Delay/Jitter Analysis for HP in the Two Transit Buffer Scheme of Darwin and Comparison

Vasan Karighattam

Necdet Uzun

Donghui Xie

Mete Yilmaz

Pinar Yilmaz

Objective

Analyze the delay-jitter for High Priority traffic, and size the Low Priority Transit buffer for the optional two transit buffer scheme in the Darwin v1.0 draft.

Model Used

- We are modeling a 16 node,10 G Ring, 2000 Km
- Use the Darwin v1.0 media access control
- Each node has a token bucket filter for each priority served
- The token bucket filter is specified as (r,b) where r is the rate and b is the depth. (It is 534 bytes deep for High Priority)
- Token buckets accumulate tokens at the SLA rate and empty tokens at line rate
- A packet is sent only if the corresponding token bucket has at least one token
- When the HP queue is silent, token bucket is saturated

Media Access Control

1. High Transit Buffer
2. If (Low Transit Buffer > High Threshold) then

   Low Transit Buffer
3. High Transmit Buffer
4. If (Medium Priority Token Available) then

   Medium Transmit Buffer .....

RULE: For guaranteed delay-jitter, line 2 should not be executed

Bandwidth Allocation

- For the previous rule to work, either the LPTB should be very large or HP traffic in the congested segment should be limited to the smaller LPTB.
- For a 10G, 2000Km ring, 10% HP traffic, the size of LPTB would be calculated as follows:

  LPTB_size = (C-Ó r_i - Ó r_k)*ring_size*5us/km*(%HP) + (TB_low_fairness_threshold)

  where C is the line rate, r_i is the reserved HP rate (if any), r_k is the sum of the HP allocations for upstream nodes
- Setting r_i & r_k to zero and setting the threshold at the end of the buffer, we calculate the size of LPTB to be 1.25MB

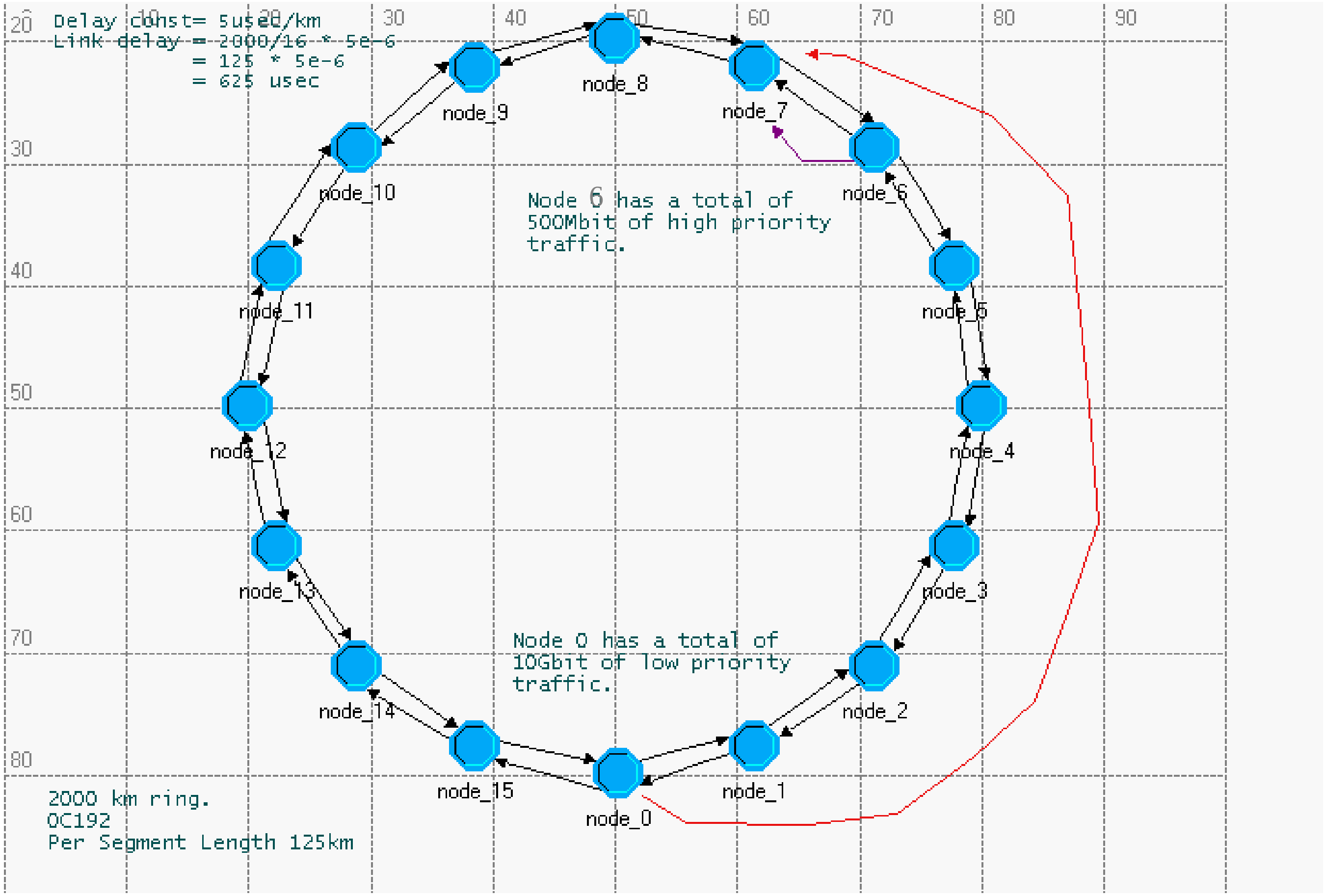
Delay Jitter Calculation

- Once we obey the previous rule and use the stated token-bucket model at every node, delay-jitter can be calculated based on the following worst case model
- All nodes are transmitting a HP packet simultaneously, adjusted by the link propagation delay. The Nth node could also have a packet already in transit at this time.
- The ideal value of delay-jitter in the fluid model is 0
- On a ring with 2N nodes, the best possible delay-jitter is (N+1)*MTU (Half the ring has N nodes and there could be a packet in transit at this time at the Nth node)

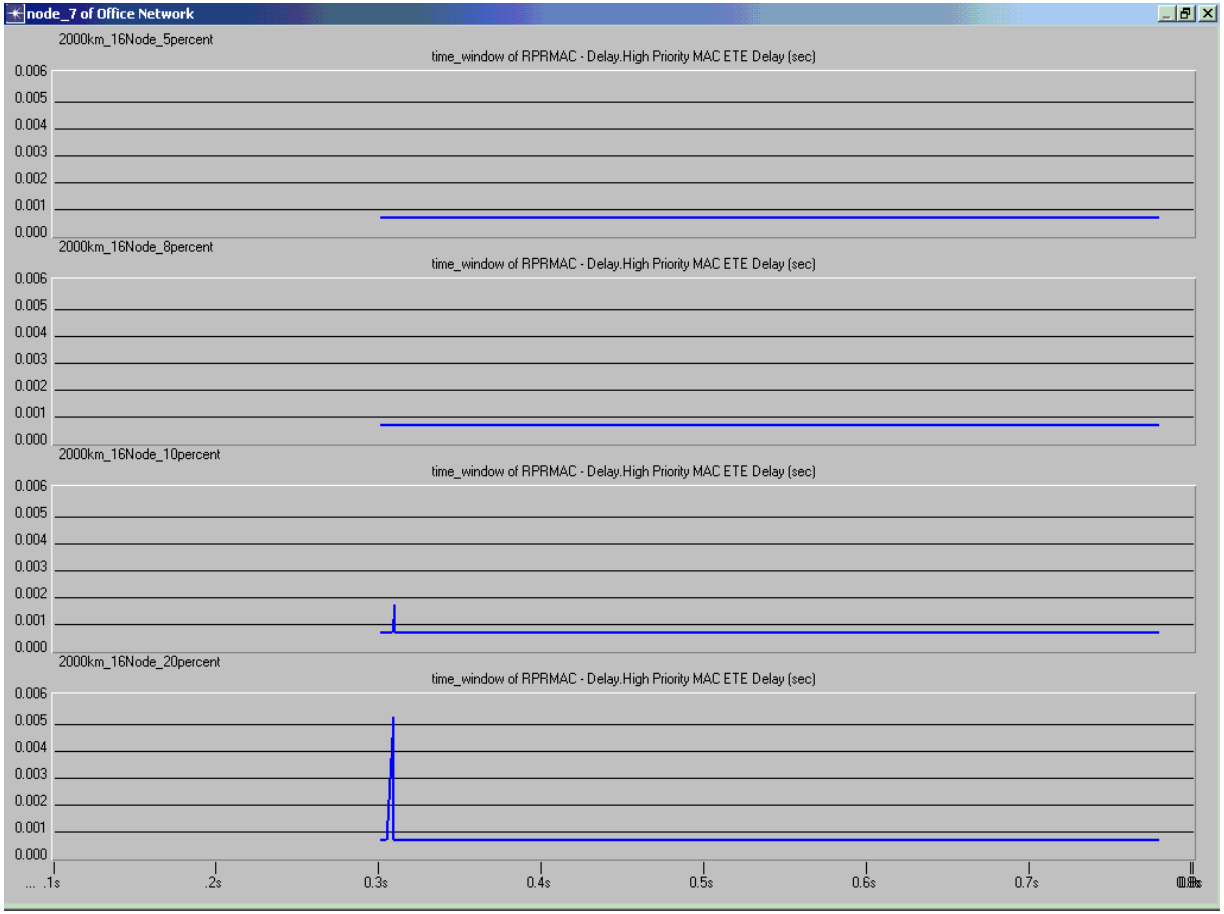
Generalization

- LPTB has to be sized according to the percentage of HP traffic passing through the node (So ½(1.25MB) for 5% and 2*(1.25MB) for 20%)
- This will eliminate the cost of bandwidth reclamation
- If Bandwidth is reserved around the ring, or ring segment for HP, there is no need for the second transit buffer

