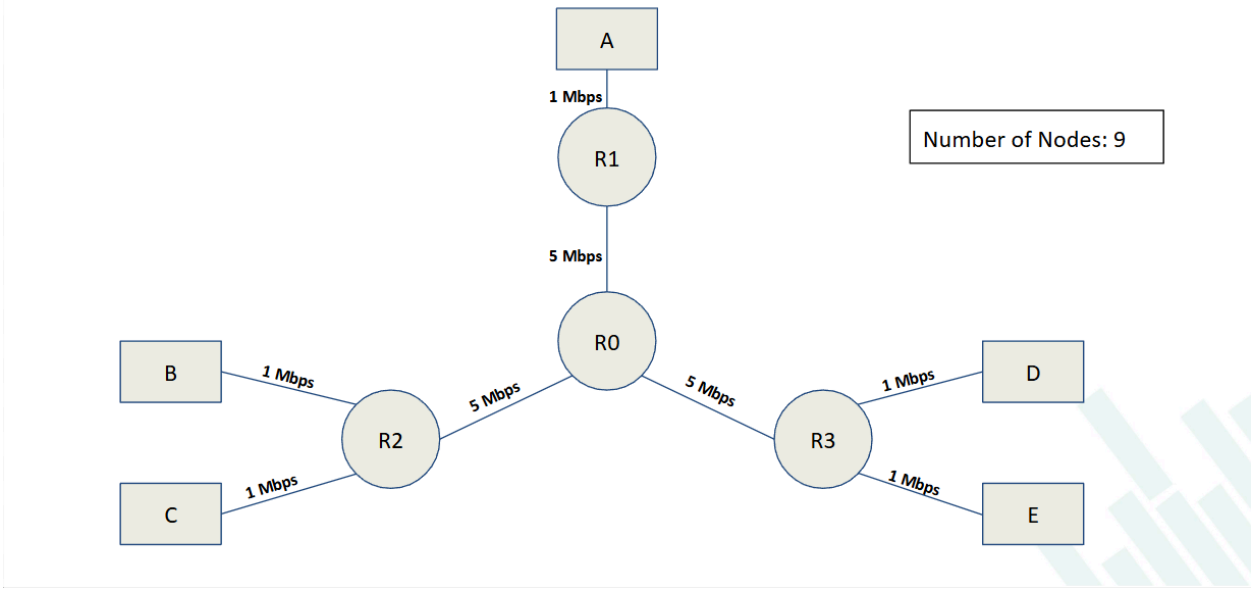
Assignment-4

NS3 Based Simulation of a Computer Network

Name: Abhinav Kumar Saxena(2022018), Akshat Gian(2022051)

Topology



Traffic matrix with source and destination workstations:

Source \ Destination	A	B	C	D	E
A	0	124	95	56	57
B	83	0	30	17	55
C	78	144	0	84	60
D	59	33	46	0	133
E	23	14	148	44	0

Complete Routing Table for All Routers and Workstations

Source \ Destination	A	B	C	D	E	R0	R1	R2	R3
A	-	R1	R1	R1	R1	R1	-	R1	R1
B	R2	-	R2	R2	R2	R2	R2	-	R2
C	R2	R2	-	R2	R2	R2	R2	-	R2
D	R3	R3	R3	-	R3	R3	R3	R3	-
E	R3	R3	R3	R3	-	R3	R3	R3	-
R0	R1	R2	R2	R3	R3	-	R1	R2	R3
R1	A	R0	R0	R0	R0	R0	-	R0	R0
R2	R0	B	C	R0	R0	R0	R0	-	R0
R3	R0	R0	R0	D	E	R0	R0	R0	-

Explanation:

- For **workstation A**:
  - To reach **B, C, D, E** or any other router, it forwards traffic to **R1** as its next hop.
- For **workstation B**:
  - It forwards traffic to **R2** as its next hop to reach any other node.
- For **workstation C**:
  - It follows the same logic as **B**. Traffic to any other node is sent to **R2**.
- For **workstations D and E**:
  - They forward traffic to **R3** to reach any other node.
- For **router R0**:
  - It chooses **R1** to reach **A**, **R2** to reach **B** and **C**, and **R3** to reach **D** and **E**.
- For **router R1**:
  - Since **R1** is directly connected to **A**, it sends traffic directly to **A**. For other nodes, it forwards packets to **R0**.

- For **router R2**:
    - **R2** forwards traffic to **R0** to reach **A, D, E**, and other routers, while directly sending packets to **B** and **C** (as it is directly connected).
- For **router R3**:
    - **R3** forwards traffic to **R0** for all other nodes except **D** and **E**, which are directly connected.

This table uses the shortest path based on the link costs shown in your topology (1 Mbps or 5 Mbps links).

Link Capacity Table:

Link	Capacity (Mbps)
A — R1	1 Mbps
B — R2	1 Mbps
C — R2	1 Mbps
D — R3	1 Mbps
E — R3	1 Mbps
R1 — R0	5 Mbps
R2 — R0	5 Mbps
R3 — R0	5 Mbps

Explanation:

- **Workstations A, B, C, D, and E** are each connected to their respective routers (**R1, R2, R2, R3, R3**) with a 1 Mbps link.
- **Routers R1, R2, and R3** are connected to the central router **R0** with 5 Mbps links.

This table outlines all the direct connections and their associated bandwidth capacity in your network.

Inducing Error and Giving Queue Length for Queueing Time

This Header File has been imported for giving packet loss error

```
#include "ns3/error-model.h"
```

Error model objects have been created namely errorModel1 and errorModel2. ‘errorModel1’ object has been given an error rate of 0.075% and ‘errorModel2’ has been given an error rate of 0.8%. Then, each device in every NetContainerDevice object has been assigned to one of the two error models simulating packet loss.

```
//Assigning Error Models
Ptr<RateErrorModel> errorModel1 = CreateObject<RateErrorModel>();
errorModel->SetAttribute("ErrorRate", DoubleValue(0.00075));
Ptr<RateErrorModel> errorModel2 = CreateObject<RateErrorModel>();
errorModel->SetAttribute("ErrorRate", DoubleValue(0.0008));

devicesAR1.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesAR1.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesR1A.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesR1A.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesR1R0.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesR1R0.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesR2R0.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesR2R0.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesR3R0.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesR3R0.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesBR2.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel)2);
devicesBR2.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesCR2.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesCR2.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesDR3.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel1));
devicesDR3.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesER3.Get(0)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
devicesER3.Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModel2));
```

