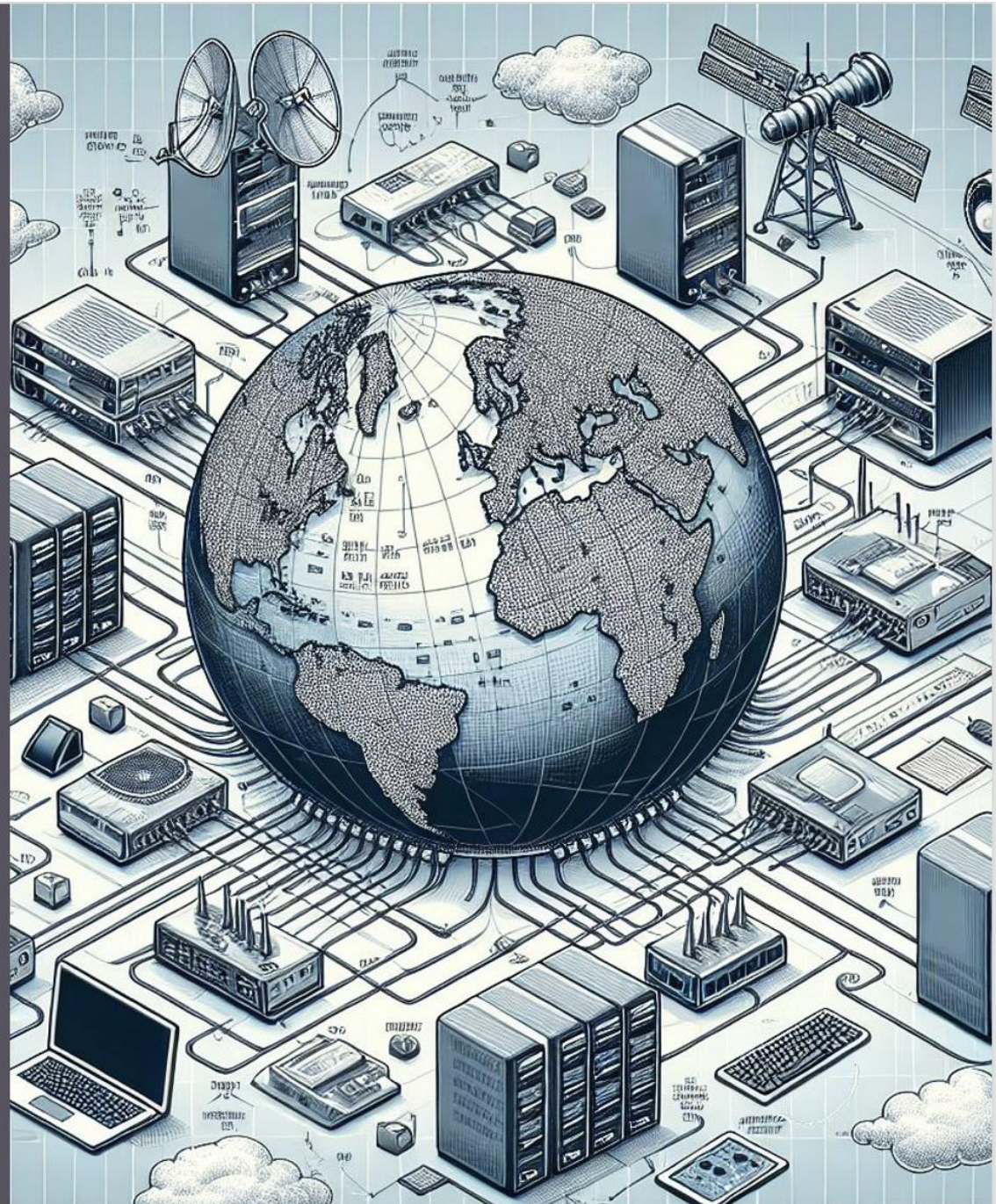
UAB COLLEGE OF ARTS AND SCIENCES
The University of Alabama at Birmingham
Department of Computer Science

CS 334/534 NETWORKING

Dr. Ragib Hasan

Lecture 4.2:
WiFi



Lecture goals

- Learn more about WiFi
- WiFi Collision Avoidance
- Hidden Nodes Problem
- Wifi Rate Scalling

