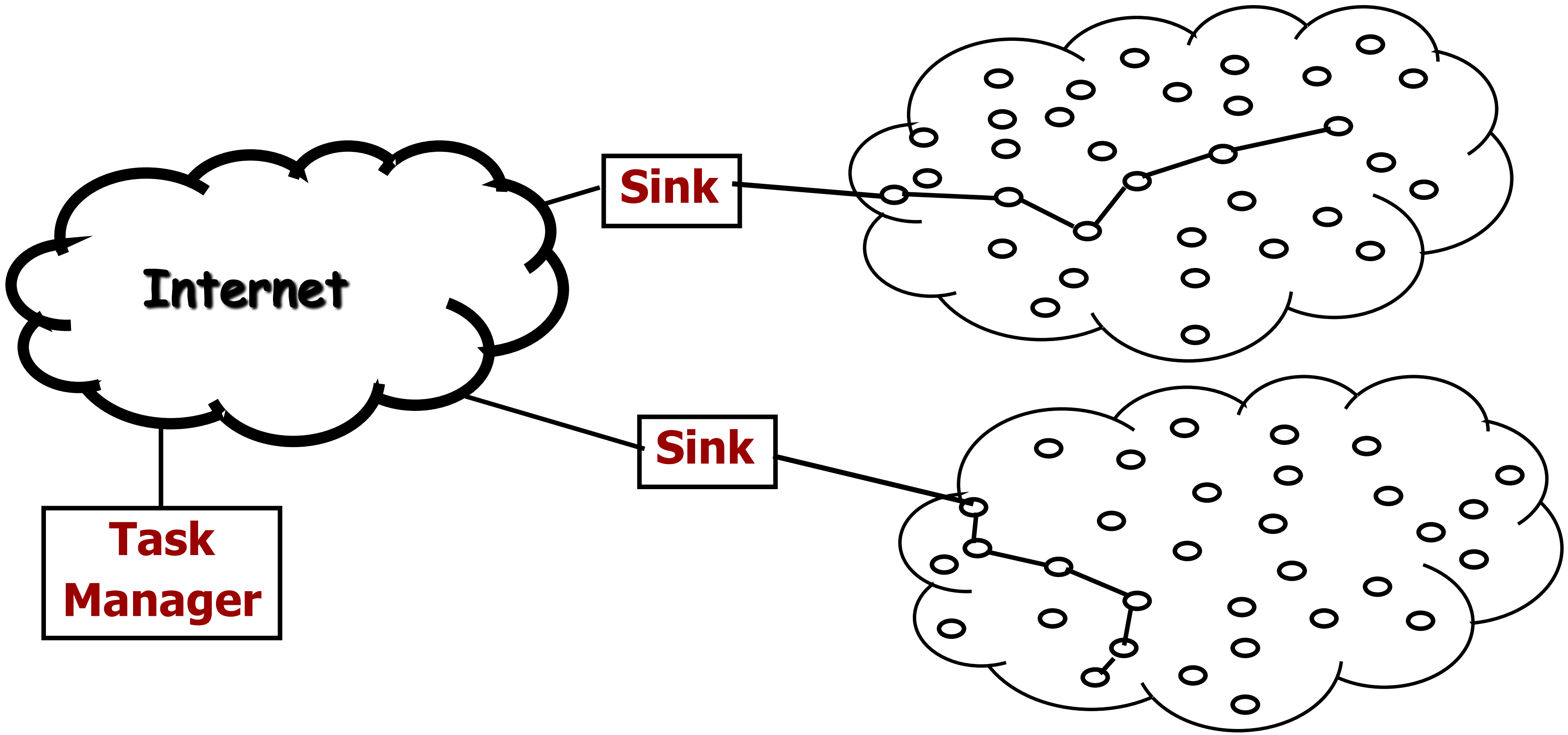
CSCE 438/838: Internet of Things

Last Classes

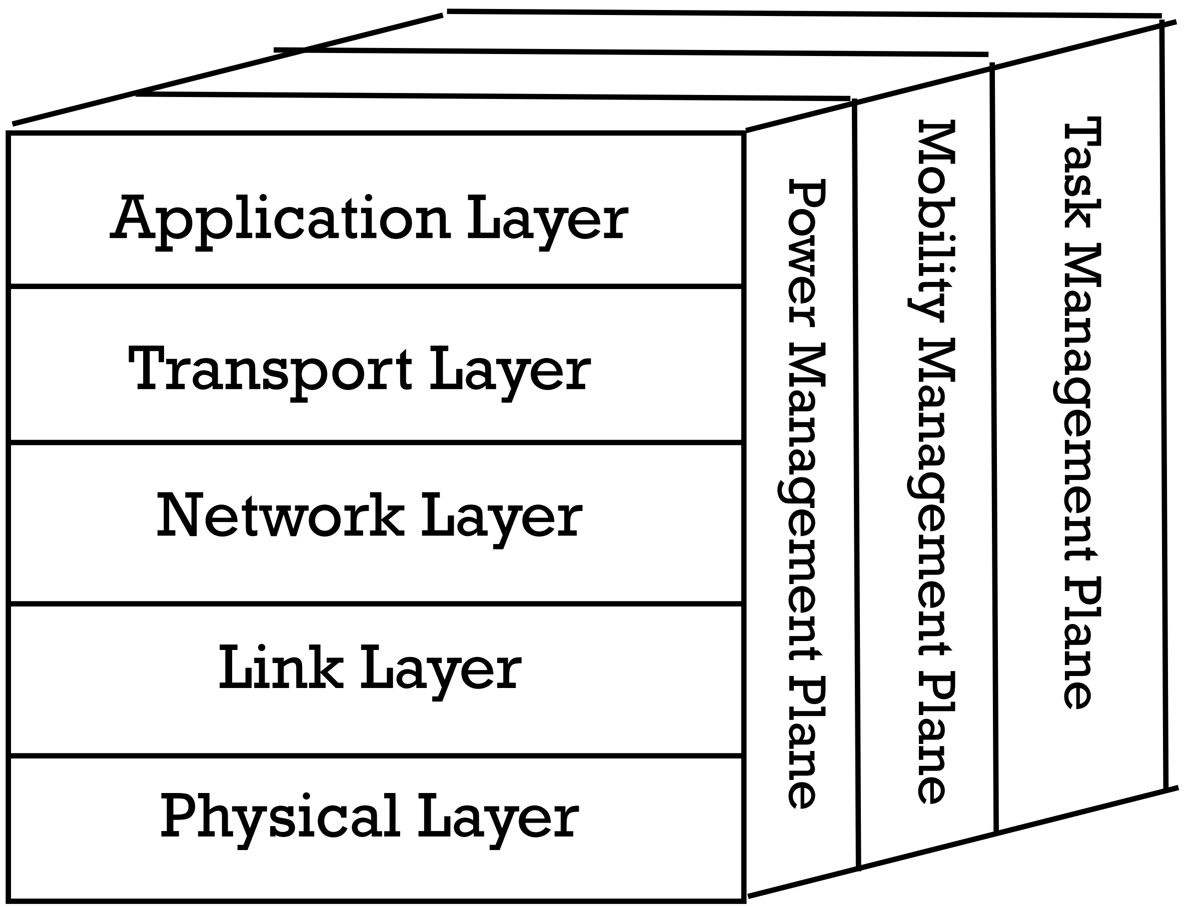
- Modulation
- LoRa

WIRELESS SENSOR NETWORK (WSN) ARCHITECTURE



I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks (Elsevier) Journal, March 2002.

PROTOCOL STACK



Medium Access Control (MAC)

What is a MAC protocol?


TDMA

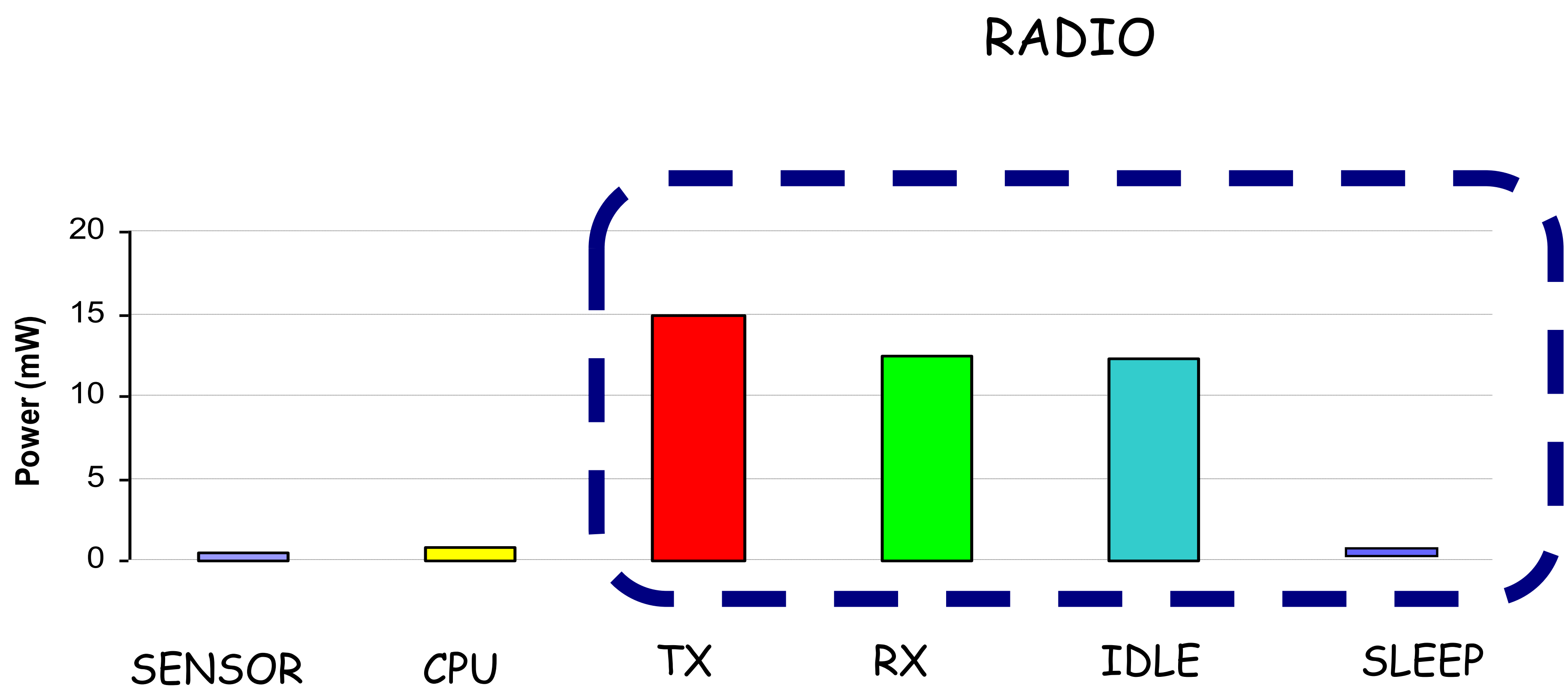

E HARRISON AVE
STOP
4-WAY
Aloha/CSMA


OFDMA

Objectives of MAC Protocols

- Collision Avoidance
- Energy Efficiency
- Scalability
- Latency
- Fairness
- Throughput
- Bandwidth Utilization

POWER CONSUMPTION



Major Sources of Energy Consumption

- Idle Listening
- Transmitter
- Receiver

Common to All Wireless Networks

- OBJECTIVE: Reduce Energy Consumption!

Challenges for MAC in IoT

- 1. Architecture
  - High density of nodes
  - Increased collision probability
  - Signaling overhead should be minimized to prevent further collisions
  - Sophisticated and simple collision avoidance protocols required

Challenges for MAC in IoT

- 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 IoT

- 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 IoT

- 4. Limited Packet Size
  - Limited header space
  - MAC protocol overhead should be minimized
- 5. Cheap Encoder/Decoders
  - 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 IoT

- 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

BOTTOMLINE: Conventional MAC protocols cannot be used for IoT!!!

MAC Protocols for WSN

- ?-MAC (pick your letter!)
- μ-MAC, A-MAC, AI-LMAC, B-MAC, Bit, BMA, CC-MAC, CMAC, Crankshaft, CSMA-MPS, CSMA/ARC, DMAC, DPS-MAC, E2-MAC, EMACs, f-MAC, FLAMA, Funneling-MAC, G-MAC, HMAC, LMAC, LEEMAC, LPL…
- MMAC, MR-MAC, MH-MAC, nanoMAC, O-MAC, PACT, PEDAMACS, PicoRadio, PMAC, PMAC, Q-MAC, QoS-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

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

Overview of MAC Protocols for WSNs -> IoT

- 1. Contention (RANDOM/CSMA)-Based MAC Protocols
  - 802.11, Sleep-MAC, BMAC, T-MAC, X-MAX, CCMAC, etc…
- 2. Reservation-Based (TDMA BASED) MAC Protocols
  - TRAMA, FLAMA, etc…
- 3. HYBRID (CSMA/TDMA) MAC Protocols
  - ZMAC, ….

