Wireless Mobile & Multimedia Networking
7COM1076
WiFi 2

Dr Tazeen Syed
t.s.syed@herts.ac.uk
School of Physics Engineering and Computer Science (SPECS)

Outline

- **Issues in Wireless MAC**
  - Hidden Node Problem
  - Exposed Node Problem
  - CSMA is not suitable
- **MACA**
  - RTS-CTS-DATA
  - Binary Exponential Backoff (BEB)
- **MACAW Message Exchange**
  - ACK-DS-RRTS
- **MACAW Backoff Algorithm**
  - Backoff Copy
  - MILD

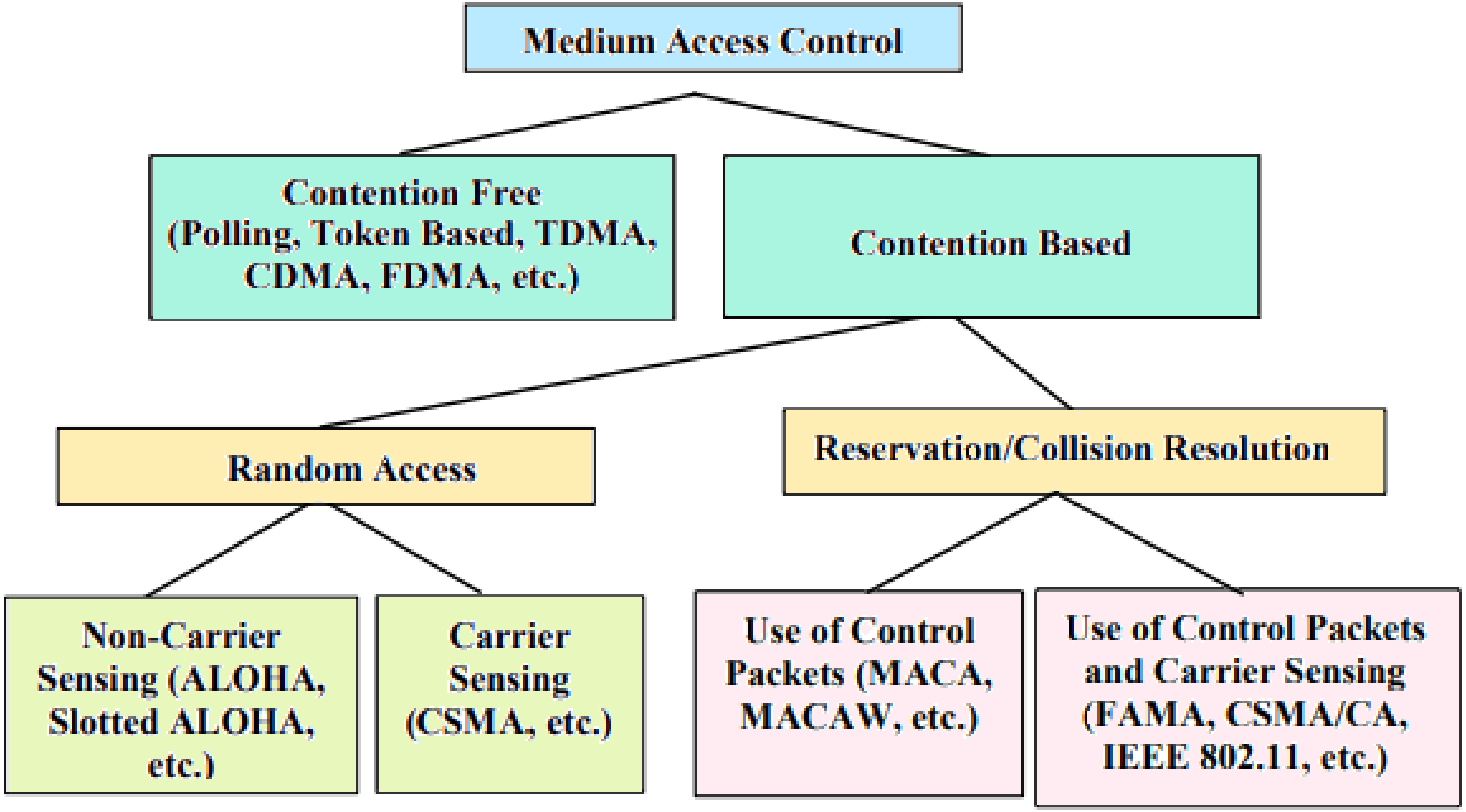
Issues in Wireless MAC

- Limited Bandwidth
- Limited Transmission Range
  - Hidden Node Problem
  - Exposed Node Problem
- Limited Power Supply
- Shared Channel
  - Collision caused by multiple nodes contending for the medium at about the same time

Performance Metrics of Wireless MAC

- Bandwidth Efficiency
- Fairness in bandwidth allocation
- Control overhead
- Effect of hidden and exposed terminal
- Scalability
- Power Consumption

