## A-MPDU Aggregation

Problems in the creation of the A-MPDU frames

#### Our objective and context

To create a "fake" AP using a wireless device in monitor mode

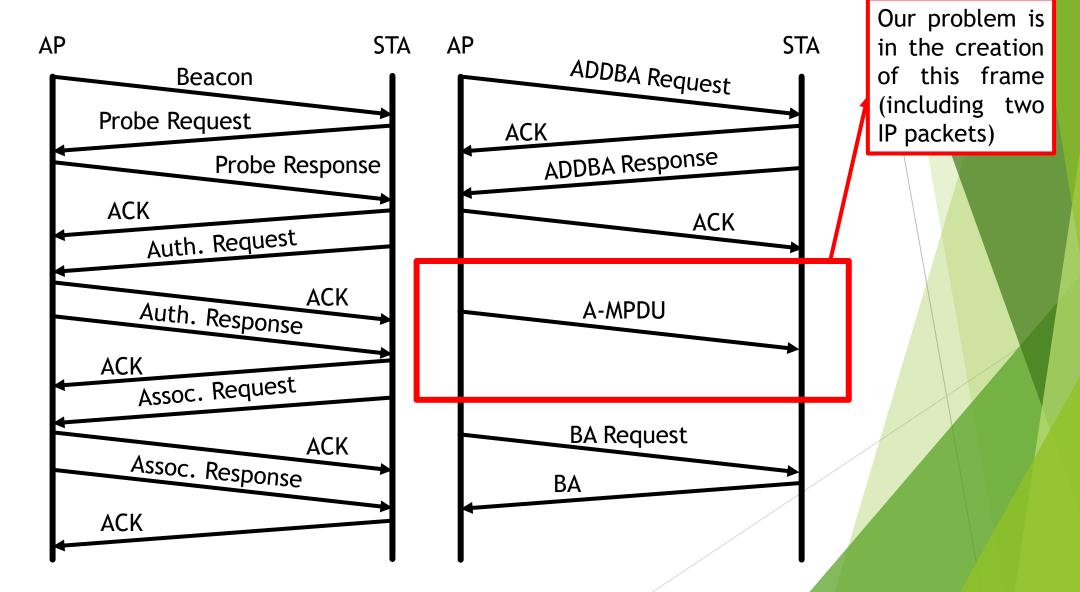
- make it able to manage the association with a normal 802.11n device
- Send a number of A-MPDUs to it, and study the savings in terms of air time and efficiency
- The source code is being shared here
  - https://github.com/Wi5/wi5-aggregation
- ► This is part of a Final Degree Project in Unizar.es. It is also a collaboration with Wi-5 project.

#### Hardware

- As an AP:
  - ► TP-Link TL-WN722N with the driver ath9k\_htc working in monitor mode

- As the STA:
  - ► AR9287 Wireless Network Adapter (PCI-Express) rev 01 with the driver ath9k working in managed mode

Traffic Scheme (Connection & Data)



#### Beacon Packet (Fields and values we fill)

Radiotap Header IEEE 802.11 Management - Beacon

-Version: 0

-Length: 9

(Radiotap Header

- + Data Rate)
- -Present flags:

Only a 1 in Rate

-Data Rate = 108

in decimal

-Version: 0

-Type: Management

-Subtype: Beacon

-Duration: 0x0000

-Destination Address: ff:ff:ff:ff:ff

-Transmitter Address: Our MAC

-Source Address: Our MAC

-Sequence number

-Timestamp: Current Timestamp

-Interval: 102.4 ms.