Contention (Random)-Based MAC Protocols

- Channel access through carrier sense mechanism
- Provide robustness and scalability to the network
- Collision probability increases with increasing node density

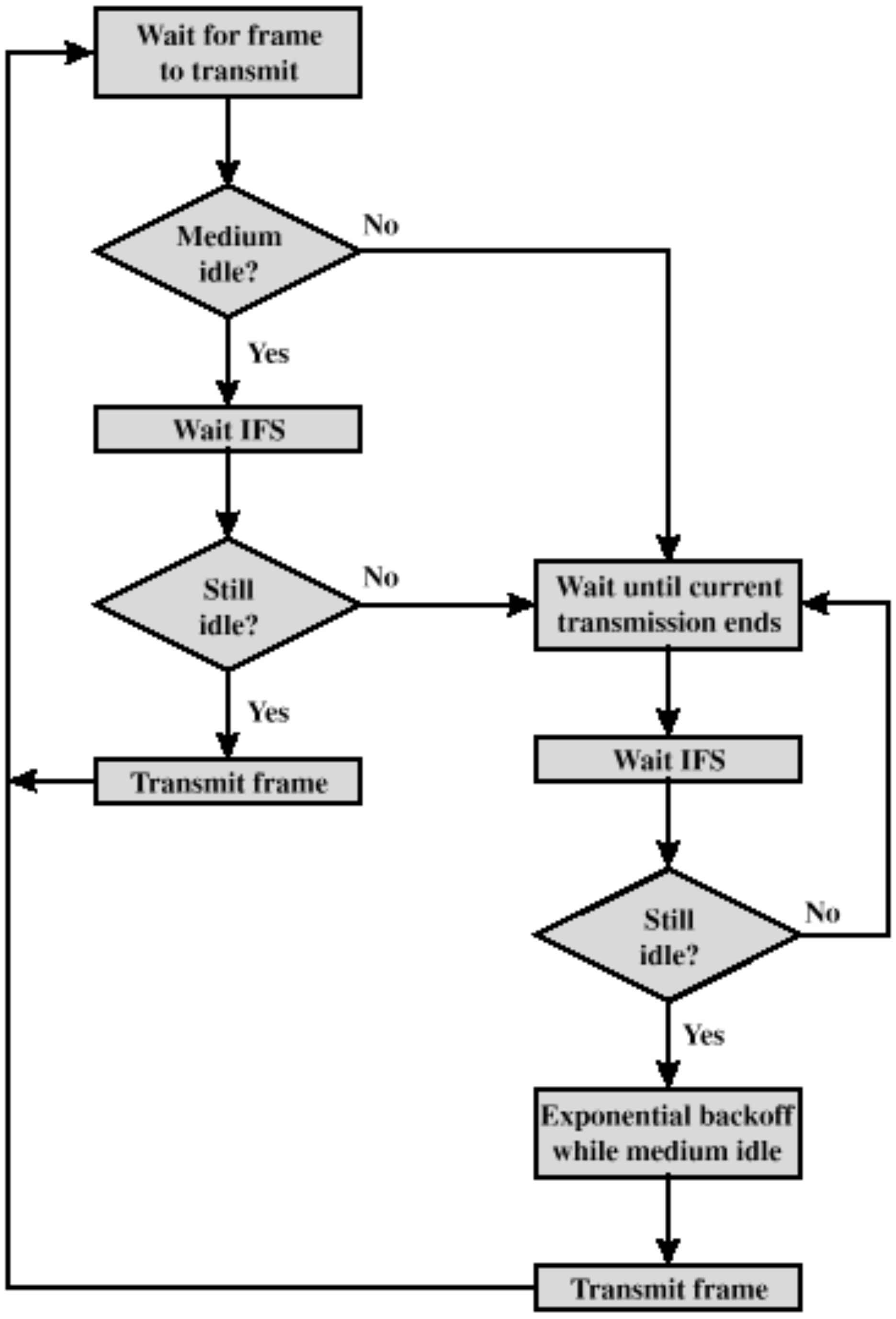
IEEE 802.11

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

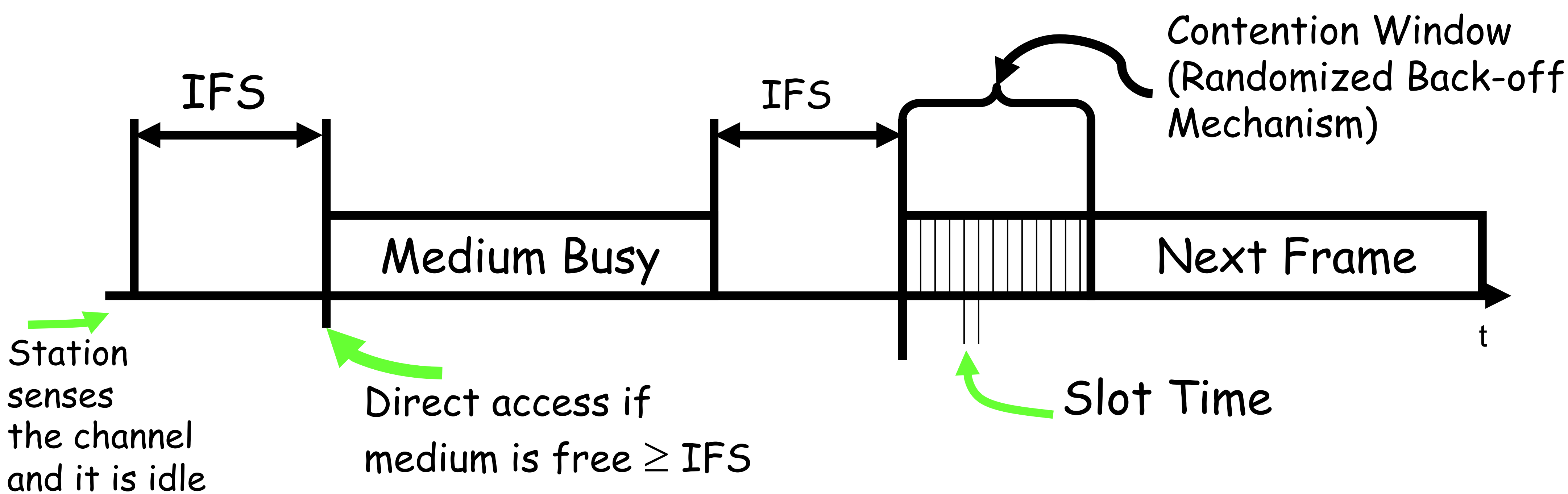
IEEE 802.11

- Let's design one…

BASIC CSMA/CA (FLOWCHART) (Distributed Foundation Wireless Medium Access Control - Distributed Coordinated Function CSMA/CA (DFWMAC-DCF) for IEEE 802.11)



BASIC CSMA/CA



BASIC CSMA/CA

- A station with a frame to transmit senses the medium (channel)
- IF IDLE
  - Wait to see if the channel remains idle for a time equal to IFS (Inter-frame spacing)
  - If so, transmit immediately
- IF BUSY
  - Why busy? Either (1) the station initially finds the channel busy or (2) the channel becomes busy during the IFS idle time
  - Defer transmission and continue to monitor the channel until the current transmission is over

BASIC CSMA/CA

- 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 maximum 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.
- Then the CONTENTION WINDOW starts in which all stations with packets for transmission run down their BACKOFF clocks.
- The first station with its clock expiring 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.

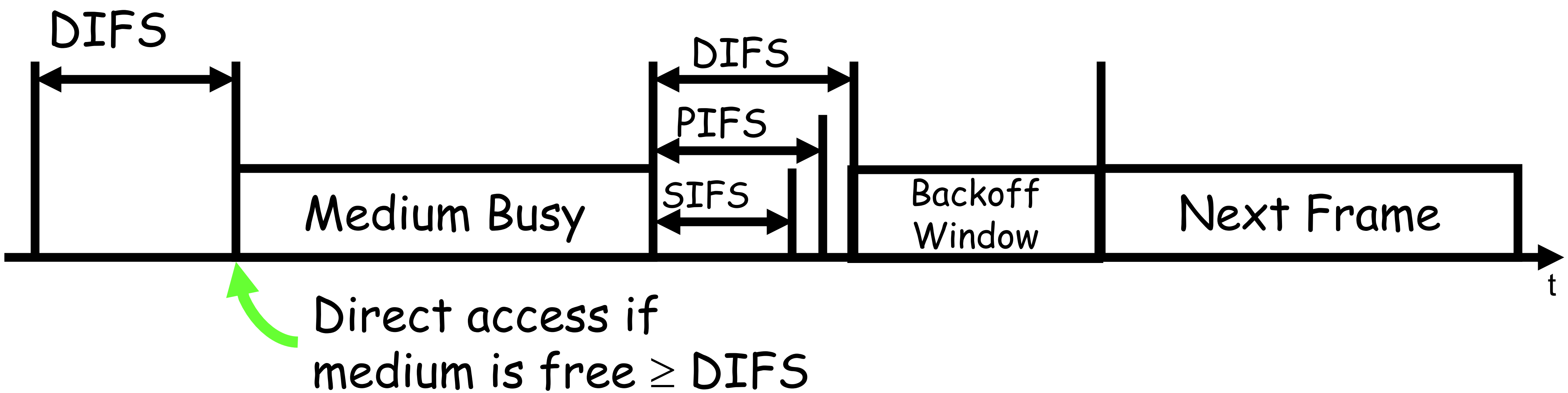
CSMA/CA Algorithm

- 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.

CSMA/CA Algorithm

- What about ACK?

Inter-frame Spaces (IFS)



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

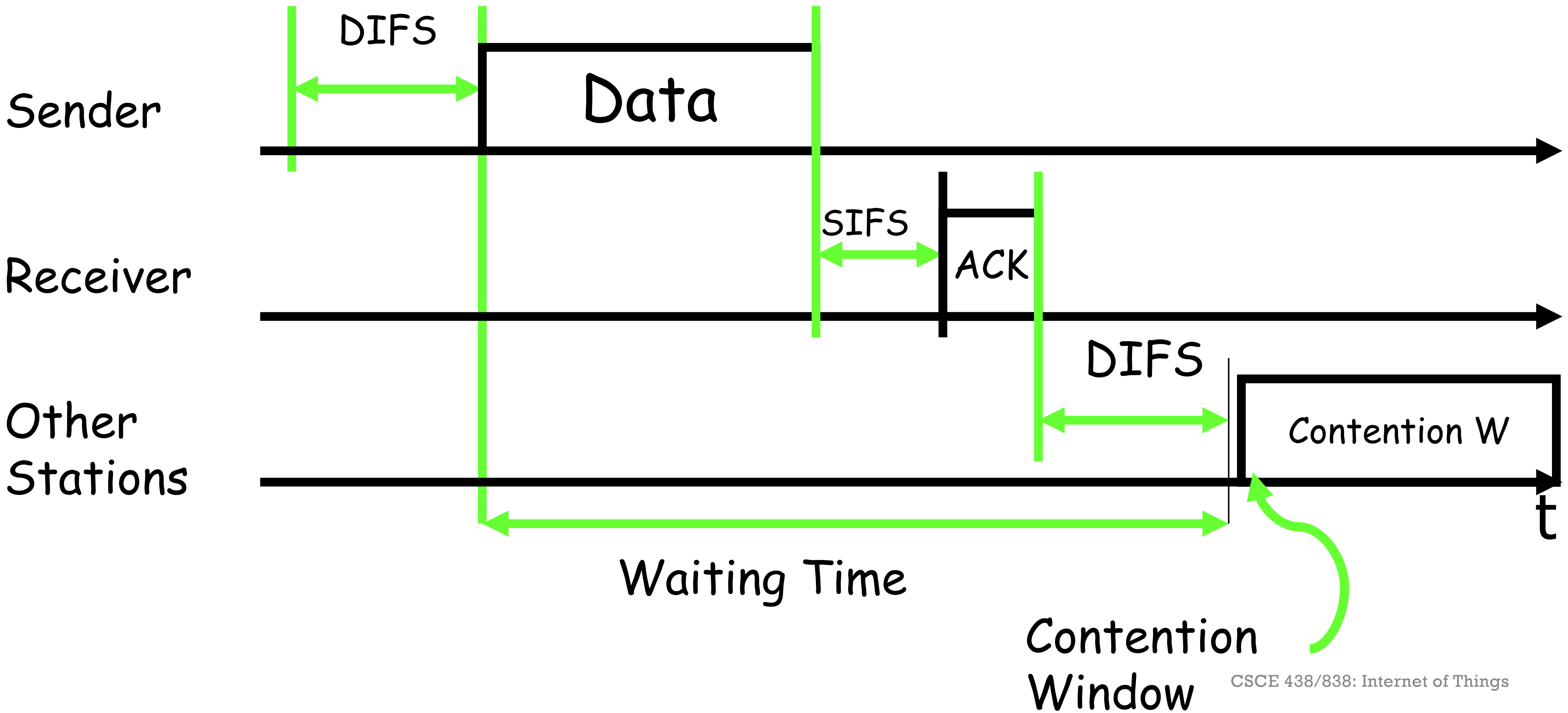
Inter-frame Spaces (IFS)

- 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

DFWMAC-DCF CSMA/CA with ACK

- Station has to wait for DIFS before sending data
- Receiver ACKs immediately (after waiting for SIFS < DIFS) if the packet was received correctly (CRC))
- Receiver transmits ACK without sensing the medium.
- If ACK is lost, retransmission is performed
- Automatic retransmission of data packets in case of transmission errors