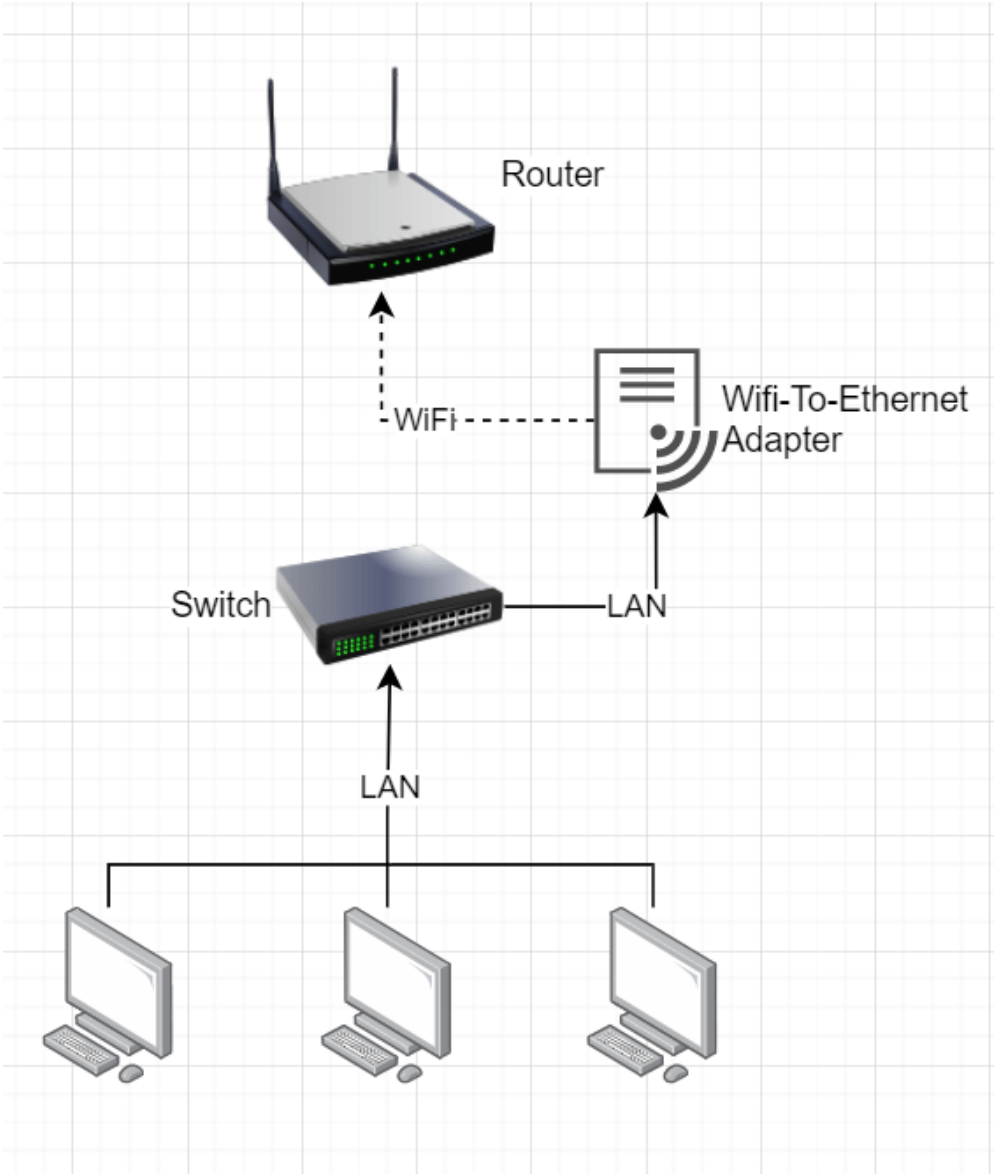
- Book reference: Chapter 4, section 4.2.1 to 4.2.2

WiFi

- Wi-Fi is a wireless networking technology that **uses radio waves** to provide **wireless high-speed Internet access**.
- IEEE wireless-networking protocols in the **802.11 family** (802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, 802.11ax)
- It uses **special control and management** packets within the Wi-Fi LAN layer to handle radio environment challenges.

Interoperability with Ethernet

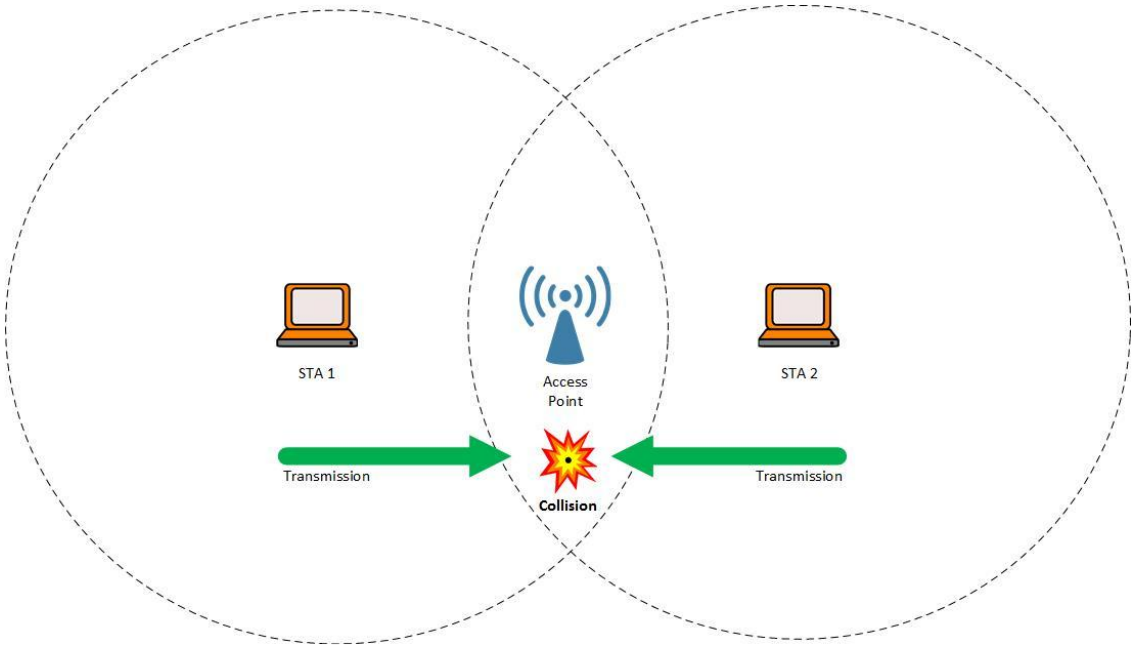
- Wi-Fi is designed to be fully compatible with Ethernet
- Wi-Fi **MAC addresses are 48-bit**, like Ethernet.
- Shares the **same address namespace** as Ethernet.
- **Switches can forward packets** between Wi-Fi and Ethernet seamlessly..



