URSI GASS 2020
Rome, Italy
29 August - 5 September 2020

Characterizing Wi-Fi Man-in-the-Middle attacks

Andy Amoordon, Christophe Gransart and Virginie Deniau

Université Gustave Eiffel

SUMMARY

The need for intermediaries

Man-in-the-Middle attacks

Fake Wi-Fi access point attacks

Levels of difficulty for the attacker

Consequences of the attack

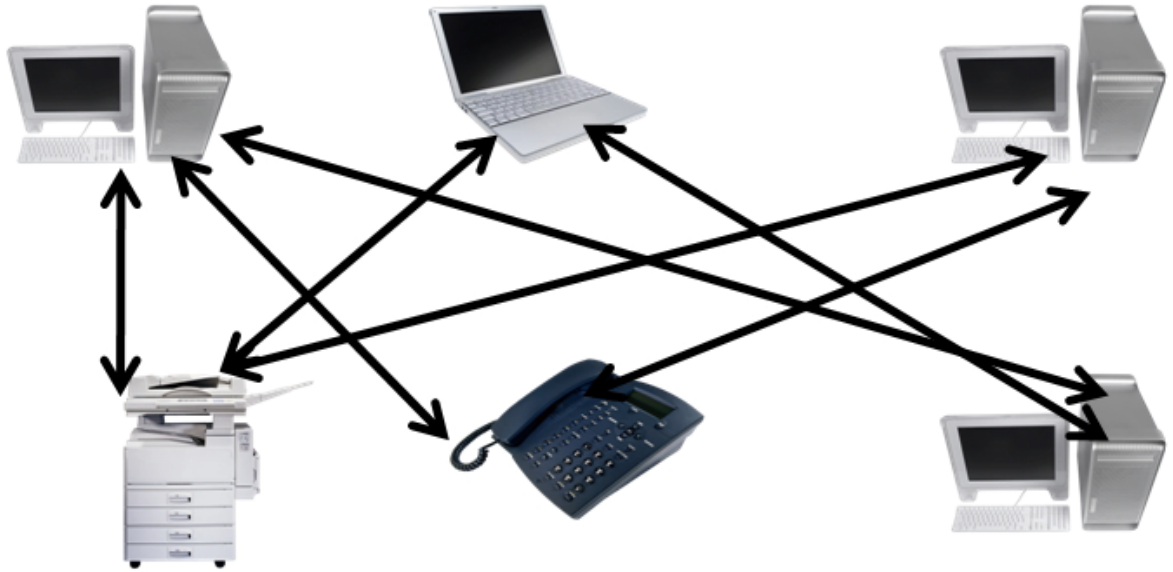
Existing countermesurement and their limitations