Simulation Model – RTT delay

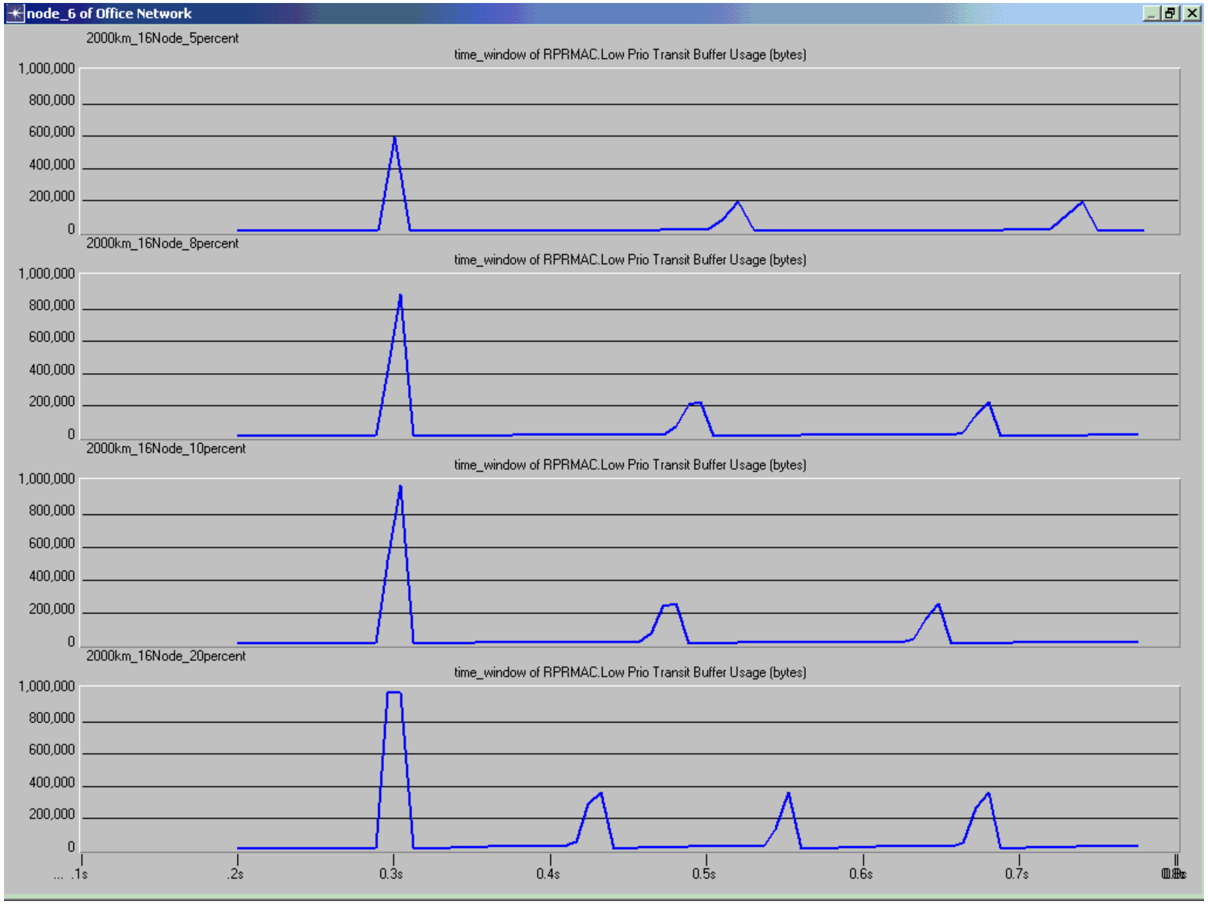


Simulation Results – RTT Delay

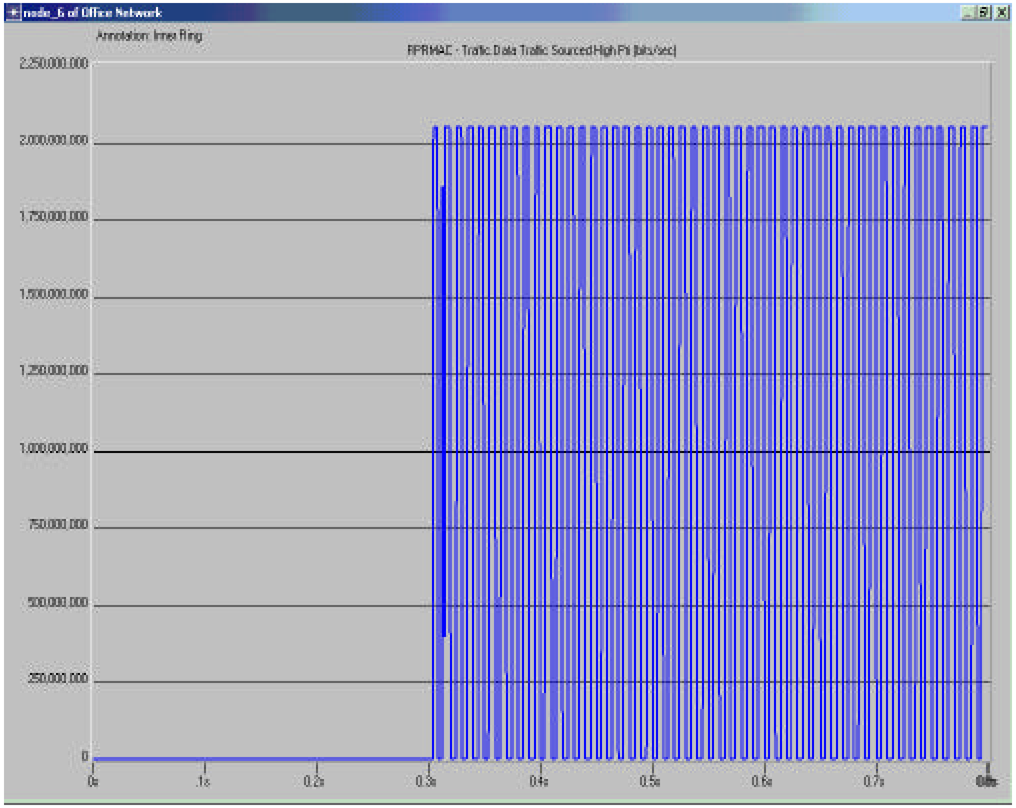


With LP Transit Buffer:
reclamation time << 2RTT

Transit Buffer Occupancy

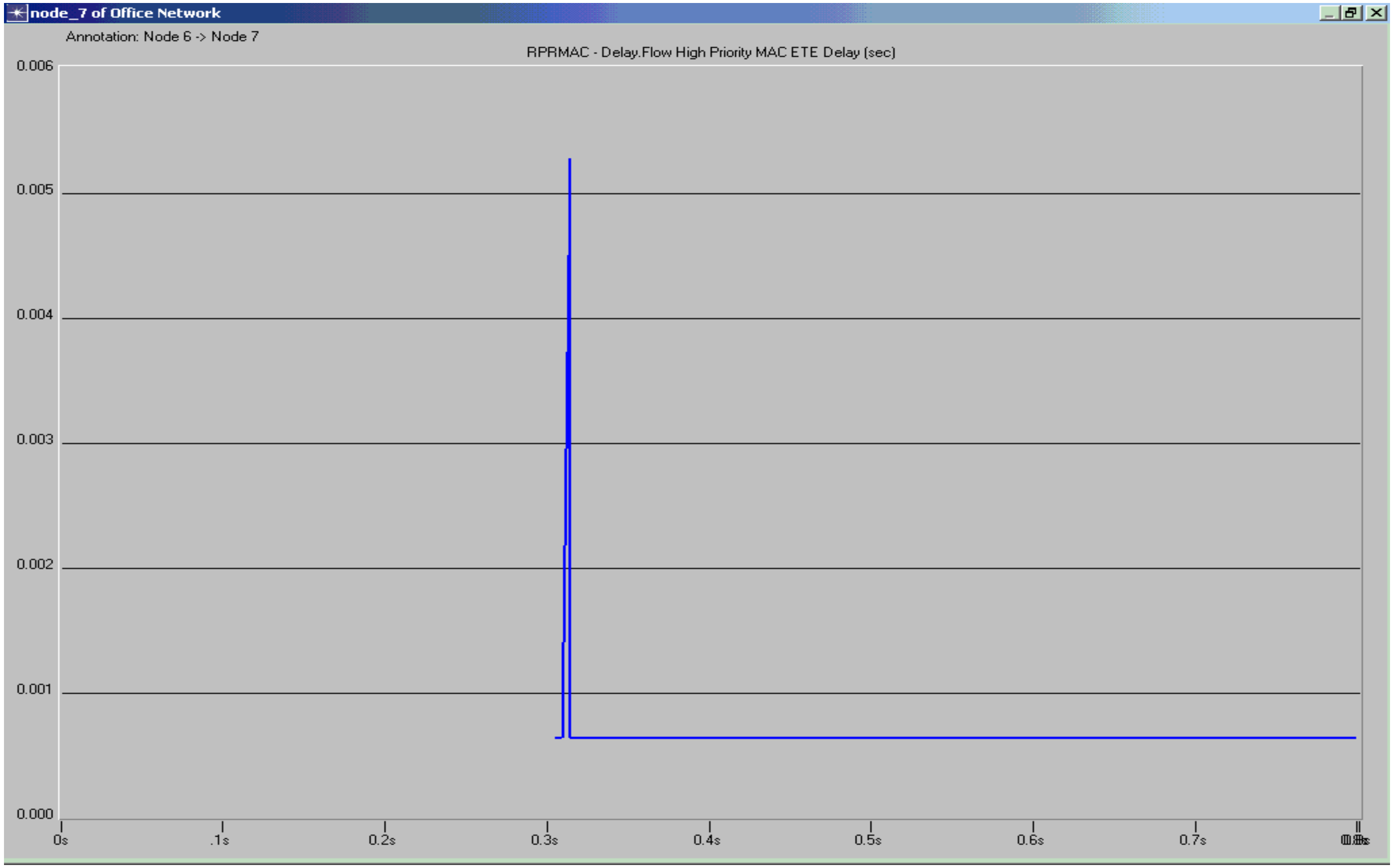


Scenario 2 – Bursty HP Traffic

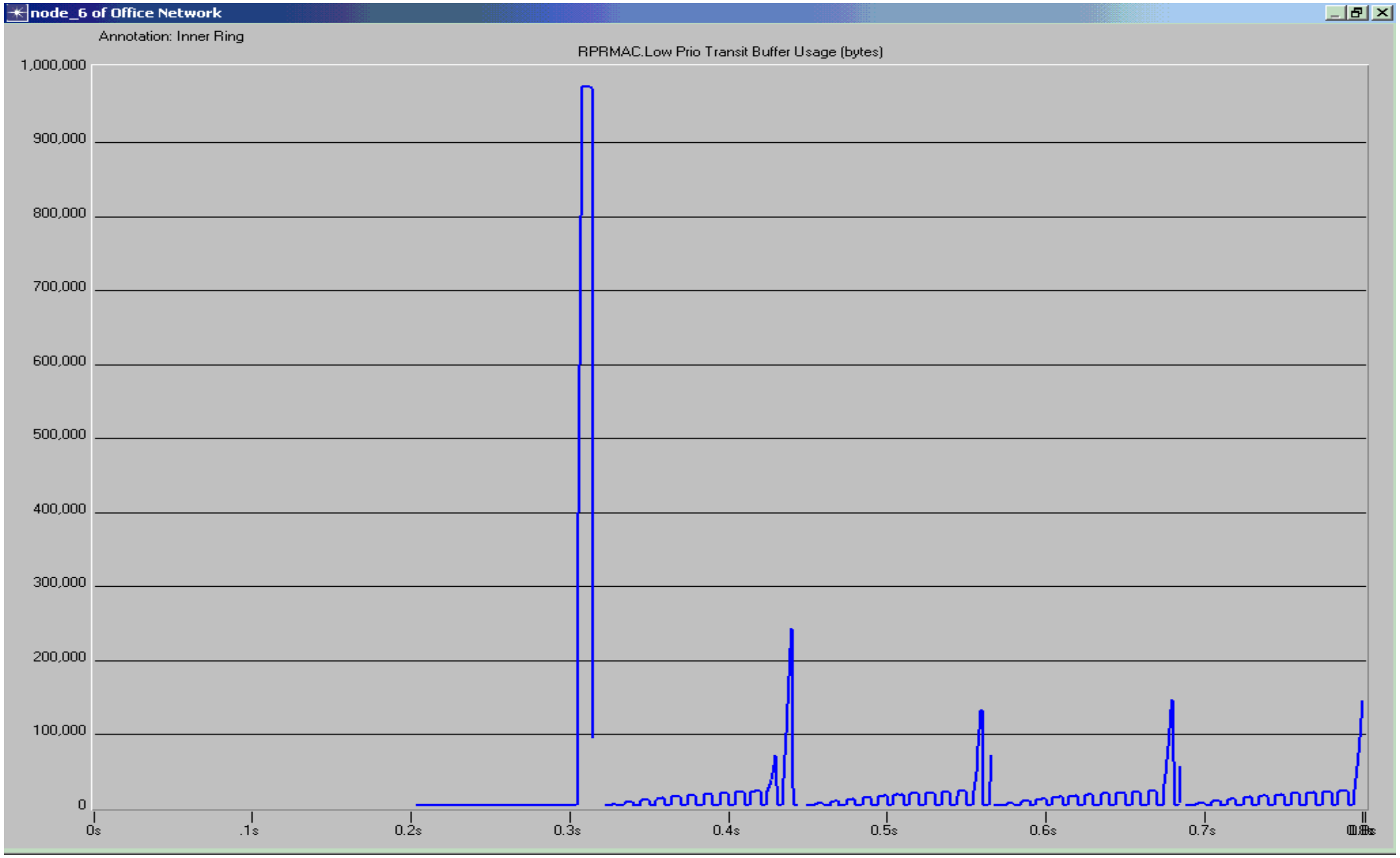


Traffic Scenario:
Bursty HP Traffic
5ms ON
5ms OFF

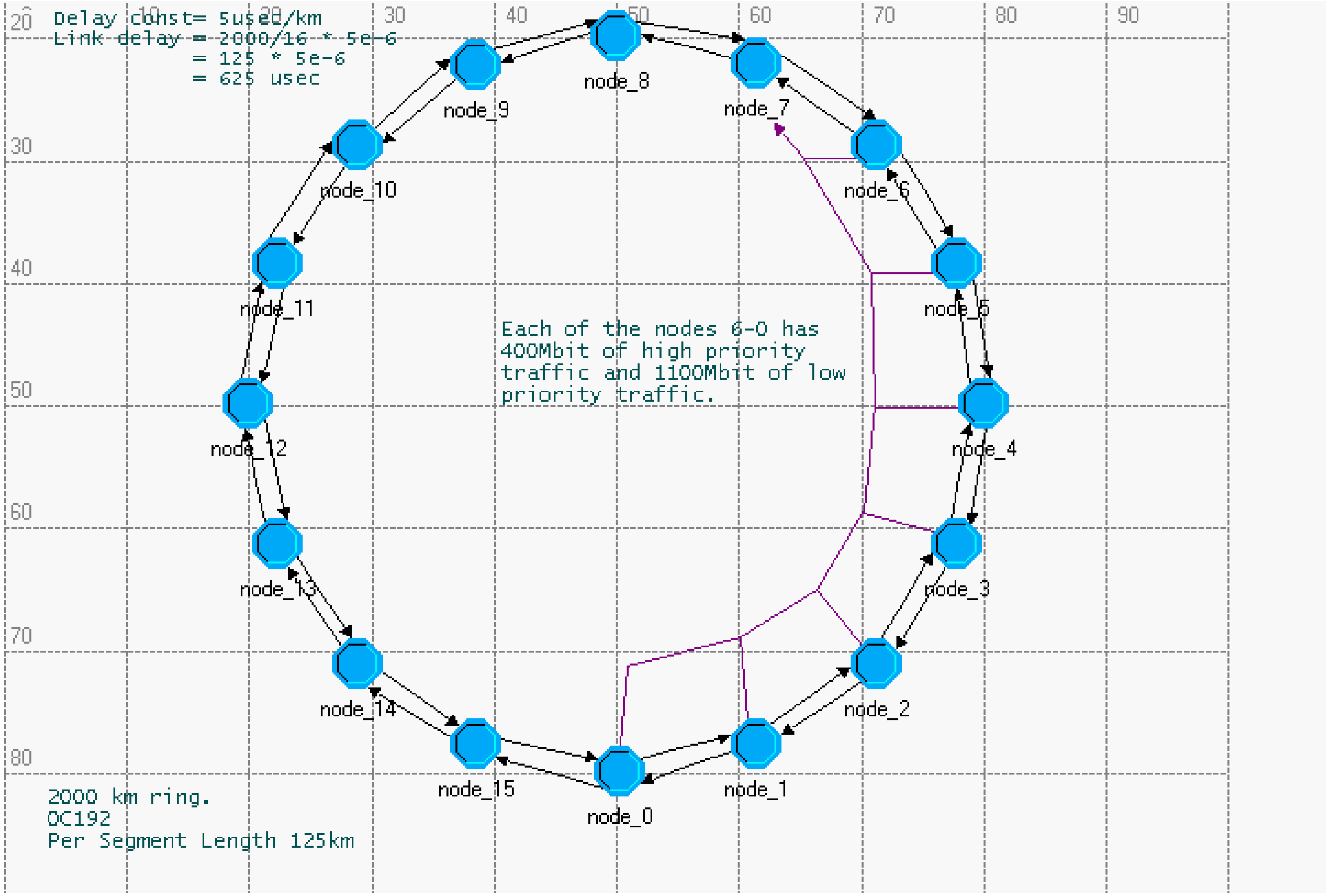
ETE Delay for HP Traffic



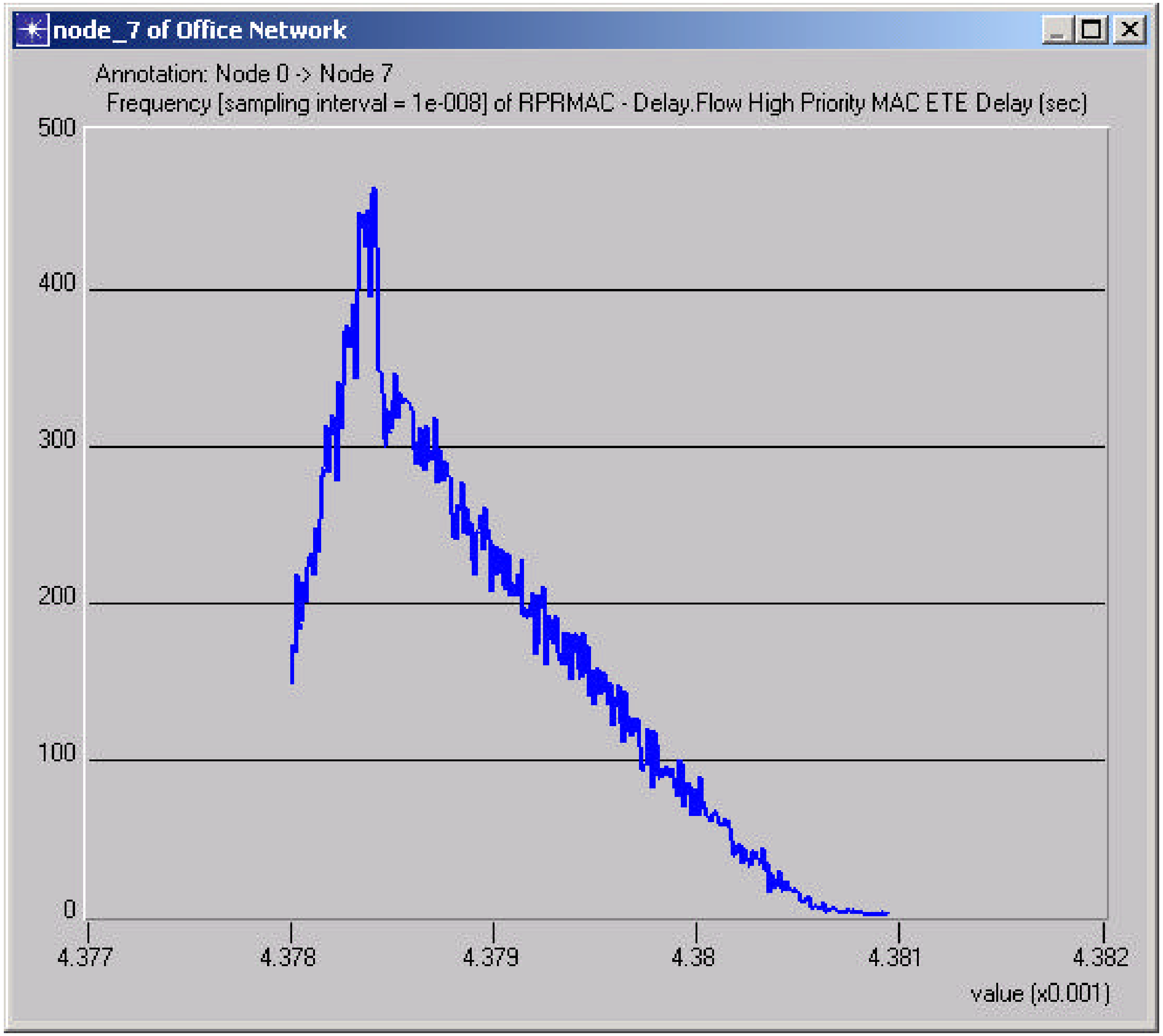
Transit Buffer Occupancy



Simulation Model – N MTU Delay



Simulation Results – N   MTU Delay



Calculated Worst Case Jitter: 7MTU = 2.99us.

Observed: 2.96us

Summary

- With the LP Transit Buffer, Bandwidth reclamation time << 2RTT
