DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

PROJECT ELEC 6181

TERM WINTER 2021

REAL-TIME AND MULTIMEDIA COMMUNICATION OVER INTERNET

TRAFFIC SHAPING AND POLICING

Submitted To

Professor Dr. ANJALI AGRAWAL

Submitted by :

Varsha Suresh (40121575)

CONTENTS

List of Figures

List of Graphs

1. OBJECTIVES

- Implementing of a network with traffic shaping and polishing by using OMNnet++ Simulator with INET Framework.
- Designing the network using 15 routers, a source host and destination host.
- Creating multiple paths between source and destination in a network topology.
- Configuring OSPF routing protocol on all the routers.
- Generating three different application such as audio, voice and video.
- Simulating to calculate end-to-end delay, queuing delay, packets dropped, and packet sent and received.

2. INTRODUCTION

The main aim of this project is to test how the traffic shaping and policing works to enhance the network resource efficiency and provide better services. It also helps us to understand the Quality of Service of the network. In this project, OMNnet++ simulator is used for the implementation for different types of network applications such as audio, voice and video.

3. BACKGROUND

Traffic shaping and policing are separate traffic conditioning systems. Both methods compare the levels of various traffic groups to a regulation or service level agreement (SLA). In terms of bandwidth, traffic rates, reliability, availability, QoS, and billing, it is typically set up between a company and a service provider.

Traffic shaping is also called as packet shaping. A congestion method and bandwidth management used in computer networks regulates data by delaying the flow as desired. Traffic shaping is the process to alter the flow of packets which ensures that a packet conforms specific specification. It police the incoming packets which can shape their traffic prior to passing it to the other network. It uses data classification, quality of system, policy rules, and queuing techniques to ensure proper bandwidth for audio, voice and video. Leaky bucket traffic shaper is used for shaping the packets in which it consist of buffering the incoming packets. There is possibility of packet loss due to buffer overflow. The incoming packets are buffered and smoothed out.

Traffic policing also known as rate limiting which monitors the flow of traffic continuously to ensure they meet the criteria of the traffic contract. The network can discard or tag the packet by giving it to low priority when the traffic contract is violated. The packets are dropped when there is a congestion. Leaky bucket algorithm is most used mechanism for policing where it can be used to police the arrival rate of packet stream. The bucket has a specific rate and specified depth for incoming packets. This is applied on the ingress bound traffic of ISP.

Traffic shaping buffers traffic that reaches the policy/agreement. Excess traffic is either reduced or marked at a lower level by policing (re-marking). These traffic conditions are deployed at the network edge. The main purpose of the traffic policing to limit the rate that is being sent in the network, send different traffic classes at different rates.

Drop tail queuing is used which is a passive queue management type where it stores the packets until the buffer gets full. When the buffer is full, the packets are dropped. This is based on FIFO mechanism to schedule the packets.

Round robin is utilized when network shares a minimum bandwidth or latency requirements, the bandwidth is guaranteed to congestion point. Queue serviced in round robin are assigned for each queue. This is done for the purpose of admission control whether to provide the flow to accepted or rejected. The flow negotiates into the network by making a contract. Through this admission control checks if sufficient resources are available for the data flow.

In this project, we have implemented these mechanisms using OMNnet++ simulator. There are 15 routers along with one source and one destination. There multiple paths connecting between host source and destination host. As Shown in figure below.

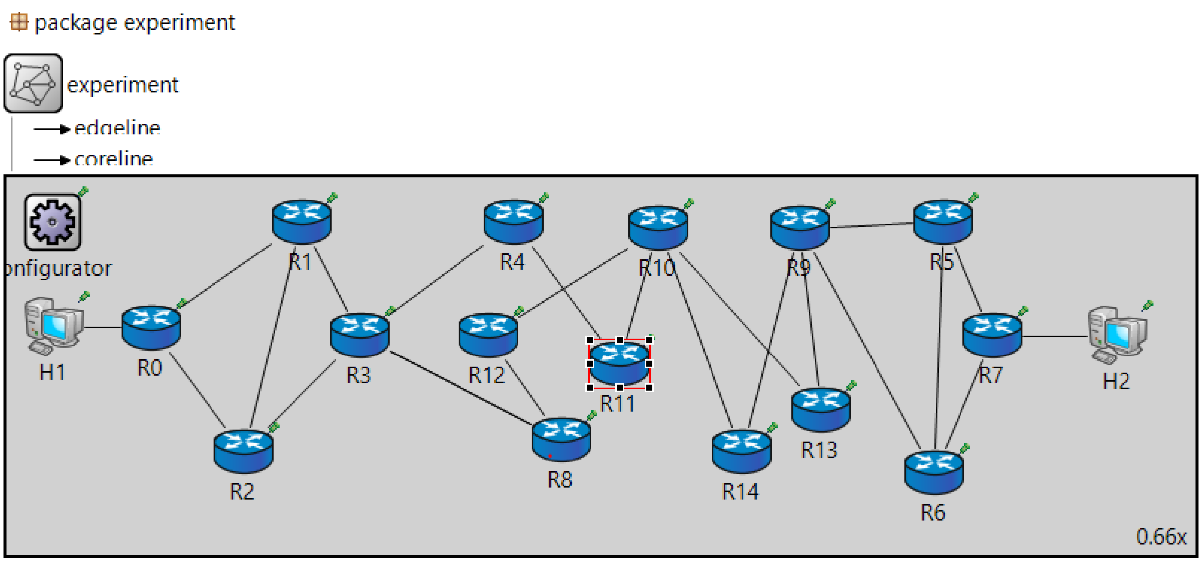


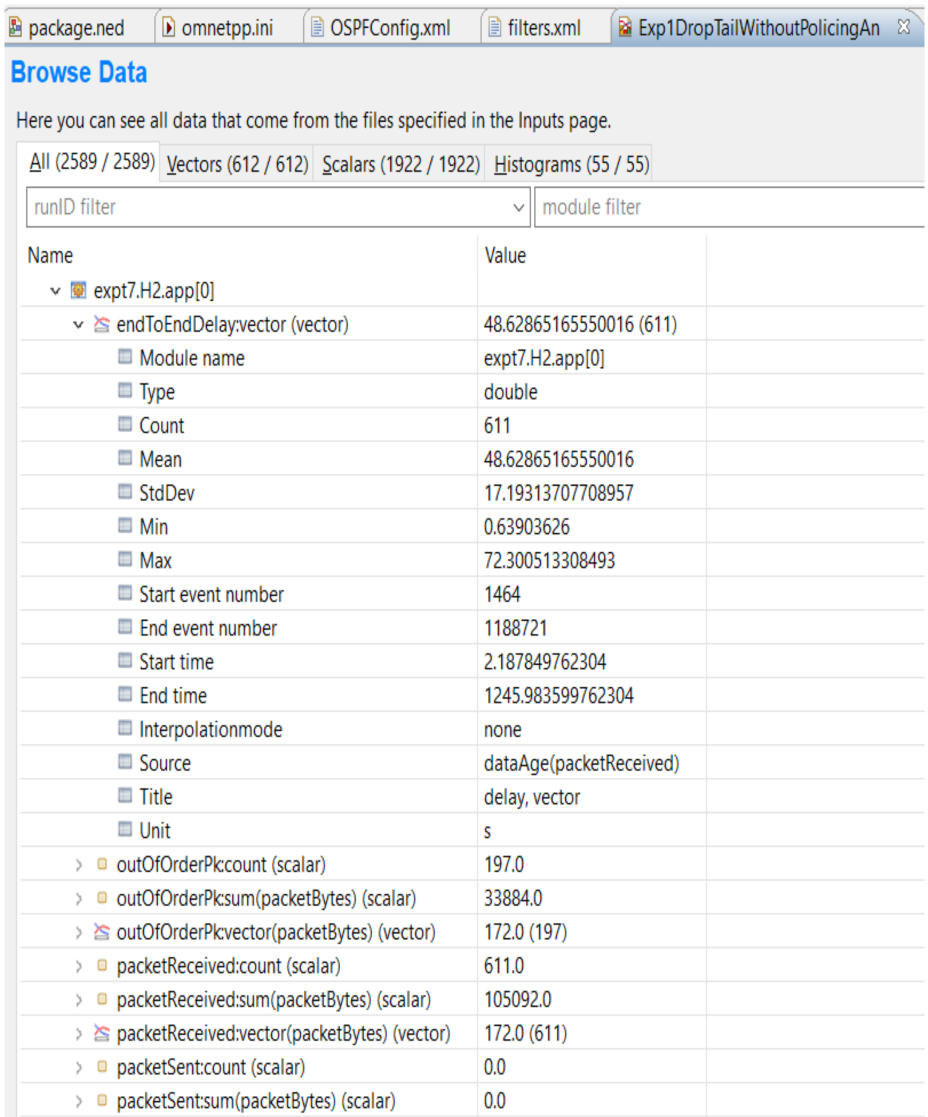
Figure 1: The design of the network topology

The policing is done Router 1 and shaping is done on Router 3. The filters.xml file and OSPFConfig.xml are used. The results are obtained in three different drop tail without shaping and policing, drop tail with shaping and policing and round robin with shaping and policing.

4. RESULTS

After simulation, new files will be created under the src file name vec and sca. Double clicking on that file we can get .anf file where we can view our results.