Classification of MAC Schemes



CSMA – Carrier Sense Multiple Access

CSMA is a MAC protocol

- Transmitting node senses the medium
- Transmits data if medium is free
- Defers its transmission if medium is busy
- Collision is avoided by attempting to sense the carrier by testing the signal strength at the vicinity of the carrier
- Collision occurs due to
  - Hidden node problem
  - Exposed node problem

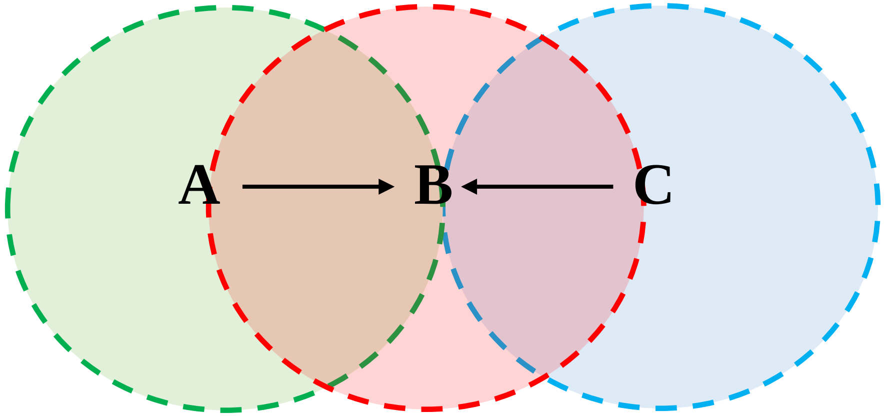
Hidden Node Problem

Node B is in the range of node A and C

Node A and node C can hear node B, but they cannot hear each other as they are out of range “hidden”.

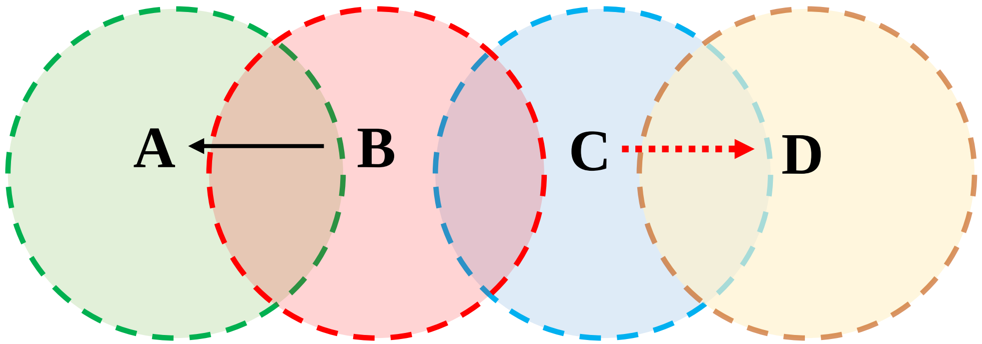
Scenario:

- A sends to B, C cannot hear it, it sense medium is “free”, so C sends to B (CS fails)
- Collision occurs at B
- A and C unaware of the collision (CD fails)
- A is hidden for C
- Collision occurs at the receiver not the sender



Exposed Node Problem

- B is transmitting to A
- C is in range of B
- C can hear B; it senses medium to be busy and waits until B’s transmission to A is completed.
- D is outside the range of A; this waiting is not necessary because collision occurs at receiver side not at sender