Queue Length has been specified in the pointer to pointer object that we have created and here the packet queue length is 50 packets.

```
p2p.SetQueue("ns3::DropTailQueue<Packet>","MaxSize", QueueSizeValue(QueueSize("50p")));
```

End-to-End one way delay matrix:

Unit : **milliseconds**

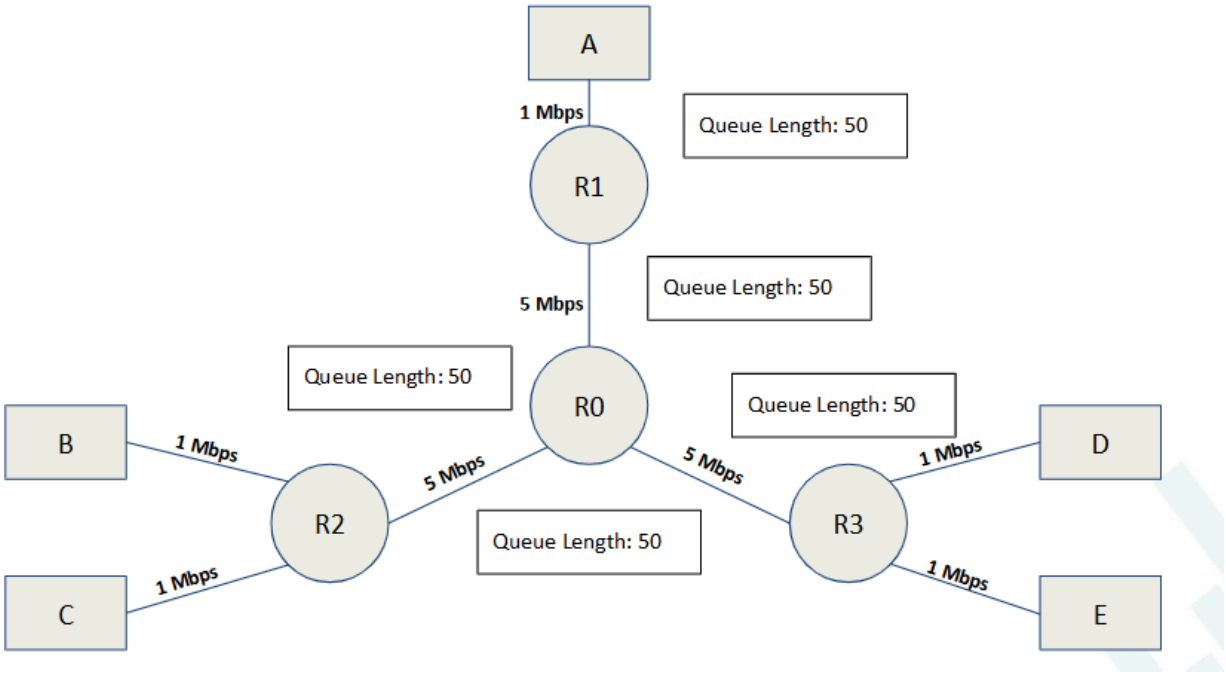
Source \ Destination	A	B	C	D	E
A	0	8.9152	8.9152	8.9152	8.9152
B	8.9152	0	6.096	8.9152	8.9152
C	8.9152	6.096	0	8.9152	8.9152
D	8.9152	8.9152	8.9152	0	6.096
E	8.9152	8.9152	8.9152	6.096	0

(Delay in **ms**)

Packet Drops in the form of source/Destination matrix:

Source \ Destination	A	B	C	D	E
A	0	3.7076	2.8405	1.6744	1.7043
B	2.4817	0	0.4482	0.5083	1.6445
C	2.3322	2.15136	0	2.5116	1.794
D	1.7641	0.9867	1.3754	0	1.98702
E	0.6877	0.4186	4.4252	0.65736	0

Queue Length at each of the outgoing links in the router

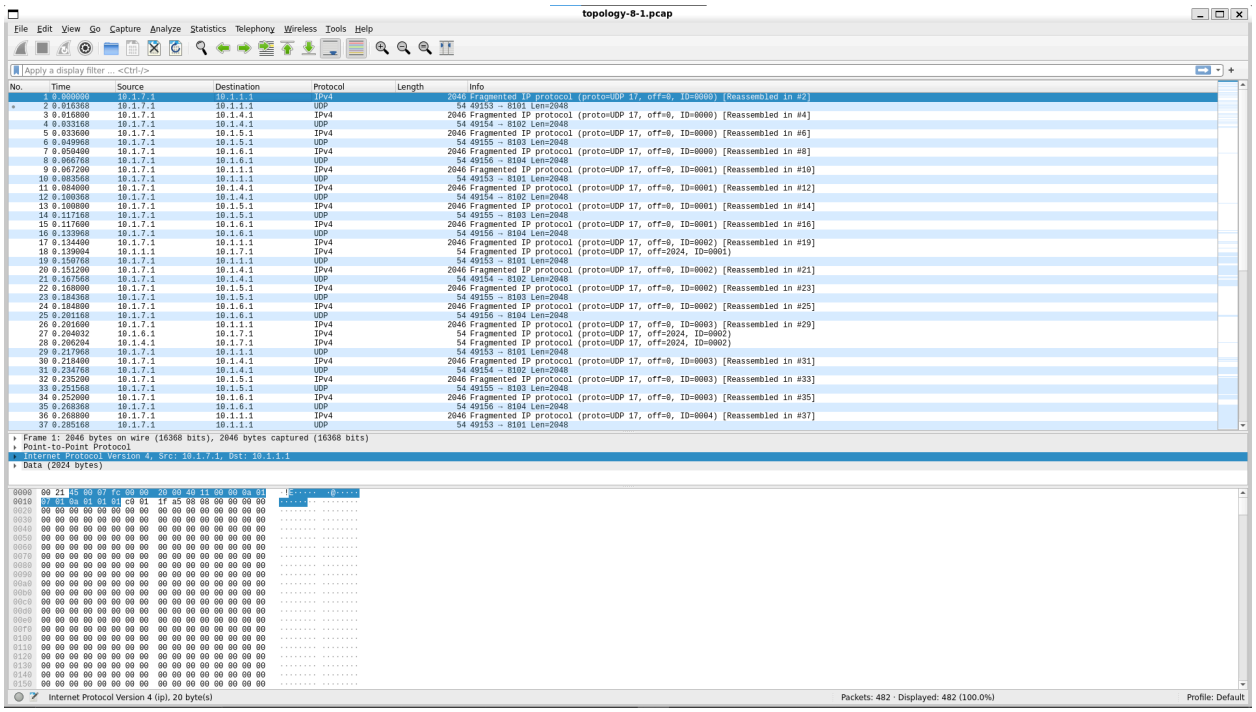


The Max Queue Length is 50 and is the same for each of the outgoing links as defined in the code. If Queue Length increases to more than 50 packet gets dropped otherwise it remains in queue.

