Chapter 03: MAC Architecture & CSMA/CA

Dr. Mayank Agarwal

Department of CSE
IIT Patna

CS6206 Selected Topics in Wireless Networks

Lecture 3 Learning Objectives

By the end of this lecture, you will be able to:

- Explain why CSMA/CD cannot work in wireless networks
- Describe the CSMA/CA mechanism in detail
- Understand the role of Inter-Frame Spaces (IFS) and backoff timers
- Differentiate between Physical and Virtual Carrier Sense
- Calculate transmission times including all MAC overhead
- Explain the hidden and exposed node problems
- Describe the RTS/CTS mechanism and its trade-offs

The Fundamental Wireless Problem: Why Not CSMA/CD?

- **Ethernet (CSMA/CD):** Listen while transmitting
  - Collision detected $\rightarrow$ Stop $\rightarrow$ Jam $\rightarrow$ Backoff $\rightarrow$ Retry
  - Works because wired signal strength is high
- **Wireless Limitations:**
  1. **Near-Far Problem:** Your own transmission drowns out others
  2. **Hidden Terminal:** Can't hear the colliding transmitter
  3. **Half-Duplex Radios:** Can't transmit and receive simultaneously

Key Insight: In wireless, **collision detection is impossible** while transmitting. Therefore, we must use **Collision Avoidance** instead.

The 802.11 MAC Architecture Overview

- **Two Coordination Functions:**
  1. **Distributed Coordination Function (DCF):** Mandatory, based on CSMA/CA
  2. **Point Coordination Function (PCF):** Optional, poll-based (rarely implemented)
- **Hybrid Coordination Function (HCF):** Modern extension with QoS (802.11e)

DCF is the Foundation:

- Completely distributed (no central coordinator)
- Uses CSMA/CA with random backoff
- Basis for all Wi-Fi communication

Today's Focus: DCF and the basic CSMA/CA mechanism

CSMA/CA: The Three-Phase Process

1. **Listen Before Talk:** Check if medium is idle
2. **Wait Mandatory Time:** Different Inter-Frame Spaces for priority
3. **Backoff If Needed:** Random delay to avoid synchronized collisions

Complete Data Transfer Sequence:

1. Wait DIFS (if medium idle)
2. Perform random backoff
3. Transmit DATA frame
4. Wait SIFS
5. Receive ACK from receiver

Key Concept: Positive acknowledgment required for every frame. No ACK = assume collision or error = retransmit.

Inter-Frame Spaces (IFS): The Priority Mechanism

Different IFS values create priority levels:

- **SIFS (Short IFS):** 10 μs (2.4 GHz) or 16 μs (5 GHz)
  - Highest priority
  - Used for: ACKs, CTS, fragments of same frame
- **PIFS (PCF IFS):** SIFS + 1 slot time
  - Medium priority
  - Used in PCF (rare)
- **DIFS (DCF IFS):** SIFS + 2 slot times
  - Standard priority for data frames
  - Typical: 50 μs (2.4 GHz) or 34 μs (5 GHz)
- **AIFS (Arbitration IFS):** Variable, for QoS (802.11e)

Slot Time: 20 μs (2.4 GHz) or 9 μs (5 GHz) - varies by PHY

Slot Time in 802.11: The Fundamental Clock Tick

- **Definition:** The **Slot Time** is the basic unit of time used in the 802.11 MAC layer's contention protocol (CSMA/CA). It is the atomic interval during which the medium's state (idle or busy) is sampled and around which all waiting and back-off procedures are structured.
- **What It Includes:** The slot time is *not* just an empty time gap. It is calculated to account for the physical realities of the network:
  - **PHY Layer Delay:** Time for the radio to switch from receive to transmit mode.
  - **Propagation Delay:** Maximum time for a signal to travel between the two farthest stations in the network.
  - **MAC Processing Delay:** Time needed to process a signal and decide the channel state.

Analogy: Think of slot time as the "reaction time" for the entire wireless network—the minimum time all devices need to sense and react to each other's transmissions before they can safely attempt to talk.