MACA protocol

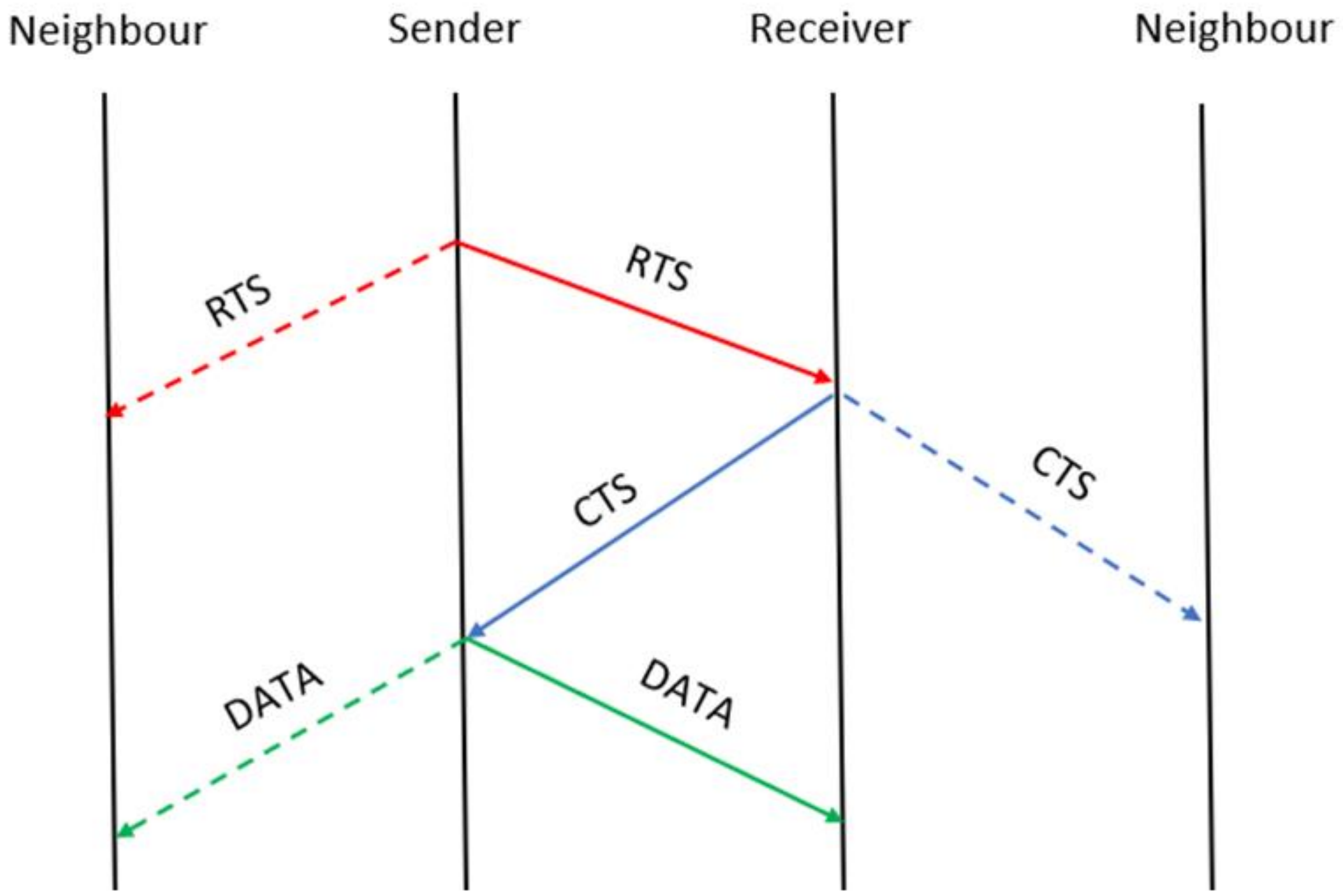
- Multiple Access Collision Avoidance (MACA) was proposed by Karn
- An alternate to the traditional CSMA
  - It does not carrier sense.
  - CSMA senses the state of the channel only at the transmitter

    - leading to hidden node and exposed node problem.
- Virtual Sensing – It "Senses" /probe a channel by short fixed-sized control packets (RTS/CTS)
- Send data packet after the control packets grasp the channel
- Collision avoided for transmission of data packet

Control Packets

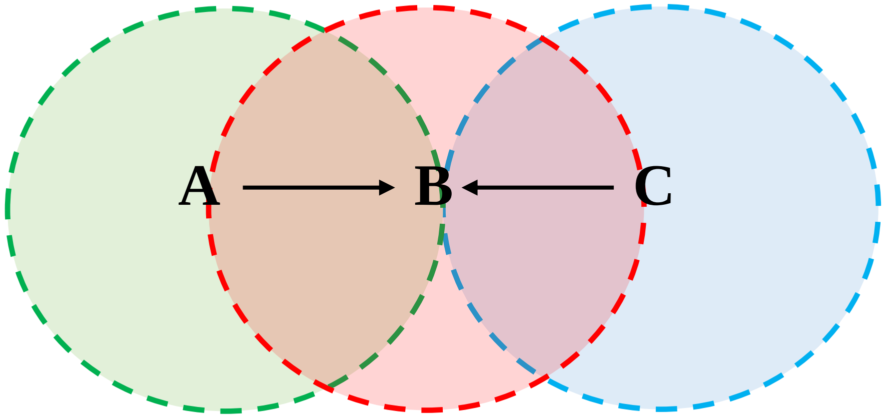
- RTS (Ready to Send)
- CTS (Clear to Send)
  - RTS and CTS carry the expected duration of data transmission

Packet Transmission in MACA



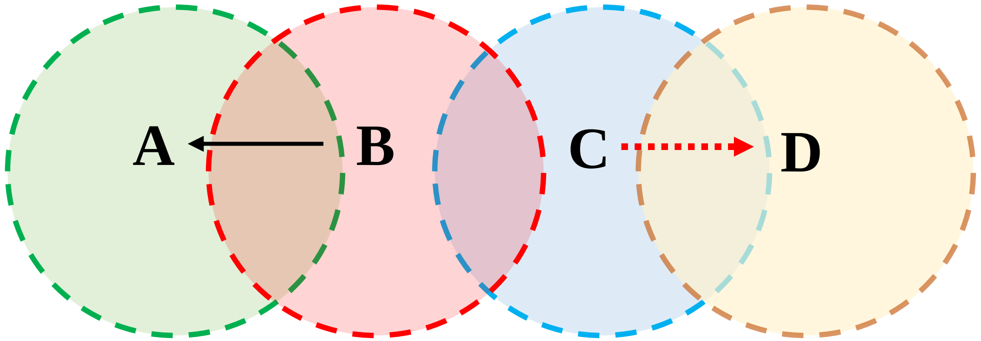
How MACA solves Hidden Node Problem?