-Capabilities: ESS, Immediate

Block ACK & QoS

-SSID: ap0

-DataRates

-Channel: 5

### Probe Response (Fields and values we fill)

Radiotap IEEE 802.11 IEEE 802.11 Header MAC Header Probe Response

-Version: 0

-Length: 9

(Radiotap Header

- + Data Rate)
- -Present flags:

Only a 1 in Rate

-Data Rate = 108

in decimal

-Version: 0

-Type: Management

-Subtype: Probe Response

-Duration: 0x0000

-Destination Address: ff:ff:ff:ff:ff

-Transmitter Address: Our MAC

-Source Address: Our MAC

-Sequence number

-Timestamp: Current Timestamp

-Interval: 102.4 ms.

-Capabilities: ESS, Immediate

Block ACK & QoS

-SSID: ap0

-DataRates

-Channel: 5

### Auth. Response (Fields and values we fill)

Radiotap IEEE 802.11 IEEE 802.11 Header MAC Header Management

-Version: 0

-Length: 9

(Radiotap Header

- + Data Rate)
- -Present flags:

Only a 1 in Rate

-Data Rate = 108

in decimal

-Version: 0

-Type: Management

-Subtype: Authentication

-Duration: 0x0000

-Destination Address: STA MAC

-Transmitter Address: Our MAC

-Source Address: Our MAC

-Sequence number

-Auth. Algorithm: Open System

-Auth. SEQ: 0x0002

-Status Code: Successful

### Assoc. Response (Field and values we fill)

Radiotap IEEE 802.11 IEEE 802.11 Header MAC Header Management

-Version: 0

-Length: 9

(Radiotap Header

+ Data Rate)

-Present flags:

Only a 1 in Rate

-Data Rate = 108

in decimal

-Version: 0

-Type: Management

-Subtype: Association Response

-Duration: 0x0000

-Destination Address: STA MAC

-Transmitter Address: Our MAC

-Source Address: Our MAC

-Sequence number

-Timestamp: Current Timestamp

-Interval: 102.4 ms.

-Capabilities: ESS, Immediate

Block ACK & QoS

-Status code: Successful

-SSID: ap0

-DataRates

-Channel: 5

#### AddBA Request (Fields and values we fill)

Radiotap IEEE 802.11 IEEE 802.11 Header MAC Header Management

-Version: 0

-Length: 9

(Radiotap Header

+ Data Rate)

-Present flags:

Only a 1 in Rate

-Data Rate = 108

in decimal

-Version: 0

-Type: Management

-Subtype: Action

-Duration: 0x0000

-Destination Address: STA MAC

-Transmitter Address: Our MAC

-Source Address: Our MAC

-Sequence number

-Category Code: Block ACK

-Action Code: Add Block ACK

Request

-Dialog Token: 0x01 and increases

in AddBA Request package sent

-Block ACK Parameters: 0x1002

(means Inmediate Block ACK and

the order of the buffer)

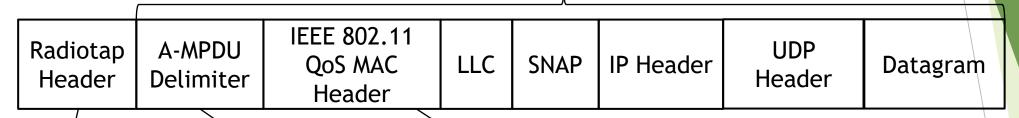
-Block ACK Timeout: 0x0000

-Block ACK SSC: The following

number of sequence of this frame

#### A-MPDU (I): structure with two IP packets

We repeat this twice in the same frame



- -Version: 0
- -Length: 9

(Radiotap Header

- + Data Rate)
- -Present flags:

Only a 1 in Rate

-Data Rate = 108 in decimal

- -Reserved&Lengh: 0x4004
- -CRC: 0x47
- -Delimiter Signature: 0x4E

('N')

