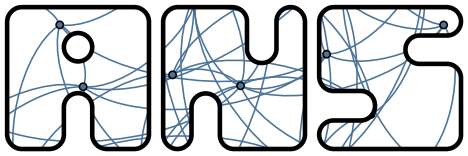

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Assignment 2

Renato Lo Cigno

Simulation & Performance Evaluation 2016-17

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Start from the simple simulator framework that is available through classroom
- Select one of the proposed extensions, design it (flow chart, modifications, . . .) implement it, run proper simulations and write a report presenting and commenting the simulation results
- Delivered the code with the report, properly commented, with all the scripts needed to compile and run it, including the scripts to analyze and plot results. The report should be concise and clear, no more than 4 pages in the given format
- Not all modifications have the same complexity . . . and they don't have the same grade

Trivial Carrier Sensing (Max: 24/30)

- Extend the protocol to include 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
- Compare results with the original simulator

Simple Carrier Sensing (Max: 27/30)

- As the **Trivial Carrier Sensing**, but a p-persistent version

- A station before transmitting must sense the channel
  - If the channel is free: transmits
  - If it is occupied, it re-schedules the transmission with persistency p.

  p defines if the transmission is scheduled immediately (when the channel becomes free) or re-scheduled after random time

- Performance evaluation must include a sensitive analysis on p

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Extend the **Trivial Carrier Sensing** protocol with a collision avoidance procedure
- Slotted contention window: Slot duration T_s; contention window size W_c.
- On busy channel wait until it becomes free, then extract an integer random variable B_k uniformly distributed in $[0, W_c - 1]$ and count-down B_k slots
- When $B_k = 0$ transmit
- If the channel becomes occupied while counting re-schedule the transmission when the channel becomes free again extracting a new value for B_k
- After a transmission the station must behave as if the channel were occupied
- Compare the results for different values of W_c

Realistic Propagation (Max. 24/30)

- Can be added to any prev. modification and makes +3 on the grade
- Change the packet reception model from a disk model to a probabilistic model
- Probability of reception:

$$\Pr(\text{correct reception}|d) = \begin{cases} 1 - \frac{d}{\text{RX}_{\text{range}}} & \text{if } d < \text{RX}_{\text{range}} \\ 0 & \text{otherwise.} \end{cases}$$

  This models the reception probability when no collision occur.
- The probability of reception for colliding packets is zero

Unicast Traffic

- Implement destinations and acknowledgment of packets
- Stations must be identifiable and packets are sent from one specific station to another one
- Received packets are ACKed
- Packets that are not ACKed must be re-sent after an exponential random time

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Two timing parameters must be defined: SIFS and DIFS (we take WiFi terminology here)
- SIFS (Short Inter-Frame Space) is 10 μs and separates a frame (packet) from its ACK.
- DIFS (Distributed Inter-Frame Space) is 50 μs and separates two different frames, or better, the end of a transmission (reception) from the attempt to send a new frame
- Carrier sensing must be implemented with $T_s = \mathrm{DIFS}$ to protect ACKs from collisions
- Collided packets are re-scheduled after a random time

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Implement the full (well a bit simplified!!) MAC of WiFi as follows

- Start from the **Unicast Traffic** extension
- Change the channel access mechanism like in the **Carrier Sensing with Collision Avoidance**
- The slot duration is $T_s = 20\,\mu$s
- The channel must be sensed free for an entire DIFS before starting to contend for the channel
- When a packet is lost (the ACK is not received), a node should double its contention window up to a maximum value W_{max}

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The "game" is repeated until the packet finally gets through or a maximum number of attempts N_a is reached
- When successfully sending a packet (or when giving up after N_a attempts) the contention window is reset to W_{min}
- In basic, standard WiFi we have $W_{\text{min}} = 32$, $W_{\text{max}} = 1024$, $N_a = 7$
- When the channel turns busy while performing the countdown for channel access, the value of the countdown should be "frozen", and the node should continue with the countdown when the channel turns free again, without extracting a new value for B_k