Our proposition

Our methodology

Preliminary results

The need for intermediairies

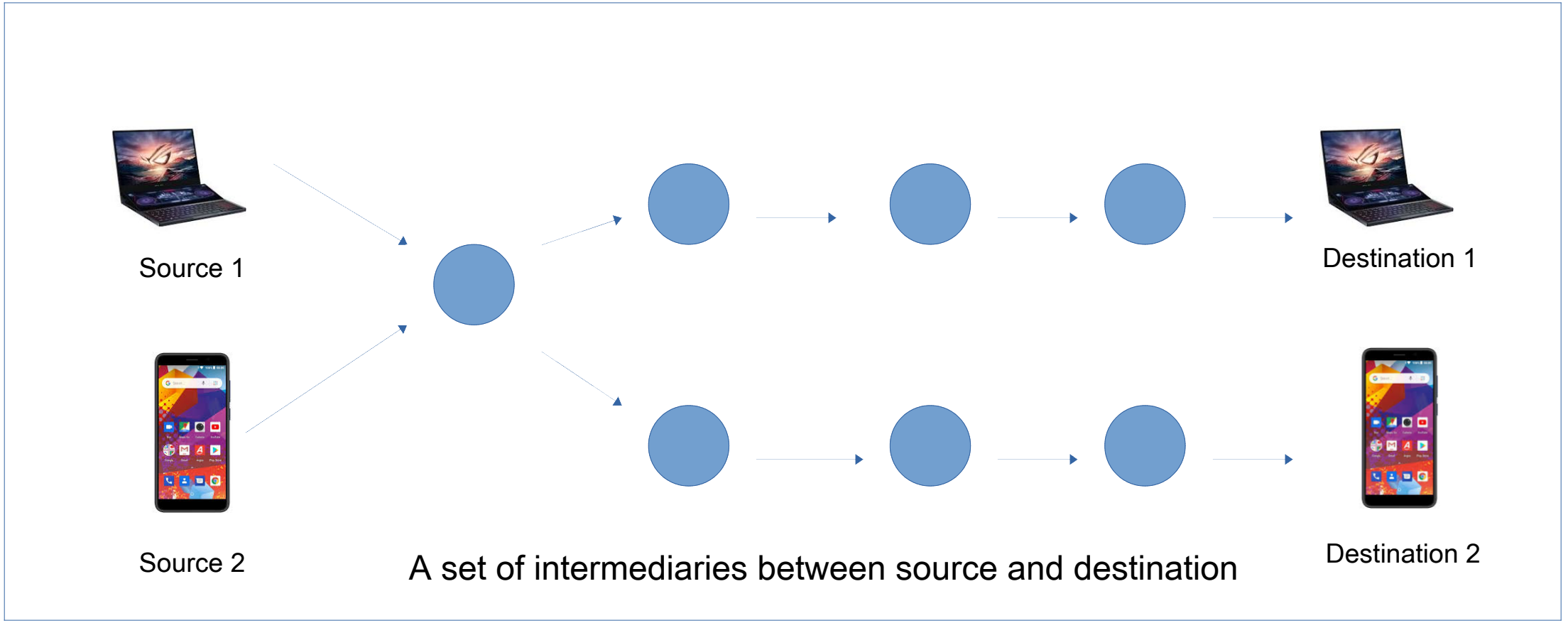


In an ideal world, all devices would be connected to each other and would communicate directly.

Under this configuration, devices do not need intermediaries and do not need to trust them to correctly forward their data .

However, this network configuration is crumbersome to make, expensive to maintain and not flexible.

The need for intermediairies (1)



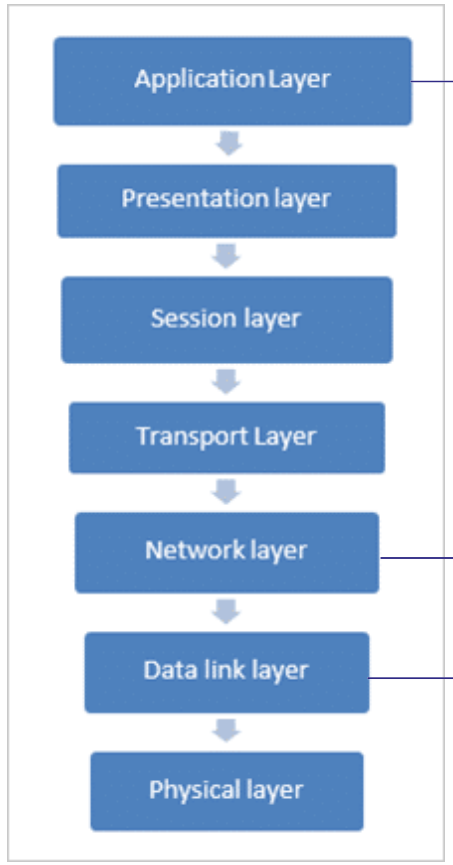
Intermediairies are therefore needed to connect devices between each other. Intermediairies allows for sharing of ressources, network expansion but devices have to trust them in doing that they advertise. Intermediairies can be also be hijacked or usurpated by attackers. These unsought situations are commonly referred to as Man-in-The-Middle (MiTM) attacks.

Man-In-The-Middle attacks

There is a Man-in-The-Middle attack when:

1) An intermediary knowingly read or modify packets it is relaying (e.g. Internet service provider doing packet sniffing or deep packet inspection)
2) An intermediary is hijacked by an attacker to read or modify relaying packet (e.g. email account of the human ressource manager is hijacked to access and to send important mail)
3) An intermediary is ursurped by an attacker to read or modify packets (e.g. fake Wi-Fi access point or fake cell tower deployed by an attacker to read or modify data packets)

Man-In-The-Middle attacks (1)



OSI Model

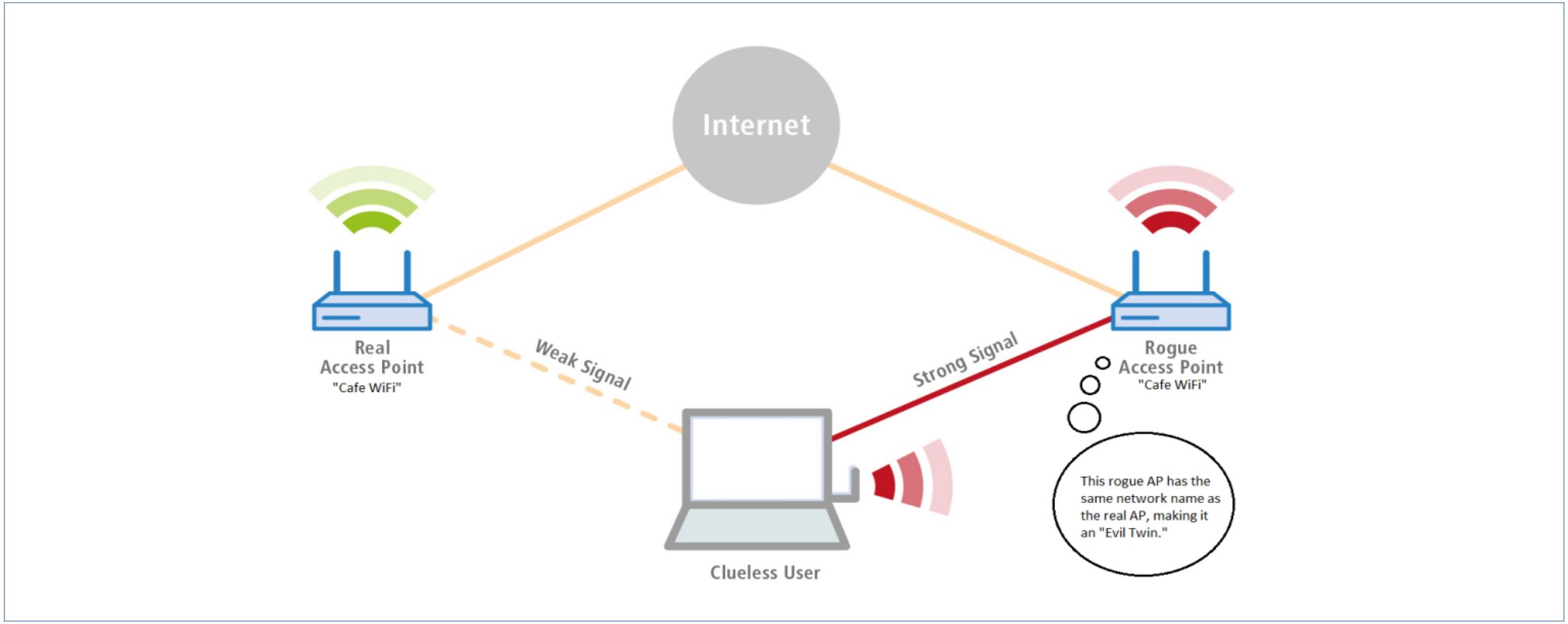
Email hijacking...

