Freie Universität Berlin

Embedded Internet and the Internet of Things WS 12/13

4. MAC Protocols

Prof. Dr. Mesut Güneş
Distributed, embedded Systems (DES)
Institute of Computer Science
Freie Universität Berlin

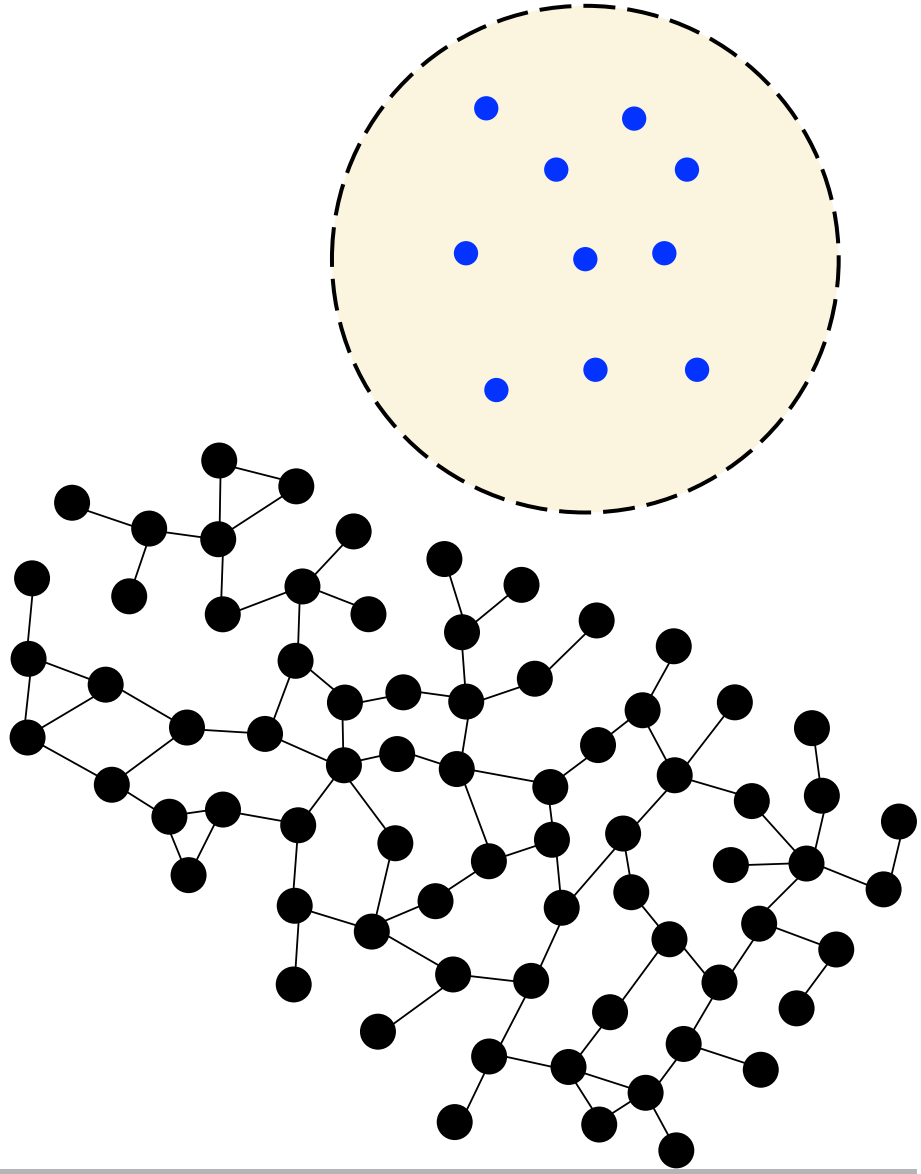
Overview

- Challenges for MAC protocols
- Contention based protocols
- Reservation based protocols
- Hybrid protocols

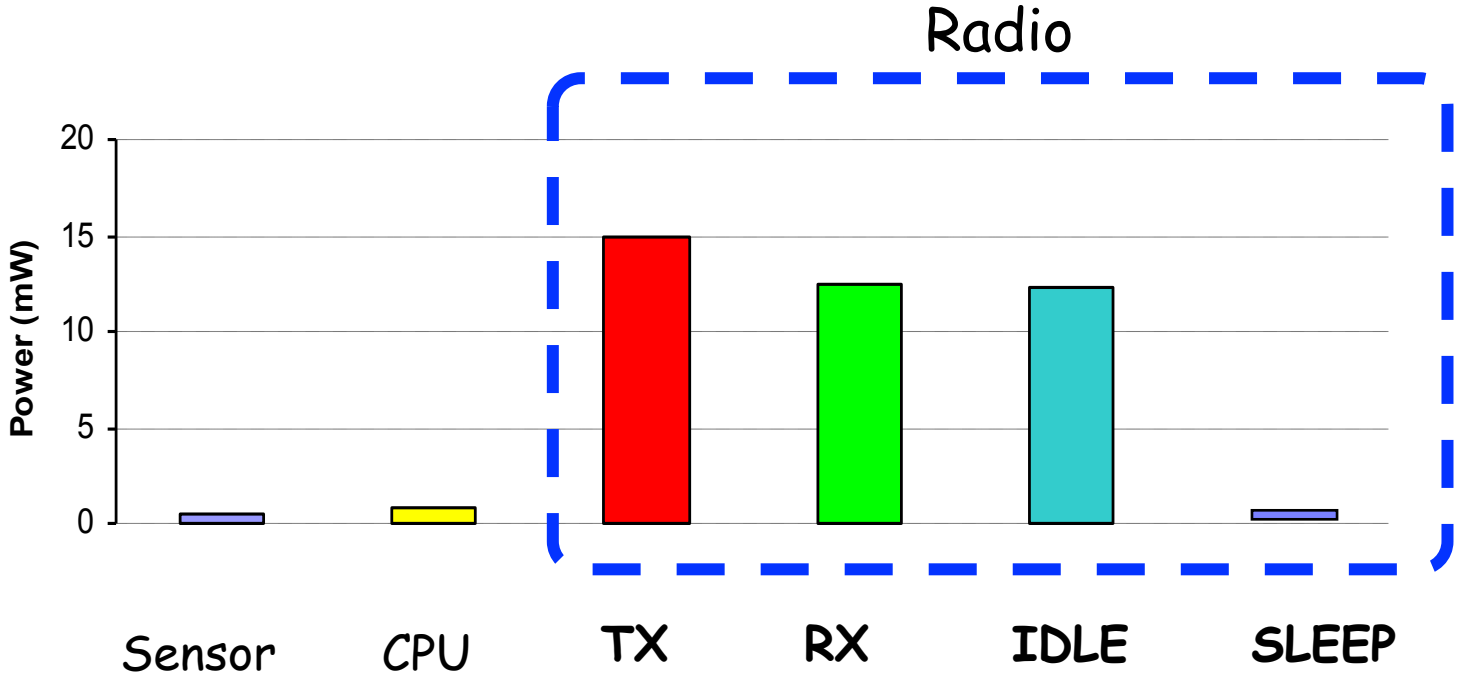
Challenges for MAC protocols

Objectives of MAC protocols

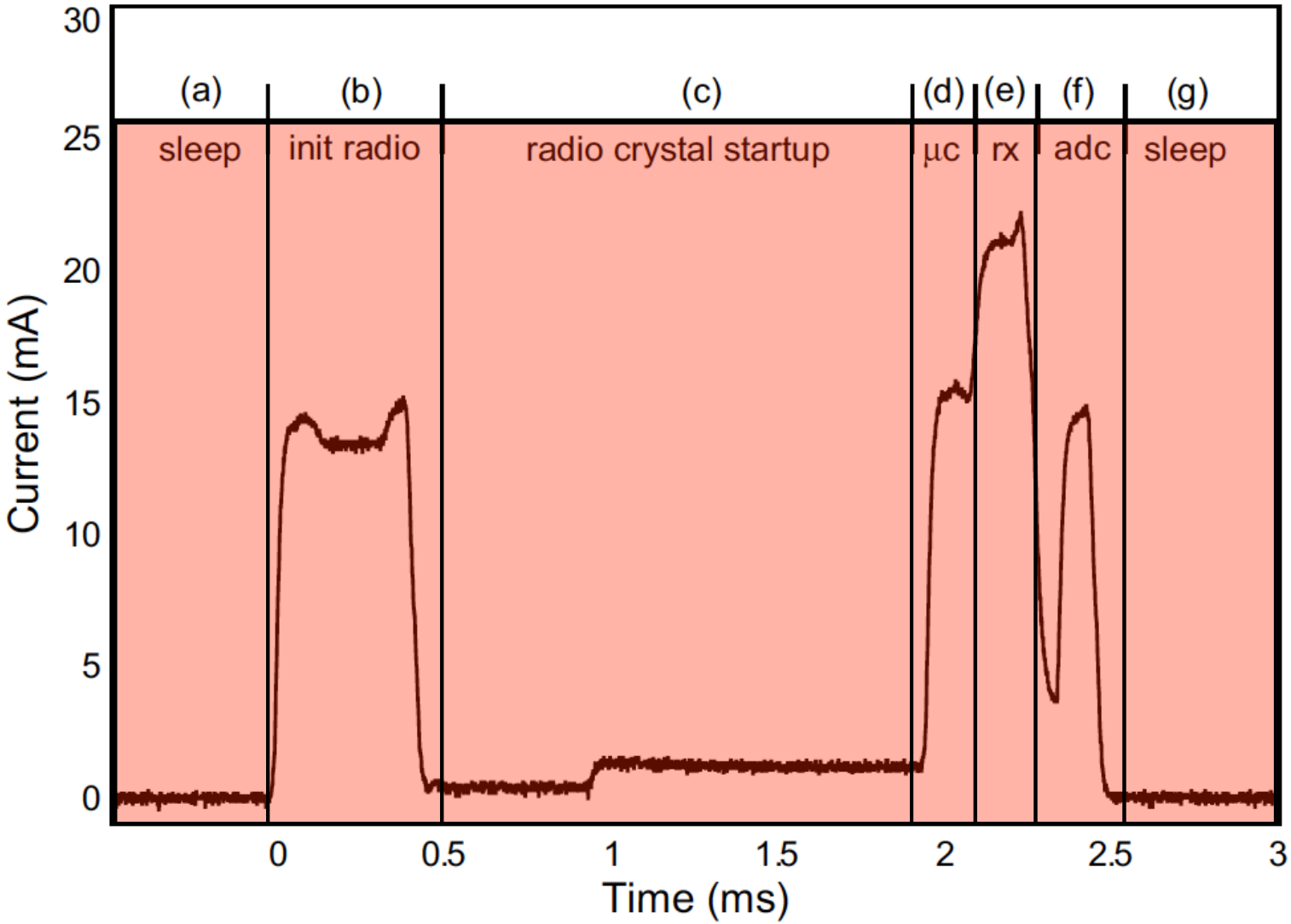
- Collision avoidance
- Energy efficiency
- Scalability
- Latency
- Fairness
- Throughput
- Bandwidth utilization



Power consumption



Power consumption



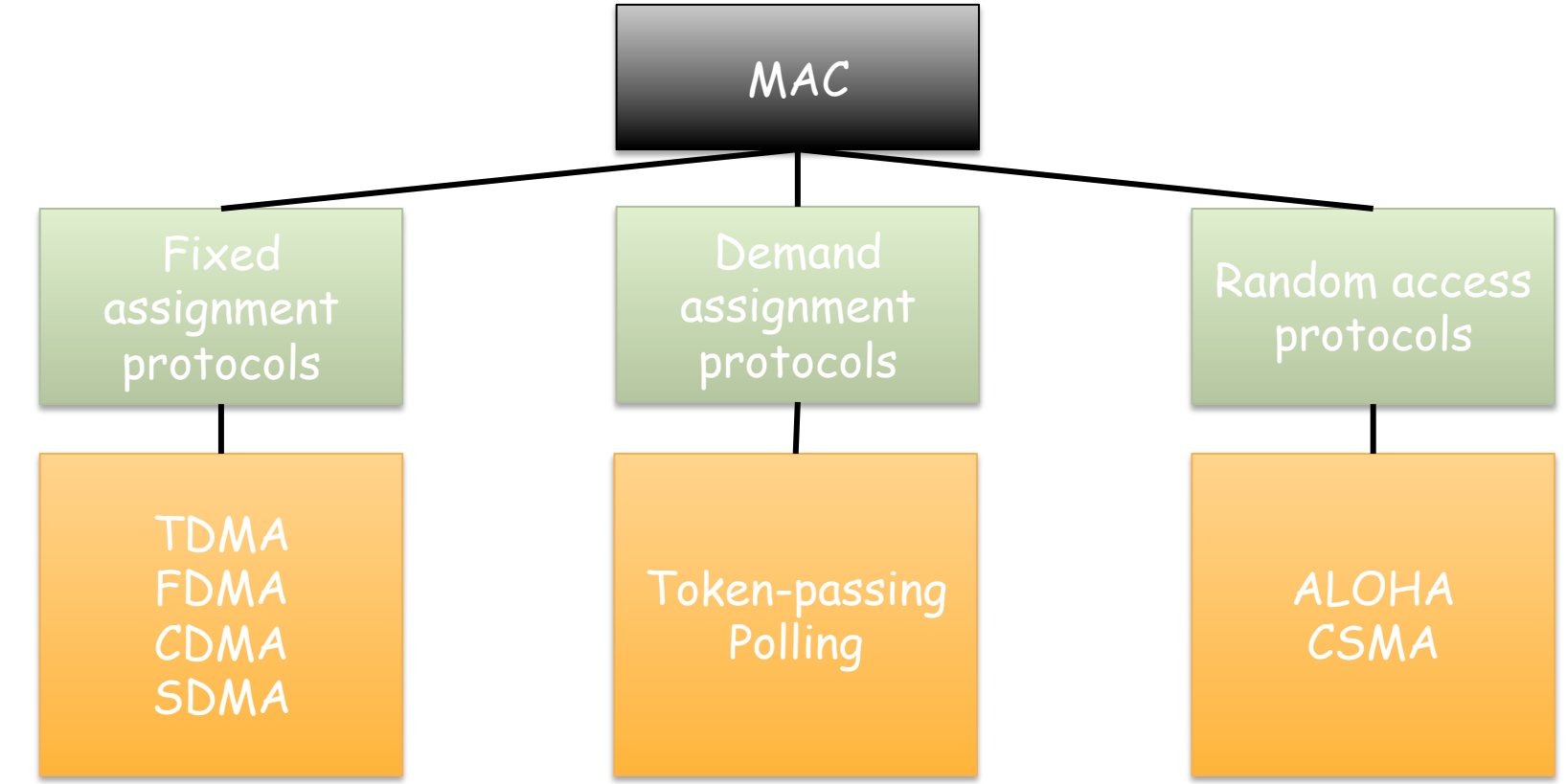
Major sources of energy waste

- Idle listening
  - Transmitter
  - Receiver

  } Common to all wireless networks

- Objective: Reduce energy consumption!

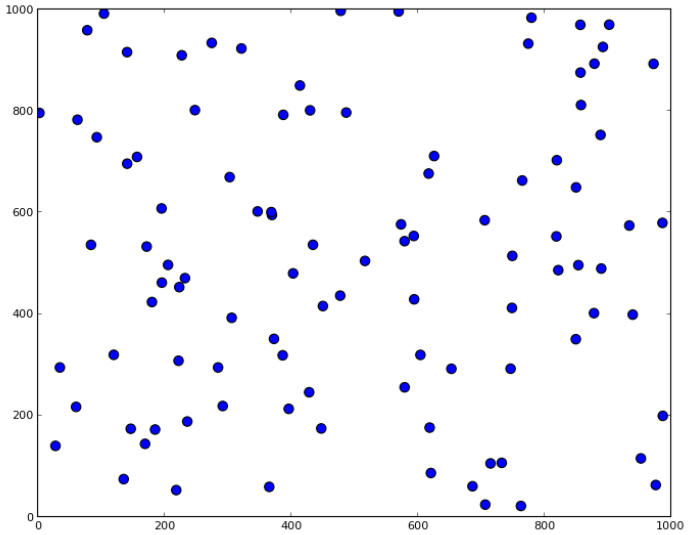
MAC protocol classes



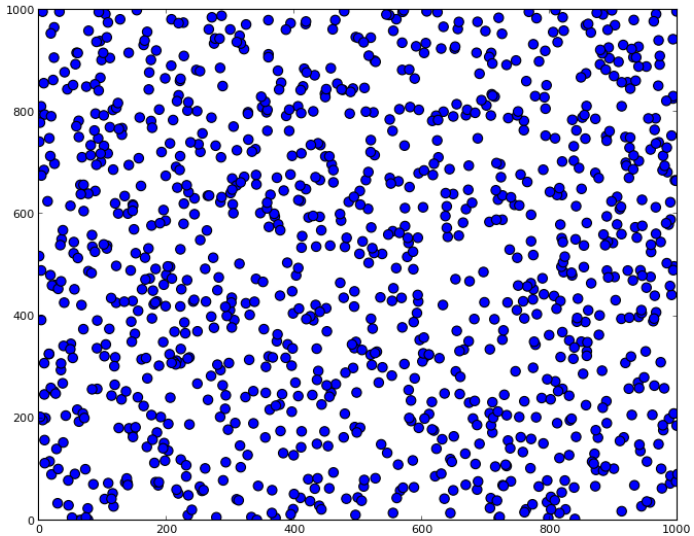
Challenges for MAC in WSNs

1. WSN Architecture

- High density of nodes
- Increased collision probability
- Minimize signaling overhead to prevent further collisions
- Sophisticated and simple collision avoidance protocols required



100 nodes on 1000mx1000m

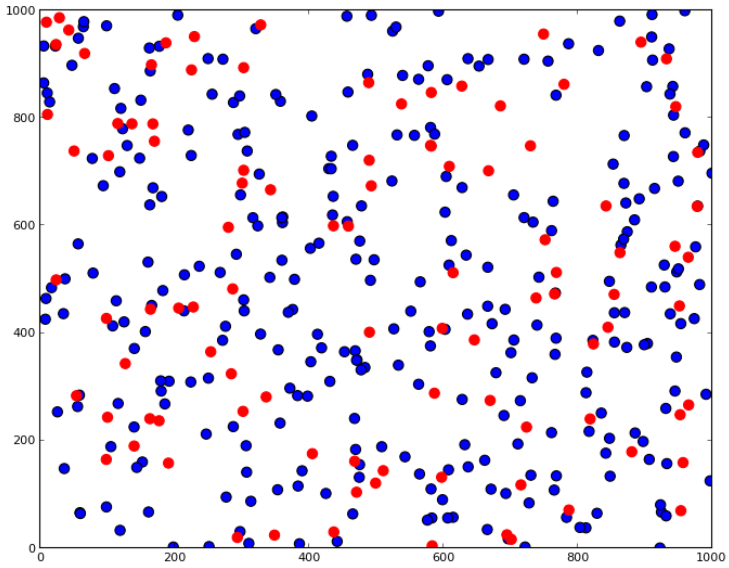


1000 nodes on 1000mx1000m

Challenges for MAC in WSNs

2. Limited energy resources

- Connectivity and the performance of the network is affected as nodes die
- Transmitting and receiving consumes almost same energy
- Frequent power up/down eats up energy
- Need very low power MAC protocols
- Minimize signaling overhead
- Avoid idle listening
- Prevent frequent radio state changes (active<->sleep)



Challenges for MAC in WSNs

3. Limited processing and memory capabilities
   - Complex algorithms cannot be implemented
   - Conventional layered architecture may not be appropriate
   - Centralized or local management is limited
   - Simple scheduling algorithms required
   - Cross-layer optimization required
   - Self-configurable, distributed protocols required

Challenges for MAC in WSNs

4. Limited packet size

- Unique node ID is not practical -> IoT?
- Limited header space
- Local IDs should be used for inter-node communication
- MAC protocol overhead should be minimized

5. Cheap encoder/decoder

- Cheap node requirement prevents sophisticated encoders/ decoders to be implemented
- Simple FEC codes required for error control
- Channel state dependent MAC can be used to decrease error rate

Challenges for MAC in WSNs

6. Inaccurate clock crystals
   - Cheap node requirement prevents expensive crystals to be implemented
   - Synchronization problems
   - TDMA-based schemes are not practical

7. Event-based networking
   - Observed data depends on physical phenomenon
   - Spatial and temporal correlation in the physical phenomenon should be exploited

Bottom line:

Existing MAC protocols cannot be used for WSNs!

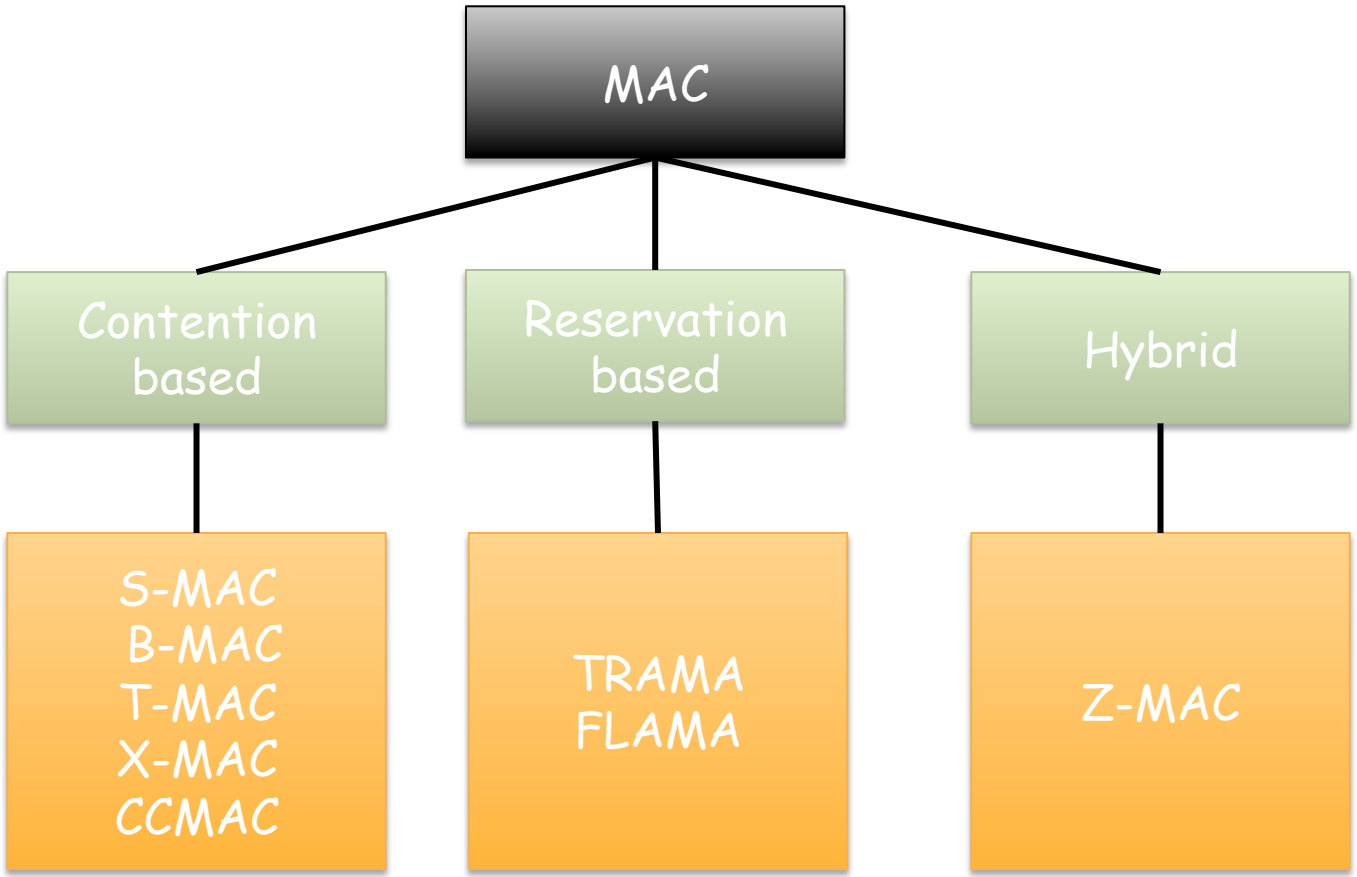
MAC Protocols for WSN

http://www.st.ewi.tudelft.nl/~koen/MACsoup/

- ?-MAC (pick your letter!)

- µ-MAC, AI-LMAC, B-MAC, Bit, BMA, CC-MAC, CMAC, Crankshaft, CSMA-MPS, CSMA/ARC, DMAC, E2-MAC, EMACs, f-MAC, FLAMA, Funneling-MAC, G-MAC, HMAC, LMAC, LPL,MMAC, MR-MAC, nanoMAC, O-MAC, PACT, PEDAMACS, PicoRadio, PMAC, PMAC, Q-MAC, Q-MAC, QMAC, RATE EST, RL-MAC, RMAC, RMAC, S-MAC, S-MAC/AL, SCP-MAC, SEESAW, Sift, SMACS, SS-TDMA, STEM, T-MAC, TA-MAC, TICER, TRAMA, U-MAC, WiseMAC, X-MAC, Z-MAC, …

Overview of MAC protocols for WSNs

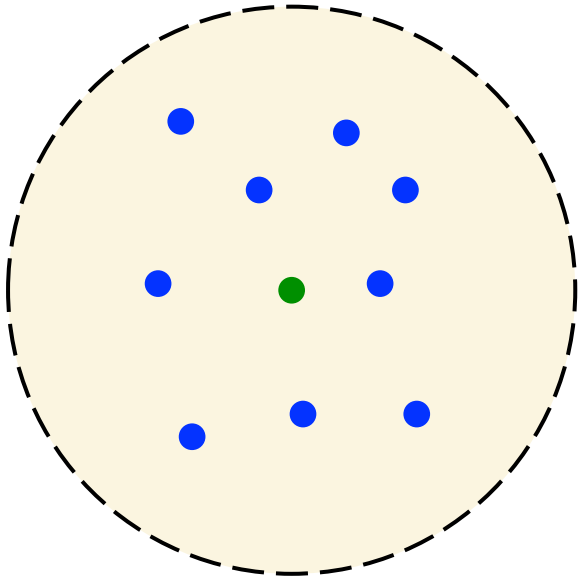


Contention based protocols

Contention based MAC Protocols

- In fact random based MAC protocol
- Channel access through carrier sense mechanism
- Provide robustness and scalability to the network
- Collision probability increases with node density

  -> Good performance with few nodes, bad performance with many nodes

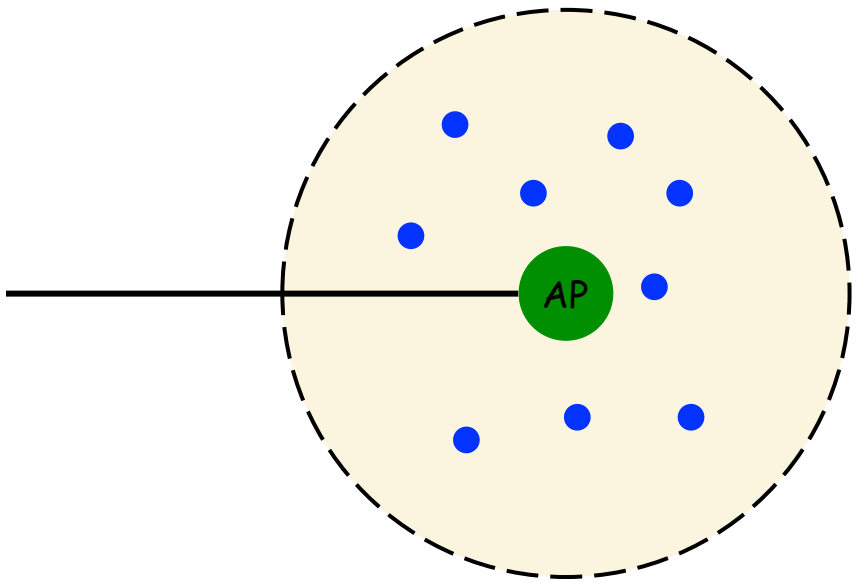


Contention based protocols

CSMA/CA

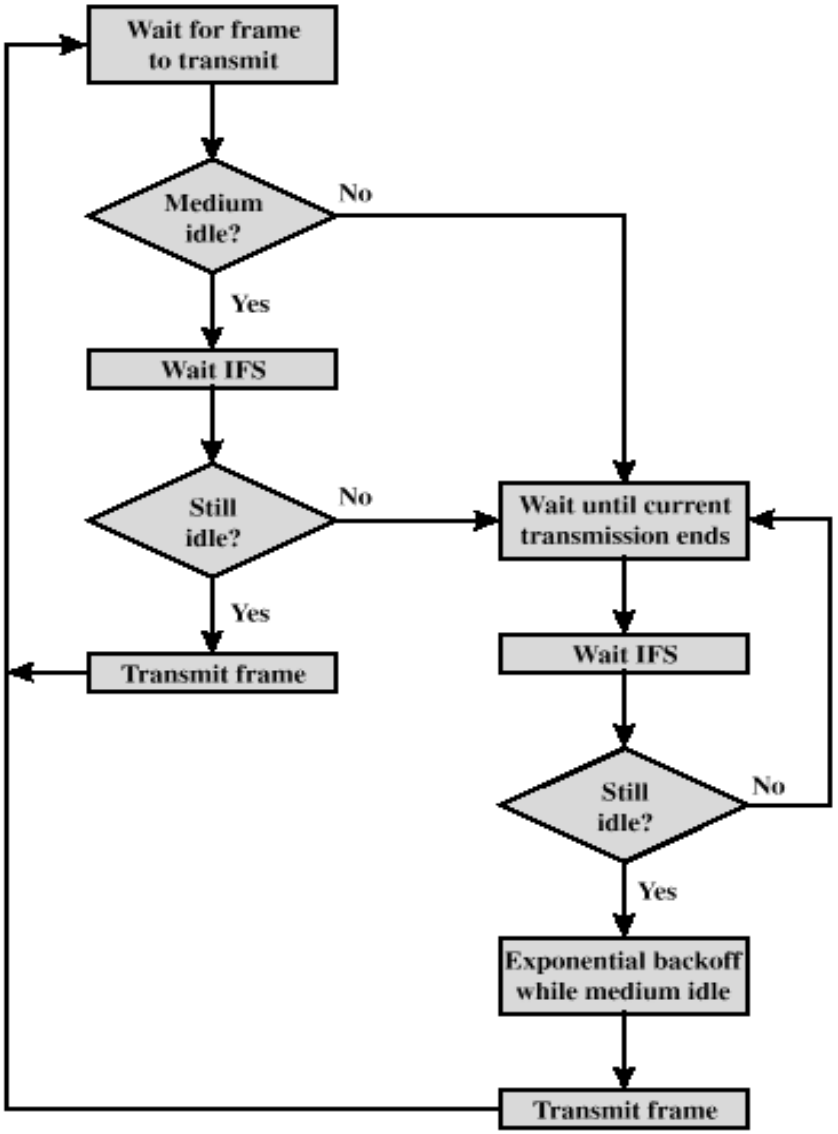
IEEE 802.11

- IEEE 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999
  - Originally developed for WLANs