- With a leaky bucket at the ingress of each node, the worst case delay-jitter can be calculated to be (N+1)*MTU
- The two transit buffer scheme can be used to support very low-jitter real-time applications without the need for bandwidth reservation

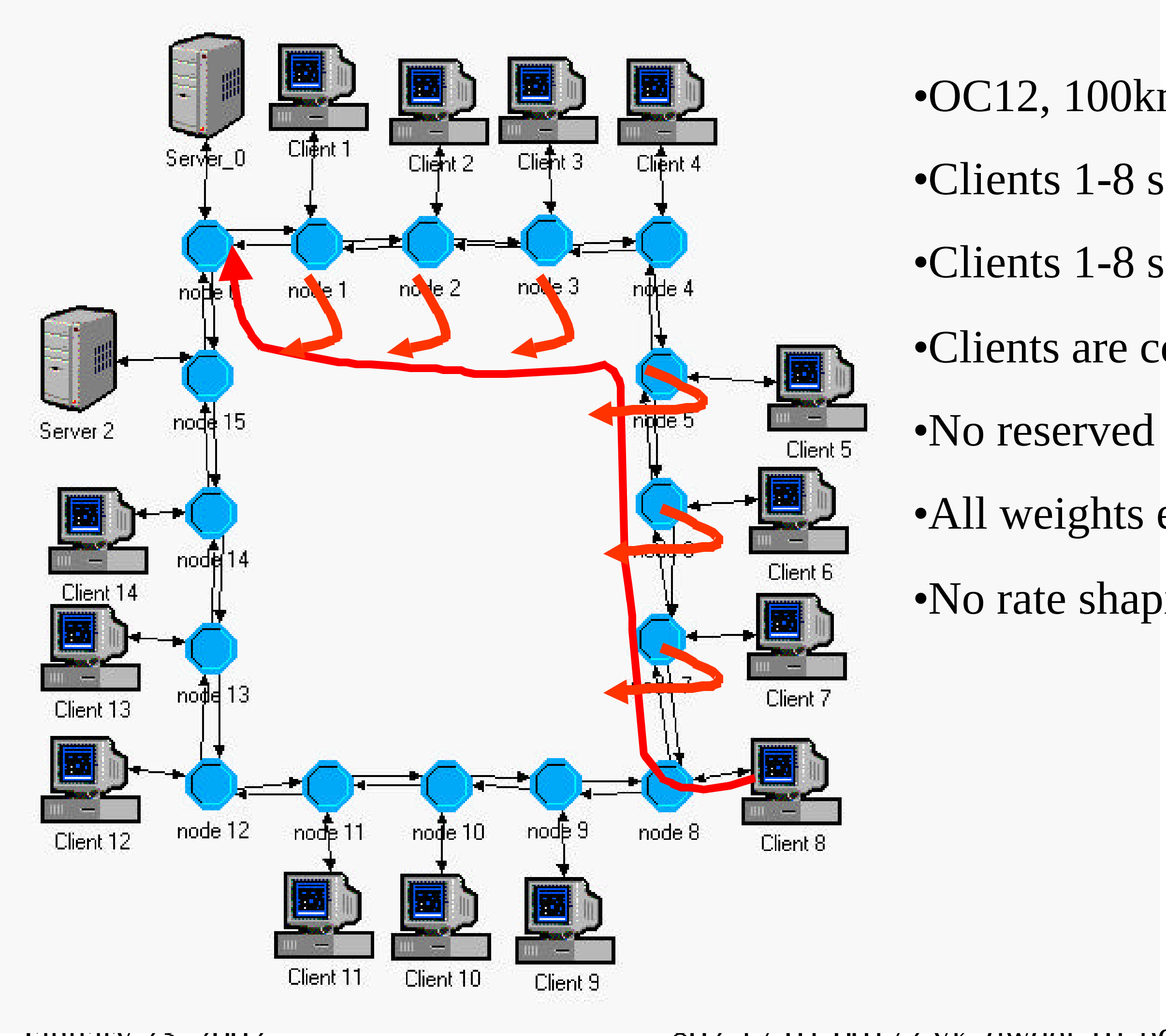
Comparison through Simulations

Simulation Scenarios

- Five sets of scenarios with VoQ_Release v2 and Gandalf v1
- Gandalf model emulates Darwin behavior with default parameters and 2 transit buffers
- Under bursty traffic (dominating characteristic of Internet), Darwin performed well
- Darwin exhibits superior performance while optimizing delay/jitter and throughput
- Darwin can react faster in bursty conditions