Internet service provider doing packet sniffing or deep packet inspection…

Fake Wi-Fi access points, fake base stations...

Man-in-The-Middle attacks can be perpetuated on different layers of the OSI model. Our research focuses on MiTM attacks on the two lower layers of the OSI model (Data link and physical layers)

Fake Wi-Fi access point attacks



1. The attacker creates a fake Access Point (AP) with same network name but often with a stronger signal

2. The user's device either automatically connects to the fake AP or is forced to connect to the fake AP by a deauthenfication attack.

Fake Wi-Fi access point attacks (1)

There are three types of Wi-Fi networks:

1. Free Wi-Fi / unprotected Wi-Fi

2. Wi-Fi Encrypted networks (shared key encryption: WPA1, WPA2...)

3. Reinforced Wi-Fi encrypted networks (unique key encryption and double authentication: WPA-E...)

Consequently, the attacker has different levels of difficulty depending on the type of the Wi-Fi network

Levels of difficulty for the attacker

Type: Free Wi-Fi or unprotected Wi-Fi

Level : Easy

The attacker simply has to deploy an AP emitting similar beacon frames as the licit AP and either wait for the client to seemlessly connect to his AP or force them to connect by sending deauthentication packets

Type: Encrypted Wi-Fi (shared password)

Level : Medium to hard

Additional step: The attacker needs to have the password else devices would not seemlessly connect to his AP. Password can be either easily or cumbersome to obtained

Type: Encrypted Wi-Fi (Unique key and double authentication)

Level : Hard

Additional step: The attacker needs to have the login and password of all targeted users and need to install certificate on targeted users' devices before deploying his fake access point

Consequences of the attack

- When a user connects to the wrong Wi-Fi access point, it performs layer 2 encryption with the attacker. The attacker can then read or modify his frames. There is the loss of anonymity and privacy.

- Moreover, Wi-Fi Man-in-The-Middle attacks are often a first step for more virulent attacks which leads to identity usurpation of the users and the theft of personal data and sensitive data (e.g. medical data, social media accounts, credit card information...)

Exisiting countermesurements

- Use upper layer encryption such as VPN and the HTTPS protocol so that an attacker cannot read or modify data of higher OSI layers
- Use authentication credentials such as tokens and double authentication to authenticate client and Wi-FI points
- Use Intrusion detection to detect the presence of MiTM attacks

Limitations of exisiting countermesurements

- VPNs do not encrypt all layers and simply move the trust to another intermediary. VPNs can therefore be attacked, hijacked and ursurpated.
- Double authentication and tokens are not implemented on all Wi-Fi protocols. Double authentication can be counteracted by certificate injection into users' devices.
- Existing Intrusion detection systems for Man-in-the-Middle attacks are designed for wired networks and not for wireless network such as Wi-Fi networks. Moreover, they are designed for upper layers which means that signals need to be demodulated and decoded up to layer 3 of the OSI model.

Our Solution : an IDS based on lower OSI layers

- An IDS on lower layers is faster as it does not have to decode the signal up to the network layer. It will therefore use less resources and will be quicker to detect the attacks. Moreover, an IDS on lower OSI layer can sweep a whole bandwidth and detect attacks on a wider spectrum

- Higher layer IDSs normally have to limit the detection on a specific Wi-Fi channel

- For these reasons, we aim at building an IDS based on the two lower layers of the OSI model