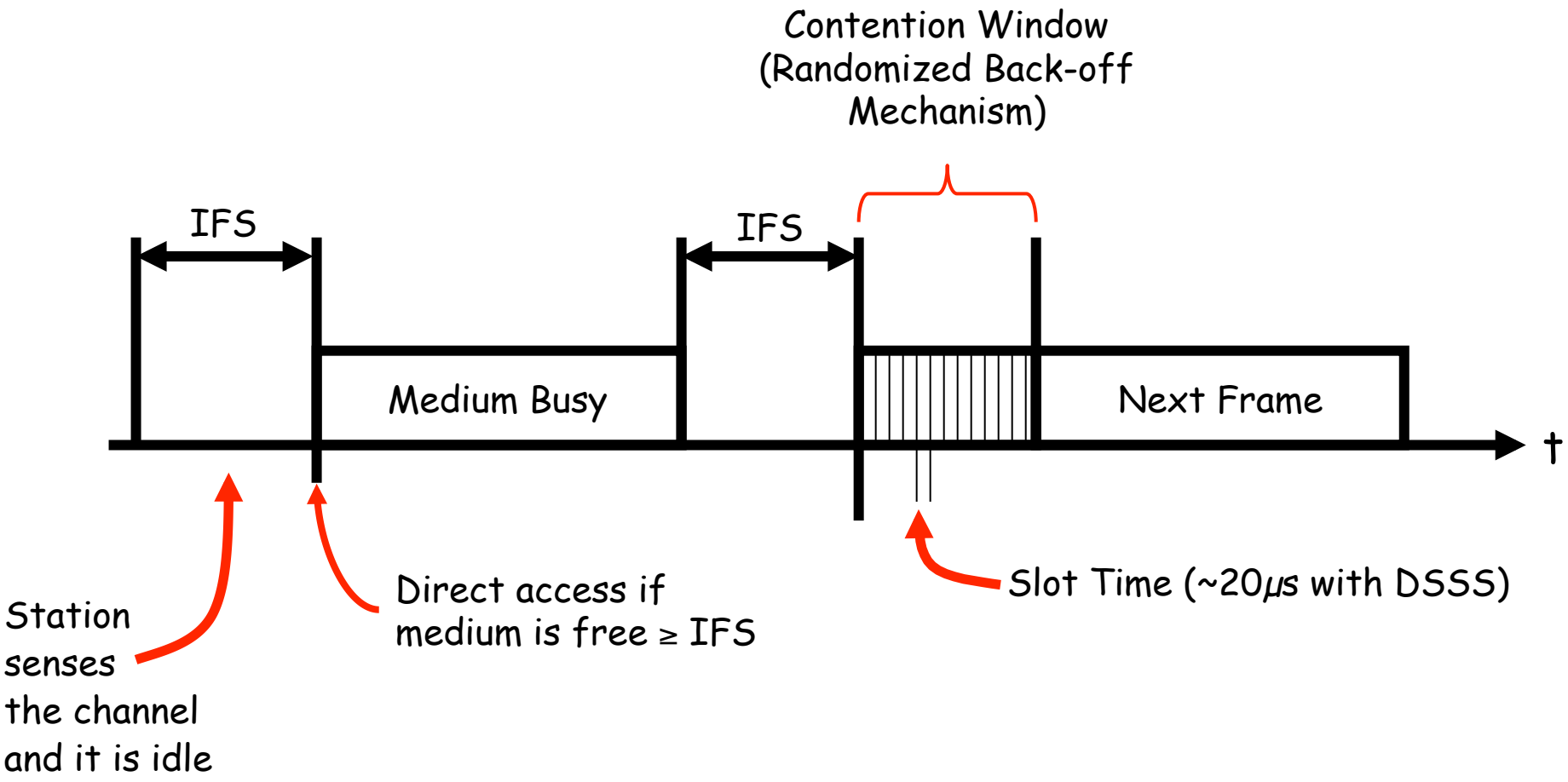


IEEE 802.11

- Reminder basic knowledge CSMA/CA (flowchart)
- Distributed Foundation Wireless Medium Access Control - Distributed Coordinated Function CSMA/CA (DFWMAC-DCF) for IEEE 802.11



Basic CSMA/CA



IFS = Inter Frame Spacing

Basic CSMA/CA

- A station with a frame to transmit senses the medium (channel).
- If IDLE -> waits to see if the channel remains idle for a time equal to IFS (Inter-frame spacing). If so, the station may transmit immediately.
- If BUSY -> (either because the station initially finds the channel busy or because the channel becomes busy during the IFS idle time), the station defers transmission and continues to monitor the channel until the current transmission is over.
- Once the current transmission is over, the station delays another IFS.
- If the medium remains idle for this period, the station backs off using a binary exponential backoff scheme and again keeps sensing the medium.
- Backoff scheme
  - The station picks up a random number of slots (the initial value of backoff counter) within a contention window to wait before transmitting its frame.

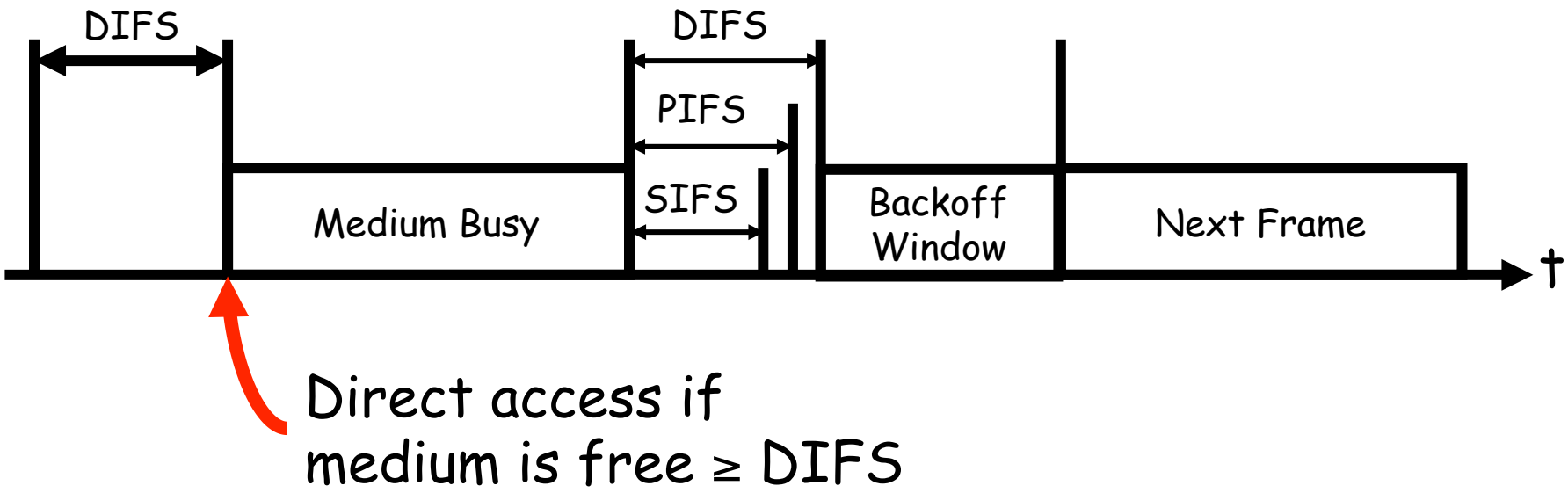
Basic CSMA/CA

- MAC runs a random number generator to set a backoff clock for every contending station
- When the contention window starts in which all stations having packets for transmission run down their backoff clocks
- The station with its clock expiring first starts transmission
- Other terminals sense the new transmission and freeze their clocks to be restarted after the completion of the current transmission in the next contention period

CSMA/CA Algorithm

- If Collisions (Control or Data)
  - Binary exponential increase (doubling) of CW
  - Length of backoff time is exponentially increased as the station goes through successive retransmissions

Inter-frame Spaces (IFS)

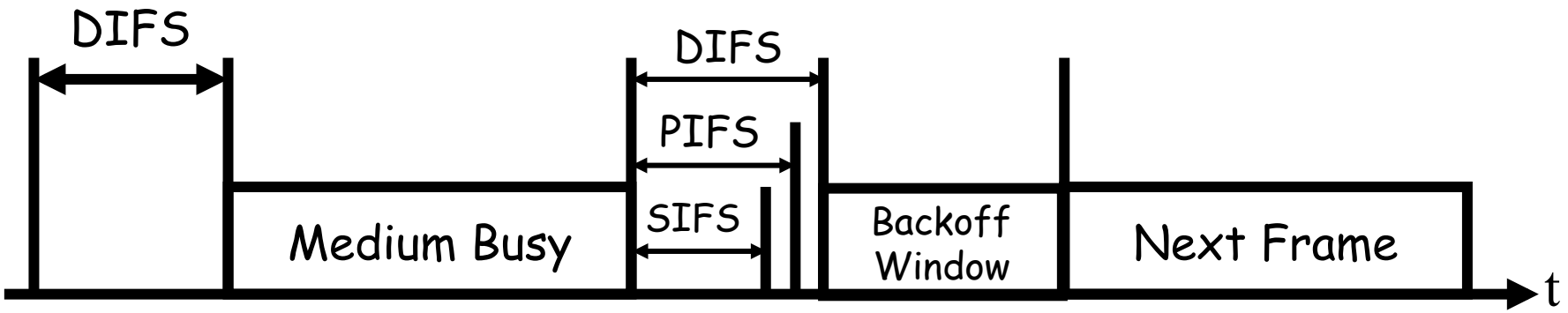


DIFS ~50μs	Distributed Coordination Function IFS
PIFS ~30μs	Point Coordination Function IFS
SIFS ~10μs	Short Inter Frame Spacing

Inter-frame Spaces (IFS)

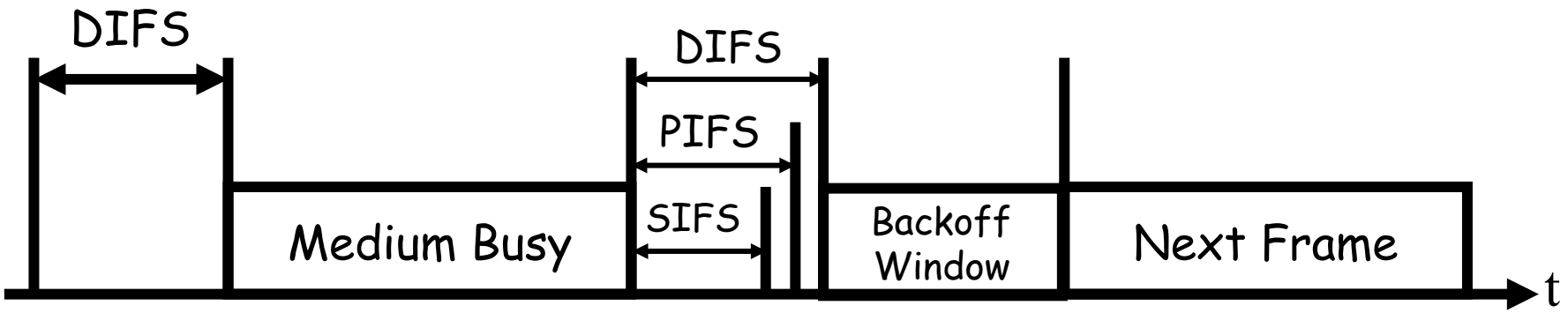
- Priorities are defined through different inter frame spaces

- SIFS (Short Inter Frame Spacing)
  - Highest priority packets such as ACK, CTS, polling response
  - Used for immediate response actions



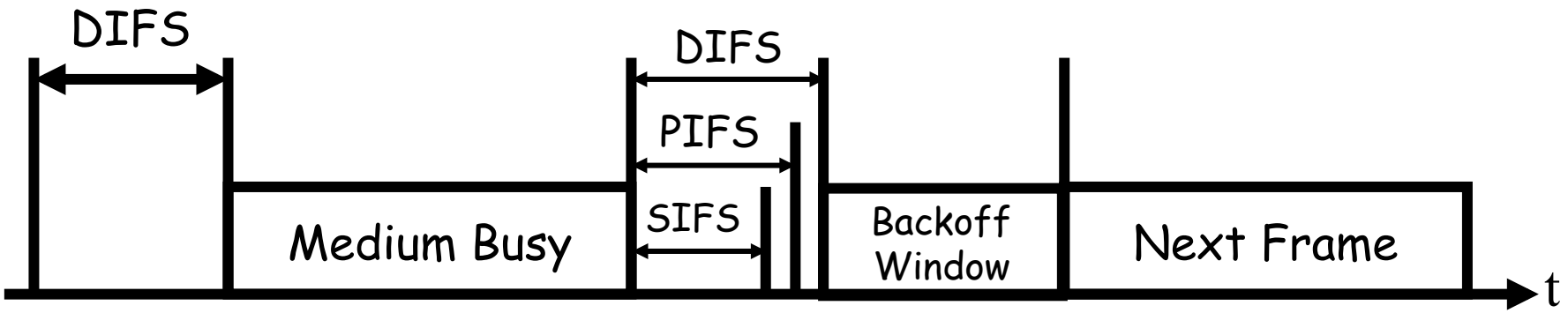
Inter-frame Spaces (IFS)

- PIFS (PCF IFS): Point Coordination Function Inter-Frame spacing
  - Medium priority, for real time service using PCF
  - SIFS + One slot time
  - Used by centralized controller in PCF scheme when using polls



Inter-frame Spaces (IFS)

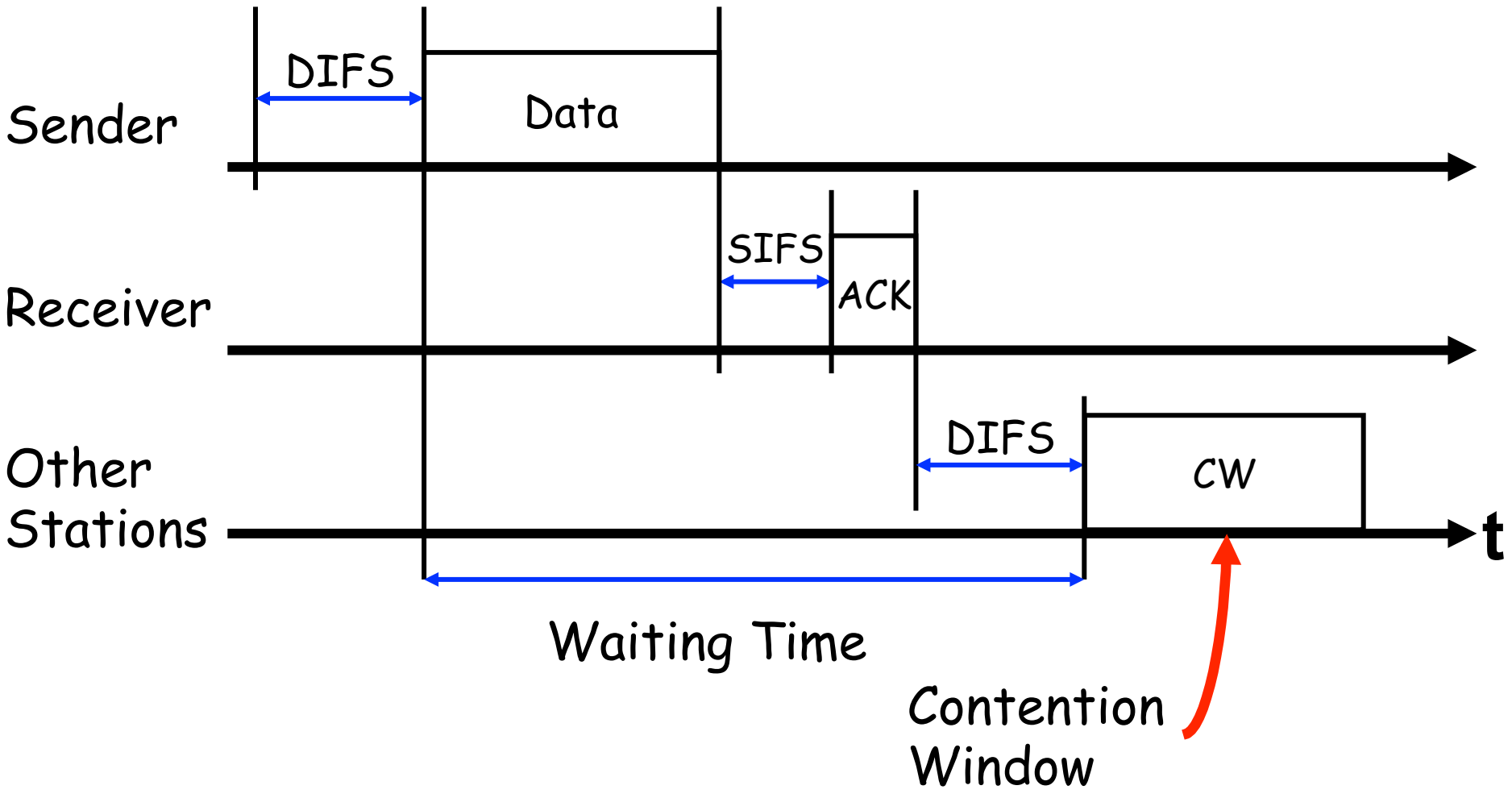
- DIFS (DCF, Distributed Coordination Function IFS)
  - Lowest priority, for asynchronous data service
  - SIFS + Two slot times
  - Used as minimum delay of asynchronous frames contending for access



CSMA/CA with ACK

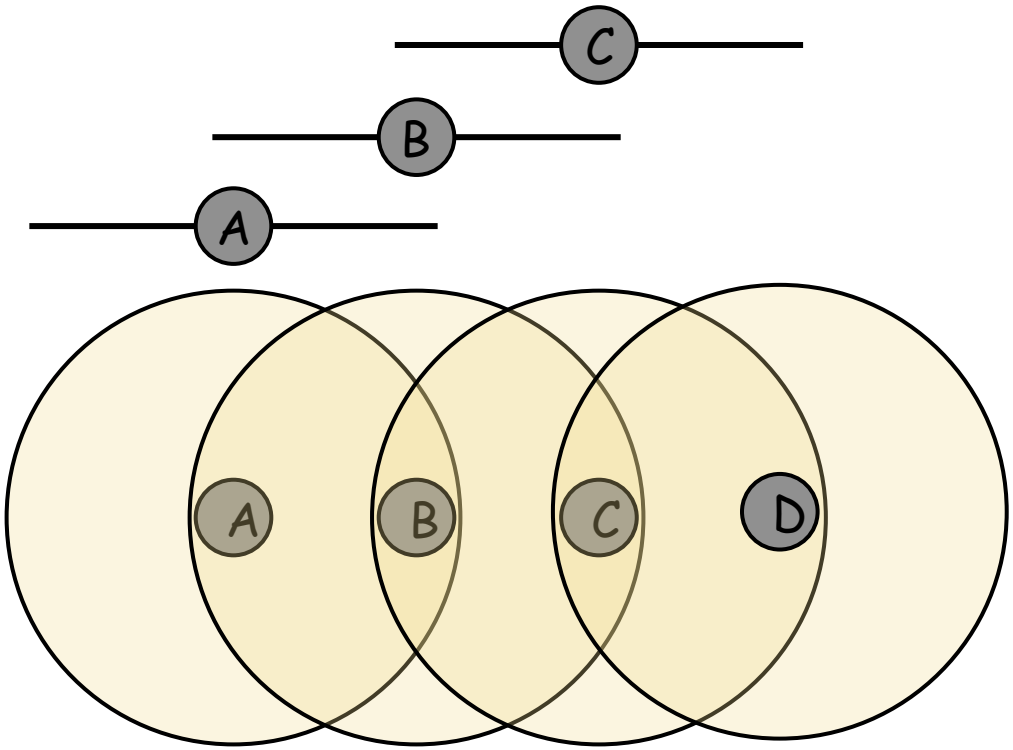
- Station has to wait for DIFS before sending data
- Receiver ACKs immediately
  - Receiver waits for SIFS < DIFS
- Receiver transmits ACK without sensing the medium
- If ACK is lost, retransmission is performed
- Automatic retransmission of data packets in case of transmission errors

CSMA/CA with ACK



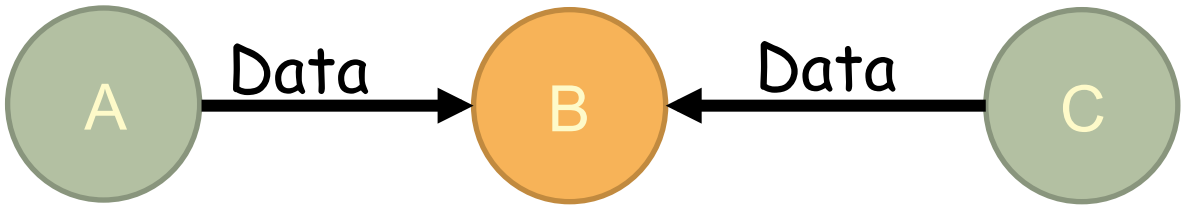
Problems with CSMA/CA

- Hidden terminal problem
- Exposed terminal problem



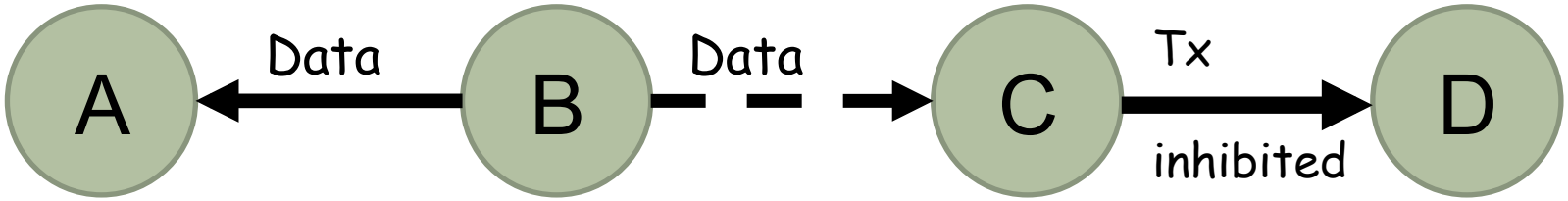
Hidden terminal problem

- A senses the channel free and sends Data
- C cannot hear A and senses the channel free
- Data packet collides at B



Exposed terminal problem

- B sends Data to A (overheard by C)
- C inhibits its transmission to D since channel is busy
- A cannot hear C
  - C-D transmission can actually take place without collisions



CSMA/CA with RTS/CTS

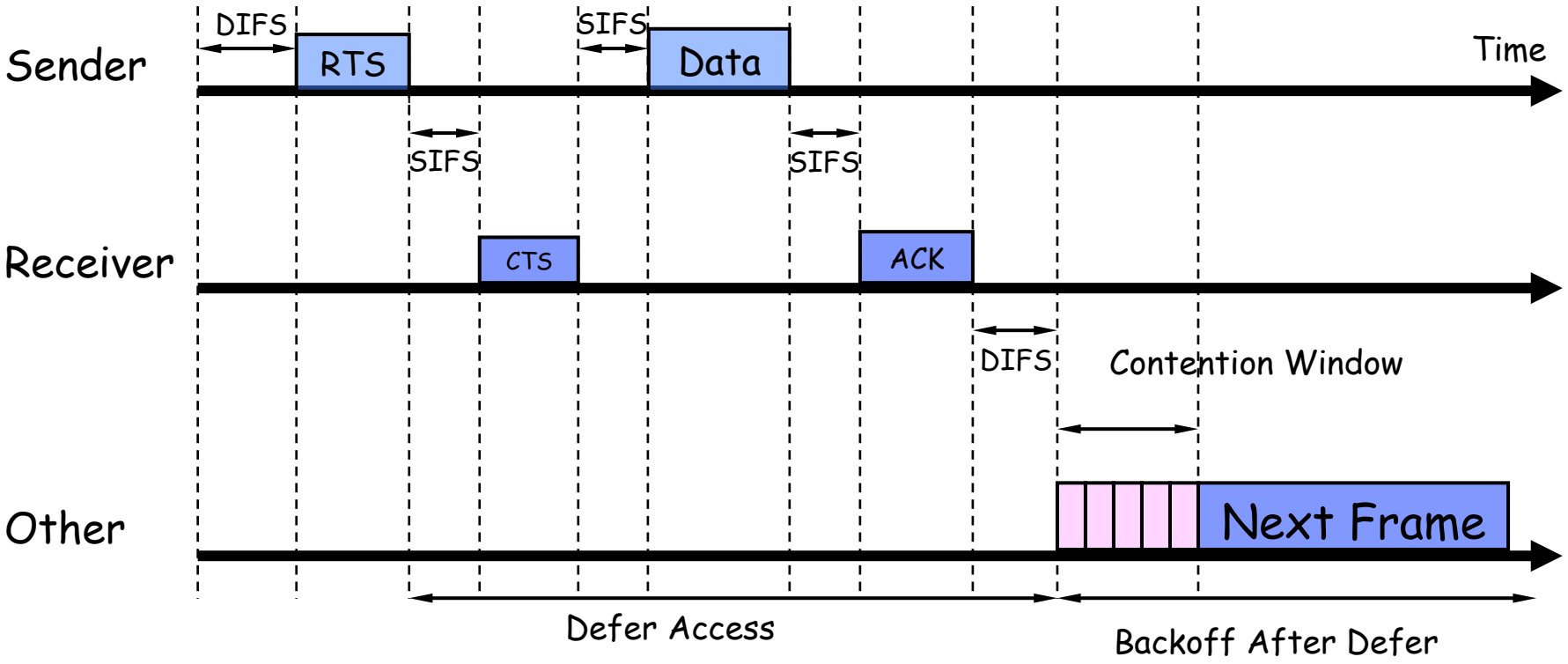
- Transmitter sends an RTS (Request To Send) after medium has been idle for time interval more than DIFS
- Receiver responds with CTS (Clear To Send) after medium has been idle for SIFS
- Then data is transmitted
- RTS/CTS is used for reserving channel for data transmission so that the collision can only occur in control message

CSMA/CA with RTS/CTS

- Use short signaling packets for Collision Avoidance
  - RTS Packet (Request To Send, 20 Bytes)
    - A sender requests the right to send from a receiver with a short RTS packet before it sends a data packet
  - CTS Packet (Clear To Send, 16 Bytes)
    - The receiver grants the right to send as soon as it is ready to receive
  - They contain