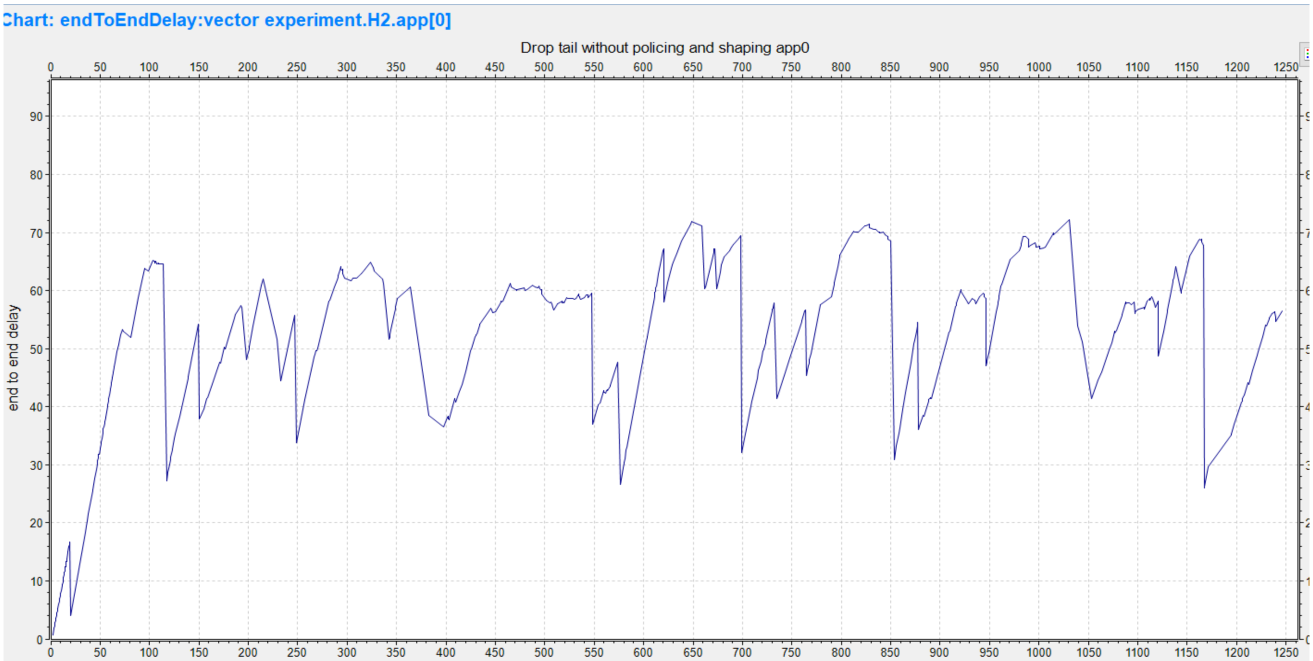
A. END TO END DELAY ,PACKET SENT AND RECEIVED FOR WITHOUT POLICING AND SHAPING



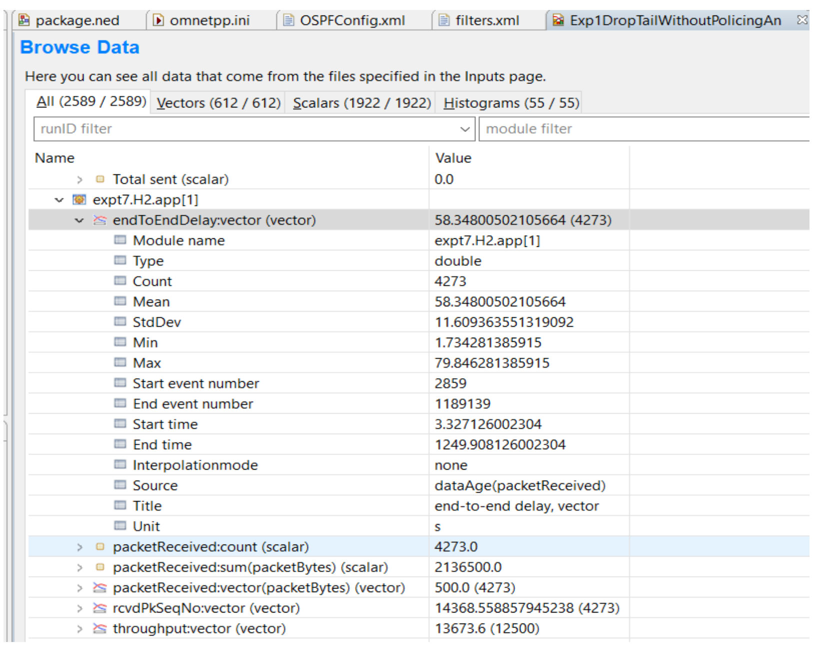
Name	Value
expt7.H2.app[0]	
endToEndDelay:vector (vector)	48.62865165550016 (611)
Module name	expt7.H2.app[0]
Type	double
Count	611
Mean	48.62865165550016
StdDev	17.19313707708957
Min	0.63903626
Max	72.300513308493
Start event number	1464
End event number	1188721
Start time	2.187849762304
End time	1245.983599762304
Interpolationmode	none
Source	dataAge(packetReceived)
Title	delay, vector
Unit	s
outOfOrderPk:count (scalar)	197.0
outOfOrderPk:sum(packetBytes) (scalar)	33884.0
outOfOrderPk:vector(packetBytes) (vector)	172.0 (197)
packetReceived:count (scalar)	611.0
packetReceived:sum(packetBytes) (scalar)	105092.0
packetReceived:vector(packetBytes) (vector)	172.0 (611)
packetSent:count (scalar)	0.0
packetSent:sum(packetBytes) (scalar)	0.0

Figure 2. End- to end delay for Drop tail without policing and shaping for application 0.

The end-to-end delay vector data is 48.62 for the application 0 that is voice streaming. The count is 611. The packets received is 611 in scalar and 172 in vector. Out of order is 197 in scalar.



Graph 1 . End- to end delay for Drop tail without policing and shaping for application 0.


package.ned | omnetpp.ini | OSPFConfig.xml | filters.xml | Exp1DropTailWithoutPolicingAn

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

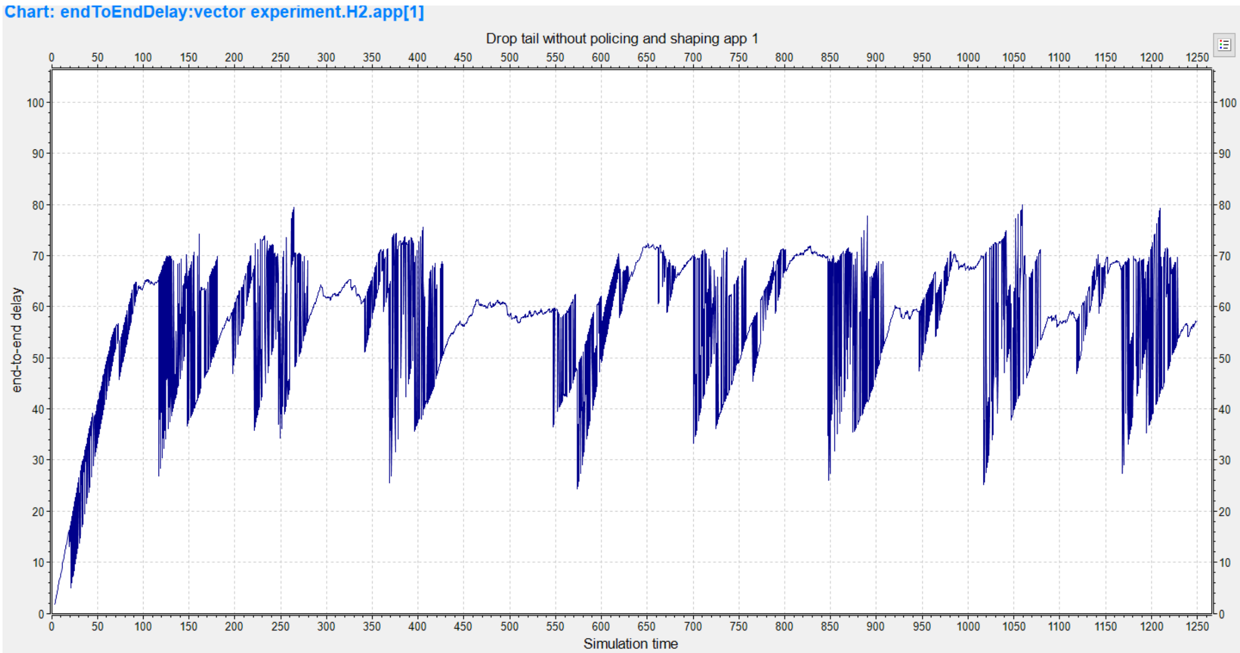
All (2589 / 2589) | Vectors (612 / 612) | Scalars (1922 / 1922) | Histograms (55 / 55)

runID filter | module filter

Name	Value
Total sent (scalar)	0.0
expt7.H2.app[1]	
endToEndDelay:vector (vector)	58.34800502105664 (4273)
Module name	expt7.H2.app[1]
Type	double
Count	4273
Mean	58.34800502105664
StdDev	11.609363551319092
Min	1.734281385915
Max	79.846281385915
Start event number	2859
End event number	1189139
Start time	3.327126002304
End time	1249.908126002304
Interpolationmode	none
Source	dataAge(packetReceived)
Title	end-to-end delay, vector
Unit	s
packetReceived:count (scalar)	4273.0
packetReceived:sum(packetBytes) (scalar)	2136500.0
packetReceived:vector(packetBytes) (vector)	500.0 (4273)
rcvdPkSeqNo:vector (vector)	14368.558857945238 (4273)
throughput:vector (vector)	13673.6 (12500)

Figure 3. End- to end delay for Drop tail without policing and shaping for application 1.

