**Objective:**

To simulate end to end data delivery in a virtual network and to observe the performance of the network for various network parameters. Through this, we gained an insight into how various parameters affect the network, how to optimize the performance of the network and the tradeoffs involved in optimizing performance.

**Simulation Approach:**

**Programming Language and Framework:**

We used HTML 5, CSS 3 and Bootstrap for UI and Javascript for the program.

HTML has very good UI facilities. It is possible to create attractive menus, controls, tables and graphs.

Further, the application is portable as it can be hosted on the Internet. It can be accessed from anywhere in the world.

Also, Javascript and HTML are supported on all operating systems. Our application will also run on smart phones with browsers.

**Implementation:**

We used discrete event simulation to simulate various events like “Attempt to transmit a packet”, “Update link costs”, “Stop Transmission”, “Process packet at router”, etc.

We implemented CSMA CD algorithm with exponential backoff for collision detection and Dijikstra’s algorithm for routing.

**Classes:**

***Packet and Packet Generator***

**Packet**

Source

Destination

RxTime

TxTime

BirthTime

QueueDelays

Path Followed

Priority

**Packet Generator**

Generate()

The “PacketGenerator” generates a list of packets with exponential inter-arrival times such that there are enough packets to last the simulation duration. Each end node uses this class to generate its packets.

The Packet contains -source, destination, timing information like birth time, transmitted time, etc. , delay information like delay experienced in the queues of routers and nodes, transmission delay, propagation delay, etc. , the path followed by the packet and priority of the packet. The priority of the packet decides the priority given to it by the router. Higher priority packets are processed first at both the input and output queues of the router.

**Core Algorithms and Related Data Structures:**

**DijikstrasAlgo**

**Run()**

GetNextRouterWithMinumumCost()

CompareUpdateCosts()

**CSMA CD**

**AttemptToTransmit()**

IsBusBusy()

ExponentialBackoff()

\*PacketAttemptCallback()

\*PacketSentCallback()

\*BusFreeCallback()

*\*Used to inform node/router of events*

**Topology**

A 2 Dimensional Array representing the connectivity of routers with each other.

*See below for details.*

**Forwarding Table**

Array of “Forwarding Entry”.

**Forwarding Entry**

Destination

Next Hop

Total Cost

Predecessor

CSMA CDinteracts with the bus and delivers packets. It listens to the bus and keep tracks of events like “Attempting to transmit”, “Transmission started by other nodes/routers”, “Transmission stopped by other nodes/routers”, “Collision occurred”, “Packet Sent Successfully”, etc.

Each node/router contains an object of this class. CSMA CD informs the node/router when a transmission has succeeded and when the bus is free, so that the node/router can send the next packet to the CSMA CD object.

The DijikstrasAlgo runs on the Topology and returns a ForwardingTable. Each router has an object of this class.

The Topology is a 2 Dimensional Array that represents the connectivity information. The row number indicates the source router and the column number represents the destination router. The value at that specific row and column number is the cost of the link , which in Infinite if the 2 routers are not connected. E.g. Topology[i][j] represents the cost of the ith router to the jth‑router.

**Core Entities:**

***Node, Router and Bus***

**Bus**

RouterIds

NodeIds

StartTransmitting()

StopTransmitting()

**Router**

RouterId

Name

Forwarding Table

CSMACD

DijikstrasAlgo

ProcessPacket()

ForwardToEndNode()

ForwardToRouter()

**Node**

NodeId

Name

Array of Packets

CSMACD

\*OnBusFree()

\*OnPacketSent()

\*\*OnMessageFromBus()

*\*Called by CSMA CD*

*\* Called by Bus*

The “Node” uses the “Packet Generator” to create a list of packets, and sends packets one by one to “CSMA CD” for delivery on the bus. Each node has a default router. This default router forwards packets that are meant for nodes on the other bus.

The “Router” uses “DijikstrasAlgo” to create its forwarding table. It receives packets from the “Bus” and from other “Router”. The router has 3 input queues and 3 output queues for different priorities of packets. The router checks the priority of the packet and places it in the appropriate queue. The router processes packets from higher priority queues first. It has a configurable processing delay. A packet may be forwarded to another router on the backhaul or delivered to end nodes on the bus using “CSMA CD”.

The “Bus” contains an array of nodes and routers connected to it. It is used by the nodes/routers through the “CSMACD” object to transmit packets. The bus forwards packets to the nodes/routers connected to it after their corresponding propagation delays.

**Other Entities:**

***Settings, LinkCostUpdater, ResultsDisplayer:***

******

**ResultsDisplayer**

DisplaySummary()

DisplayCollisions()

DisplayDelays()

DisplayPathTrace()

**LinkCostUpdater**

UpdateCosts()

InformRouters()

“Settings” contains various network parameters that are editable by the user via UI. These are used by various entities in their operations – e.g. the bus uses the InterNode distance to calculate the propagation delay.

“LinkCostUpdater” updates the costs after the “Link Update Interval” which is configurable ( 2ms by default) and informs all routers. The routers then use “DijikstrasAlgo” to update their forwarding table.

“ResultsDisplayer” has various functions for displaying results.

**Results and Observations:**

**Network Parameters:**

Frame Slot Duration = 0.5 milli sec = 500 micro sec

Inter Node Distance = 2000 meters

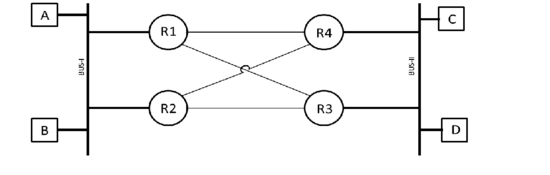
Transmission Rate = 100 Mbps

Packet Size = 1000 bytes

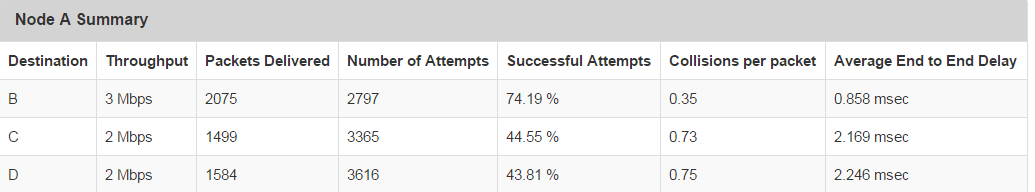
Router Processing Delay = 0.1 milli sec = 100 micro sec

Packet Inter Arrival Time = 2 frame slots

Topology :



**Node A Summary:**



**Observation:** The throughput from A to B is 3 Mbps and throughputs from A to C and A to D are 2 Mbps each.

**Comment:** This is expected, as packets travelling from A to C and A to D will face collisions at 2 buses, will face the processing delays of 2 routers and will travel more distance. So, less packets will be delivered from A to C and A to D , contributing to a lower throughput.