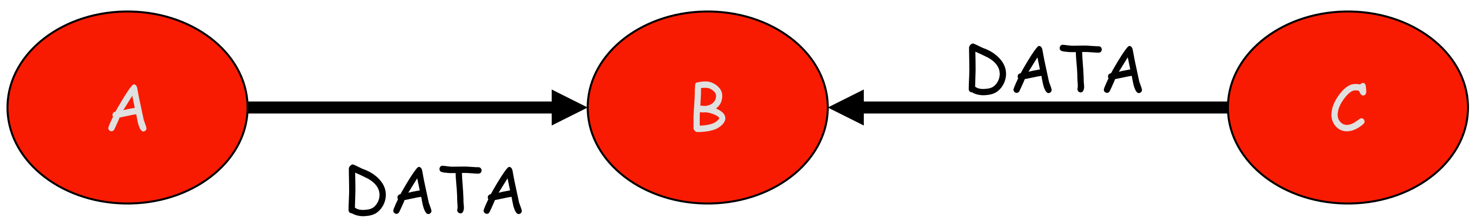
DFWMAC-DCF CSMA/CA with ACK



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 has a packet for A, C has a packet for D
- 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

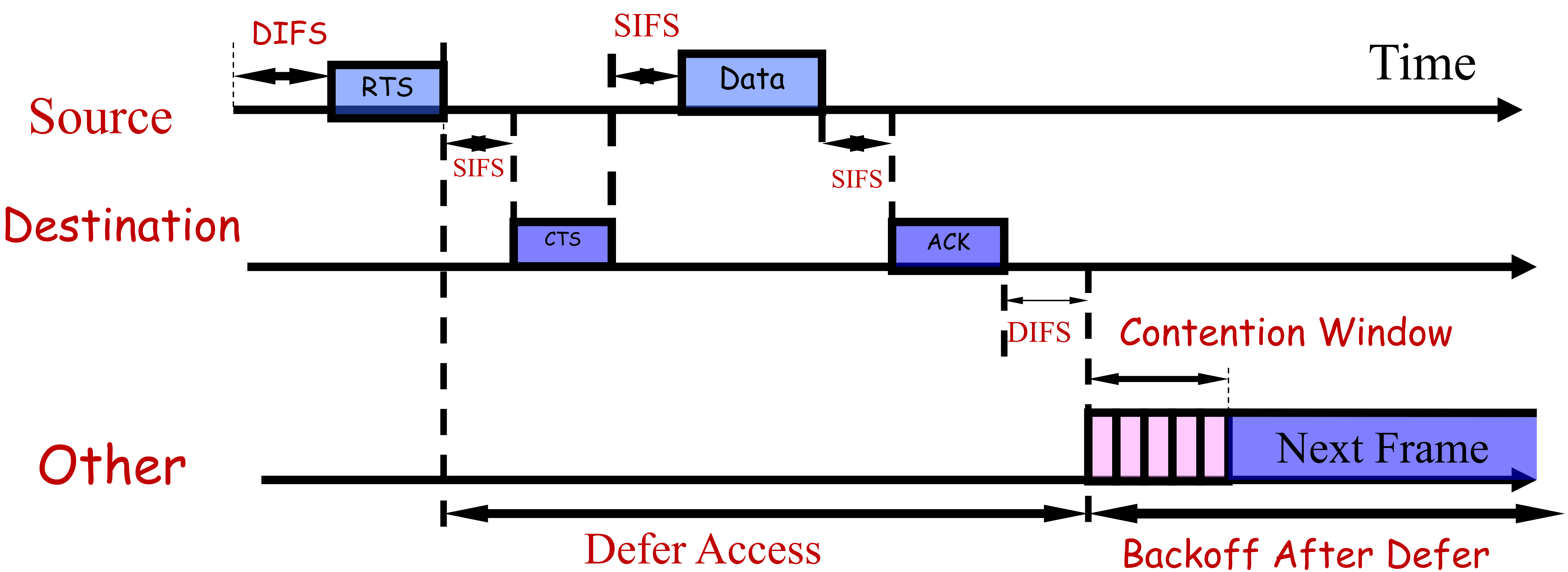
DFWMAC-DCF CSMA/CA with RTS/CTS

- 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.

DFWMAC-DCF CSMA/CA with RTS/CTS

- Use short signaling packets for Collision Avoidance
- RTS (Request To Send) Packet (20 Bytes):
  - A sender requests the right to send from a receiver with a short RTS packet before it sends a data packet
- CTS (Clear To Send) Packet (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)

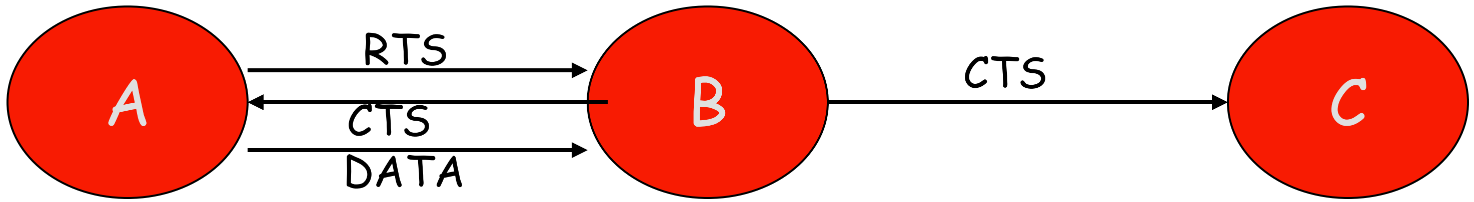
DFWMAC-DCF CSMA with RTS/CTS



Problems with CSMA/CA

- Hidden terminal problem
- Exposed terminal problem

Hidden Terminal Problem

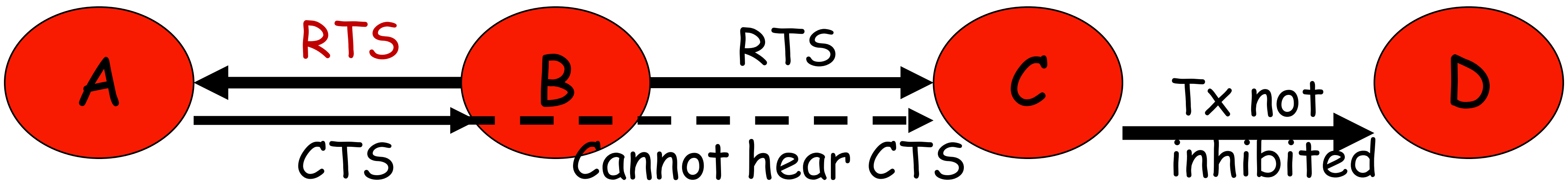


- A sends RTS
- B sends CTS
- C overhears CTS
- C inhibits its own transmitter
- A successfully sends DATA to B

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

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)
  - For traditional wireless networks!

DFWMAC-DCF CSMA/CA with RTS/CTS (Network Allocation Vector (NAV))

- Both Physical Carrier Sensing and Virtual Carrier Sensing are 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 the NAV (Network Allocation Vector)

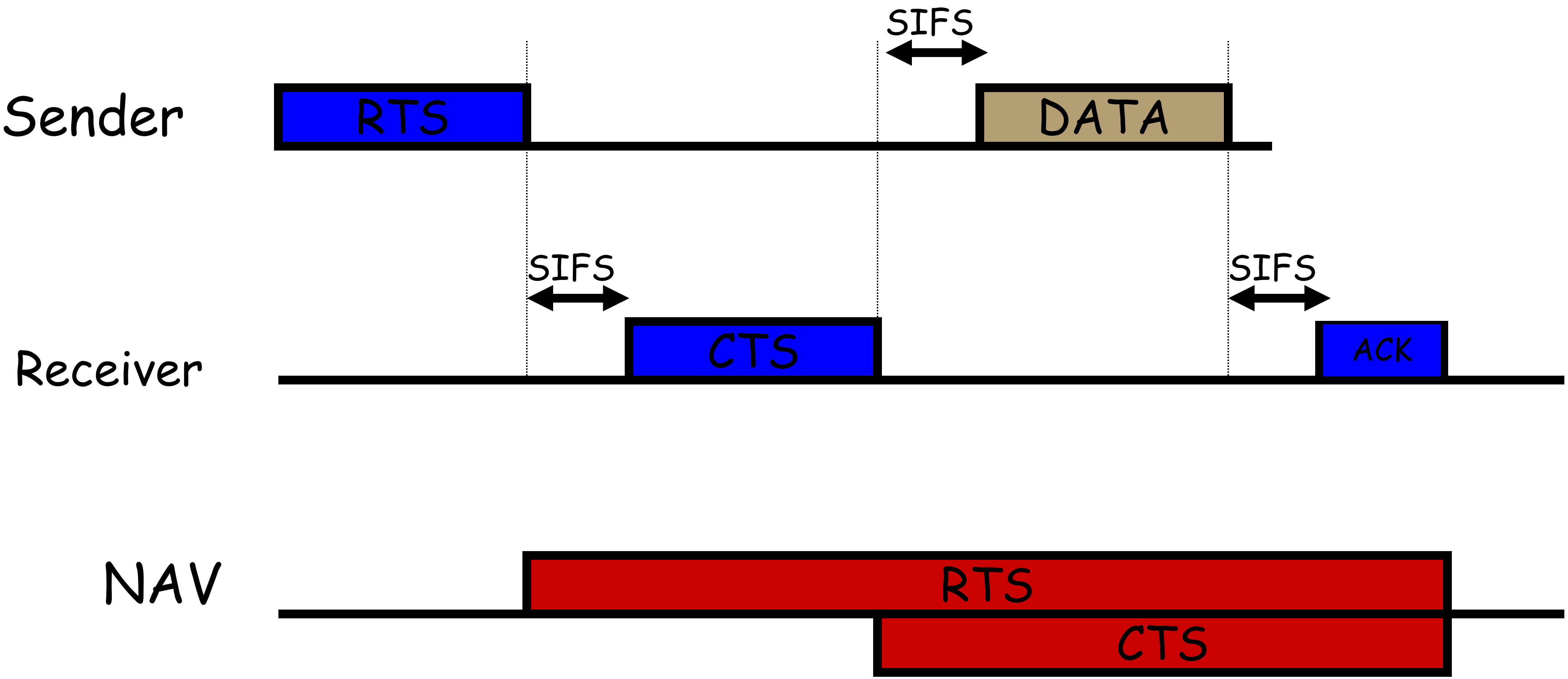
DFWMAC-DCF CSMA/CA with RTS/CTS (Network Allocation Vector (NAV))

- 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

DFWMAC-DCF CSMA/CA with RTS/CTS (Network Allocation Vector (NAV))

- 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.

Illustration



CSMA/CA with RTS/CTS (NAV)

- 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

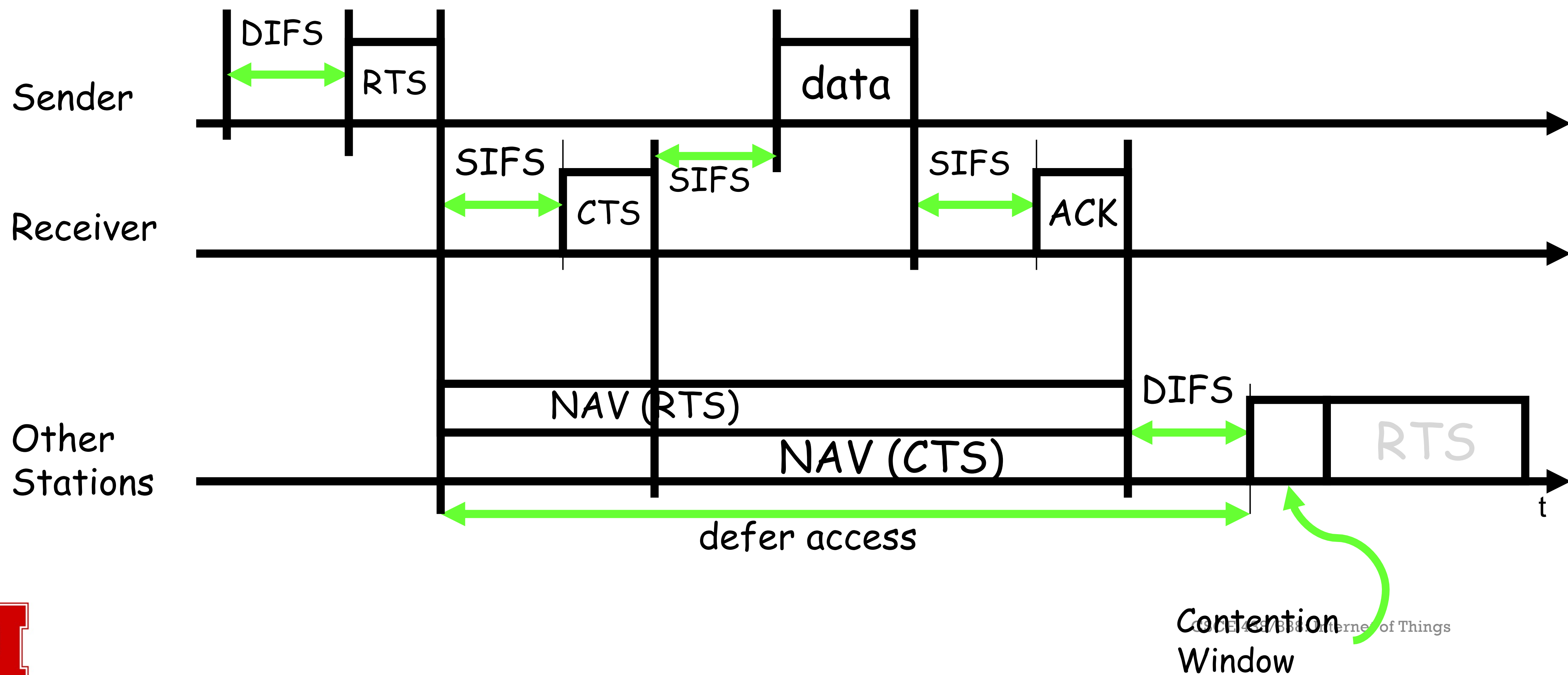
CSMA/CA with RTS/CTS (NAV)

- 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 RTC/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

CSMA/CA with RTS/CTS (NAV)

- 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 (NAV)



MAC Protocols for WSNs

- Contention (RANDOM/CSMA)-Based MAC Protocols
  - BMAC

Goals of BMAC

- Low Power Operation
- Effective Collision Avoidance
- Simple Implementation, Small Code and RAM Size
- Efficient Channel Utilization
- Reconfigurable by Network Protocols
- Tolerant to Changing RF/Networking Conditions
- Scalable to Large Numbers of Nodes

B-MAC

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

- 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)

B-MAC Interfaces

```
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();
}
```

- Interfaces for flexible control of BMAC 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

B-MAC Design

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

Clear Channel Assessment

- Effective collision avoidance
- Find out whether the channel is idle
  - If too pessimistic: waste bandwidth
  - If too optimistic: more collisions

Clear Channel Assessment

- 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
- -> BMAC 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 is added to an exponentially weighted moving average with decay *a*
- Median signal strength is used as a simple low pass filter to add robustness to the noise floor estimate.
- $A_t = a * S_t + (1 - a) * S_t\text{-}1$
- where a value is assumed to be 0.06 and FIFO queue size of 10.