The code for defining Queue Length is:

```
p2p.SetQueue("ns3::DropTailQueue<Packet>","MaxSize", QueueSizeValue(QueueSize("50p")));
```

Packet Observation on Wireshark

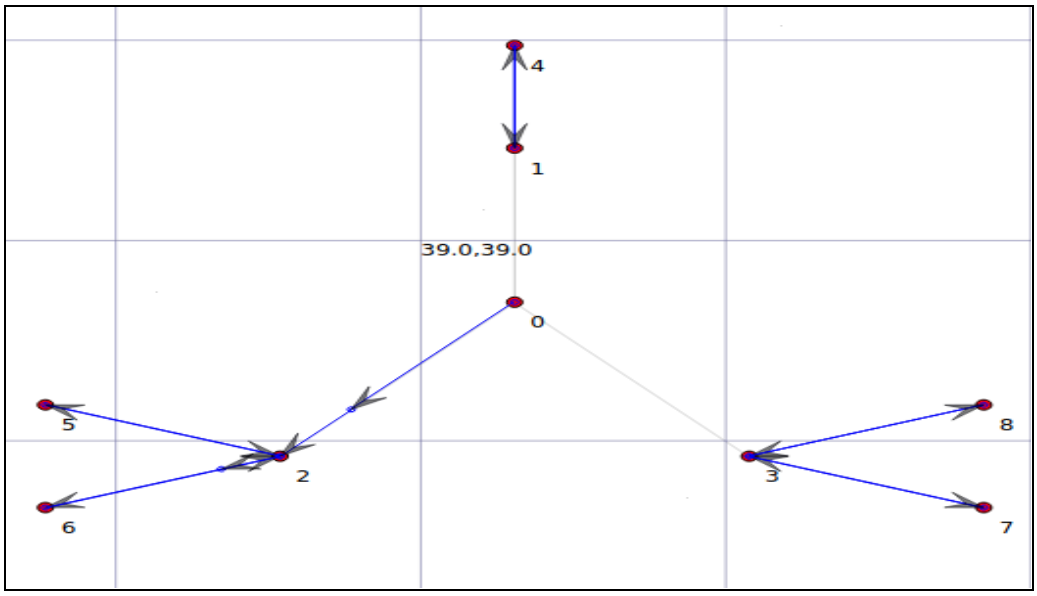


Here Frame Number, Time of Packet Sent, Source, Destination, Protocol and Length of Frame and some information regarding that can be seen.

MTU can be changed by changing the numerical values in below code

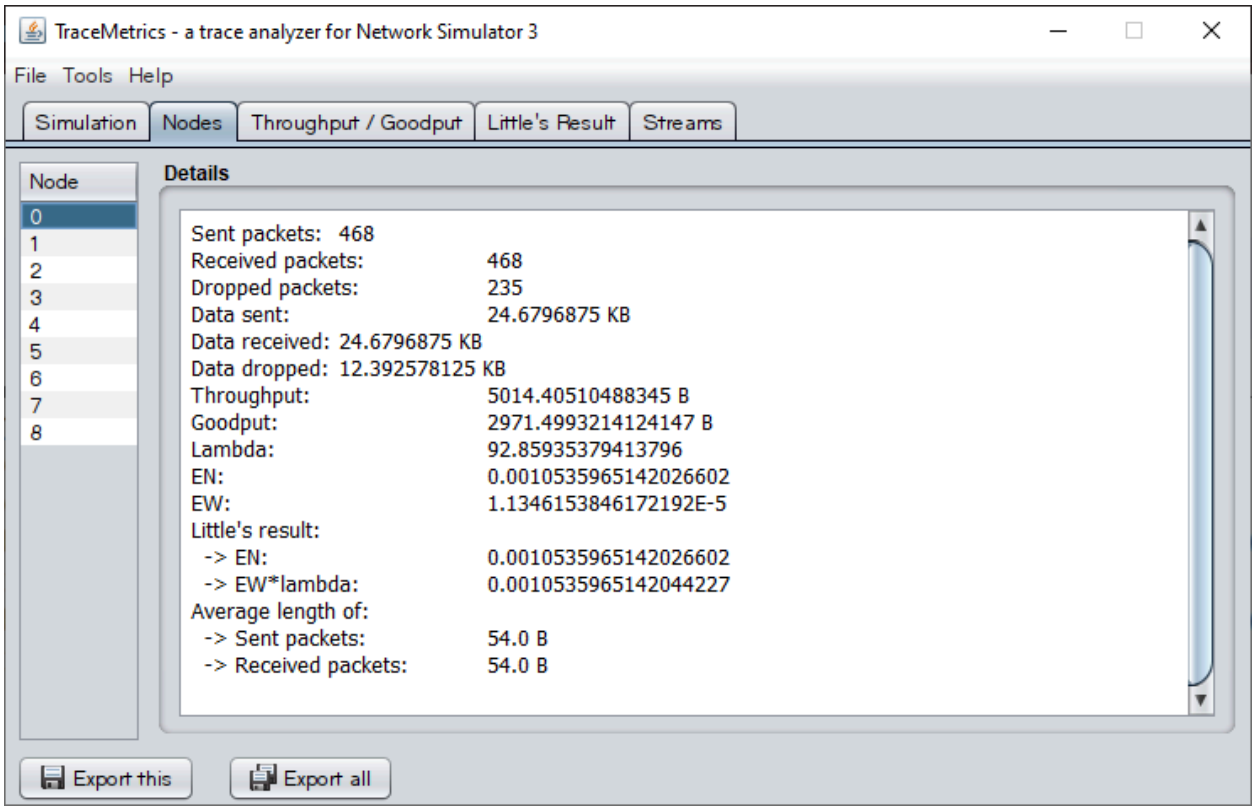
```
for(uint32_t i=0; i<devices.size();i++){
Ptr<NetDevice> device1 = devices[i].Get(0);
Ptr<NetDevice> device2 = devices[i].Get(1);
Ptr<PointToPointNetDevice> p2pDevice1 = DynamicCast<PointToPointNetDevice>(device1);
p2pDevice1->SetMtu(2048);
Ptr<PointToPointNetDevice> p2pDevice2 = DynamicCast<PointToPointNetDevice>(device2);
p2pDevice2->SetMtu(2048);
}
```