**Observation:** The percentage of successful attempts from A to B is approximately 75% and that from A to C and A to D are approximately 44%.

**Comment:** This is expected, as packets travelling from A to C and A to D will face more collisions as they go through 2 buses. So, they will need more retransmissions and more attempts.

**Observation:** The average end to end delay from A to B is 0.86 milli sec and that from A to C and A to D is approximately 2.2 milliseconds.

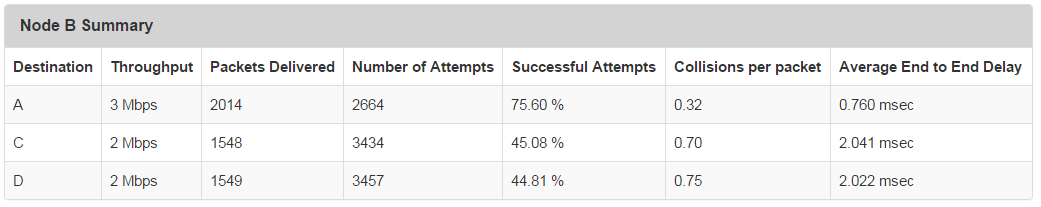
**Comment:** This is expected, as packets travelling from A to C and A to D will face more delays as they pass through routers and face collisions at 2 buses.

**Observation:** Packets from A to B face an average of 0.35 collisions per packet and packets from A to C and A to D face an average of 0.74 collisions per packet.

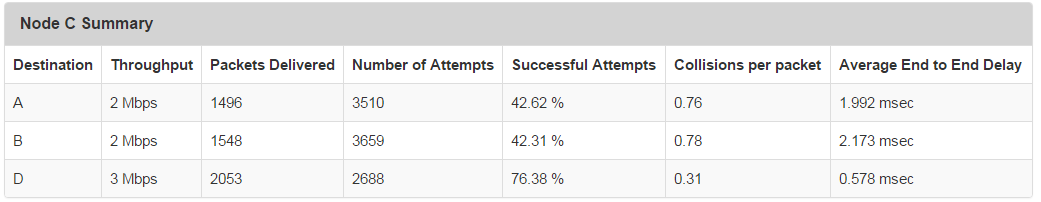
**Comment:** This is expected, as packets travelling from A to C and A to D will face collisions at 2 buses.

The summaries for packets sent by B, C and from D are similar:

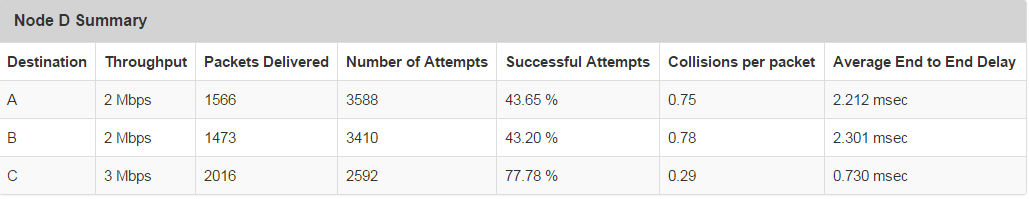
**Node B Summary:**

****

**Node C Summary :**

****

**Node D Summary :**

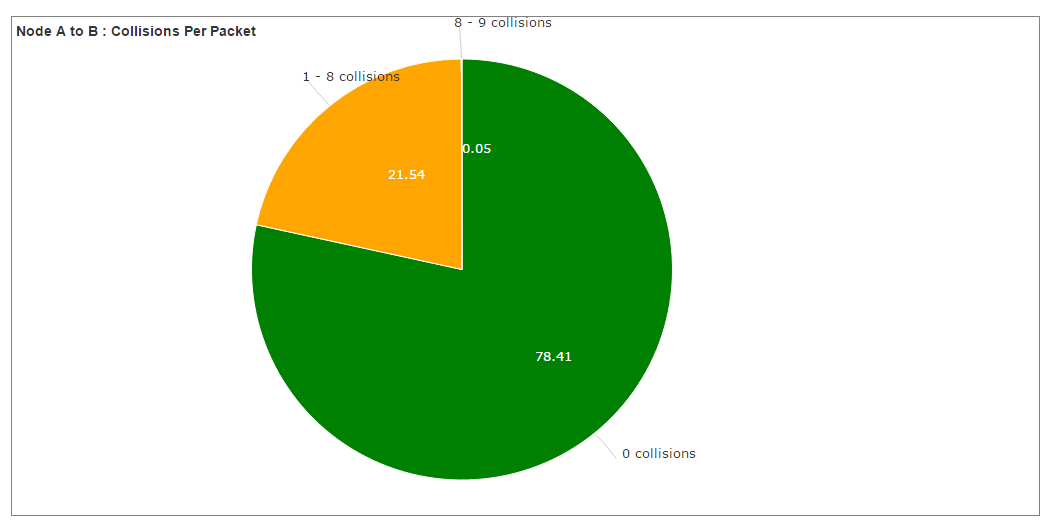
****

**Collisions Per Packet Details:**

**Packets sent from A to B Observations:**

Around 78% of the packets ( Green Section) experience 0 collisions.