- All neighbours of the receiver hear CTS
- Node which hears only CTS message (but not RTS), stops from transmitting (hidden terminal)



How MACA solves Exposed Node Problem?

- All neighbours of the sender hear RTS
- Node which only hears RTS (but not CTS), is free to transmit (exposed terminal)



MACA Protocol

- Did MACA solve the problem?
- Can collision still occur?

Binary Exponential Backoff (BEB)

- When control packets (RTS/CTS) result in collision, MACA uses binary exponential backoff (BEB) to select the retransmission time.
- The backoff is doubled after every collision and reduced to the minimal backoff after every successful RTS/CTS exchange.
- Each node sets a backoff counter locally
  - Initially set the counter to 1
  - After a handshaking failure, increase the counter by doubling the value
  - After a successful handshake, reset the counter to 1
- For example, Assume node A has a backoff counter of 8 and node B has a backoff counter of 16, which node is more likely to win the contention next?

Is MACA good Enough?

- Discussion
  - Bandwidth efficiency
  - Fairness in bandwidth allocation
  - Control overhead
  - Effect of hidden and exposed terminal
  - Scalability
  - Power consumption

MACAW – Multiple Access Collision Avoidance for Wireless

- MACAW: A improved protocol based on MACA
- Problems with MACA:
  - Possible Collision over RTS
  - Fairness issues
    - Backoff algorithm: starvation problem for unlucky stations
    - Less bandwidth for station who want to send more stream
- Q: How to define “fairness” on wireless network?
- Multicast problems
  - What happens when multicasting with RTS/CTS?

MACAW GOAL

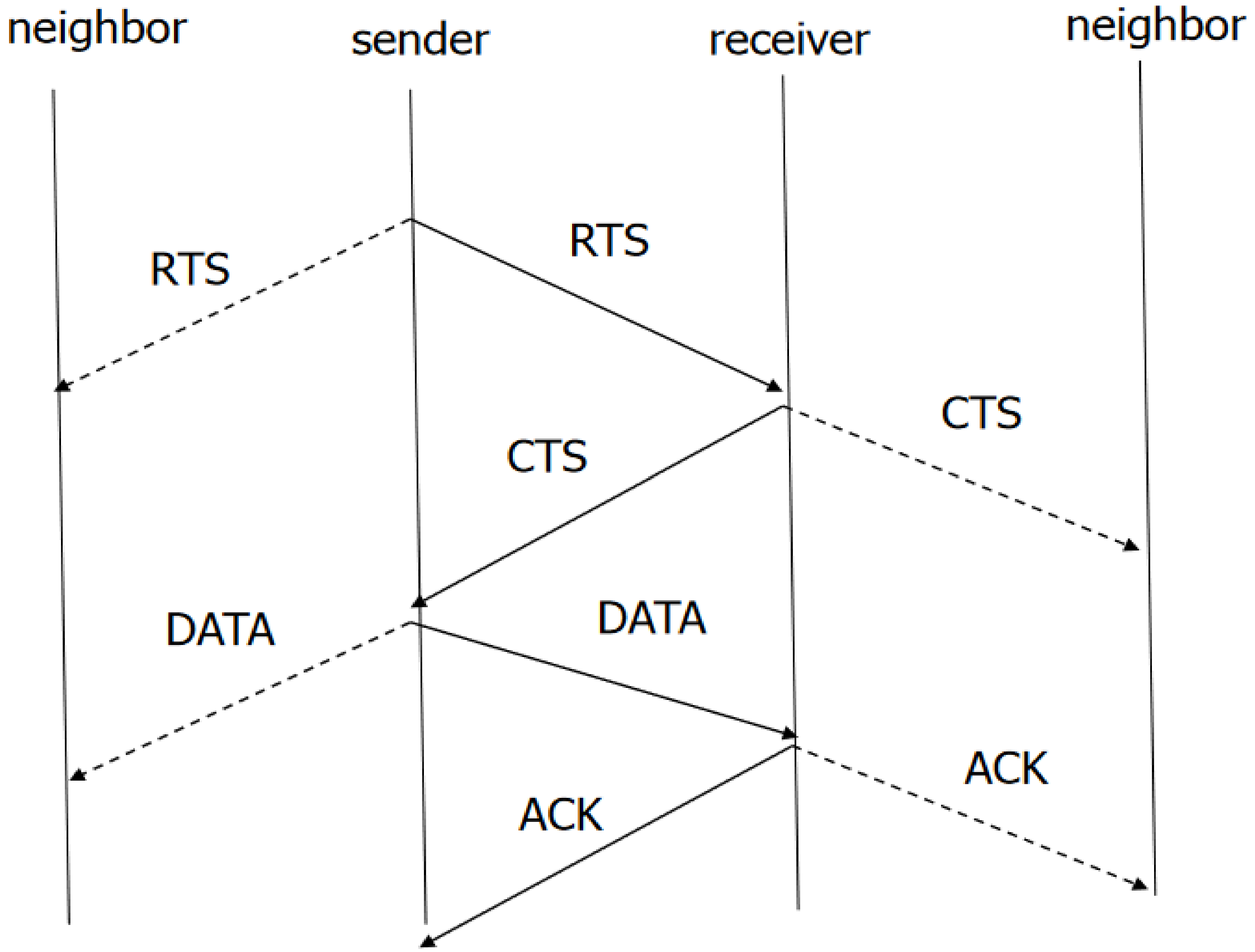
- The Objective of MACAW:
  - Achieve high network utilization (efficiency)
  - Provide fair access to the media (stream-wise)
- Fairness has higher priority over efficiency (throughput)
- Major features of MACAW:
  - Improved message exchange
  - Revised backoff algorithm to avoid starvation
  - Multicast