The end-to-end delay vector data is 58.34 for the application 1 that is video streaming. The count is 4273 which is same as packet received in vector. The packet received in scalar is 2136500.



Graph 2 . End- to end delay for Drop tail without policing and shaping for application 1.

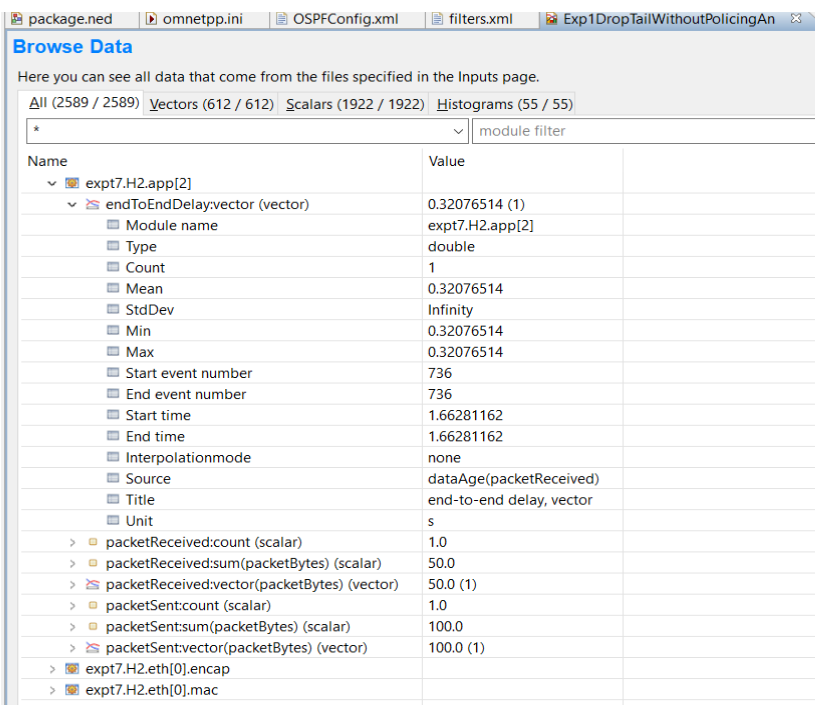
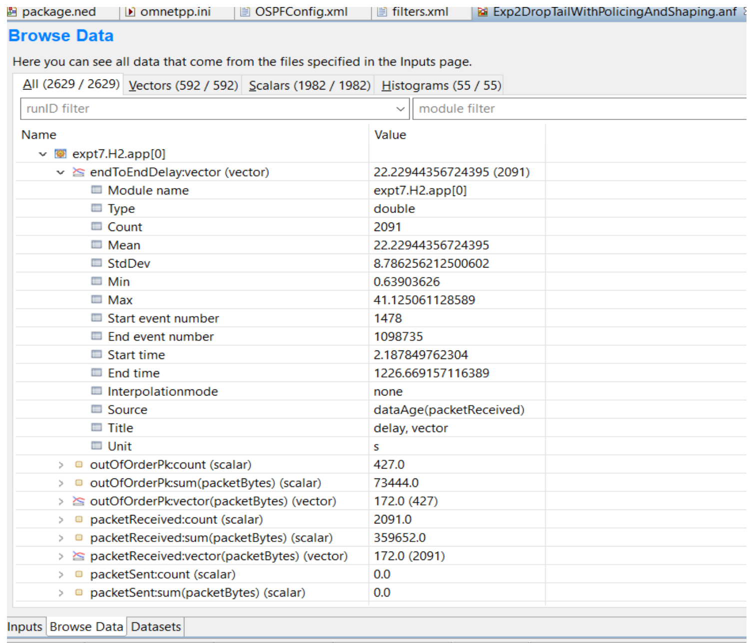


Figure 4. End- to end delay for Drop tail without policing and shaping for application 2.

The end-to-end delay vector data is 0.32 for the application 2 that is TCP Traffic streaming. The count is 1. The packet sent is 50 and packet received is 100.

B. END-TO-END DELAY, PACKET SENT AND RECEIVED FOR TRAFFIC SHAPING AND POLICING



package.ned | omnetpp.ini | OSPFConfig.xml | filters.xml | Exp2DropTailWithPolicingAndShaping.anf

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

All (2629 / 2629) | Vectors (592 / 592) | Scalars (1982 / 1982) | Histograms (55 / 55)

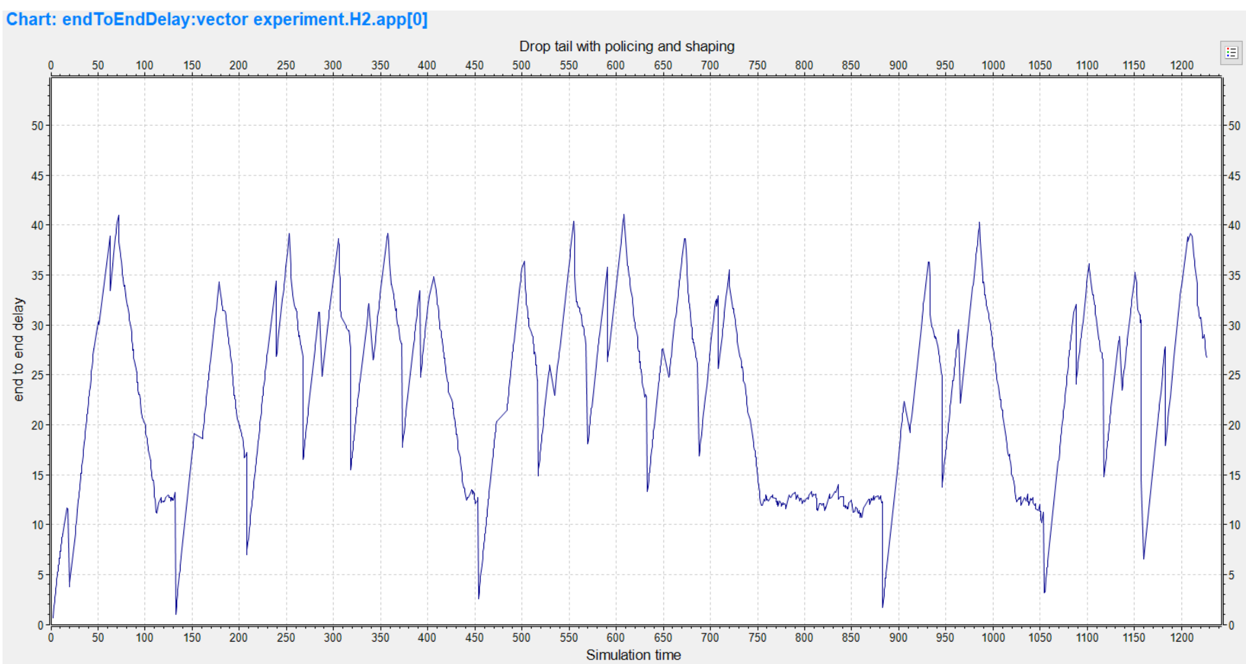
runID filter | module filter

Name	Value
expt7.H2.app[0]	
endToEndDelay:vector (vector)	22.22944356724395 (2091)
Module name	expt7.H2.app[0]
Type	double
Count	2091
Mean	22.22944356724395
StdDev	8.786256212500602
Min	0.63903626
Max	41.125061128589
Start event number	1478
End event number	1098735
Start time	2.187849762304
End time	1226.669157116389
Interpolationmode	none
Source	dataAge(packetReceived)
Title	delay, vector
Unit	s
outOfOrderPk:count (scalar)	427.0
outOfOrderPk:sum(packetBytes) (scalar)	73444.0
outOfOrderPk:vector(packetBytes) (vector)	172.0 (427)
packetReceived:count (scalar)	2091.0
packetReceived:sum(packetBytes) (scalar)	359652.0
packetReceived:vector(packetBytes) (vector)	172.0 (2091)
packetSent:count (scalar)	0.0
packetSent:sum(packetBytes) (scalar)	0.0

Inputs | Browse Data | Datasets

Figure 5. End- to end delay for Drop tail with policing and shaping for application 0.

The end-to-end delay vector data is 22.22 for the application 0 that is voice streaming. The count is 2091. The packet received in scalar is 2091 and in vector is 172.



Graph 3 . End- to end delay for Drop tail with policing and shaping for application 0.