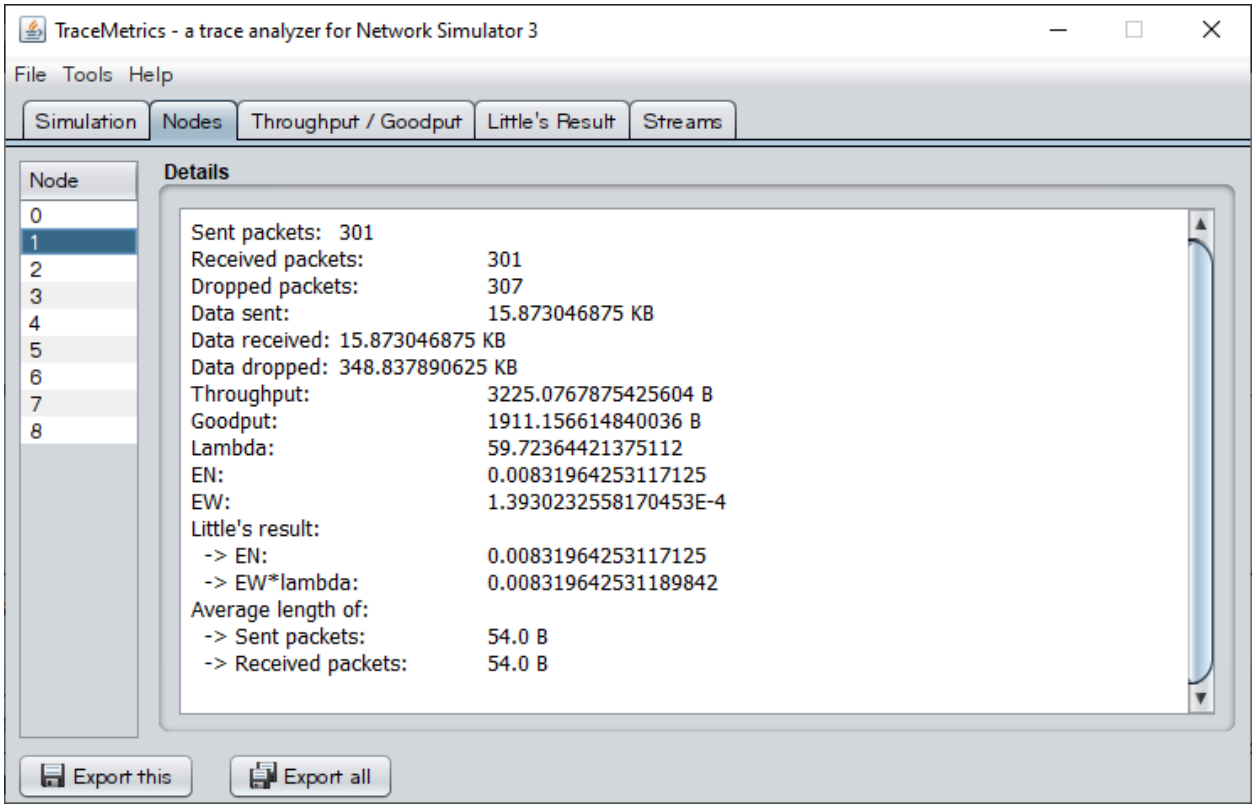
The packets can be traced using the NetAnim Simulator in NS3.