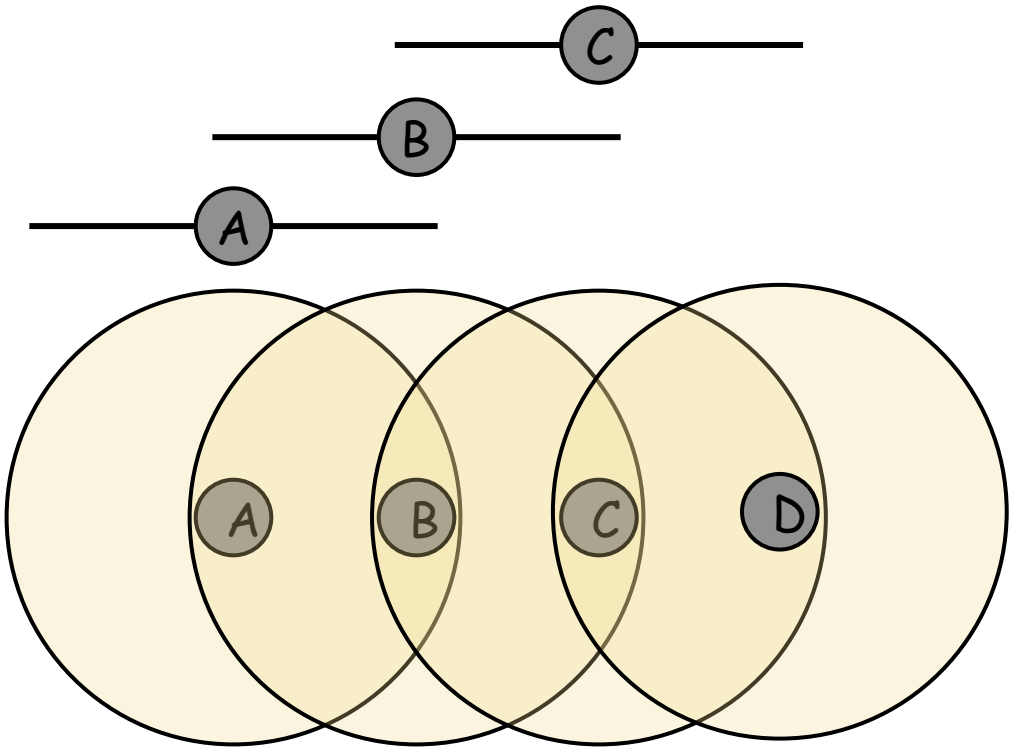
    (Sender Address, Receiver Address, Packet Size)

CSMA with RTS/CTS



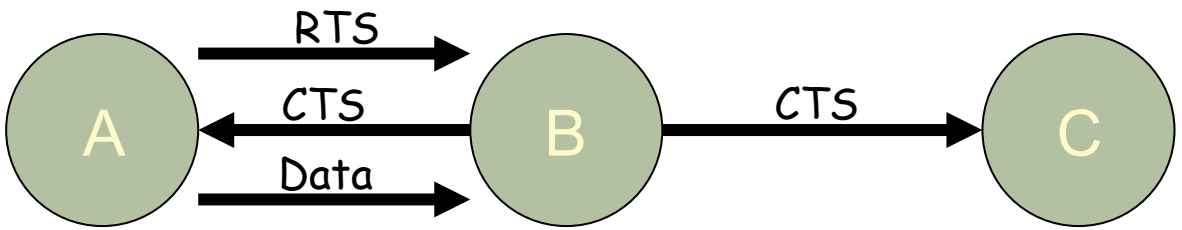
Problems with CSMA/CA

- The problems are still not solved
  - Hidden terminal problem
  - Exposed terminal problem



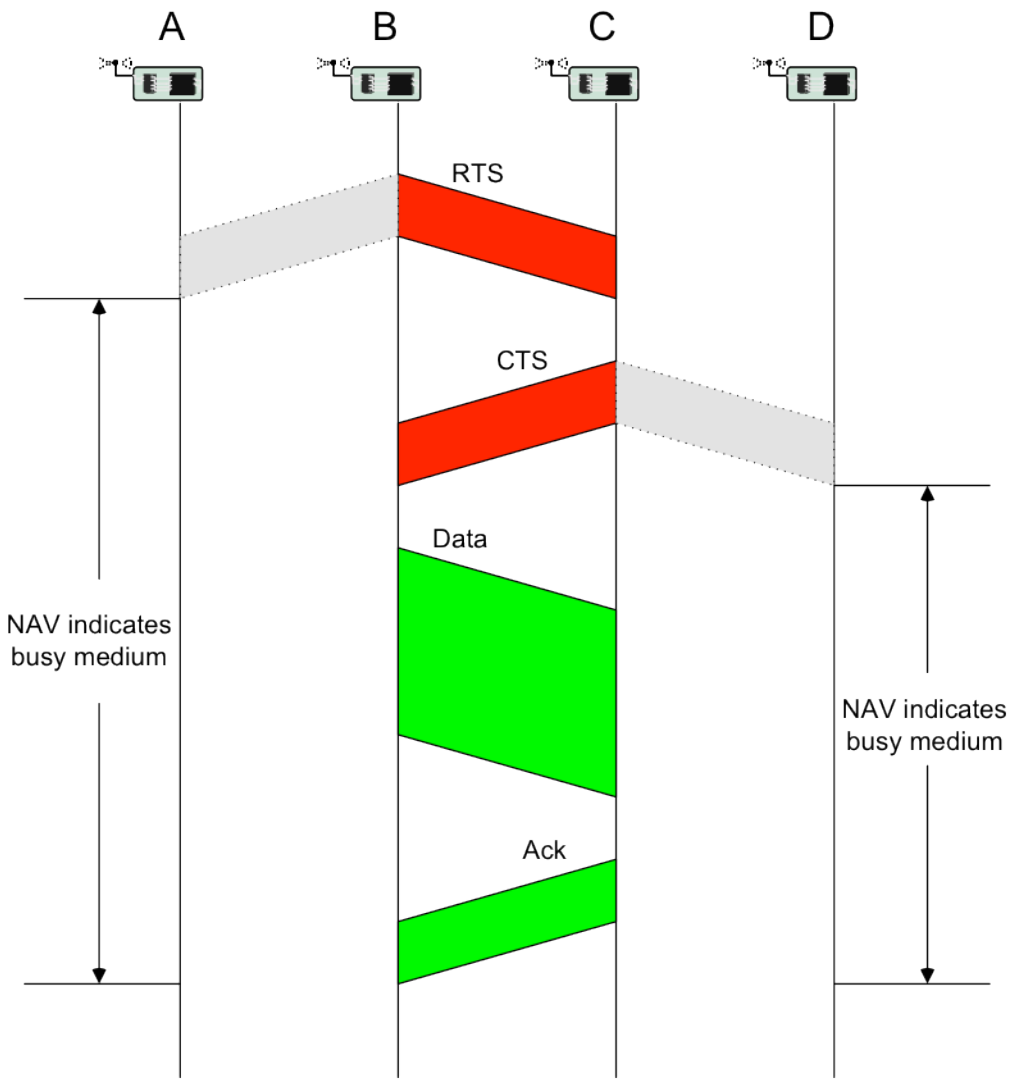
Hidden terminal problem

- A sends RTS
- B sends CTS
- C overhears CTS
- C inhibits its own transmitter
- A successfully sends Data to B



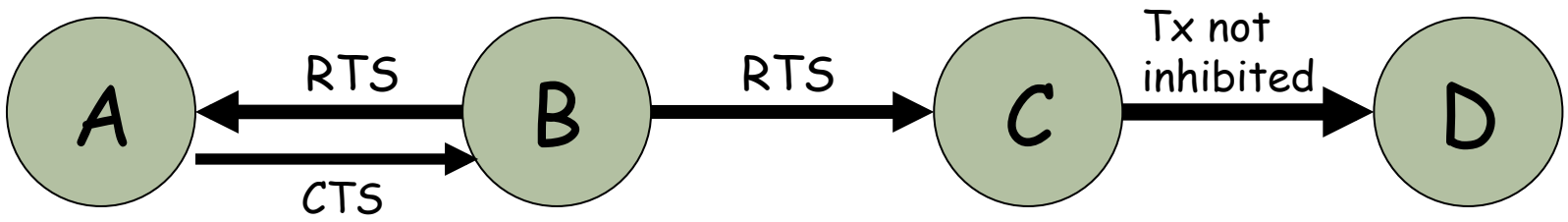
Hidden terminal problem

- How does C know how long to wait before it can attempt a transmission?
  - A includes length of Data that it wants to send in the RTS packet
  - B includes this information in the CTS packet
  - C, when it overhears the CTS packet, retrieves the length information and uses it to set the inhibition time
- Network Allocation Vector (NAV)
  - MACA protocol used in 802.11



Exposed terminal problem

- B sends RTS to A (overheard by C)
- A sends CTS to B
- C cannot hear A's CTS
- C assumes A is either down or out of range
- C does not inhibit its transmissions to D



Collisions

- Still possible – RTS packets can collide!
- Binary exponential backoff performed by stations that experience RTS collisions
- RTS collisions not as bad as data collisions in CSMA (since RTS packets are typically much smaller than DATA packets)

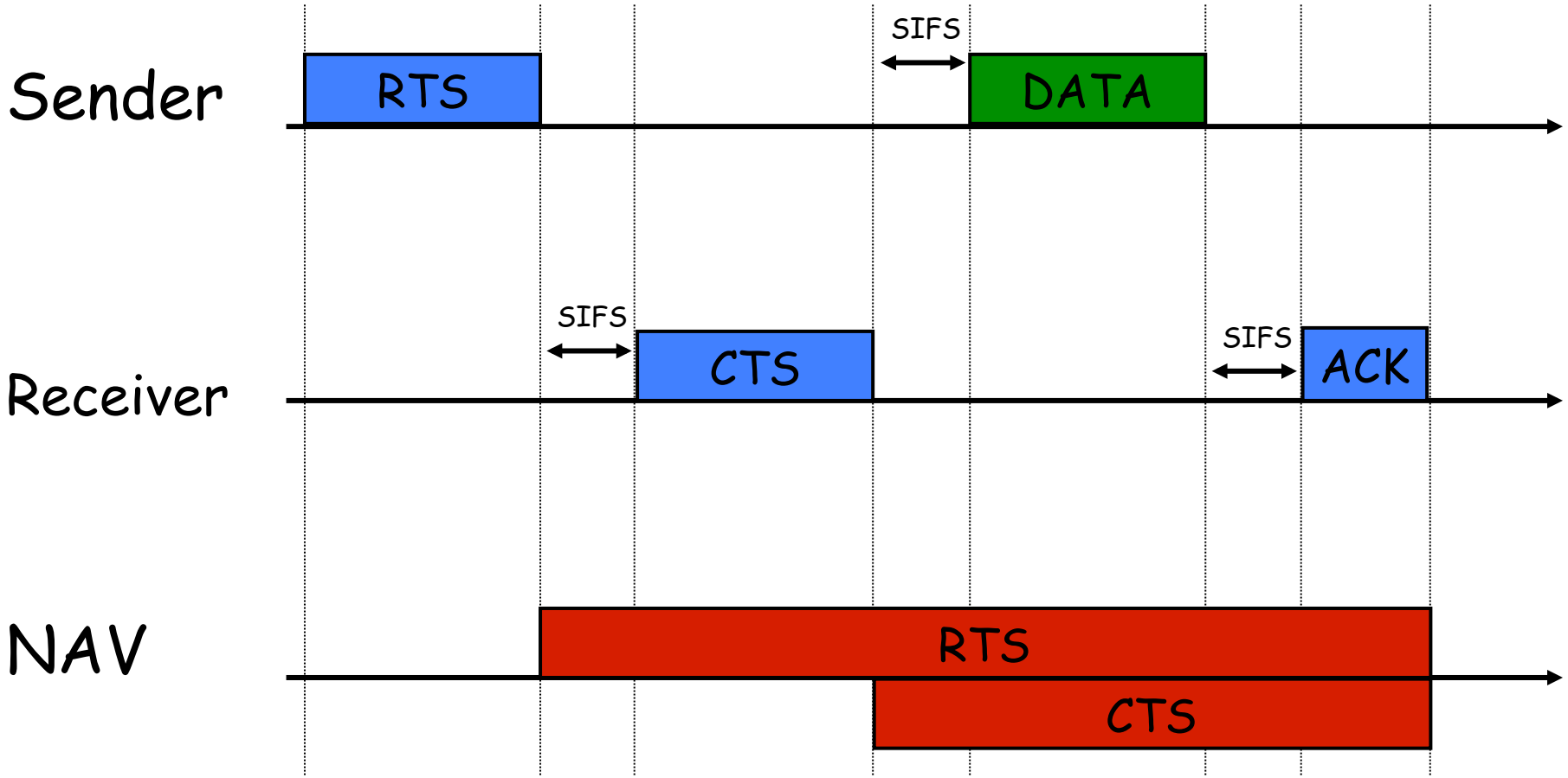
CSMA/CA with RTS/CTS

- Both Physical Carrier Sensing and Virtual Carrier Sensing used in 802.11
- If either function indicates that the medium is busy, 802.11 treats the channel to be busy
- Virtual Carrier Sensing is provided by NAV (Network Allocation Vector)

CSMA/CA with RTS/CTS

- Most 802.11 frames carry a duration field which is used to reserve the medium for a fixed time period
- Tx sets the NAV to the time for which it expects to use the medium
- Other stations start counting down from NAV to 0
- As long as NAV > 0, the medium is busy
- CHANNEL VIRTUALLY BUSY -> a NAV is turned on!
- The transmission will be delayed until the NAV expires
- When the channel is virtually available, then MAC checks for PHY condition of the channel

CSMA/CA with RTS/CTS



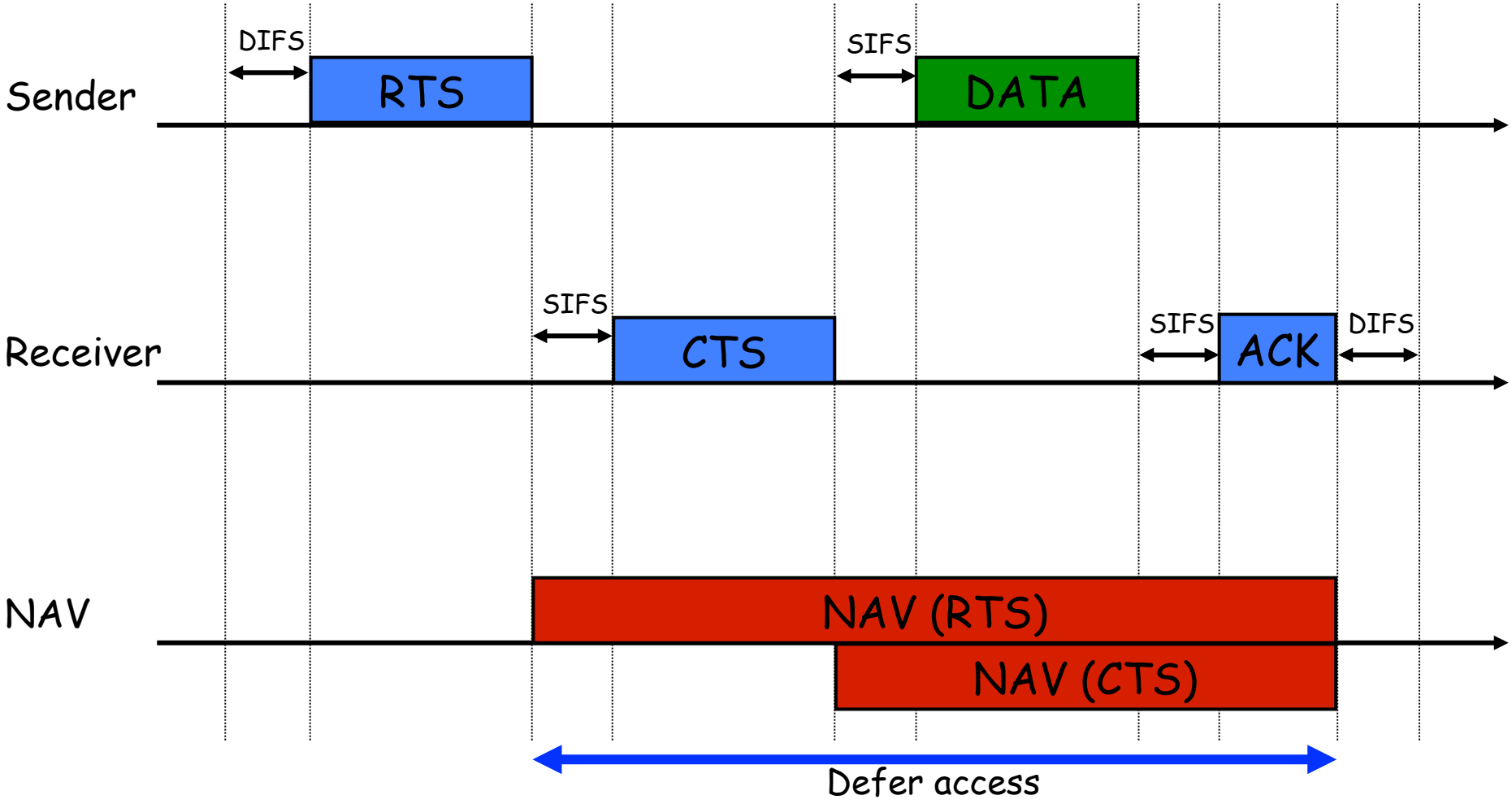
CSMA/CA with RTS/CTS

- If receiver receives RTS, it sends CTS (Clear to Send) after SIFS
- CTS again contains duration field and all stations receiving this packet need to adjust their NAV
- Sender can now send data after SIFS, acknowledgement via ACK by receiver after SIFS

CSMA/CA with RTS/CTS

- Every station receiving the RTS that is not addressed to it, will go to the Virtual Carrier Sensing Mode for the entire period identified in the RTS/CTS communication, by setting their NAV signal on
- Network Allocation Vector (NAV) is set in accordance with the duration of the field
- NAV specifies the earliest point at which the station can try to access the medium
- Thus, the source station sends its packet without contention
- After completion of the transmission, the destination terminal sends an ACK and NAV signal is terminated, opening the contention for other users

CSMA/CA with RTS/CTS

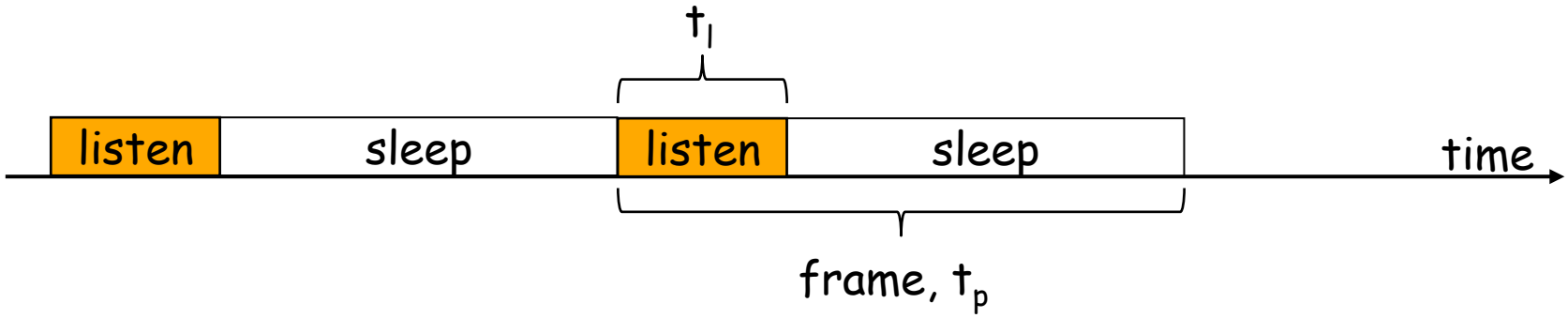


Contention based protocols

S-MAC

S-MAC: Sensor-MAC

- Problem: "Idle Listening" consumes significant energy
- Solution: Periodic listen and sleep



- During sleeping, radio is turned off
- Important parameter: "duty cycle" -> $\text{duty cycle} = \frac{t_l}{t_p}$
- Reduce duty cycle to ~ 10%
  - Listen for 200ms and sleep for 1.8s
- Property: High latency, low energy

S-MAC

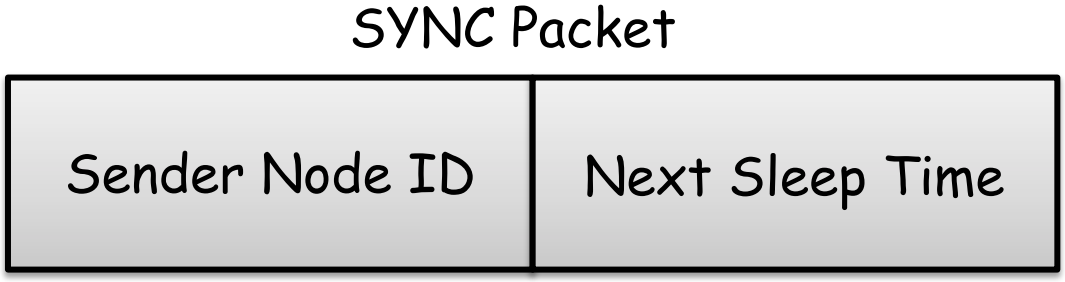
- Each node goes into periodic sleep mode during which it switches the radio off and sets a timer to awake later
- When the timer expires it wakes up and listens to see if any other node wants to talk to it
- The duration of the sleep and listen cycles are application dependent and they are set the same for all nodes
- Requires a periodic synchronization among nodes to take care of any type of clock drift

Periodic Sleep and Listen

- All nodes are free to choose their own listen/sleep schedules
- To reduce control overhead, neighboring nodes are synchronized together
- They listen at the same time and go to sleep at the same time
  - -> synchronized sleep

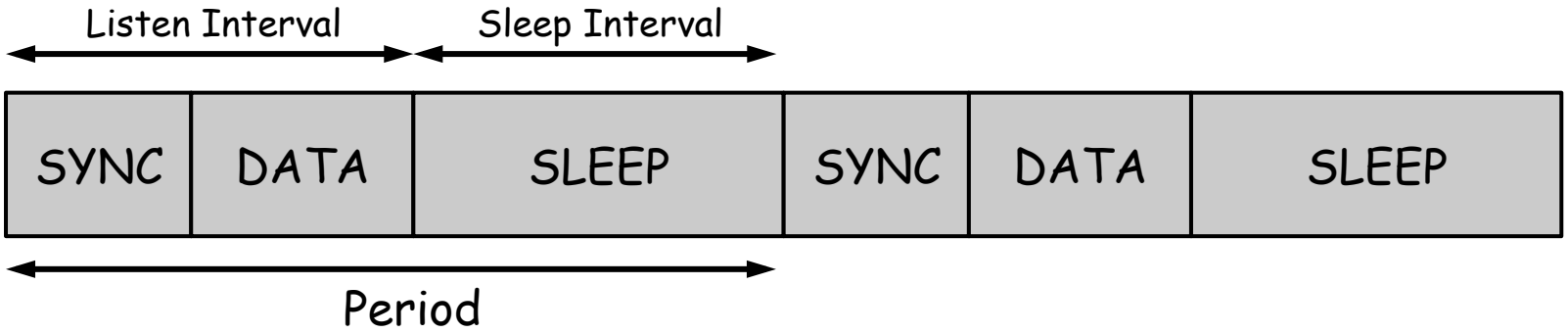
Synchronization

- SYNC packets are exchanged periodically to maintain schedule synchronization.

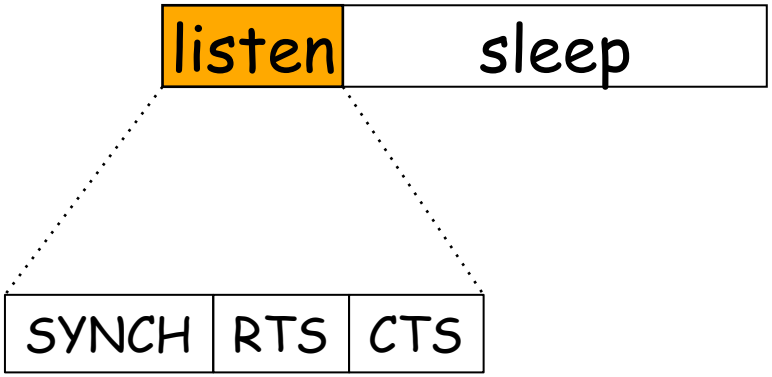


- Synchronization period:
  - Period for a node to send a SYNC packet
- Receivers will adjust their timer counters immediately after they receive the SYNC packet

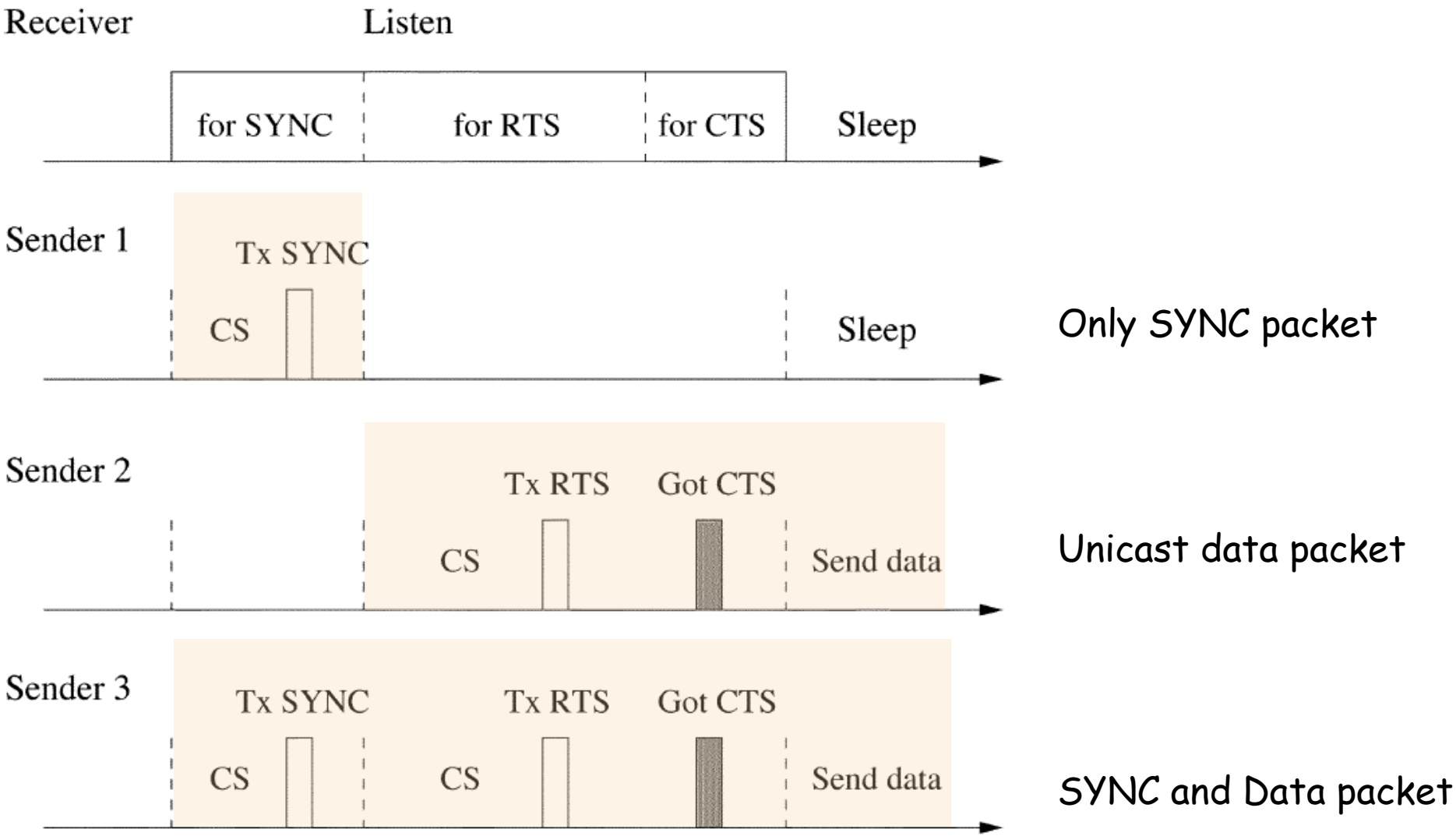
Periodic listen and sleep



Periodic listen and sleep



Maintaining synchronization

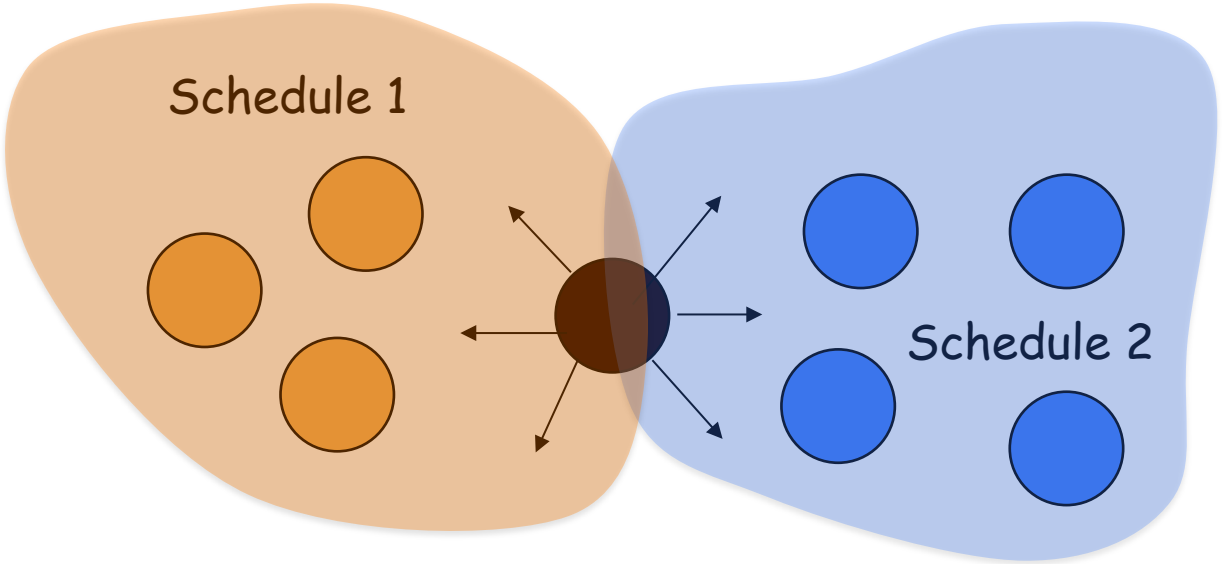


Choosing and maintaining schedules

- Each node maintains a schedule table that stores schedules of all its known neighbors
- For initial schedule:
  - A node first listens to the medium for a certain amount of time (at least the synchronization period)
  - If it does not hear a schedule (SYNC packet) from another node, it randomly chooses a schedule and broadcasts its schedule with a SYNC packet immediately
  - This node is called a Synchronizer
- If a node receives a schedule from a neighbor before choosing its own schedule
  - It follows this neighbor's schedule
  - Becomes a Follower
  - Waits for a random delay and broadcasts its schedule