ACK (Acknowledgement) Packet

RTS—CTS—DATA—ACK

- Receiver sends back ACK as soon as getting data correctly
- Sender retransmits if not receiving ACK
- Sender increases backoff value if not receiving CTS or ACK
- MACAW allows much faster error recovery at the data link layer by using the acknowledgment packet (ACK) that is returned from the receiving node to the sending node as soon as data reception is completed.

Packet transmission with ACK



ACK in MACAW

- Why do MACAW use ACK at MAC layer?
  - In MACA, there was no reliability at MAC layer, it was left to the higher layer to provide reliability
- How does the sender distinguish data loss from ACK loss?
  - If ACK is not received, the sender reschedules the Data packet for transmission.
    - Sender would send RTS
    - How would the receiver respond?
- Using ACK improves fairness or bandwidth efficiency?

Performance of using ACK

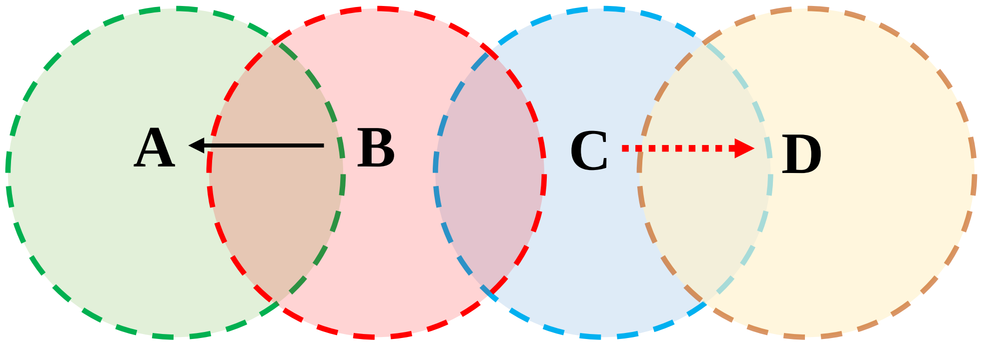
Error Rate	RTS-CTS-DATA	RTS-CTS-DATA-ACK
0	40.41	36.76
0.001	36.58	36.67
0.01	16.65	35.52
0.1	2.48	9.93

The throughput in packets per second achieved by a single TCP data stream between a node and a Base Station in the presence of noise