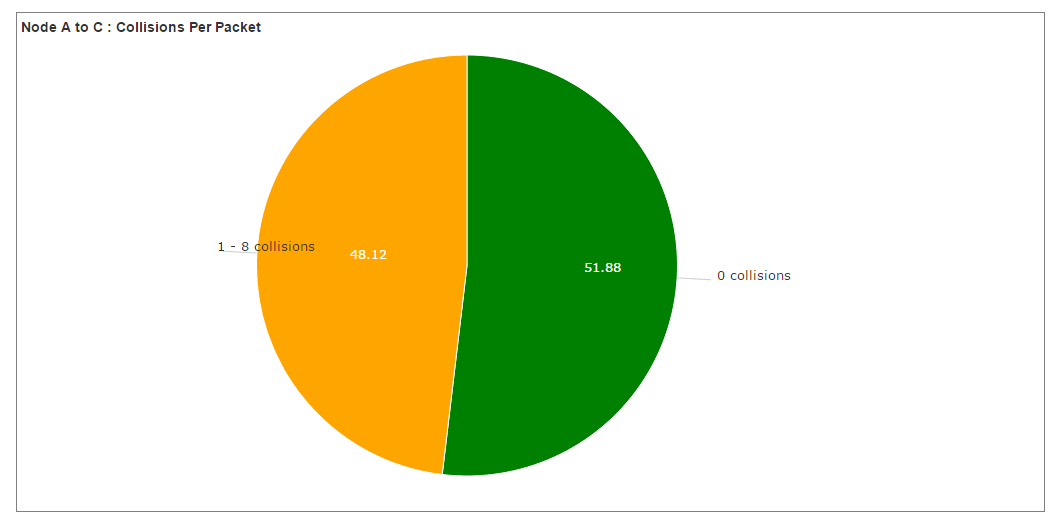
Around 21.5% of the packets ( Yellow Section )experience 1 to 8 collisions.

****

**Packets sent from A to C:**

Around 50% of the packets (Green Section) experience 0 collisions.

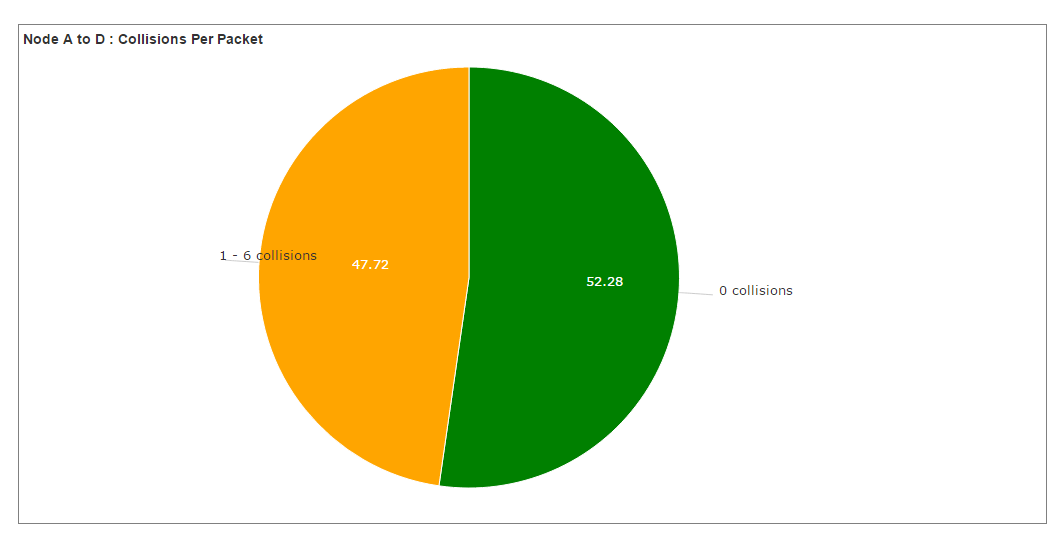
Around 48% of the packets ( Yellow Section )experience 1 to 8 collisions.

****

**A to D:**

Around 52% of the packets (Green Section) experience 0 collisions.

Around 48% of the packets ( Yellow Section )experience 1 to 8 collisions.

****

**Comments:**

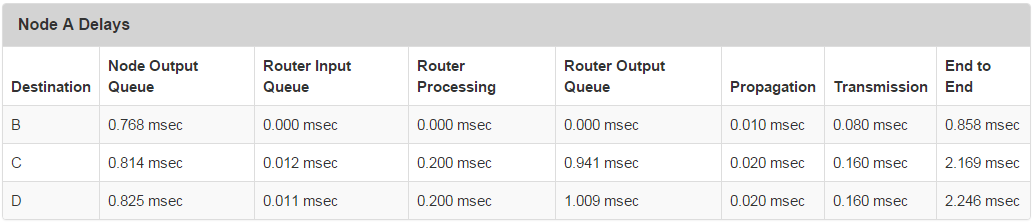
A smaller percentage of packets sent from A to C and A to D experience 0 collisions and a higher percentage of packets experience between 1 to 8 collisions compared to packets sent from A to B.

This is because such packets may potentially face collision at 2 buses.

**Packet Delay Details:**

This presents a breakup of the delays faced by a packet as it travels from source node to destination node. The end to end delay is the sum of all these delays.

**Node A:**

****

**Observation:**

Packets from A to B face around 0.77 millisec delay at the output queue of the source node, which is similar to the delay faced by packets from A to C ( 0.814 millisec) and from A to D (0.825 millisec).

**Comment:**

This delay is the same for all packets as each packet has to wait at the output queue of the source node for the bus to become free.

**Observation:**

Packets from A to C and from A to D face a delay of around 0.01 millisec at the input queue of routers. Packets from A to B don’t face this delay.

**Comment:**

This is because packets from A to B are sent directly via the bus and they do not go through any router.

**Observation:**

Packets from A to C and from A to D face an overall processing delay of 0.2 millisec inside routers.

Packets from A to B don’t face this delay

**Comment:**

This is because packets from A to C and D go through 2 routers and the processing delay at each router has been set to 0.1 millisec, making the overall processing delay = 2 \* 0.1 millisec = 0.2 millisec.

**Observation:**

Packets from A to C and from A to D face an output queue delay of around 1 millisec at routers.

Packets from A to B don’t face this delay

**Comment:**

This is because packets from A to C and D have to wait at the output queue of a router, till the bus of the destination node becomes free.

**Observation:**

Packets from A to C and from A to D face a higher overall propagation delay of 0.020 millisec compared to packets from A to B, which face a delay of 0.010 millisec.