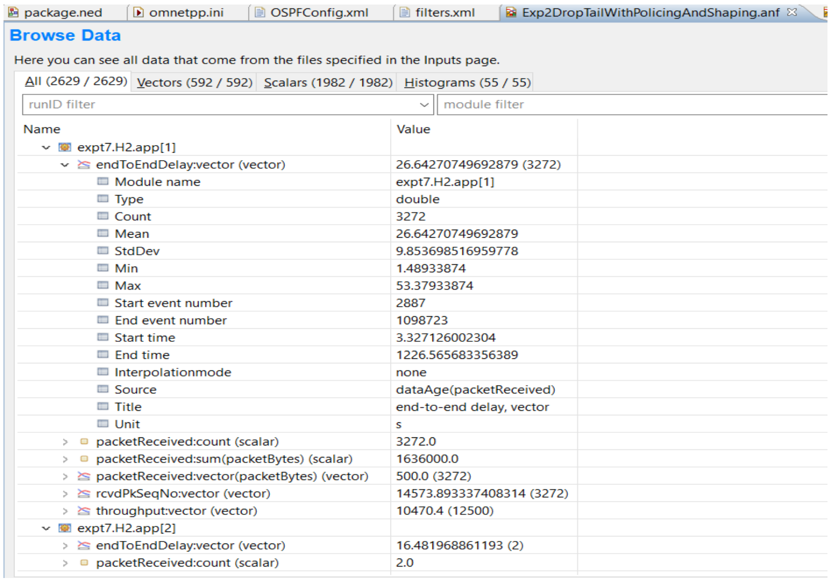
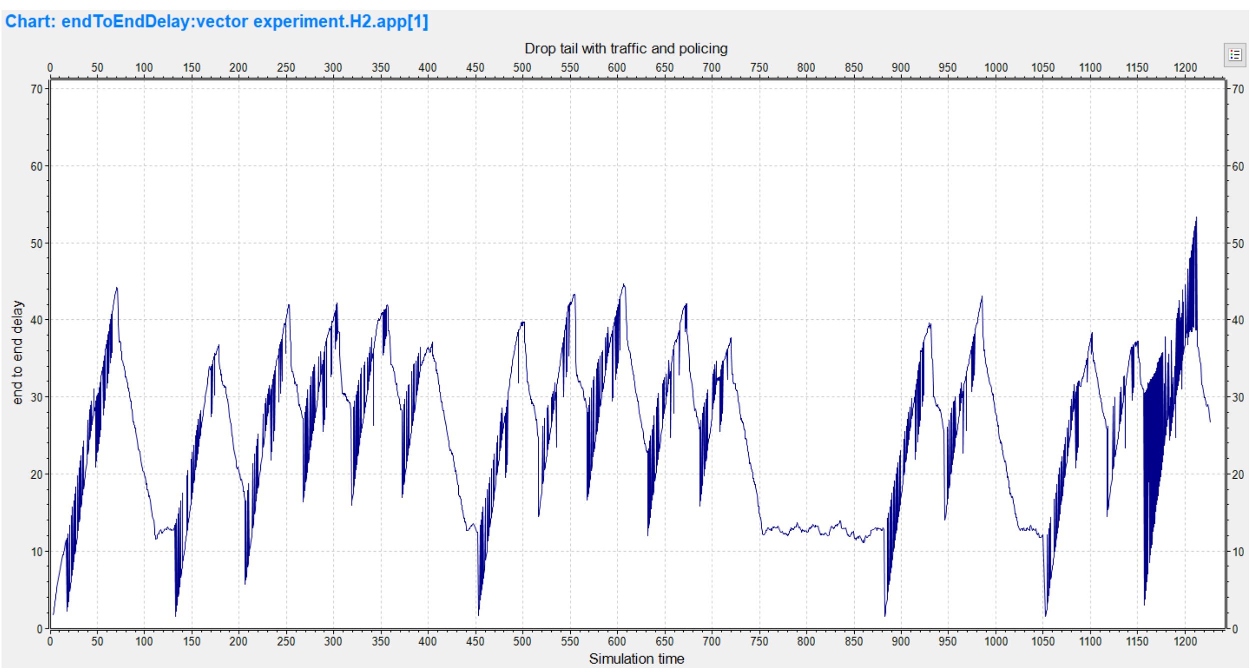
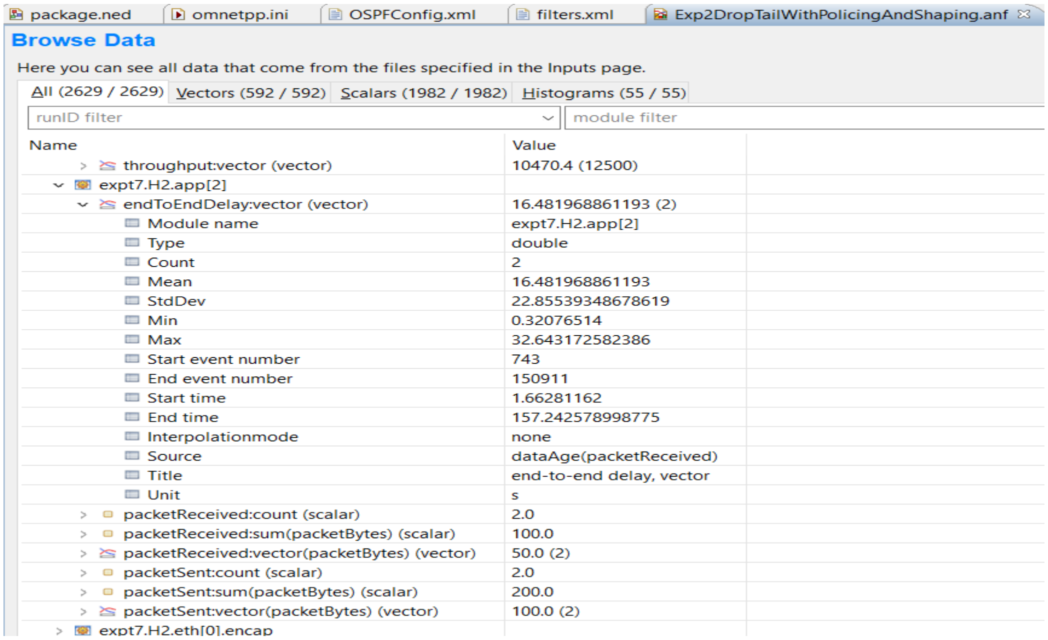


Figure 6. End- to end delay for Drop tail with policing and shaping for application 1.

The end-to-end delay vector data is 26.64 for the application 1 that is video streaming. The count is 3272 is same as packet received.



Graph 4 . End- to end delay for Drop tail with policing and shaping for application 1.



package.ned | omnetpp.ini | OSPFConfig.xml | filters.xml | Exp2DropTailWithPolicingAndShaping.anf

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

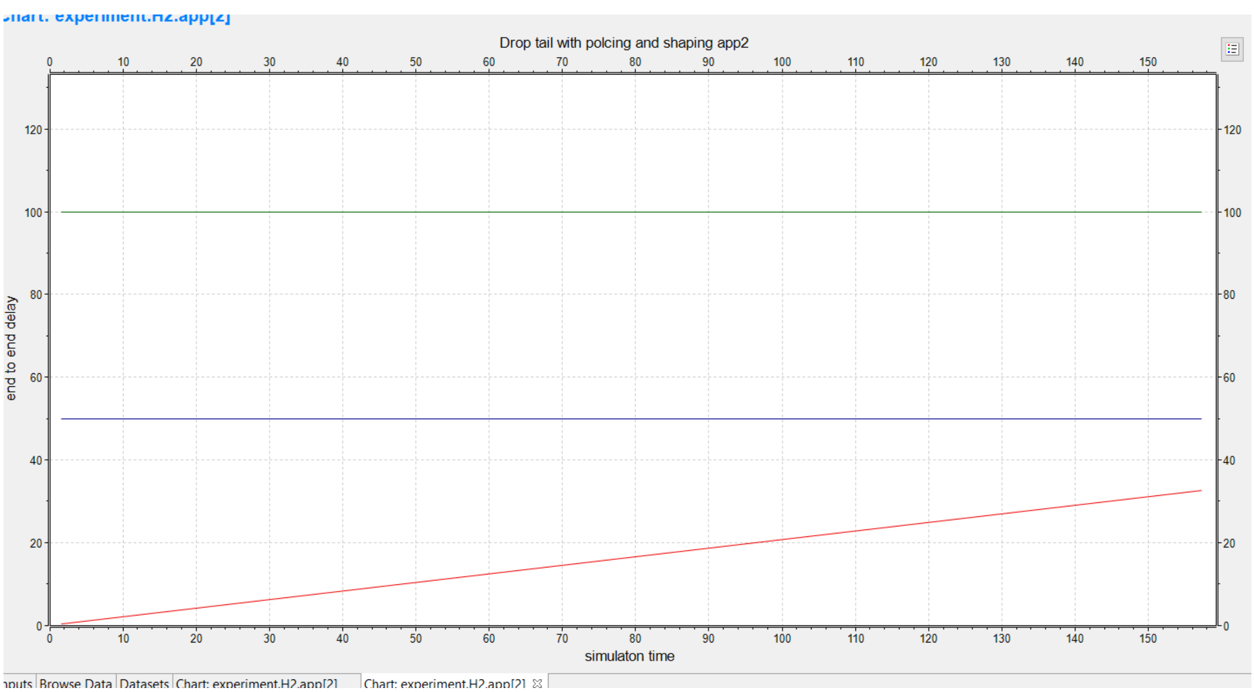
All (2629 / 2629) | Vectors (592 / 592) | Scalars (1982 / 1982) | Histograms (55 / 55)

runID filter | module filter

Name	Value
throughput:vector (vector)	10470.4 (12500)
expt7.H2.app[2]	
endToEndDelay:vector (vector)	16.481968861193 (2)
Module name	expt7.H2.app[2]
Type	double
Count	2
Mean	16.481968861193
StdDev	22.85539348678619
Min	0.32076514
Max	32.643172582386
Start event number	743
End event number	150911
Start time	1.66281162
End time	157.242578998775
Interpolationmode	none
Source	dataAge(packetReceived)
Title	end-to-end delay, vector
Unit	s
packetReceived:count (scalar)	2.0
packetReceived:sum(packetBytes) (scalar)	100.0
packetReceived:vector(packetBytes) (vector)	50.0 (2)
packetSent:count (scalar)	2.0
packetSent:sum(packetBytes) (scalar)	200.0
packetSent:vector(packetBytes) (vector)	100.0 (2)
expt7.H2.eth[0].encap	