Our Methodology

1. Model and build SDR chains for normal, jamming attack and fake access point attack situations using GNU radio [1]

2. Test the chains by simulation under a similar setup on GNU radio

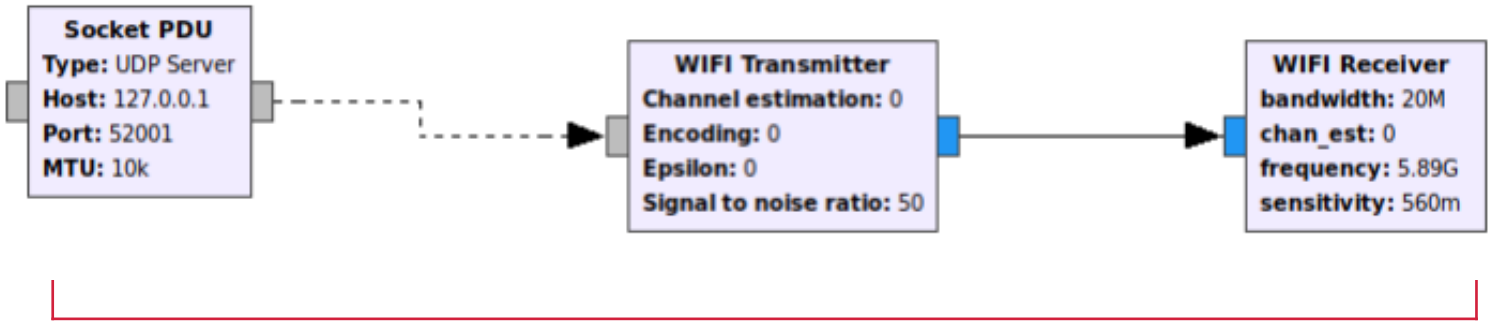
3. Record, for the three situations, fluctuations in parameters on physical and data link layers of the OSI model (number of lost packets, signal strength, transmission time, throughput...)

4. Compare normal situation values with attack situation values to find indicators that allows us to detect the presence of jamming and fake access point attacks

5. Reproduce the simulations with hardware equiment (Hackrf, USRP and Wi-Fi devices) to refine the results

[1] GNU Radio is a free & open-source software development toolkit that provides signal processing blocks to implement software radios

GNU Radio Chain - Normal situation



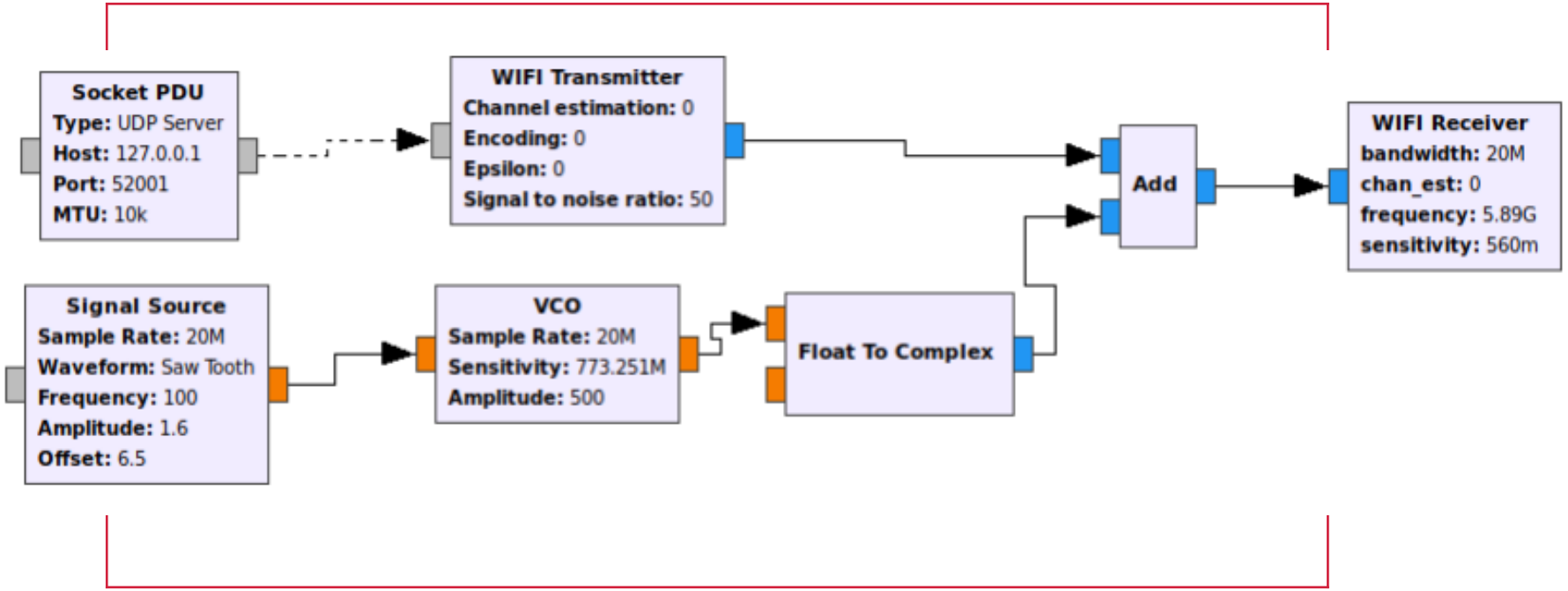
Wi-Fi beacons are transmitted periodically, as signals, by a Wi-Fi transmitter. They are received and decoded back into frames by a Wi-Fi receiver. The Wi-Fi receiver demodulates the signal and sent the frames to Wireshark [1] for frame analysis and calculations.

A strong Signal to noise ratio is used to cater for jamming and fake access point situations. Beacons are created using a python script and sent by the UDP server

[1] Wireshark is a free and open-source protocol analyzer.

