1. A-MSDU (Aggregate MAC Service Data Unit)

A-MSDU aggregates multiple **MSDUs (MAC Service Data Units)** into a single **MAC frame** for transmission. An MSDU is typically an Ethernet frame (or higher-layer packet, e.g., IP packet) passed to the MAC layer for processing. A-MSDU combines these MSDUs into one frame with a single **MAC header**, reducing overhead by sharing the header across multiple payloads.

- **MAC Header**: A single MAC header (e.g., 30 bytes in 802.11n) containing source/destination addresses, QoS fields, and sequence control.
- **A-MSDU Subframes**: Multiple MSDU subframes, each with:
  - **Subframe Header**: 14 bytes, including:
    - **Destination Address (DA)**: 6 bytes (recipient's MAC address).
    - **Source Address (SA)**: 6 bytes (sender's MAC address).
    - **Length**: 2 bytes (size of the MSDU payload).
  - **MSDU Payload**: The actual data (e.g., Ethernet frame, up to 2304 bytes).
  - **Padding**: 0–3 bytes to align each subframe to a 4-byte boundary.
- **Frame Check Sequence (FCS)**: A 4-byte CRC for the entire A-MSDU, ensuring integrity.
- **Maximum Size**: Up to 7935 bytes (802.11n) or 11,467 bytes (802.11ac/ax, depending on AP/station capabilities).

Operation:

1. **Aggregation**:
   - The sender (station or AP) collects multiple MSDUs destined for the same recipient, typically from the same **Traffic Identifier (TID)** (QoS Access Category, e.g., AC_VI for video).
   - The MSDUs are encapsulated into A-MSDU subframes, each with a subframe header.
   - A single MAC header and FCS are added to form the A-MSDU.
2. **Transmission**:
   - The A-MSDU is transmitted as a single MAC frame within a **PHY Protocol Data Unit (PPDU)**, with one PHY header.
   - The recipient processes the A-MSDU, extracting each MSDU using the subframe headers.
3. **Acknowledgment**:
   - Typically used with **Block ACK** to acknowledge individual MSDUs, as a single frame loss affects the entire A-MSDU (see limitations).
   - In some cases, a traditional ACK is used for small A-MSDUs.

Advantages:

- **Reduced MAC Header Overhead**: A single MAC header serves multiple MSDUs, significantly lowering overhead compared to sending individual frames (e.g., 30 bytes vs. 30 bytes per MSDU).
- **Efficient for Small Frames**: Ideal for aggregating small packets (e.g., TCP ACKs, VoIP packets), as it minimizes header redundancy.

Limitations:

- **Error Propagation**: If any part of the A-MSDU is corrupted (e.g., due to interference), the entire frame is discarded, as there's only one FCS for the whole frame.
- **No Selective Retransmission**: All MSDUs must be retransmitted if the A-MSDU fails, reducing efficiency in noisy environments.
- **Processing Overhead**: The recipient must parse and de-aggregate subframes, increasing computational complexity.

Use Cases:

- **Small Packet Traffic**: Aggregating short frames (e.g., TCP ACKs, DNS queries, VoIP packets) in low-error environments.
- **High-Throughput Applications**: Streaming or file transfers with consistent packet sizes and stable channels.
- **Low-Density Networks**: Environments with minimal interference, where error rates are low.

---

2. A-MPDU (Aggregate MAC Protocol Data Unit)

A-MPDU aggregates multiple **MPDUs (MAC Protocol Data Units)** into a single **PHY frame** for transmission. An MPDU is a complete MAC frame (MAC header + payload + FCS). A-MPDU combines multiple MPDUs, each with its own MAC header and FCS, into one transmission with a single **PHY header**, reducing PHY overhead while maintaining individual frame integrity.

Structure:

- **PHY Header**: A single PHY header and preamble for the entire A-MPDU.
- **A-MPDU Subframes**: Multiple MPDU subframes, each with:
  - **Delimiter**: 4 bytes, including:
    - **Length**: Size of the MPDU.
    - **Signature**: Unique pattern to identify the delimiter.
    - **CRC**: 8-bit check for delimiter integrity.
  - **MPDU**: A complete MAC frame, including:
    - **MAC Header**: 30 bytes (addresses, QoS, sequence number).
    - **Payload**: Data (e.g., MSDU or fragment).
    - **FCS**: 4-byte CRC for the individual MPDU.

    - **Padding**: 0–3 bytes to align subframes to a 4-byte boundary.
- **Maximum Size**: Up to 65,535 bytes (802.11n) or 1,048,576 bytes (802.11ac/ax), depending on AP/station capabilities and TXOP duration.

Operation:

1. **Aggregation**:
    - The sender collects multiple MPDUs, which may belong to the same or different TIDs, destined for the same recipient.
    - Each MPDU is encapsulated with a delimiter to form an A-MPDU subframe.
    - The A-MPDU is transmitted within a single PPDU, with one PHY header.
2. **Transmission**:
    - The recipient processes the A-MPDU, using delimiters to separate MPDUs and checking each MPDU's FCS for integrity.
    - Corrupted MPDUs are identified without affecting others, as each has its own FCS.
3. **Acknowledgment**:
    - A **Block ACK** is used to acknowledge the A-MPDU, with a bitmap indicating which MPDUs were received correctly.
    - Lost MPDUs are selectively retransmitted in the next TXOP, preserving efficiency.