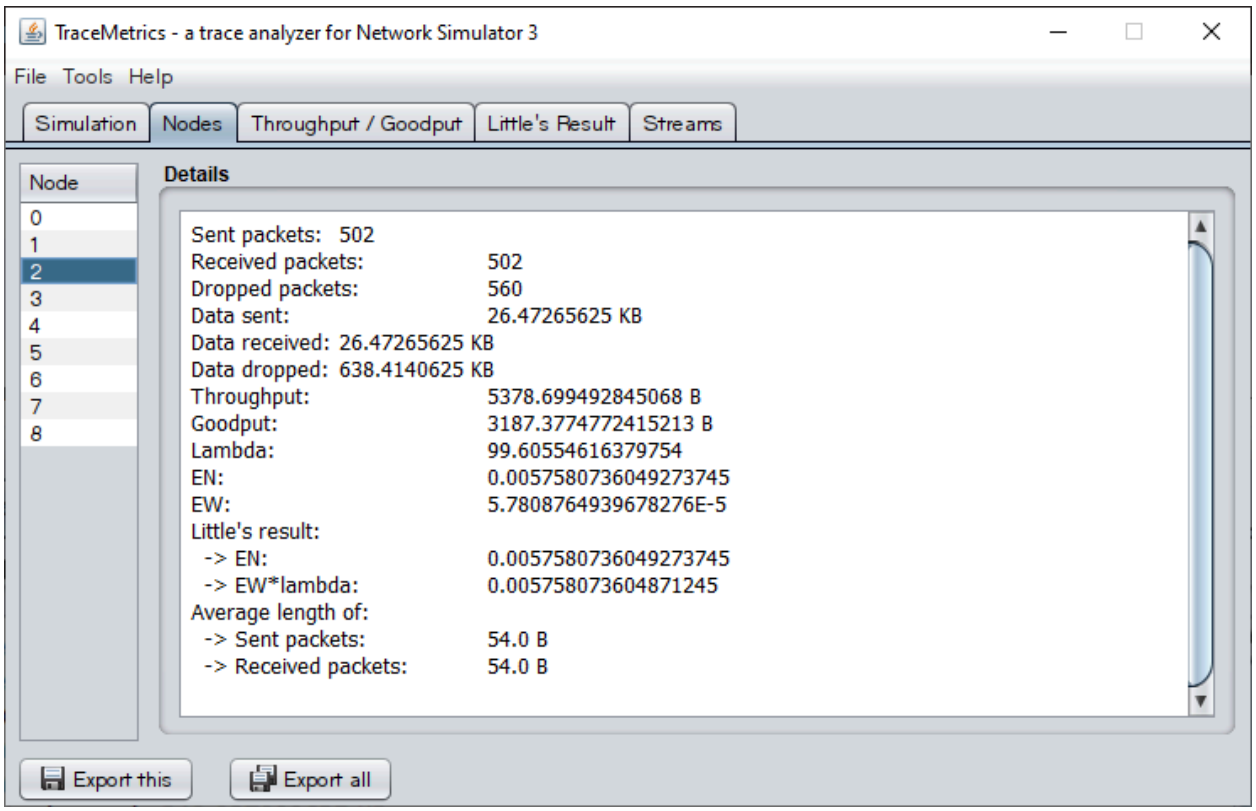


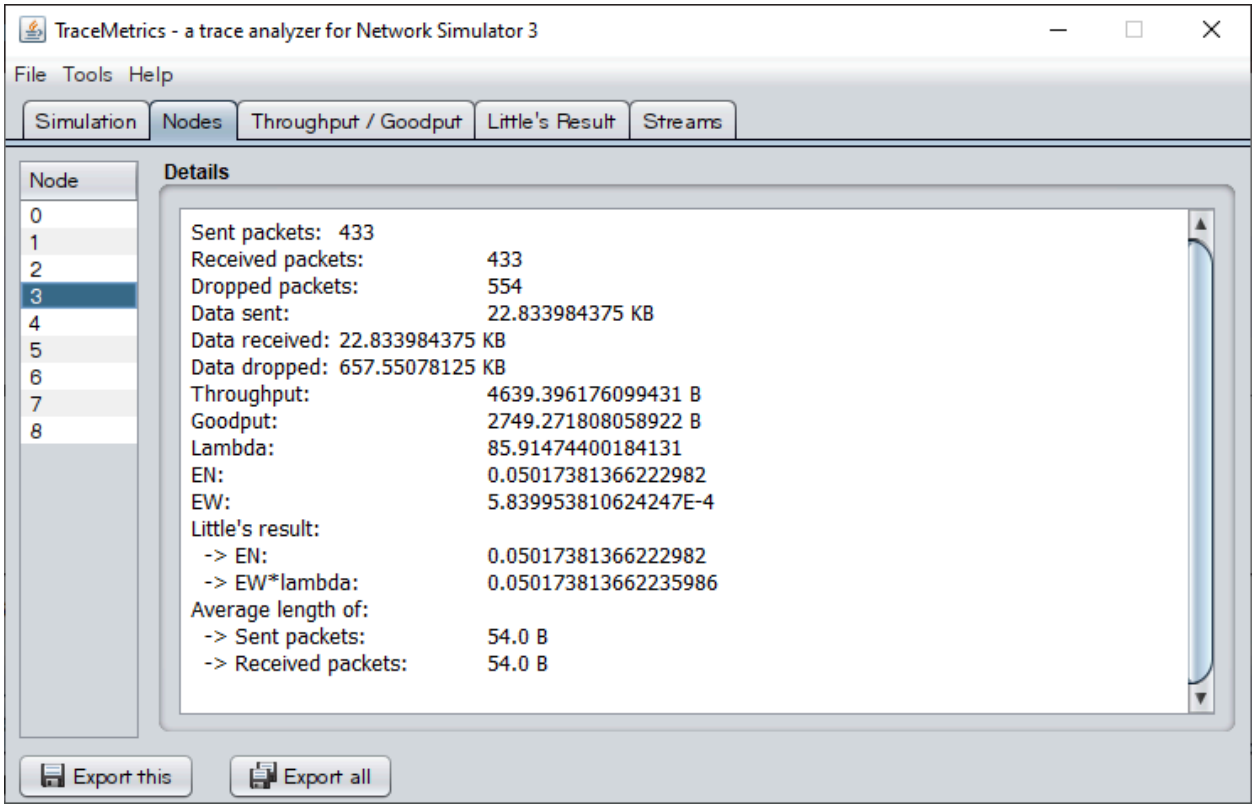
The Blue arrows in this image represent the packages travelling on the network. The Nodes representation is the same as the topology in the beginning for easier understanding of this. The packets are travelling from source to destination nodes (workstations) through the routers.