**Comment:**

This is because packets from A to C and D have to travel double the distance on the bus – 2000 meters from source node to the router and then again 2000 meters from the router to the destination on the other bus. It is assumed that propagation delay on the backhaul connecting 2 routers is 0.

**Observation:**

Packets from A to C and from A to D face a higher overall transmission delay of 0.160 millisec compared to packets from A to B, which face a delay of 0.080 millisec.

**Comment:**

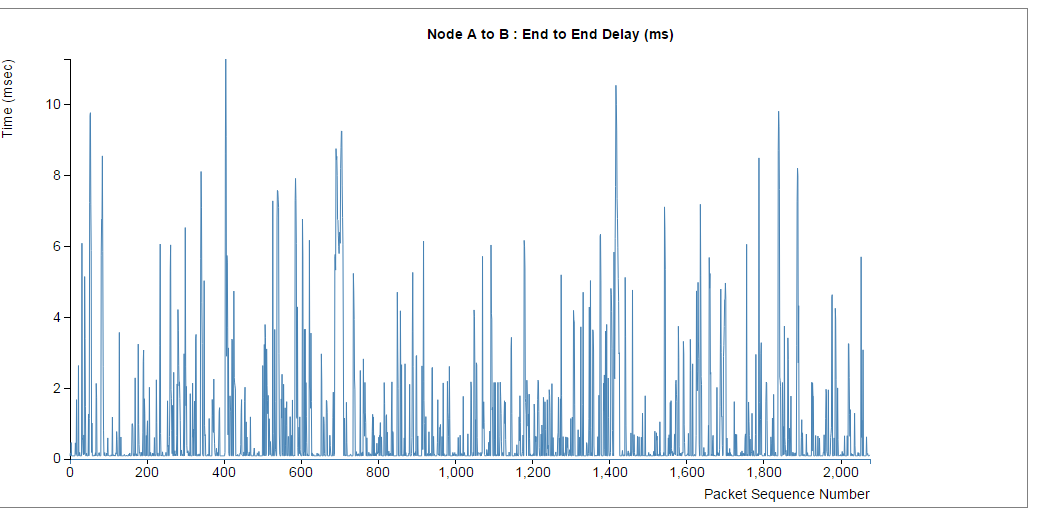
This is because packets from A to C and A to D are first transmitted from source node to the bus at 100 Mbps and then from the router to the destination bus also at 100 Mbps. So, they face 2 transmission delays. It is assumed that transmission delay on the backhaul is 0.

The results for packets sent by other nodes are similar.

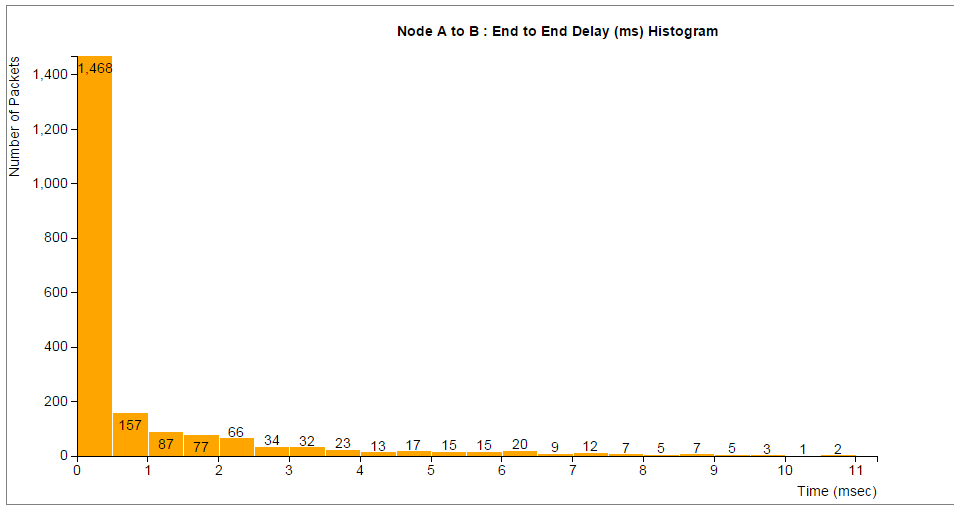
**End to End Delay Graphs:**

This show the end to end delay faced by each packet :