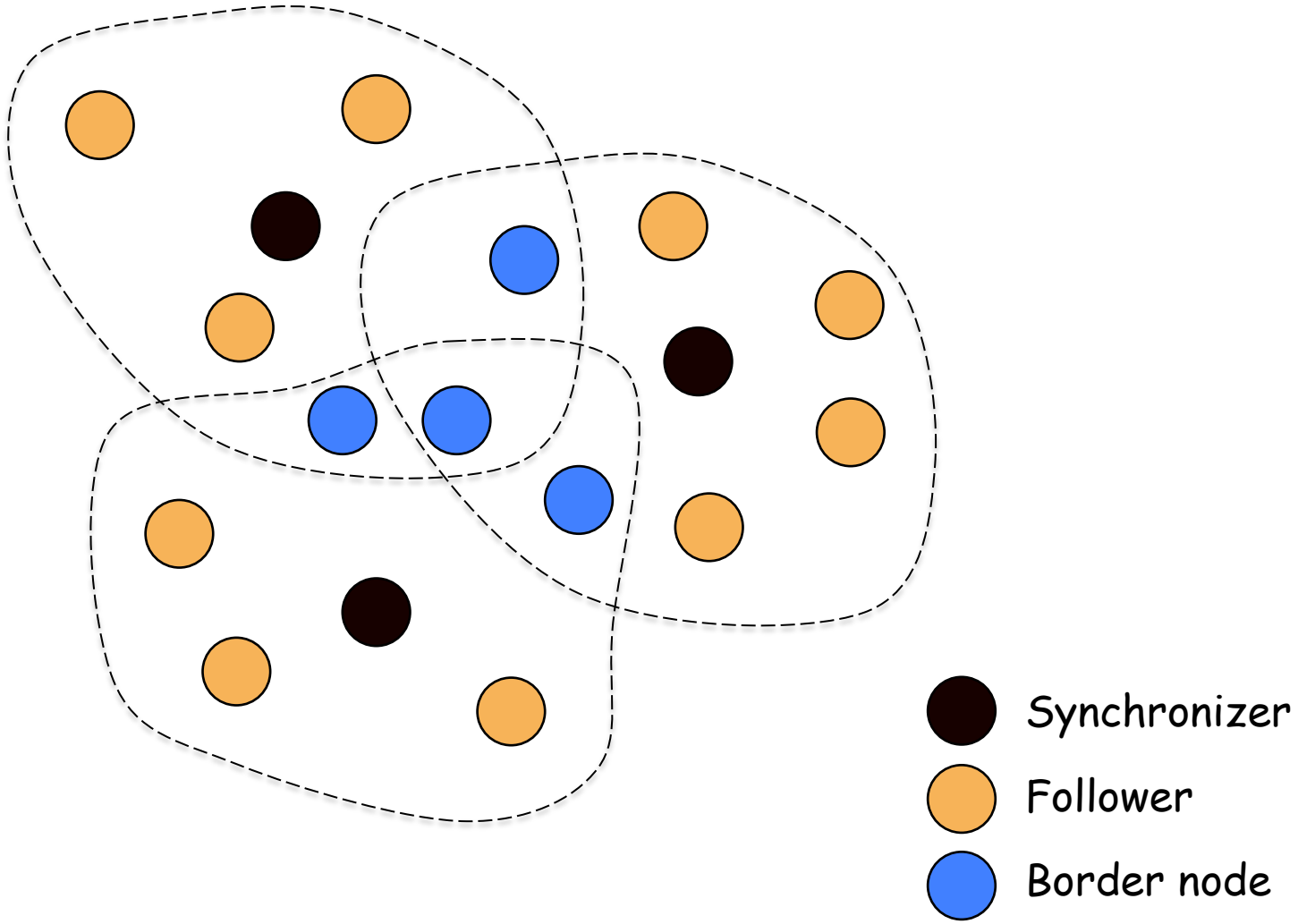
Coordinated sleeping

- In a large network, we cannot guarantee that all nodes follow the same schedule.
- The node on the border will follow both schedules.
- When it broadcasts a packet, it needs to do it twice, first for nodes on schedule 1 and then for those on schedule 2.



Coordinated sleeping

Virtual clusters in S-MAC



Border nodes

- Border nodes have less time to sleep and consume more energy than others.

- Option: Let border nodes adopt only one schedule
  - -> First received

Collision avoidance

- S-MAC is based on contention, i.e., if multiple neighbors want to talk to a node at the same time, they will try to send when the node starts listening.

- Similar to IEEE802.11, i.e., use RTS/CTS mechanism to address the hidden terminal problem

- Perform carrier sense before initiating a transmission

Collision avoidance

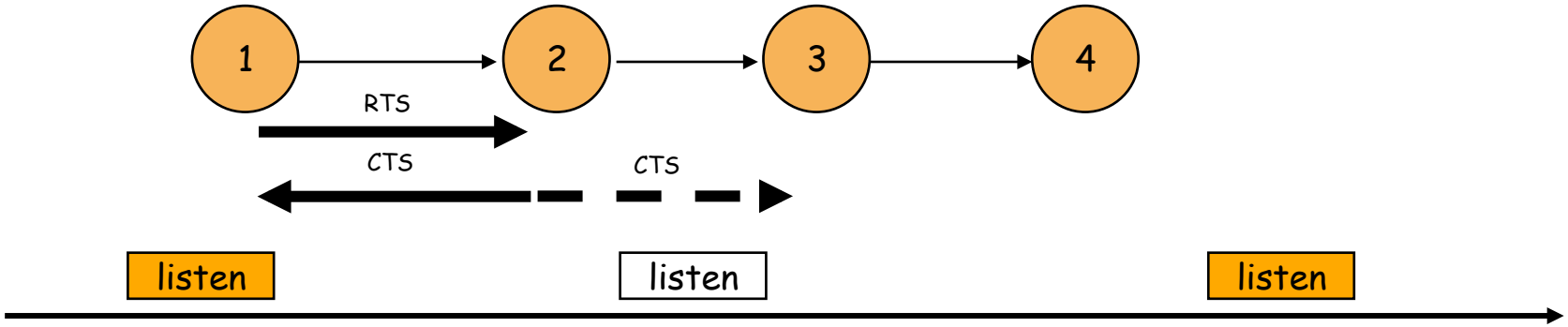
- If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again
- Broadcast packets are sent without using RTS/CTS
- Unicast data packets follow the sequence of RTS/CTS/ DATA/ACK between the sender and receiver
- Duration field in each transmitted packet indicates how long the remaining transmission will be so if a node receives a packet destined to another node, it knows how long it has to keep silent
- The node records this value in network allocation vector (NAV) and sets a timer for it
- When a node has data to send, it first looks at NAV
- If this value is not zero, then medium is busy (virtual carrier sense)

Collision avoidance

- The medium is determined as free if both virtual and physical carrier sense indicate the medium is free

- All immediate neighbors of both the sender and receiver should sleep after they hear RTS or CTS packet until the current transmission is over

Adaptive listening feature



- Reduce multi-hop latency due to periodic sleep
  - BASIC IDEA: Let the node which overhears its neighbor's transmission stay awake
- Both neighbors will learn about how long the transmission is from the duration field in the RTS and CTS packets
- They are able to adaptively wake up when the transmission is over
- Reduce latency by at least half (e.g., CTS of 2 is heard by 3 also, 3 remains awake!!)

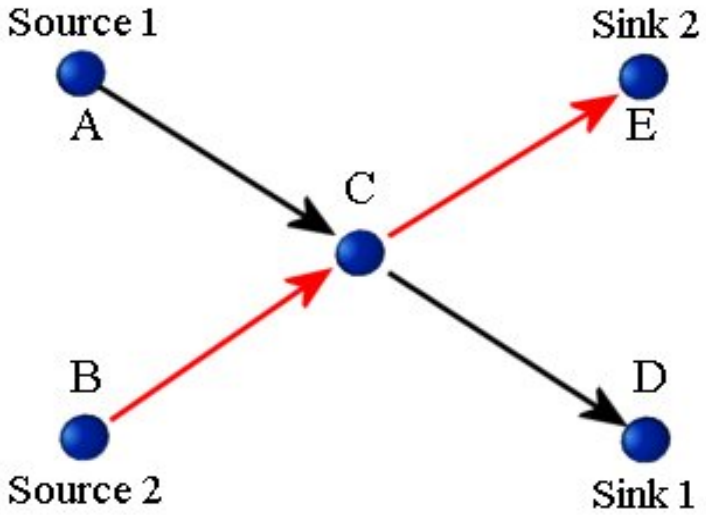
Message passing feature

- Long messages are broken down in to smaller packets and sent continuously once the channel is acquired by RTS/ CTS handshake

- Increases the sleep time, but leads to fairness problems

S-MAC: Example

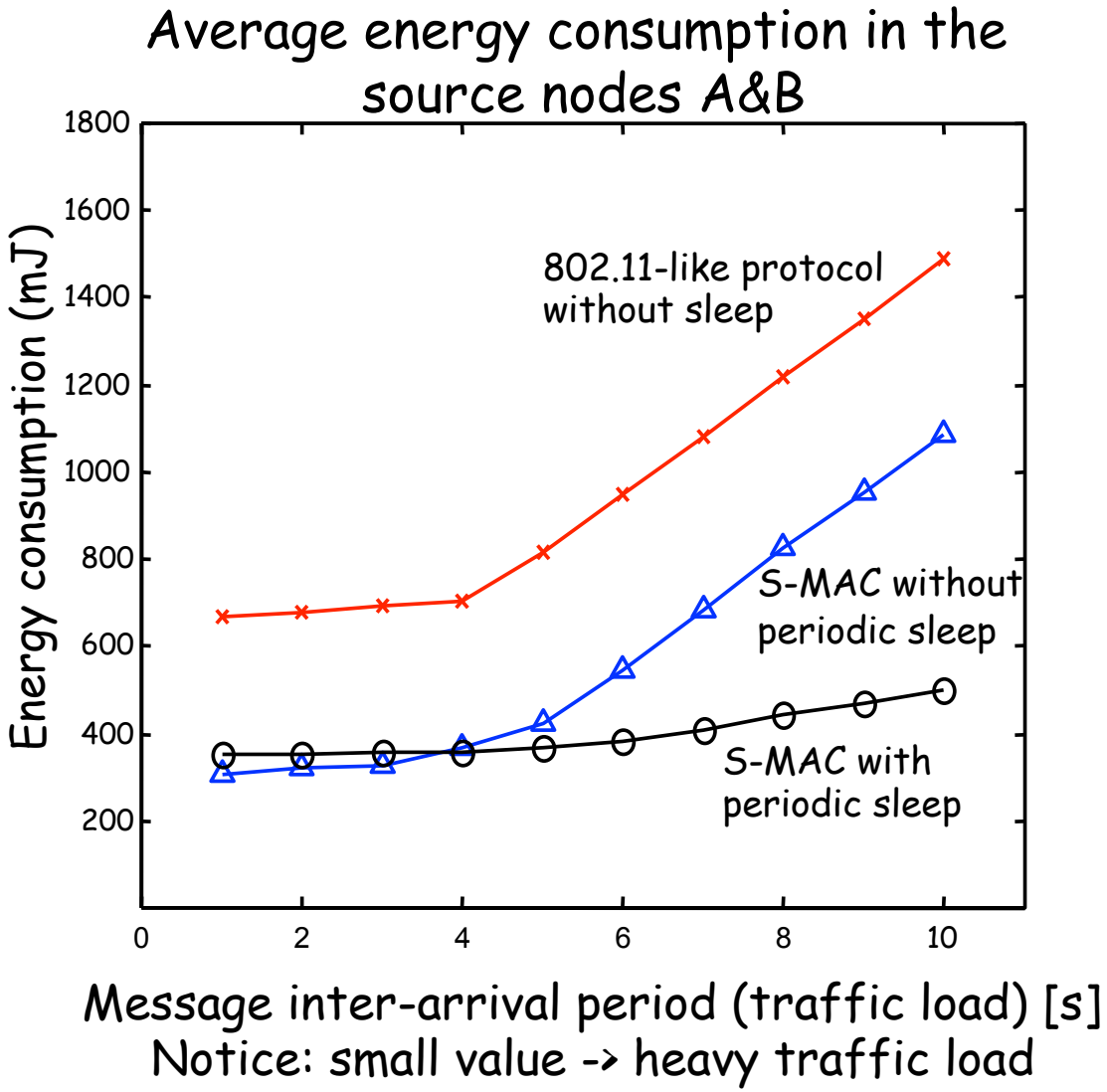
- Topology
  - Two-hop network with two sources and two sinks
- Sources periodically generate a sensing message which is divided into fragments
- Traffic load is changed by varying the inter-arrival period of the messages:
  - For inter-arrival period of 5s, message is generated every 5s by each source
  - Here inter-arrival period varies between 1-10s



S-MAC: Example

- In an experiment, a source node generates 10 messages
- Each message has 10 fragments
  - 200 data packets to be passed from source to sink nodes
- Each fragment has 40 bytes
  - 8000 bytes at all
- The total energy consumption of each node is measured for sending this fixed amount of data

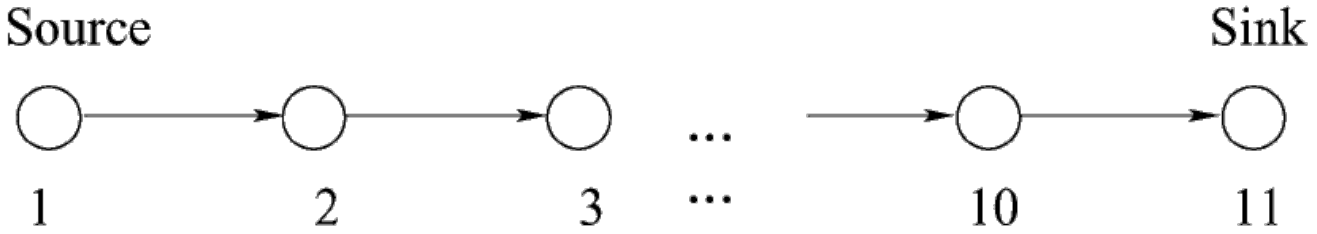
Experiments



- S-MAC consumes much less energy than 802.11-like protocol without sleeping
- At heavy load, idle listening rarely happens, energy savings from sleeping is very limited. SMAC achieves energy savings by avoiding overhearing and efficiently transmitting long messages
- At light load, periodic sleeping plays the key role

Energy consumption over multi-hops

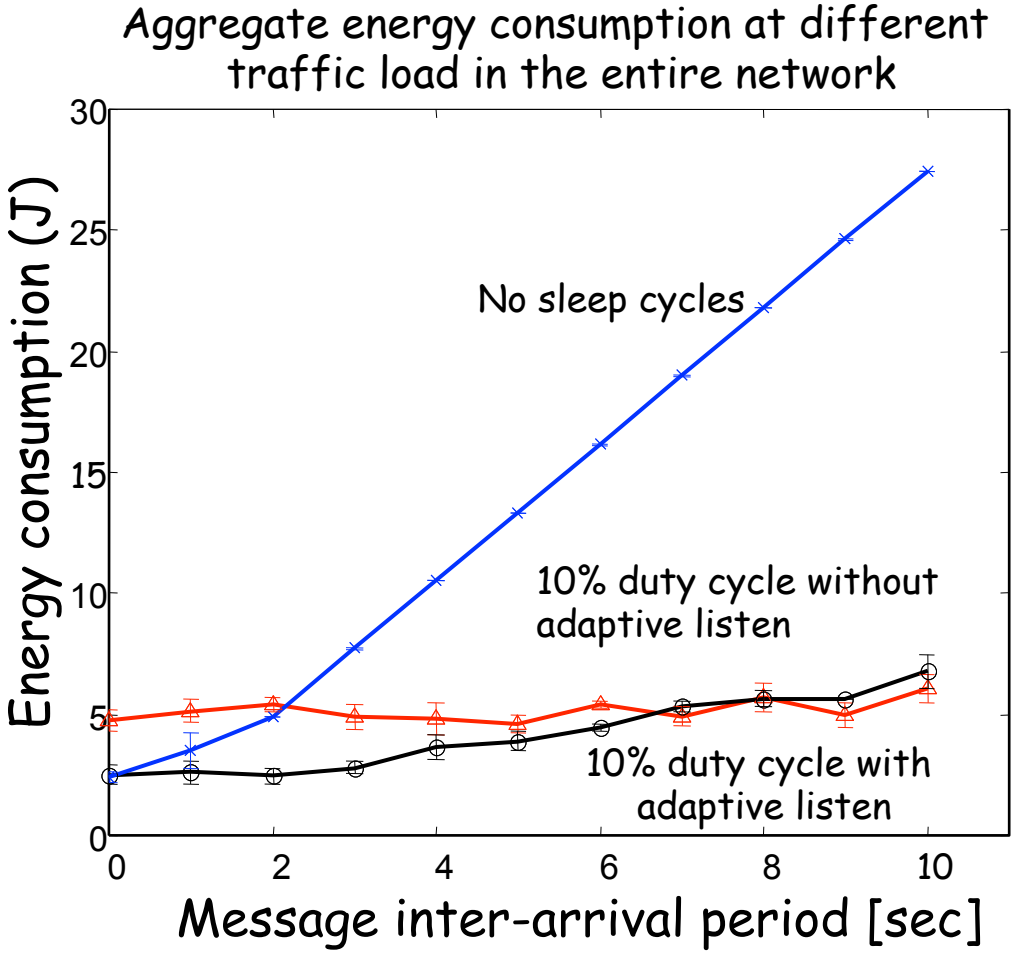
- Ten-hop linear network at different traffic



- Load
  - inter-arrival time 0-10s
  - source node sends 20 messages with each 100 bytes long
- 3 configurations for S-MAC
  - No sleep cycles
  - 10% duty cycle without adaptive listening
  - 10% duty cycle with adaptive listening
- Periodic listen interval 115ms
  - 10% duty cycle means a frame length for 1.15sec

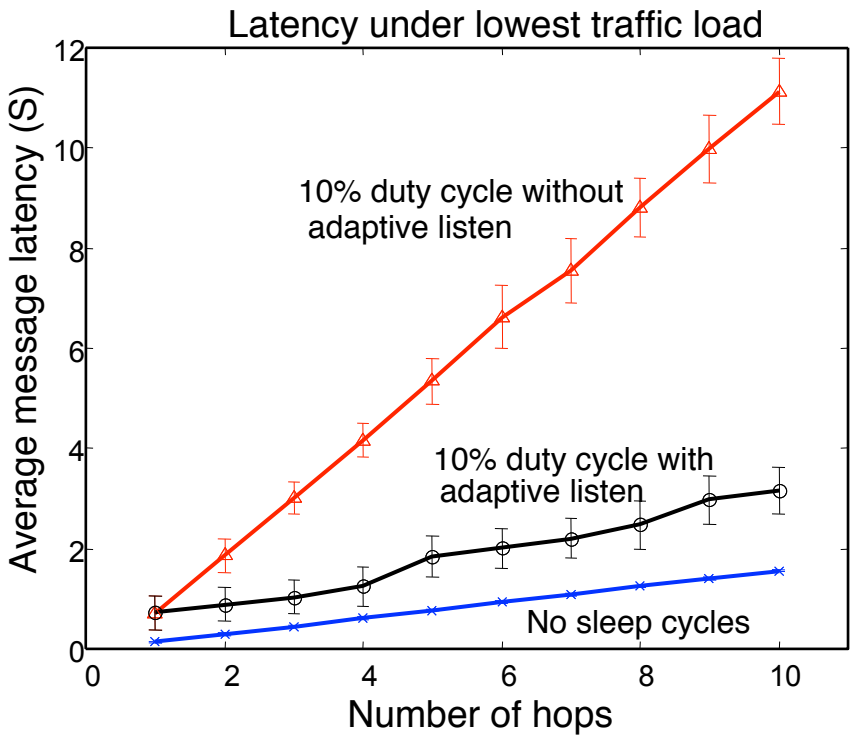
Energy consumption over multi-hops

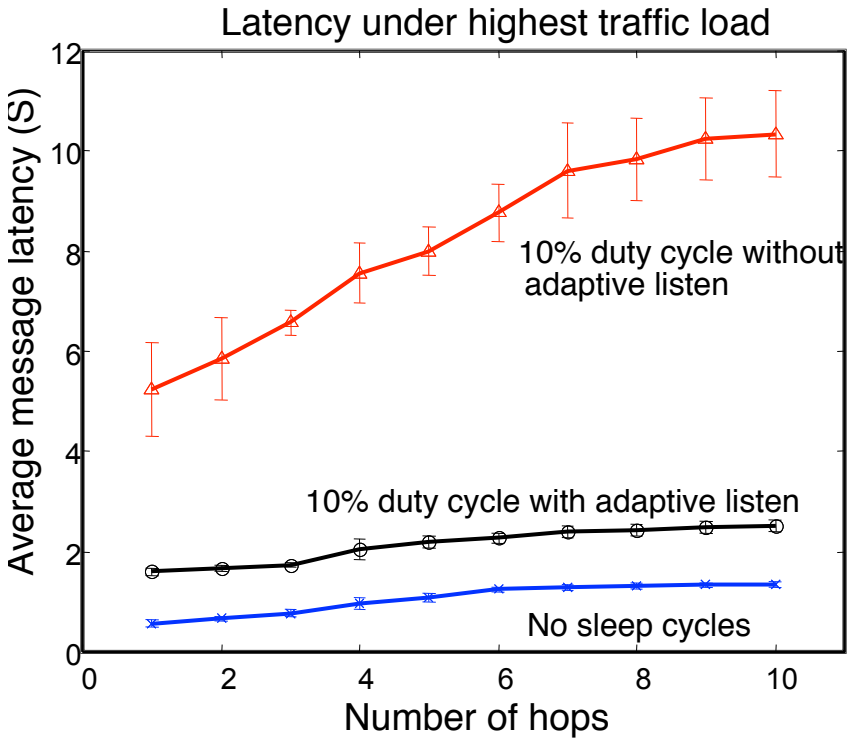
- At light traffic load, periodic sleeping has significant energy savings over fully active mode
- Adaptive listen saves more at heavy load by reducing latency



Latency as hops increase

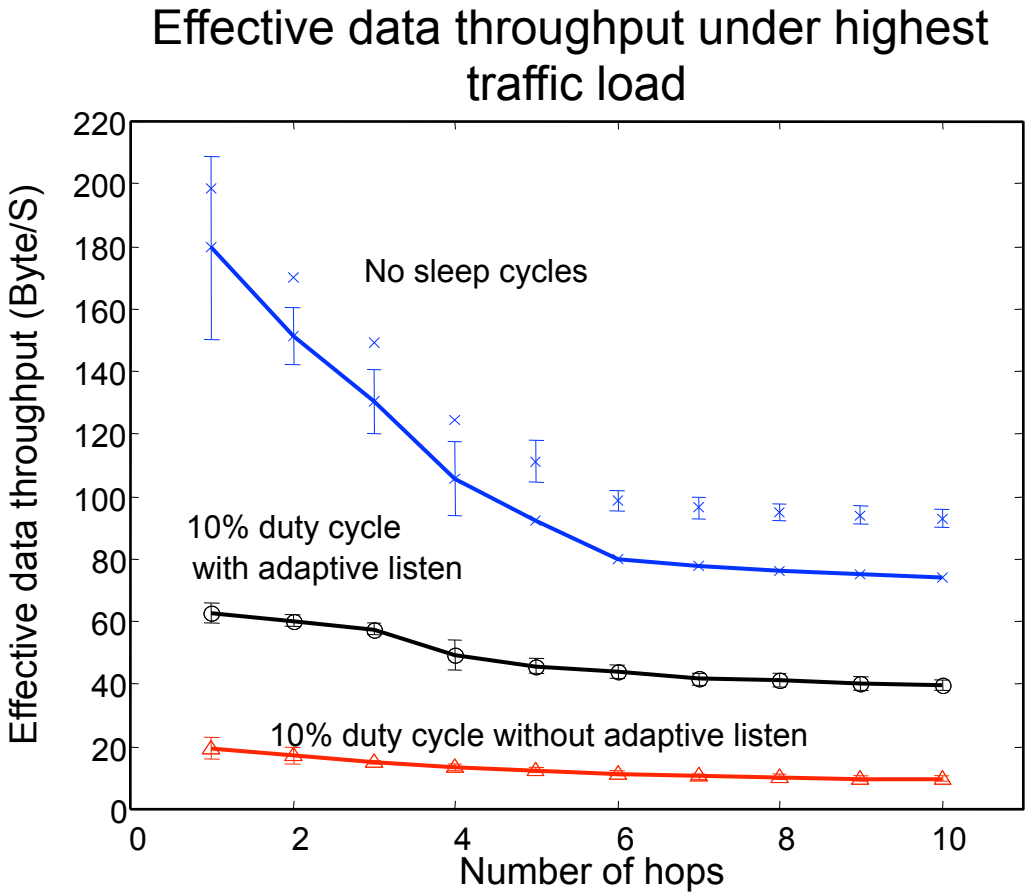
- Adaptive listen significantly reduces latency causes by periodic sleeping





Throughput as hops increase

- Adaptive listen significantly increases throughput
- Uses less time to pass the same amount of data



S-MAC: Summary

- A mainly static network is assumed
- Trades off latency for reduced energy consumption
- Redundant data is still sent with increased latency
- Increased collision rate due to sleep schedules

Contention based protocols

B-MAC

B-MAC: Berkeley MAC

- Properties
  - Keep core MAC simple
  - Provides basic CSMA access
  - Optional link level ACK -> no link level RTS/CTS
  - CSMA backoffs configurable by higher layers
  - Carrier sensing using Clear Channel Assessment (CCA)
  - Sleep/Wake scheduling using Low Power Listening (LPL)

Goals of B-MAC

- Low Power Operation
- Effective Collision Avoidance
- Simple Implementation, Small Code and RAM Size

Comparison of the size of B-MAC and S-MAC in bytes. Both protocols are implemented in TinyOS.

Protocol	ROM	RAM
B-MAC	3046	166
B-MAC w/ ACK	3340	168
B-MAC w/ LPL	4092	170
B-MAC w/ LPL & ACK	4386	172
B-MAC w/ LPL & ACK + RTS-CTS	4616	277
S-MAC	6274	516

- Efficient Channel Utilization
- Reconfigurable by Network Protocols
- Tolerant to Changing RF/Networking Conditions
- Scalable to Large Numbers of Nodes

B-MAC TinyOS Interfaces

- B-MAC Design
  - Clear Channel Assessment (CCA)
  - Packet Backoffs
  - Link Layer Acknowledgments
  - Low Power Listening (LPL)

- TinyOS Interface
  - Flexible control of B-MAC by higher layer services
  - Allow services to toggle CCA and ACKs
  - Set backoffs on a per message basis
  - Change the LPL mode for transmit and receive

```
interface MacControl {
  command result_t EnableCCA();
  command result_t DisableCCA();
  command result_t EnableAck();
  command result_t DisableAck();
  command void* HaltTx();
}

interface MacBackoff {
  event uint16_t initialBackoff(void* msg);
  event uint16_t congestionBackoff(void* msg);
}

interface LowPowerListening {
  command result_t SetListeningMode(uint8_t mode);
  command uint8_t GetListeningMode();
  command result_t SetTransmitMode(uint8_t mode);
  command uint8_t GetTransmitMode();
  command result_t SetPreambleLength(uint16_t bytes);
  command uint16_t GetPreambleLength();
  command result_t SetCheckInterval(uint16_t ms);
  command uint16_t GetCheckInterval();
}
```

Clear Channel Assessment

- Effective collision avoidance
- Find out whether the channel is idle
  - If too pessimistic: waste bandwidth
  - If too optimistic: more collisions
- Key observation
  - Ambient noise may change significantly depending on the environment
  - Packet reception has fairly constant channel energy
  - Need to tell what is noise and what is a signal

- Software approach to estimating the noise floor
  -> B-MAC solution!