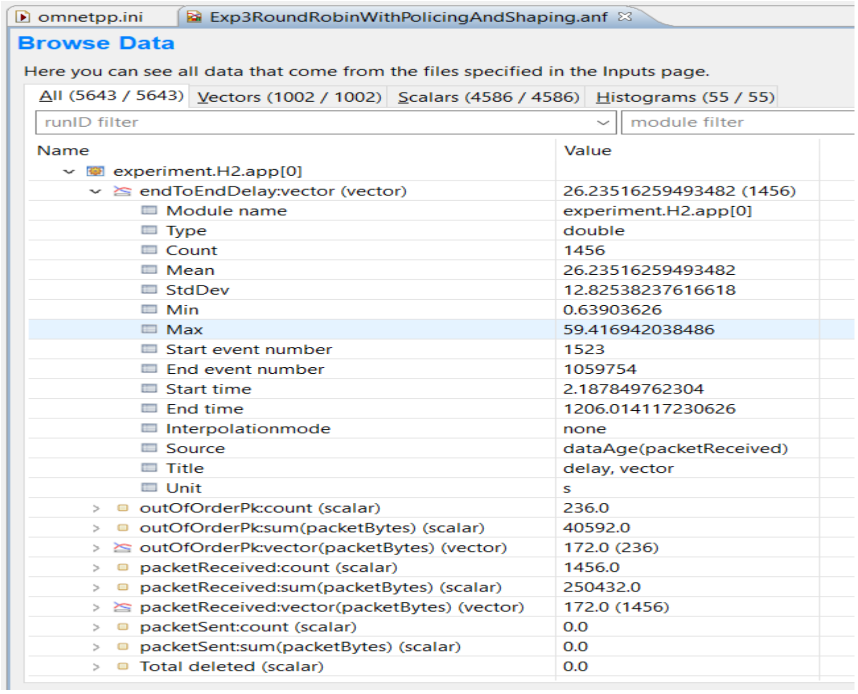
Figure 7. End- to end delay for Drop tail with policing and shaping for application 2.

The end-to-end delay vector data is 16.48 for the application 2 that is TCP Traffic streaming. The count is 2. The packet received is in scalar is 100 and sent is 200.



Graph 5. End- to end delay for Drop tail with policing and shaping for application 2.

C. END-TO-END DELAY, PACKET SENT AND RECEIVED FOR ROUND ROBIN WITH TRAFFIC SHAPING AND POLICING



omnetpp.ini | Exp3RoundRobinWithPolicingAndShaping.anf

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

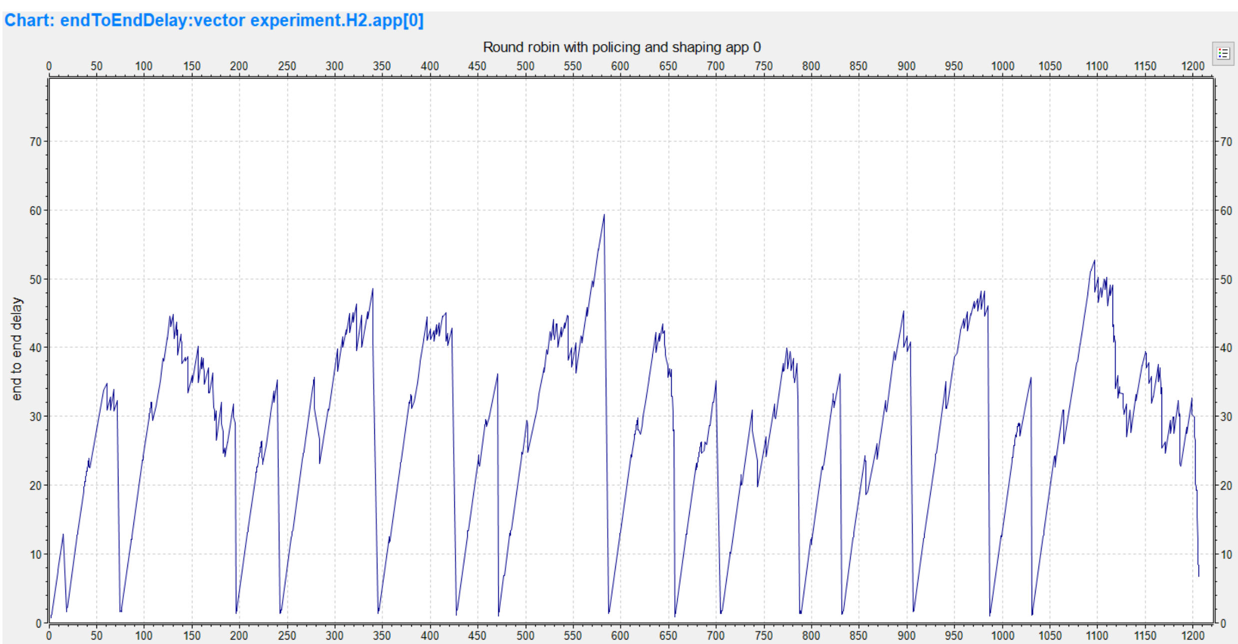
All (5643 / 5643) | Vectors (1002 / 1002) | Scalars (4586 / 4586) | Histograms (55 / 55)

runID filter | module filter

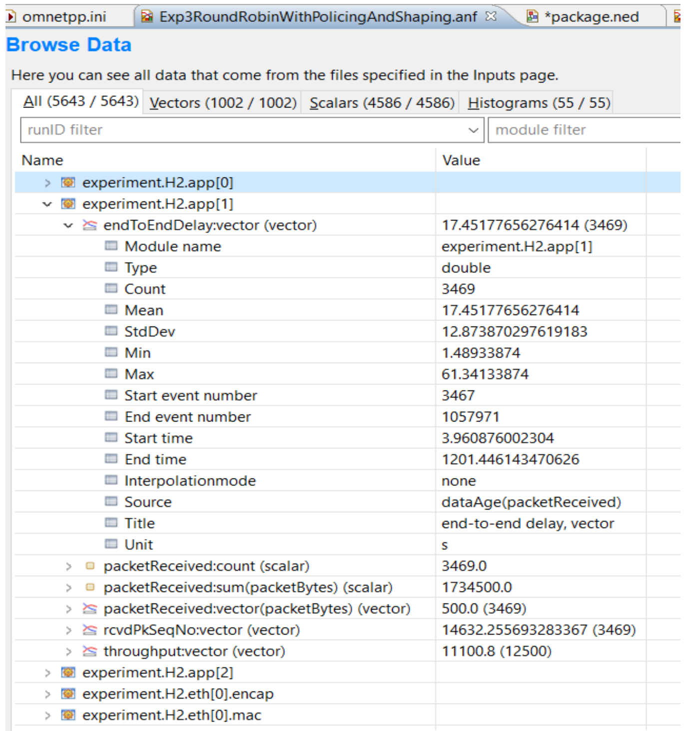
Name	Value
experiment.H2.app[0]	
endToEndDelay:vector (vector)	26.23516259493482 (1456)
Module name	experiment.H2.app[0]
Type	double
Count	1456
Mean	26.23516259493482
StdDev	12.82538237616618
Min	0.63903626
Max	59.416942038486
Start event number	1523
End event number	1059754
Start time	2.187849762304
End time	1206.014117230626
Interpolationmode	none
Source	dataAge(packetReceived)
Title	delay, vector
Unit	s
outOfOrderPk:count (scalar)	236.0
outOfOrderPk:sum(packetBytes) (scalar)	40592.0
outOfOrderPk:vector(packetBytes) (vector)	172.0 (236)
packetReceived:count (scalar)	1456.0
packetReceived:sum(packetBytes) (scalar)	250432.0
packetReceived:vector(packetBytes) (vector)	172.0 (1456)
packetSent:count (scalar)	0.0
packetSent:sum(packetBytes) (scalar)	0.0
Total deleted (scalar)	0.0

Figure 8. End- to end delay for Round Robin with policing and shaping for application 0.

The end-to-end delay vector data is 26.23 for the application 0 that is voice streaming. The count is 1456. The packet received is 20432 in scalar and out of order packets are 40592.



