Network Project Report

TCP Flavours: SACK vs VEGAS

Submitted by:

Abhinav Sharma 9915103035
Karan Arora 9915103045
Prafull Parashar 9915103058

Under the supervision of:
Mr. Bansidhar Joshi

Department of CSE/IT
Jaypee Institute of Information Technology University, Noida

November 2017

ACKNOWLEDGEMENT

We express my sincere gratitude to Mr. Bansidhar Joshi Dept. of CSE/IT, Jaypee Institute of Information Technology, Noida, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We also wish to extend our thanks to other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signature(s) of Students

Abhinav Sharma 9915103035
Karan Arora 9915103045
Prafull Parashar 9915103058

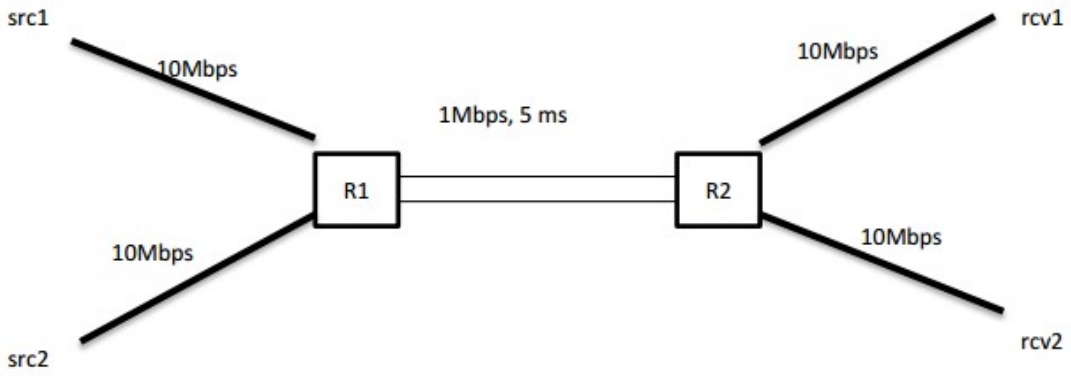
Abstract

This report presents a study of TCP flavors: SACK and VEGAS. We study the behavior of the flavors (queue size and throughput) by simulating different test cases. And at the end present a conclusion comparing both of them with each other.

SIMULATION

We use NS-2 simulator to build a simulation of the following topology and configuration:

• Two routers (R1, R2) connected with a 1 Mbps link and 5ms of latency
• Two senders (src1, src2) connected to R1 with 10 Mbps links
• Two receivers (rcv1, rcv2) connected to R2 with 10 Mbps links
• Application sender is FTP over TCP



(1) Test setup

First we need to set our configuration in our codes, and then set the variable parameter for three case:

Case 1:

•src1-R1 and R2-rcv1 end-2-end delay = 5 ms
•src2-R1 and R2-rcv2 end-2-end delay = 12.5 ms

Case 2:

•src1-R1 and R2-rcv1 end-2-end delay = 5 ms
•src2-R1 and R2-rcv2 end-2-end delay = 20 ms

Case 3:

•src1-R1 and R2-rcv1 end-2-end delay = 5 ms
•src2-R1 and R2-rcv2 end-2-end delay = 27.5 ms

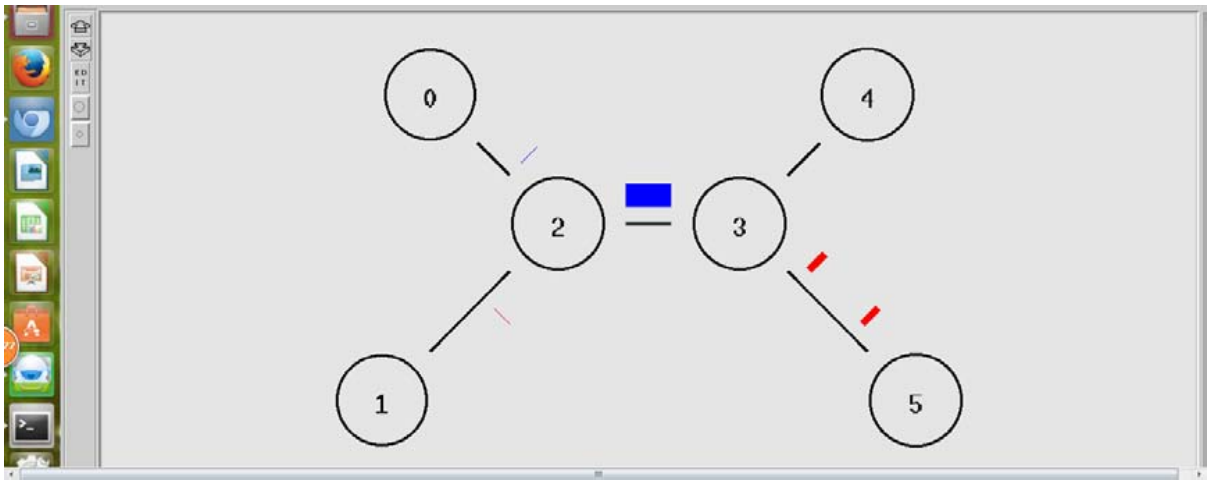
The following part of how to finish our code can be found in readme file.

