Vysoké učení technické v Brně
Fakulta informačních technologií

Předmět IPK.

Bandwidth Measurement.

Varianta 1

Autor: Bolshakova Ksenia (xbolsh00)

Brno
09.04.2018

Content:

1. What is Bandwidth Measurement?

The first task was to understand what Bandwidth Measurement is and how it works. The definition comes from the field of engineering where bandwidth represents the distance between the highest and lowest signals on a communication channel (band). Implicitly, it most typically refers to the maximum rate at which data can be transferred, either at a single interface or across an end-to-end path.

1.1 How can you imagine it?

The simplest explanation I can think of is to compare bandwidth to the rate which you look at things. It doesn't really cover how continuous systems work but given that most people deal with digital or discontinuous systems (even if they don't know it) it's a relevant, if not perfect, analogy.

Compare a low bandwidth with having your eyes closed and then opening them briefly once every 5 seconds. Not only are you going to be slow to react to changes because you only get an update every 5 seconds (latency) but you will also miss all of the things happening while your eyes are closed (filtering). A higher bandwidth would be opening your eyes once a second. You could increase the sampling rate up until your eyes are always open.

The more rapidly you open your eyes, the higher your bandwidth. So higher bandwidth means not only can you update you output more quickly but you also capture more of the things going on.

2. The suggestion of solution.

2.1 UDP communication.

When I understood what Bandwidth Measurement is, my next task was to program an UDP. This protocol uses a simple connectionless communication model with a minimum of protocol mechanism, therefore this protocol is unreliable and datagrams may not come in order, be duplicated or even disappear without a trace. Applications what depends on time often use UDP, because it is easier to send socket again or reset it than to wait it.

2.2 Bandwidth Measurement

My program is separated to 2 parts: “reflect” and “meter”.

2.2.1 Meter.

```
./ipk-mtrip meter -h remote_host -p remote_port -s sample_size -t time
```

remote_host - the domain name or IP address of the station on which the reflector is running.

remote_port - the port number on which the spotlight is running.
sample size - the size of the data used in the "sample" of the packet
time - total time of measurement

In the part “meter” of my program I am doing all calculating. I calculated all these values: average baud rate, maximum measured speed, minimum measured speed, standard deviation and average RTT communication sockets.

But before it in this part I would send to “reflect” part a message and receive form it sockets and I will count how long it will be between one send and one receive. This time I will need for calculating. And with time I will do bigger and bigger sockets (bigger max 20% from user’s size).
In array I will save all values (size of socket / time from one send to receive) and after the time specified by the user is over I finish all calculating.
In this array I will find the max value and min value. Average value I will calculate like the sum of the whole values in these array and after divide it by number of times when send and receive were done. Standard deviation I will calculate as the sqrt() from sum of the differences between average value and every value in array. And the last is average RTT communication sockets is time divide by number of times successfully passed from send to receive.

2.2.2 Reflect.

```
./ipk-mtrip reflect -p port
```

port - The port number on which the reflector will be triggered.

Reflect part will only send to ***meter*** and receive messages from “Meter” part and it returns the length of the receive buffer.

3. Implementation

At first was the implementation of the function what would process arguments by function getopt().
The first argument was ipk-mtrip, based on the second argument (reflect / meter) the program would release *reflect* or *meter* part and after it by getopt() I processed other arguments. In *reflect* part is port, in *meter* part is remote_host, remote_port, size and time.

meter_function is for Meter part, ***reflect_function*** is for Reflect part.

3.1 reflect_function

First of all I created a socket and control address of reflect for attributing the port.
I allocated buffer for the first **recvfrom()**, because I would receive the size of buffer what I would use for messages from meter_function(). After I got a size of buffer I deleted it. When I have a size of messages I will receive. I do an array of a size of this value.
Further I have a while cycle, where I receive and send messages.

3.2 meter_function

Initialization of structure reflect_address and find IP address, after it I created a socket.
For *metter_function()* I use function *full_buffer()* to fill a buffer by letters. I allocated buffer to send to reflect the size of a messages, after the first send() I would allocate this buffer again, but the size is different. The new size is user’s size. *While cycle* depends on timer, what this function got from user. In this cycle I send messages to reflect and receive it back. The size and time of each message I save to the array. At the end of this function I calculate all needed values, using this array and after it I print it.

4. Tests

This program was tested in Linux, Windows and Merlin.

```
./ipk-mtrip meter -h merlin.fit.vutbr.cz -p 55545 -s 10000 -t 5

Průměrná přenosová rychlost: 137.100191 Mbit/s.
Maximální naměřená rychlost: 163.779525 Mbit/s.
Minimální naměřená rychlost: 108.051950 Mbit/s.
Standardní odchylka: 54.147667.
Průměrný RTT paketů komunikace: 0.030518 seconds.

./ipk-mtrip meter -h merlin.fit.vutbr.cz -p 55545 -s 10000 -t 10

Průměrná přenosová rychlost: 136.044031 Mbit/s.
Maximální naměřená rychlost: 407.843125 Mbit/s.
Minimální naměřená rychlost: 96.744181 Mbit/s.
Standardní odchylka: 106.393216.
Průměrný RTT paketů komunikace: 0.030963 seconds.

./ipk-mtrip meter -h merlin.fit.vutbr.cz -p 55545 -s 8000 -t 10

Průměrná přenosová rychlost: 167.635837 Mbit/s.
Maximální naměřená rychlost: 603.594200 Mbit/s.
Minimální naměřená rychlost: 116.011138 Mbit/s.
Standardní odchylka: 141.539469.
Průměrný RTT paketů komunikace: 0.025906 seconds.

./ipk-mtrip meter -h merlin.fit.vutbr.cz -p 55545 -s 4000 -t 10

Průměrná přenosová rychlost: 226.858741 Mbit/s.
Maximální naměřená rychlost: 756.654550 Mbit/s.
Minimální naměřená rychlost: 134.245163 Mbit/s.
Standardní odchylka: 224.295692.
Průměrný RTT paketů komunikace: 0.019012 seconds.
```

5. Sources:

Lectures,

https://robotics.stackexchange.com/questions/2620/how-to-explain-bandwidth-of-a-measurement-to-a-noob

K UROSE , J. a R OSS , K. W. Computer networking : a top-down approach.
6. vyd. New Jersey: Pearson Education, Inc., 2013. ISBN 0–13–285620–4.

http://pages.cs.wisc.edu/~suman/courses/740/papers/jain02sigcomm.pdf