The Critical Role of Slot Time in CSMA/CA

The Distributed Coordination Function (DCF) uses slot time to orchestrate access:

- **Back-off Procedure:** When a device needs to transmit, it must wait for the channel to be idle for a DIFS period, then choose a random number of slot times for its **back-off counter**.
- **Deferral Mechanism:** The back-off counter only decrements by one for each *empty slot time* observed. If the channel becomes busy, the counter **pauses**. This ensures fair access.
- **Collision Window:** The initial range from which the random back-off number is chosen (the Contention Window) is measured in slot times. A shorter slot time allows for faster resolution of contention.

The Critical Role of Slot Time in CSMA/CA

The Distributed Coordination Function (DCF) uses slot time to orchestrate access:

Action	**Dependency on Slot Time**
Wait after DIFS	Slot time defines the granularity of waiting.
Choose back-off number	Random integer (0 to CW) $\times$ Slot Time.
Defer to other transmissions	Counter pauses/resumes per slot.
Detect collision	Based on no ACK within a slot-based timeout.

PHY-Dependency and The Importance of Accurate Slot Time

- **Not a Fixed Value:** Slot time is **PHY-layer dependent**. Different standards (802.11b, 802.11a/g, 802.11n/ac/ax) use different slot times because their underlying radio characteristics differ.
  - **E.g., 802.11b (DSSS):** 20 µs slot time.
  - **E.g., 802.11a/g (OFDM):** 9 µs slot time.
  - **E.g., 802.11n/ac (HT/VHT OFDM):** 9 µs slot time.
- **Importance #1: Collision Detection & Avoidance** Since true collision detection is impossible in wireless (a device cannot transmit and listen simultaneously), the slot-time-based back-off is the primary method to **probabilistically avoid** collisions. A correctly sized slot time ensures all stations have a consistent view of the medium state before proceeding.

PHY-Dependency and The Importance of Accurate Slot Time

- **Importance #2: Efficiency & Throughput** A shorter slot time reduces the overhead of contention. Networks can resolve "who talks next" more quickly, leaving more time for actual data transmission. This is a key reason 802.11a/g/n/ac are more efficient than 802.11b.
- **Importance #3: Network Scale & Fairness** The slot time must be long enough to account for propagation delay across the entire network. If set too short, stations at the edge may act on an outdated view of the channel, causing hidden-node collisions. It enforces a basic synchronization for fairness.

Impact on Performance and Coexistence

- **Performance Trade-off:** A shorter slot time (like 9 μs) allows faster channel access and higher aggregate throughput but requires a more precisely synchronized PHY layer (like OFDM provides). A longer slot time (like 20 μs) is more robust for slower, longer-range technologies but adds significant contention overhead.
- **Coexistence in Mixed Networks:** When 802.11b (20 μs slot) and 802.11g (9 μs slot) devices share a network, they must use a **protection mechanism** (RTS/CTS or CTS-to-Self) that forces everyone to use the longer, legacy slot times. This ensures fairness but dramatically reduces the efficiency for the newer, faster devices.

Impact on Performance and Coexistence

- **Relationship with Air Time Fairness:** Modern Wi-Fi manages "air time" not just packet count. Slot time is the fundamental quantum of this air time during the contention period. Efficient scheduling algorithms must account for how many slot times are wasted in contention versus used for data.

Conclusion: The slot time is a critical bridge parameter. It is defined by the **PHY layer's capabilities** (switching speed, propagation) but dictates the fundamental rhythm and efficiency of the **MAC layer's contention algorithm**. Getting it right is essential for stability, fairness, and performance across all devices on the network.

Numerical Example 1: IFS Timing Calculation

Problem: In 802.11ac (5 GHz), SIFS = 16 μs, Slot Time = 9 μs.

1. Calculate DIFS duration.
2. If a station waits DIFS then backoff of 3 slots, what's total wait time?

Solution Steps:

1. DIFS = SIFS + (2 $\times$ Slot Time) = 16 + (2 $\times$ 9) = 16 + 18 = 34 μs
2. Total wait = DIFS + (3 $\times$ Slot Time) = 34 + (3 $\times$ 9) = 34 + 27 = 61 μs

Model Answer:

1. DIFS duration = **34 μs**
2. Total wait time = **61 μs**

Observation: In 5 GHz, timing is faster (shorter slot times) than 2.4 GHz, allowing more efficient channel usage.