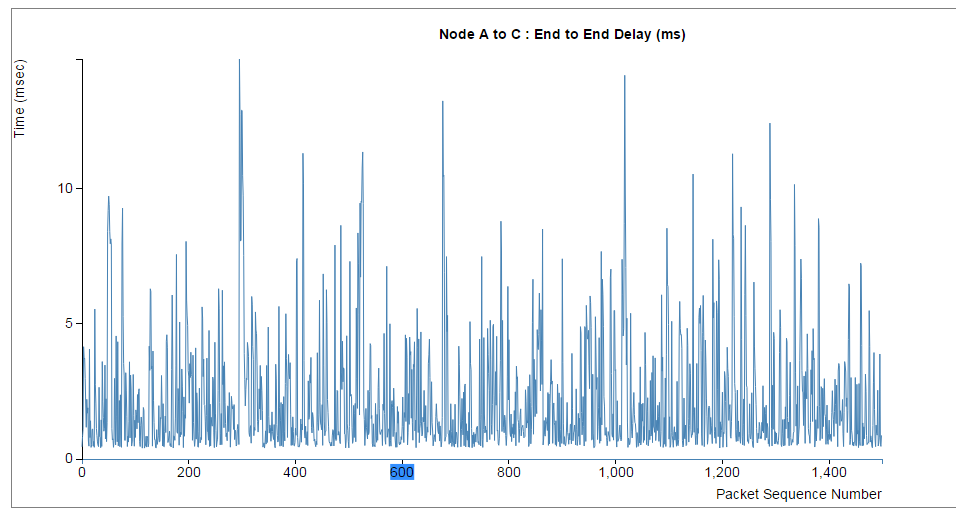
**Node A to B :**

****

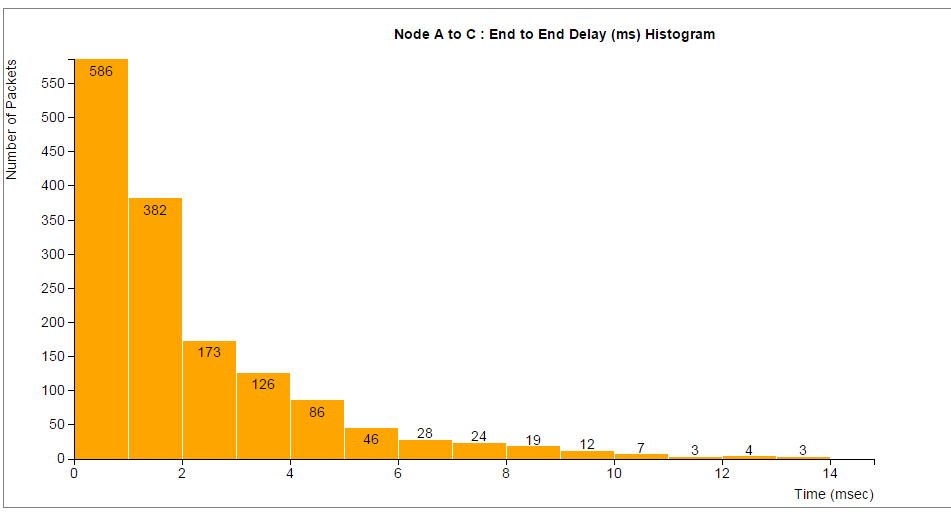
**Histogram:**

****

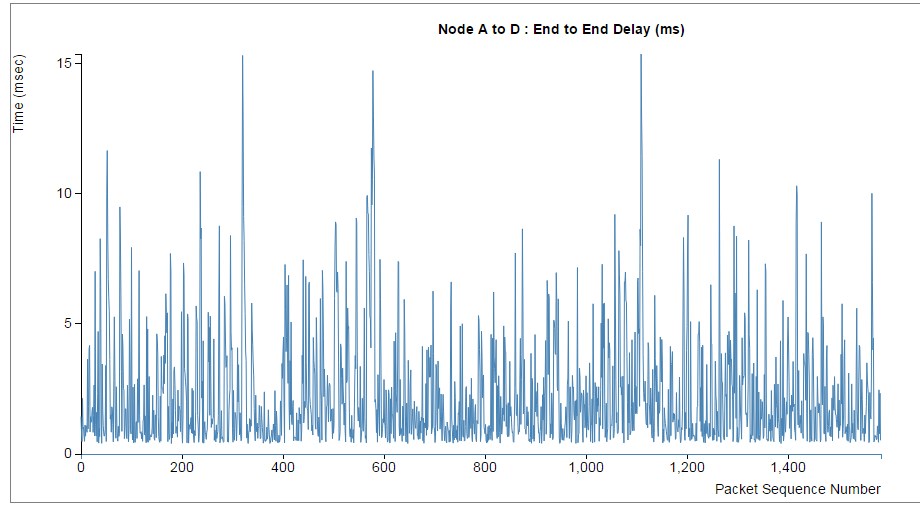
**Node A to C End to End Delay:**

****

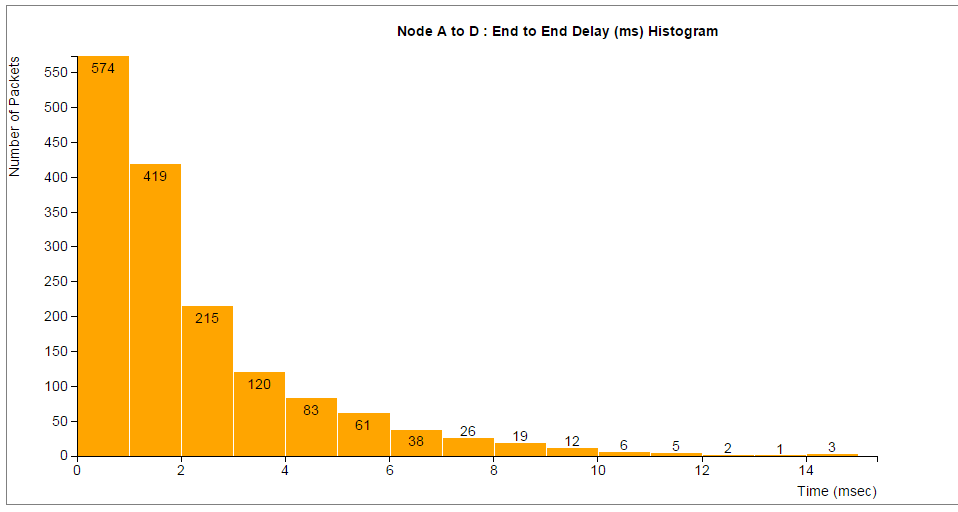
**Histogram:**

****

**Node A to D End to End Delay:**

****

**Histogram:**

****

**Observations:**

The end to end delay values for A to B vary from 0 to 10 millisec with majority of the spikes below 5 millisec. The end to end delay values for A to C and D also vary from 0 to 10 millisec with majority of the spikes above 5 millisec.

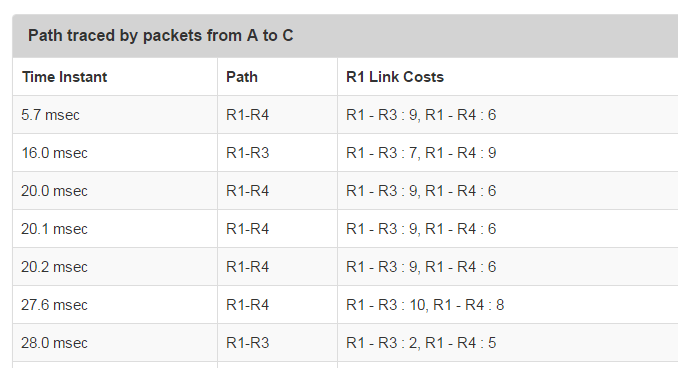
The histogram for A to B shows that maximum packets experience an end to end delay in between 0 to 1 millisec and very few experience larger delays. The histogram for packets from A to C and from A to D shows a comparatively more flatter distribution. i.e. more packets experience delays greater than 1 millisec than packets that experience a delay less than 1 millisec

**Comments:**

A higher end to end delay from A to C and from A to D is expected as explained in the Packet Delays section.