Graph 6. End- to end delay for Round Robin with policing and shaping for application 0.



omnetpp.ini | Exp3RoundRobinWithPolicingAndShaping.anf | *package.ned

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

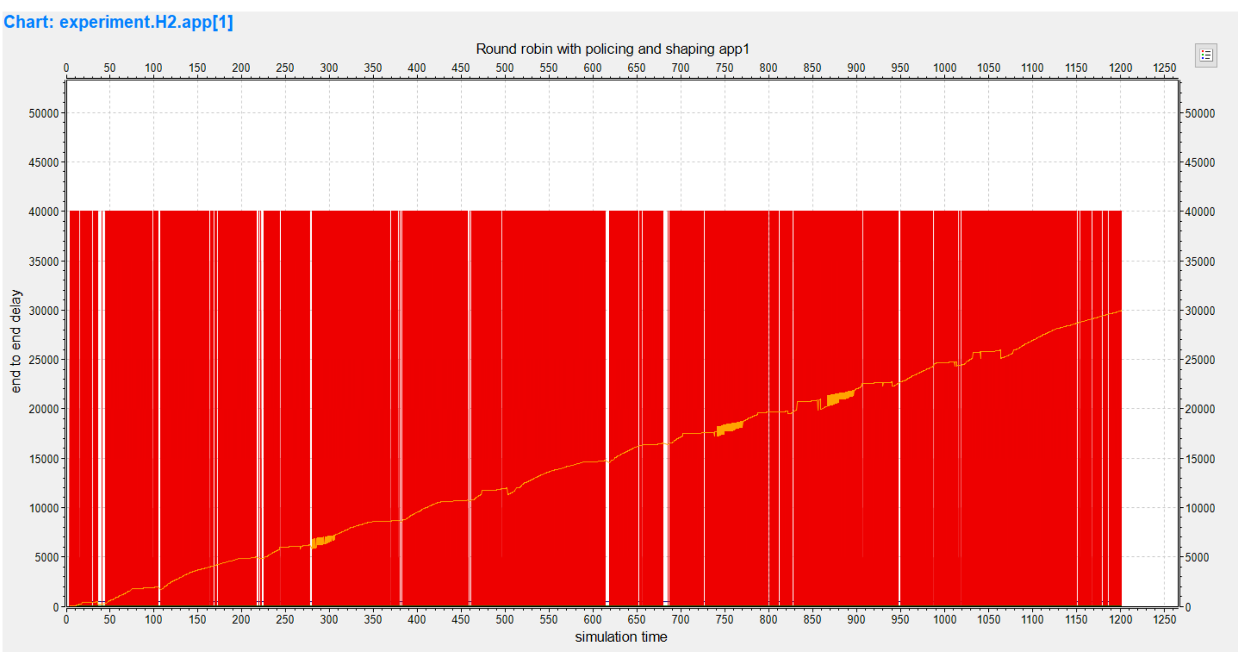
All (5643 / 5643) | Vectors (1002 / 1002) | Scalars (4586 / 4586) | Histograms (55 / 55)

runID filter | module filter

Name	Value
experiment.H2.app[0]	
experiment.H2.app[1]	
endToEndDelay:vector (vector)	17.45177656276414 (3469)
Module name	experiment.H2.app[1]
Type	double
Count	3469
Mean	17.45177656276414
StdDev	12.873870297619183
Min	1.48933874
Max	61.34133874
Start event number	3467
End event number	1057971
Start time	3.960876002304
End time	1201.446143470626
Interpolationmode	none
Source	dataAge(packetReceived)
Title	end-to-end delay, vector
Unit	s
packetReceived:count (scalar)	3469.0
packetReceived:sum(packetBytes) (scalar)	1734500.0
packetReceived:vector(packetBytes) (vector)	500.0 (3469)
rcvdPkSeqNo:vector (vector)	14632.255693283367 (3469)
throughput:vector (vector)	11100.8 (12500)
experiment.H2.app[2]	
experiment.H2.eth[0].encap	
experiment.H2.eth[0].mac	

Figure 9. End- to end delay for Round Robin with policing and shaping for application 1.

The end-to-end delay vector data is 17.45 for the application 0 that is video streaming. The count is 3469. The packets received is 3496 in vector.



Graph 7. End- to end delay for Round Robin with policing and shaping for application 1.

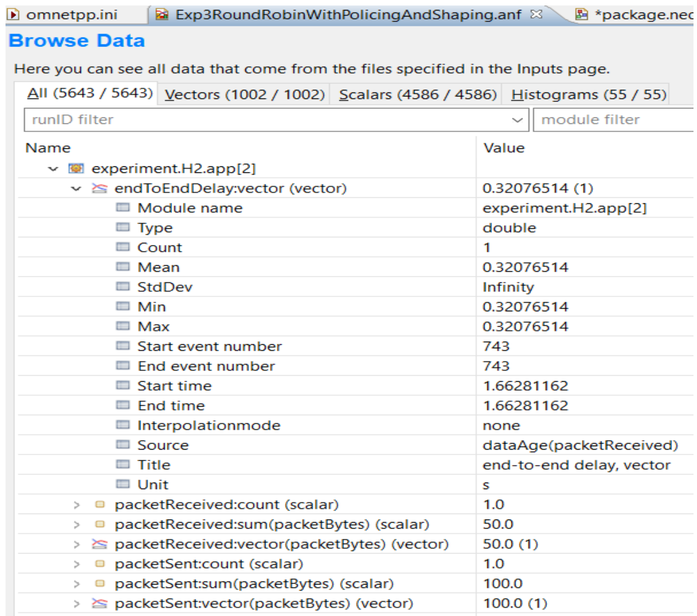
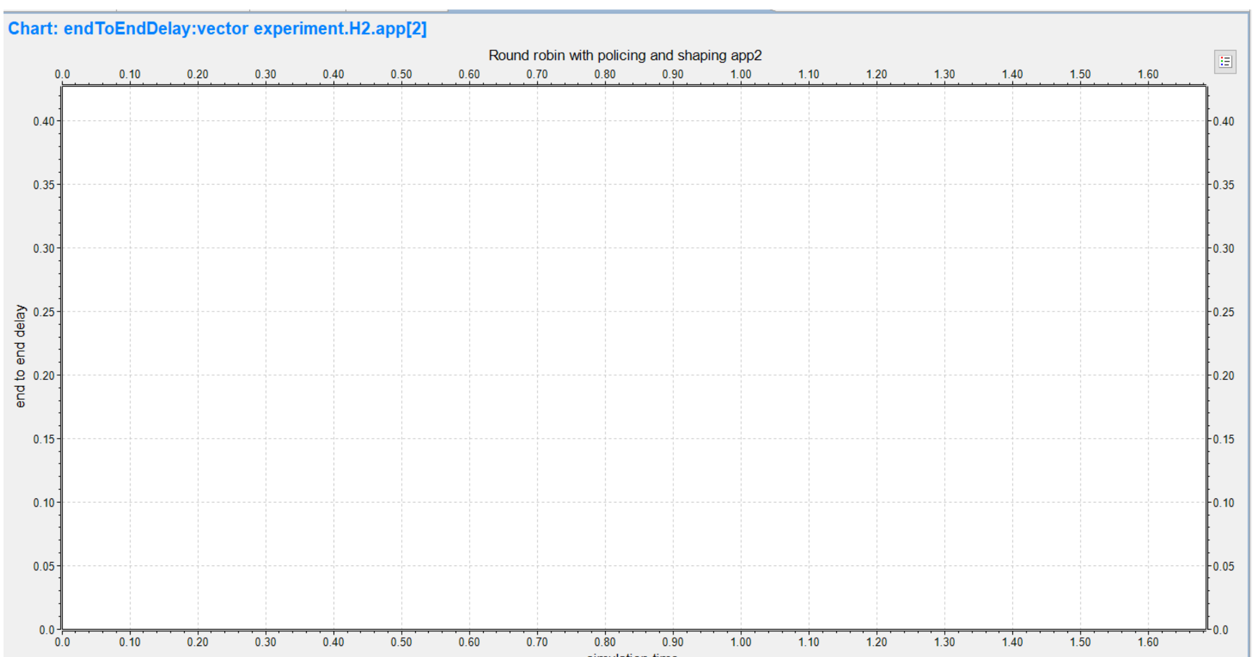


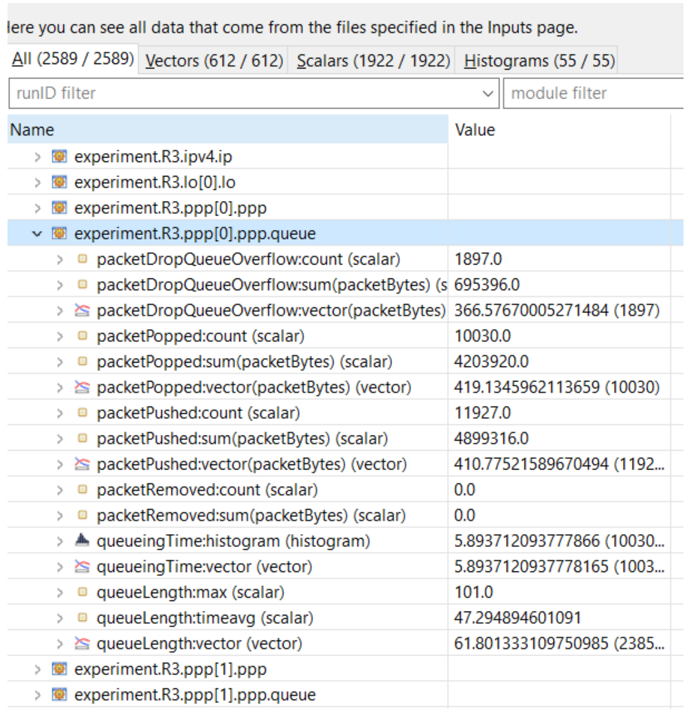
Figure 10. End- to end delay for Round Robin with policing and shaping for application 2.

The end-to-end delay vector data is 0.32 for the application 0 that is video streaming. The count is 1. The packet sent and received is 1 in scalar. The packet received is 100 and sent is 50 in vector.



Graph 8. End- to end delay for Round Robin with policing and shaping for application 2.