GNU Radio Chain - Jamming attack situation

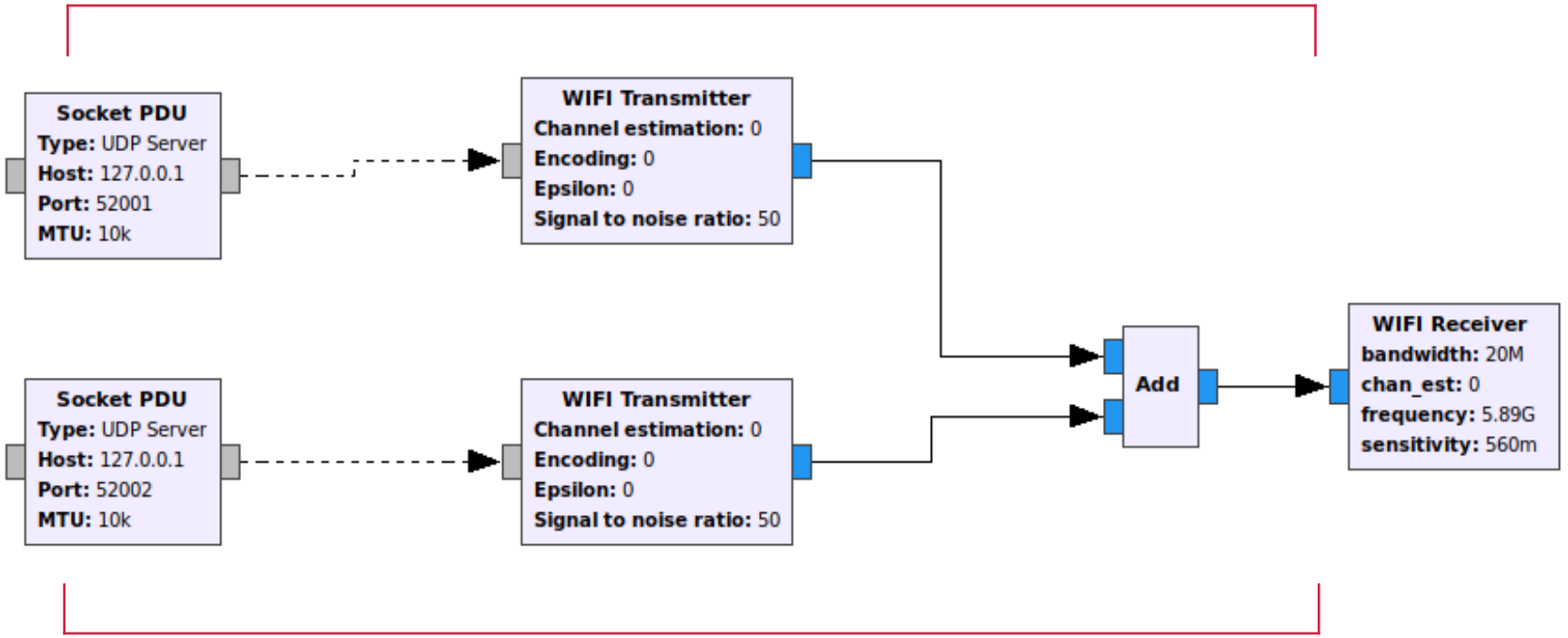
Licit access point



Powerful jamming is added to the licit transmission

GNU Radio Chain - Fake access point attack situation

Licit access point



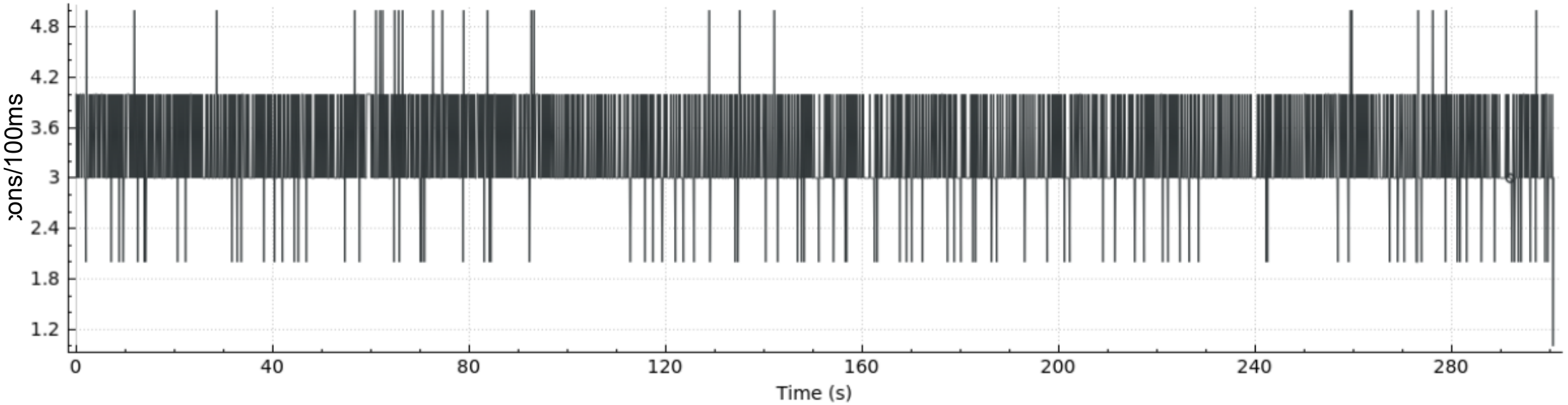
Fake access point transmitting same beacon on the same channel, but at an interval of 250ms