DS (Data Sending) Packet

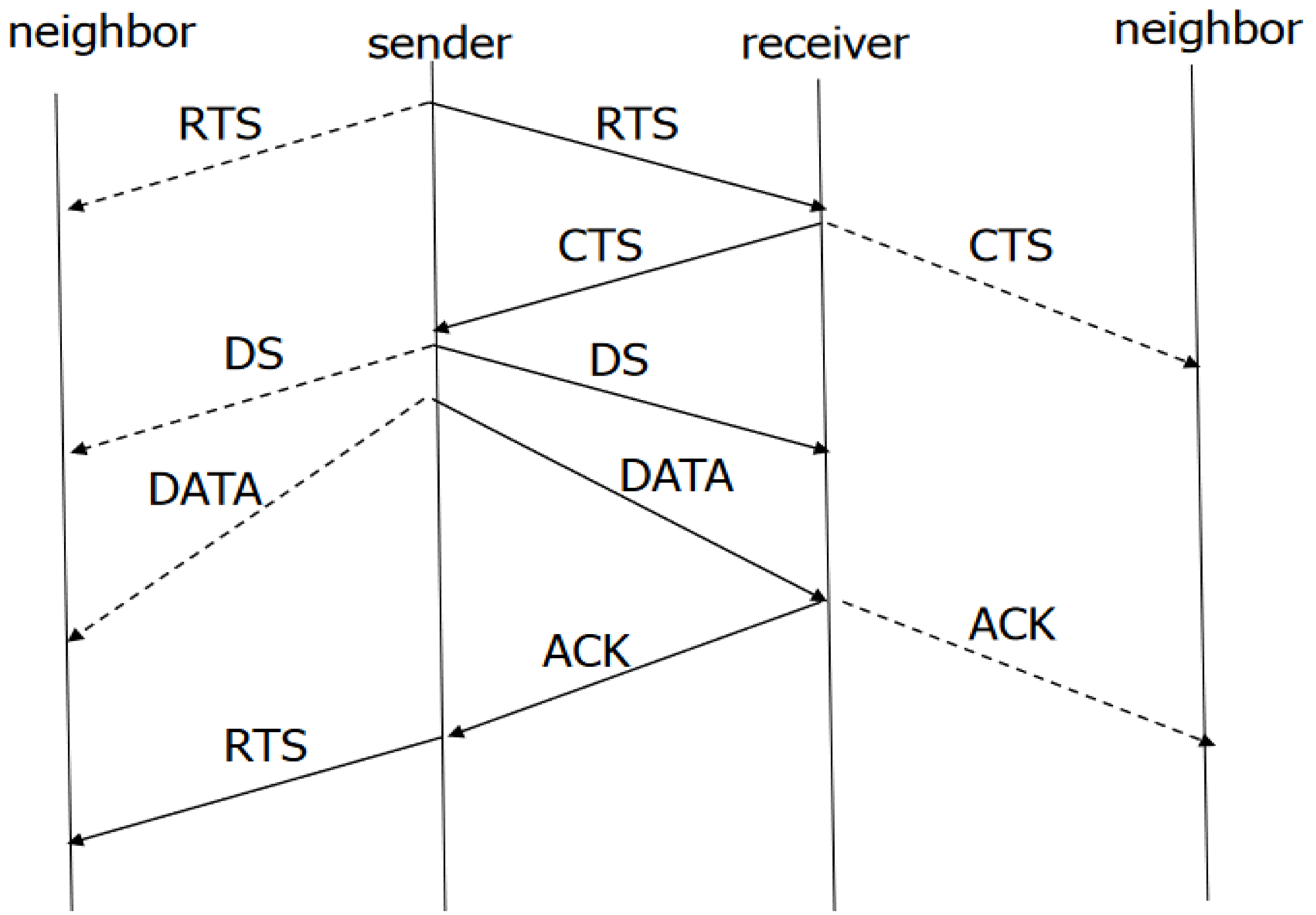
RTS—CTS—DS—DATA—ACK

- DS is a control packet (30- byte) sent before data to tell all neighbours about the transmission length.
- Stations overhearing DS defer all transmissions until after the ACK packet slot has passed.



In the exposed node scenario, node C can hear the RTS message of node B to node A, but it cannot hear the CTS message from node A. At this point, node C does not know if RTS-CTS handshake of node A and Node B was successful. Assuming medium is free, node C will send RTS to node B. But node B does not send CTS reply to node C. Because of this, the backoff counter of node C increases rapidly and will increase congestion at both node C and node B. To solve this issue, node B will send a short **DS packet** before sending Data. Every station which overhears this packet will know that the RTS-CTS exchange was successful and that a data transmission is about to occur; these overhearing stations defer all transmissions until the ACK packet slot

Packet transmission with DS



DS in MACAW

- Why do MACAW use DS packet?

  - In MACA, an exposed node is free to transmit simultaneously when the source node is transmitting packets.
  - But RTS transmission by the exposed node is useless, since the CTS reception will end in collision.
  - Using DS can tell the exposed node when the source node will finish transmission, so the exposed node will wait until the DATA-ACK packet slot has passed.
  - Using DS improves fairness.
  - DS can also be viewed as a way of synchronization.

Performance of Using DS Packet

Considering a two-node configuration where both wireless nodes (P1 & P2) are in range of their respective base stations (B1 & B2) and in range of each other. The nodes are sending data to their base stations, and each stream is generating data at a rate of 64 packets per second and using UDP for transport. Node P2 does not get any chance to access the medium when DS is not used

	RTS-CTS-DATA-ACK	RTS-CTS-DS-DATA-ACK
P1-B1	46.72	23.35
P2-B2	0	22.63

Table shows the throughput in packets per second achieved by the streams with and without the DS packet

