

## ECE545: Supplementary Information for NS3 Project

### 1. Patching Traffic Control Layer

In newer NS3 versions, an additional traffic control layer has been added between network layer and MAC layer to emulate the traffic control behavior of Linux operating system. This layer comes with a queue buffer of its own, which causes a minor inconvenience for this experiment, because we are expecting to see the queuing effects at MAC layer, that is, at the queue of `PointToPointNetDevice`. You can observe this by changing the queue size of it and can find that the total throughput is not affected.

To disable the effects of this layer, make sure to execute the following steps:

1) In your script, **after** setting up IP addresses with helpers like `Ipv4AddressHelper`, add the following:

```
TrafficControlHelper tc;  
tc.Uninstall (devices);
```

And of course you need to add the corresponding header in your own script first

```
#include "ns3/traffic-control-module.h"
```

2) Find the file `src/traffic-control/model/traffic-control-layer.cc` under your ns3 files

and locate the line

```
if (!devQueueInterface->GetTxQueue (txq)->IsStopped ())
```

and comment it out.

### 2. Trace files, calculation

NS-3 provide APIs to generate trace files that record the receiving and dropping of packets. We will use `AsciiTraceHelper`.

You can calculate the throughput and loss by counting the packet records in the trace files. Alternatively one may directly do this counting inside the NS simulation program file. To do this, it is necessary to know how to set up the **tracing and callback**. For example, one may increment a counter variable in the callback function whenever the tracing source `Rx` is activated, so as to count the total number of received packets. This can be used to calculate the throughput, and the difference with the number of transmitted packets can be used to calculate the loss rate.