Advantages:

- **Error Isolation**: Each MPDU has its own FCS, so a corrupted MPDU does not affect others, allowing selective retransmission via Block ACK.
- **High Reliability**: Robust in noisy environments, as only lost MPDUs are retransmitted, minimizing data loss.
- **Flexible Aggregation**: Supports MPDUs with different TIDs, destinations (within limits), or QoS requirements, increasing aggregation opportunities.
- **Scalability**: Handles large bursts (up to 64 MPDUs or more), ideal for high-throughput applications.
- **Integration with Block ACK**: Optimized for Block ACK, reducing acknowledgment overhead and improving efficiency.

Limitations:

- **Higher MAC Overhead**: Each MPDU includes a full MAC header (e.g., 30 bytes), increasing overhead compared to A-MSDU for small packets.
- **Delimiter Overhead**: Delimiters (4 bytes per MPDU) add minor airtime usage, though less significant than multiple PHY headers.
- **Processing Complexity**: The recipient must parse delimiters and process multiple FCS checks, increasing computational load.
- **Buffer Requirements**: Large A-MPDUs require significant buffering at the recipient, especially in high-throughput scenarios.

Use Cases:

- **High-Throughput Applications**: 4K video streaming, large file transfers, or gaming in 802.11n/ac/ax networks.
- **Noisy Environments**: Channels with interference, where selective retransmission is critical.
- **Dense Networks**: Scenarios with diverse traffic, leveraging Block ACK and EDCA for QoS.

---

3. A-MSDU in A-MPDU (Hybrid Aggregation)

A-MSDU in A-MPDU is a hybrid aggregation technique where multiple **A-MSDUs** are encapsulated as payloads within an **A-MPDU**. This combines the benefits of A-MSDU's reduced MAC header overhead with A-MPDU's error isolation and selective retransmission, optimizing efficiency for specific traffic patterns.

- **PHY Header**: A single PHY header for the entire A-MPDU.
- **A-MPDU Subframes**: Multiple A-MPDU subframes, each containing:
    - **Delimiter**: 4 bytes (Length, Signature, CRC).
    - **MPDU**: A complete MAC frame, where the payload is an **A-MSDU**:
        - **MAC Header**: Single MAC header for the MPDU.
        - **A-MSDU**: Contains multiple MSDU subframes, each with:
            - **Subframe Header**: 14 bytes (DA, SA, Length).
            - **MSDU Payload**: Data (e.g., Ethernet frame).
            - **Padding**: 0–3 bytes for alignment.
        - **FCS**: 4-byte CRC for the entire MPDU (covering the A-MSDU).
    - **Padding**: 0–3 bytes for alignment.
- **Maximum Size**:
    - Each A-MSDU is limited to 7935/11,467 bytes.
    - The A-MPDU can be up to 65,535/1,048,576 bytes, depending on the standard and TXOP.

Operation:

1. **Aggregation**:
    - The sender creates multiple A-MSDUs, each aggregating MSDUs with the same TID and destination.
    - Each A-MSDU is encapsulated as the payload of an MPDU, with its own MAC header and FCS.
    - The MPDUs are combined into an A-MPDU, with delimiters and a single PHY header.
2. **Transmission**:
    - The A-MPDU is transmitted within a PPDU.

    - The recipient processes the A-MPDU, using delimiters to separate MPDUs and FCS to check each MPDU's integrity.
    - For each valid MPDU, the recipient de-aggregates the A-MSDU to extract individual MSDUs.
3. **Acknowledgment**:
    - A **Block ACK** acknowledges the A-MPDU, with a bitmap indicating which MPDUs (and their A-MSDUs) were received correctly.
    - Lost MPDUs (containing A-MSDUs) are retransmitted, but a corrupted A-MSDU within an MPDU requires retransmission of all its MSDUs.

Advantages:

- **Combined Benefits**:
    - Inherits A-MSDU's reduced MAC header overhead for small packets within each A-MSDU.
    - Inherits A-MPDU's error isolation and selective retransmission, as each MPDU has its own FCS.
- **Optimized for Mixed Traffic**: Balances efficiency for small packets (via A-MSDU) and reliability for large bursts (via A-MPDU).

Limitations:

- **Partial Error Propagation**: While A-MPDU isolates errors at the MPDU level, a corrupted A-MSDU within an MPDU requires retransmission of all its MSDUs, reducing efficiency compared to pure A-MPDU.
- **Increased Complexity**: Combining A-MSDU and A-MPDU processing requires significant computational resources for aggregation, de-aggregation, and error handling.
- **Buffer Requirements**: Large A-MPDUs with multiple A-MSDUs demand substantial buffering at the recipient.

Use Cases:

- **Mixed Traffic Scenarios**: Networks with both small packets (e.g., TCP ACKs, VoIP) and large data bursts (e.g., video streaming, file transfers).
- **High-Throughput Networks**: 802.11ac/ax environments requiring maximum efficiency and reliability.
- **Moderately Noisy Environments**: Channels where A-MPDU's selective retransmission mitigates errors, but A-MSDU's header savings are beneficial.