Preliminary results - Setup

- For the three situations, 10,000 beacons are sent at an interval of 200 ms by the licit access point
- A jammer with a high amplitude is added to the licit access point beacon transmission signal in the second scenario
- A fake access point emitting similar beacons, but at an interval of 250 ms is added to the licit access point beacon transmission signal in the third scenario
- Parameters are measured at the receiver's end

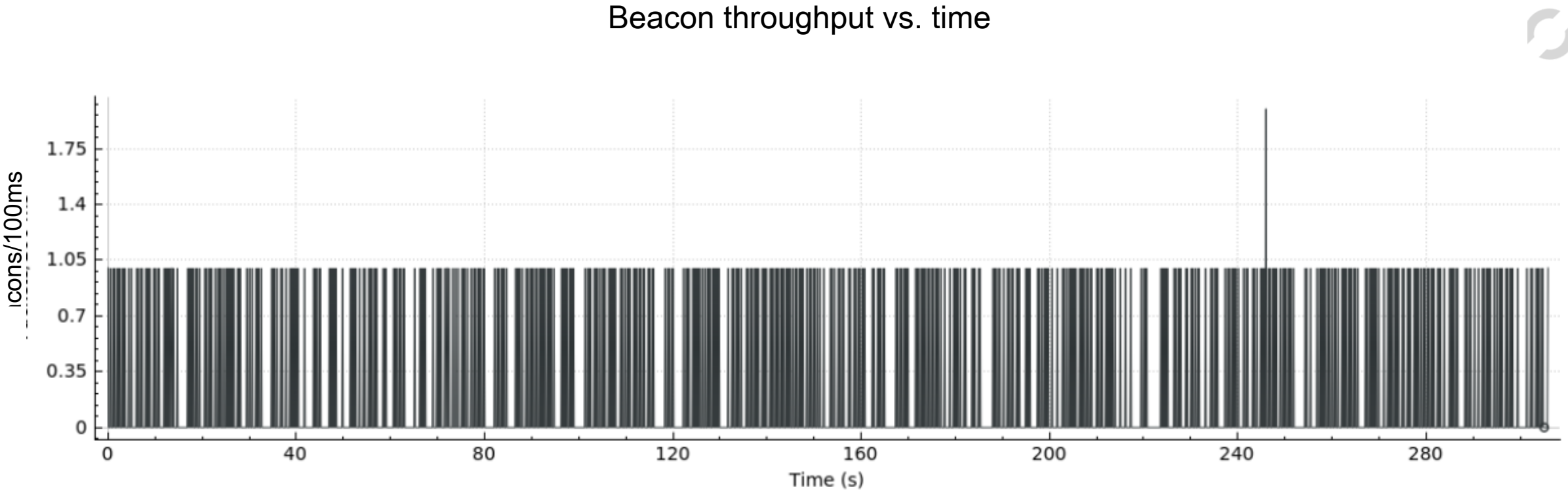
Preliminary results - Normal Situation

Beacon throughput vs. time



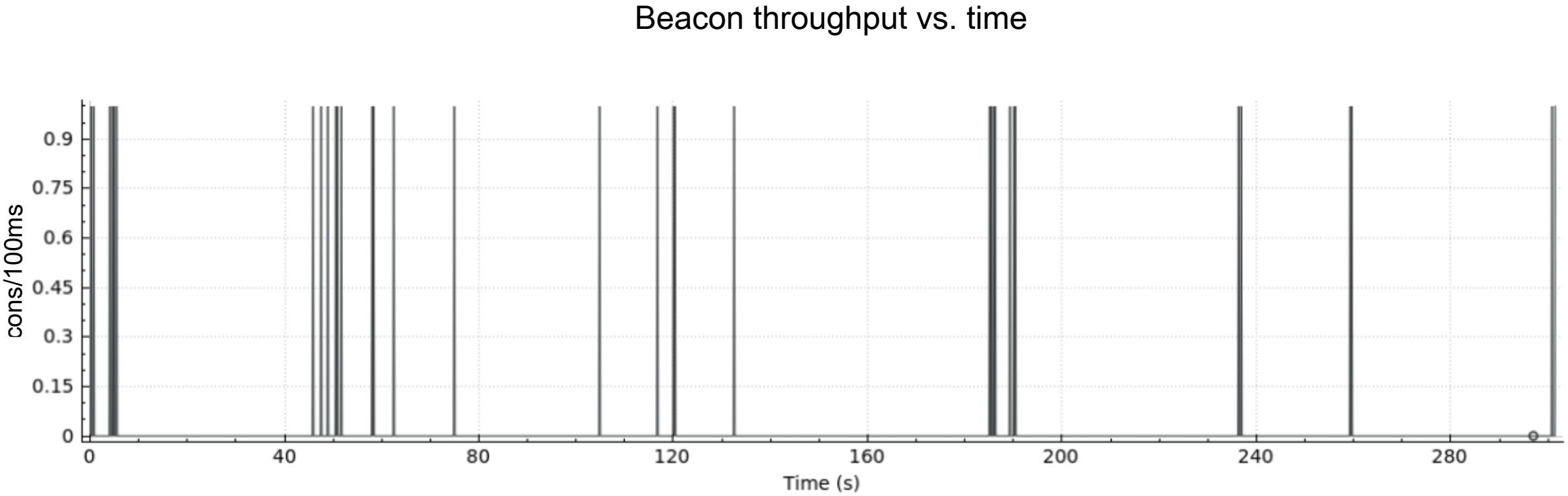
We observe that under the normal situation, the number of received beacons/100 ms is between 0.6 to 5 beacons/100 ms There are some fluctuations in the number of beacons/100 ms but the receiver received at least 0.6 beacon/100ms throughout the reception time. Provided that beacons are sent each 200 ms (0.5 beacon/100ms) and that packets cannot be lost under simulation, the amount of beacon loss for normal situation should be null.