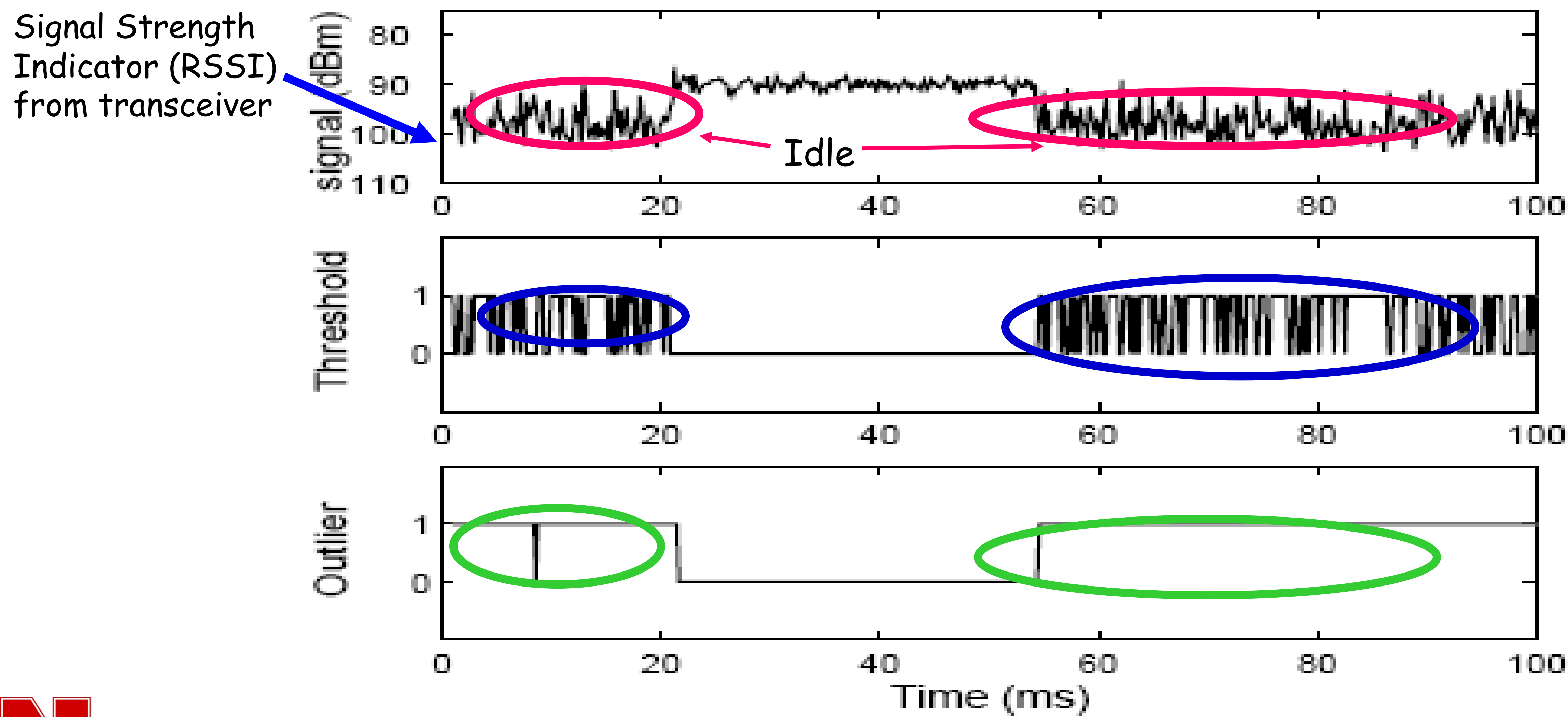
Clear Channel Assessment

- 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
- BMAC: 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

Clear Channel Assessment

- CCA can be turned on/off (see BMAC-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

Clear Channel Assessment

- 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.

Clear Channel Assessment

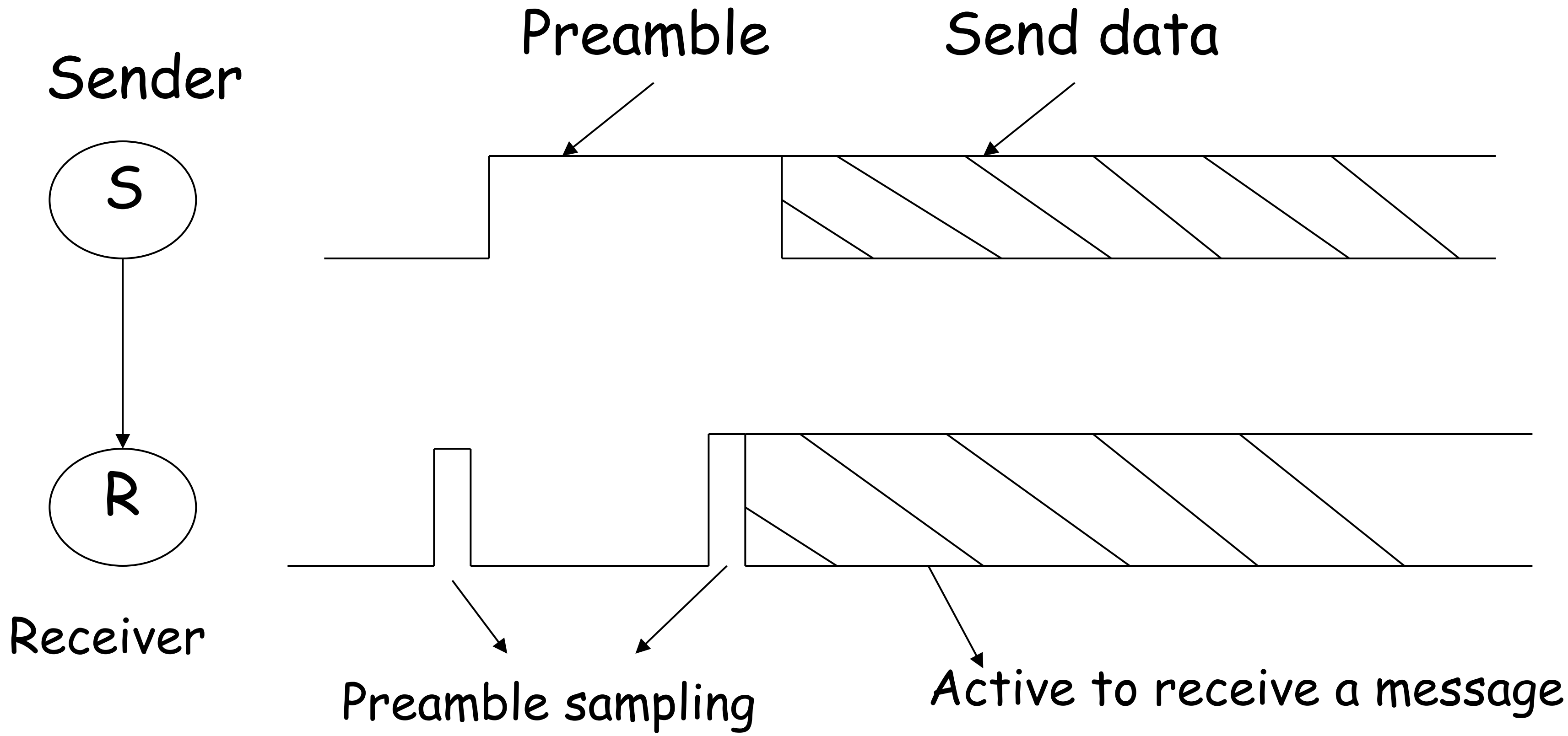
- If no backoff time is given, again a small random backoff is used.
- Enabling or disabling CCA and configuring the backoff allows services to change the fairness and available throughput.

Low Power Listening (LPL)

- Goal: Minimize "Listen Cost"
- Principles
  - Node periodically wakes up, turns radio on and checks activity on the channel
    - Wakeup time fixed (time spent 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

LPL- Preamble Sampling

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



|Preamble| >= Sampling period

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

Check Interval for Channel Activity

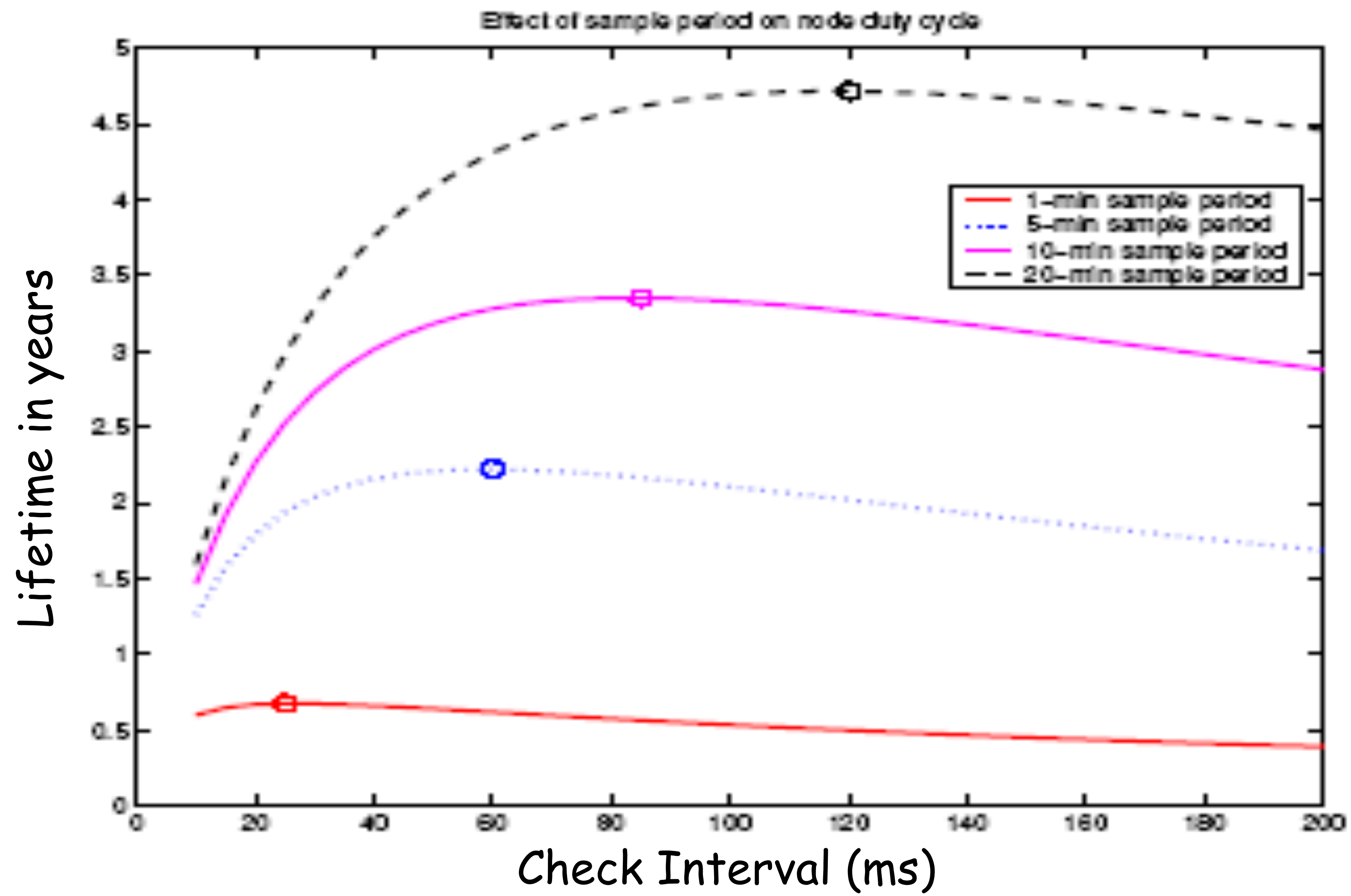
- Interval between LPL samples is maximized so that the time spent sampling the channel is minimized.