Wi-Fi versions

IEEE name	maximum bit rate	frequency	channel width	new name
802.11a	54 Mbps	5 GHz	20 MHz	
802.11b	11 Mbps	2.4 GHz	20 MHz	
802.11g	54 Mbps	2.4 GHz	20 MHz	
802.11n	65-150 Mbps	2.4/5 GHz	20-40 MHz	Wi-Fi 4
802.11ac	78-867 Mbps	5 GHz	20-160 MHz	Wi-Fi 5
802.11ax	Up to 1200 Mbps	2.4/5+ GHz	20-160 MHz	Wi-Fi 6

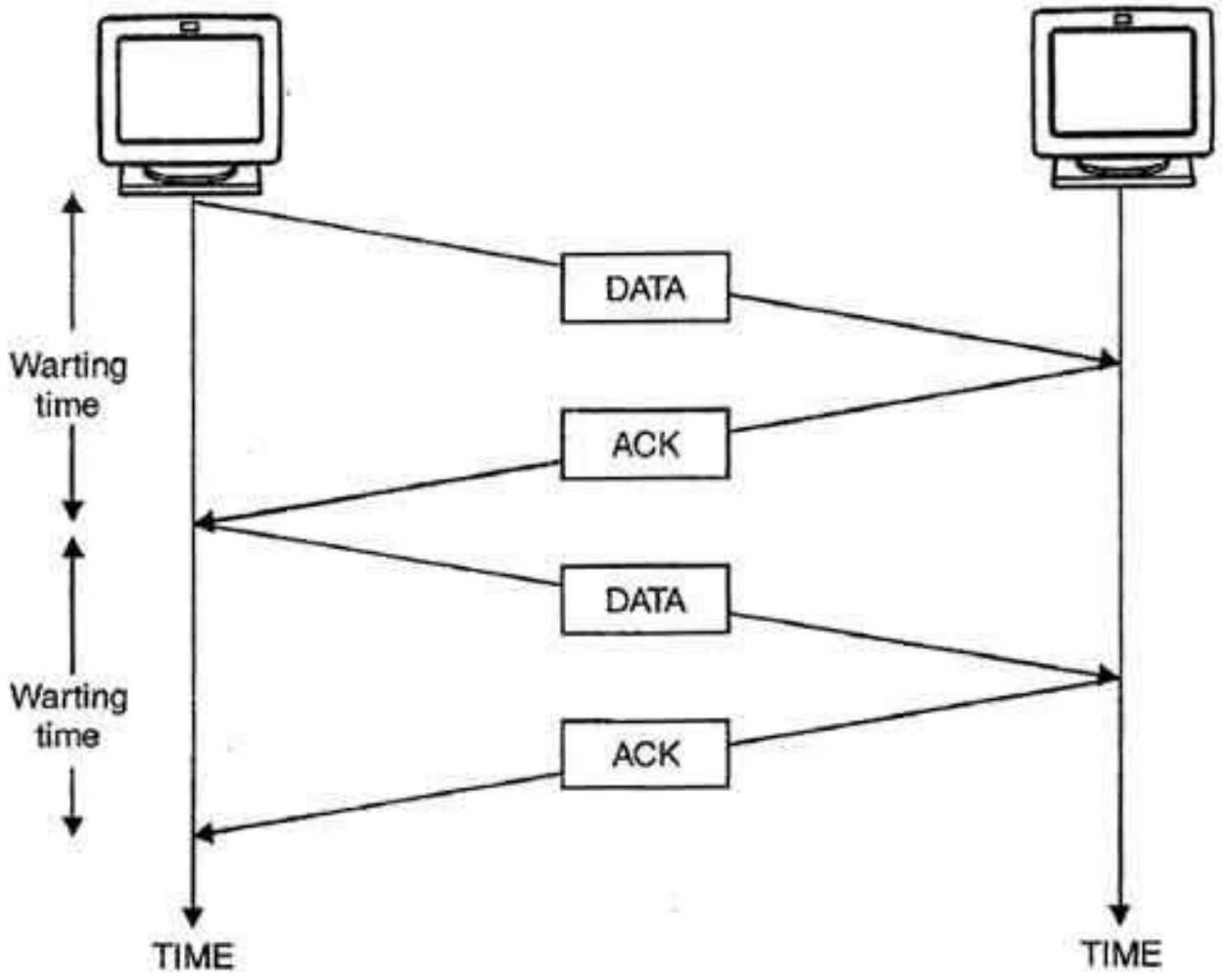
WiFi and Collision



- Unlike Ethernet, Wi-Fi **cannot detect collisions in progress.**
- If another device sends data at the same time, the Wi-Fi sender **won't notice anything wrong**, but its signal won't reach the receiver.
- It makes its backoff and retransmission **algorithms more complex**.
- **Collision-Free Mode** exists but is **rarely used.**
- **Collision management** remains **a critical factor** in Wi-Fi performance.