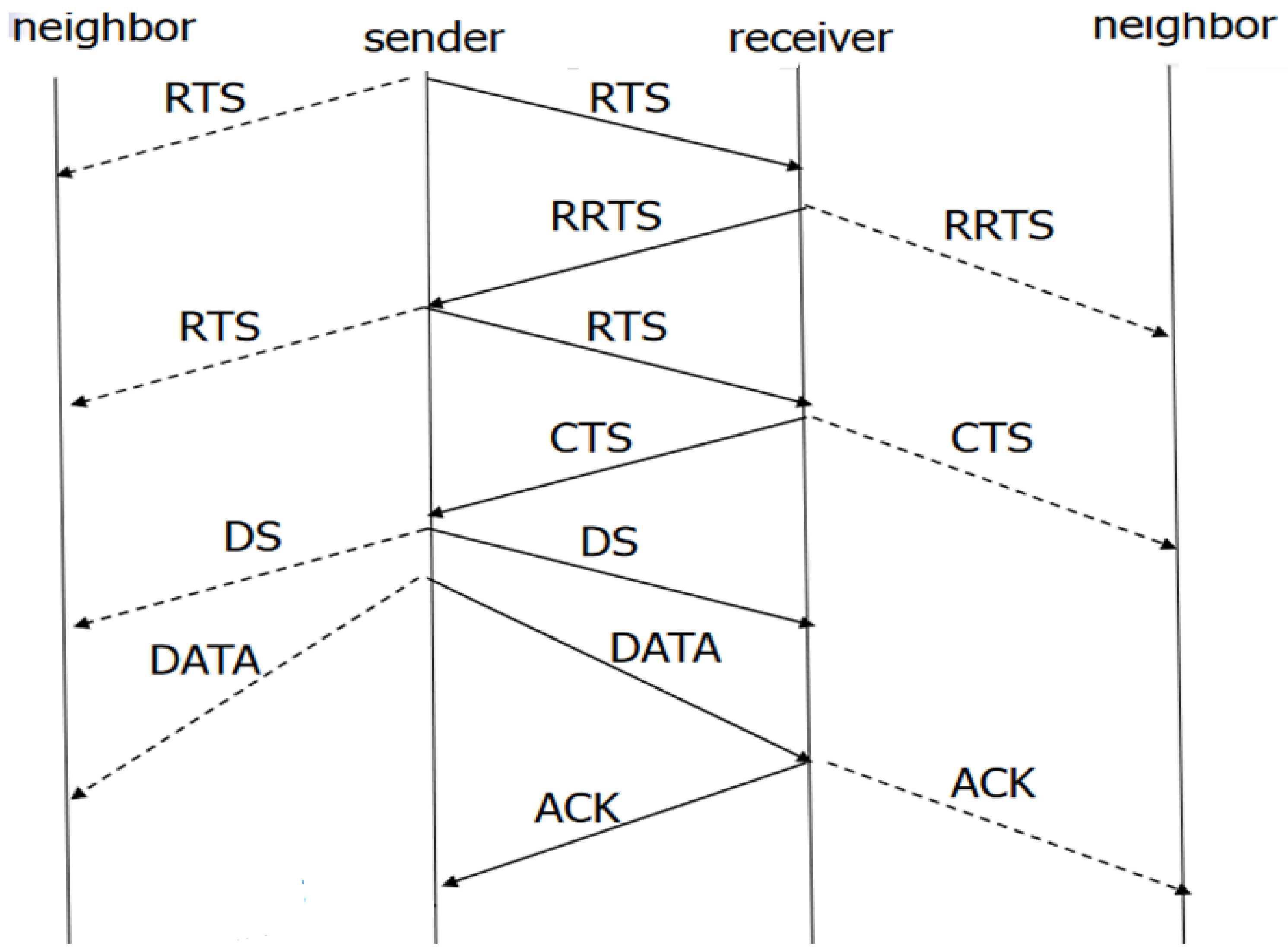
Without DS packet, the exposed node cannot identify when the next contention period (i.e., the slots after the ACK packet and before the next RTS) starts, therefore it is unable to compete effectively for accessing the medium. Usually DATA transmissions are long and the exposed node will transmit its RTS during the middle of an ongoing data transmission which, results in a collision. The node must send its RTS packets during the contention periods, and this requires knowing when the data transmissions start and finish. This need for '**synchronizing**" information is crucial; in this configuration it is supplied by the DS packet (which informs the other stations about the existence and length of the following DATA packet).

RRTS (Request-for-Request-to-send) Packet

RTS—RRTS—RTS—CTS—DS—DATA—ACK

- RRTS is a control packet (30- byte)
- Whenever a station receives an RTS to which it cannot respond (due to deferral), it then contends during the next contention period and sends an RRTS to the sender of RTS
- The recipient of an RRTS immediately responds with an RTS
- Stations overhearing an RRTS defer for two slot

Packet transmission with RRTS



RRTS in MACAW

- Why do MACAW use RRTS packet?
  - RRTS lets the recipient of an RTS to contend the channel on behalf of the sender of the RTS.
  - RRTS can also be viewed as a way of synchronization.
  - Using RRTS improves fairness.