- Transmit mode ~~ Preamble length
- Listening mode ~~ Check interval

LPL 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

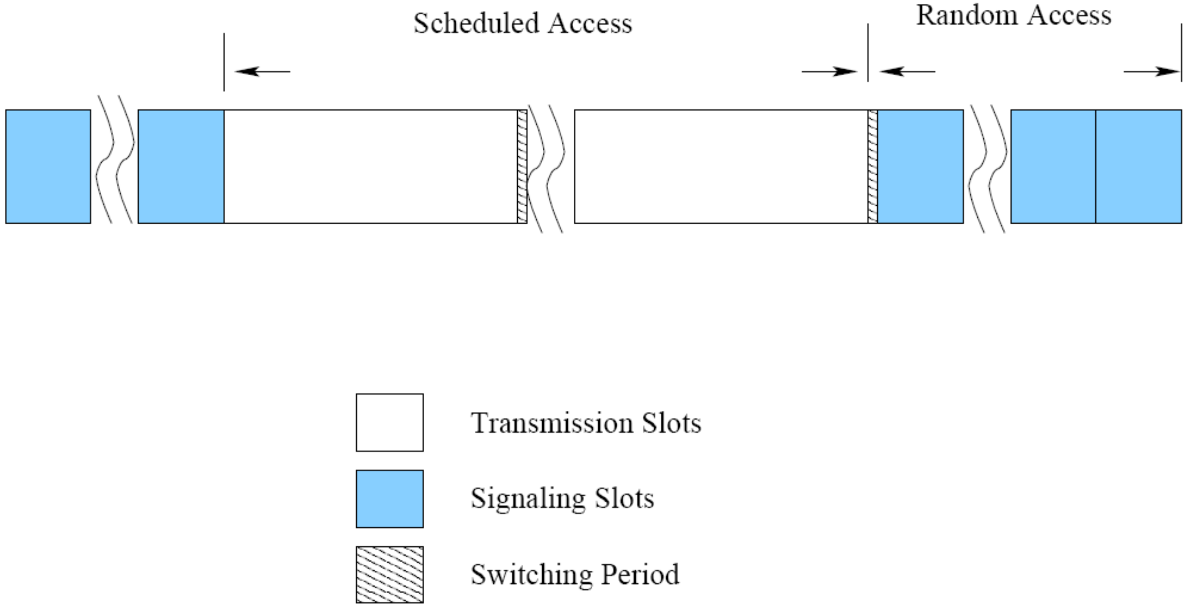


MAC Protocols for WSNs

- 1. Contention (RANDOM/CSMA)-Based MAC Protocols
  - Sleep-MAC, BMAC, CCMAC, etc…
- 2. Reservation-Based (TDMA BASED) MAC Protocols
  - TRAMA, FLAMA, etc…
- 3. HYBRID (CSMA/TDMA) MAC Protocols
  - ZMAC, ….

TRAMA: TRaffic-Adaptive MAC
V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, ``Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," Proc. ACM SenSys 2003, LA, CA, Nov. 2003.

- A time-slotted structure



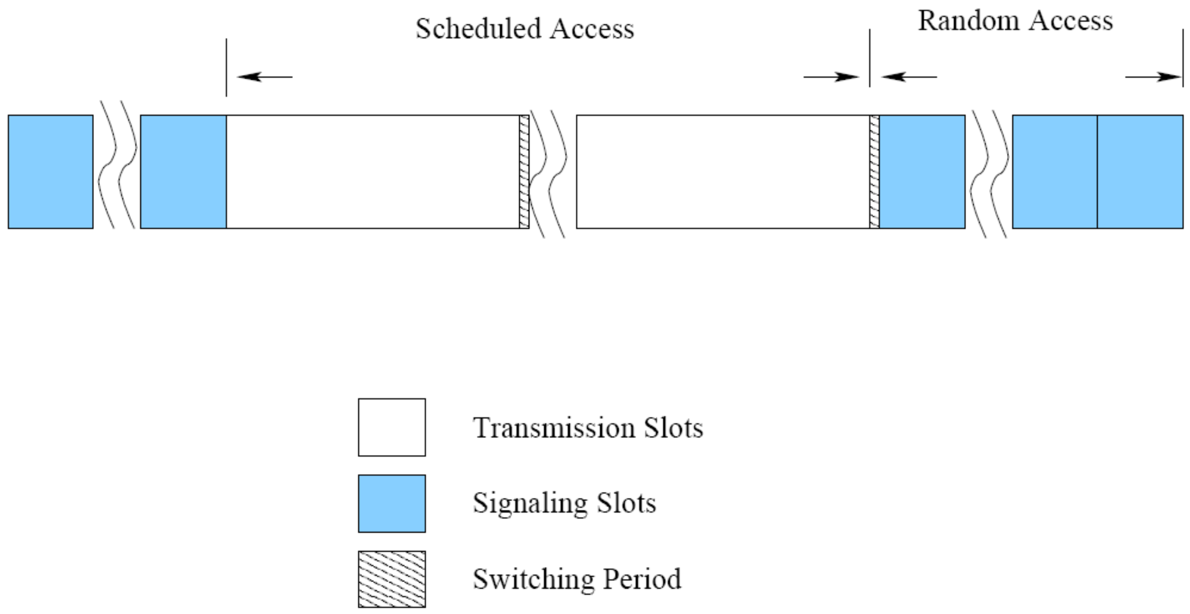
115.2 kbps → transmission slot 46ms (512-byte segments)

TRAMA

- Time is divided into PERIODS:

- Random Access Period
  - Used for signaling: synchronization and updating two-hop neighbour information.
  - Collision!!

- Scheduled Access Period:
  - Used for contention free data exchange between nodes.
  - Supports unicast, multicast, and broadcast communication.



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
  - Other nodes can switch to low power mode using the NP and SEP results

Neighbor Protocol (NP)

- 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

Type	SourceAddr	DestAddr	DeleteNum	AddNum	Deleted NodeID's	Added NodeID's

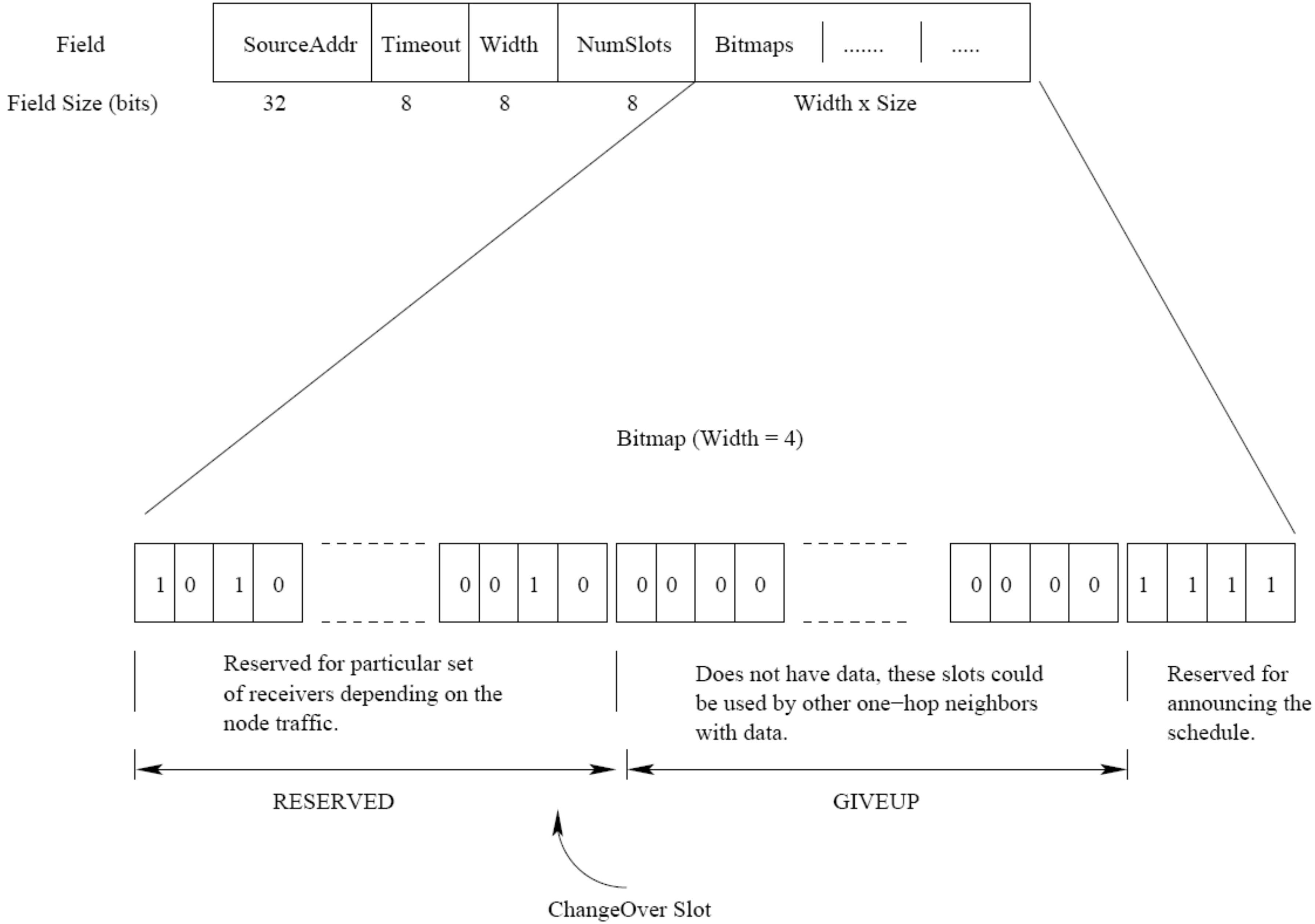
Neighbor Protocol

- A node times out if nothing is heard from its neighbor
- Updates retransmitted to guarantee packet delivery

Schedule Exchange Protocol (SEP)

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

Schedule Packet Format



Schedule Exchange Protocol (SEP)

- The node pre-computes # of slots in the interval
- [t, t+SCHED]
- for which it has the highest priority among its two-hop neighbors (contenders) → WINNING SLOTS

Adaptive Election Algorithm (AEA)

- 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 x and globally known hash function h
  - For time slot t, compute priority p = h (x © 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

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

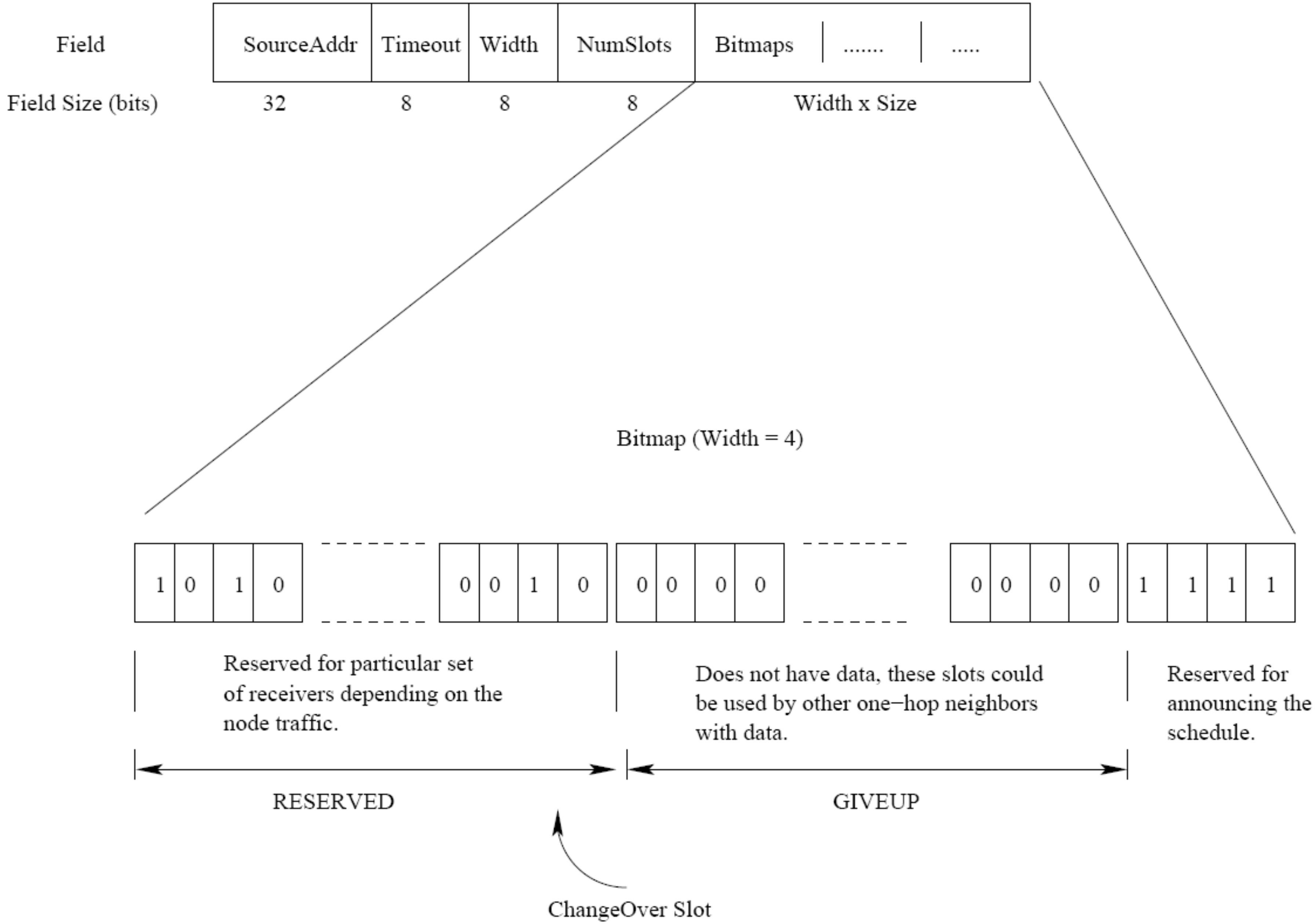
Schedule Exchange Protocol (SEP)