Results for packets sent by Node B, Node C and Node D are similar.

**Path Followed:**

****

**Observations:**

We see that the path followed changes with time as link costs are updated every 2 milliseconds.

For e.g. at t = 27.6 millisec, the cost from R1-R3 is 10 and R1-R4 is 8, so packets follow the path through R1-R4.

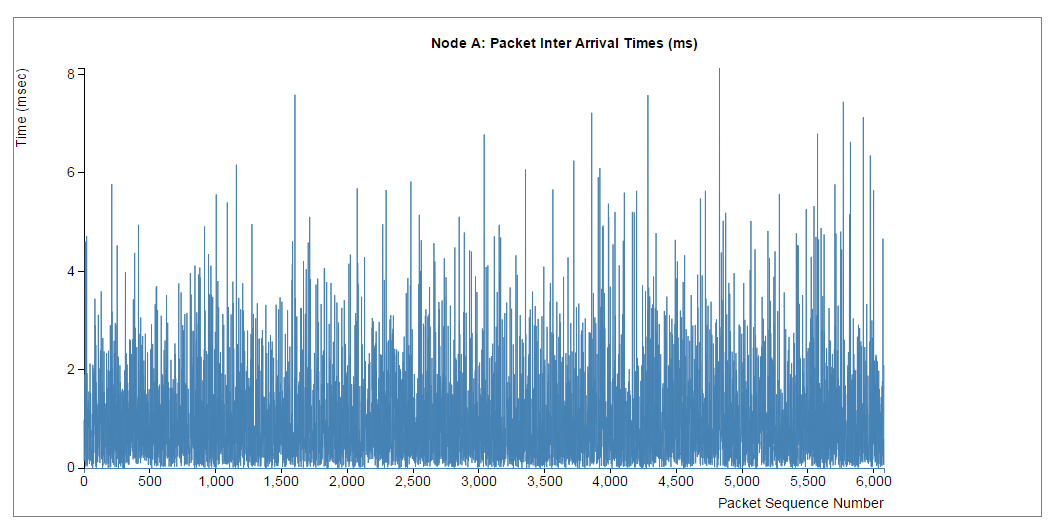
**Comment:**

Whenever the link cost changes, routers update their forwarding table accordingly. This ensures that packets do not follow a higher cost path.

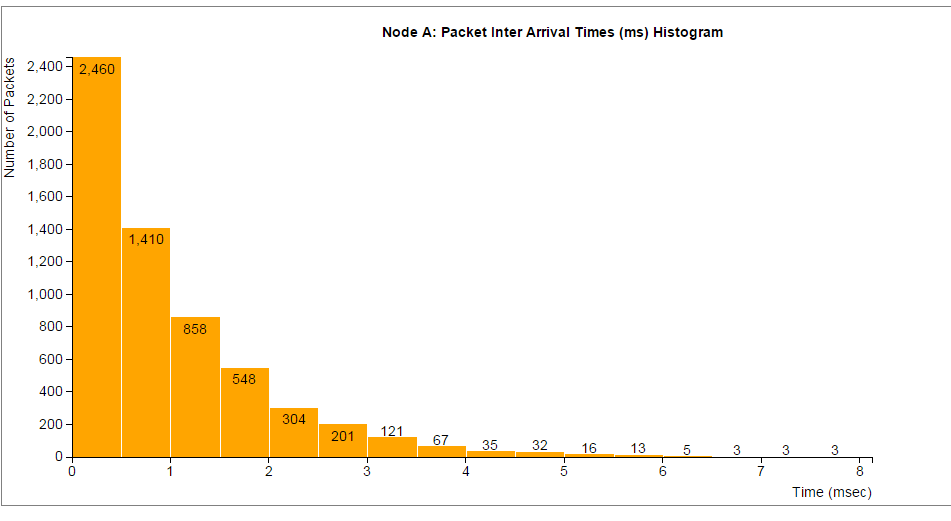
Results for other nodes are similar.

**Interpacket Arrival Time per packet:**

**Node A**

****

**Histogram:**

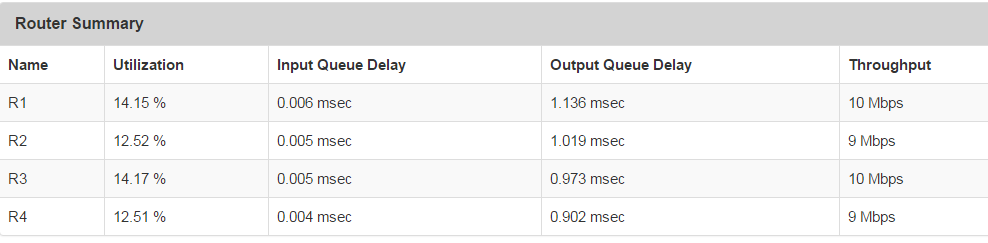


**Observation**:

The interarrival time is exponentially distributed, as expected.

Results for other nodes are similar.

**Router Summary:**

****

**Observations:**

The utilization of each router is near 13 %, output queue delay is near 1 millisec and Throughput is around 10 Mbps.

**Comments:**

With the given inter packet arrival of 2 frame slots and a frame slot duration of 0.5 millisec, on an average 1 packet arrives every 1 millisec. Also, the router processing delay is 0.1 millisec, meaning that it can process 10 packets every 1 millisec. So, the theoretical utilization of the router should be

1/10 \* 100 = 10%. The observed value 13 % is close to this value.

