Lab number: 7

Lab Title : RED (Random Early Detection) Queue Management

### Abstract:

*Observed the code in the red.tcl file to determine what it did, then attempted to analyse results in XGraph to no avail before researching the differences between the RED mechanism and the DropTail queue.*

### Aim objectives and goal:

*The goal is to get a better understanding of the RED queue monitor and how it functions in ns2 by looking at an example files code and modifying its value to see what changes so I can learn more about what happens.*

### Observation/ Results and Discussion:

*(note: I couldn’t get XGraph to work like mentioned in a previous lab so I couldn’t do exercise 2)*

*After looking at the code, the first thing that stood out aside from code that was already in other .tcl files I’ve been working with was the ‘create-connection TCP/Reno’ line, presumably this is for establishing TCP connections. There was also the section for tracing the queue for data packets so that the queue size can be tracked so I can see it in the XGraph plot. Then the code would let the simulator check the queue size and print results to the trace file before said queue size is used to determine whether packets go through or are discarded.*

*Some differences that I could find between the RED mechanism and the DropTail queue include the DropTail queue dropping packets if the amount of packets received by the queue is more than the current buffer size whilst the RED mechanism checks the queue length and if the queue length is too large then it drops packets randomly due to congestion, so at a glance the DropTail queue is more passive with dealing with congestion whilst the RED mechanism actively attempts to prevent congestion.*