(2) Test procedure After finishing the codes, firstly we need to debug our code. Run ns ns2.tcl <flavor> <case> to start our simulation:

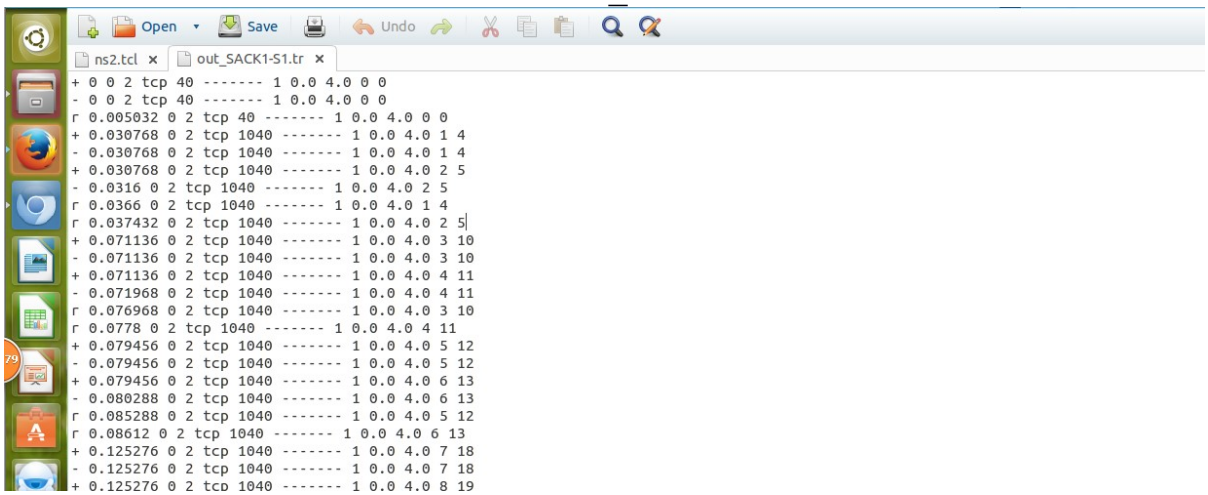
```
wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl SACK 1
Avg throughput for Src1=0.52330800000000588 MBits/sec

Avg throughput for Src2=0.47523919999999636 MBits/sec

wanli@wanli-virtual-machine:~/Downloads$
```

The then we can run the nam to see the simulation process.

And



And we can see the detail data in the out_SACK-S1.tr



After this we run the six cases one by one.

(3) Test results

After we successfully ran our codes, we can get finally results in the command line.

```
wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl VEGAS 1
Avg throughput for Src1=0.58252000000000148 MBits/sec

Avg throughput for Src2=0.41599999999999765 MBits/sec

wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl VEGAS 2
Avg throughput for Src1=0.68663999999999692 MBits/sec

Avg throughput for Src2=0.31185999999999814 MBits/sec

wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl VEGAS 3
Avg throughput for Src1=0.74901999999999647 MBits/sec

Avg throughput for Src2=0.24948000000000081 MBits/sec
```

```
wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl SACK 1
Avg throughput for Src1=0.52330800000000588 MBits/sec

Avg throughput for Src2=0.47523919999999636 MBits/sec

wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl SACK 2
Avg throughput for Src1=0.54514800000000518 MBits/sec

Avg throughput for Src2=0.45339919999999467 MBits/sec

wanli@wanli-virtual-machine:~/Downloads$ ns ns2.tcl SACK 3
Avg throughput for Src1=0.565032800000007 MBits/sec

Avg throughput for Src2=0.43349359999999548 MBits/sec
```

(4) Throughput ratio and explanation

(i)

After running all six cases, we read the simulation results from the table and then computed the ratio of the throughput. And create the following table:

TCP_flavor Case	Throughput src1 Mbps	Throughput scr2 Mbps	ratio
SACK 1	0.5233	0.4752	1.101
SACK 2	0.5451	0.4534	1.202
SACK 3	0.5650	0.4335	1.303
VEGAS 1	0.5825	0.4160	1.400
VEGAS 2	0.6866	0.3119	2.201
VEGAS 2	0.7490	0.2494	3.003

(ii)

As we can see from the table, for the TCP SACK, the RTT of the links have little impacts on the throughput. We can see the RTT ratios of three cases are 1:2, 1:3, 1:4 respectively, the ratios of the throughput are 1.101, 1.202, 1.303, which are not consistent with the RTT ratio; for the TCP VEGAS, the RTT of the links will have an obvious impact on their throughput. For the RTT ratio of 1:2, 1:3, 1:4, the trend of their throughput ratio are similar (1.400, 2.201, 3.003). We can conclude that the RTT has a significant influence on the throughput of TCP VEGAS, which means TCP VEGAS can have a bigger throughput when the delay is small. For the case 1, VEGAS has a bigger throughput compared to TCP SACK in the same ratio of delay. From this point, TCP VEGAS performs better than TCP SACK.

There are several reasons for better performance of VEGAS.

(1) VEGAS has a good estimation of incipient congestion, and efficient estimation of congestion by measuring change in throughput rather than packet loss.
(2) VEGAS is more stable than SACK. The reason for this being that SACK uses packet losses to denote congestion. So that the sender continually increases sending rate until there is congestion and then they cut back. This cycle continues and the system keeps on oscillating.
(3) SACK is not very easy to incorporate SACK in the current TCP.

Reference:

http://inst.eecs.berkeley.edu/~ee122/fa05/projects/Project2/SACKRENEVEGAS.pdf