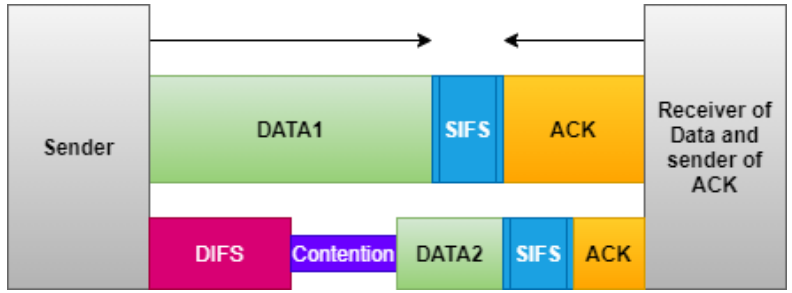
Collision Handling

- After receiving a packet, the receiver sends an **ACK (Acknowledge) packet** to the sender to confirm successful delivery.
- If no ACK is received, the sender **assumes a collision**.
- Other causes of packet loss (e.g., interference, distance) are ignored.
- Assuming collisions leads to slight delays but **ensures retransmission**.



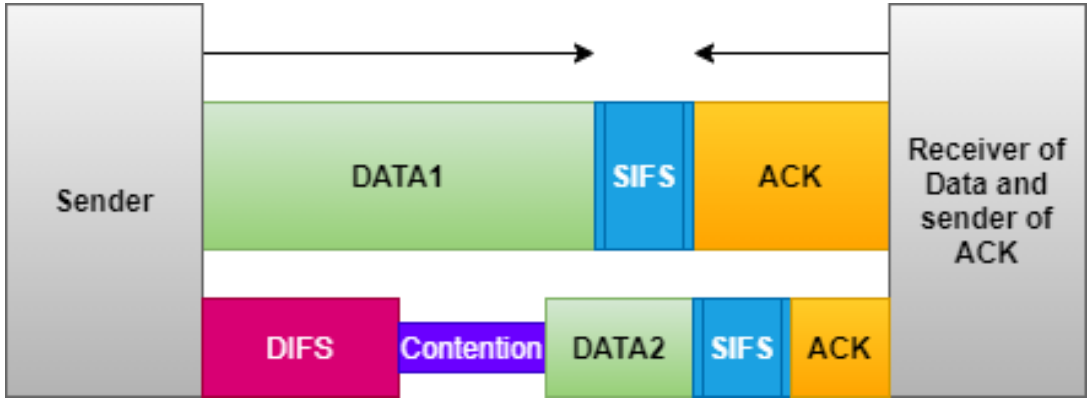
Collision Handling

- Timing Intervals for ACKs:
  - Slot Time: 20 µsec
  - IFS (InterFrame Spacing): 50 µsec
  - SIFS (Short IFS): 10 µsec
- IFS is more formally known as DIFS (Distributed IFS)
- After getting a data packet, the receiver waits SIFS time before sending an ACK.
- Other stations wait a longer IFS time before transmitting, preventing interference.
- RTT (Round Trip Time) between two Wi-Fi stations 100 meters apart is less than 1 µsec.



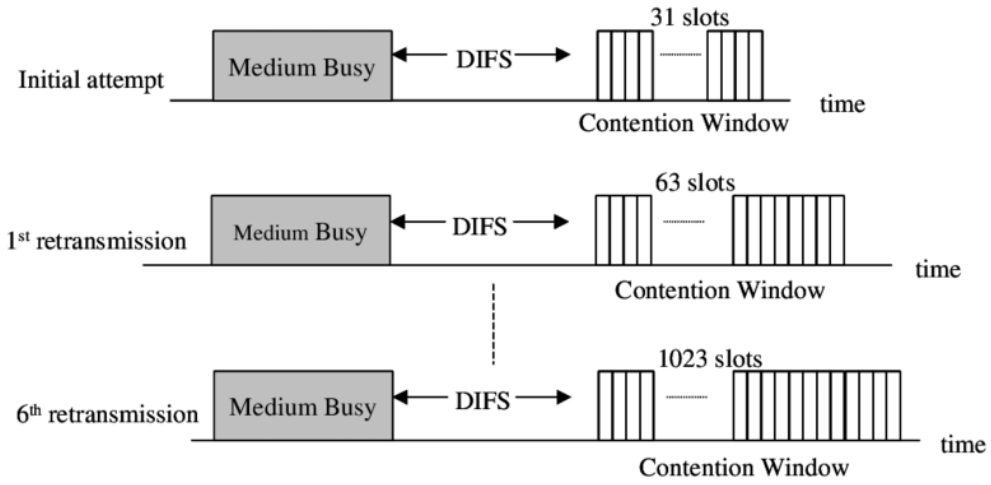
Avoiding Collisions in Wi-Fi

- Before sending data, a device checks if the channel is idle.
- If the channel is free for IFS time, the device transmits immediately.
- If the channel is busy, the device waits using exponential backoff.