Oversubscribed Ring
in
Hub Configuration

Hub Scenario – OC12

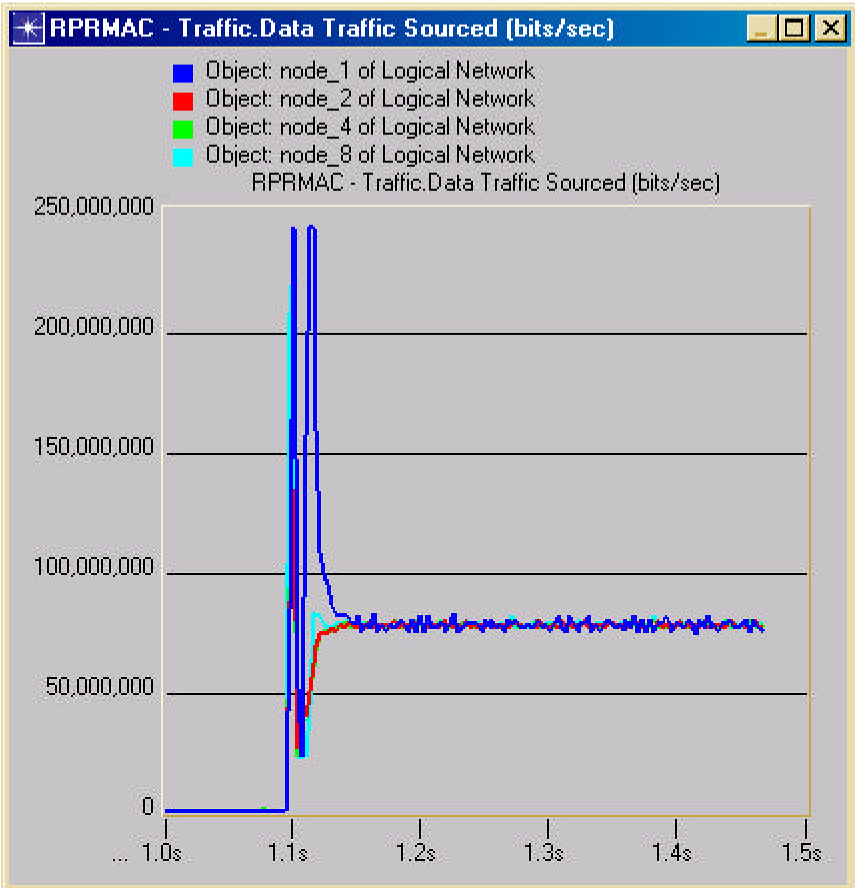


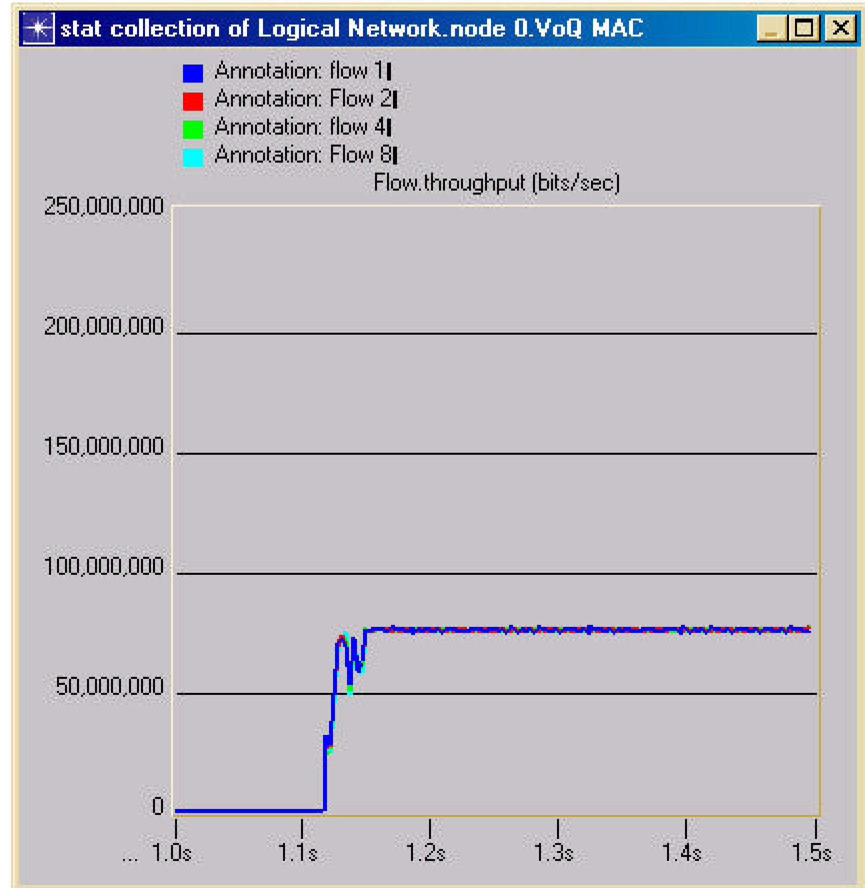
- OC12, 100km, 16 nodes
- Clients 1-8 send 22.5 Mbps CBR HP to Server 0
- Clients 1-8 send 750Mbps CBR LP to Server 0
- Clients are connected to RPR nodes via 10GE
- No reserved BW
- All weights equal to 1
- No rate shaping (Darwin)