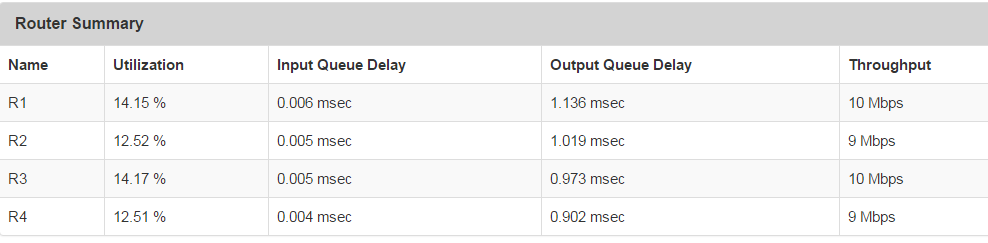
**Results for a different Topology:**

**Improving Network Performance by changing Network Parameters:**

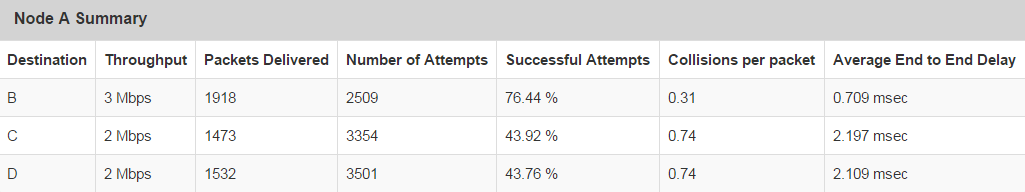
**1. Effect of Router Processing Delay:**

Router Processing Delay = 0.1 millisec

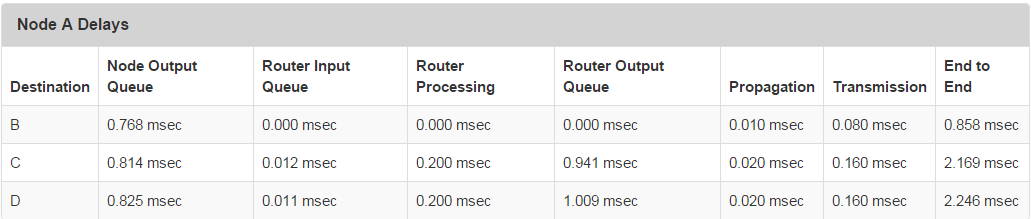
**Router Summary:**



**Node Summary:**

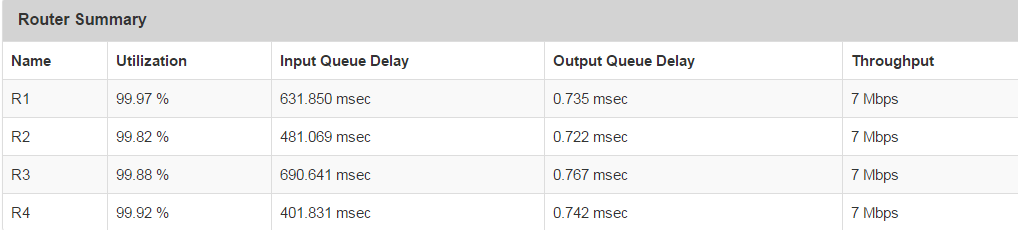


**Packet Delays:**

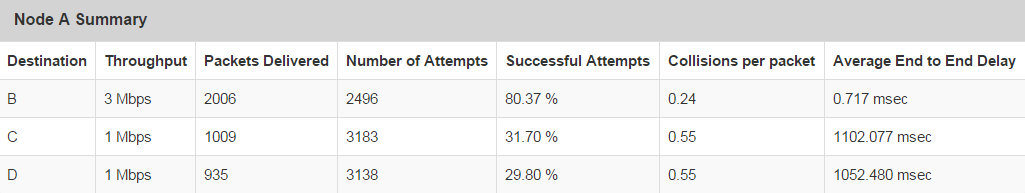


Router Processing Delay = 1 millisec

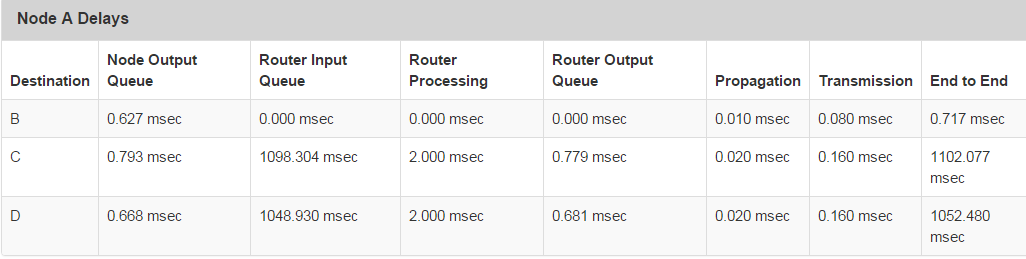
**Router Summary:**



**Node Summary:**



**Packet Delays:**



**Observations:**

The “Router Summary “shows that the utilization of routers increase to approximately 100% from 13% and the input queue delay at the routers increases to 400 – 600 millisec from roughly 0 millisec.

The “Node Summary” shows that the throughput for Node A to C and A to D drops to 1 Mbps from 2 Mbps and the end to end delay increase to around 1.1 sec from 2.16 millisec.

The “Packet Delays” shows that the major contributing factor to this increase in end to end delay is the Router Input Queue Delay.

**Comments:**

When we increase the router processing time to 1 millisec, then it can process roughly 1 packet every 1 millisec. This is also the rate at which packets are arriving. So, the router is utilized for nearly 100% of the time. As a result, packets have to wait to be processed at the input queue of the router and the queue size would keep on increasing. This leads to the increase in the router input queue delay, which causes the end to end delay to increase and the throughput to reduce.

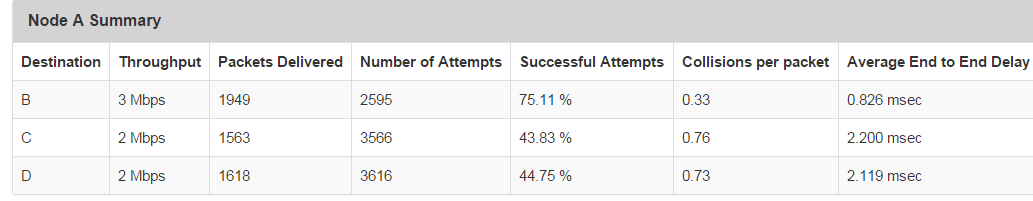
**Conclusion:**

For maximum throughput and minimum end to end delay, the router should be able to handle the demands of the network, i.e. the processing rate of the router should be greater than the inter arrival rate of the packets.