All these values were taken from an example

-Version: 0

-Type: Data

-Subtype: QoS

-Duration: 0x0000

-Destination Address: STA MAC

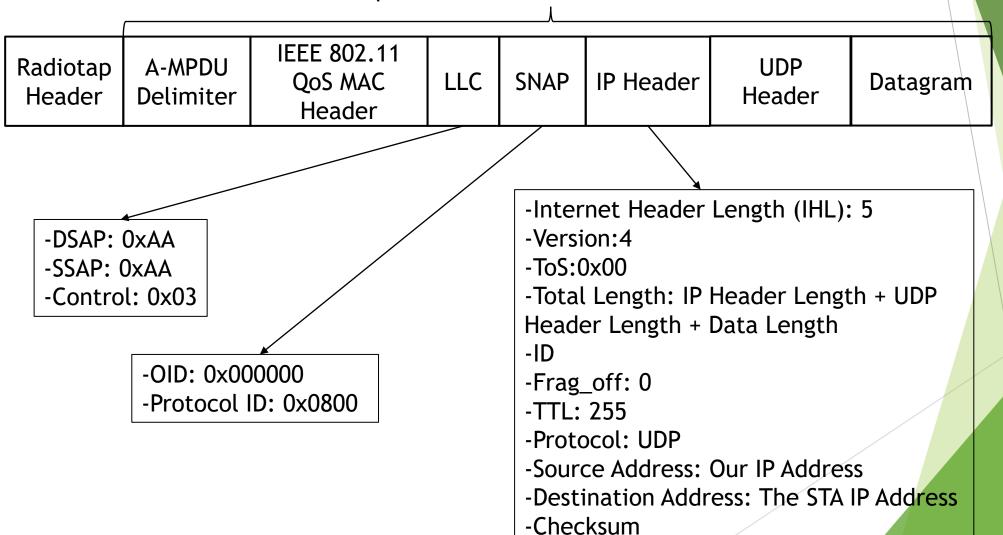
-Transmitter Address: Our

MAC

- -Source Address: Our MAC
- -Sequence number
- -QoS: 0x0000

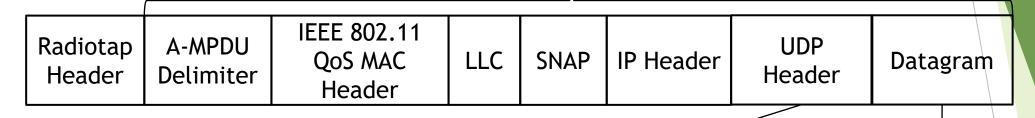
#### A-MPDU (II): structure with two IP packets

We repeat this twice in the same frame



#### A-MPDU (III): structure with two IP packets

We repeat this twice in the same frame



-Source Port: 6666

-Destination Port: 8622

-Length: UDP Header Length +

Data Length

-Checksum

In this example 8 zeros due to the fact of using the example A-MPDU delimiter values

#### A-MPDU (IV): Questions

- ► This frame is giving us the problems, it could be happening because we didn't notice that some flag of this frame or the previous ones must be enabled, we tried to enable in this frame the "A-MPDU status" that belongs to the radiotap header but we couldn't.
- ▶ Other thing that could be wrong are the values of the A-MPDU delimiter but we try to use the values used in the following program as we said previously:
  - https://github.com/rpp0/aggr-inject
- ▶ But even using those values, Wireshark thinks that the A-MPDU delimiter is the Frame Control Field, as you can **see in the attached capture**. Wireshark thinks there were Fragmented IEEE802.11 frames

#### **BA Request**

-Version: 0

-Length: 9

(Radiotap Header

+ Data Rate)

-Present flags:

Only a 1 in Rate

-Data Rate = 108

in decimal

Radiotap | IEEE 802.11 Header | MAC Header IEEE 802.11 BAR Frame

-Version: 0

-Type: CTL

-Subtype: BAR

-Duration: 0x0000

-Destination Address: STA MAC

-Transmitter Address: Our MAC

-Source Address: Our MAC

-Sequence number

-BAR Control: 0x0001 (means

Immediate Block ACK)

-Block ACK SSC: First MPDU

sequence number

# THANKS FOR YOUR HELP