Clear Channel Assessment

- Take a signal strength sample when the channel is assumed to be free/idle
  - WHEN?
  - Right after a packet is transmitted or when no valid data is received
  - Samples are entered into a FIFO queue

Clear Channel Assessment

- Median of the queue S_t is added to an exponentially weighted moving average with decay α

- Median signal strength is used as a simple low pass filter to add robustness to the noise floor estimate A_t

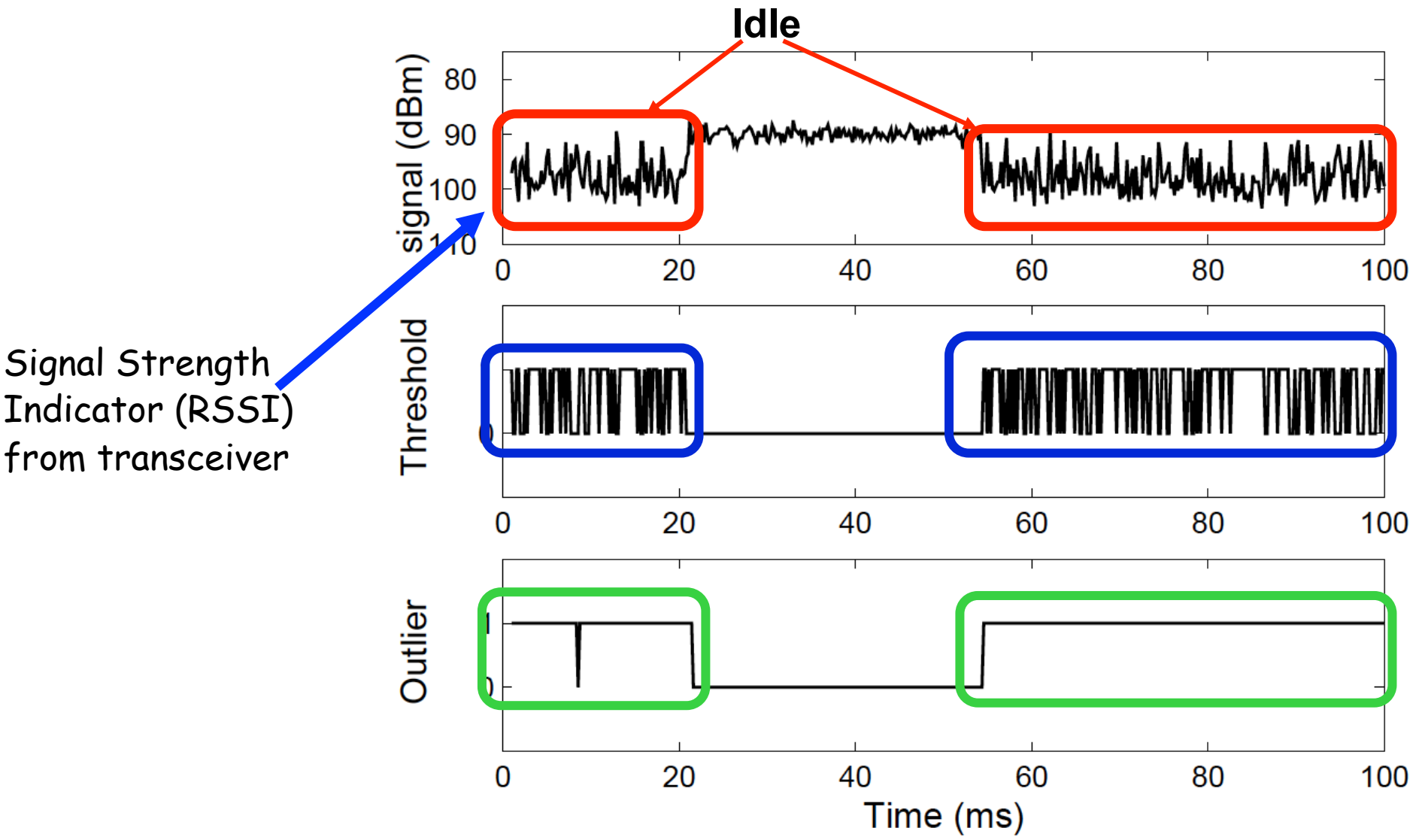
$$A_t = \alpha S_t + (1-\alpha) S_{t-1}$$

- Assumptions: α=0.06 and FIFO queue size=10
- Once a good estimate of the noise floor is established, a request to transmit a packet starts the process of monitoring the received signal from the radio

Single-Sample Thresholding vs Outlier Detection

- Common approach: take single sample, compare to noise floor
  - Large number of false negatives -> lower effective channel BW
- B-MAC: search for outliers in received signal (RSSI)
  - If a sample has significantly lower energy than the noise floor during the sampling period, then the channel is clear
  - If 5 samples are taken and no outlier is found, the channel is busy

CCA vs. Threshold Techniques



CCA vs. Threshold Techniques

- Threshold: waste channel utilization
- CCA: Fully utilize the channel since a valid packet could have no outlier significantly below the noise floor
- A packet arrives between 22 and 54ms.
  - The middle graph shows the output of a thresholding CCA algorithm. (1: channel clear, 0: channel busy)
  - Bottom shows the output of an outlier detection algorithm

Clear Channel Assessment

- Before transmission, take a sample of the channel
- If the sample is below the current noise floor, channel clear, send immediately
- If five samples are taken, and no outlier found -> channel busy, take a random backoff
- Noise floor updated when the channel is known to be clear, e.g., just after packet transmission
- CCA can be turned on/off (see B-MAC-TinyOS interface)
- If turned off, a schedule-based protocol can be implemented above B-MAC
- If turned on, B-MAC uses an initial channel backoff when sending a packet

Clear Channel Assessment

- B-MAC does not set the backoff time, instead an event is signaled to the service that sent the packet via the MacBackoff interface
- The service may either return an initial backoff time or ignore the event
- If ignored, a small random backoff is used
- After the initial backoff, the CCA outlier algorithm is run
- If the channel is not clear, an event signals the service for a congestion backoff time
- If no backoff time is given, again a small random backoff is used
- Enabling or disabling CCA and configuring the backoff allows services to change the fairness and available throughput

Low Power Listening

- Goal: Minimize "Listen Cost"
- Principles
  - Node periodically wakes up, turns radio on and checks activity on the channel
    - Wakeup time fixed (time spend sampling RSSI?)
    - "Check time" variable
  - If energy/activity on the channel is detected, node powers up and stays awake for the time required to receive the incoming packet

Low Power Listening

- Node goes back to sleep
  - If the packet is received successfully
  - After a timeout (if no packet received (a false positive))
- Preamble length matches channel checking period
  - No explicit synchronization required
- Noise floor estimation used to detect channel activity during LPL

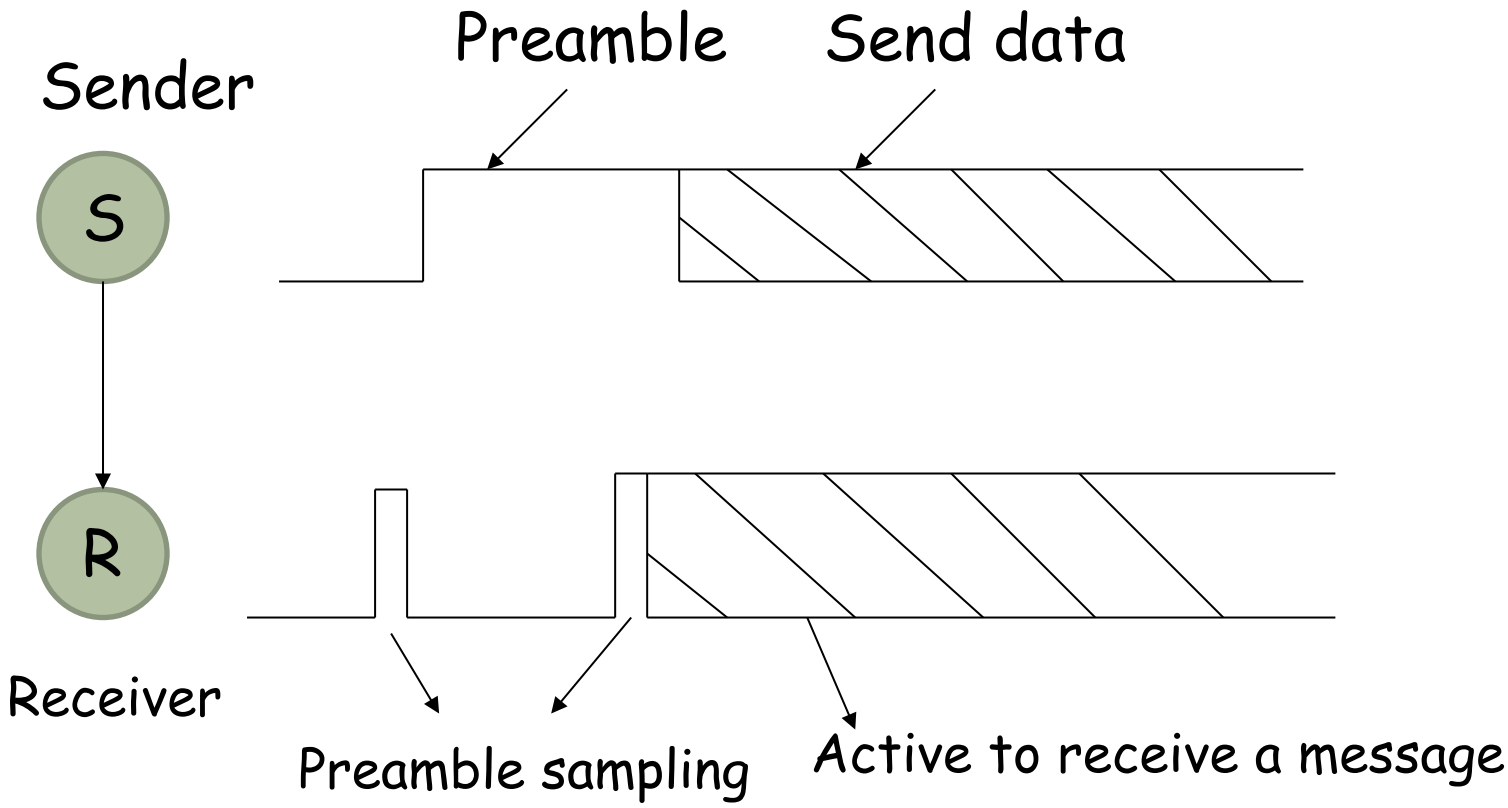
Check Interval for Channel Activity

- To reliably receive data, the preamble length is matched to the interval that the channel is checked for activity
- If the channel is checked for every 100 ms, the preamble must be at least 100 ms long for a node to wake up, detect activity on the channel, receive the preamble and then receive the message

- Interval between LPL samples is maximized so that the time spent sampling the channel is minimized.
- Transmit mode ~ Preamble length
- Listening mode ~ Check interval

LPL- Preamble Sampling

Preamble is not a packet but a physical layer RF pulse (Minimize overhead)

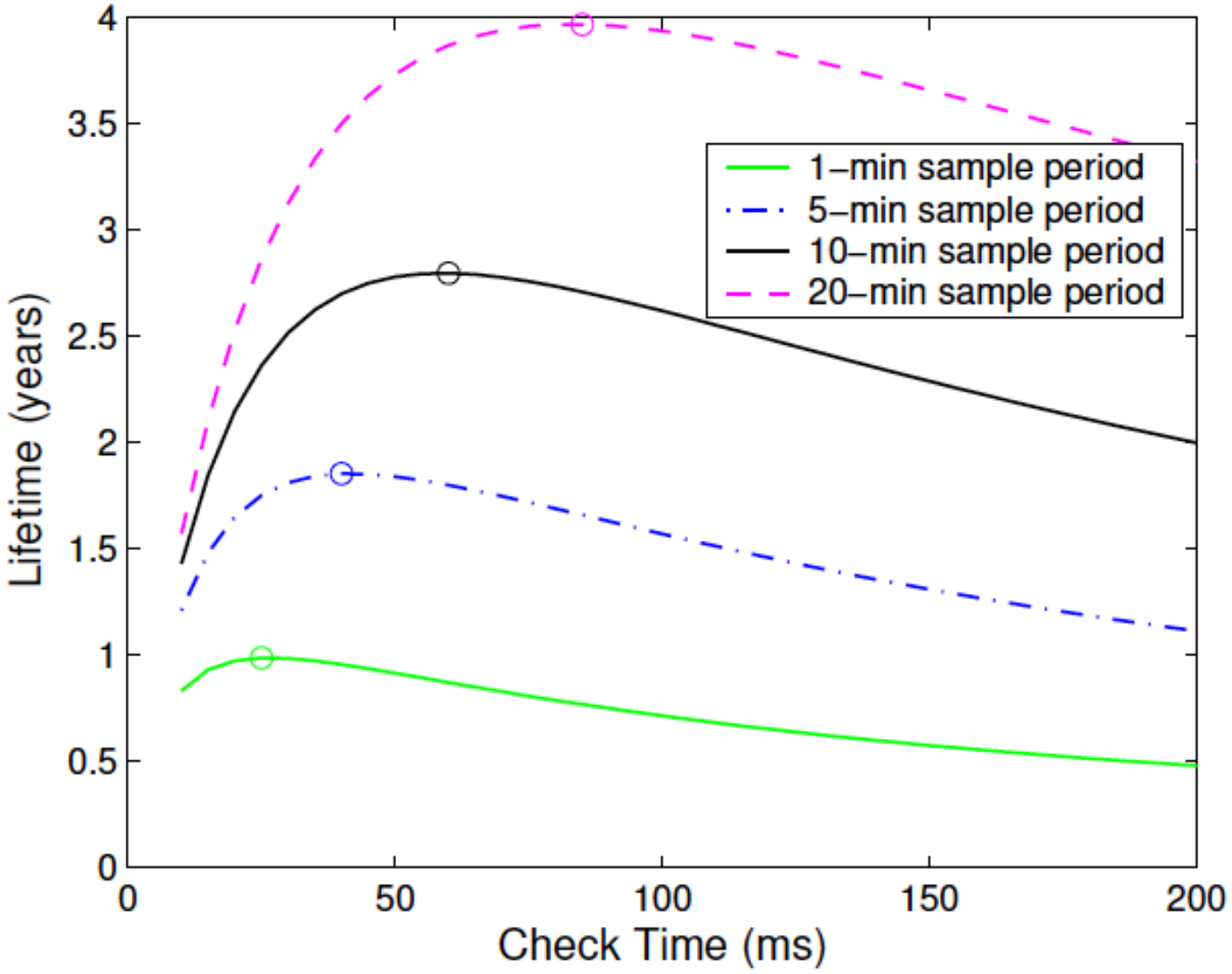


|Preamble| >= Sampling period

LPL Check Interval

- Sampling rate (traffic pattern) defines optimal check interval
- Check interval
  - Too small: energy wasted on idle listening
  - Too large: energy wasted on transmissions (long preambles)
- In general, it is better to have larger preambles than to check more often!
- More frequent checking of the radio
  - Shorter transmission time
  - More energy consumption

LPL Check Interval

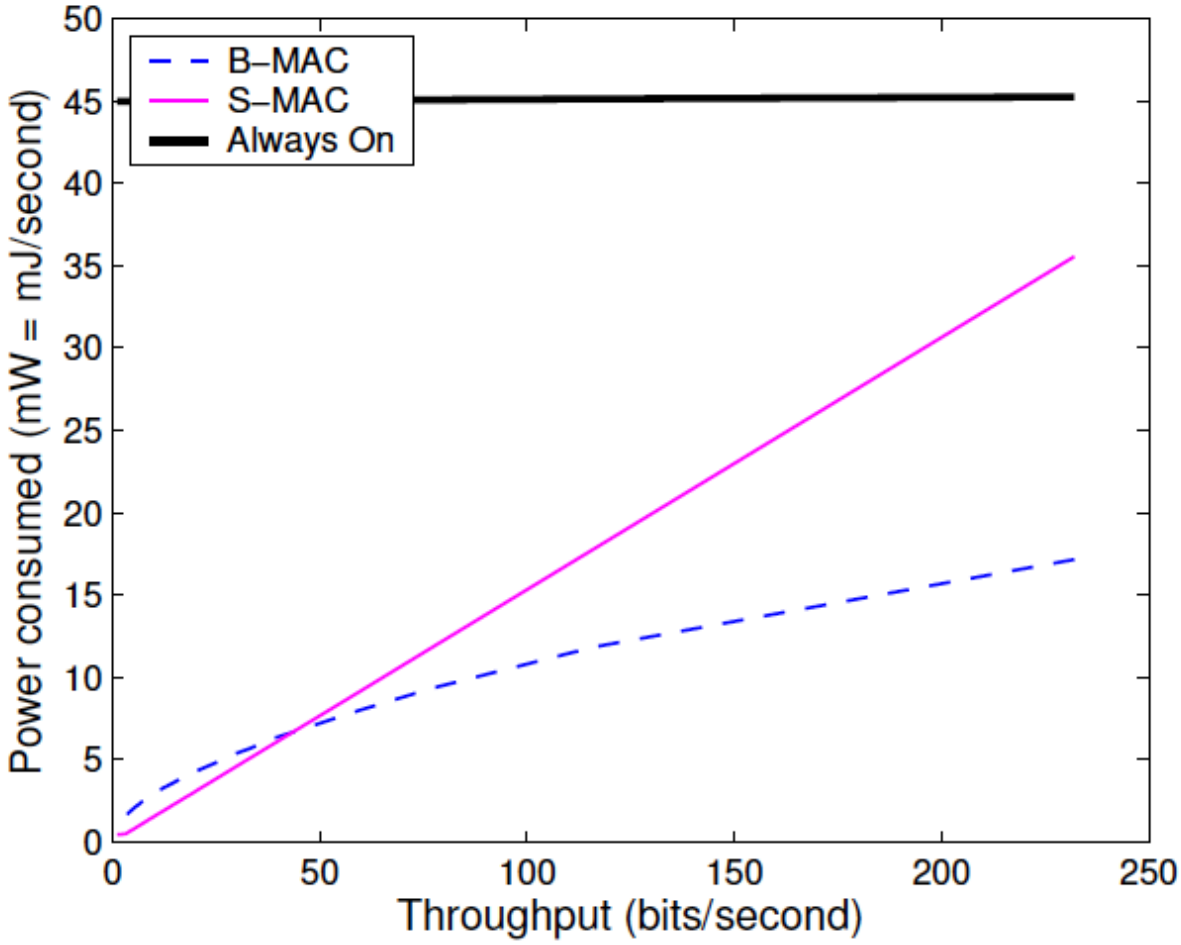


Experimental Results: Throughput

- "B-MAC is about 4.5 faster than SMAC-unicast"
  - Not as fast when ACK or RTS/CTS is used
  - Differences less pronounced as number of nodes increases
  - Another issue: B-MAC has CCA, thus it backs off less frequently (and perhaps the backoff timer is faster)
  - What about hidden terminal without RTS/CTS?

Throughput vs Power Consumption

- 10 nodes in a neighborhood
- Data must arrive within 10 seconds
- Average power consumption per node



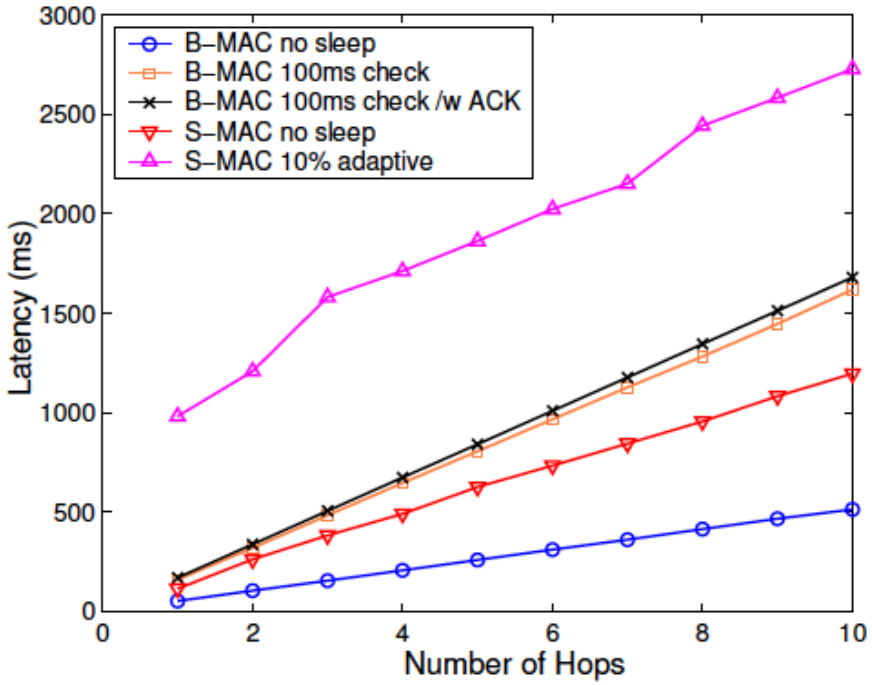
Throughput vs Power Consumption

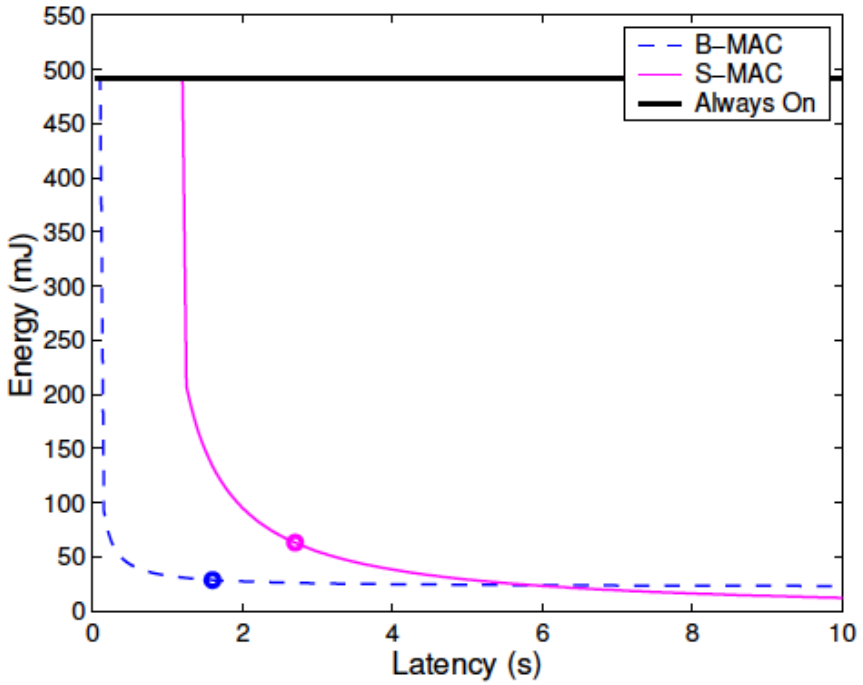
- Low data rates: S-MAC is better
  - Very low duty cycle
- Power vs throughput
  - S-MAC: linear
  - B-MAC: sub-linear
- Reason: S-MAC duty cycle must increase
  - More active periods, more SYNC periods
  - Cost of resynchronizing?
- B-MAC: larger preambles at low throughput, progressively becoming smaller

Length (bytes)	B-MAC	S-MAC
Preamble	8	18
Synchronization	2	2
Header	5	9
Footer (CRC)	2	2
Data Length	29	29
Total	**46**	**60**

Energy vs Latency

- 10-hop network
- Source sends a 100-byte packet every 10 seconds
- S-MAC: 10% Duty Cycle with Adaptive Listening
- B-MAC: choose optimal check interval
- SMAC again performs worse…
  - Reason: sync packets, prob. of multiple schedules->less time to sleep





Comparison of S-MAC and B-MAC

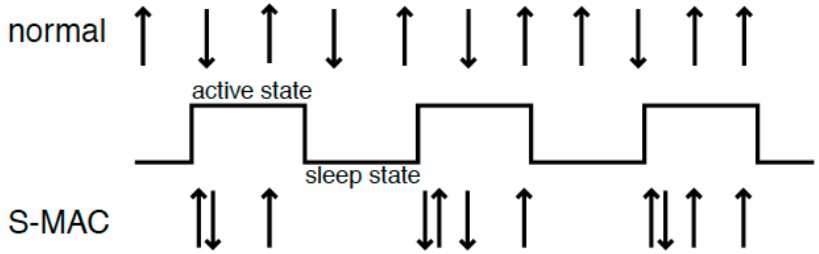
	S-MAC	B-MAC
Collision avoidance	CSMA/CA	CSMA
ACK	Yes	Optional
Message passing	Yes	No
Overhearing avoidance	Yes	No
Listen period	Pre-defined + adaptive listen	Pre-defined
Listen interval	Long	Very short
Schedule synchronization	Required	Not required
Packet transmission	Short preamble	Long preamble
Code size	6.3KB	4.4KB (LPL & ACK)

Contention based protocols

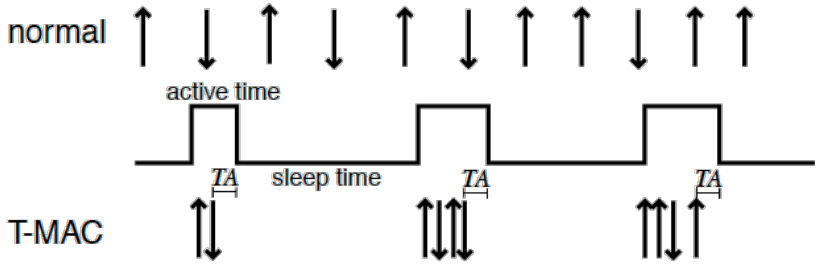
T-MAC

T-MAC

- S-MAC is based on static frame length



- T-MAC improves energy efficiency by using an adaptive duty cycle
  - Node goes in sleep mode after TA sec idle time (no traffic)



T-MAC

- T-MAC is an improvement on S-MAC
- All traffic is burst at the begin of an listen interval
- How to determine idle time TA?

$$TA > C + R + T$$

- C Length of contention interval
- R Time of RTS packet transfer
- T Turnaround time between RTS and CTS

Contention based protocols

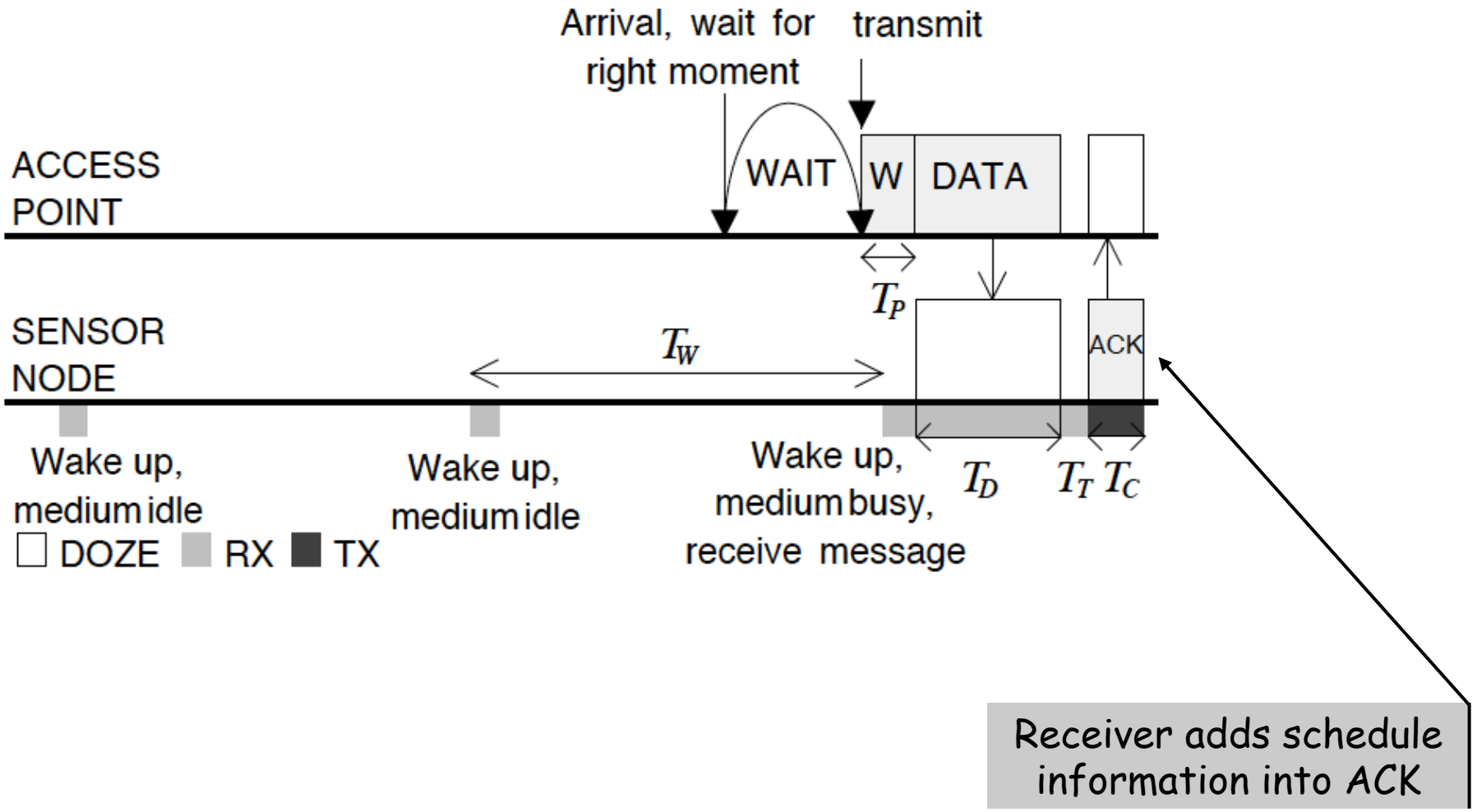
WiseMAC

WiseMAC

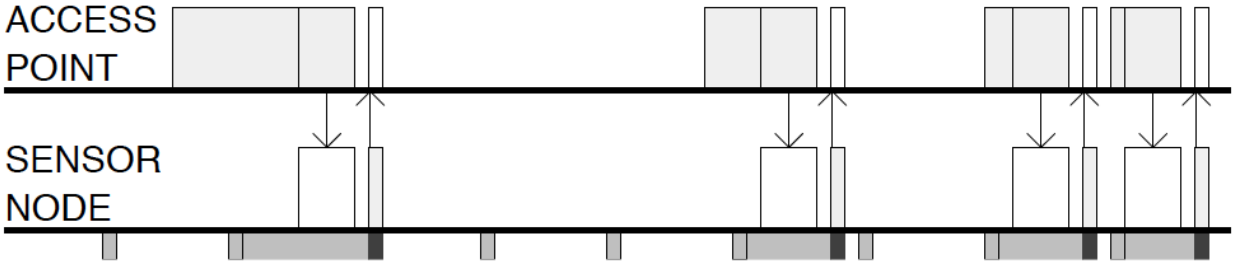
- WiseMAC ~ An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks

