# Python Simulator Extension

Rodrigo Joni Sestari, Mat. number 179020
rodrigo.sestari@unitn.it

***Abstract*—This document describes the modifications made to the implementation of Trivial Carrier Sensing and analyses the results.**

## I. Introduction

- general objective this document explain the extending of simulator framework using the concepts of DES.
- problem: extend the protocol including a simple carrier sensing function that prevents transmission if there are packets being transmitted within the communication range of the node. The packet is transmitted when the channel becomes free.
- strategy: create a new state on protocol that represents the trivial Carrier Sensing.

## II. Base information

The protocol is composed by 4 states; *IDLE, TX, RX, and PROC* where:

- IDLE: the node *Thinks* the channel is free .
- TX: the node is currently transmitting a packet.
- RX: the node is currently receiving one or more packets.
- PROC: the node is performing a little processing after a TX or an RX.

The network used to evaluate the extension simulator is composed by 10 nodes, in the Figure 1 show how the network is distributed.

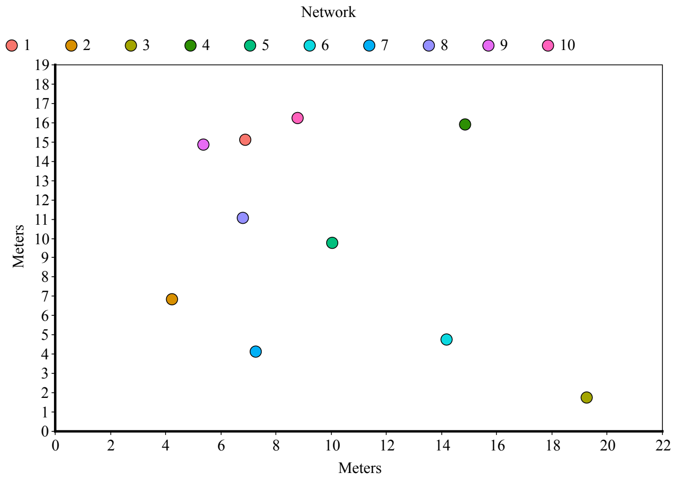

Figure 1. Network

## III. Implementation

The implantation consists in create a new state on protocol to simulate the Carrier Sense, This new state is called *CS*, the main idea is that at moment that arrives some packet to transmit, the protocol verify if there any signal in the air, if positive, then the protocol state becomes *CS*, once in the state of *CS*, the protocol change the state when there is not signal the in air. Is possible see a sample of state machine of protocol in the Figure 2, the red color represent the extension protocol released by this project.

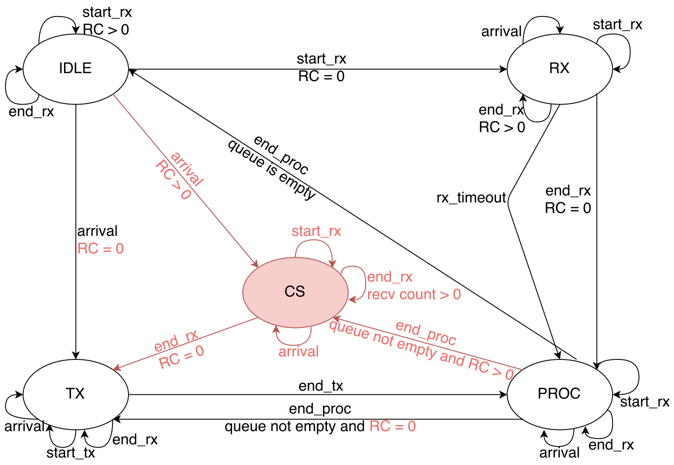

Figure 2. State Machine protocol

To extent the protocol was necessary modificate three events, bellow will be describe the single modification for each events. The first event that will be described is *end-Proc*, visible on Figure 3, this state is used by a node to know when processing is over, the modification was realised at moment that the queue is not empty, is controlled if there is any signal in the air, in positive case, then the state is changed to *CS*, in negative case the protocol put the packet on the queue.