Hub Scenario – OC12

Darwin

Alladin

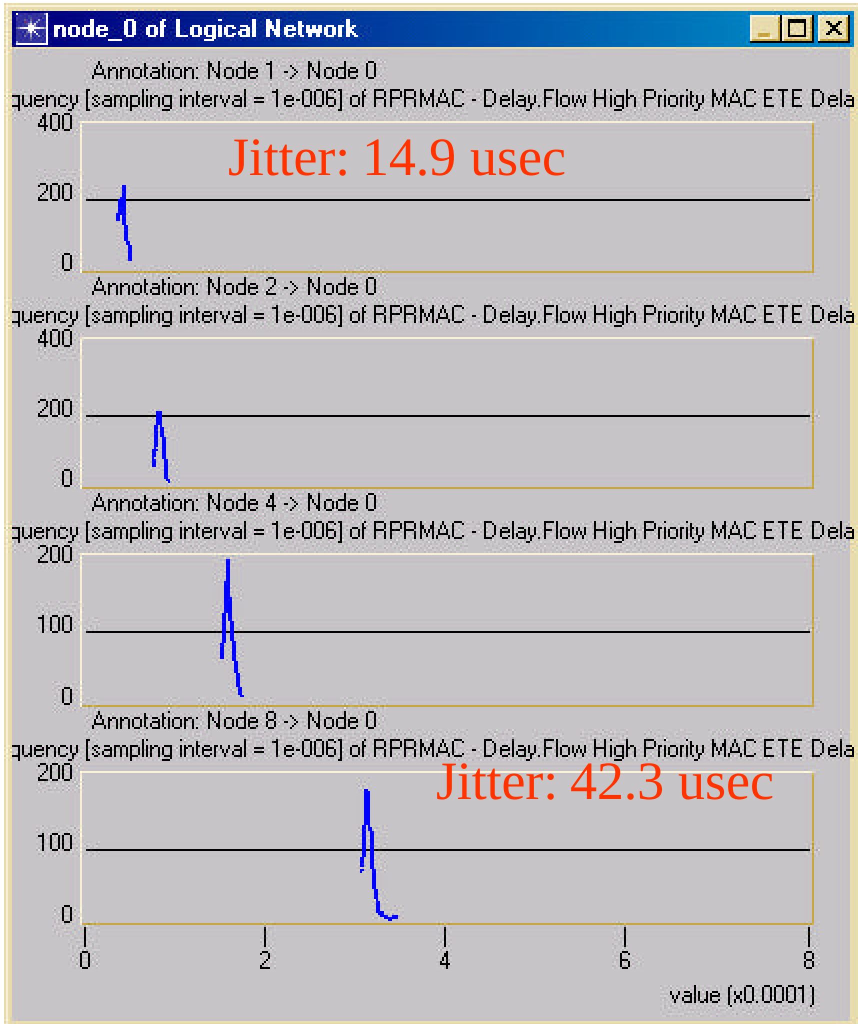




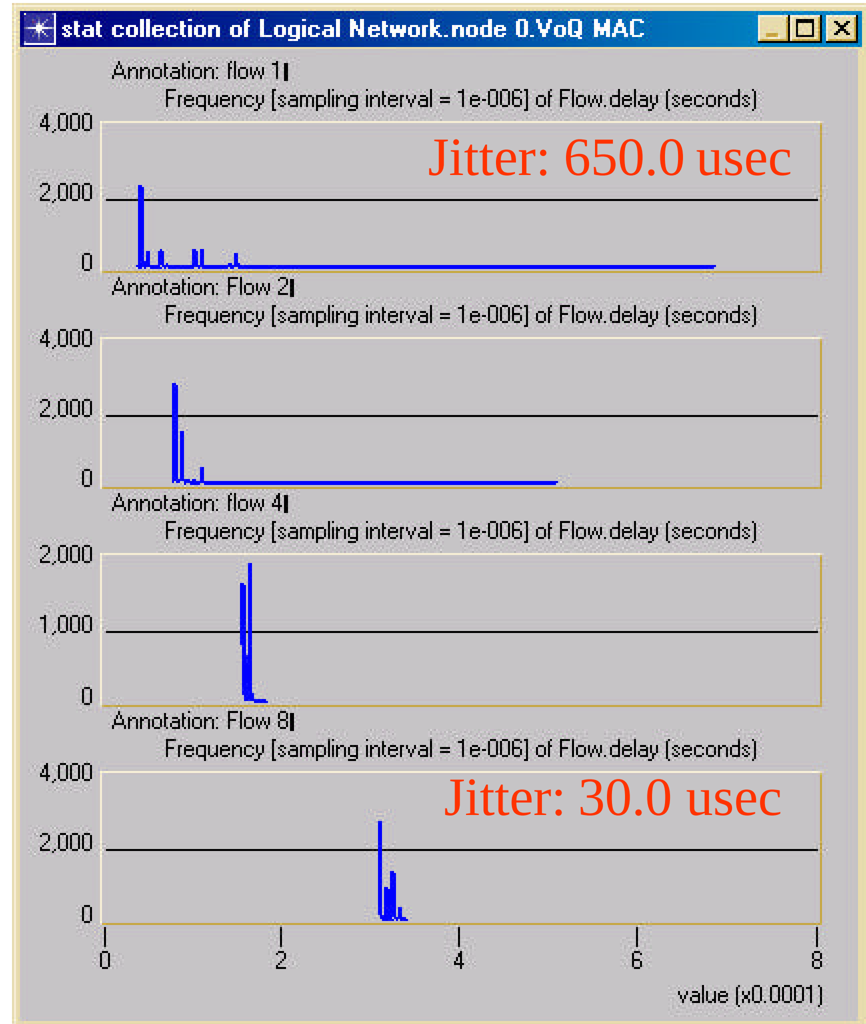
Traffic sourced from RPR Nodes

Hub Scenario – OC12

Darwin

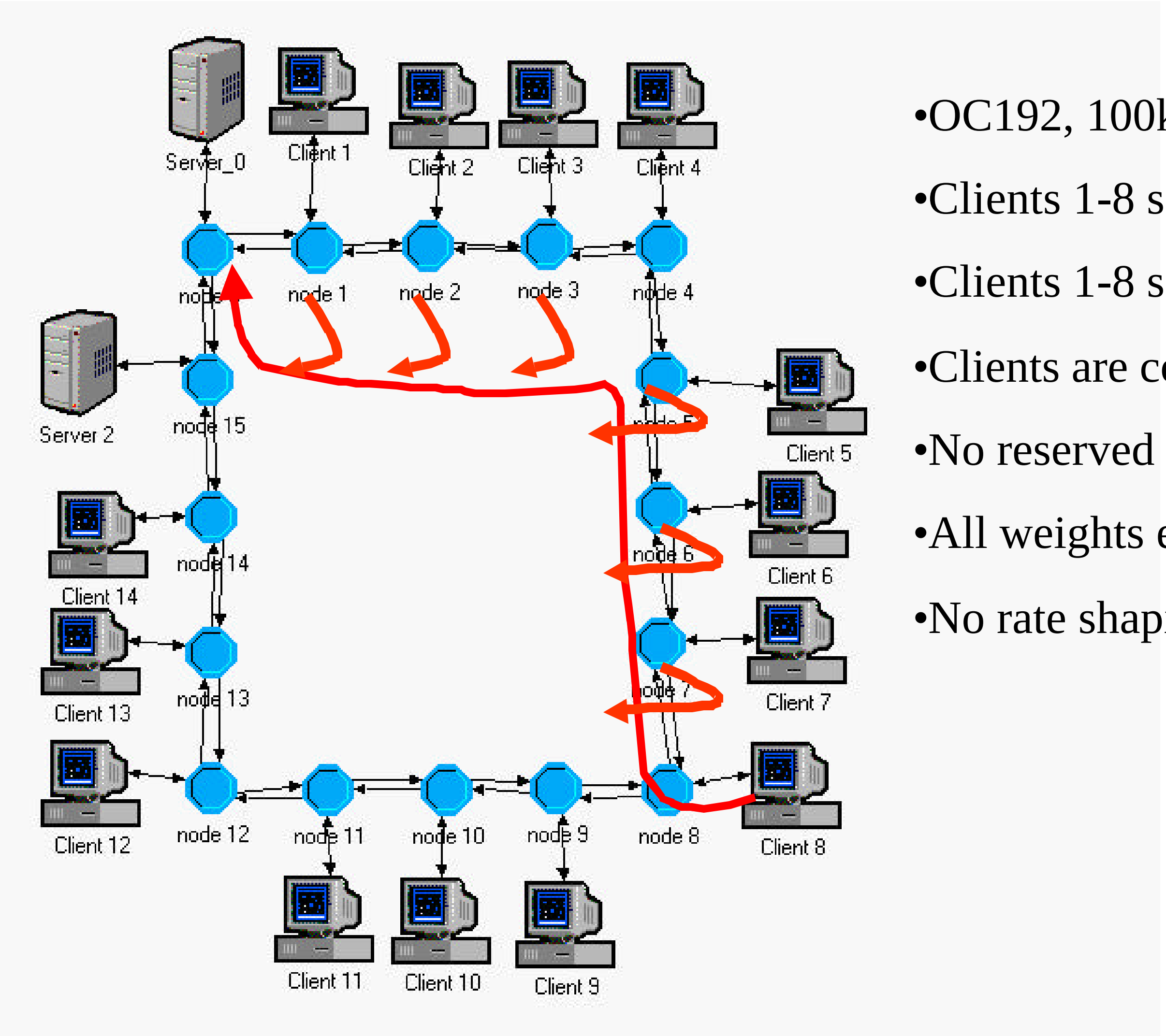


Alladin



Delay Value and Histogram

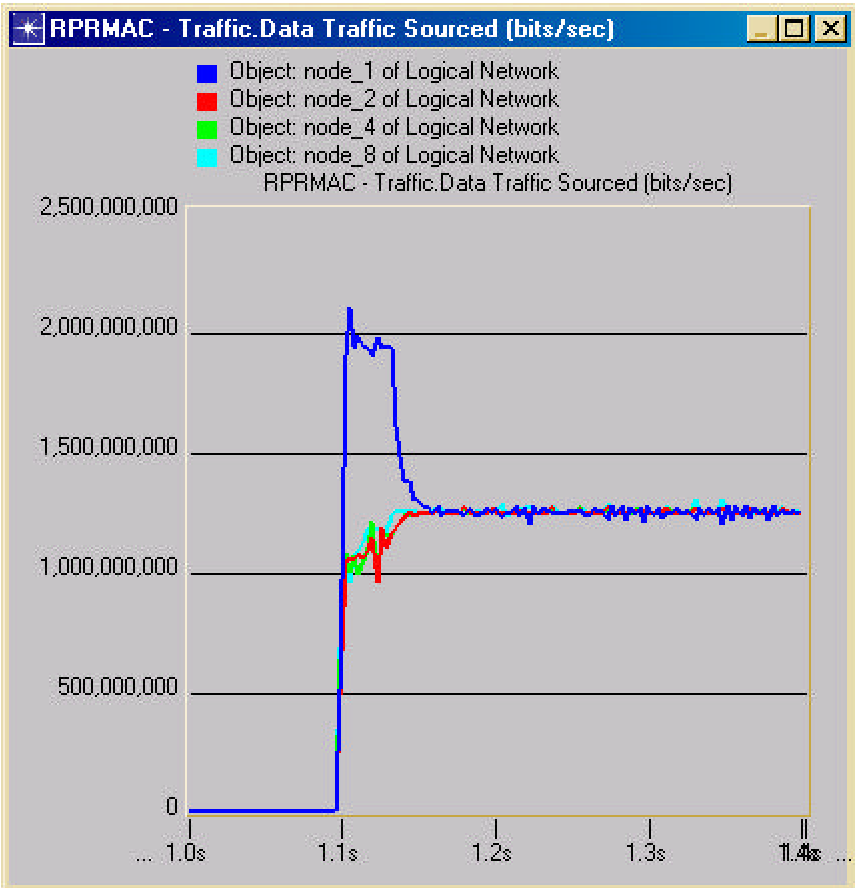
Hub Scenario – OC192



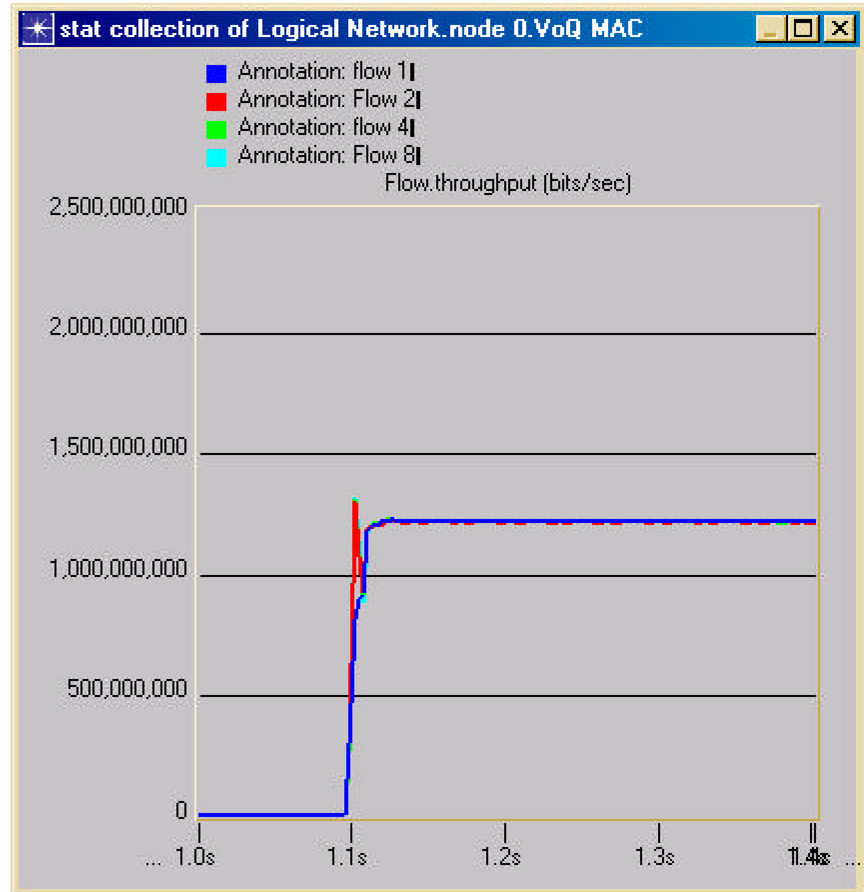
- OC192, 100km, 16 nodes
- Clients 1-8 send 483 Mbps CBR HP to Server 0
- Clients 1-8 send 1.45Gbps CBR LP to Server 0
- Clients are connected to RPR nodes via 10GE
- No reserved BW
- All weights equal to 1
- No rate shaping (Darwin)

Hub Scenario – OC192 (100km)

Darwin



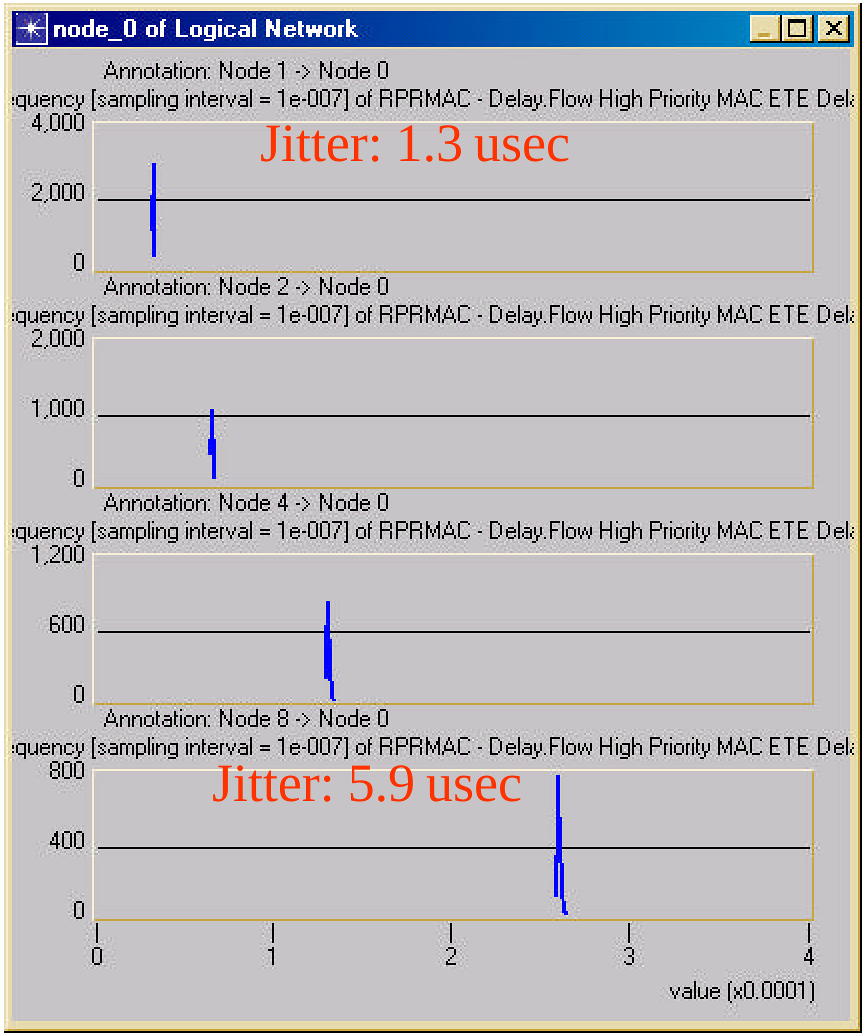
Alladin



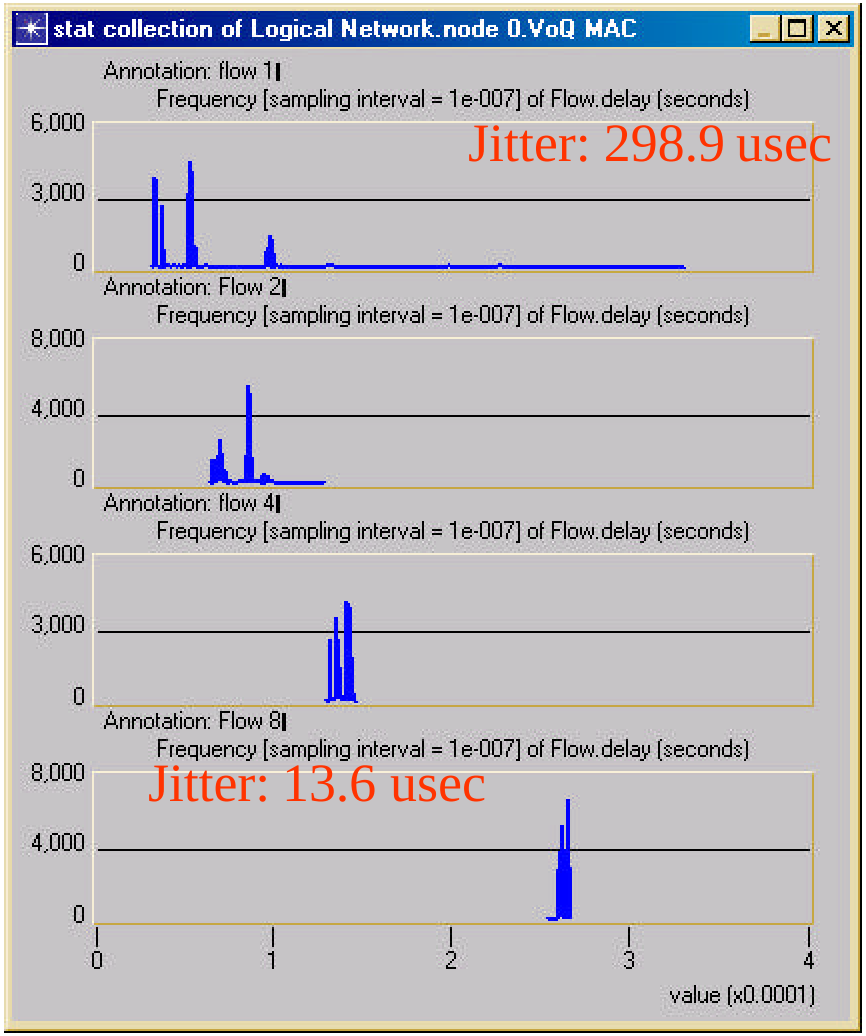
Traffic sourced from RPR Nodes

Hub Scenario – OC192 (100km)