- Another problem with basic preamble sampling
  - Sender has to wait for the preamble and the wakeup of receiver
  - Sender wastes resources if receiver sleeps during preamble transmission
- Improvement
  - Sender transmits preamble just when receiver wakes up
  - Problem: How to know when receiver wakes up?
  - Approach: Sender learns the schedule of the receiver

WiseMAC


Arrival, wait for right moment
transmit
ACCESS POINT
WAIT
W
DATA
T_P
SENSOR NODE
T_W
ACK
Wake up, medium idle
Wake up, medium idle
Wake up, medium busy, receive message
T_D
T_T T_C
DOZE
RX
TX
Receiver adds schedule information into ACK

WiseMAC

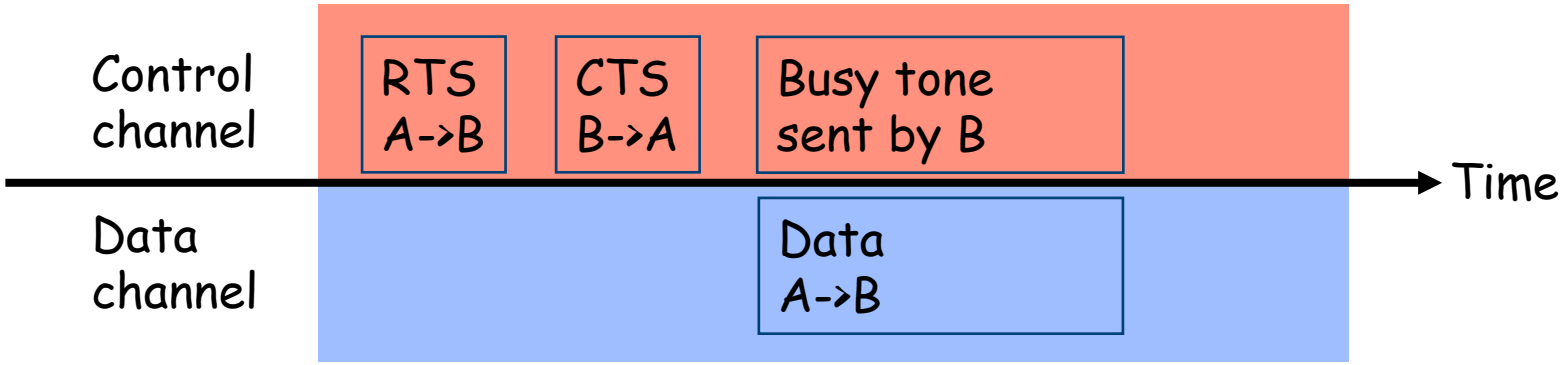


Contention based protocols

PAMAS

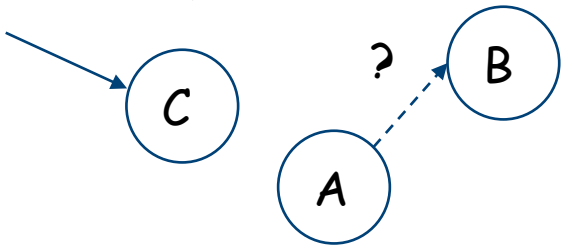
PAMAS: Power Aware Multiaccess with Signaling

- Idea: combine busy tone with RTS/CTS
  - Results in detailed overhearing avoidance, does not address idle listening
  - Uses separate ***data*** and ***control channels***
- Procedure
  - Node A transmits RTS on control channel, does not sense channel
  - Node B receives RTS, sends CTS on control channel if it can receive and does not know about ongoing transmissions
  - B sends busy tone as it starts to receive data

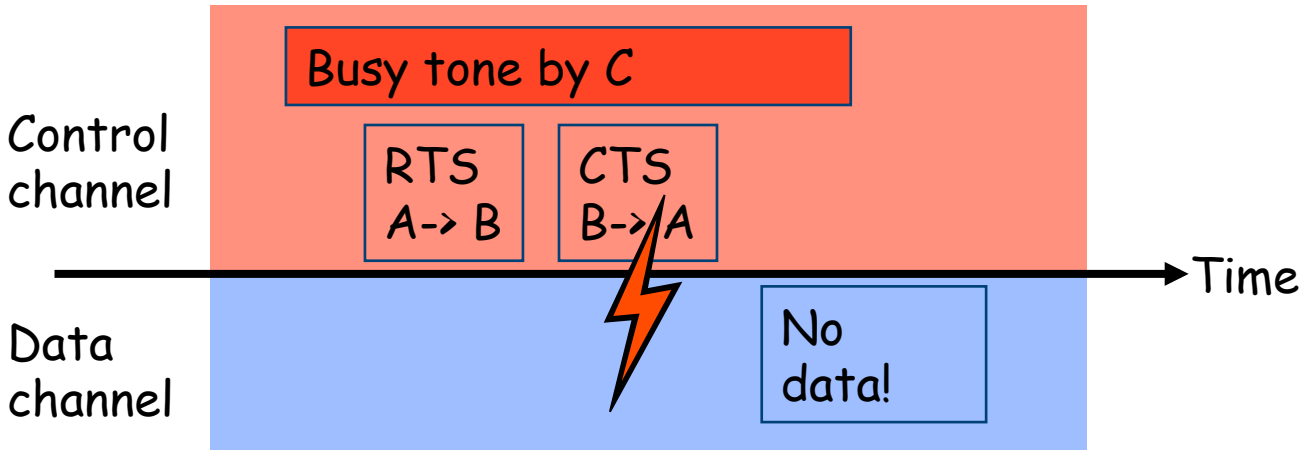


PAMAS: Already ongoing transmission

- Suppose a node C in vicinity of A is already receiving a packet when A initiates RTS



- Procedure
  - A sends RTS to B
  - C is sending busy tone (as it receives data)
  - CTS and busy tone collide, A receives no CTS, does not send data



Similarly: Ongoing transmission near B destroys RTS by busy tone

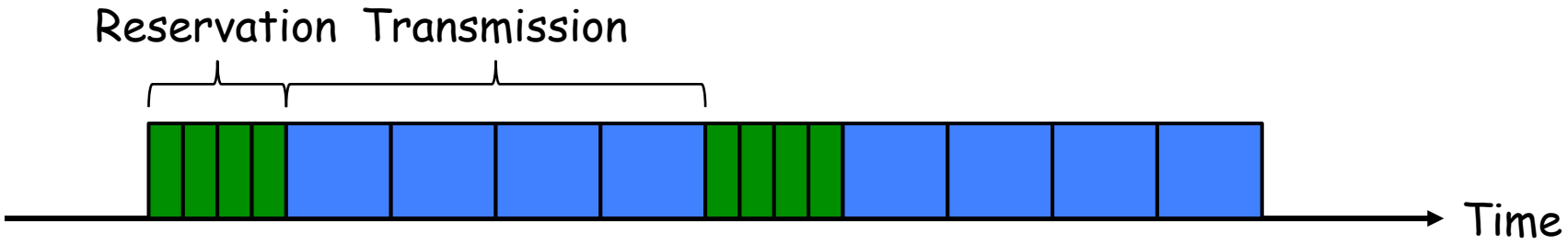
Reservation based protocols

TRAMA

Reservation based protocols

- Reservation based
  - First reserve a transmission slot
  - Then transmit

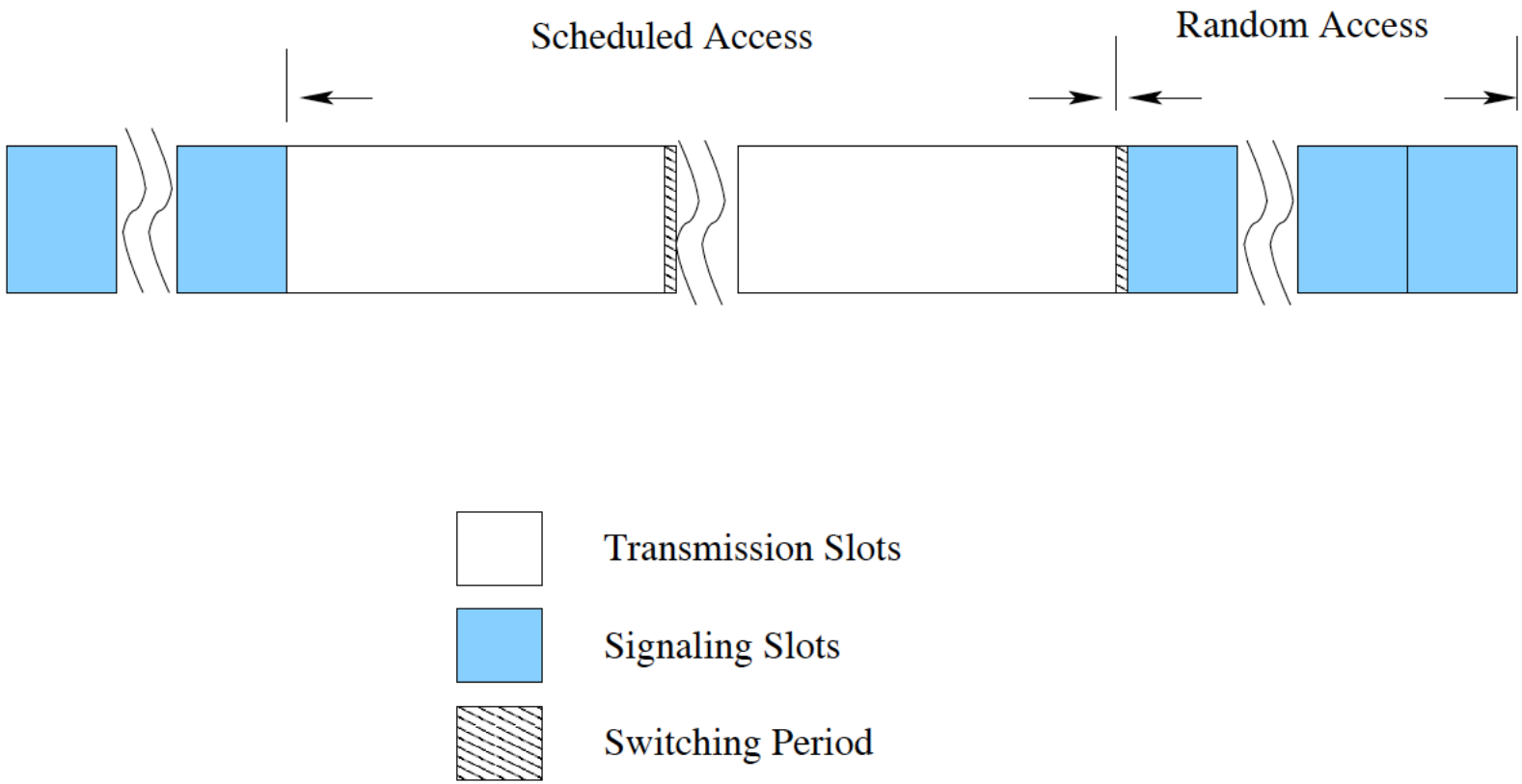
} Collision free



  - Traffic Adaptive Medium Access (TRAMA)

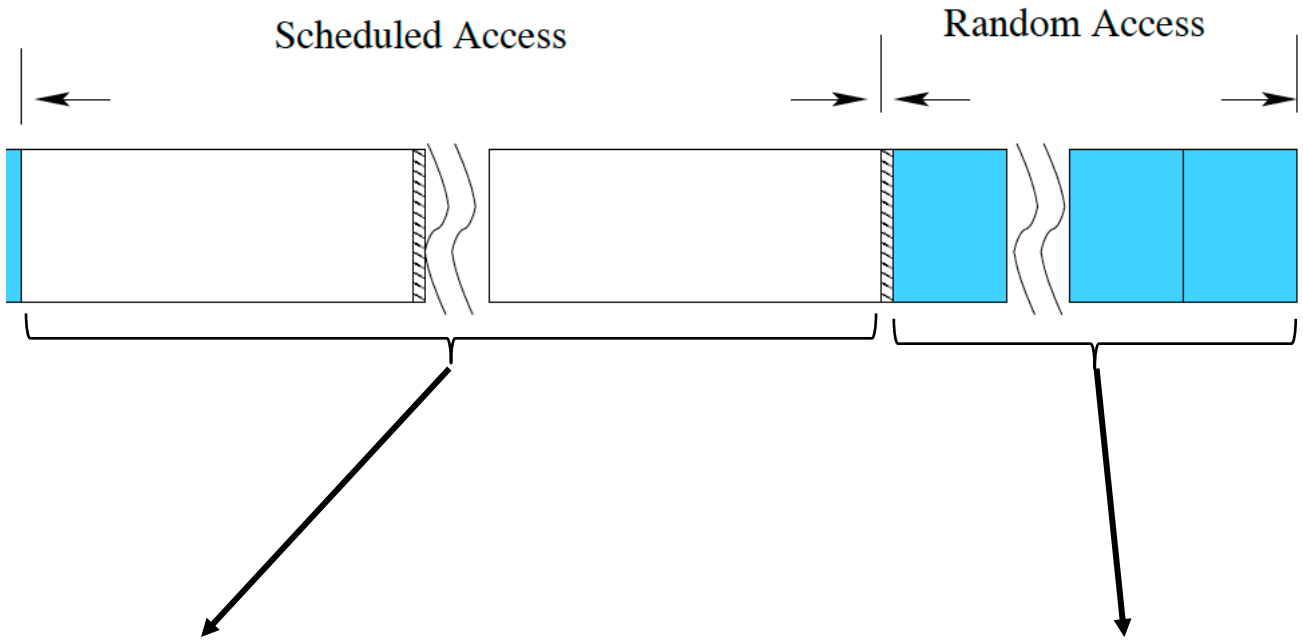
TRAMA

- A time-slotted structure



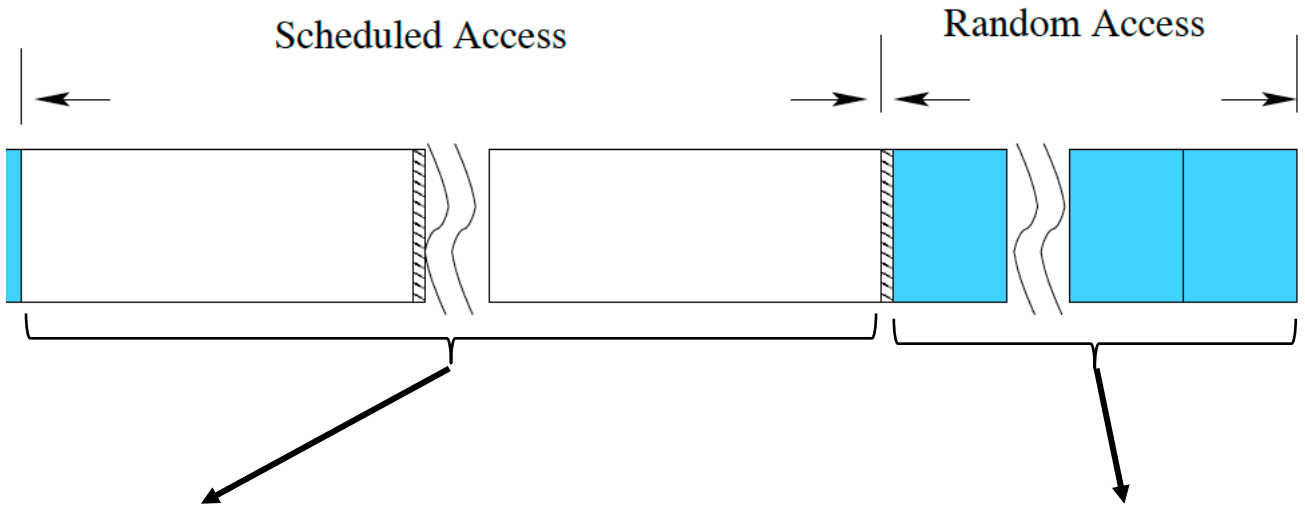
- 115.2 kbps -> transmission slot 46ms (512-byte segments)

TRAMA



- Scheduled access period
  - Used for contention free data transmission
  - Supports unicast, multicast, and broadcast communication
- Random access period
  - Used for signalling
  - Synchronization and updating of two-hop neighbourhood information
  - Collision!!

TRAMA



- Transmission slots:
  - For collision free data exchange
  - For schedule propagation
- Nodes use SEP (Schedule Exchange Protocol) to
  - exchange traffic based information or schedules with neighbors
- Signaling slots:
  - are used by the Neighbor protocol (NP)
  - Propagate one-hop neighbor information among neighboring nodes during the random access period
- A consistent two-hop topology information across all nodes is obtained

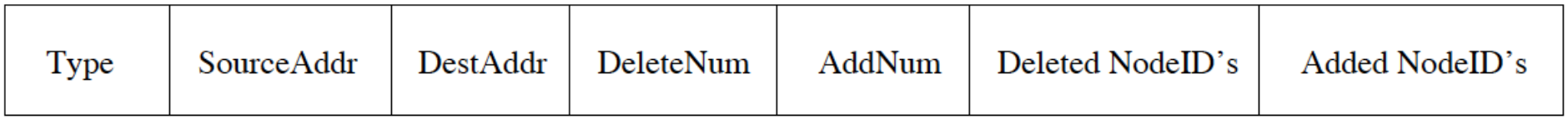
TRAMA Components

- Neighbor Protocol (NP)
  - Gather 2-hop neighborhood information
- Schedule Exchange Protocol (SEP)
  - Gather 1-hop traffic information for SCHEDULING
- Adaptive Election Algorithm (AEA)
  - Select transmitters, receivers for current time slot leave other nodes in liberty to switch to low power mode using the NP and SEP results

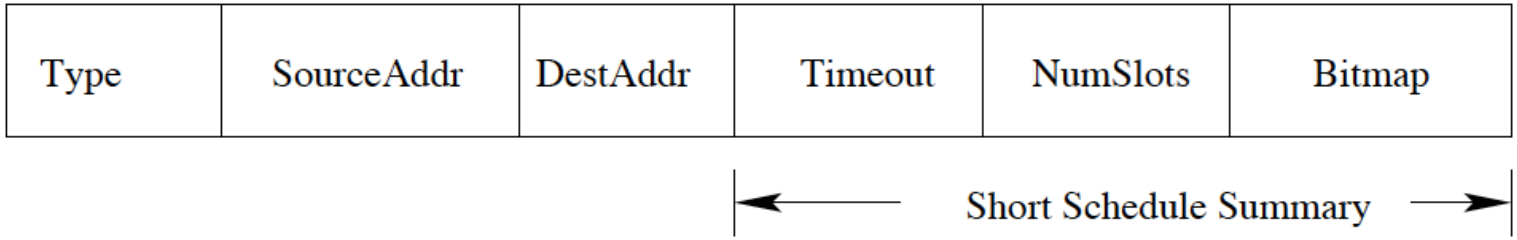
Neighbor Protocol (NP)

- Main Function:
  - Gather two-hop neighborhood information by using signaling packets during the random access period
- If no updates, signaling packets are sent as "keep alive" beacons
- A node times out if nothing is heard from its neighbor
- Updates retransmitted to guarantee packet delivery

Packet Formats



(a) Signal Header



(b) Data Header

Schedule Exchange Protocol (SEP)

- Each node computes a SCHEDULE INTERVAL (SCHED) based on the rate at which packets are produced.
- SCHED represents number of slots for which the node can announce the schedule to its neighbors according to its current state (queue)
- The node pre-computes number of slots in the interval

$$[t, t+SCHED]$$

for which it has the highest priority among its two-hop neighbors (contenders) → WINNING SLOTS

Schedule Exchange Protocol (SEP)

- The node announces the intended receivers for these slots
- The last winning slot is used for broadcasting the node's schedule for the next interval
- If these winning slots cannot be filled by the node the remaining vacant slots can be released to other nodes

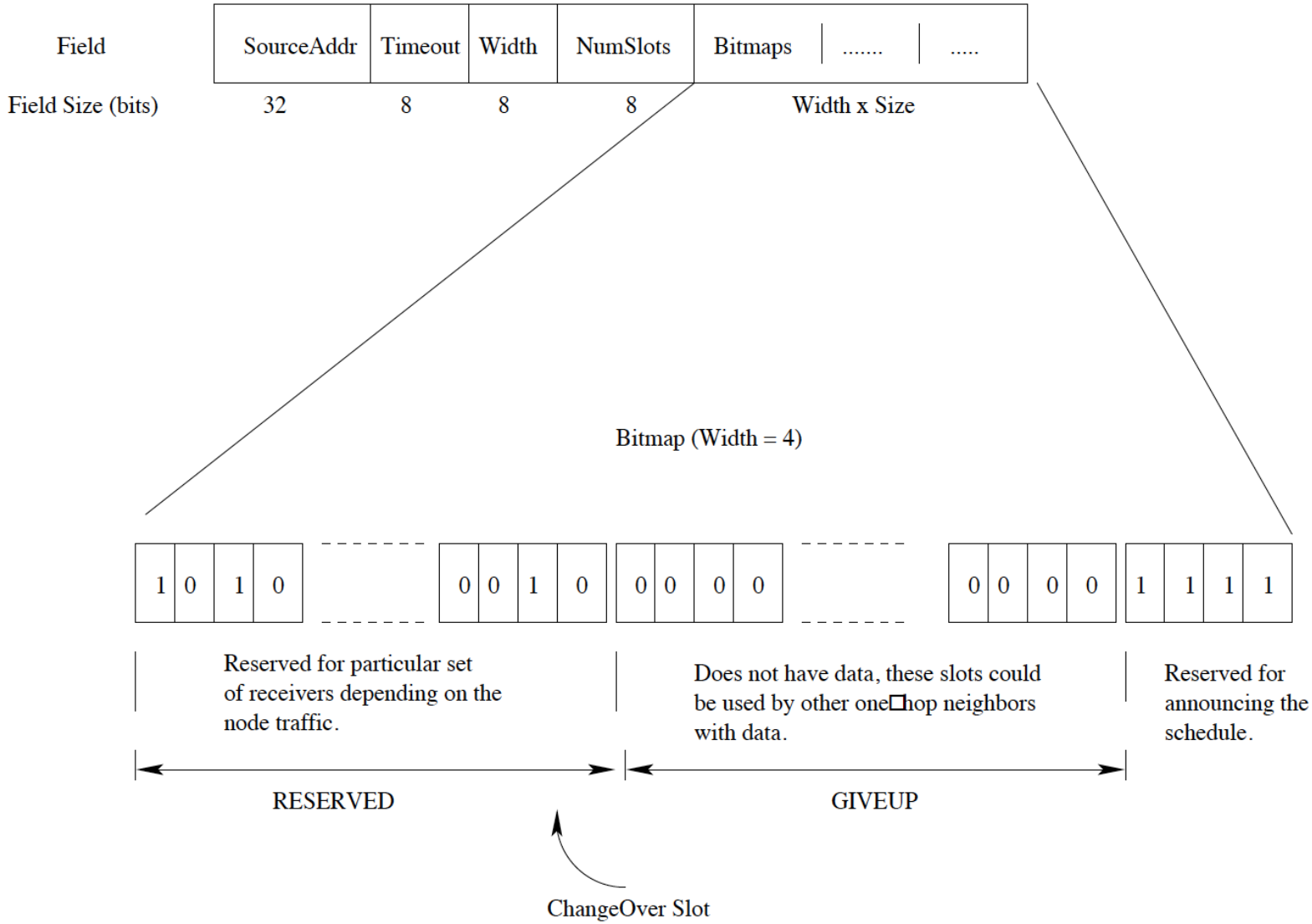
- Example: Node u -> SCHED is 100 slots.
  - During time slot 1000, u computes its winning slots between [1000,1100]
  - Assume: These slots are 1009, 1030, 1033, 1064, 1075, 1098
  - u uses slot 1098 to announce its next schedule by looking ahead from [1098,1198]

Schedule Exchange Protocol (SEP)

- Nodes announce their schedules via SCHEDULE PACKETS
- BITMAP: with the length equal to number of one-hop neighbors
- Each bit corresponds to one particular receiver

- Example: One node with 4 neighbors 14,7,5, and 4
  - BITMAP -> size 4
  - For broadcast: all bitmap bits are set to 1

Schedule Packet Format



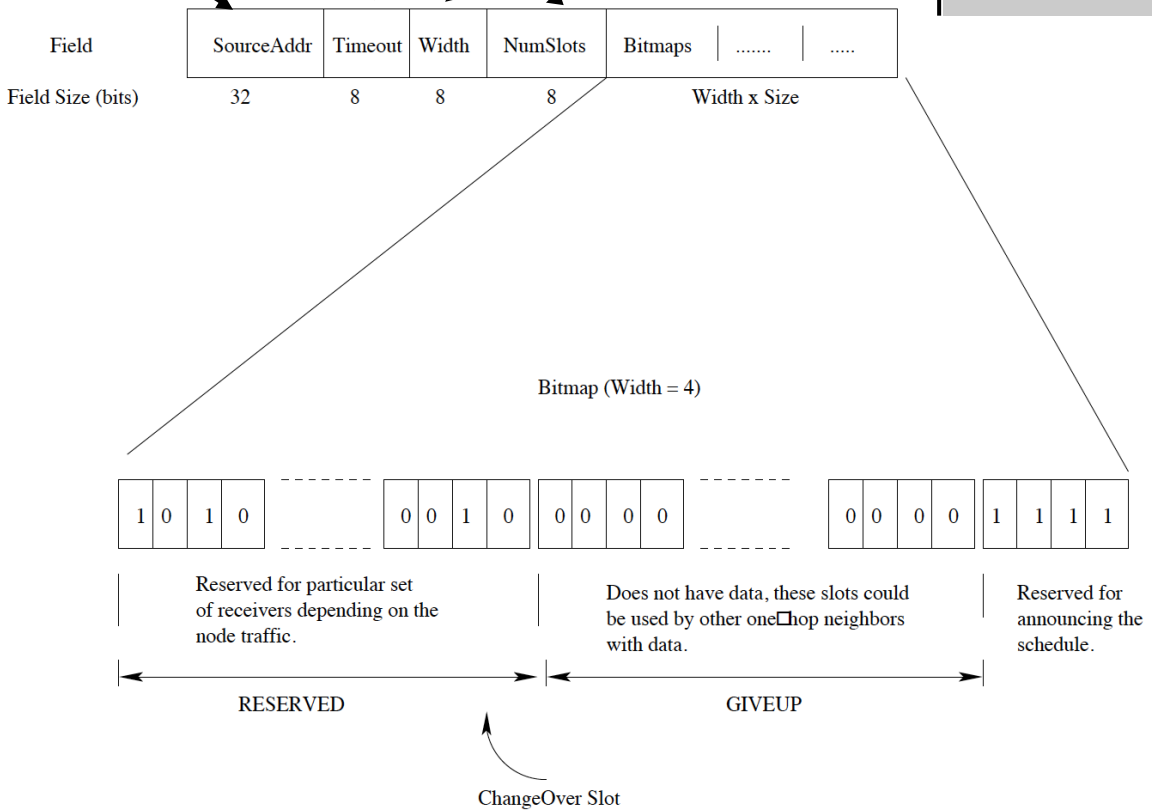
Schedule Packet Format

Number of slots for which the schedule is valid

Length of the neighbor bitmap (number of one hop neighbors)

Node announcing the schedule

Total number of winning slots (number of bitmaps contained in the packet)