D. QUEUING TIME FOR DROP TAIL WITHOUT TRAFFIC SHAPING AND POLICING

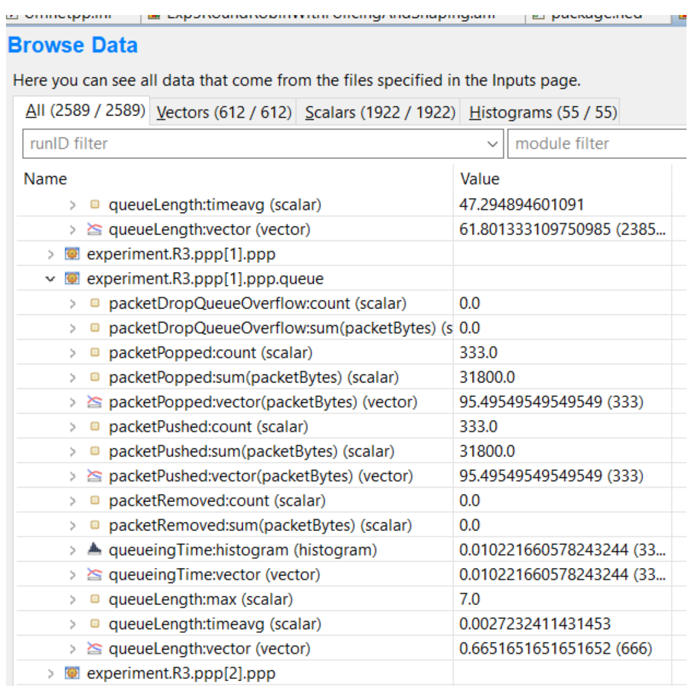

lere you can see all data that come from the files specified in the Inputs page.

All (2589 / 2589) | Vectors (612 / 612) | Scalars (1922 / 1922) | Histograms (55 / 55)

runID filter | module filter

Name	Value
experiment.R3.ipv4.ip	
experiment.R3.lo[0].lo	
experiment.R3.ppp[0].ppp	
experiment.R3.ppp[0].ppp.queue	
packetDropQueueOverflow:count (scalar)	1897.0
packetDropQueueOverflow:sum(packetBytes) (s	695396.0
packetDropQueueOverflow:vector(packetBytes)	366.57670005271484 (1897)
packetPopped:count (scalar)	10030.0
packetPopped:sum(packetBytes) (scalar)	4203920.0
packetPopped:vector(packetBytes) (vector)	419.1345962113659 (10030)
packetPushed:count (scalar)	11927.0
packetPushed:sum(packetBytes) (scalar)	4899316.0
packetPushed:vector(packetBytes) (vector)	410.77521589670494 (1192...
packetRemoved:count (scalar)	0.0
packetRemoved:sum(packetBytes) (scalar)	0.0
queueingTime:histogram (histogram)	5.893712093777866 (10030...
queueingTime:vector (vector)	5.8937120937778165 (1003...
queueLength:max (scalar)	101.0
queueLength:timeavg (scalar)	47.294894601091
queueLength:vector (vector)	61.801333109750985 (2385...
experiment.R3.ppp[1].ppp	
experiment.R3.ppp[1].ppp.queue	

Figure 11. The queuing time for application 0 at Router 3 is 5.89.


Browse Data

Here you can see all data that come from the files specified in the Inputs page.

All (2589 / 2589) | Vectors (612 / 612) | Scalars (1922 / 1922) | Histograms (55 / 55)

runID filter | module filter

Name	Value
queueLength:timeavg (scalar)	47.294894601091
queueLength:vector (vector)	61.801333109750985 (2385...
experiment.R3.ppp[1].ppp	
experiment.R3.ppp[1].ppp.queue	
packetDropQueueOverflow:count (scalar)	0.0
packetDropQueueOverflow:sum(packetBytes) (s	0.0
packetPopped:count (scalar)	333.0
packetPopped:sum(packetBytes) (scalar)	31800.0
packetPopped:vector(packetBytes) (vector)	95.49549549549549 (333)
packetPushed:count (scalar)	333.0
packetPushed:sum(packetBytes) (scalar)	31800.0
packetPushed:vector(packetBytes) (vector)	95.49549549549549 (333)
packetRemoved:count (scalar)	0.0
packetRemoved:sum(packetBytes) (scalar)	0.0
queueingTime:histogram (histogram)	0.010221660578243244 (33...
queueingTime:vector (vector)	0.010221660578243244 (33...
queueLength:max (scalar)	7.0
queueLength:timeavg (scalar)	0.0027232411431453
queueLength:vector (vector)	0.6651651651651652 (666)
experiment.R3.ppp[2].ppp	

Figure 12. The queuing time for application 1 at Router 3 is 0.01.

E. QUEUING TIME FOR DROP TAIL WITH TRAFFIC SHAPING AND POLICING

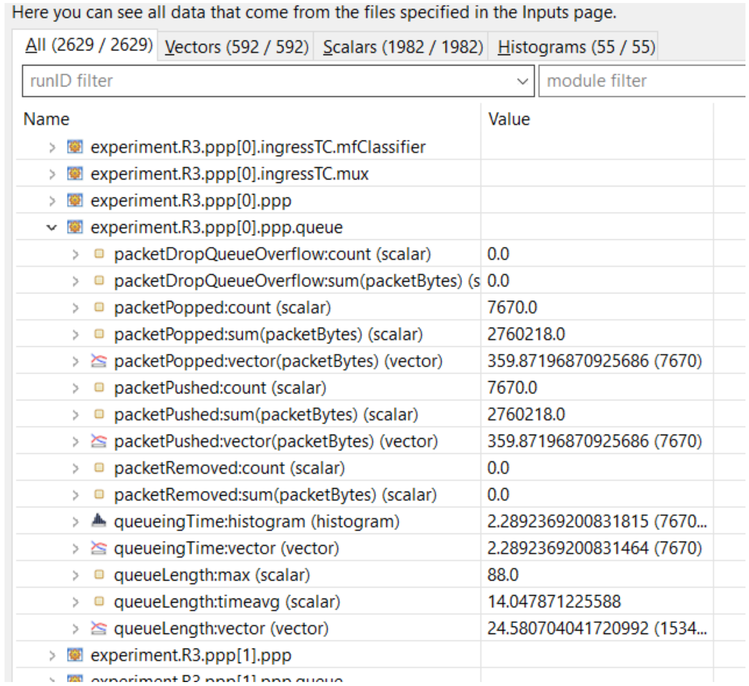


Figure 13. The queuing time for application 0 at Router 3 is 2.28.

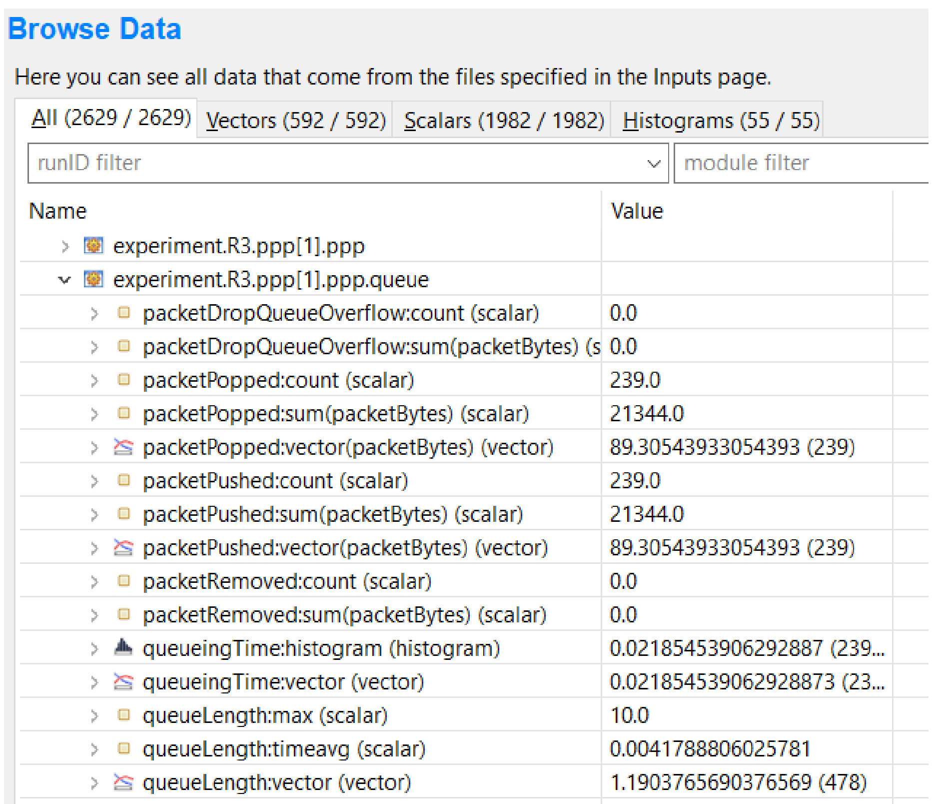
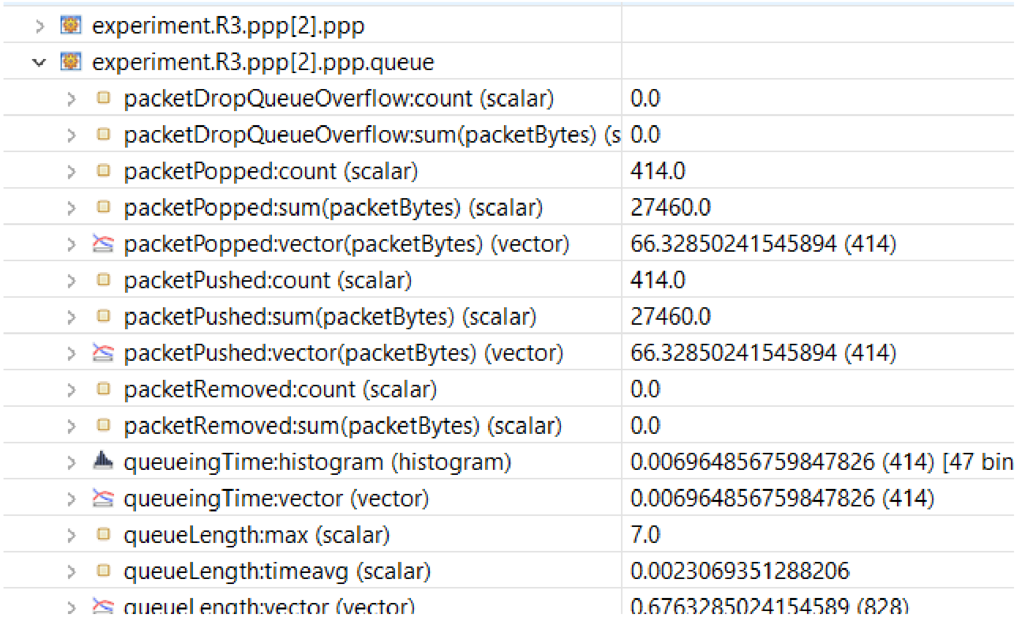