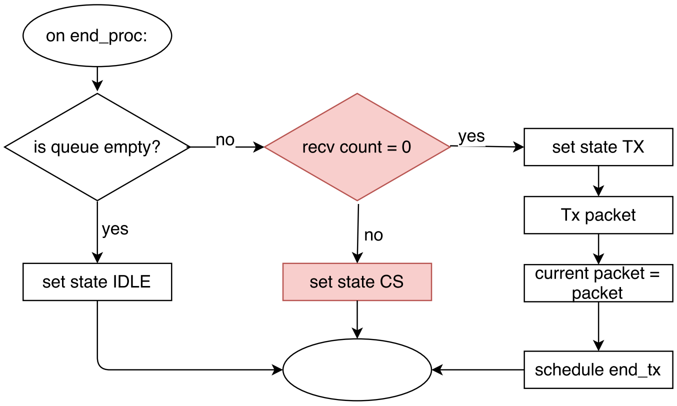

Figure 3. end-Proc event

The next event is *end-Rx*, visible in the Figure 4, this event notifies the node the end of an incoming packet. The modification created in the extension protocol is concentrated on *CS* state when there is no signal in the air, in this case, the state is changed to *TX* state, it happens because if you are in *CS* state, means that you want transmit a packet but there is some signal in the air, it characterise the Carrier Sensing.

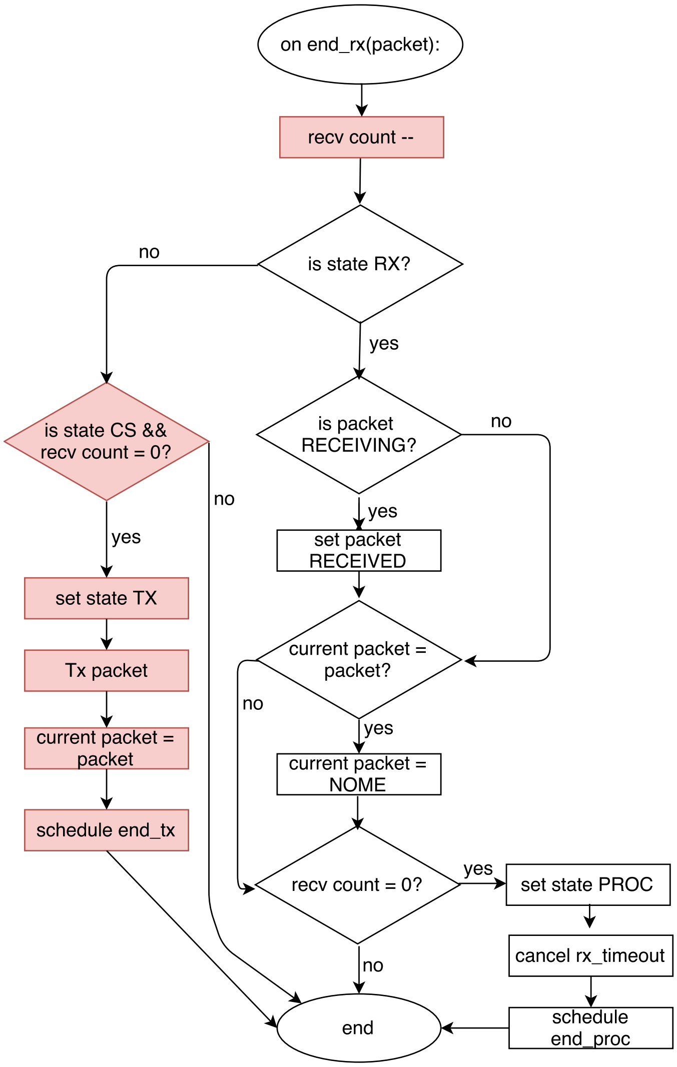

Figure 4. end-RX event

The last event changed is *on-Arrival*, visible on Figure 5, this event happens when a new packet is generated by a node. The modification in this event is concentrated on *IDDLE* state, if a new packet is generated and there is not any signal in the air, then the state is changed to *TX*, but if there is some signal in the air, the packet just arrived is put in the queue and then the state is changed to *CS*. Once time on *CS*, the state is changed only there is no signal on air.