- 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 # 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

- SourceAddr: Node announcing the schedule.
- Timeout: # of of slots for which the schedule is valid (starting from the current slot)
- Width: Length of the neighbor bitmap (# of one hop neighbors)
- numSlots: total # of winning slots (# of bitmaps contained in the packet)

Data Packet

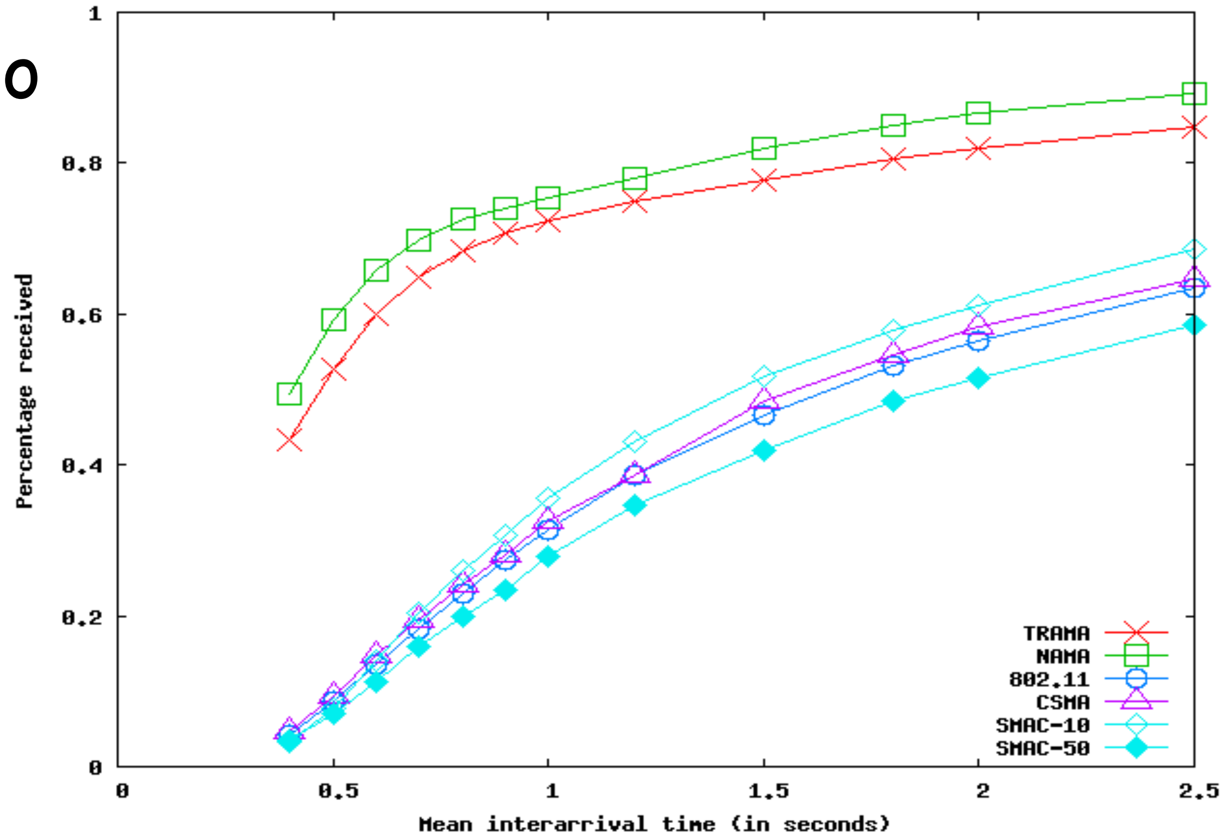
- Timeout: # of slots for which the schedule is valid
- NumSlots: # of winning slots
- Bitmap: Indicates whether the node is transmitting or not (|Bitmap| = NumSlots)

Type	SourceAddr	DestAddr	Timeout	NumSlots	Bitmap

Short Schedule Summary

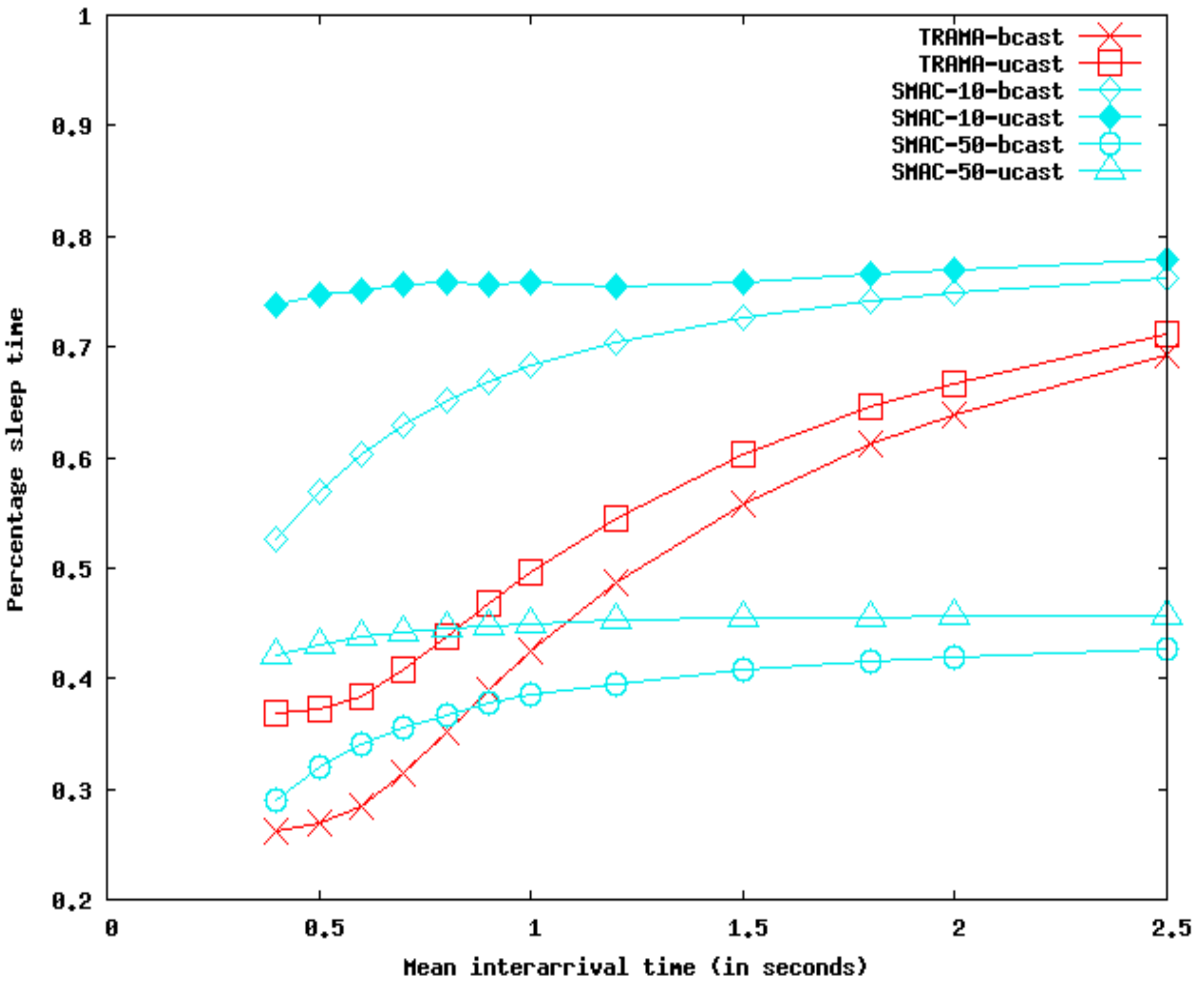
Delivery Ratio

Simulation Results



- Broadcast traffic using Poisson arrivals.
- 50 nodes, 500x500 area.
- 512 byte data.
- Average node density: 6

Energy Savings



Percentage Energy Savings

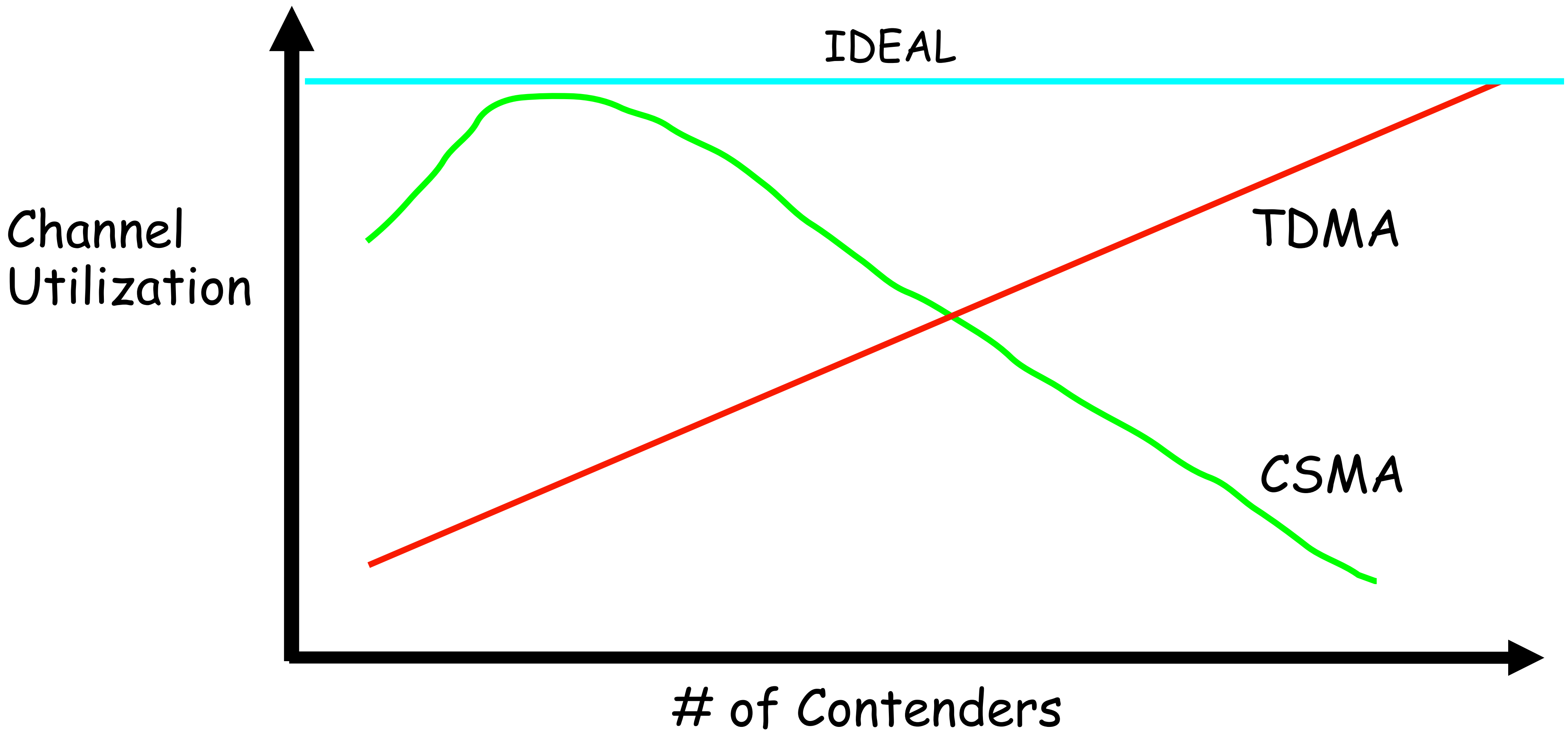
TRAMA Limitations

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

MAC Protocols for WSNs

- 1. Contention (RANDOM/CSMA)-Based MAC Protocols
  - Sleep-MAC, BMAC, T-MAC, CCMAC, etc…
- 2. Reservation-Based (TDMA BASED) MAC Protocols
  - TRAMA, PMAC, Energy-aware TDMA, BMA-MAC, Adaptive Low Power Res-based…
- 3. HYBRID (CSMA/TDMA) MAC Protocols
  - ZMAC, ….

Effective Throughput
CSMA vs. TDMA



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

Z(ebra)-MAC: A HYBRID MAC PROTOCOL
I. Rhee, A. Warrier,M. Aia, J. Min, ACM SenSys 2005, Nov 2005.