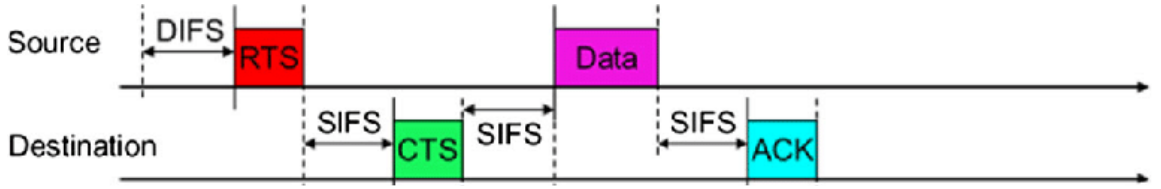
Exponential Backoff in Wi-Fi

- First backoff range: random number between 0-31 slot times.
- If a collision happens: backoff range doubles → 64, 128, 256, 512, 1024 (max).
- After 7 failed attempts, the packet is discarded, and the sender restarts.



RTS/CTS in Wi-Fi

- Wi-Fi can use **Request-to-Send (RTS) / Clear-to-Send (CTS)** to avoid collisions.
- Mainly used for **large data packets** to prevent wasted bandwidth.
- Often **disabled by default** because many packets are small.
- Ensures **smoother network performance** in crowded areas.

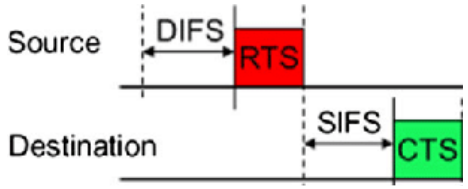


How RTS/CTS Works

- **RTS (Request-to-Send)**
  - The sender sends a small RTS packet to the receiver.
  - It includes the receiver's identity and the data size.

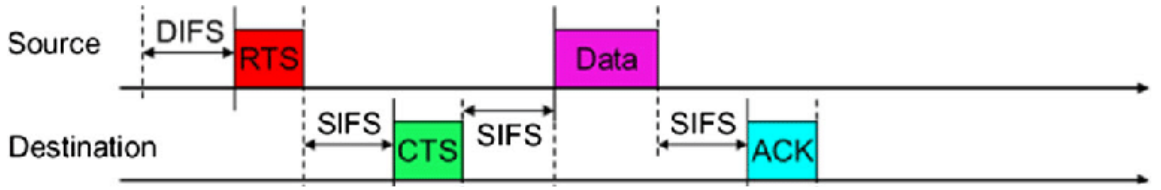
How RTS/CTS Works

- **RTS (Request-to-Send)**
  - The sender sends a small RTS packet to the receiver.
  - It includes the receiver's identity and the data size.
- **CTS (Clear-to-Send)**
  - The receiver waits SIFS (Short InterFrame Space) and sends CTS.
  - CTS informs all nearby devices to pause transmission.



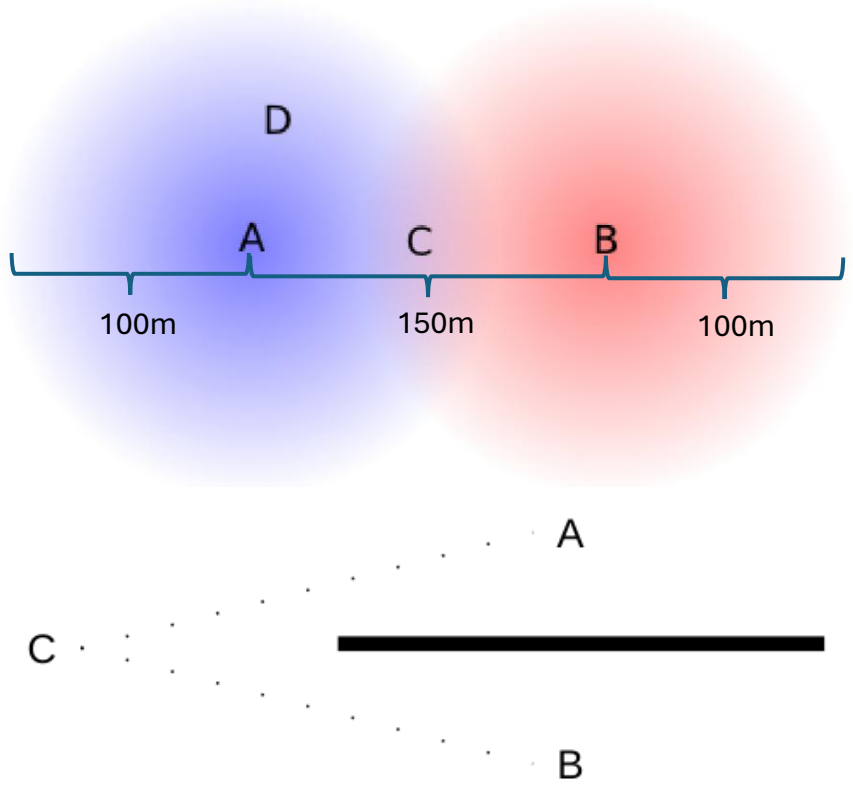
How RTS/CTS Works

- **RTS (Request-to-Send)**
  - The sender sends a small RTS packet to the receiver.
  - It includes the receiver's identity and the data size.
- **CTS (Clear-to-Send)**
  - The receiver waits SIFS (Short InterFrame Space) and sends CTS.
  - CTS informs all nearby devices to pause transmission.
- **Data Transmission**
  - The sender waits SIFS, then sends the full data packet.
  - Since RTS/CTS was successful, no collisions should happen.



