School of
Applied Computing

Network Simulation

NetSim

Exercise 2

Faculty of Architecture, Computing and Engineering

School of Applied Computing

Mount Pleasant Campus

Swansea

SA1 6ED

2. Study the throughput of Slow start + congestion avoidance (Old Tahoe) and Fast Retransmission (Tahoe) congestion control Algorithms.

2.1 Theory:

One of the important functions of a TCP Protocol is congestion control in the network. Given below is a description of how Old Tahoe and Tahoe variants (of TCP) control congestion.

Old Tahoe:

Congestion can occur when data arrives on a big pipe (i.e. a fast LAN) and gets sent out through a smaller pipe (i.e. a slower WAN). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets.

The assumption of the algorithm is that the packet loss caused by damaged is very small (much less than 1%), therefore the loss of a packet signals congestion somewhere in the network between the source and destination. There are two indications of packets loss: a timeout occurring and the receipt of duplicate ACKs

Congestion avoidance and slow start are independent algorithms with different objectives. But when congestion occurs TCP must slow down its transmission rate and then invoke slow start to get things going again. In practice they are implemented together.

Congestion avoidance and slow start requires two variables to be maintained for each connection: a Congestion Window (i.e. cwnd) and a Slow Start Threshold Size (i.e. ssthresh). Old Tahoe algorithm is the combination of slow start and congestion avoidance. The combined algorithm operates as follows,

1. Initialization for a given connection sets cwnd to one segment and ssthresh to 65535 bytes.
2. When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one-half of the current window size (the minimum of cwnd and the receiver's advertised window, but at least two segments) is saved in ssthresh. Additionally, if the congestion is indicated by a timeout, cwnd is set to one segment (i.e. slow start).
3. When new data is acknowledged by the other end, increase cwnd, but the way it increases depends on whether TCP is performing slow start or congestion avoidance.

If cwnd is less than or equal to ssthresh, TCP is in slow start. Else TCP is performing congestion avoidance. Slow start continues until TCP is halfway to where it was when congestion occurred (since it recorded half of the window size that caused the problem in step 2). Then congestion avoidance takes over.

Slow start has cwnd begins at one segment and be incremented by one segment every time an ACK is received. As mentioned earlier, this opens the window exponentially: send one segment, then two, then four, and so on. Congestion avoidance dictates that cwnd be incremented by 1/cwnd, compared to slow start's exponential growth. The increase in cwnd should be at most one segment in each round trip time (regardless of how many ACKs are received in that RTT), whereas slow start increments cwnd by the number of ACKs received in a round-trip time.

Tahoe (Fast Retransmit):

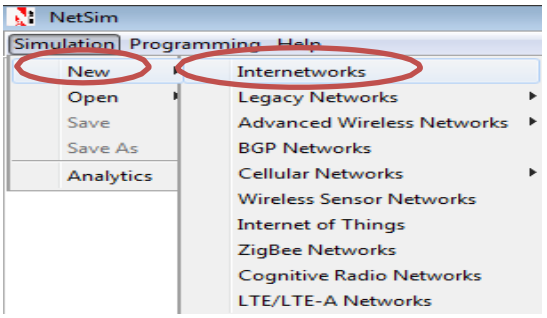
The Fast retransmit algorithms operating with Old Tahoe is known as the Tahoe variant.

TCP may generate an immediate acknowledgement (a duplicate ACK) when an out-of-order segment is received out-of-order, and to tell it what sequence number is expected.

Since TCP does not know whether a duplicate ACK is caused by a lost segment or just a re-ordering of segments, it waits for a small number of duplicate ACKs to be received. It is assumed that if there is just a reordering of the segments, there will be only one or two duplicate ACKs before the re-ordered segment is processed, which will then generate a new ACK. If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then performs a retransmission of what appears to be the missing segment, without waiting for a re-transmission timer to expire.

2.2 Procedure:

Go to Simulation → New → Internetworks



Sample Inputs:

Follow the steps given in the different samples to arrive at the objective.

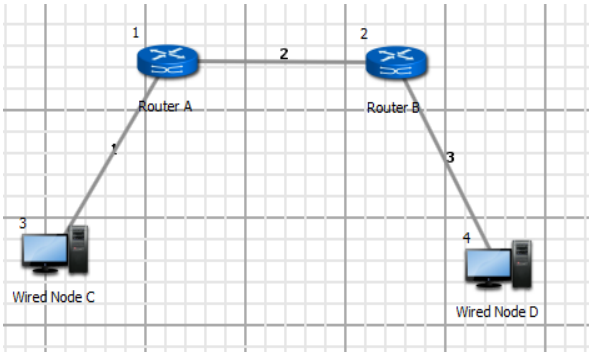
Sample 1.a: Old Tahoe (1 client and 1 server)

In this Sample,

- Total no of Node used: 2
- Total no of Routers used: 2

The devices are inter connected as given below,

- Wired Node C is connected with Router A by Link 1.
- Router A and Router B are connected by Link 2.
- Wired Node D is connected with Router B by Link 3.