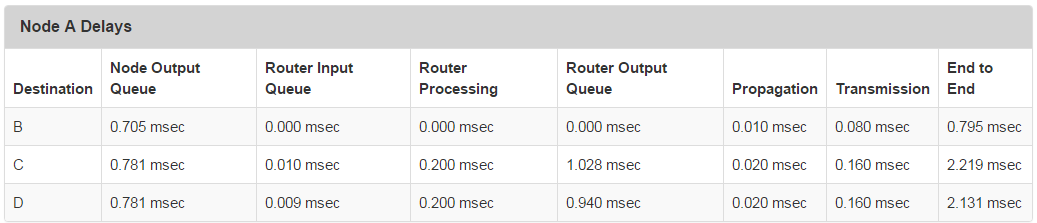
**2. Effect of Transmission Rate:**

Transmission Rate = 100 Mbps

**Node Summary:**

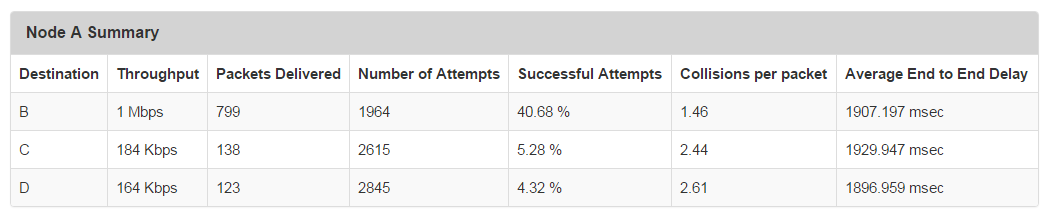
****

**Packet Delays:**

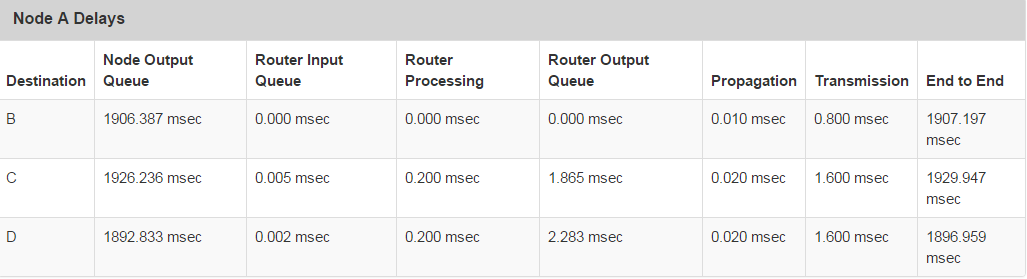


Transmission Rate = 10 Mbps

**Node Summary:**

****

**Packet Delays:**



**Observations:**

The “Node Summary” shows the throughput from A to B reduces from 3 Mbps to around 1 Mbps, and the throughput from A to C and A to C reduce from around 2 Mbps to around 180 kbps.

The “Packet Delays” show that the delay at the output of nodes increase from around 0.75 millisec to around 1.9 sec.

**Comments:**

If we reduce the transmission rate, then a packet will take a larger duration to be transmitted out. Till this duration, other packets will be waiting in the queue. This leads to a large queue delay, which results in large end to end delay and lower throughput.

Also, in our simulation, we have chosen a large packet size of 1000 bytes. So, the transmission time is not negligible as can be seen.

**Conclusion:**

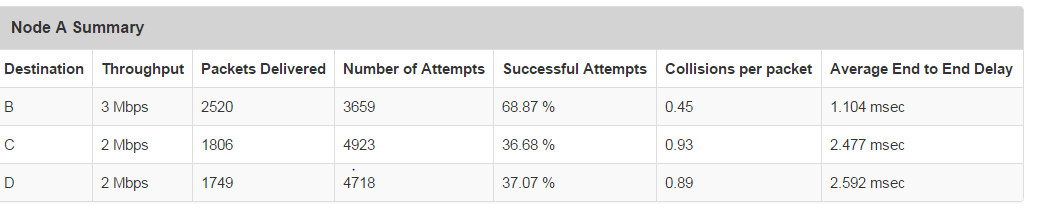
For maximum throughput, we should use a higher transmission rate, especially if the packet size is large.

**3. Effect of Inter Node Distance**

(Note Frame Slot = 0.4 millisec ( rather than 0.5 millisec ) for this section to take advantage of shorter distance and smaller wait time to detect collision. A smaller frame slot leads to more packets generated per second and smaller backoff delays in case of collisions )

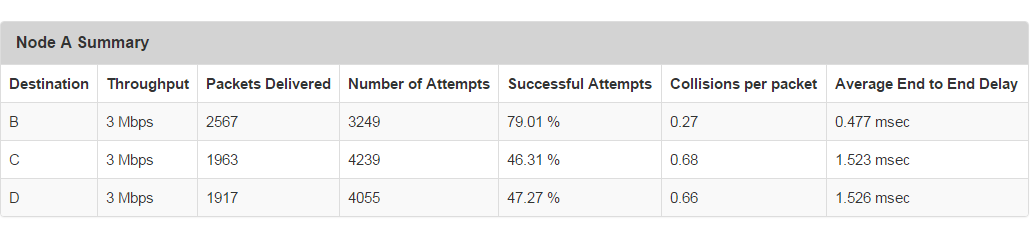
Inter Node Distance = 2000m

**Node Summary:**



Inter Node Distance = 200m

**Node Summary:**

****

**Observations:**

From the “Node Summary”, we observe that the number of collisions per packet has reduced from around 0.45 per packet from A to B to around 0.27 per packet and from 0.90 per packet from A to C and D to around 0.70 per packet.

The percentage of successful attempts has also increased from around 69% to 79% for packets from A to B and from around 37% to 47% for packets from A to C and D.

**Comments:**

A smaller inter node distance reduces the chances of collisions as packets travel for a shorter distance and can reach the destination faster, before the next node has a packet ready to send and collide with the existing packet. This results in an increase in the number of successful attempts. This, in turn, leads to higher throughput.

**Conclusion:**

A smaller inter node distance can help to reduce the number of collisions and retransmissions and increase the throughput.