Performance of RRTS in MACAW

From the table, it shows the B1-P1 stream is denied access, while the B2-P2 stream is receiving all of its requested throughput. B1 initiates a data transfer by sending an RTS, the receiving node P1 cannot respond with a CTS because it is deferring to the data transmission to P2. The only way B1 can successfully initiate a transfer is when its RTS happens to arrive during those very short gaps in between a completed data transmission and the completion of P2's next CTS. The key problem is lack of synchronization information

	no RRTS	RRTS
B1-P1	0	20.39
P2-B2	42.87	20.53

The throughput, in packets per second, achieved by the streams

Binary Exponential backoff (BEB) in MACAW

- MACAW uses a backoff algorithm, in which the transmission of RTS messages are delayed for a random number of slots, to reduce the probability of collision and to resolve collisions once they occur.
- The calculation of this random delay is based on the backoff counter.
- The value of the backoff value should therefore reflect the level of contention for the media.
- For a backoff algorithm to be efficient, the backoff counters must accurately reflect the level of contention.
- The backoff algorithm in MACAW plays a crucial role in achieving high overall throughput and fair allocation.

Backoff Copy

- Operations:
  - Include in the packet header of data packet a field containing the current value of the backoff counter.
  - Whenever a station hears a packet, it copies that value into its own backoff counter.
  - After each successful transmission, all nodes have the same backoff counter.
- Advantages:
  - Sharing of congestion information with neighbouring nodes.
  - Using backoff copy improves fairness.

Performance of using Backoff Copy

- The table shows that using BEB copy, the throughput allocation is now completely fair. Thus, having the congestion information disseminated explicitly by the media access protocol produced a fairer allocation of resources.

	BEB	BEB copy
P1-B	48.5	23.82
P2-B	0	23.32

The throughput, in packets per second, achieved by the streams

Backoff MILD (Multiplicative Increase and Linear Decrease)

- In cases of heavy network conditions, backoff strategy is unfair to the already existing nodes that are backing off due to collisions. To improve this situation, MILD algorithm is used in MACAW.
- This scheme however reduces throughput in light traffic conditions.

Operations:

- Basic idea is to adopt a gentler adjustment algorithm
- Upon a collision, the backoff interval is increased by a multiplicative factor (1.5)
- Upon a success it is decreased by 1

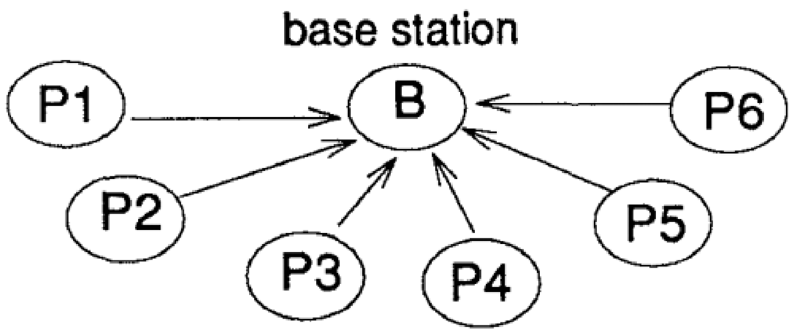
Advantages:

- Using backoff MILD improves bandwidth efficiency.

Performance of using Backoff MILD

A single cell configuration where all stations are in range of each other. All six nodes are sending data to the base station. Each stream is generating data at a rate of 32 packets per second and using UDP for transport. Table shows the data from the two backoff algorithms and illustrates a clear advantage for the MILD algorithm.

	BEB copy	MILD copy
P1-B	2.96	6.10
P2-B	3.01	6.18
P3-B	2.84	6.05
P4-B	2.93	6.12
P5-B	3.00	6.14
P6-B	3.05	6.09



The throughput, in packets per second, achieved by the streams

Comparing Backoff Algorithms

- MACA
  - When CTS is received, reduce BO
    - $BO = BO_{min}$
  - When CTS is not received (collision), increase BO
    - $BO = \min(2*BO, BO_{max})$

- MACAW: add extra field in packet header containing current value of BO (BO_{cur})
  - Every node updates $BO = Bo_{cur}$ upon overhearing new packet
  - Use MILD (Multiple Increase Linear Decrease) to adjust BO:
    - No collision→ $BO = MAX [BO-1, Bo_{min}]$
    - Collision → $BO = MIN [1.5BO, BO_{max}]$

References

- Ad Hoc Wireless Networks, architectures and protocols. C. Siva Ram Murthy and B. S. Manoj, 1st edition.
  - Sections 6.1, 6.2, and their subsections.
- V. Bharghavan, A. Demers, S. Shenker, L. Zhang, “MACAW: A Media Access Protocol for Wireless LANs”, Proc. Of ACM SIGCOMM 1994, pp. 212-225, August 1994.
- Reading list on Canvas

Thank you

Any Questions?

t.s.syed@herts.ac.uk