Set the properties for each device by following the tables,

Application Properties	
Application Type	Custom
Source_Id	4(Wired Node D)
Destination_Id	3(Wired Node C)

Packet Size	
Distribution	Constant
Value (bytes)	1460
Inter Arrival Time	
Distribution	Constant
Value (micro secs)	1300

Node Properties: In Transport Layer properties, set

TCP Properties	
MSS(bytes)	1460
Congestion Control Algorithm	**Old Tahoe**
Window size(MSS)	8

Router Properties: Accept default properties for Router.

Link Properties	**Link 1**	**Link 2**	**Link 3**
Max Uplink Speed (Mbps)	8	10	8
Max Downlink Speed(Mbps)	8	10	8
Uplink BER	0.000001	0.000001	0.000001
Downlink BER	0.000001	0.000001	0.000001

Simulation Time - 10 Sec

Upon completion of simulation, ‒Save‖ the experiment.

(Note: *The Simulation Time can be selected only after doing the following two tasks,*

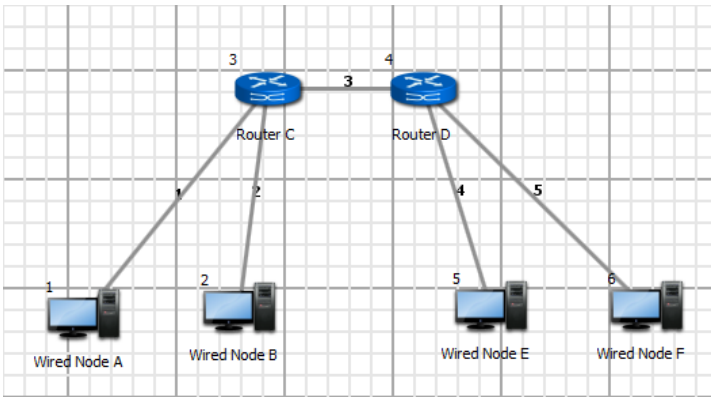
- *Set the properties of Node, Router& Link*
- *Then click on Run Simulation button)***.**

Sample 1.b: Tahoe (1 client and 1 server)

Open sample 1.a, and change the TCP congestion control algorithm to Tahoe (in Node Properties). Upon completion of simulation, ‒Save‖ the experiment as sample 1.b.

Sample 2.a: Old Tahoe (2 clients and 2 servers)

In this Sample,



- Total no of Wired Nodes used: 4
- Total no of Routers used: 2

The devices are inter connected as given below,

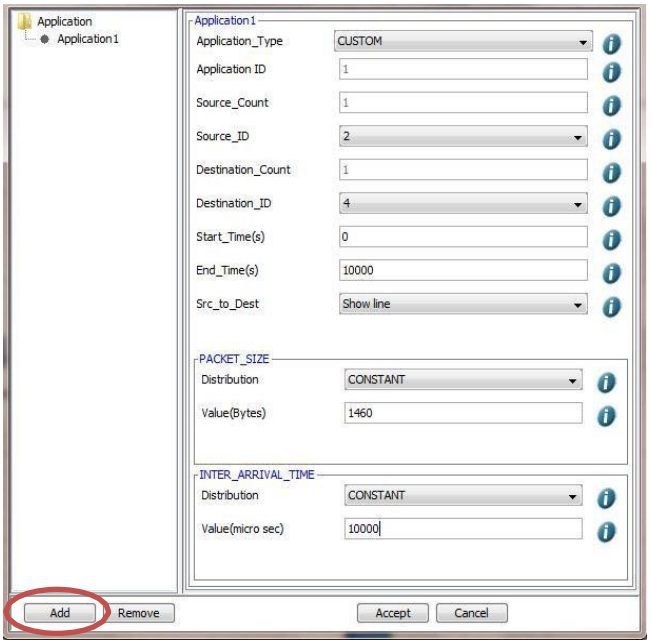
- Wired Node A and Wired Node B are connected with Router C by Link 1 and Link 2.
- Router C and Router D are connected by Link 3.
- Wired Node E and Wired Node F are connected with Router D by Link 4 and Link 5.
- Wired Node A and Wired Node B are not transmitting data in this sample.

Set the properties for each device by following the tables,

Application Properties	**Application 1**	**Application 2**
Application Type	Custom	
Source_Id	5	6
Destination_Id	1	2
Packet Size		
Distribution	Constant	Constant
Value (bytes)	1460	1460
Inter Arrival Time		
Distribution	Constant	Constant
Value (micro secs)	1300	1300

NOTE: The procedure to create multiple applications are as follows:

Step 1: Click on the ADD button present in the bottom left corner to add a new application.



Node Properties: In Transport Layer properties, set

TCP Properties	
MSS(bytes)	1460
Congestion Control Algorithm	**Old Tahoe**
Window size(MSS)	8

Router Properties: Accept default properties for Router.

Link Properties	**Link 1**	**Link 2**	**Link 3**	**Link 4**	**Link 5**
Max Uplink Speed (Mbps)	8	8	10	8	8
Max Downlink Speed (Mbps)	8	8	10	8	8

Uplink BER	0.000001	0.000001	0.000001	0.000001	0.000001
Downlink BER	0.000001	0.000001	0.000001	0.000001	0.000001

Simulation Time - 10 Sec

Upon completion of simulation, ‒Save‖ the experiment.