Darwin



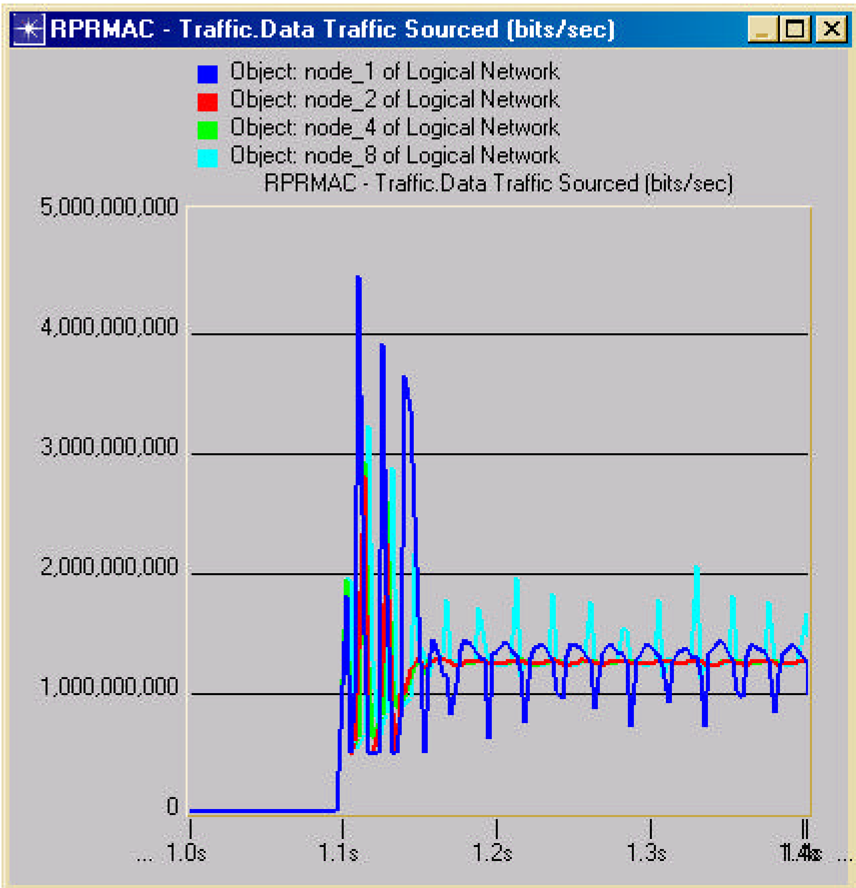
Alladin



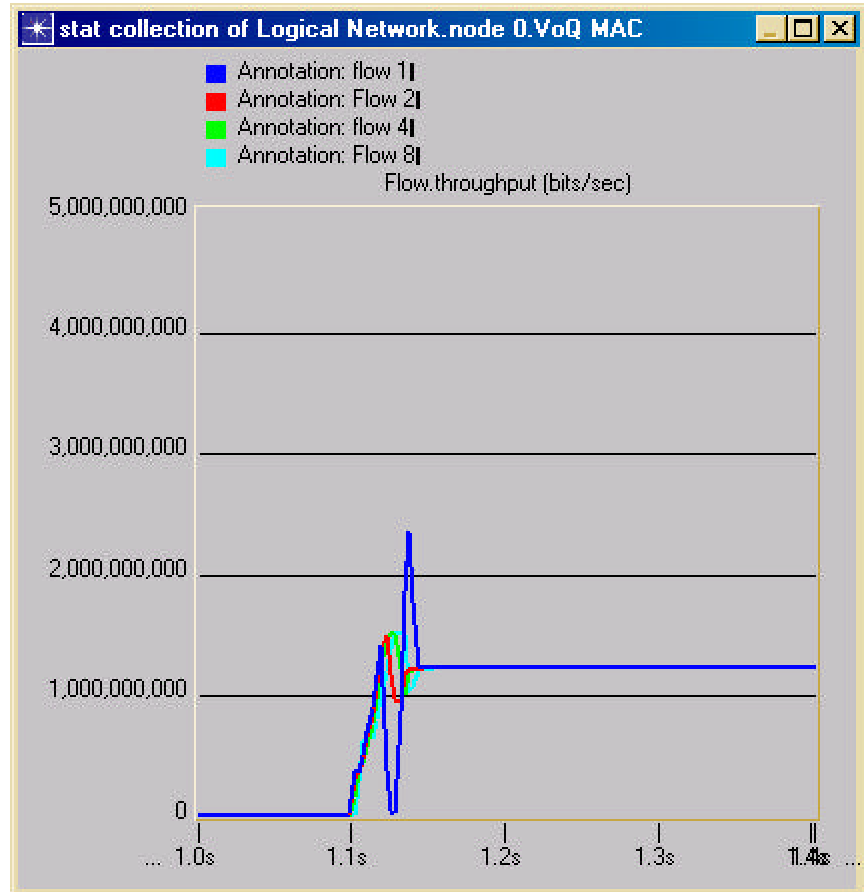
Delay Value and Histogram

Hub Scenario – OC192 (2000km)

Darwin



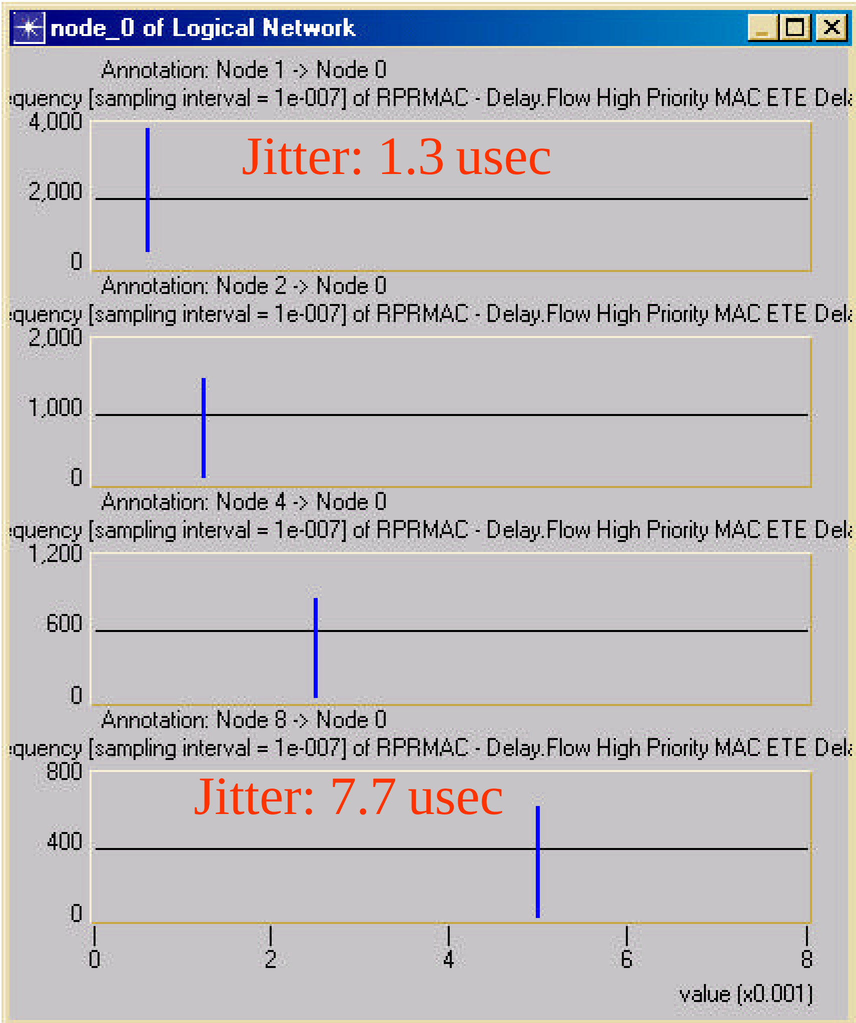
Alladin



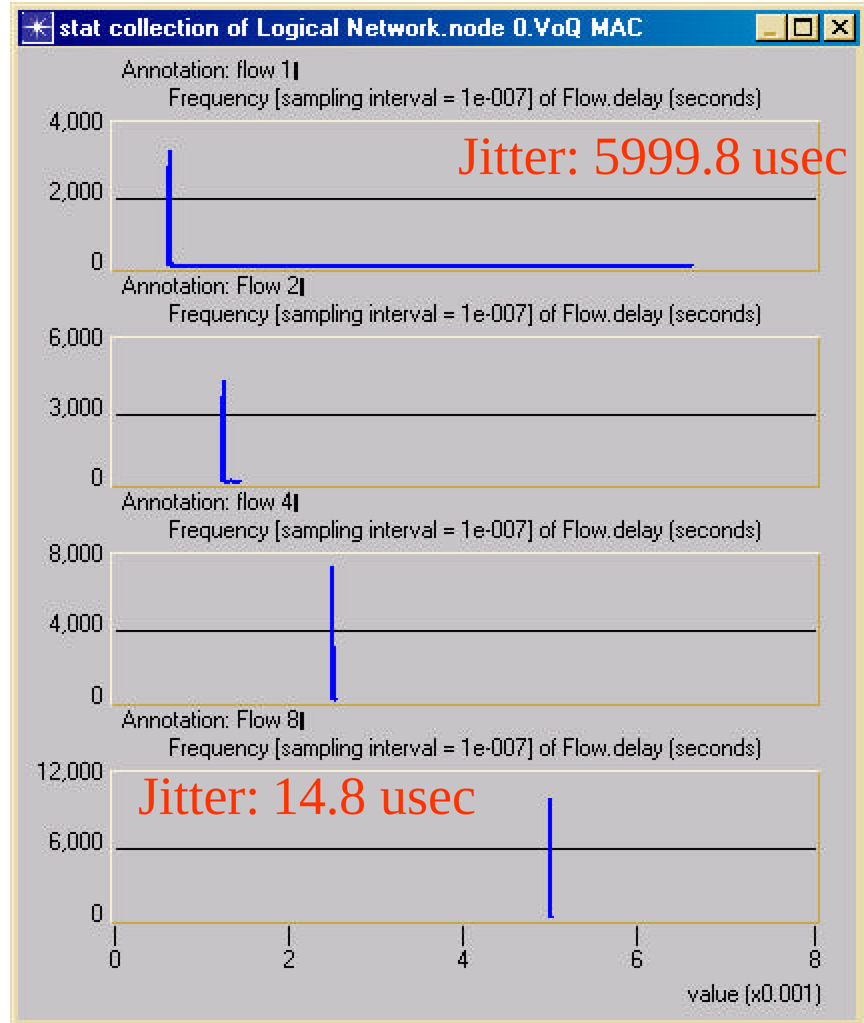
Traffic sourced from RPR Nodes

Hub Scenario – OC192 (2000km)