Carrier Sense: Two Methods

- **Physical Carrier Sense**: Actual RF energy detection
  - PHY layer measures signal strength
  - Compares to Clear Channel Assessment (CCA) threshold
  - Problem: Can't distinguish Wi-Fi from microwave oven
- **Virtual Carrier Sense:** Network Allocation Vector (NAV)
  - Timer maintained by each station
  - Set by duration field in frame headers
  - Counts down while medium is "virtually busy"
  - Even if physically idle, station waits if NAV > 0

Medium is considered "busy" if: Physical Carrier Sense OR Virtual Carrier Sense (NAV > 0)

Network Allocation Vector (NAV): The "Reservation" System

- **Duration Field:** In every frame header, specifies how long medium will be busy
- **NAV Timer:** Each station maintains its own countdown timer
- **Setting NAV:** When station hears a frame, it sets NAV = duration value
- **Using NAV:** Station won't transmit while NAV > 0

Example:

- Station A transmits DATA to AP with duration = 500 μs
- All other stations hear this, set NAV = 500 μs
- AP replies with ACK after SIFS (16 μs)
- Other stations wait until NAV expires

Key Benefit: Solves hidden node problem partially (if all can hear the transmitter)

The Random Backoff Algorithm

- **Purpose:** Avoid synchronized collisions after DIFS
- **Contention Window (CW):** Range for random backoff selection
- **CWmin, CWmax:** Minimum and maximum CW sizes

Algorithm:

1. Choose random backoff = random(0, CW) $\times$ SlotTime
2. Decrement backoff counter while medium idle
3. Freeze counter when medium busy
4. Resume countdown when medium idle for DIFS
5. Transmit when counter reaches 0

Binary Exponential Backoff (like Ethernet):

- Start with CW = CWmin. [Band=2.4 GHz, CWmin=31, CWmax=1023; Band=5 GHz, CWmin=15, CWmax=1023]
- After collision: CW = min(2 $\times$ (CW + 1) - 1, CWmax)
- After successful transmission: Reset CW = CWmin

Numerical Example 2: Backoff Calculation

Problem: 802.11ac has CWmin = 15, CWmax = 1023, Slot Time = 9 μs.

1. What's the initial backoff range (in slots and time)?
2. After 2 collisions, what's the new CW and backoff range?
3. What's the probability two stations pick the same slot?

Solution Steps:

1. Initial: CW = 15, slots = 0 to 15, time = 0 to (15×9) = 0 to 135 μs
2. After 1st collision: CW = $2\times(15+1)-1$ = 31 After 2nd collision: CW = $2\times(31+1)-1$ = 63 Slots = 0 to 63, time = 0 to (63×9) = 0 to 567 μs
3. Probability = 1/(CW+1) = 1/16 6.25% initially

Model Answer:

1. Initial backoff: **0-15 slots (0-135 μs)**
2. After 2 collisions: **CW=63, range 0-63 slots (0-567 μs)**
3. Collision probability: **6.25% initially**

Complete CSMA/CA Timeline Example

Scenario: Station A wants to send 1500-byte frame at 433 Mbps (27.7 μs data time from Lecture 1).

Timeline Calculation:

1. Wait DIFS: 34 μs
2. Random backoff (assume 5 slots): $5 \times 9 = 45$ μs
3. Transmit DATA: PHY preamble (20 μs) + MAC header (36 μs) + data (27.7 μs) = 83.7 μs
4. Wait SIFS: 16 μs
5. Receive ACK: 20 μs (preamble) + 14 μs (ACK frame) = 34 μs

Total Time: $34 + 45 + 83.7 + 16 + 34 = 212.7$ μs

Efficiency: Data time / Total time = 27.7 / 212.7 13%

Key Insight: Massive overhead! This is why frame aggregation (sending multiple packets) was introduced in 802.11n.

DCF Performance Characteristics

- **Fairness:** Approximately fair in long term
  - Each station gets equal share of transmission opportunities
  - Not perfectly fair due to random backoff
- **Efficiency:** Poor for small frames
  - Overhead dominates
  - As low as 10-20% efficiency for VoIP packets
- **Collision Probability:** Increases with number of stations
  - More stations → more contention → more collisions
  - Binary exponential backoff mitigates this
- **Throughput vs Load:** Classic "S-curve"
  - Increases linearly at low load
  - Peaks at optimal load
  - Decreases at high load (thrashing)