Adaptive Election

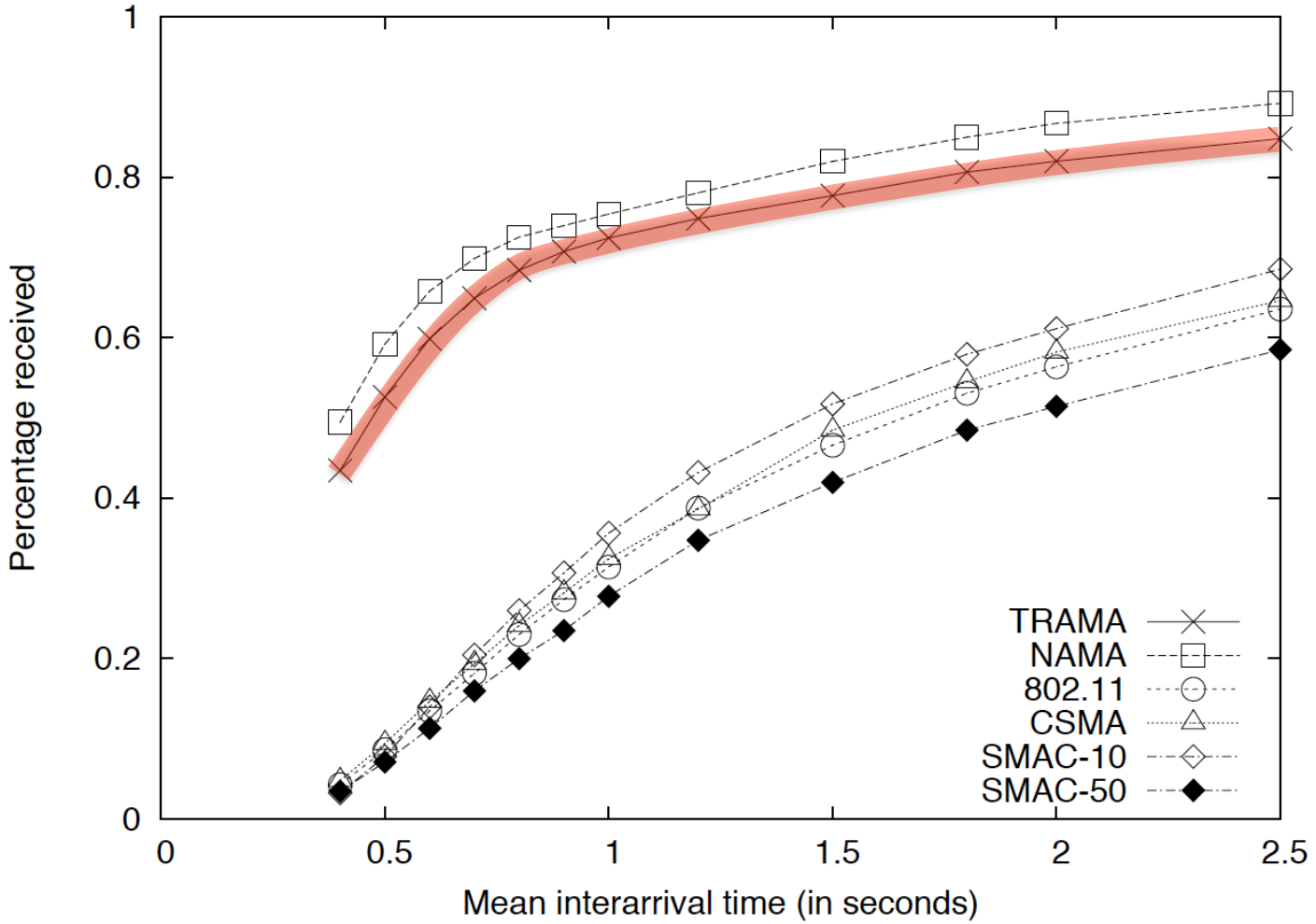
- Given: Each node knows its two-hop neighborhood and their current schedules
- How to decide which slot (in scheduled access period) a node can use?
  - Use node identifier u and globally known hash function h
  - For time slot t, compute priority

$$prio(u,t) = h(u \oplus t)$$

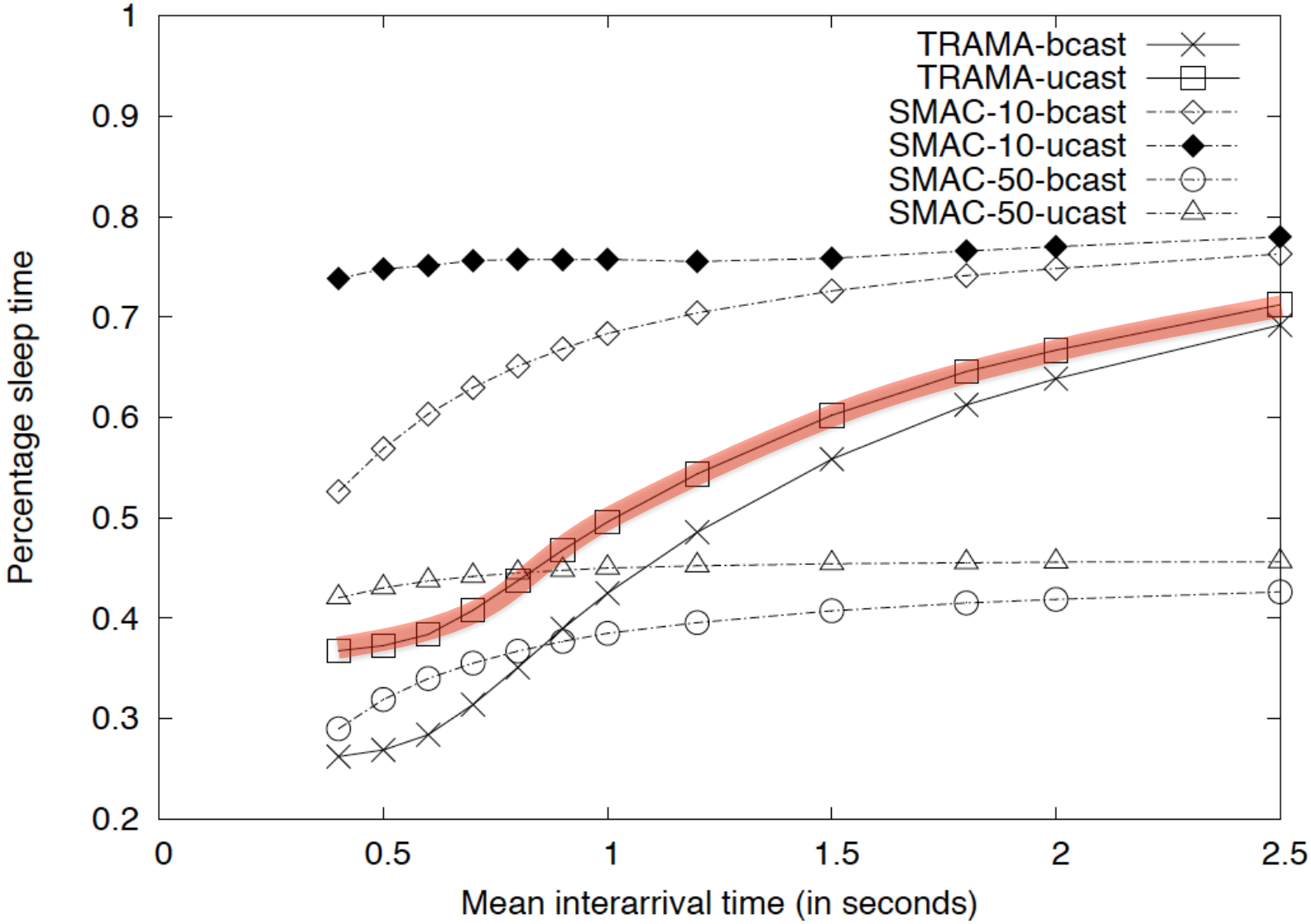
  - Compute this priority for next k time slots for node itself and all two-hop neighbors
  - Node uses these time slots for which it has the highest priority

Simulation Results

- Broadcast traffic using Poisson arrivals
- 50 nodes, 500mx500m area, 512 byte data, Average node density: 6



Energy Savings



TRAMA: Summary

- Complex election algorithm and data structure
- Overhead due to explicit schedule propagation
- Higher queueing delay

Hybrid protocols

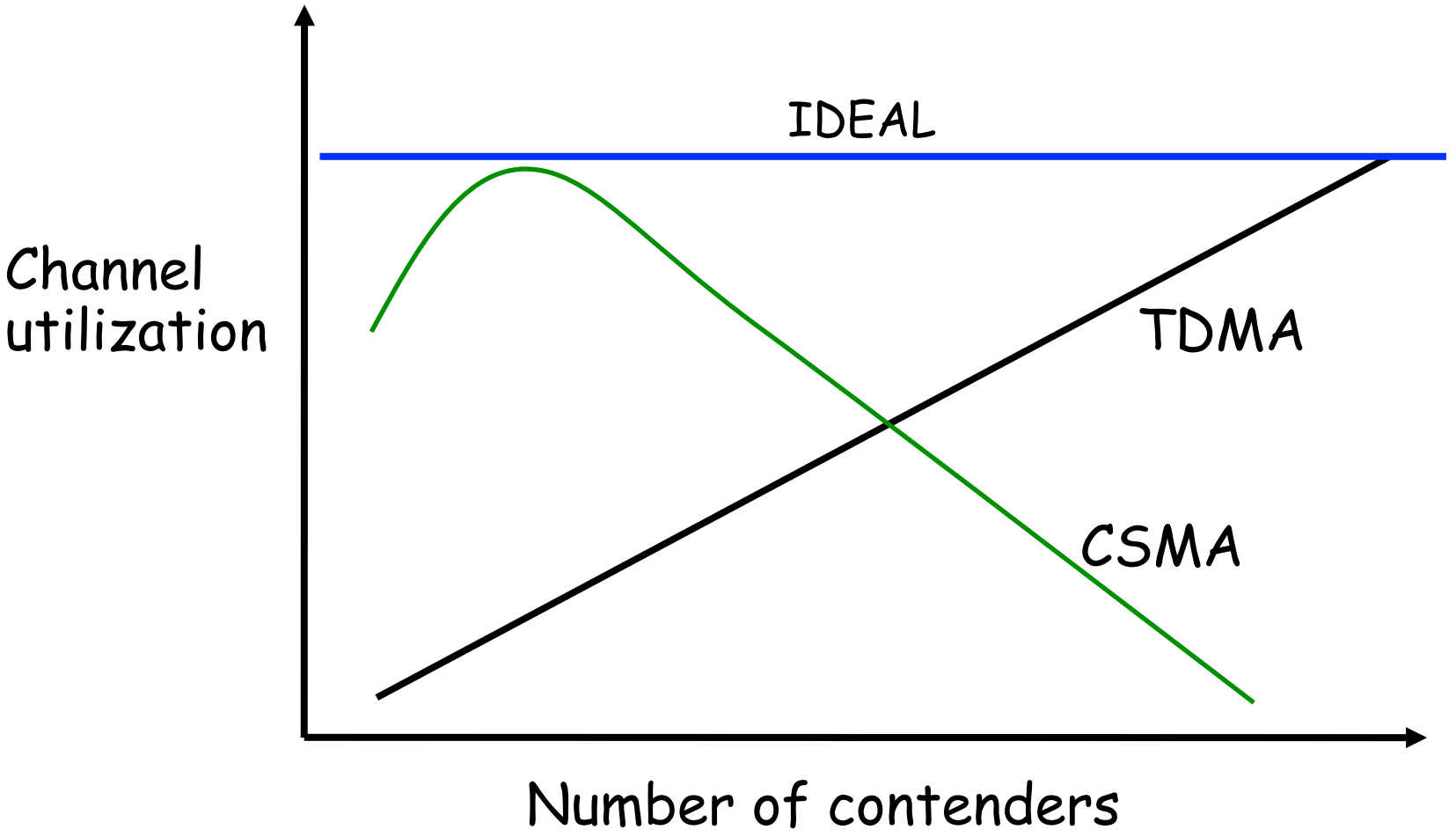
Z-MAC

Z(ebra)-MAC

- Combines the strengths of both CSMA and TDMA at the same time offsetting their weaknesses
- High channel efficiency and fair

	Channel Utilization	
	Low Contention	High Contention
CSMA	High	Low
TDMA	Low	High

Effective throughput: CSMA vs. TDMA



Z-MAC

- Uses TDMA schedule created by DRAND as a "hint" to schedule transmissions
  - Distributed RAND
- The owner of a time-slot always has priority over the non-owners while accessing the medium
- Unlike TDMA, non-owners can "steal" the time-slot when the owners do not have data to send

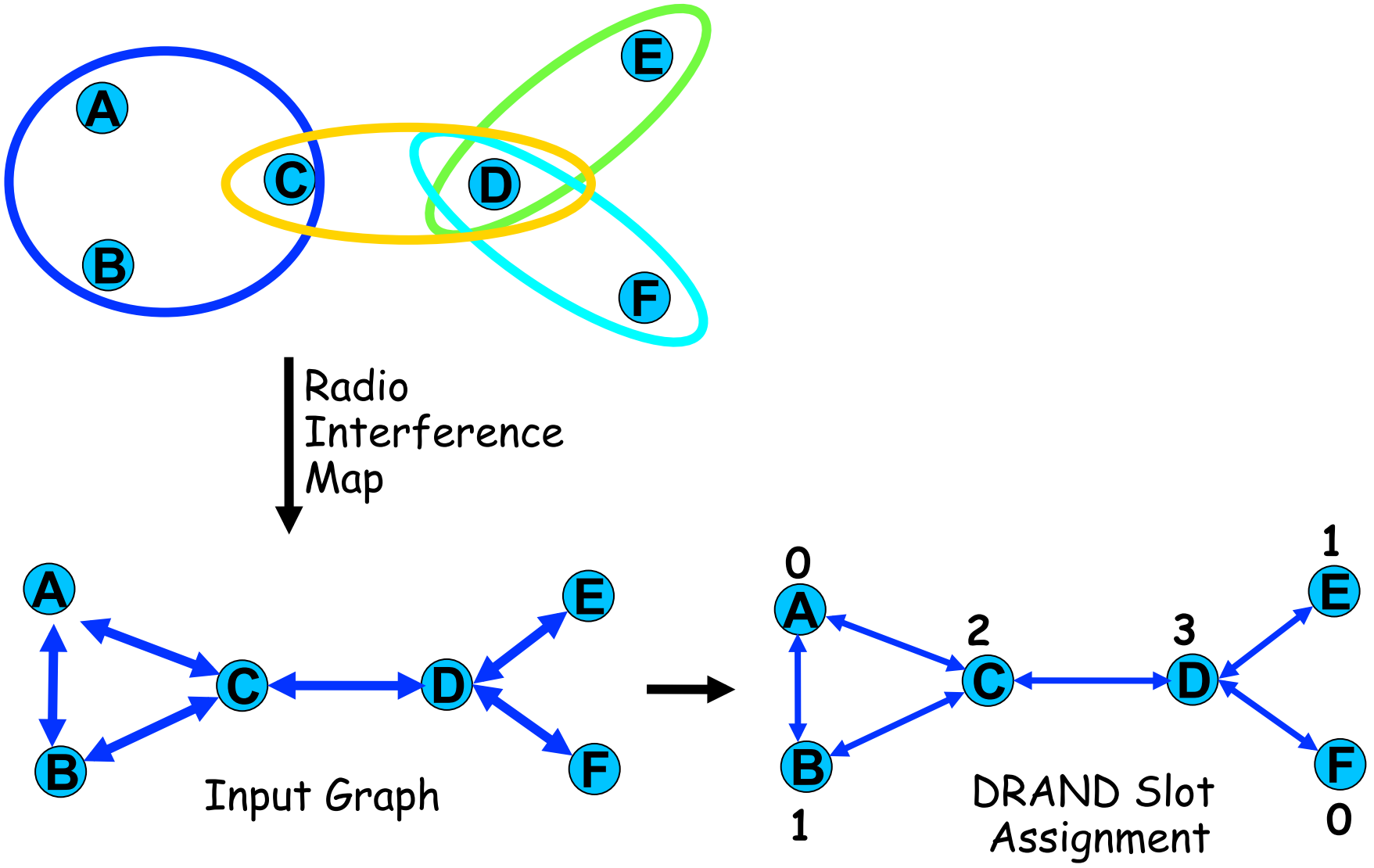
Z-MAC

- This enables Z-MAC to switch between CSMA and TDMA depending on the level of contention

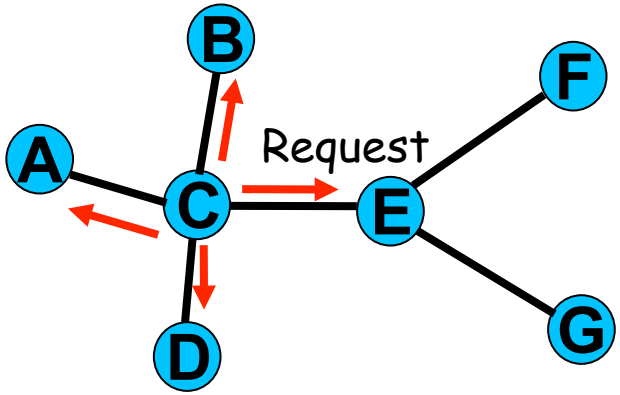
- Under low contention
  - Z-MAC acts like CSMA
  - i.e., high channel utilization and low latency

- Under high contention
  - Z-MAC acts like TDMA
  - i.e., high channel utilization, fairness, and low contention overhead

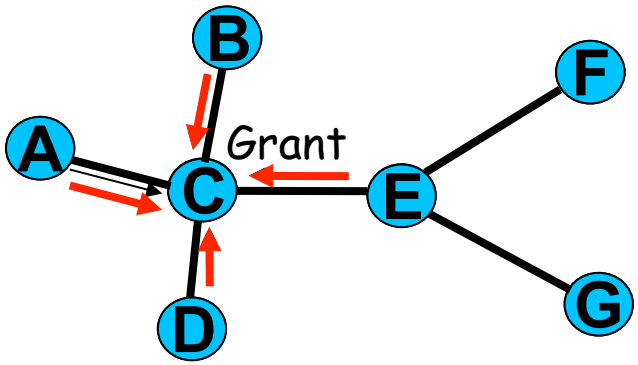
DRAND-Algorithm
Distributed TDMA Scheduling



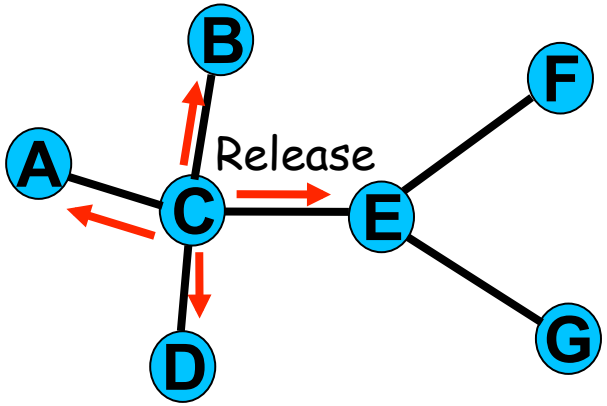
DRAND-Algorithm: Successful Round



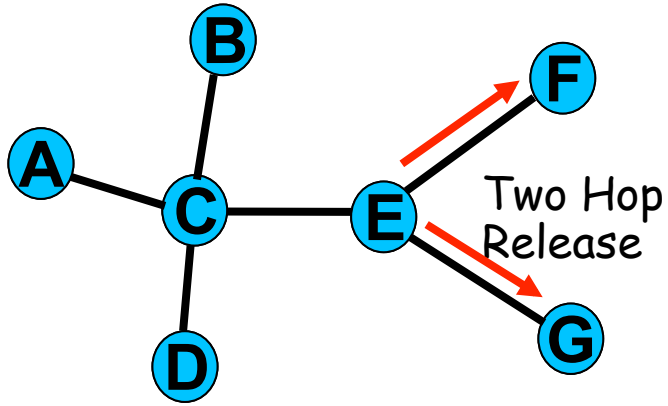
Step 1: Broadcast Request



Step 2: Receive Grants

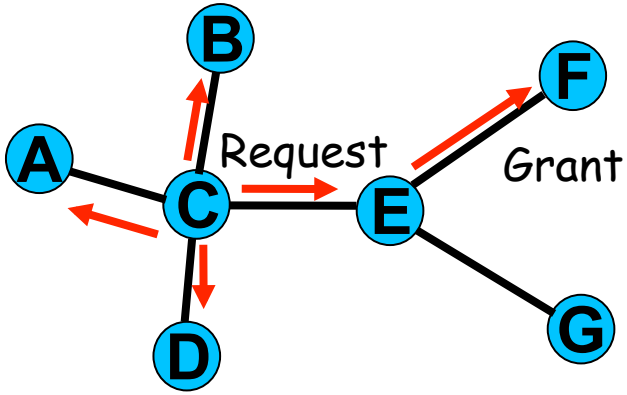


Step 3: Broadcast Release

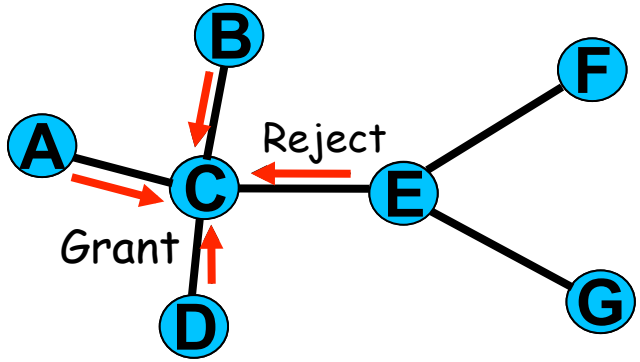


Step 4: Broadcast Two Hop Release

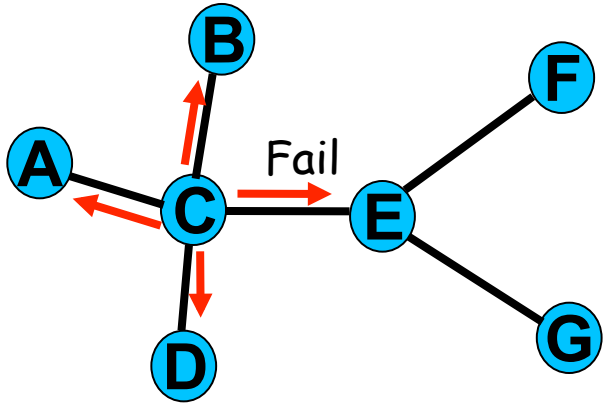
DRAND-Algorithm: Unsuccessful Round



Step 1: Broadcast Request



Step 2: Receive Grants from A,B,D but Reject from E



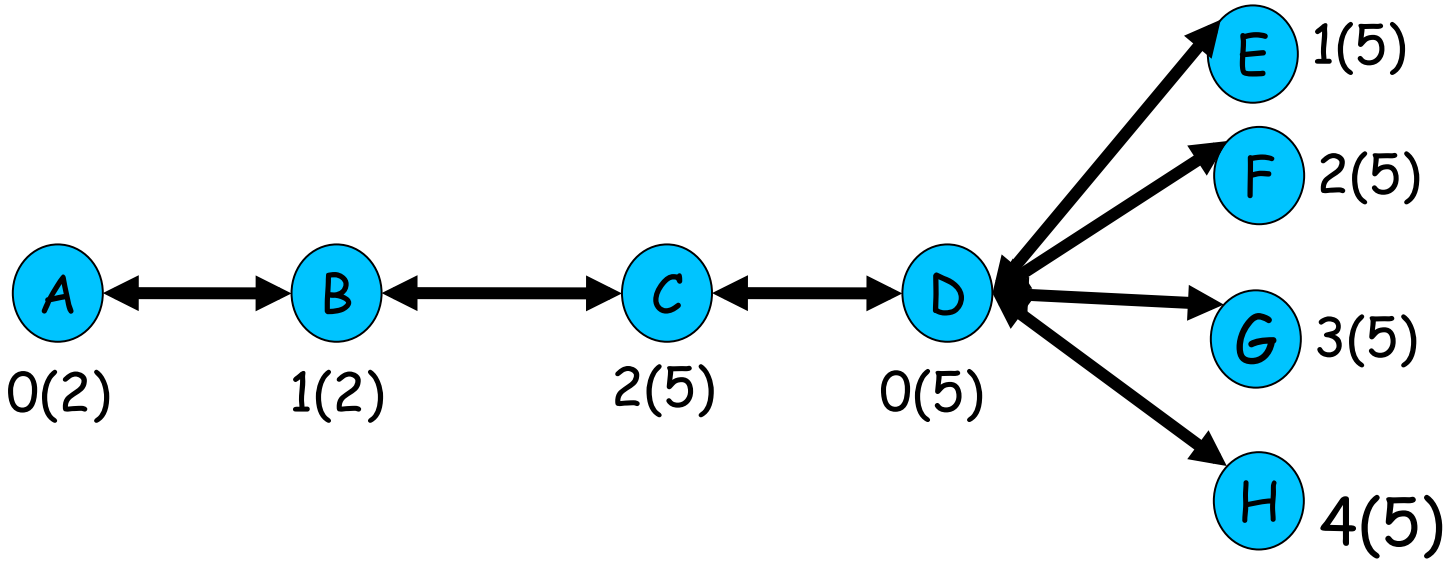
Step 3: Broadcast Fail

Local Frames

- After DRAND, each node needs to decide on the frame size
- Conventional wisdom: Synchronize with rest of the network on **Maximum Slot Number (MSN)** as the frame size

- Disadvantage:
  - MSN must be broadcast across the entire network.
  - Unused slots if neighborhood small, e.g., A and B would have to maintain frame size of 8, in spite of having small neighborhood

Local Frames: Example



Label is the assigned slot, number in parenthesis is maximum slot number within two hops

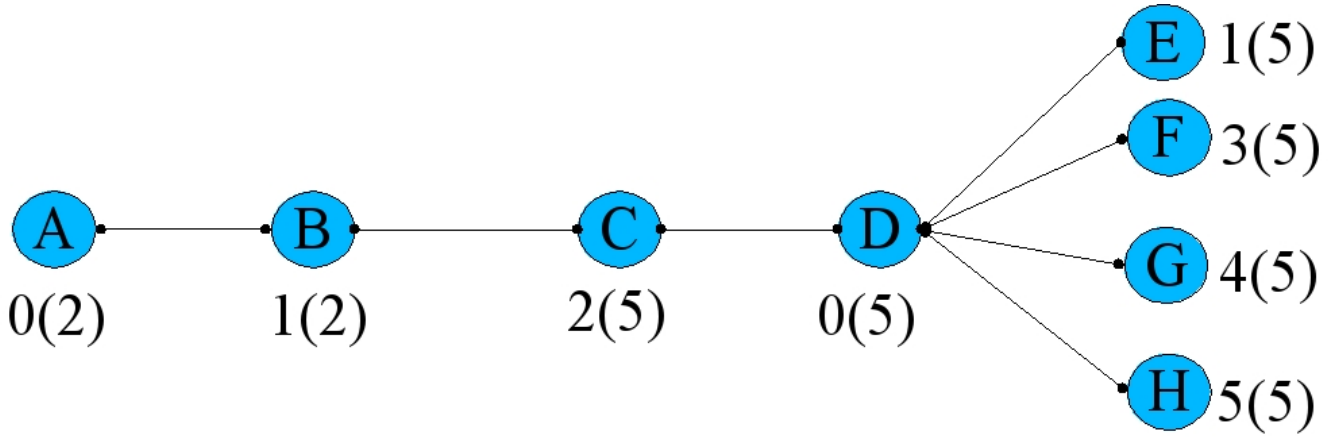
Local Frames

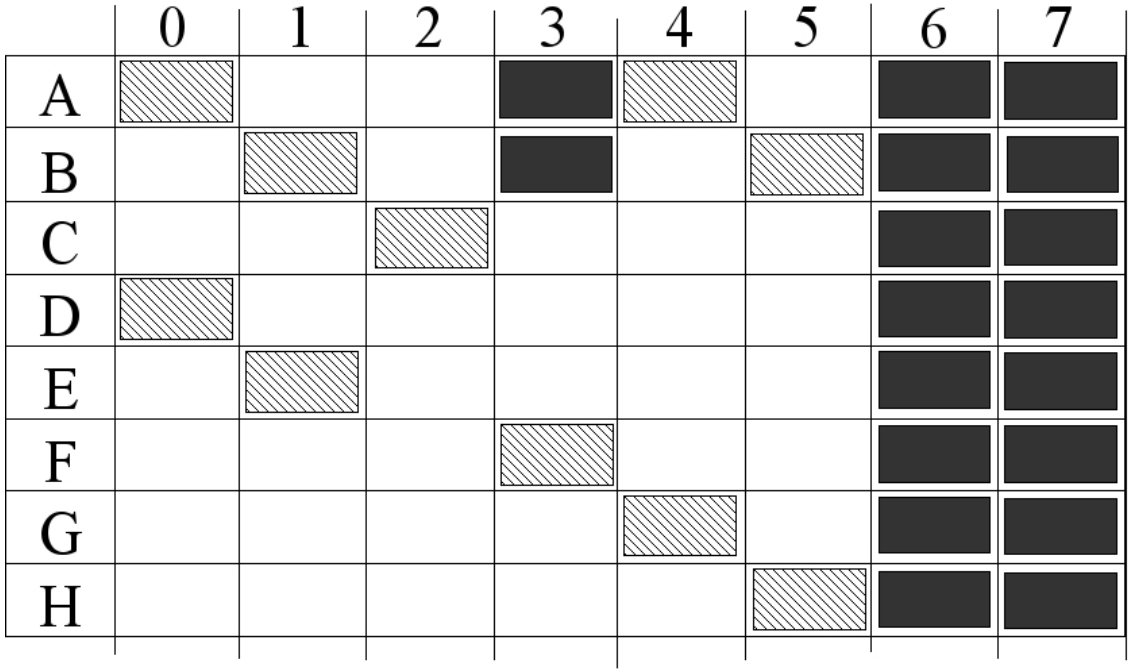
- Time Frame Rule (TF Rule)
  - Let node i be assigned to slot s_i, according to DRAND and MSN within two hop neighborhood be F_i, then i's time frame is set to be 2^a, where positive integer a is chosen to satisfy condition

$$2^{a-1} \leq F_i < 2^a - 1$$

  - In other words, i uses the s_i-th slot in every 2^a time frame
    - i's slots are $L \cdot 2^a + s_i$, for all L=1,2,3,…