Darwin



Alladin



Delay Value and Histogram

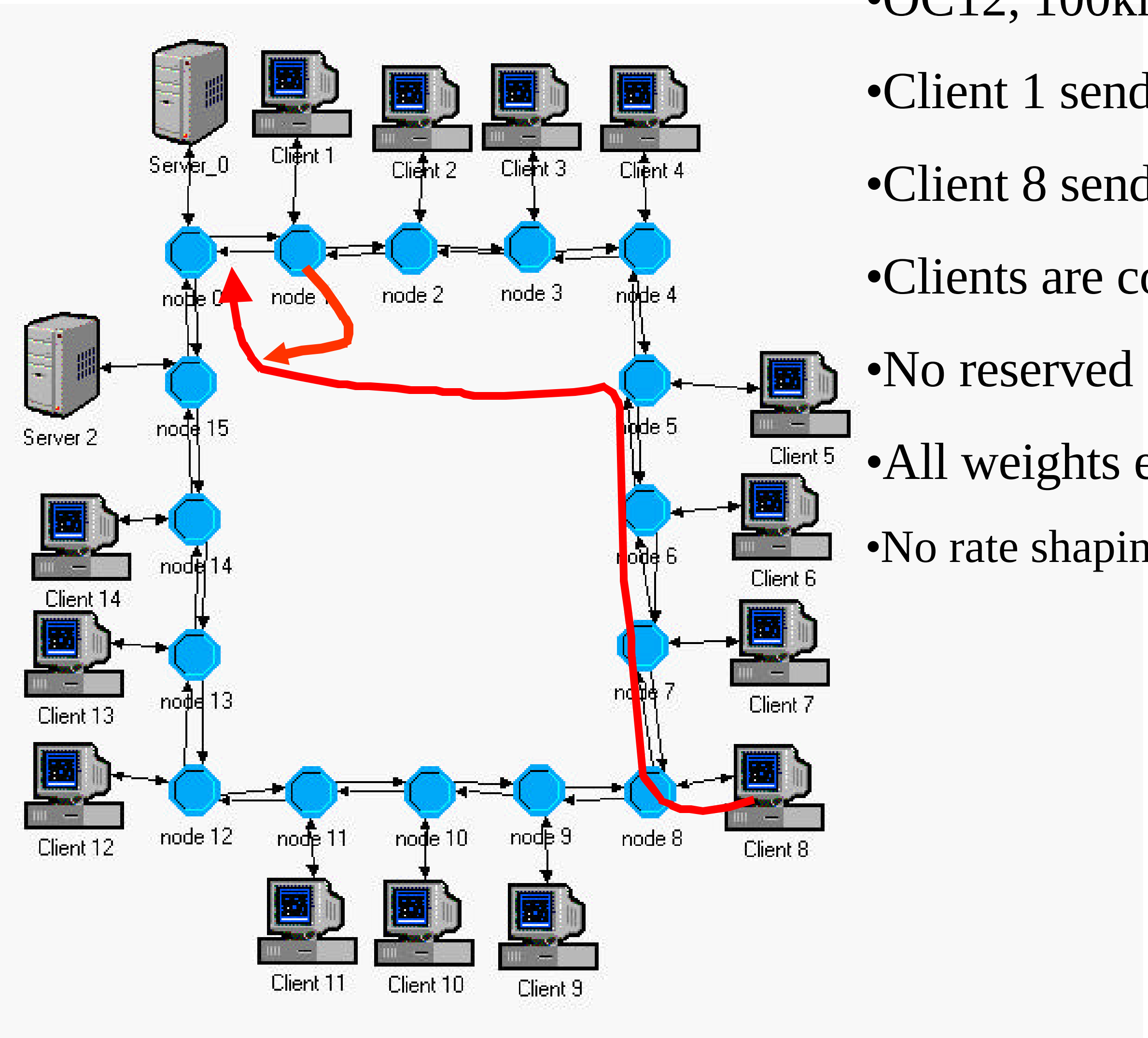
Delay Values

(usec)

		Alladin		Darwin	
		Node 1	Node 8	Node 1	Node 8
OC12 (100km)	min	39.4	312.0	38.5	307.1
	max	689.4	342.0	53.4	350.1
OC192 (100km)	min	32.5	255.3	32.5	259.8
	max	331.4	268.9	33.8	265.7
OC192 (2000km)	min	625.2	5008.0	626.2	5009.8
	max	6625.0	5022.8	627.5	5017.5

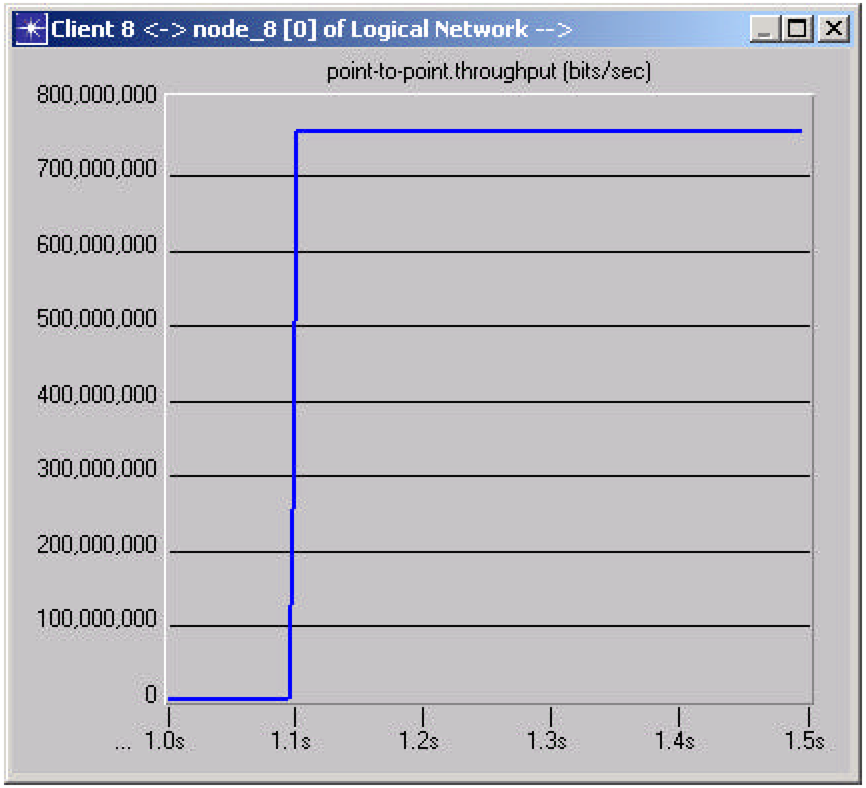
Pathological Scenario (Non-Bursty)

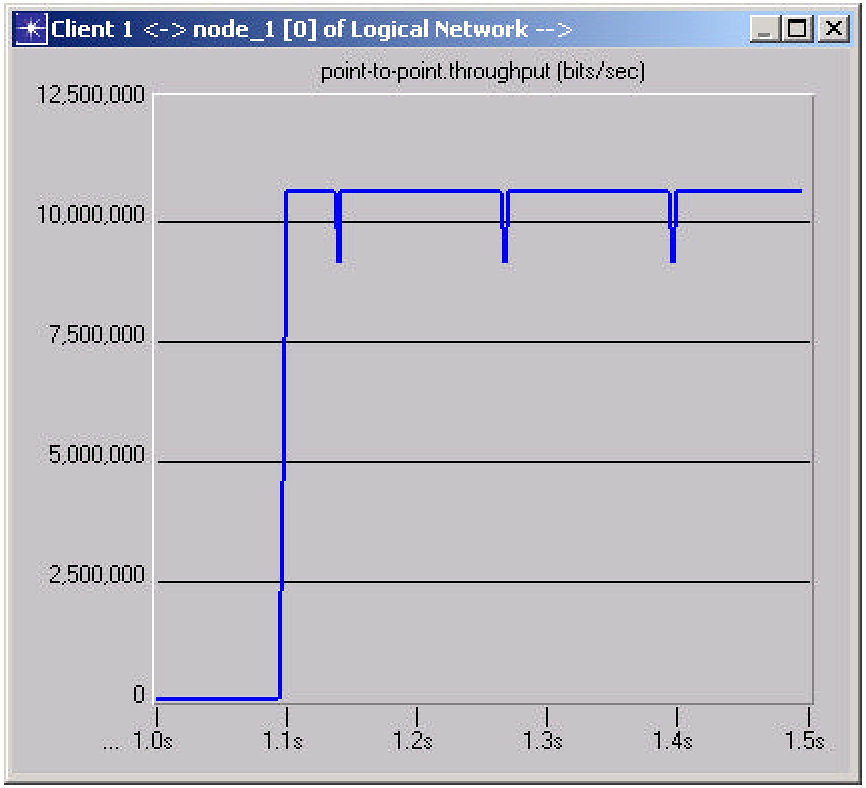
Hub Scenario



- OC12, 100km
- Client 1 sends 10Mbps CBR HP to Server 0
- Client 8 sends 750Mbps CBR LP to Server 0
- Clients are connected to RPR nodes via 10GE
- No reserved BW
- All weights equal to 1
- No rate shaping (Darwin)