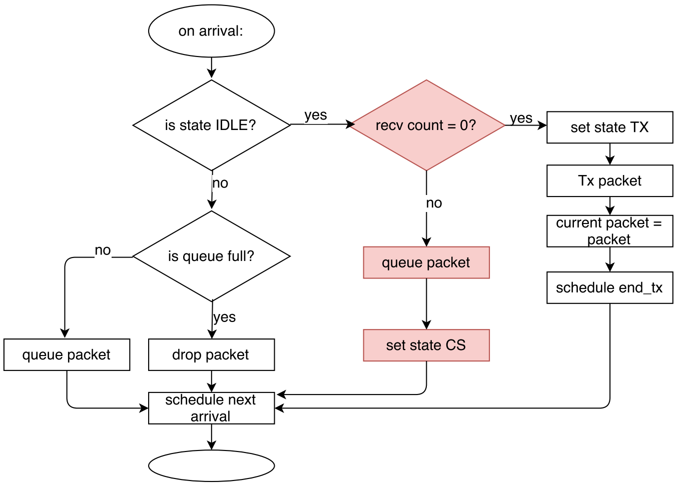

Figure 5. on-Arrival event

## IV. Evaluation

The evaluation is divided in two parts, the first one represents the results of Simulator Extension, the second one represents the comparison between the base and extension Simulator. The evaluation is made by throughput and packet collision rate at receiver and packet drop rate at sender.

### A. Simulator Extension

The graph on Figure 6 represents the throughput at receiver by the total offered load. Is interesting highlight the result at receiver 3 and 8. Is possible see on Figure 1 their network position. The receiver node 3 is clearly the more remote node in the network, instead the node 8 is the node with more neighbours near. Is possible in the Figure 6 that when the throughput decrease for the nodes 1-2,4-10 *more quickly for the node 8*, the throughput increases for the node 3. Been the node 3 more distance of others nodes, the rate of packet collision and drop is less than of theirs, it allows the throughput to grow in when others nodes decrease.

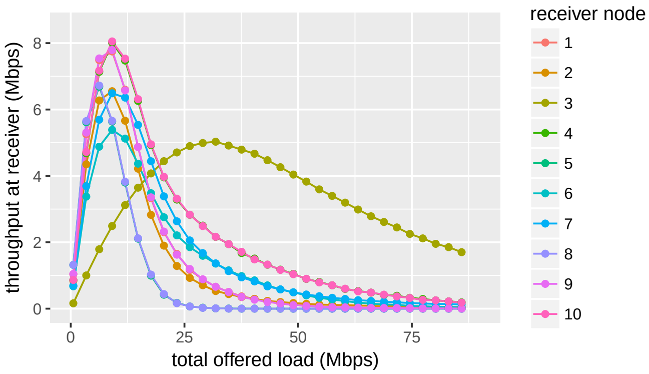

Figure 6. Throughput Simulator Extension

The graph on Figure 7 represents the packet collision rate by the total offered load. Another time focus is on receiver 3 and 8. The receiver 3 has clearly the rate lowest, instead the node 8 has the highest. The node range in the network is 10 meters, the node 3 is seen only by the node 6. its allows the node 3 has a less packet collision rate.

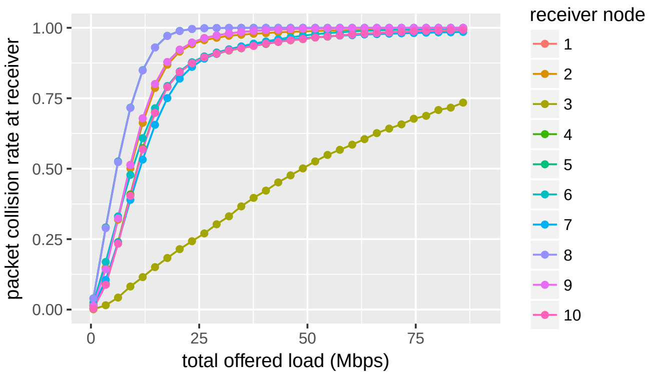

Figure 7. packet collision Simulator Extension

The graph on Figure 8 represents the packet dropped rate by the total offered load. Here another the the nodes 3 and 8 are distinct. The packet drop rate reflects the time the protocol is in the *CS* state.