Hidden-Node Problem

- A and B are not in range of one another
- If A is transmitting and B senses the medium, B will find the medium is idle and will start sending.
- If A and B transmit simultaneously then there will be a collision at C.
- C receives nothing usable.
- Neither A nor B can possibly detect this.
- The general scenario is known as the hidden-node problem.
- D receives only A's signal, and so no collision occurs at D.



How RTS/CTS protocol solves Hidden-Node Problem

- A sends an RTS packet to C

How RTS/CTS protocol solves Hidden-Node Problem

- A sends an RTS packet to C
- C responds with CTS

How RTS/CTS protocol solves Hidden-Node Problem

- A sends an RTS packet to C
- C responds with CTS
- B has not heard the RTS packet from A, but does hear the CTS from C

How RTS/CTS protocol solves Hidden-Node Problem

- A sends an RTS packet to C
- C responds with CTS
- B has not heard the RTS packet from A, but does hear the CTS from C
- A will begin transmitting after a SIFS interval

How RTS/CTS protocol solves Hidden-Node Problem

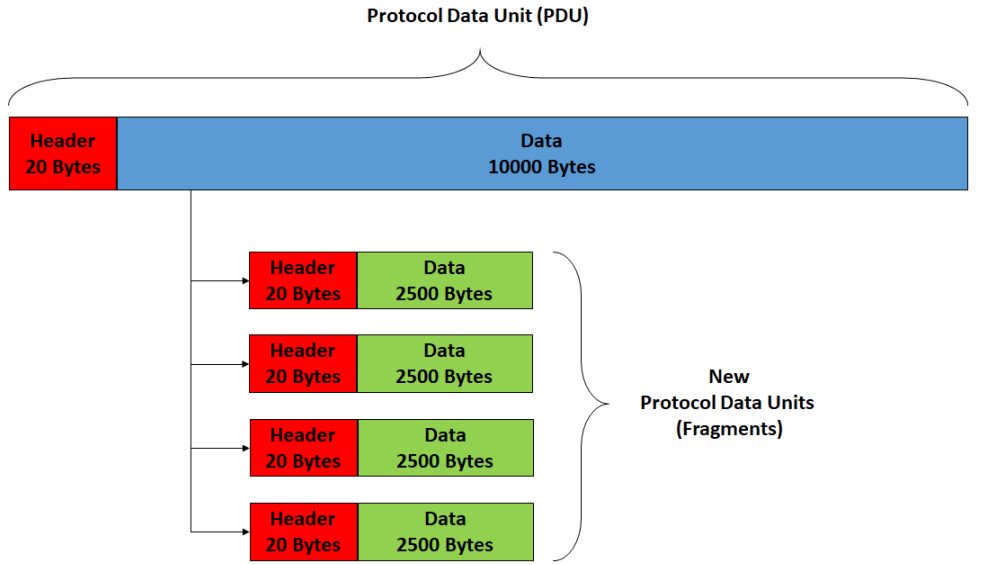
- A sends an RTS packet to C
- C responds with CTS
- B has not heard the RTS packet from A, but does hear the CTS from C
- A will begin transmitting after a SIFS interval
- B waits because CTS includes the data packet size, signaling how long the channel will be occupied

How RTS/CTS protocol solves Hidden-Node Problem

- A sends an RTS packet to C
- C responds with CTS
- B has not heard the RTS packet from A, but does hear the CTS from C
- A will begin transmitting after a SIFS interval
- B waits because CTS includes the data packet size, signaling how long the channel will be occupied
- RTS packets are short, reducing their chances of collision compared to data packets.