Preliminary results - Jamming attack Situation



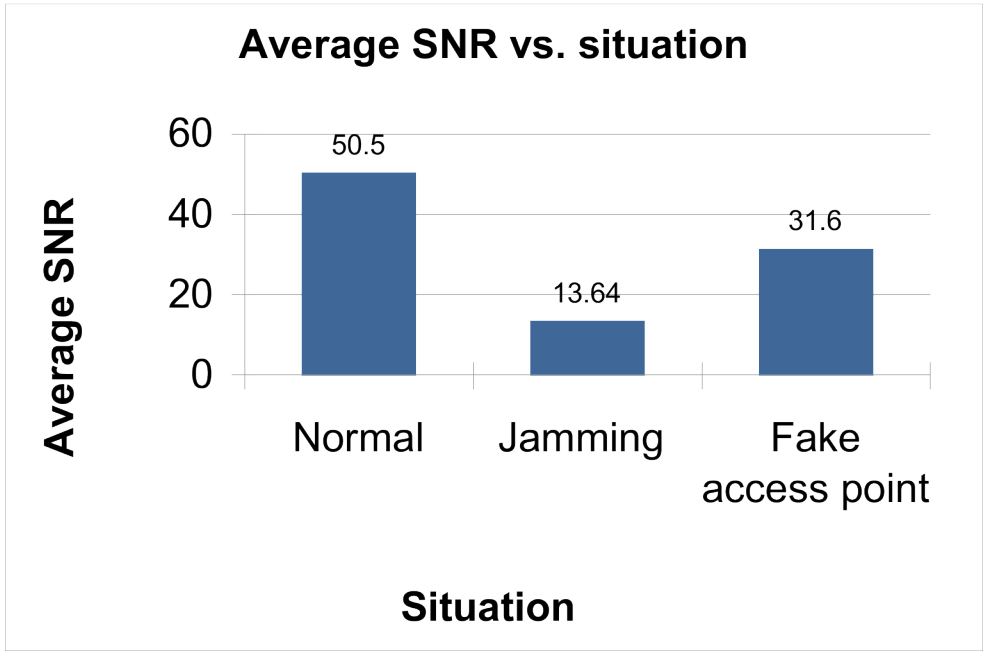
In contrary to the normal situation, we observe here that the number of beacons/100ms is within a smaller range and that there are several instants where the number of beacons/100ms is equal to zero. This indicates that the jamming had an effect on the reception and has led to the loss of beacons and more generally performance degradation.

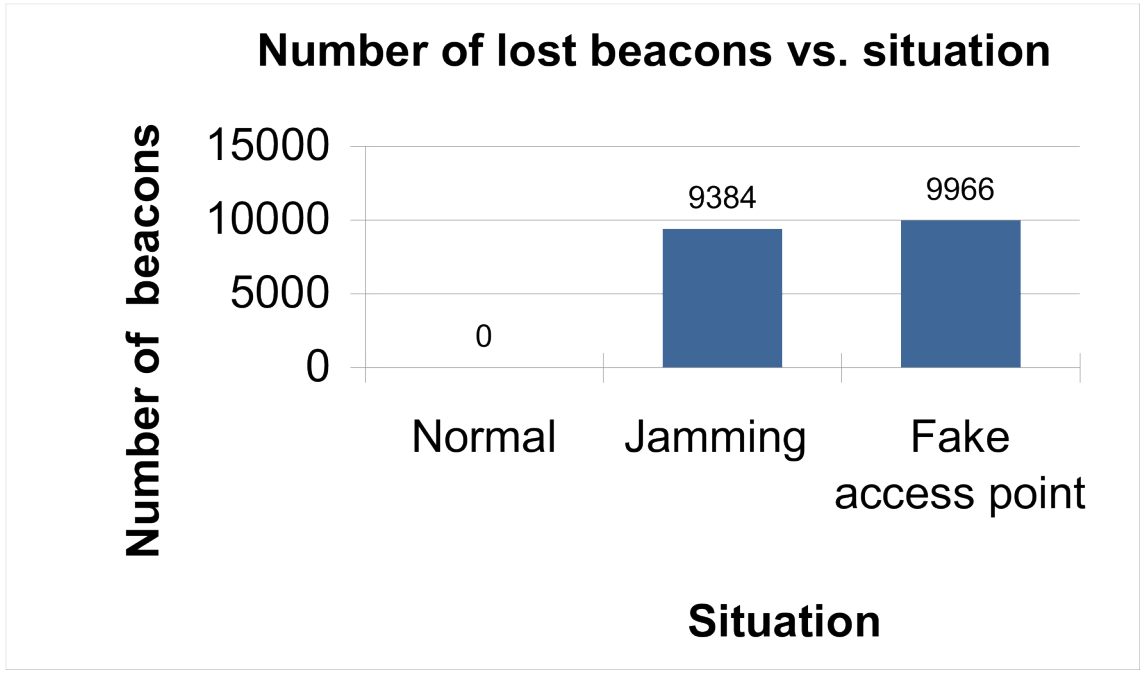
Preliminary results - Fake AP attack Situation