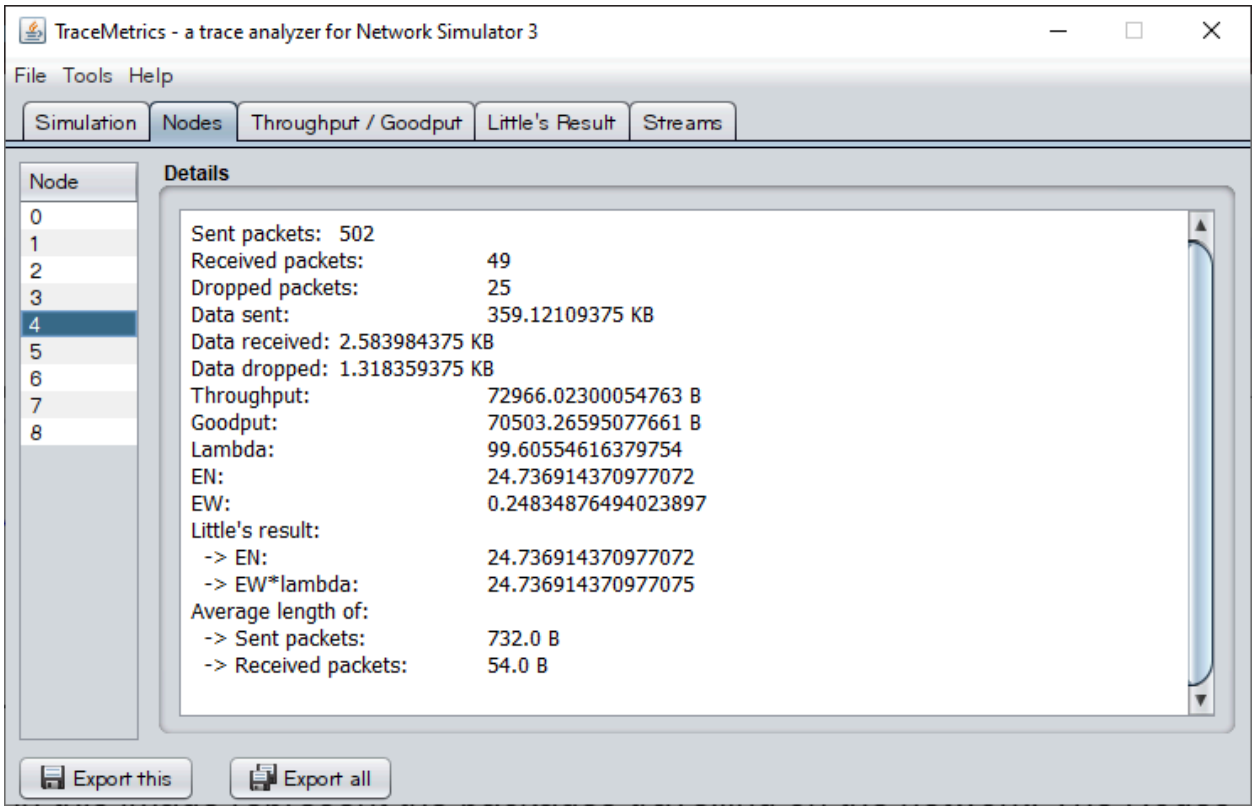
Information On Nodes

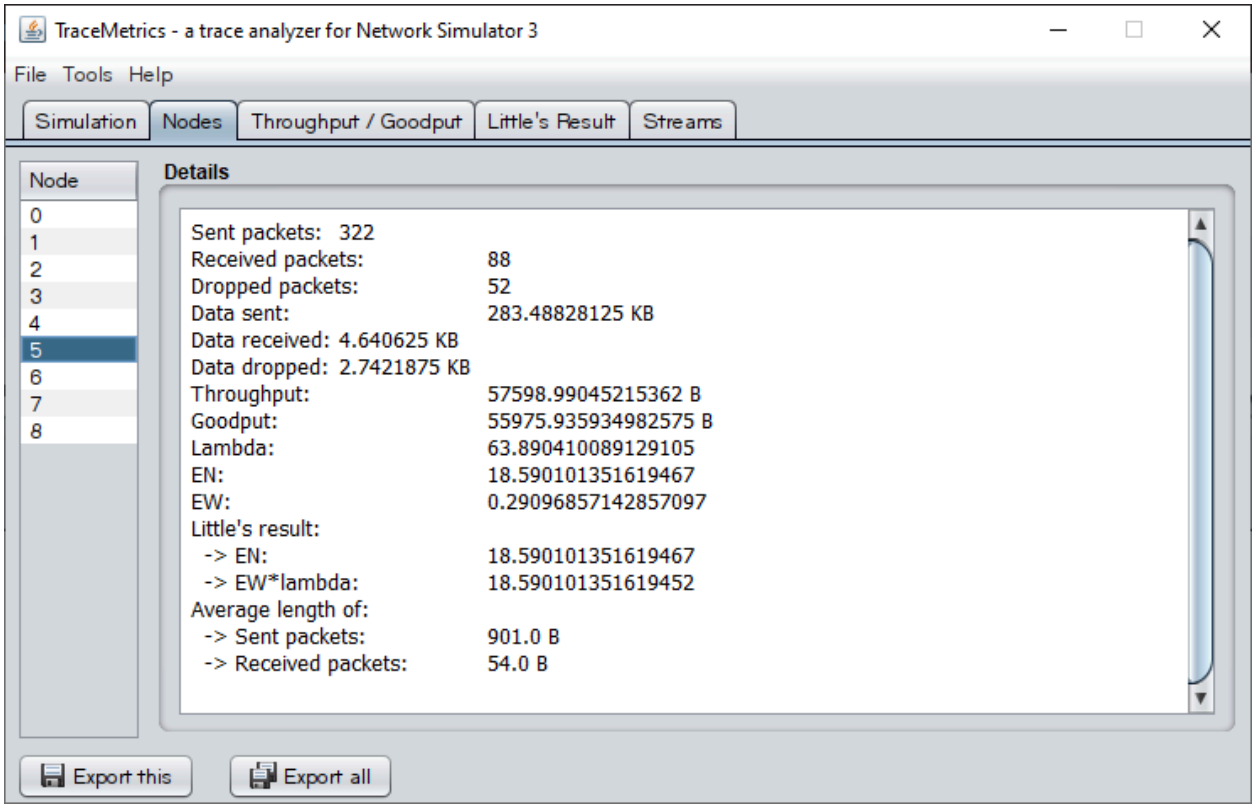


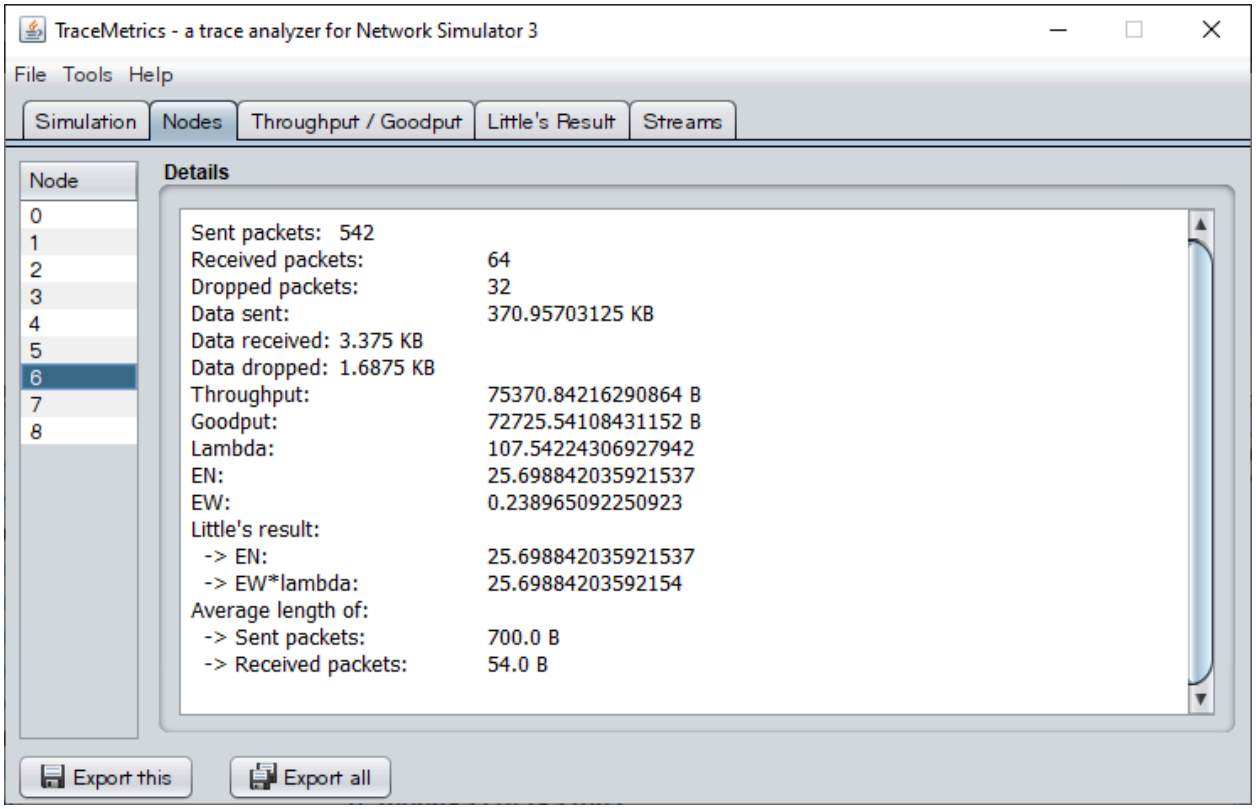

TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0
1
2
3
4
5
6
7
8
Details
Sent packets: 301
Received packets: 301
Dropped packets: 307
Data sent: 15.873046875 KB
Data received: 15.873046875 KB
Data dropped: 348.837890625 KB
Throughput: 3225.0767875425604 B
Goodput: 1911.156614840036 B
Lambda: 59.72364421375112
EN: 0.00831964253117125
EW: 1.3930232558170453E-4
Little's result:
-> EN: 0.00831964253117125
-> EW*lambda: 0.008319642531189842
Average length of:
-> Sent packets: 54.0 B