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

Z-MAC

- Uses the TDMA schedule as a 'hint' to schedule transmissions
- 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

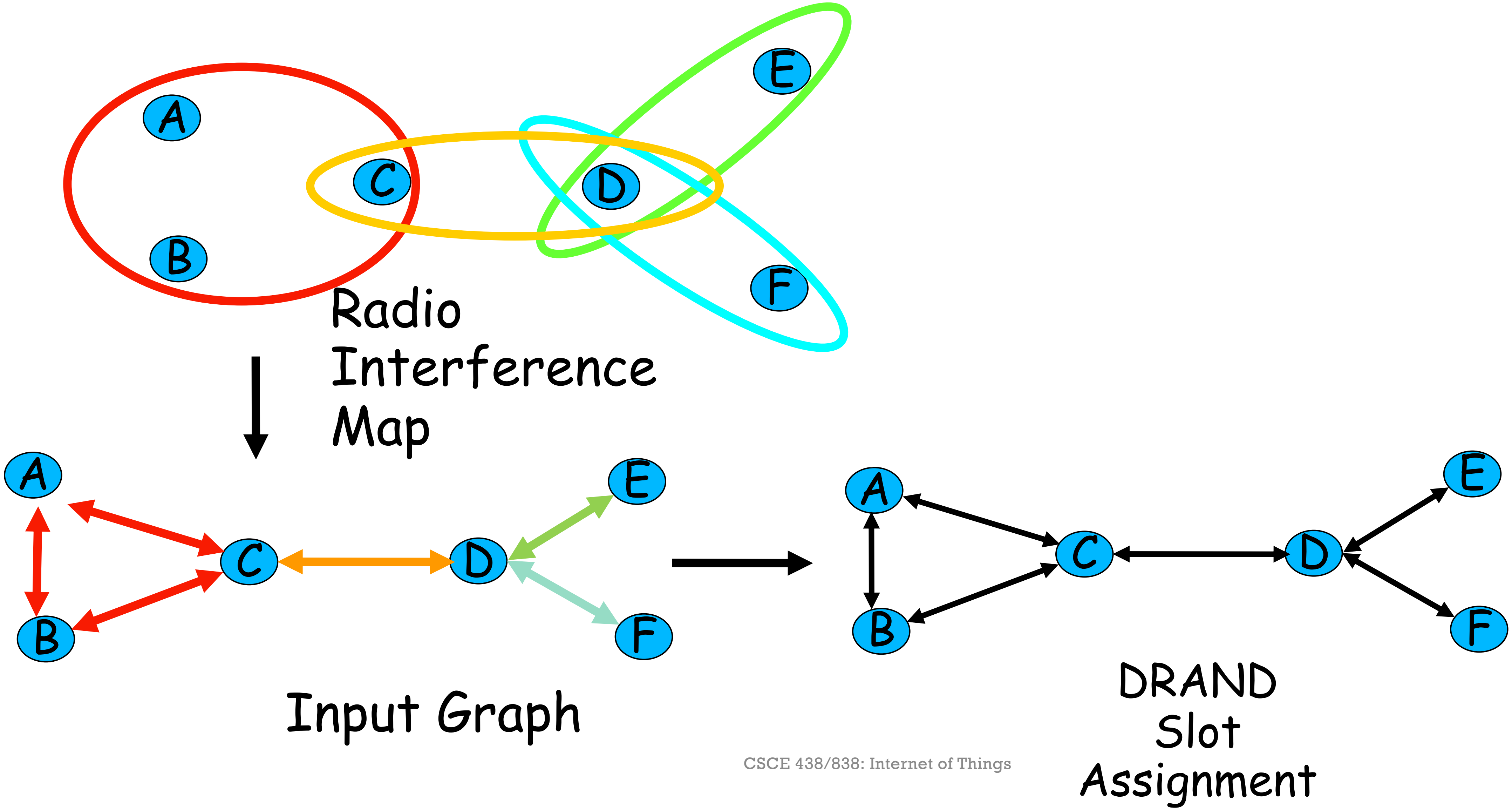
Z-MAC

- 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)

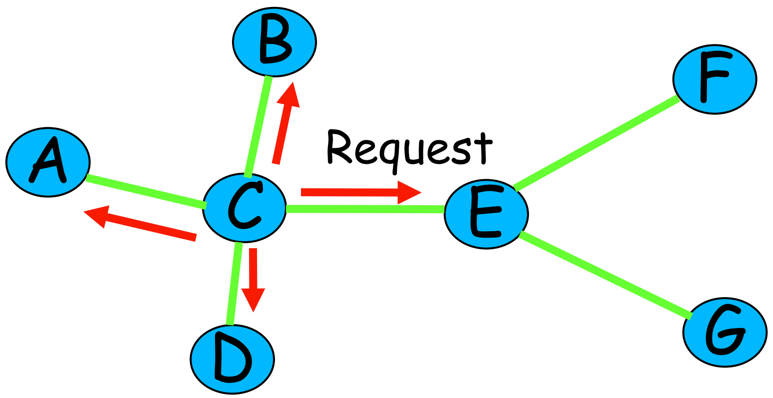
Z-MAC Operation

- Setup phase
  - Neighbour discovery
  - Slot assignment
  - Local frame exchange
  - Global time synchronization
- Communication phase
  - Setup phase is not used until a significant change in topology

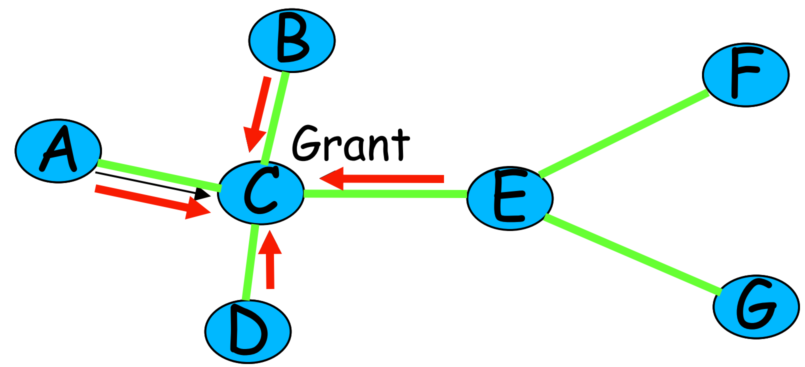
DRAND (Distributed TDMA Scheduling) – Algorithm



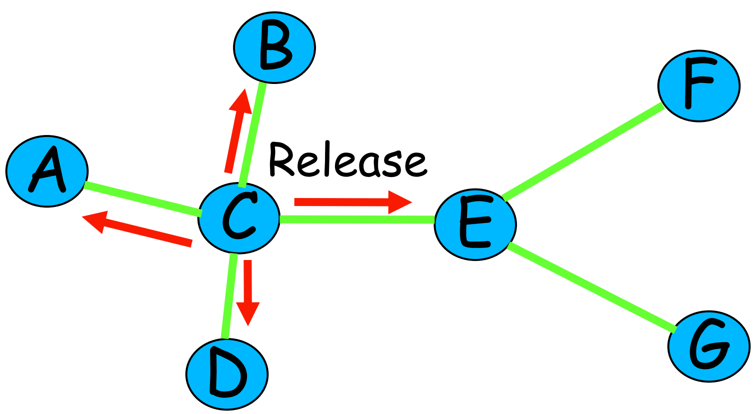
DRAND – Algorithm – Successful Round



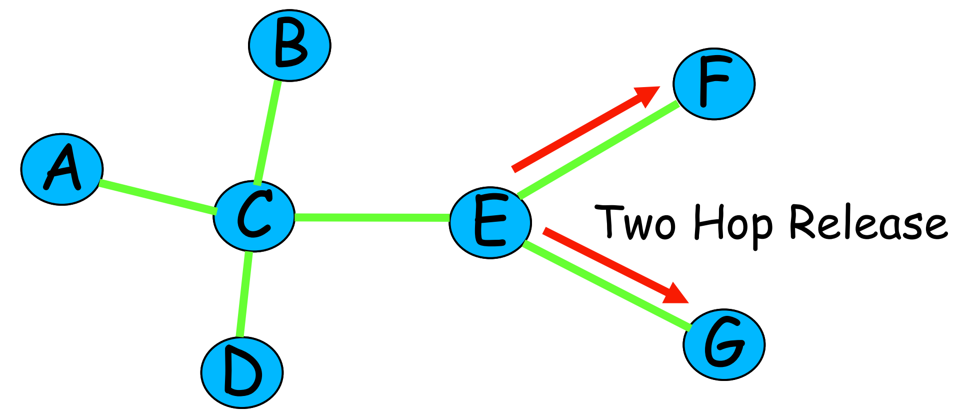
Step 1. Broadcast Request



Step 2. Receive Grants

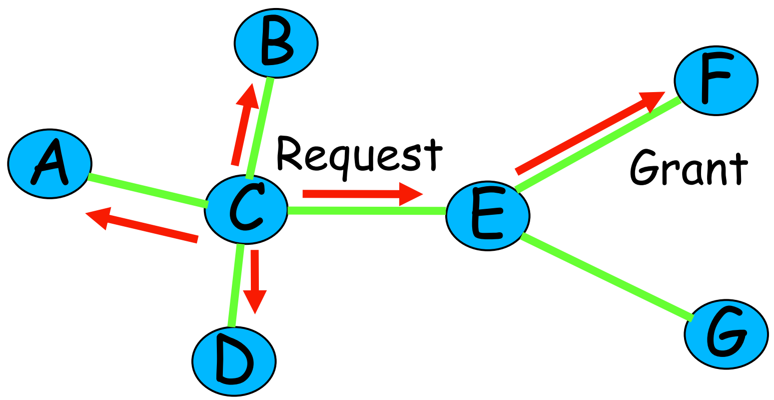


Step 3. Broadcast Release

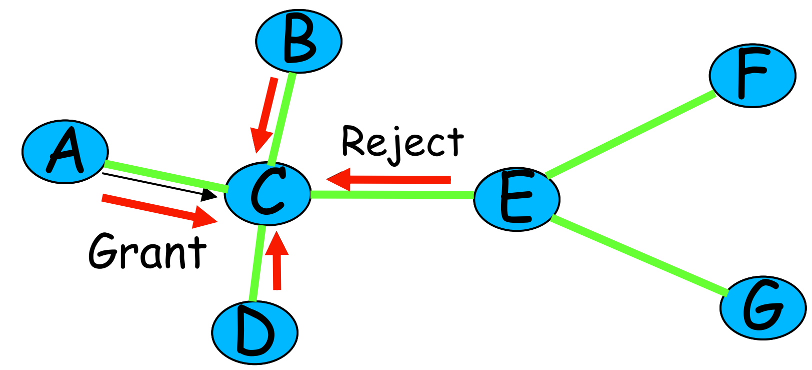


Step 4. Broadcast Two Hop Release

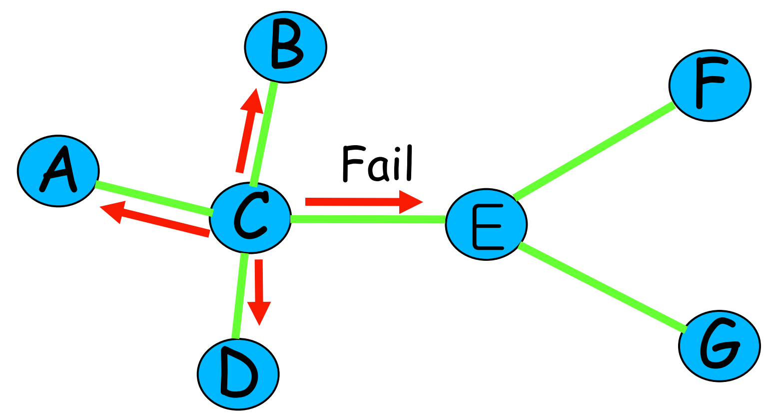
DRAND – Algorithm – Unsuccessful Round



Step 1. Broadcast Request



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



Step 3. Broadcast Fail

Transmission Control

- Slot Ownership
  - If current timeslot is the node's assigned time-slot, then it is the Owner, and all other neighboring nodes are Non-Owners.

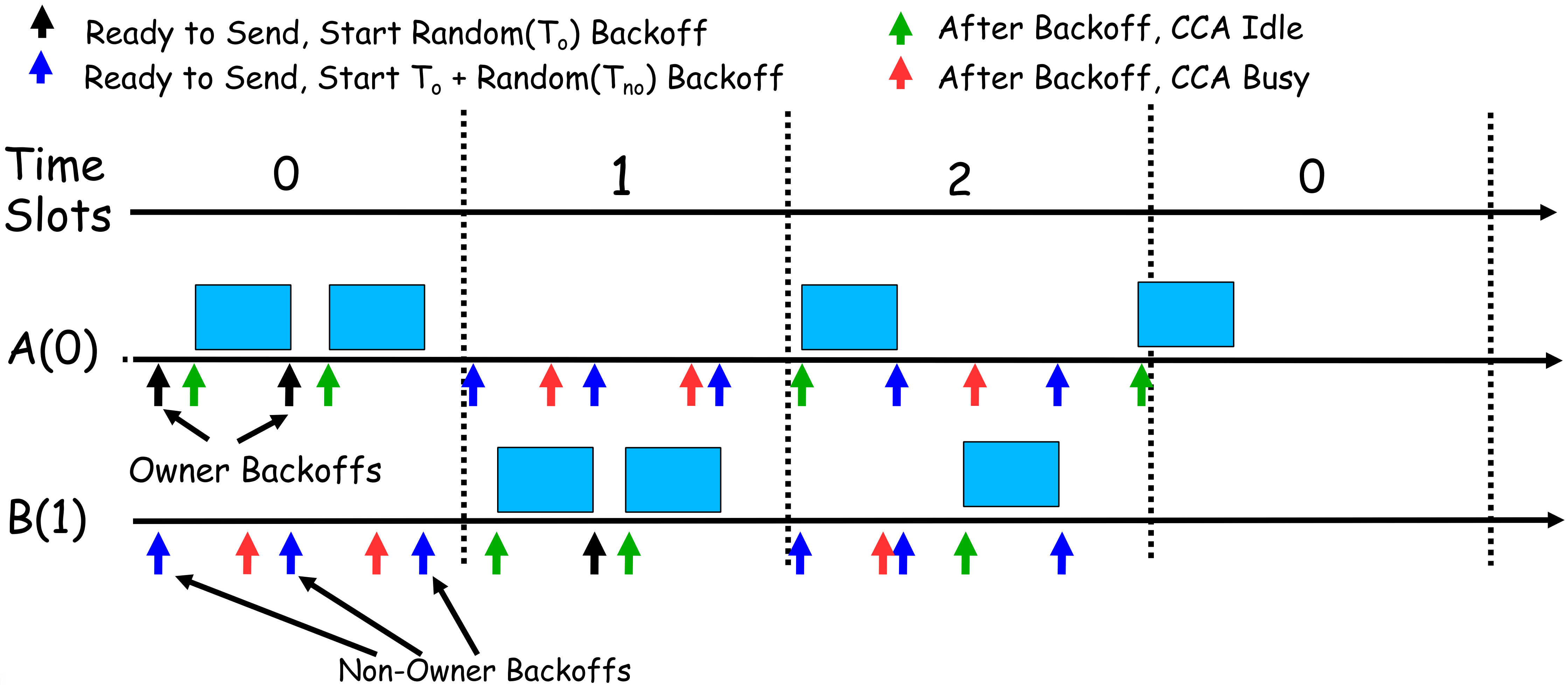
Transmission Control

- 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.