Here, we can clearly observe that at several instants, the beacons/100ms is zero. The reason can be due to the fact that the access points are emitting beacons on the same channel at a close interval of time (200 ms for the licit access point and 250ms for the fake access point), there is overlapping (interference) and the receiver can only decode beacons at instants where the interference is not so high.

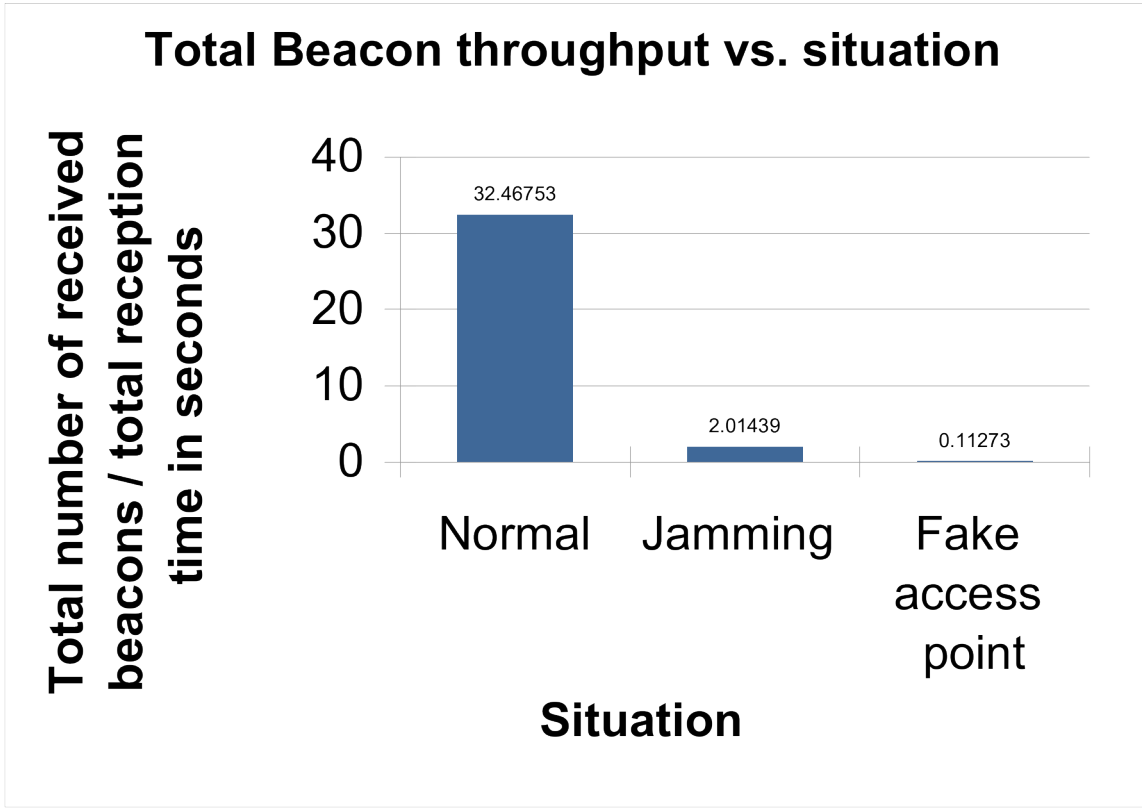
Preliminary results - Side-by-side Comparision





The average SNR vs situation and the number of lost beacons vs situation graphs validate our previous assertions that jamming and fake AP deteriorates performance. We observe here that the SNR is the lowest for the jamming situation. Note that the SNR for the fake access point situation might seems high,but the average is calculated on the received packets so when interference (overlapping) between two access points is the lowest while for in the jamming situation, the jamming signal is constant. This alsoo indicate that the SNR value from layer 2 at reception might not be a proper indicator of the fake access point attack.

Preliminary results - Side-by-side Comparision



The number of lost beacons and the total beacon throughput confirms that in this experiment, the fake access point attack situation is the worst in beacon reception. A way to detect the fake access point on layer 2 of the OSI model would be to check if the beacon interval is the same at transmission and reception. In this case, the licit AP is emitting 1 beacon each 200ms (5 beacons per second) but the receiver has only received 0.11273 beacon in one second.

Andy AMOORDON

andy.amoordon@univ-eiffel.fr

Université Gustave Eiffel