Real-World DCF Behavior

- **Capture Effect:** Strong signal "captures" medium
  - Station close to AP has advantage
  - Can lead to unfairness
- **Rate Anomaly:** Low-rate clients hurt everyone
  - 1 Mbps client consumes 10$\times$ more airtime than 10 Mbps client
  - Reduces overall network throughput
- **Performance Anomaly:** All stations get similar throughput
  - Counter-intuitive: Faster stations slowed down
  - Due to fair channel access in time, not data

Example: 1 client at 1 Mbps, 9 clients at 54 Mbps

- Each gets $\sim$10% of airtime
- Fast clients: 10% $\times$ 54 Mbps = 5.4 Mbps each
- Slow client: 10% $\times$ 1 Mbps = 0.1 Mbps
- Total network: $9\times5.4 + 0.1 = 48.7$ Mbps (much less than $9\times54=486$ Mbps!)

Limitations of Basic DCF

1. **No QoS Support**: All traffic treated equally
   - VoIP and file download compete equally
2. **Poor Efficiency**: High overhead per frame
3. **Fairness Issues:** Capture effect, rate anomaly
4. **No Service Differentiation**: Urgent vs. background traffic
5. **Poor Performance in Dense Networks:** Too many collisions

These limitations led to enhancements:

- 802.11e: QoS (EDCA)
- 802.11n: Frame Aggregation (A-MSDU, A-MPDU)
- 802.11ac: MU-MIMO
- 802.11ax: OFDMA, BSS Coloring

Comparison: DCF vs. Switched Ethernet

Characteristic	Wi-Fi (DCF)	Switched Ethernet
Access Method	CSMA/CA	Dedicated link
Collision Handling	Avoidance	Detection
Duplex	Half	Full
Fairness	Temporal	None needed
Efficiency	30-50%	Near 100%
Scalability	10-30 stations	Thousands
Latency	Variable, higher	Low, consistent
Jitter	High	Low

Key Insight: Wi-Fi will never match wired performance, but provides mobility trade-off.

The Hidden Node Problem Revisited

Nodes: A — B — C
Ranges: A hears B, C hears B, but A cannot hear C

Scenario:

1. A transmits to B
2. C cannot hear A (out of range)
3. C thinks medium is idle
4. C also transmits to B
5. Collision at B!

Consequences:

- B receives corrupted data from both
- Neither A nor C knows about collision
- Both will timeout waiting for ACK
- Both will retransmit, likely colliding again

The Exposed Node Problem

Nodes: A — B — C — D
Ranges: All can hear immediate neighbors only

Scenario:

1. B is transmitting to A
2. C hears B's transmission
3. C wants to transmit to D
4. C refrains (medium busy)
5. But C$\rightarrow$D wouldn't interfere with B$\rightarrow$A!

Consequences:

- Unnecessary reduction in spatial reuse
- Wasted capacity
- Both transmissions could occur simultaneously

RTS/CTS: Solving Hidden Node Problem

- **RTS (Request To Send):** Short control frame
- **CTS (Clear To Send):** Short control frame
- **Four-way handshake:** RTS - CTS - DATA - ACK

Mechanism:

1. Sender transmits RTS with duration = time for (CTS + DATA + ACK + $3\times$SIFS)
2. Receiver replies with CTS, echoing duration
3. All stations hearing RTS or CTS set their NAV
4. Hidden nodes hear CTS and defer
5. DATA and ACK proceed with protection

Key Benefit: Hidden nodes hear CTS and defer, even if they didn't hear RTS

RTS/CTS Example with Timeline

Parameters: SIFS = 16 μs, DATA = 100 μs, ACK = 34 μs, RTS = 44 μs, CTS = 38 μs

Duration Calculation in RTS:

- Time needed: CTS + DATA + ACK + 3$\times$SIFS
- $= 38 + 100 + 34 + (3\times16) = 38 + 100 + 34 + 48 = 220\ \mu s$

Complete Timeline:

1. DIFS + Backoff: 34 + (assume 20) = 54 μs
2. RTS: 44 μs
3. SIFS: 16 μs
4. CTS: 38 μs
5. SIFS: 16 μs
6. DATA: 100 μs
7. SIFS: 16 μs
8. ACK: 34 μs

Total: $54 + 44 + 16 + 38 + 16 + 100 + 16 + 34 = 318\ \mu s$

RTS/CTS Trade-offs: When to Use It?