Wi-Fi Fragmentation

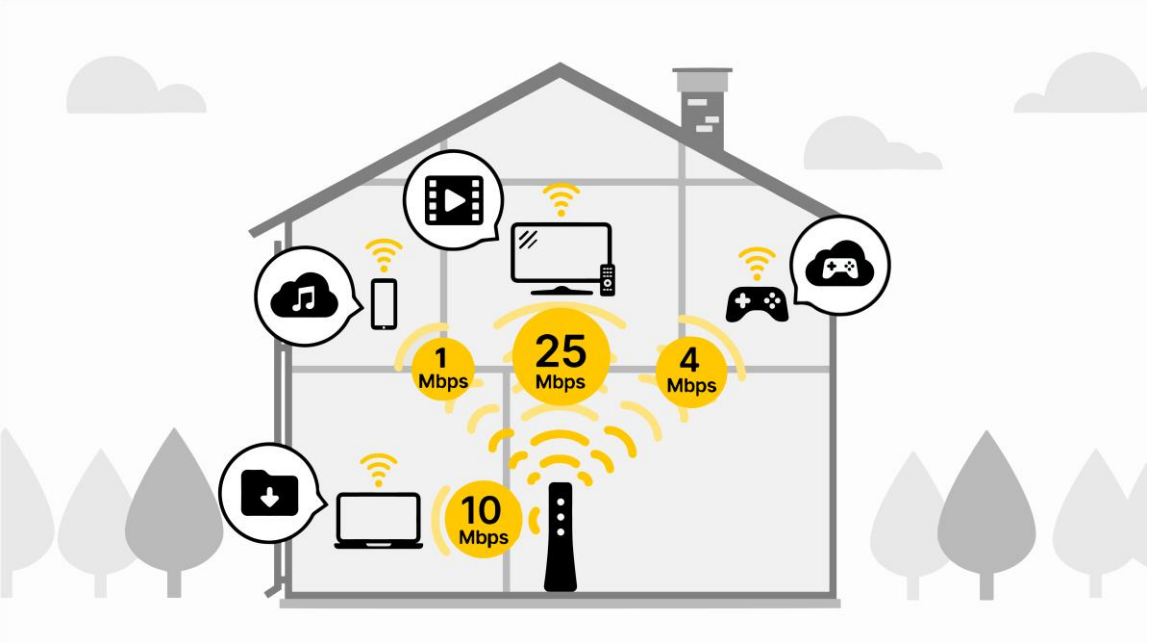
- Wi-Fi fragmentation splits large packets into smaller fragments.
- Each fragment gets its own link-layer ACK.
- Helps in high error or collision environments.
- Reduces retransmission overhead if errors occur.
- Improves transmission efficiency.
- Fragments are reassembled at the receiving node.



Wi-Fi Rate Scaling

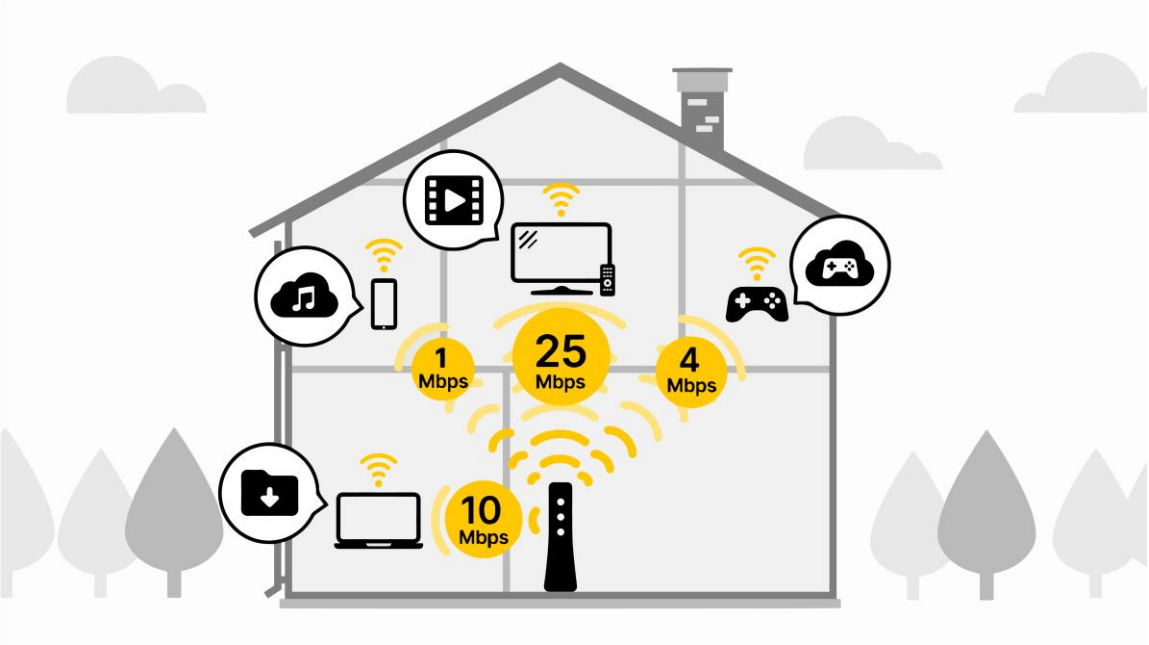
Rate scaling (or rate control) allows Wi-Fi senders to **adjust transmission bit rates** based on connection quality:

- Lower bit rates **reduce noise-related errors, improving reliability.**
- Rates are adjusted based on **distance and interference**.
- Goal: **Maximize throughput**, not just speed.