Local Frames: Example

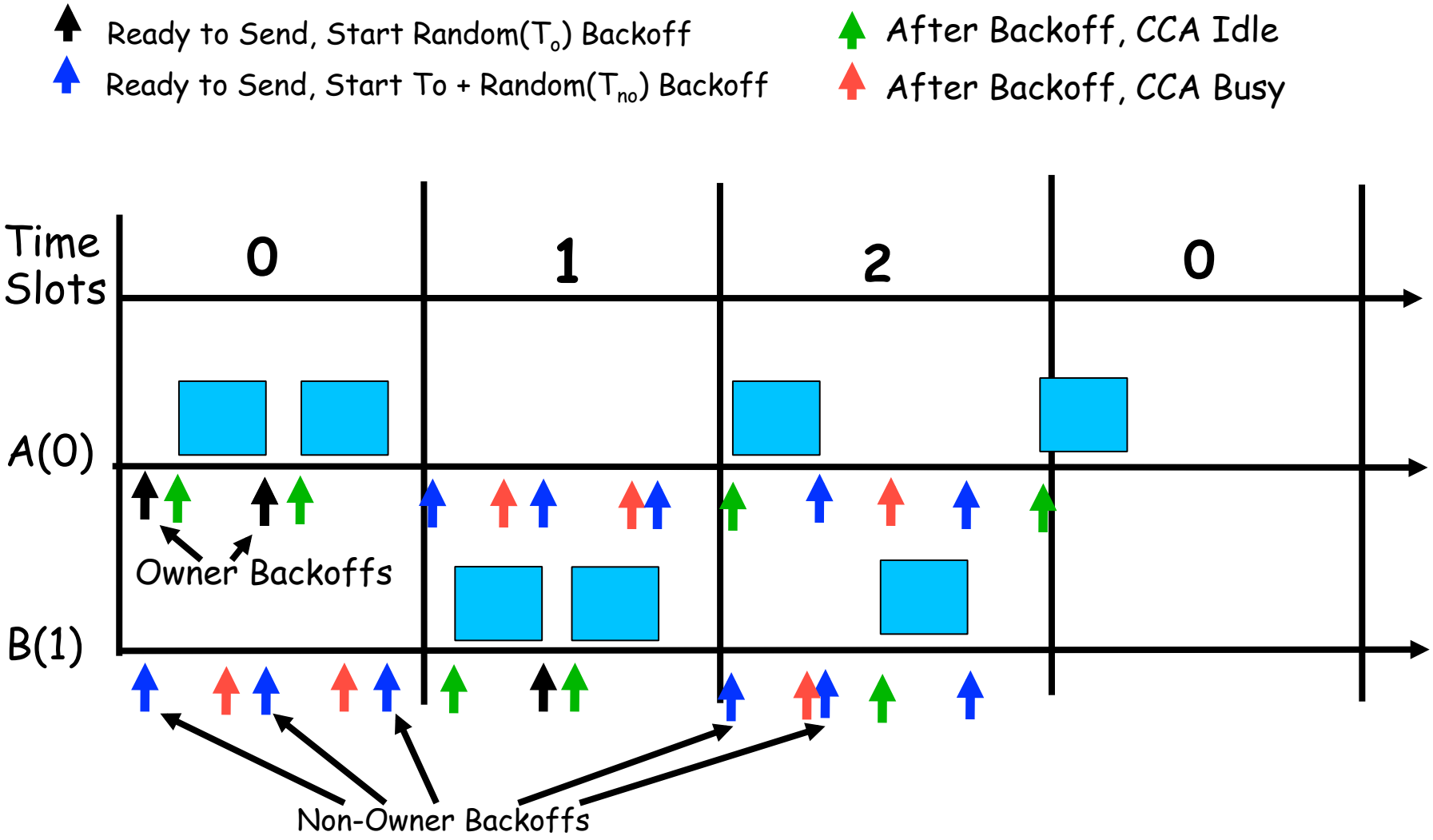




Transmission Control

- Slot Ownership
  - If current timeslot is the node's assigned time-slot, then it is the **Owner**, and all other neighbour nodes are **Non-Owners**
- Low Contention Level: Nodes compete in all slots, albeit with different priorities. Before transmitting:
  - Owner: Backoff = Random(T_o)
  - Non-Owner: Backoff = T_o + Random(T_{no})
  - After backoff, sense channel, if busy repeat above, else send.
- Switches between CSMA and TDMA automatically depending on contention level
- Performance depends on specific values of T_o and T_{no}
  - From analysis, T_o = 8 and T_{no} = 32 are used for best performance

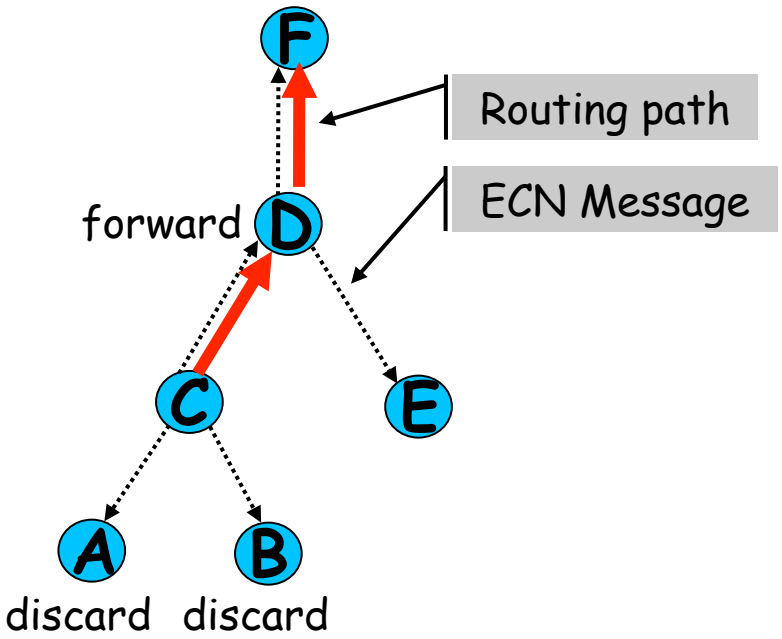
Transmission Control



Explicit Contention Notification (ECN)

- Informs all nodes within two-hop neighborhood not to send during its time-slot
- When a node receives ECN message, it sets its HCL flag
- ECN is sent by a node if it experiences high contention
- High contention detected by lost ACKs or congestion backoffs.
- On receiving one-hop ECN from i, forward two-hop ECN if it is on the routing path from i.

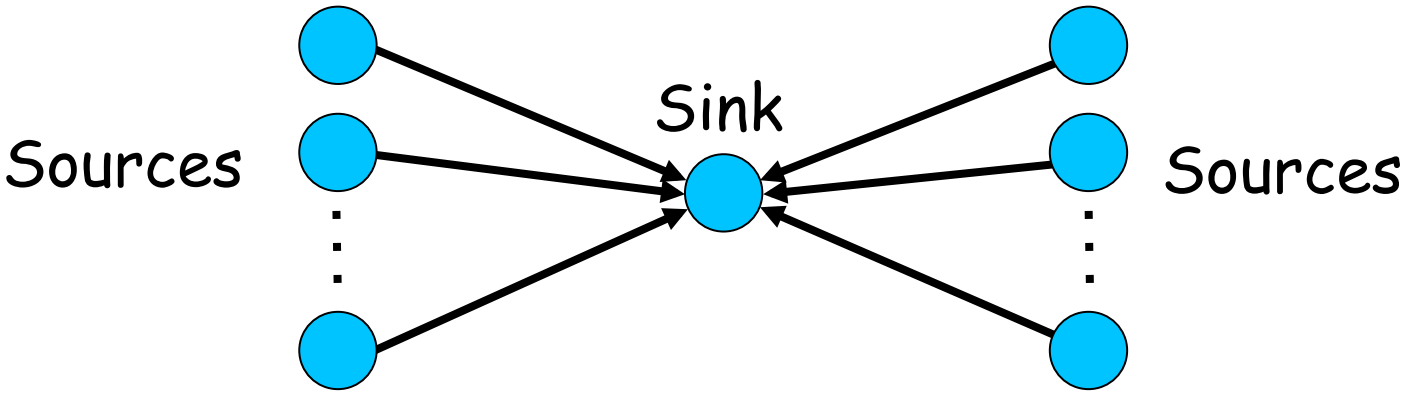
Explicit Contention Notification



- C experiences high contention
- C broadcasts one-hop ECN message to A, B, D.
- A, B not on routing path, so discard ECN.
- D on routing path, so it forwards ECN as two-hop ECN message to E, F.
- Now, E and F will not compete during C's slot as Non-Owners.
- A, B and D are eligible to compete during C's slot, albeit with lesser priority as Non-Owners.

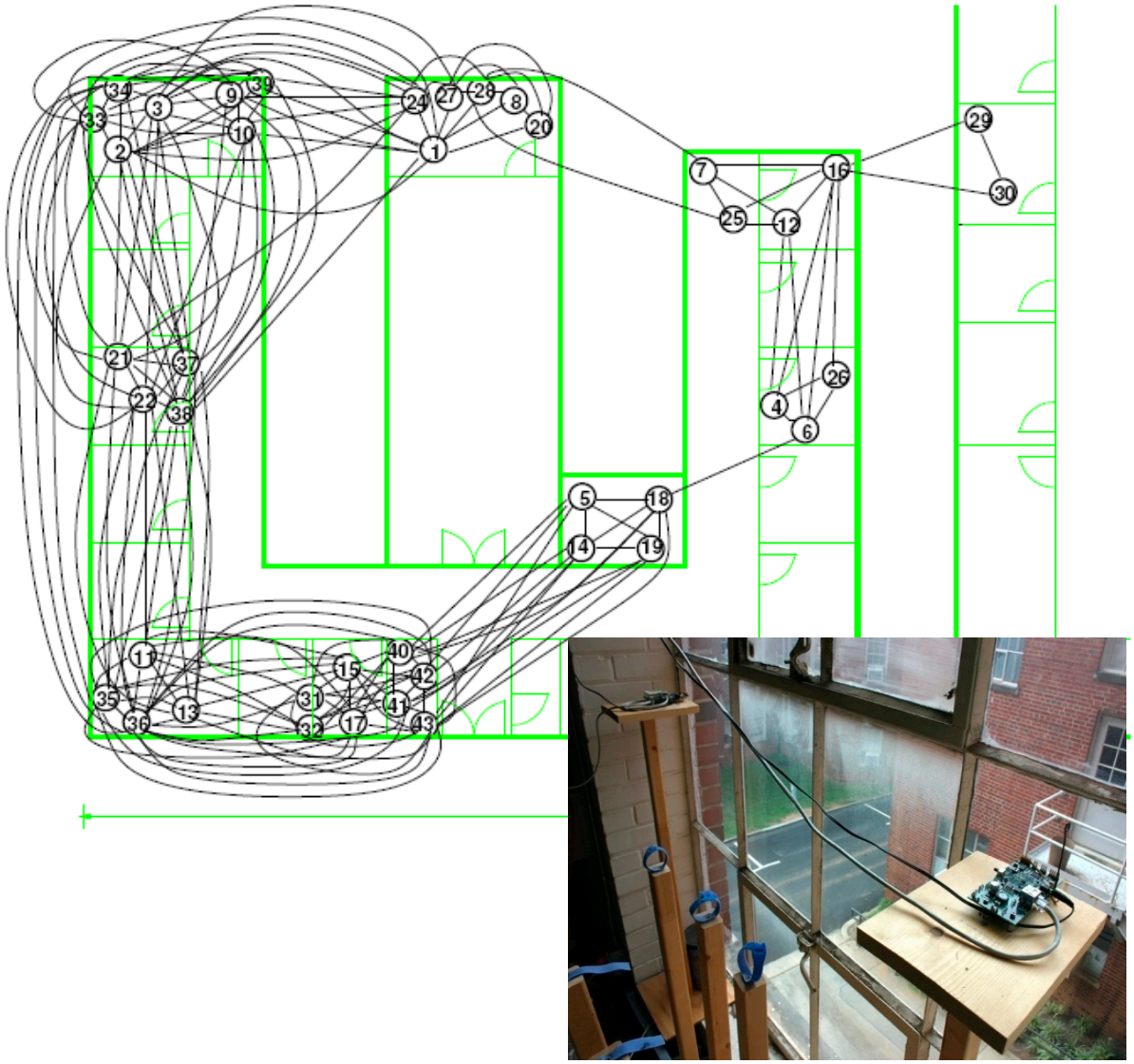
Two-Hop Experiments

- Setup: Two-Hop
  - Dumbbell shaped topology
  - Transmission power varied between low (50) and high (150) to get two-hop situations.
  - Aim – See how Z-MAC works when Hidden Terminal Problem manifests itself.

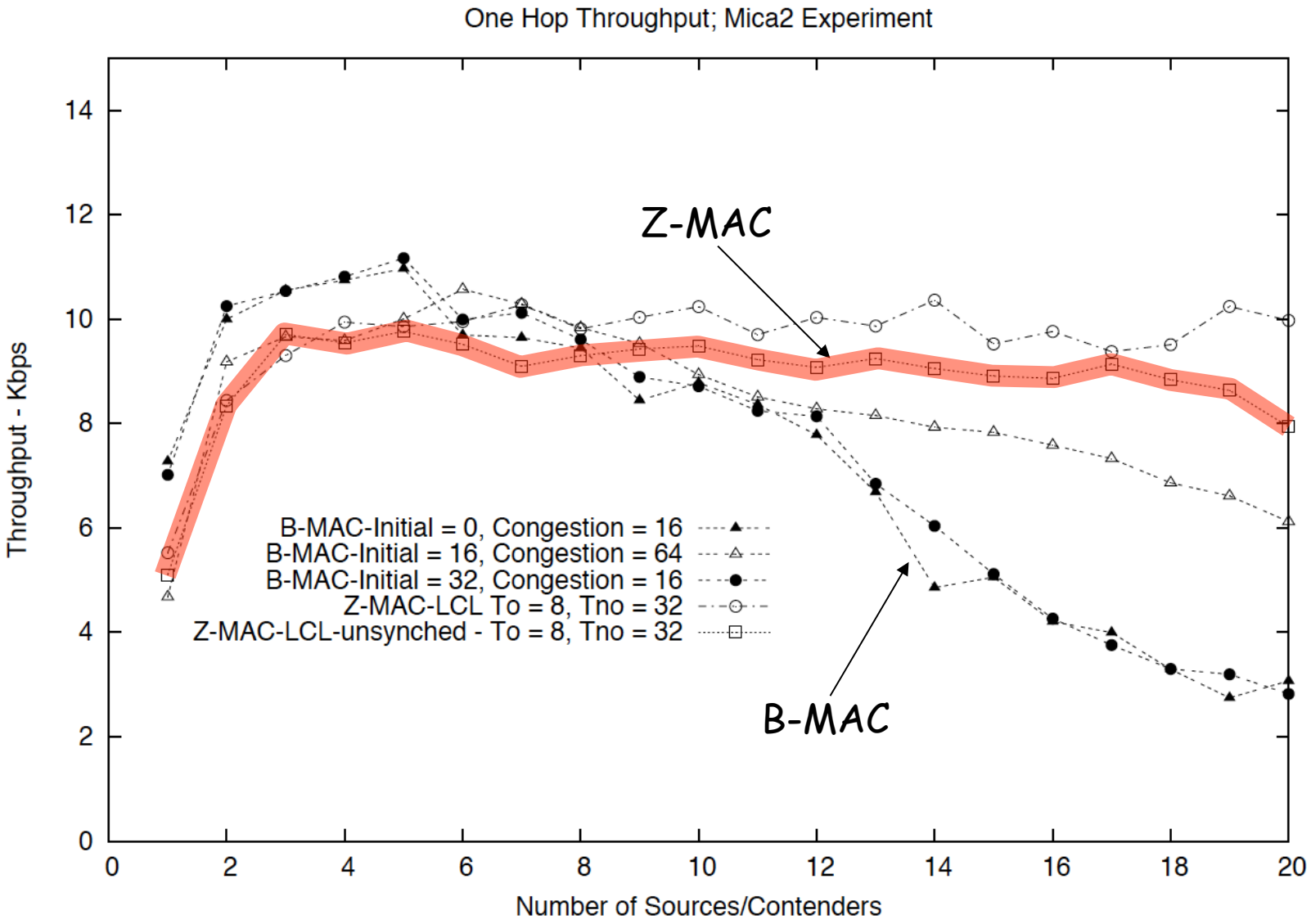


Experimental Setup - Testbed

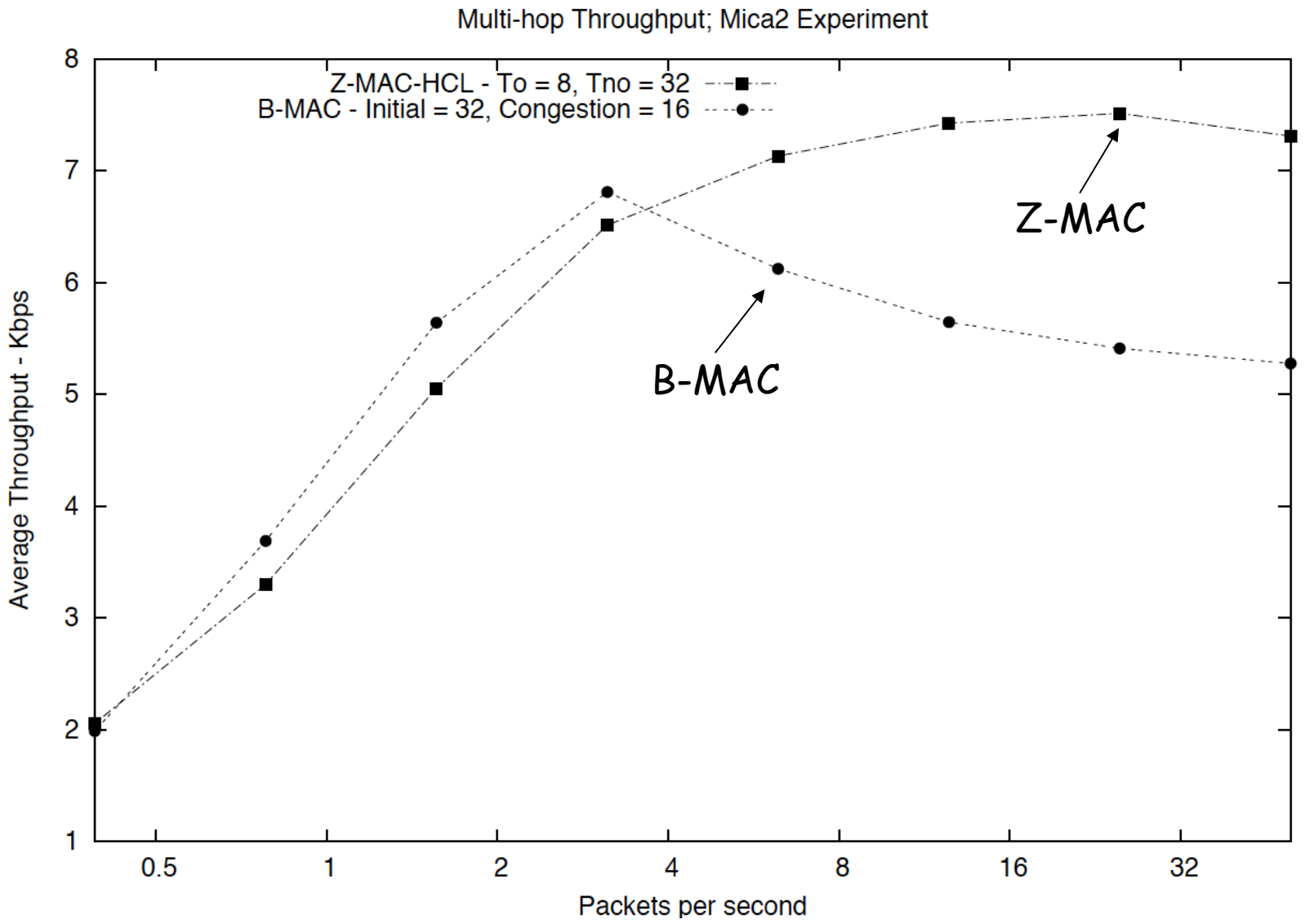
- 40 Mica2 sensor motes
- Wall-powered and connected to the Internet via Ethernet ports
- Programs uploaded via the Internet, all mote interaction via wireless
- Links vary in quality, some have loss rates up to 30-40%
- Asymmetric links also present (14->15)



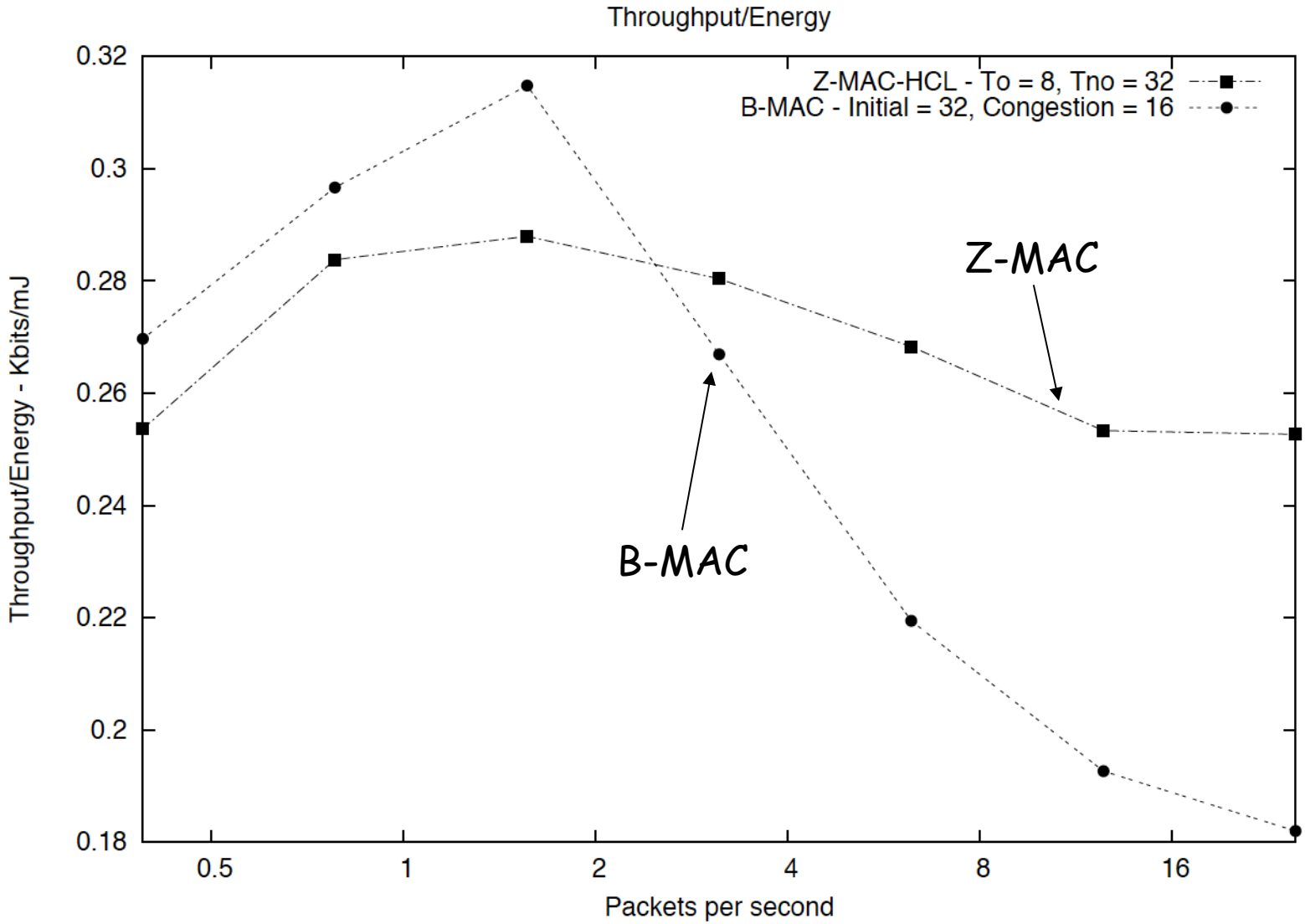
Single-Hop Results: Throughput



Multi Hop Results: Throughput



Multi Hop Results: Energy Efficiency



Z-MAC: Summary

- Hybrid protocol
  - Improved throughput in high traffic load
  - Acceptable latency in low traffic

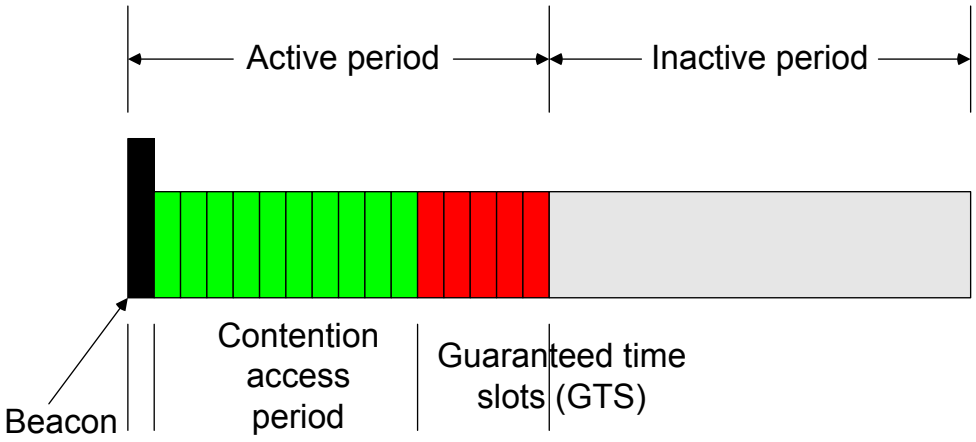
IEEE 802.15.4

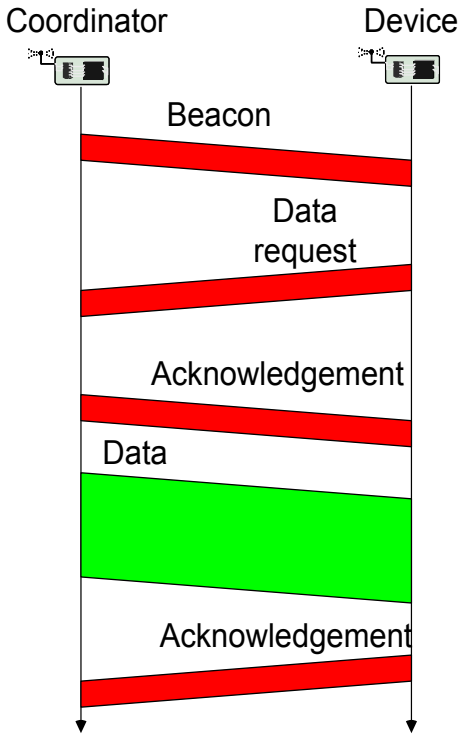
IEEE 802.15.4

- IEEE standard for low-rate WPAN applications
- Goals: low-to-medium bit rates, moderate delays without too stringent guarantee requirements, low energy consumption
- Physical layer
  - 20 kbps over 1 channel @ 868-868.6 MHz
  - 40 kbps over 10 channels @ 905 – 928 MHz
  - 250 kbps over 16 channels @ 2.4 GHz
- MAC protocol
  - Single channel at any one time
  - Combines contention-based and schedule-based schemes
  - Asymmetric: nodes can assume different roles

IEEE 802.15.4 MAC overview

- Star networks: ***devices*** are associated with ***coordinators***
  - Forming a PAN, identified by a PAN identifier
- Coordinator
  - Bookkeeping of devices, address assignment, generate beacons
  - Talks to devices and peer coordinators
- Beacon-mode superframe structure
  - GTS assigned to devices upon request





Summary

Summary

- Plenty of MAC protocols

- Discussed
  - Contention-based protocols
  - Reservation-based protocols
  - Hybrid protocols

- Open issues
  - Mobility support
  - Real-time communication
  - Which MAC suites best an application?
  - How to compare different MAC protocols?
  - Influence of MAC to higher layers!

Literature

[Ye] W. Ye, et. al., "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Trans. on Networking, June 2004.

[Polastre] J. Polastre, J. Hill, D. Culler, "Versatile Low Power Media Access for WSNs", Proc. of ACM SenSys, Nov. 2004.

[Dam] Tijs van Dam, Koen Lagendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", ACM SenSys, 2003

[Hoiydi] A. El-Hoiydi, J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks", Proc. of 9. International Symposium on Computers and Communications 2004, (ISCC 04)

[Singh] Suresh Singh, C. S. Raghavendra , "PAMAS-Power aware multi-access protocol with signalling for ad hoc networks", ACM SIGCOMM Computer Communication Review, Volume 28 Issue 3, July 1998

[Rajendran] V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks", Proc. ACM SenSys 2003.

[Rhee] I. Rhee, A. Warrier,M. Aia, J. Min, "ZMAC: A Hybrid MAC for Wireless Sensor Networks", ACM SenSys 2005, Nov 2005.

[Vuran] M.C. Vuran, and I. F. Akyildiz, "Spatial Correlation-based Collaborative Medium Access Control in Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol. 14, no. 2, pp. 316 -329, April 2006