- **Pros:**
  - Solves hidden node problem
  - Reduces collision cost (RTS/CTS frames are smaller than data)
  - Especially useful for large data frames
- **Cons:**
  - Additional overhead (RTS + CTS frames)
  - Can exacerbate exposed node problem
  - More control traffic

RTS Threshold: Configuration parameter

- Frame size > threshold: Use RTS/CTS
- Frame size threshold: Send directly
- Typical default: 2347 bytes (disables RTS/CTS)
- Set lower (e.g., 500 bytes) in environments with hidden nodes

Numerical Example 3: RTS/CTS Efficiency Analysis

Problem: Compare efficiency with/without RTS/CTS for:

- Large frame: 1500 bytes data (150 μs at 80 Mbps)
- Overhead: RTS=44 μs, CTS=38 μs, ACK=34 μs, SIFS=16 μs, DIFS=34 μs

Solution Steps:

Small frame (100B) without RTS:

- Total = DIFS + Backoff(avg=31.5 slots) + Data + SIFS + ACK
- $= 34 + (31.5 \times 9) + 10 + 16 + 34 = 34 + 283.5 + 10 + 16 + 34 = 377.5$ μs
- Efficiency = 10/377.5 2.65%

Small frame with RTS:

- Add RTS + CTS + 2 extra SIFS = 44 + 38 + 32 = 114 μs more
- Total = 377.5 + 114 = 491.5 μs
- Efficiency = 10/491.5 2.03% (worse!)

Numerical Example 3: RTS/CTS Efficiency Analysis

Problem: Compare efficiency with/without RTS/CTS for:

- Small frame: 100 bytes data (10 μs at 80 Mbps)
- Large frame: 1500 bytes data (150 μs at 80 Mbps)
- Overhead: RTS=44 μs, CTS=38 μs, ACK=34 μs, SIFS=16 μs, DIFS=34 μs

Solution Steps:

Large frame (1500B) without RTS:

- Total $= 34 + 283.5 + 150 + 16 + 34 = 517.5\ \mu$s
- Efficiency = 150/517.5 29.0%

Large frame with RTS:

- Total $= 517.5 + 114 = 631.5\ \mu$s
- Efficiency = 150/631.5 23.7% (slightly worse)

Model Answer: For both sizes, RTS/CTS reduces efficiency. But without RTS, hidden nodes could cause collisions requiring retransmissions, making actual throughput even lower. RTS/CTS is an insurance policy against costly collisions.

Fragmentation: Dealing with Noisy Channels

- **Problem:** Long frames more likely to have errors
- **Solution:** Break large frames into smaller fragments
- **Each fragment:** Individually acknowledged
- **Burst transmission:** SIFS between fragments, NAV protects entire burst

Example: 1500-byte frame $\rightarrow$ 3 $\times$ 500-byte fragments

- Each fragment has its own ACK
- If fragment 2 fails, only retransmit fragment 2
- More efficient than retransmitting entire 1500 bytes

Fragmentation Threshold: Like RTS threshold

- Frame size > threshold $\rightarrow$ Fragment
- Trade-off: More fragments = more headers = more overhead

Power Save Mode: Basic Mechanism

- **Problem:** Battery-powered devices (phones, IoT)
- **Solution:** Sleep most of the time, wake periodically

Original PS Mode (Legacy):

1. Client tells AP "I'm going to sleep" (sets Power Management bit)
2. AP buffers frames for sleeping client
3. Client wakes every **Listen Interval** (e.g., 100 ms)
4. Checks **Traffic Indication Map (TIM)** in beacon
5. If AP has buffered frames, client sends PS-Poll to retrieve them

Limitations: High latency, inefficient for periodic traffic

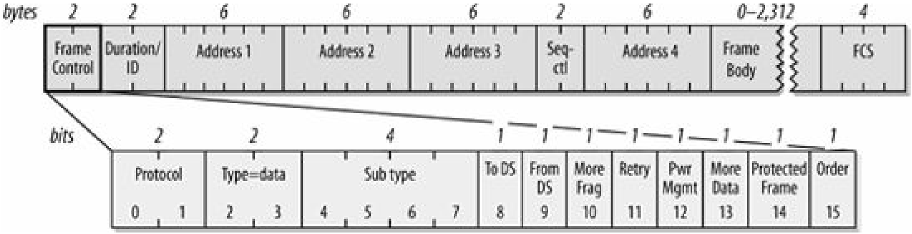
Enhanced Power Save Mechanisms

- **Automatic Power Save Delivery (APSD - 802.11e):**
  - Scheduled wake times
  - Better for VoIP, streaming
- **Target Wake Time (TWT - 802.11ax):**
  - AP schedules exact wake times for each client
  - Clients can sleep longer between beacons
  - Critical for IoT battery life

