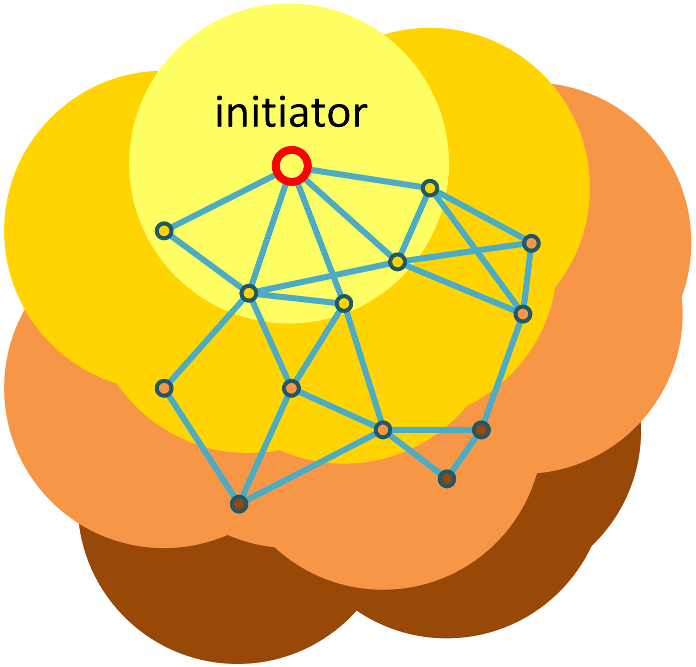

*(In this example a node transmits at most twice)*



# Time synchronization in Glossy

- Estimate the **relay length** during propagation
- Compute a common **reference time**



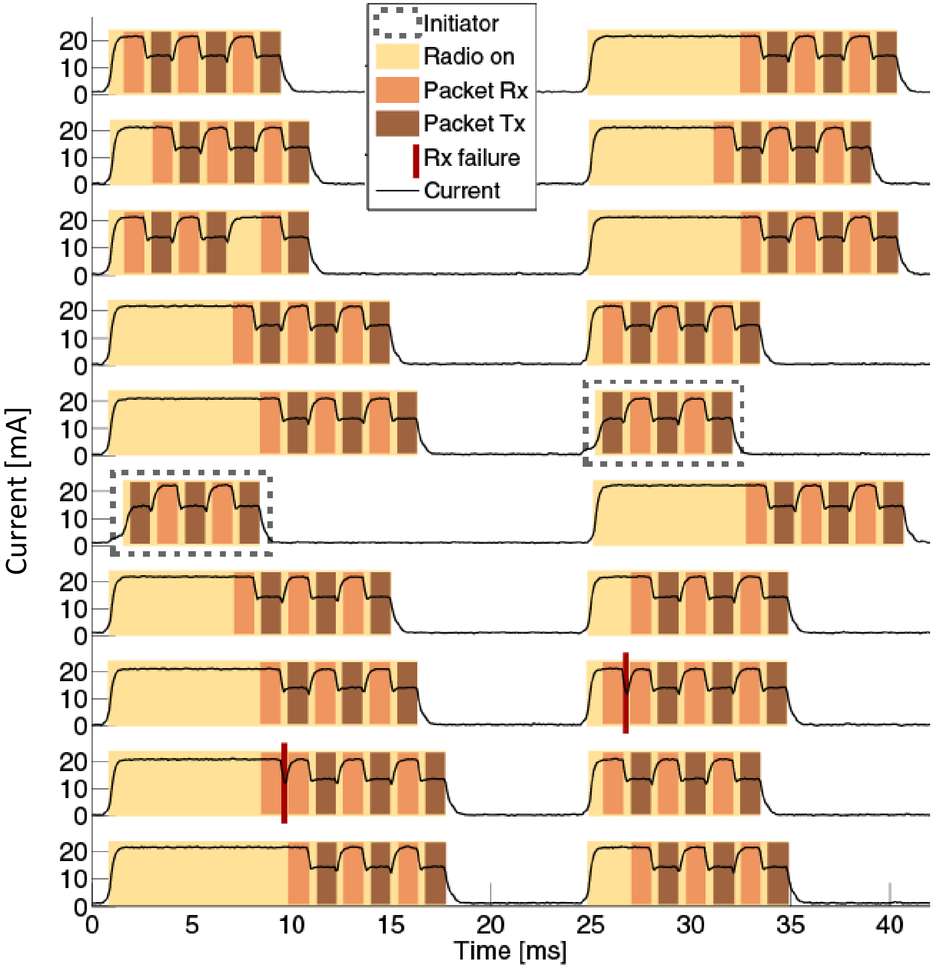



# Glossy: Main Evaluation Findings

- **A few ms** to flood packets to hundreds of nodes
- Reliability **> 99.99 %** in most scenarios
- Synchronization error **< 1 µs** even after 8 hops

# Evaluation of Glossy on FlockLab

- Multi-modal monitoring at network scale
- Flooding protocol (Glossy)
  - Packet transmissions overlap
- Power
  - Find current consumption for each state
  - Expected behavior?
- Activity
  - Packet exchange



# Today's Hot Researcher & Paper

- David Culler
  - Faculty at UC Berkeley
- (Distributed) systems background
  - Many well-known systems implementations
  - Founder of TinyOS initiative
  - Drove first large-scale WSN applications (habitat monitoring)
- Now focusing on sustainable energy use (buildings)

J Hill, R Szewczyk, A Woo, S Hollar, D Culler, K Pister: *System architecture directions for networked sensors.* ACM SIGOPS operating systems review 34 (5), 93-104

# Recap of Today

- Networked Embedded Systems focus on cross-layer solutions
  - No strict division across interfaces (like OSI model)

- (Temporal) Co-ordination helps a lot
- Most protocols employ a mix of stochastic elements (contention) and schedule based elements

- State-of-the-Art protocols allow reliable communication at very little energy cost