(Note: *The Simulation Time can be selected only after doing the following two tasks,*

- *Set the properties of Node , Router & Link*
- *Then click on Run Simulation button)***.**

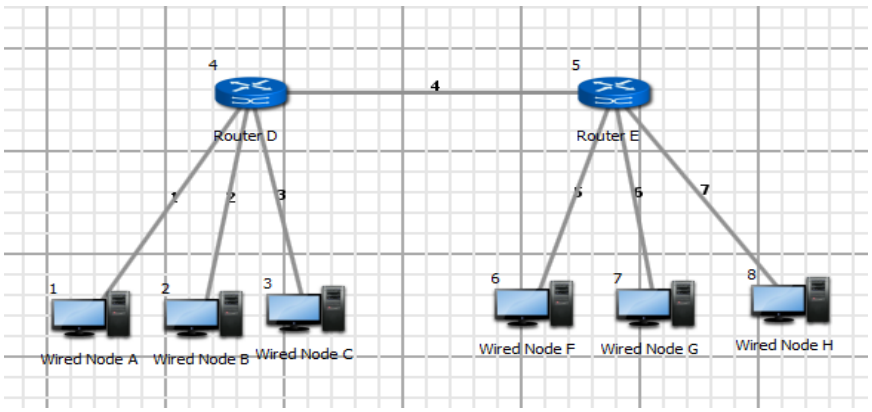
Sample 2.b: Tahoe (2 clients and 2 servers)

Do the experiment as sample 2.a, and change the congestion control algorithm to Tahoe.

Upon completion of simulation, ‒Save‖ the experiment.

Sample 3.a: Old Tahoe (3 clients and 3 servers)

In this Sample,



- Total no of Nodes used: 6
- Total no of Routers used: 2

The devices are inter connected as given below,

- Wired Node A, Wired Node B & Wired Node C is connected with Router D by Link 1, Link 2 & Link 3.
- Router D and Router E are connected by Link 4.
- Wired Node F, Wired Node G & Wired Node H is connected with Router E by Link 5, Link 6 & Link 7.
- Wired Node A, Wired Node B and Wired Node C are not transmitting data in this sample.

Set the properties for each device by following the tables,

Application Properties	**Application 1**	**Application 2**	**Application 3**
Application Type	Custom		
Source_Id	6	7	8
Destination_Id	1	2	3
Packet Size			
Distribution	Constant	Constant	Constant
Value (bytes)	1460	1460	1460
Inter Arrival Time			
Distribution	Constant	Constant	Constant
Value (micro sec)	1300	1300	1300

Node Properties: In Transport Layer properties, set

TCP Properties			
MSS(bytes)	1460	1460	1460
Congestion Control Algorithm	**Old Tahoe**	**Old Tahoe**	**Old Tahoe**
Window size(MSS)	8	8	8

Router Properties: Accept default properties for Router.

Link Properties	Link 1	Link 2	Link 3	Link 4	Link 5	Link 6	Link 7
Max Uplink Speed (Mbps)	8	8	8	10	8	8	8
Max Downlink Speed(Mbps)	8	8	8	10	8	8	8
Uplink BER	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
Downlink BER	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001

Simulation Time- 10 Sec

Upon completion of simulation, ‒Save‖ the experiment.

(Note: *The Simulation Time can be selected only after doing the following two tasks,*

- *Set the properties of Node, Router & Link*
- *Then click on Run Simulation button)***.**

Sample 3.b: Tahoe (3 clients and 3 servers)

Do the experiment as sample 3.a, and change the TCP congestion algorithm to Tahoe. Upon completion of simulation, ‒Save‖ the experiment.

2.3 Output

Compare your output with the table is below after running the simulation. Graphically compare and analyse the metrics.

Comparison Table:

TCP Downloads	Metrics	Slow start + Congestion avoidance	Fast Retransmit
1 client and 1 server	Throughput(Mbps)	5.926432	6.120320
	Segments Retransmitted + Seg Fast Retransmitted	195	200
	Throughput(Mbps)	8.796208	8.810224

2 clients and 2 servers	Segments Retransmitted + Seg Fast Retransmitted	343	378
3 clients and 3 servers	Throughput(Mbps)	9.144272	9.23304
	Segments Retransmitted + Seg Fast Retransmitted	401	434

Note: To calculate the ‒Throughput (Mbps)‖ for more than one application, add the individual application throughput which is available in Application Metrics (or Metrics.txt) of Performance Metrics screen. In the same way calculate the metrics for ‒Segments Retransmitted + Seg Fast Retransmitted‖ from TCP Metrics → Connection Metrics.

2.4 Inference:

User lever throughput: User lever throughput of Fast Retransmit is higher when compared then the Old Tahoe (SS + CA). This is because, if a segment is lost due to error, Old Tahoe waits until the RTO Timer expires to retransmit the lost segment, whereas Tahoe (FR) retransmits the lost segment immediately after getting three continuous duplicate ACK's. This results in the increased segment transmissions, and therefore throughput is higher in the case of Tahoe.