Power Save Trade-off:

- **More sleep:** Better battery life, higher latency
- **Less sleep:** Lower latency, worse battery life
- **TWT**: Optimizes both by precise scheduling

MAC Frame Formats: Diagram



MAC Frame Formats: Three Types

- **Management Frames:** Association, authentication, beacons, de-authentication, dis-Association
  - Frame Control, Duration, Address 1-3, Sequence Control, Frame Body, FCS
  - 24-byte MAC header minimum
- **Control Frames:** RTS, CTS, ACK, PS-Poll
  - Shorter, typically 10-14 bytes
  - No Frame Body
- **Data Frames:** Carry upper-layer data
  - Can be combined with management/control (Data+CF-Ack, etc.)
  - Address fields vary by To/From DS bits

Key Fields:

- **Duration/ID:** Sets NAV, power save info
- **Address 1-4:** Receiver, Transmitter, Source, Destination
- **Sequence Control:** Fragmentation and sequence numbers
- **FCS:** Frame Check Sequence (CRC-32)

Address Fields Demystified

To DS	From DS	Addr 1	Addr 2	Addr 3
0	0	DA	SA	BSSID
0	1	DA	BSSID	SA
1	0	BSSID	SA	DA
1	1	RA	TA	DA

Where:

- DA = Destination Address (ultimate receiver)
- SA = Source Address (original sender)
- BSSID = AP's MAC address
- RA = Receiver Address (next hop)
- TA = Transmitter Address (previous hop)

Common Case (STA to AP): To DS=1, From DS=0

- Addr1 = BSSID (AP)
- Addr2 = SA (STA)
- Addr3 = DA (ultimate destination)

Inter-BSS Communication Scenario

Network Setup:

- BSS1: STA1 associated with AP1
- BSS2: STA2 associated with AP2
- AP1 and AP2 connected via the Distribution System (DS)

Communication Path:

$$\text{STA1} \rightarrow \text{AP1} \rightarrow \text{DS} \rightarrow \text{AP2} \rightarrow \text{STA2}$$

Key Point:

- 4-address MAC frames are used **only inside the DS**

STA1 to AP1 Transmission (To DS = 1)

Frame Characteristics:

- Data frame
- To DS = 1, From DS = 0
- 3-address format

Address Fields:

- Address 1 (Receiver): AP1 (BSSID)
- Address 2 (Transmitter): STA1
- Address 3 (Destination): STA2
- Address 4: Not used

Interpretation:

- STA1 sends data to AP1
- Final destination is STA2 in another BSS

AP1 to AP2 Forwarding (To DS = 1, From DS = 1)

Frame Characteristics:

- Data frame within Distribution System
- To DS = 1, From DS = 1
- 4-address format

Address Fields:

- Address 1 (Receiver): AP2
- Address 2 (Transmitter): AP1
- Address 3 (Destination): STA2
- Address 4 (Source): STA1

Why 4 Addresses?

- Preserves original source and final destination
- Required for AP-to-AP forwarding

AP2 to STA2 Transmission (From DS = 1)

Frame Characteristics:

- Data frame
- To DS = 0, From DS = 1
- 3-address format

Address Fields:

- Address 1 (Receiver): STA2
- Address 2 (Transmitter): AP2 (BSSID)
- Address 3 (Source): STA1
- Address 4: Not used

Interpretation:

- AP2 delivers data to STA2
- STA2 identifies STA1 as the source

Summary of Addressing Across the Path

Addressing Summary:

- STA $\leftrightarrow$ AP communication uses **3-address frames**
- AP $\leftrightarrow$ AP communication uses **4-address frames**
- 4-address mode carries:
  - Receiver
  - Transmitter
  - Original Source
  - Final Destination

Key Takeaway:

