Monday, 7 January 2019

Network Simulation

Laboratory 4 - “Buffering at the link layer”

DEADLINE: January 18, 2019

In this lab, you will use the same topology from Lab 2 and the same txApp and rxApp modules developed in Lab 3, and you will implement buffering at the link layer. The purpose is to simulate the transmission time of packets toward other nodes.

Start from the link layer module as defined in Lab 3. In this module, add two queues, i.e., one for each outgoing port. When a message (in transit from another node or generated by the connected txApp/rxApp) is to be transmitted on one of the outgoing ports, it is instead inserted in the corresponding queue, provided the queue length is not greater than q_{limit}, in which case the message is discarded. When the queue is not empty, the message at the top of the queue is popped every t_{TX} seconds and immediately sent out on the corresponding gate (this behaviour can obviously be implemented using self-messages, one for each queue).

PARAMETER STUDY

Set the following default values (as usual, let *G* be your group number):

- Maximum queue size of all ports, i.e., q_{limit} = *ceil[10+G/3]*;
- Transmission time t_{TX} = 0.05s;
- Message generation interval (see Lab3) T_{ia} = exponential(G/100);
- Delay on longChannel L = 0.03s.

STEP 1 - Let the simulation end when each node has generated 5000+G*100 messages and compute the average queue length Q of the output transmission queue connecting node 0 to node 1. Adjust the value of the exponential parameter used to compute T_{ia} such that the 95% confidence interval of the estimate of Q, out of 10 different runs, loosely falls between 10% and 50% of q_{limit}.

STEP 2 - When you have determined a suitable value of T_{ia} according to STEP 1, plot the value of the average delivery delay (defined as in Lab 3) and its 99% confidence interval (out of 10 runs) as a function of 5 values of the transmission time t_{TX} .. The choice for these values is yours to make: pick values that will yield a “meaningful” plot, i.e., not a simple flat line (for example, 0.05, 0.0501, 0.0503, 0.0507 0.0509 are probably not a good choice). As for drawing the plots with confidence intervals, you will find example scripts on the portal, using the open source application *gnuplot*.

STEP 3 - For the values of t_{TX} chosen at STEP 2, draw a plot with the 99% confidence interval queue loss ratio. Consider both data messages and ACK and include only the losses *due to full queue* (do not include losses due to exceeded hop number). The queue loss ratio is thus: (the sum of dropped messages and ACKs in all queues) / (the sum of all generated messages and ACKs).

By January 18, submit a written report with:

- The metrics and plots requested in STEPs 1 to 3
- The .ini, .ned and C++ files you wrote for each of the tasks. Be sure to exhaustively comment your code.