-> Received packets: 54.0 B
Export this
Export all

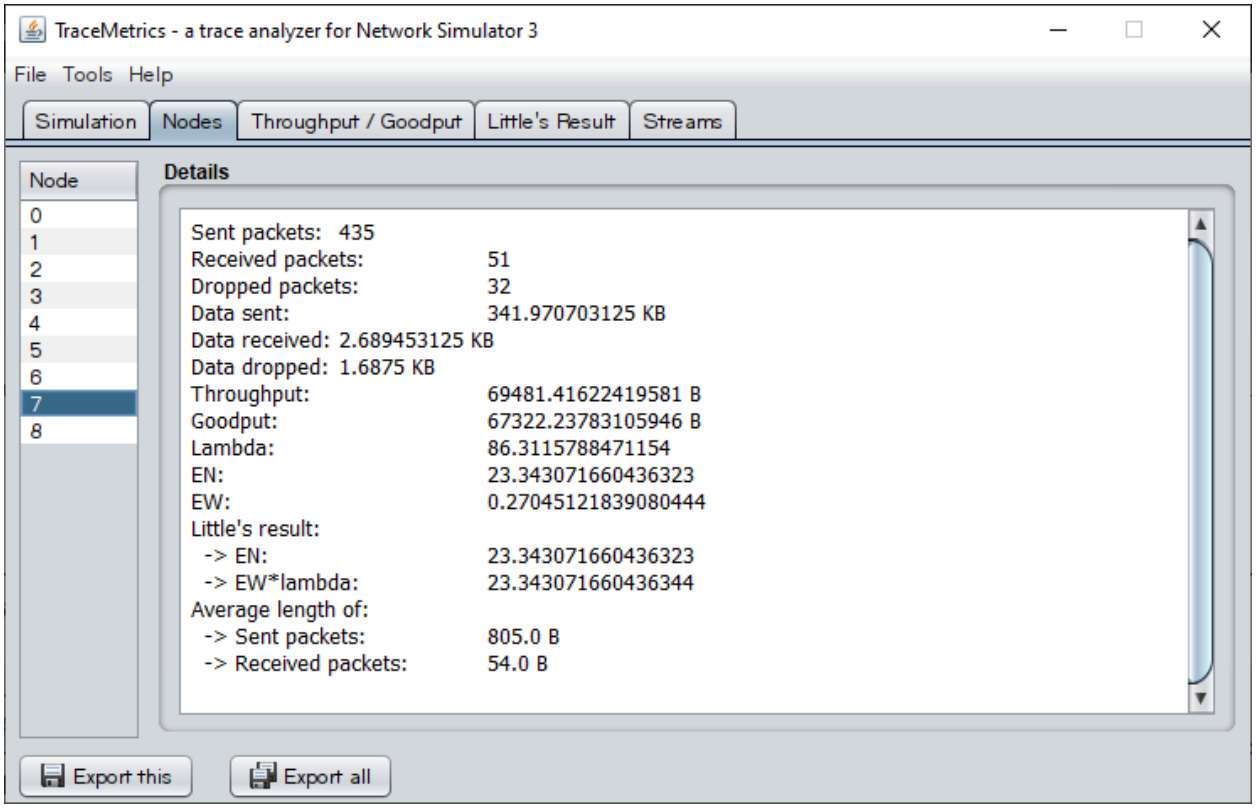

TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0
1
2
3
4
5
6
7
8
Details
Sent packets: 502
Received packets: 502
Dropped packets: 560
Data sent: 26.47265625 KB
Data received: 26.47265625 KB
Data dropped: 638.4140625 KB
Throughput: 5378.699492845068 B
Goodput: 3187.3774772415213 B
Lambda: 99.60554616379754
EN: 0.0057580736049273745
EW: 5.7808764939678276E-5
Little's result:
-> EN: 0.0057580736049273745
-> EW*lambda: 0.005758073604871245
Average length of:
-> Sent packets: 54.0 B
-> Received packets: 54.0 B
Export this
Export all