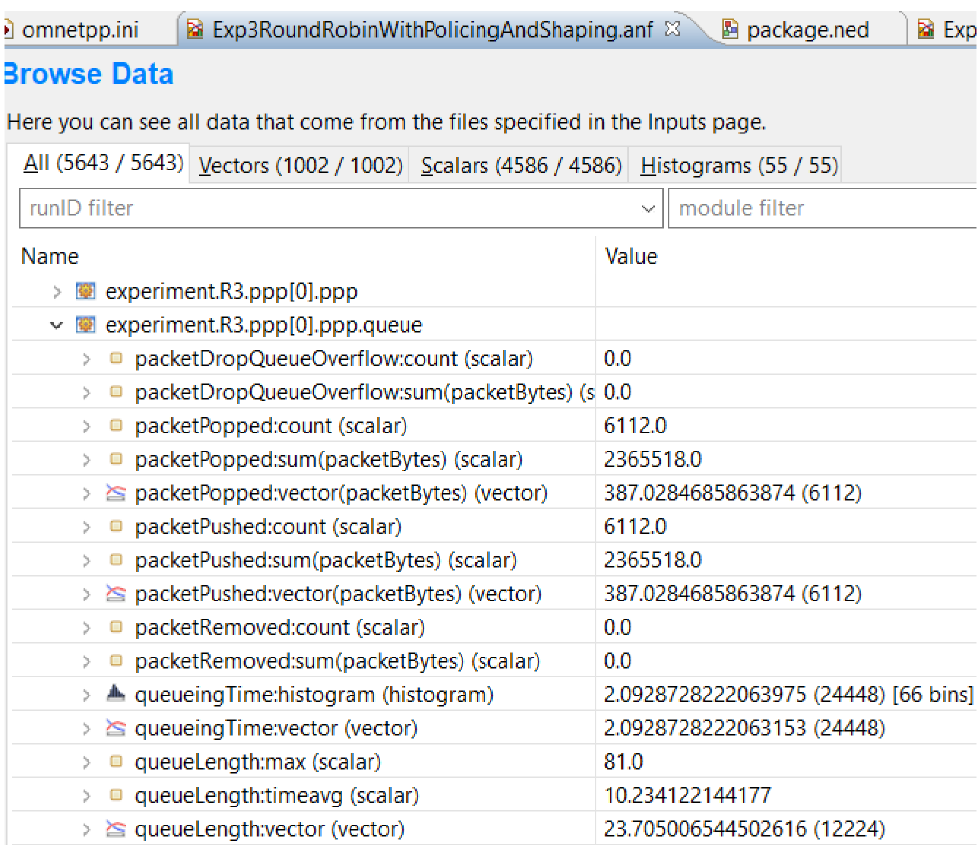
Figure 14. The queuing time for application 1 at Router 3 is 0.02.



Name	Value
experiment.R3.ppp[2].ppp	
experiment.R3.ppp[2].ppp.queue	
packetDropQueueOverflow:count (scalar)	0.0
packetDropQueueOverflow:sum(packetBytes) (s	0.0
packetPopped:count (scalar)	414.0
packetPopped:sum(packetBytes) (scalar)	27460.0
packetPopped:vector(packetBytes) (vector)	66.32850241545894 (414)
packetPushed:count (scalar)	414.0
packetPushed:sum(packetBytes) (scalar)	27460.0
packetPushed:vector(packetBytes) (vector)	66.32850241545894 (414)
packetRemoved:count (scalar)	0.0
packetRemoved:sum(packetBytes) (scalar)	0.0
queueingTime:histogram (histogram)	0.006964856759847826 (414) [47 bin
queueingTime:vector (vector)	0.006964856759847826 (414)
queueLength:max (scalar)	7.0
queueLength:timeavg (scalar)	0.0023069351288206
queueLength:vector (vector)	0.6763285024154589 (828)

Figure 15. The queuing time for application 3 at Router 3 is 0.0069.

F. QUEUING TIME FOR ROUND ROBIN WITH TRAFFIC SHAPING AND POLICING



omnetpp.ini | Exp3RoundRobinWithPolicingAndShaping.anf | package.ned | Exp

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

All (5643 / 5643) | Vectors (1002 / 1002) | Scalars (4586 / 4586) | Histograms (55 / 55)

runID filter | module filter

Name	Value
experiment.R3.ppp[0].ppp	
experiment.R3.ppp[0].ppp.queue	
packetDropQueueOverflow:count (scalar)	0.0
packetDropQueueOverflow:sum(packetBytes) (s	0.0
packetPopped:count (scalar)	6112.0
packetPopped:sum(packetBytes) (scalar)	2365518.0
packetPopped:vector(packetBytes) (vector)	387.0284685863874 (6112)
packetPushed:count (scalar)	6112.0
packetPushed:sum(packetBytes) (scalar)	2365518.0
packetPushed:vector(packetBytes) (vector)	387.0284685863874 (6112)
packetRemoved:count (scalar)	0.0
packetRemoved:sum(packetBytes) (scalar)	0.0
queueingTime:histogram (histogram)	2.0928728222063975 (24448) [66 bins]
queueingTime:vector (vector)	2.092872822063153 (24448)
queueLength:max (scalar)	81.0
queueLength:timeavg (scalar)	10.234122144177
queueLength:vector (vector)	23.705006544502616 (12224)

Figure 16. The queuing time for application 0 at Router 3 is 2.09.

experiment.R3.ppp[1].ppp.queue	
packetDropQueueOverflow:count (scalar)	0.0
packetDropQueueOverflow:sum(packetBytes) (s	0.0
packetPopped:count (scalar)	258.0
packetPopped:sum(packetBytes) (scalar)	22272.0
packetPopped:vector(packetBytes) (vector)	86.32558139534883 (258)
packetPushed:count (scalar)	258.0
packetPushed:sum(packetBytes) (scalar)	22272.0
packetPushed:vector(packetBytes) (vector)	86.32558139534883 (258)
packetRemoved:count (scalar)	0.0
packetRemoved:sum(packetBytes) (scalar)	0.0
queueingTime:histogram (histogram)	0.022877610558798448 (1032) [54 bi...
queueingTime:vector (vector)	0.022877610558798455 (1032)
queueLength:max (scalar)	12.0
queueLength:timeavg (scalar)	0.0047222696178935
queueLength:vector (vector)	1.244186046511628 (516)

Figure 17. The queuing time for application 1 at Router 3 is 0.022.

experiment.R3.ppp[3].ppp.queue	
packetDropQueueOverflow:count (scalar)	0.0
packetDropQueueOverflow:sum(packetBytes) (s	0.0
packetPopped:count (scalar)	286.0
packetPopped:sum(packetBytes) (scalar)	25324.0
packetPopped:vector(packetBytes) (vector)	88.54545454545455 (286)
packetPushed:count (scalar)	286.0
packetPushed:sum(packetBytes) (scalar)	25324.0
packetPushed:vector(packetBytes) (vector)	88.54545454545455 (286)
packetRemoved:count (scalar)	0.0
packetRemoved:sum(packetBytes) (scalar)	0.0
queueingTime:histogram (histogram)	0.010050583517947551 (11...
queueingTime:vector (vector)	0.01005058351794755 (114...
queueLength:max (scalar)	7.0
queueLength:timeavg (scalar)	0.0022997598847913

Figure 18. The queuing time for application 1 at Router 3 is 0.01.

5. DISCUSSION

Policing is done on Router 3 and shaping is done Router 0. From the above results, end-to-end delay is more for application 1 i.e for voice application when compared to video, TCP traffic. The queuing delay is more in application 0 at router 3 for all the three configurations. The count is least in TCP traffic for all the applications in all three configurations. The OSPF is configured for all the routers.

6. CONCLUSION

This project allowed to learn how the policing and shaping works in a network. The use of OMNnet++ simulator helped to understand the real- time working of the network topology. It helped to understand the use of OSPF Configuration on routers. It also helped us to understand the round robin mechanism used in the network.

7. Difficulties Faced in implementing RSVP

- Not able to configure configure rsvp on all the routers.
- Not able to implement fec file properly.
- Not sure, how to implement with multiple sources and destination hosts.
- Faced some errors while building the project and got implicit serialization.

8. REFERENCES

1. Lecture notes and lab manual.
2. http://what-when-how.com/ccnp-ont-exam-certification-guide/traffic-shaping-and-policing-congestion-avoidance-policing-shaping-and-link-efficiency-mechanisms/
3. https://support.huawei.com/enterprise/en/doc/EDOC1000047423?section=j008