- 4-address frames enable transparent inter-BSS communication in IEEE 802.11

Summary: Key MAC Concepts

1. **CSMA/CA:** Listen, wait DIFS, random backoff, transmit, wait ACK
2. **Carrier Sense:** Physical (RF energy) + Virtual (NAV timer)
3. **NAV:** Reservation system via duration field
4. **Hidden Node:** Transmitters can't hear each other, collide at receiver
5. **Exposed Node:** Unnecessary deferral, reduces spatial reuse
6. **RTS/CTS:** Four-way handshake protects against hidden nodes
7. **Fragmentation:** Smaller frames for error-prone channels
8. **Power Save:** Sleep/wake cycles for battery devices

Reading & Preparation for Next Lecture

- **Required Reading:**
  - Textbook (Gast): Chapter 3 - "802.11 MAC Fundamentals"
  - Review the DCF algorithm steps
- **Optional Reading:**
  - IEEE 802.11-2020: Clause 9 (MAC sublayer)
  - Bianchi's DCF performance model papers
- **Next Lecture (Lecture 4): Frame Formats & Basic Operations**
  - Detailed frame structure
  - Authentication and association process
  - Scanning mechanisms
  - Be ready to analyze frame captures

Review Questions

1. Why can't wireless use CSMA/CD like Ethernet?
2. Explain the difference between physical and virtual carrier sense.
3. How does the NAV timer help with the hidden node problem?
4. When should you enable RTS/CTS? What's the trade-off?
5. Why does a 1 Mbps client significantly reduce throughput for 54 Mbps clients?
6. Calculate: DIFS=34μs, SIFS=16μs, Slot=9μs, Backoff=7 slots. How long before transmission starts?
7. What problem does fragmentation solve, and what's its downside?

Discussion Question: "If you were designing MAC from scratch today, what would you do differently than 802.11 DCF? Consider modern use cases like IoT, dense deployments, and mixed traffic types."

Laboratory Exercise Preview (Related to This Lecture)

Exercise: MAC Layer Analysis with Wireshark

- **Objective:** Analyze 802.11 MAC behavior in real captures
- **Tools:** Wireshark with 802.11 dissector, Wi-Fi adapter in monitor mode
- **Tasks:**
  - Capture and identify different frame types
  - Measure IFS timing between frames
  - Observe NAV setting from duration fields
  - Detect RTS/CTS usage
  - Calculate efficiency from actual traces
- **Deliverable:** Analysis report with annotated captures

Learning Outcome: Connect theoretical MAC concepts to real protocol behavior.

Appendix: Key MAC Parameters by Standard

Parameter	**802.11b (2.4GHz)**	**802.11a/g (2.4/5GHz)**	**802.11n/ac (5GHz)**
Slot Time (μs)	20	9 / 20	9
SIFS (μs)	10	10 / 16	16
DIFS (μs)	50	28 / 34	34
CWmin	31	15	15
CWmax	1023	1023	1023
Max Frame Size	2346	2346	7955*
Frag Threshold	2346	2346	7955
RTS Threshold	2347	2347	2347

Common Configuration Mistakes

1. **RTS Threshold too low:** Excessive overhead for small frames
2. **RTS Threshold too high:** Hidden node collisions for large frames
3. **Fragmentation Threshold too low:** Too much header overhead
4. **Fragmentation Threshold too high:** Excessive retransmissions
5. **CWmin too small:** More collisions in dense networks
6. **CWmin too large:** Excessive delays in sparse networks
7. **Disabling RTS/CTS entirely:** Hidden node problems persist
8. **Using RTS/CTS always:** Unnecessary overhead most of the time

Best Practice: Defaults are reasonable for most cases. Adjust based on measured problems.

Historical Note: Why These Values?

- **Slot Time:** Based on worst-case propagation delay + PHY processing
  - 20 μs in 2.4GHz = round trip for 3km! (Very conservative)
  - 9 μs in 5GHz = more aggressive, assumes indoor distances
- **SIFS:** Time for radio to switch from Rx to Tx
  - Modern radios faster, but backward compatibility maintains values
- **CWmin=15:** Compromise between collision probability and delay
  - Theoretical analysis shows near-optimal for moderate contention
- **Legacy Constraints:** Many values frozen for interoperability

Observation: Modern standards (ax, be) can't change these without breaking backward compatibility. Instead, they add new mechanisms (OFDMA, MLO) that work alongside DCF.

Thank you!