Laboratory Assignment - Active Queue Management and Congestion Control

10/5/2025

Försök 1

Pågår
NÄSTA: Lämna in uppgift

Lägg till kommentar

4 försök tillåts

Information

This laboratory exercise is comprised of two parts; a practical part, where you will get some hands-on experience with some Active Queue Management (AQM) algorithms, and a theoretical part that will be examined orally. The oral examination will be based on what you have done in the practical part, so please start there.

For the practical part, you may work in pairs if you like. The oral examination will be done *individually*. If you worked together with another student for the practical part, then please add a comment stating who you worked with when submitting your results.

Make sure to read through the instructions carefully and in order.

Practical part

For this lab, we will use a Mininet setup. You can download all of the required files **here (https://canvas.kau.se/courses/24886/files/3096578?wrap=1) (https://canvas.kau.se/courses/24886/files/3096578/download?download_frd=1)** .

Step 1

The Mininet environment consists of two hosts, **h1** and **h2**, that are connected with a switch, **s1**. The environment can be started by running:

```
$ sudo ./run.sh
```

Let's begin by configuring the queuing algorithm that is to be used on the switch. Open up another terminal window (or tab), and run:

```
$ sudo ./tc.sh pfifo_fast
```

This will set the queuing algorithm on the switch to *pfifo_fast*. Now that we have configured the queuing algorithm, we are ready to take our first simple measurement. In the Mininet environment, run the following command:

```
$ xterm h1 h2
```

This will open up two new terminals, one for each of the two hosts. Now, let's generate some traffic between the two hosts. You may run any traffic generator that you want, though iperf3 is a good option. Run the following command in the **h1** terminal to perform a bandwidth test using the iperf3 tool:

```
$ iperf3 -t60 -c10.0.0.2
```

The test should report a throughput of around 8-9 Mbit/s on the receiver-side.

Now, in the window for **h2**, start a continuous ping process to observe the round-trip time between the two hosts:

```
$ ping 10.0.0.1
```

You should expect to see a round-trip time that is quite short. Now, run the iperf3 command again. Observe what happens to the round-trip time reported by the ping command. You should see that the ping times increase significantly while the iperf3 test is running.

Next, reconfigure the queuing algorithm and observe it's effect on the queuing latency. Run:

```
$ sudo ./tc.sh codel
```

This will switch the queuing algorithm to *codel*. With the ping process still running, rerun the iperf3 command and observe the ping times. This time they should be much lower. Finally, change the queuing algorithm to *fq_codel*, and rerun the test once again:

```
$ sudo ./tc.sh fq_codel
```

Step 2

With the simple measurements out of the way, let's turn to something more sophisticated. For this part of the exercise, we will use a tool called Flent. Flent is a wrapper around a number of popular network benchmarking tools. Flent allows us to easily perform advanced tests and then produce various plots of the results. We will perform a number of "real-time response under load" (rrul) tests. This type of test creates a number of competing TCP flows in both directions, while measuring packet latency for ICMP and UDP packets (you can find more info about the tests **here (https://www.bufferbloat.net/projects/bloat/wiki/RRUL_Spec/)**). We will run a version of the rrul test that also runs an emulated VoIP flow. Run the test once for each of the three queuing algorithms (i.e. pfifo_fast, codel, and fq_codel).

To run the test, run the following command on **h1**:

```
flent voip-rrul -H 10.0.0.2
```

To make it easier to distinguish between the output results, you may use the -t flag to tag the output. For example, to tag your run when using pfifo_fast, run the command:

```
flent voip-rrul -H 10.0.0.2 -t 'pfifo_fast'
```

After the test is finished, a statistical overview is printed on the screen, and a file containing the results will also have been created in the current directory (voip-rrul-[timestamp].[tag].flent.gz). Once you have run all three tests, you may use the Flent GUI to explore the results you have produced:

```
flent --gui voip-rrul*gz
```

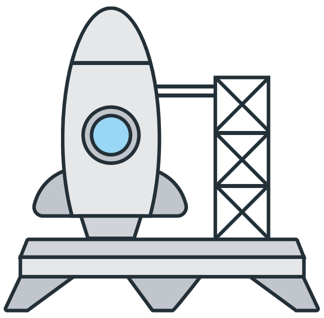
The above command will open up three tabs in the Flent GUI -- one for each dataset, and also a number of plots that you may explore. You may also save the plots as an image or pdf-file.

Theoretical part

You have now produced a number of plots that you may use for analysis. The next step is to prepare for the oral examination of the lab.

Compare the results that you have produced for the three different AQM algorithms. Which of the three algorithms would you say performed the best? Keep in mind that there may be multiple aspects to keep into consideration (e.g. throughput, latency, loss, etc.). Bring (some of) the plots that you have produced to the oral examination in order to motivate your answers. Also, upload the plots here before the oral examination.

Be prepared to explain how the different algorithms work, and how they relate to your results. Also, be prepared to explain how the AQM and Congestion Control concepts are related.



Välj en fil att ladda upp

Tillåten fil: PDF

eller

Canvas-filer

☐ Den inlämnade uppgiften är mitt egna originalarbete *

Skicka uppgift