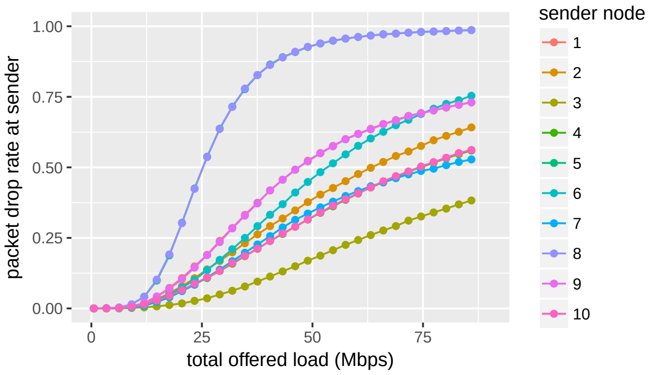

Figure 8. packet dropped Simulator Extension

### *B. Comparison between base and extension Simulator*

The Figure 9 represents the throughput at receiver, is possible see that the extension version has a slight more peak than the base version. The reason is that there are already nodes in the *CS* state, allowing the other nodes to transmit without collision, the throughput decrease after the highest peak, probably the reason is that all nodes are in the state *CS*, after which the throughput returns to increase in relation to the base version, probably because the nodes try to retransmission.

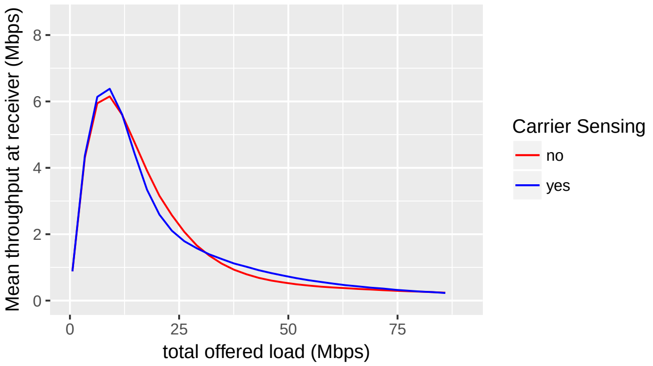

Figure 9. Throughput

The Figure 10 represents the packet collision at receiver. The extension version has a mean slightly lower than the base version. The reason is that when the protocol is in the state *CS*, the node stop transmitting, decreasing the collision.

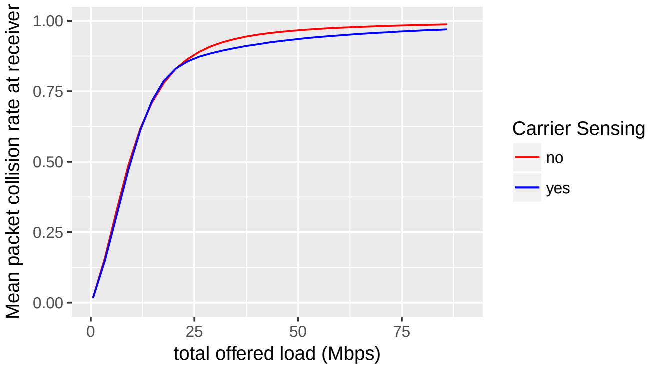

Figure 10. packet collision

The Figure 10 represents the packet dropped at sender, is possible to see that the mean of extension version is greater than the base version, it happens because when the protocol is *CS* state, when new packets arrives they are dropped.

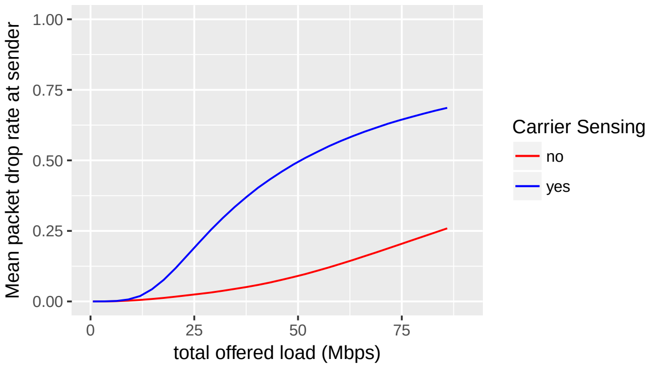

Figure 11. packet dropped

## V. Conclusion

The extension of the protocol with trivial Carrier Sensing allowed to decrease the collision rate, in addition to increasing the throughput, the trivial Carrier Sensing implemented in this project does not contain *slotted aloha* methodology, being one of the reasons that caused the average packet dropped to be high in respect to the base version.