Transmission Control

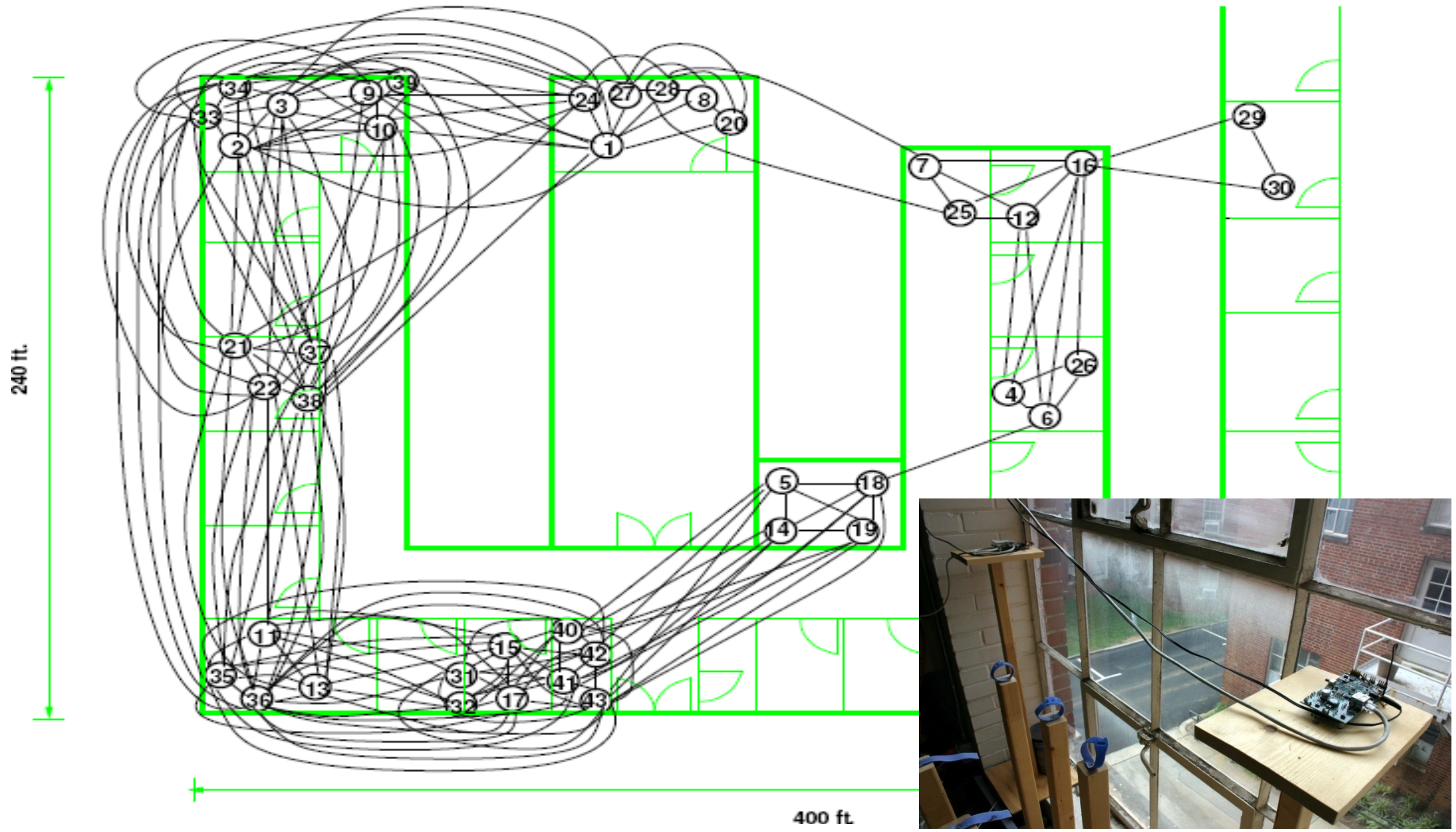
- Switches between CSMA and TDMA automatically depending on contention level
- Performance depends on specific values of T_o and T_{no}
- From analysis, $T_o = 8$ and $T_{no} = 32$ are used for best performance

Transmission Control

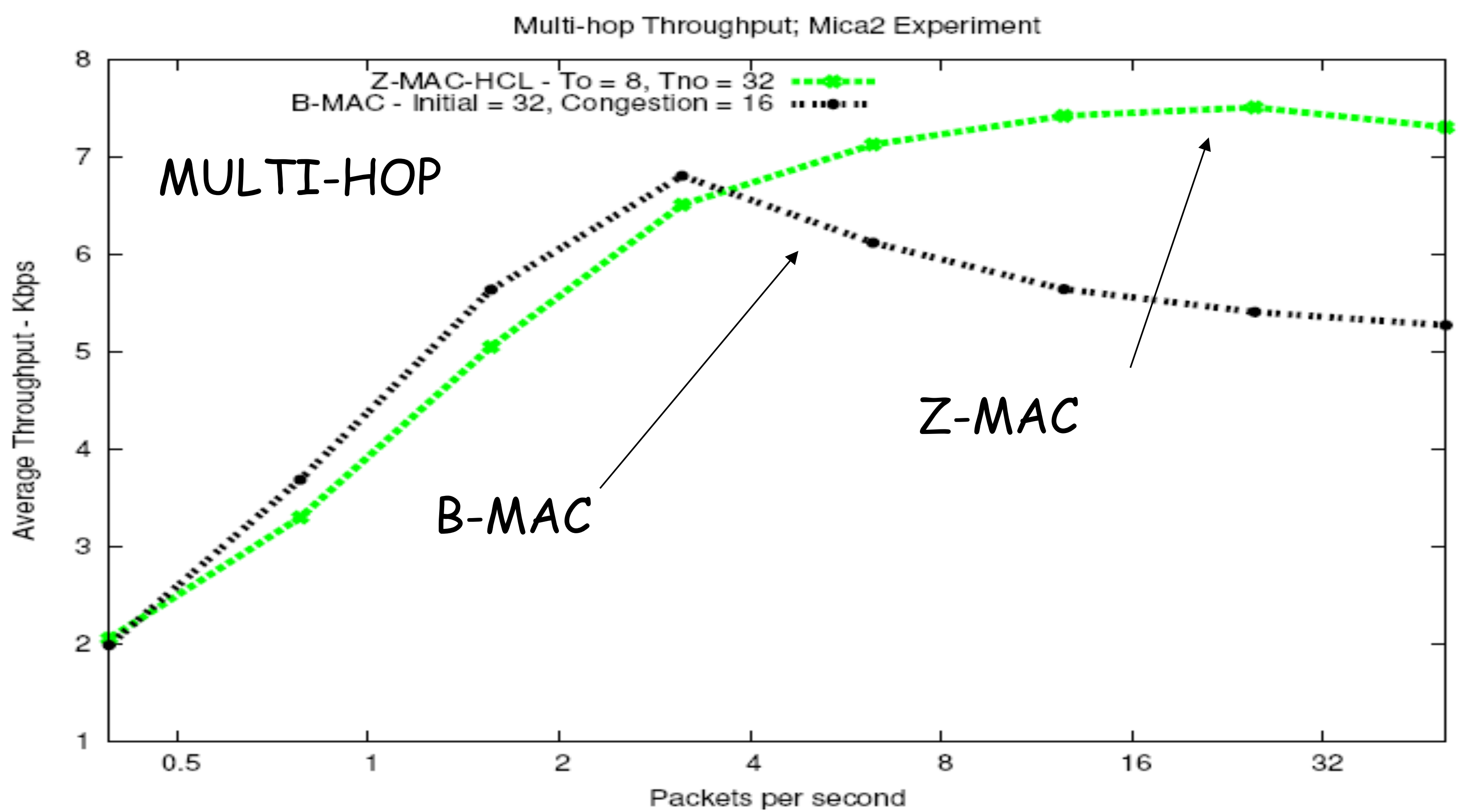


Experimental Setup - Testbed

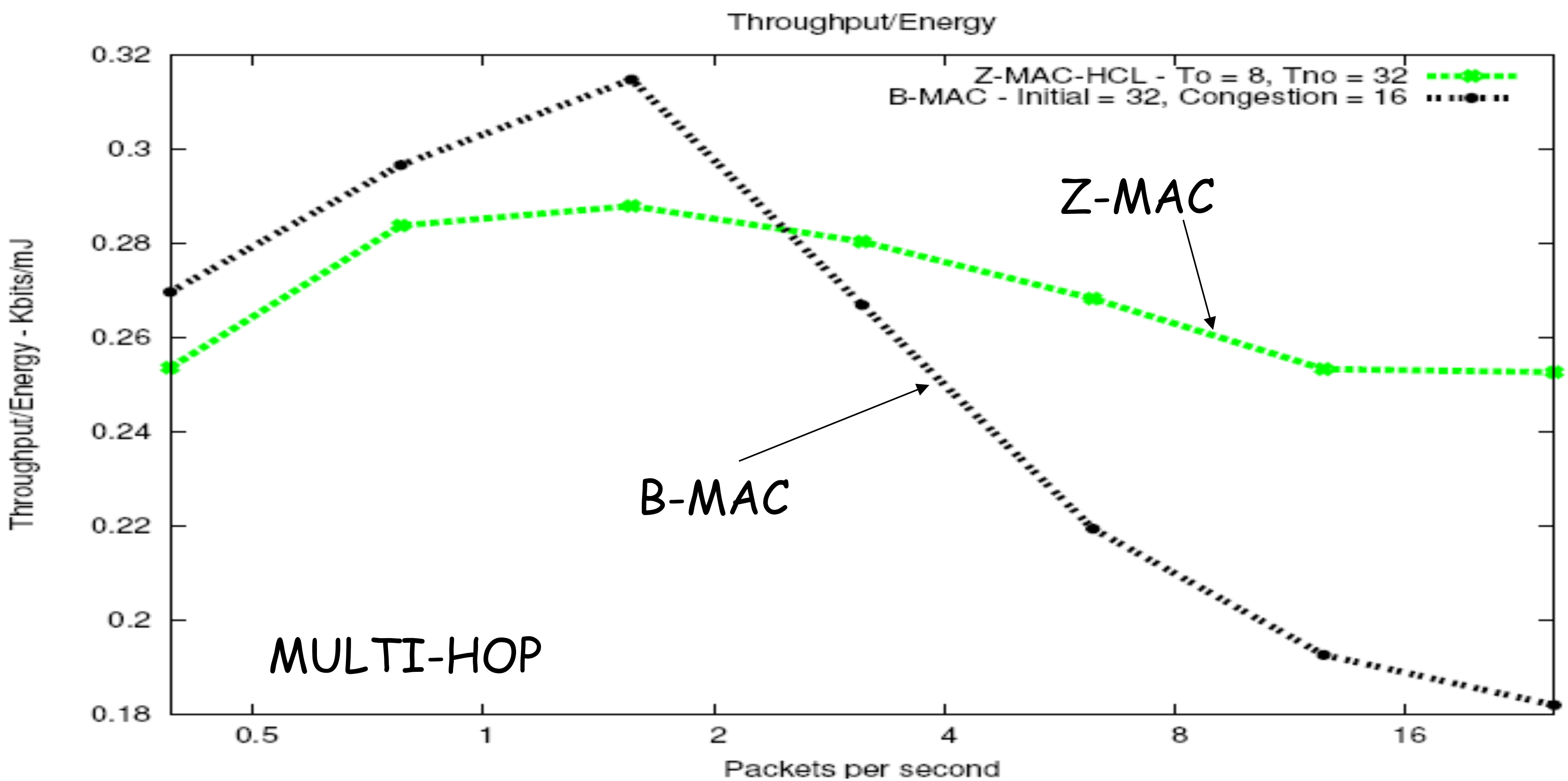
- 40 sensor motes
- Links vary in quality, some have loss rates up to 30-40%
- Asymmetric links also present (14-->15)



Multi Hop Results – Throughput



Multi Hop Results – Energy Efficiency (KBits/Joule)



MAC Protocols for WSNs

- 1. Contention (RANDOM/CSMA)-Based MAC Protocols
  - Sleep-MAC, BMAC, T-MAC, CCMAC, etc…
- 2. Reservation-Based (TDMA BASED) MAC Protocols
  - TRAMA, FLAMA, etc…
- 3. HYBRID (CSMA/TDMA) MAC Protocols
  - ZMAC, ….

Which concept was the most intriguing? (one word)

Total Results: 0

Powered by Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Questions?