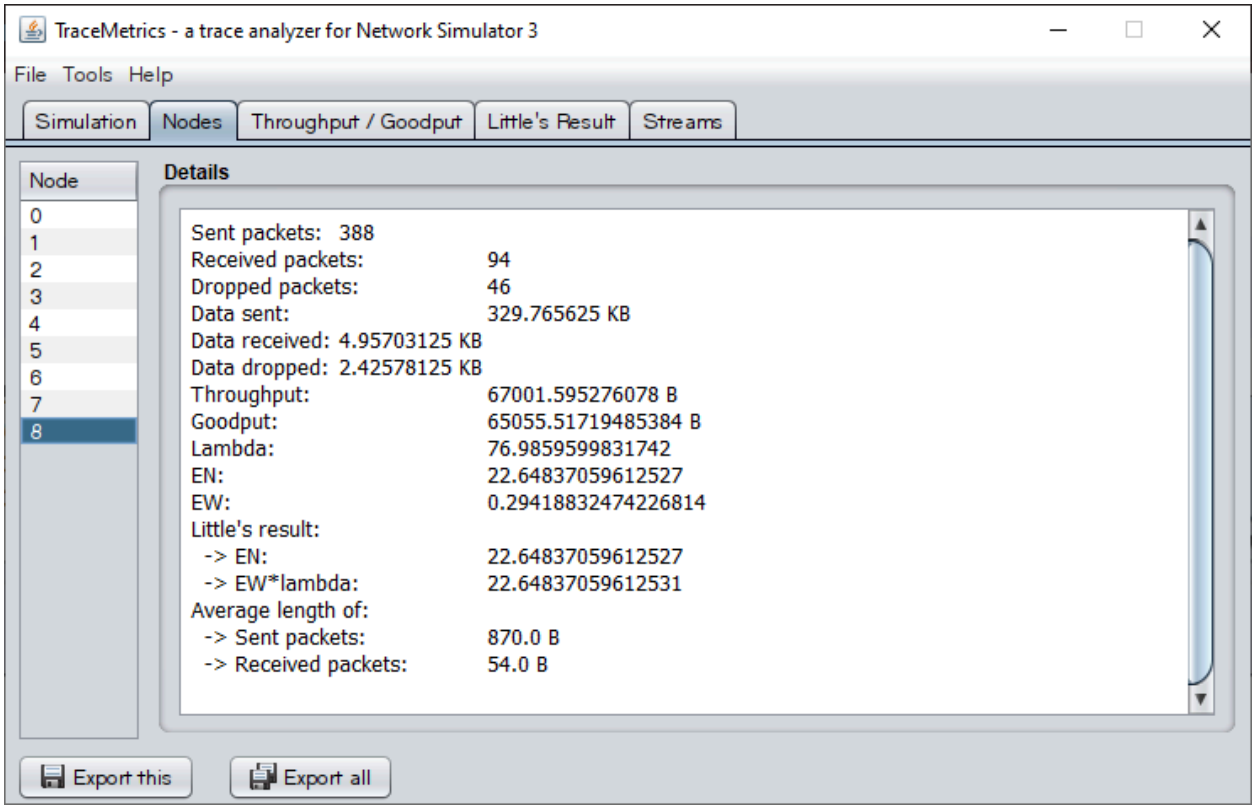

TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0
1
2
3
4
5
6
7
8
Details
Sent packets: 433
Received packets: 433
Dropped packets: 554
Data sent: 22.833984375 KB
Data received: 22.833984375 KB
Data dropped: 657.55078125 KB
Throughput: 4639.396176099431 B
Goodput: 2749.271808058922 B
Lambda: 85.91474400184131
EN: 0.05017381366222982
EW: 5.839953810624247E-4
Little's result:
-> EN: 0.05017381366222982
-> EW*lambda: 0.050173813662235986
Average length of:
-> Sent packets: 54.0 B
-> Received packets: 54.0 B
Export this
Export all


TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0
1
2
3
4
5
6
7
8
Details
Sent packets: 502
Received packets: 49
Dropped packets: 25
Data sent: 359.12109375 KB
Data received: 2.583984375 KB
Data dropped: 1.318359375 KB
Throughput: 72966.02300054763 B
Goodput: 70503.26595077661 B
Lambda: 99.60554616379754
EN: 24.736914370977072
EW: 0.24834876494023897
Little's result:
-> EN: 24.736914370977072
-> EW*lambda: 24.736914370977075
Average length of:
-> Sent packets: 732.0 B
-> Received packets: 54.0 B
Export this
Export all


TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0 1 2 3 4 5 6 7 8
Details
Sent packets: 322
Received packets: 88
Dropped packets: 52
Data sent: 283.48828125 KB
Data received: 4.640625 KB
Data dropped: 2.7421875 KB
Throughput: 57598.99045215362 B
Goodput: 55975.935934982575 B
Lambda: 63.890410089129105
EN: 18.590101351619467
EW: 0.29096857142857097
Little's result:
-> EN: 18.590101351619467
-> EW*lambda: 18.590101351619452
Average length of:
-> Sent packets: 901.0 B
-> Received packets: 54.0 B
Export this
Export all


TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0 1 2 3 4 5 6 7 8
Details
Sent packets: 542
Received packets: 64
Dropped packets: 32
Data sent: 370.95703125 KB
Data received: 3.375 KB
Data dropped: 1.6875 KB
Throughput: 75370.84216290864 B
Goodput: 72725.54108431152 B
Lambda: 107.54224306927942
EN: 25.698842035921537
EW: 0.238965092250923
Little's result:
-> EN: 25.698842035921537
-> EW*lambda: 25.69884203592154
Average length of:
-> Sent packets: 700.0 B
-> Received packets: 54.0 B
Export this
Export all


TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0
1
2
3
4
5
6
7
8
Details
Sent packets: 435
Received packets: 51
Dropped packets: 32
Data sent: 341.970703125 KB
Data received: 2.689453125 KB
Data dropped: 1.6875 KB
Throughput: 69481.41622419581 B
Goodput: 67322.23783105946 B
Lambda: 86.3115788471154
EN: 23.343071660436323
EW: 0.27045121839080444
Little's result:
-> EN: 23.343071660436323
-> EW*lambda: 23.343071660436344
Average length of:
-> Sent packets: 805.0 B
-> Received packets: 54.0 B
Export this
Export all


TraceMetrics - a trace analyzer for Network Simulator 3
File Tools Help
Simulation Nodes Throughput / Goodput Little's Result Streams
Node
0
1
2
3
4
5
6
7
8
Details
Sent packets: 388
Received packets: 94
Dropped packets: 46
Data sent: 329.765625 KB
Data received: 4.95703125 KB
Data dropped: 2.42578125 KB
Throughput: 67001.595276078 B
Goodput: 65055.51719485384 B
Lambda: 76.9859599831742
EN: 22.64837059612527
EW: 0.29418832474226814
Little's result:
-> EN: 22.64837059612527
-> EW*lambda: 22.64837059612531
Average length of:
-> Sent packets: 870.0 B
-> Received packets: 54.0 B
Export this
Export all