

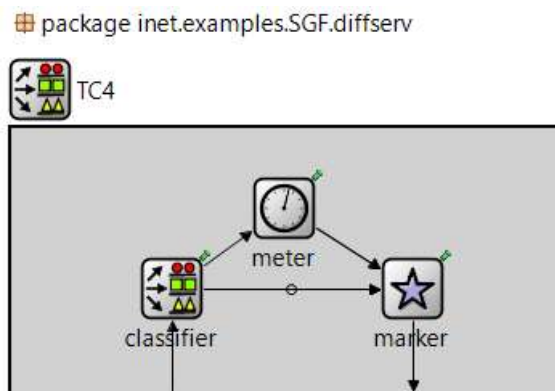
ASSIGNMENT III DIFFSERV AND MPLS

This assignment is to be done by a group of two students. The students are encouraged to prepare the assignment before the class considering the indicated parameter values. In the classroom, the professor will indicate the specific parameters to be used by each group, so the students should make themselves ready to easily adapt their calculations and software to the new parameters.

ATTENTION: Please, don't forget to show the exercises to the professor before leaving the classroom, since the progress will be evaluated.

1. Run the SGF/diffserv example.
 - Compare the configurations *wQoS* and *woQoS* regarding the following metrics, taken at the applications running at the DCC, :
 - i. Number of received packets for audio and video.
 - ii. Average delay of received packets of audio and video.
 - Change the DSCP marking of audio and video (suggestion: take advantage of the *filters.xml* file). Compare with the results obtained in a). Justify.

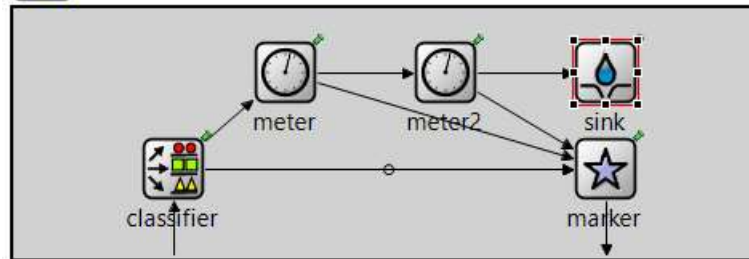
2. Create a new Traffic Conditioner *TC4*, in *TC4.ned*, which applies a *TokenBucketMeter* to EF traffic¹. EF traffic exceeding the token bucket capacity will be remarked as BE (see figure). In *omnetpp.ini*, create a new configuration named *wQoS_TC4*, similar to *wQoS*, but setting *TC4* as the traffic conditioner. Configure the *TC4* meter to limit EF traffic to 80 kbit/s, with token bucket size equal to 60 bytes. Configure the *UdpBasicBurst* (UDP port 2000 at the DCC) application running on the hosts to generate traffic at an average rate of 160 kbit/s. Check that some *UdpBasicBurst* packets that were always marked EF by the ingress routers in *wQoS*, are now marked as BE in *wQoS_TC4*.



¹ This traffic conditioner is hybrid between *TC1.ned* and *TC2.ned* in the DiffServ example scenario at [\[INETMANET\]/examples/diffserv/onedomain](#).

3. **(OPTIONAL)** Extend *TC4.ned* as a new traffic conditioner *TC5.ned*, so that the red output of the *meter* enters another *TokenBucketMeter* named *meter2*. This *meter2* imposes another token bucket, sending compliant traffic to the marker (as done in *TC4*), and eliminating excess traffic by sending it to a *sink* module² (see figure). Test *TC5.ned* in a scenario where EF traffic is marked as BE when it exceeds 80 kbit/s but does not exceed 160 kbit/s, while being eliminated when it exceeds 160 kbit/s.

package inet.examples.SGF.diffserv



4. Modify the *SGF/mpls* example so that:
- In the beginning, the traffic destined to *host4* follows the path *LSR1-LSR2-LSR6-LSR5-host4*;
 - The new path for traffic destined to *host3* at $t=2s$ is *LSR1-LSR3-LSR4-LSR5-host3*.

Check that the Path Error message is not sent anymore after the new path is created at $t=2s$. Justify.

² This kind of use of a *sink* to eliminate excess traffic is used in *TC3.ned*, in the DiffServ example scenario at [INETMANET]/examples/diffserv/onedomain.