18756 – Project 5 – MPLS with DiffServ

Overview

You have been given some Java code that represents some label switching routers on a network. Your job is to complete the code to allow communication over the MPLS network and to support DiffServ. You should implement a basic label distribution and label switching scheme based on RSVP. (Your Project 3 label switching code with a few modifications for the new format will suffice.) Additionally you will need to implement DiffServ.

The skeleton of what is needed is in place, however you will need to add a few functions and variables to make everything work correctly. You can add functions, variables, and classes to the project in whatever way you want. As students have been simply making their code match the examples and are not seriously considering other cases, we will not be providing samples to match your code to. In this project you will need to implement a routing algorithm. You could use Dijkstra's algorithm to compute the shortest path. The only example we have provided is an example network setup and sending a few packets. This is to familiarize you with the syntax for the slightly modified framework. Remember our code is documented via JavaDoc. You can use this as a reference for what the provided methods do and what parameters they take.

Grading

The output results from your code will be worth 70% of your project grade. We will test your code against test vectors and check that your outputs match the correct values. (Due to the randomness of some of the algorithms your output will not precisely match ours. This is ok.)

1. [10%] Can the code support an arbitrary network topology?
2. [20%] Can MPLS LSPs be set up from a source router to a destination router.
3. [40%] Does your network support DiffServ? (All PHBs, classes, and drop preferences.)

A report will compose 30% of your project grade. See the report section of the project guide for what should be included in your report.

4. [30%] Report.

Deliverables

1. All the Java source files only (src folder in eclipse). Delete the .class files from the working folder. (In eclipse this is the bin folder which at the time of submitting needs to be empty)
2. Your report in PDF format.

Zip the working folder into a ZIP file only. Name the file as *andrewid*_18756_project5.zip, where *andrewid* is replaced with your personal andrewid.

Upload the zip file in the blackboard under Assignments > Coursework > Project 5

Hints

You have already implemented most (almost all) of this project in your previous projects. Feel free to cannibalize ***your*** previous projects to complete this project.

For example, when performing the LSP setup you can use most of your VC setup code from project 3. However you should change the command names to the following:

Command	Explanation
Path	Establishes path from source to destination
Resv	Reserves resources along the path
ResvConf	Confirms resource reservation
PathErr	Reports issues related to Path state/setup
ResvErr	Reports issues related to Reservation state/setup

You should output

COMMAND: Router X sent a COMMAND to Router Y

COMMAND: Router Y received a COMMAND from Router X

- You don’t need to worry about label contention in our test script!

____

You should refer to the lecture notes on MPLS, RSVP, and DiffServ for examples of how messages pass through the network.

____

We are implementing a limited form of DiffServ. While we are implementing all of the PHBs, Classes, and Drop Preferences, we are only doing simple bandwidth reservation for each one. A skeleton method for bandwidth reservation is provided within each router. In our implementation, each PHB and class can only reserve bandwidth per Label (LSP) in each router (LSR). You can implement DiffServ in the LSR or in LSRNIC class. Either one is acceptable.

____

A large component of this project is reading about MPLS/RSVP/DiffServ and figuring out how they work. Feel free to ask conceptual questions about the project or these technologies on blackboard. I have created a Project 5 discussion board for this purpose. Other students are encouraged to answer questions if they know the answer.

Report

30% of the grade for this project is related to the report. This report is similar in concept to your project 3.5 report but substantially smaller in scale and scope.

Problem Setup

You should **modify** the provided **example.java** to conform to the following specifications:

In this component create a network of two routers, each with a single nic, connected via an OtoOlink. You are going to send traffic from the first router to the second router. You should send the following traffic from Router 1 to Router 2 with the provided bandwidth allocations:

Traffic Type	Bandwidth Allocation (packets per time unit)	Traffic Rate (Packets per time unit)
EF	5	5
AF – Class 1 – Drop Pref 1	5 (You only allocate per class…)	3
AF – Class 1 – Drop Pref 2		2
AF – Class 2 – Drop Pref 1	4 (You only allocate per class…)	3
AF – Class 2 – Drop Pref 2		2
AF – Class 3 – Drop Pref 1	8 (You only allocate per class…)	4
AF – Class 3 – Drop Pref 2		5
BE		40

Table 1. Traffic Type and Traffic Rate per Time Unit.

You should send the above traffic for **15 time units**.

At the end of the run, your code should output to command line: Avg Latency, Avg Jitter, Avg Throughput, and Avg Drop Rate for each of the 8 traffic types.

The required format is as follows:

<Traffic Type> : Latency - <Ave Latency> ms : Jitter - <Ave Jitter> ms : Throughput - <Ave Throughput> % : Drop Rate - <Ave Drop Rate >% :

Using the data you have generated (please include figures or tables of the above data in your report), please provide **a 1-2 page report** on the effects of using DiffServ for each traffic type.

In an additional **1-2 pages** discuss the following questions: Based on your results do you recommend implementation of DiffServ in a triple-play type network service (cable, telephone, and internet)? In these networks there are quite a few different types of traffic: standard television, on demand video through cable box, internet traffic, on demand video through web browser, and telephone traffic. Based on your experiences, how should you break these different types of traffic into different traffic categories?