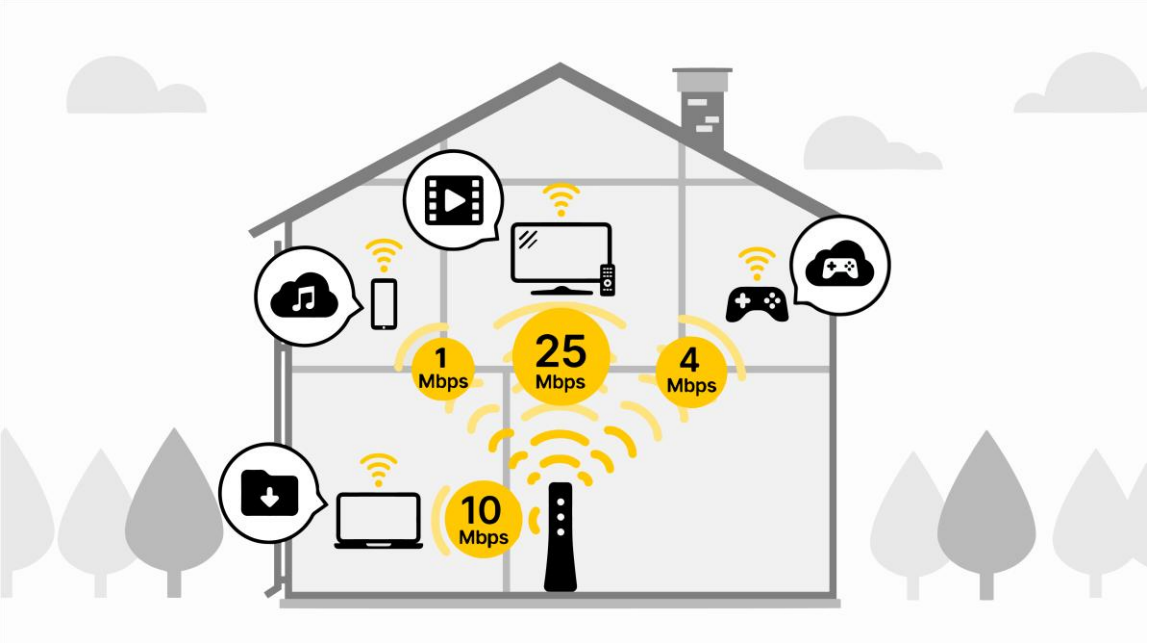
Wi-Fi Rate Scaling

- 802.11g standard rates: 54, 48, 36, 24, 18, 12, 9, and 6 Mbps
- Example:
  - 36 Mbps with 25% loss → 27 Mbps effective
  - Better than 24 Mbps with no loss.



Rate Scaling Implementation

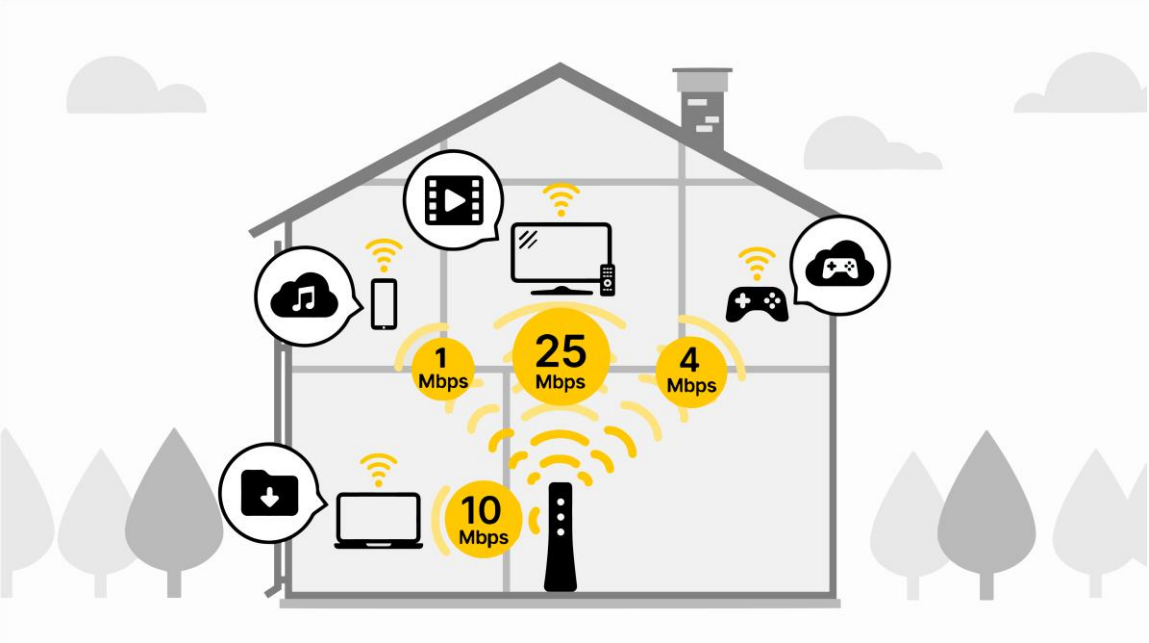
- Updates can occur on a **per-packet basis**
- Different **rates** can be used for **different recipients**
- Rate selection algorithms are **implemented in Wi-Fi drivers**
- Different nodes may **use different algorithms**



Automatic Rate Fallback (ARF)

The earliest rate-scaling algorithm:

- Rate decreases after two consecutive transmission failures
- Rate increases after ten transmission successes
- Failure is determined by missing link-layer ACK



Advanced Rate Adaptation Algorithms

- **Receiver-Based Auto Rate (RBAR)**
  - Uses signal-to-noise ratio (SNR) to select rate.
  - Less effective since Wi-Fi reports signal strength, not SNR.
- **Collision-Aware Rate Adaptation (CARA)**
  - Detects collisions by monitoring busy channel after SIFS.
  - Works best when hidden-node problem isn’t present