Hub Scenario – Not Bursty

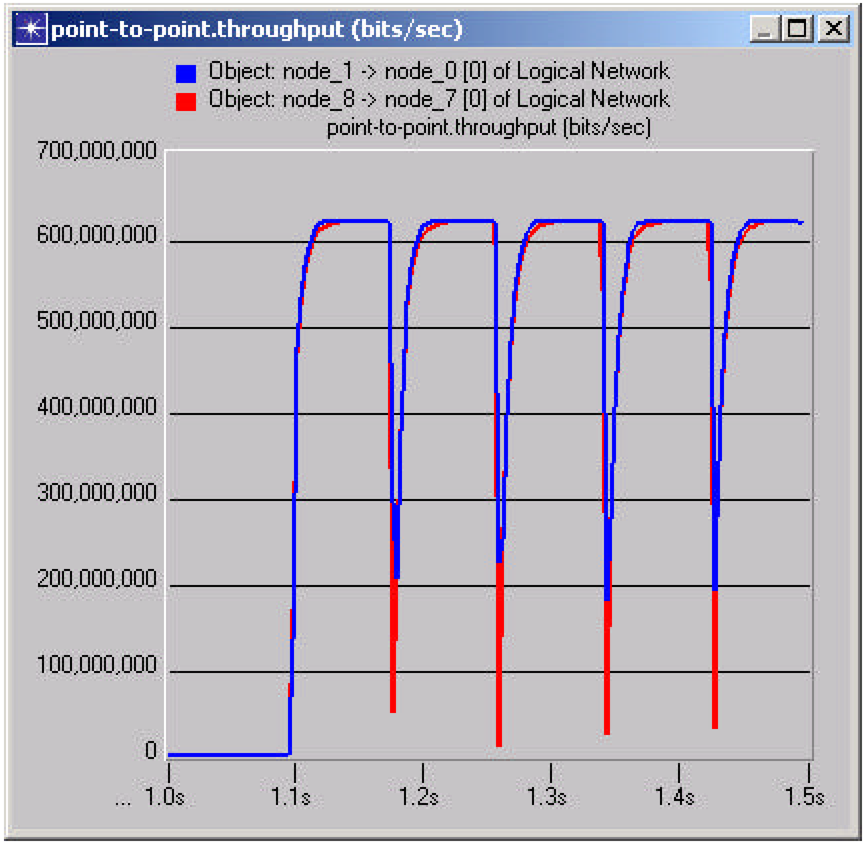




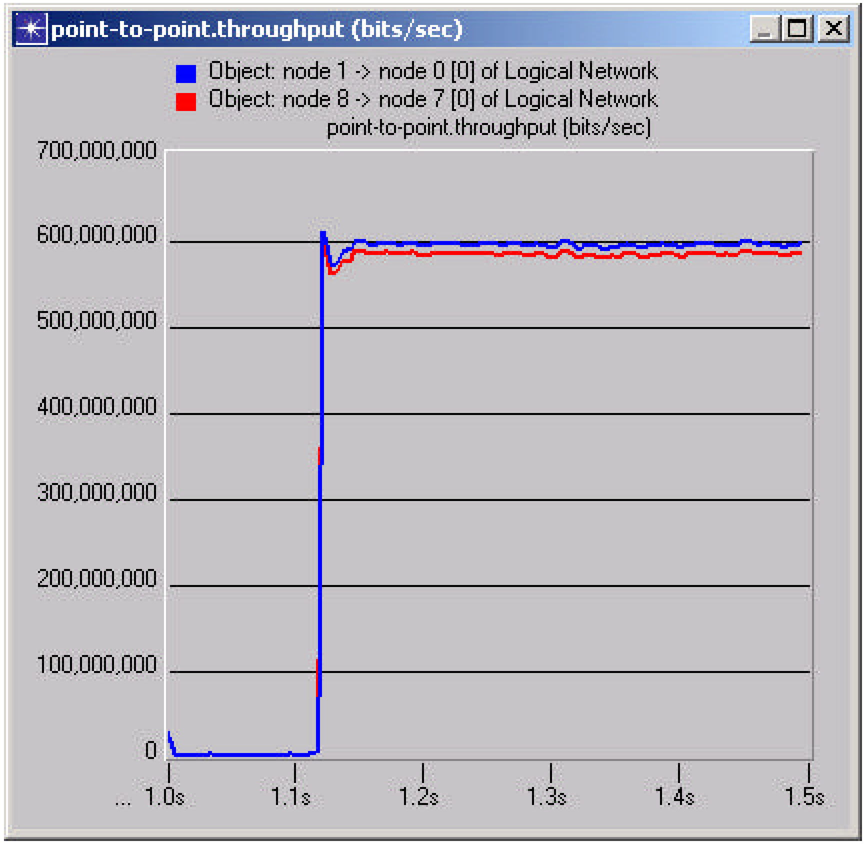
Traffic from Client to RPR Node

Hub Scenario – Not Bursty

Darwin



Alladin



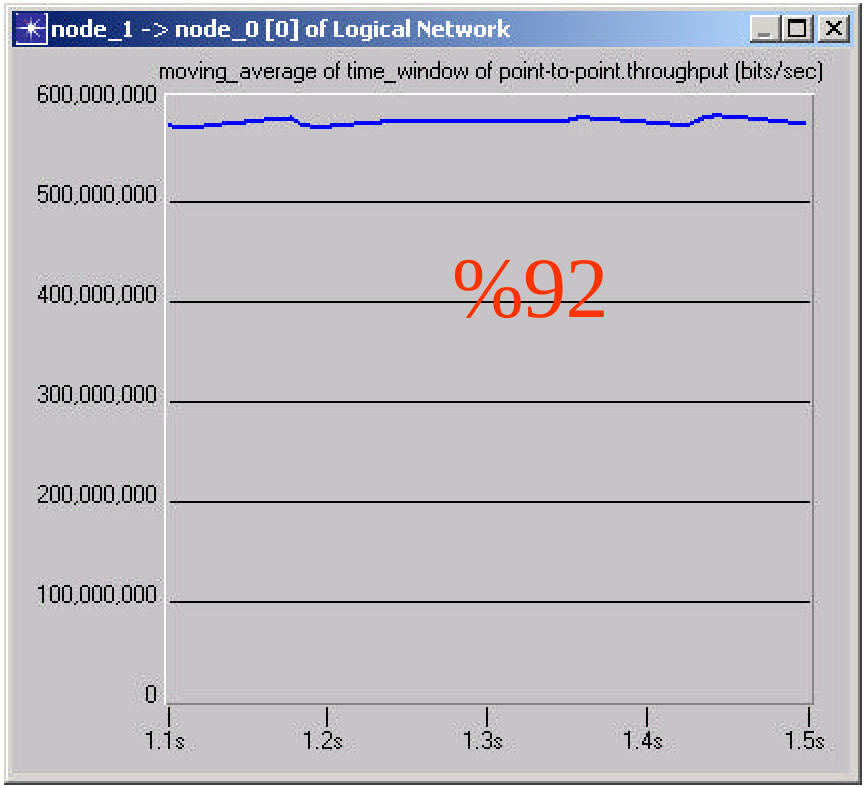
Throughput

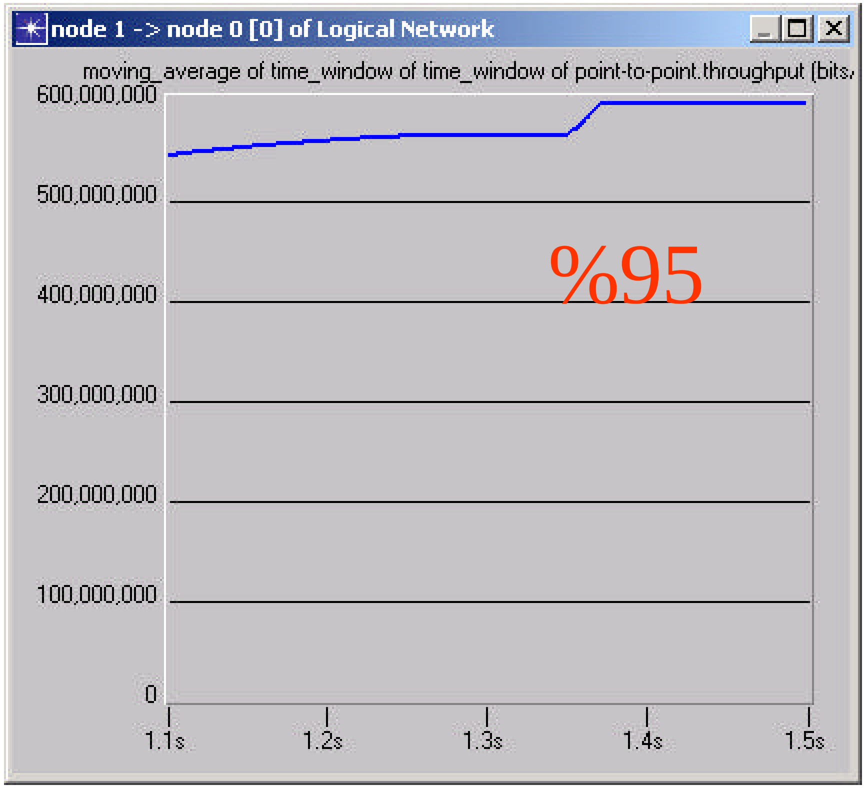
Node 8→Node 7 and Node 1→Node 0

Hub Scenario – Not Bursty

Darwin

Alladin

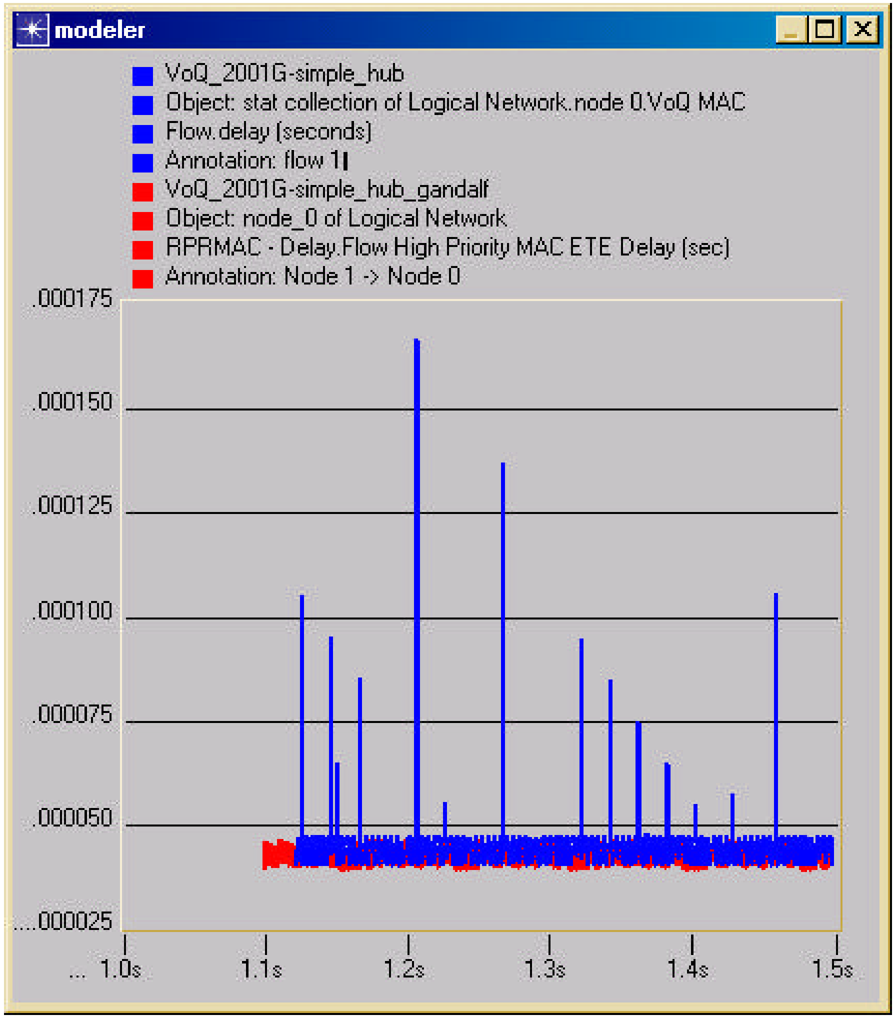


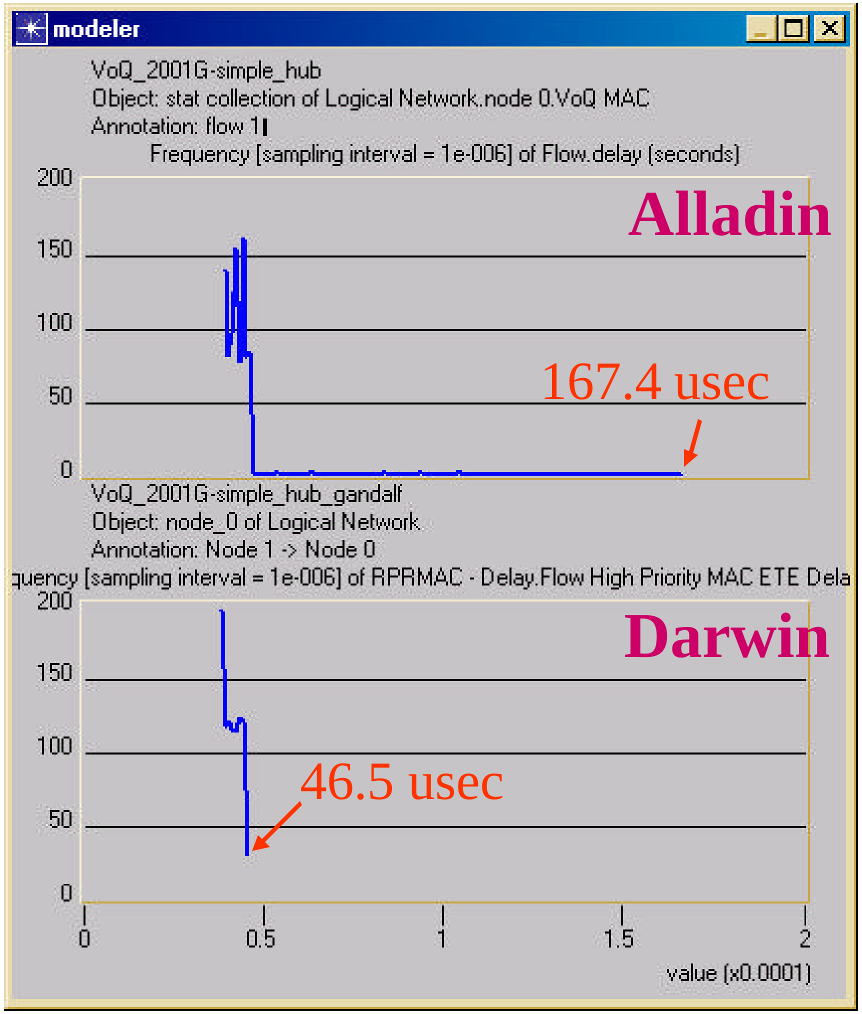


Moving Average of Throughput

Node 1→Node 0

Hub Scenario – Not Bursty

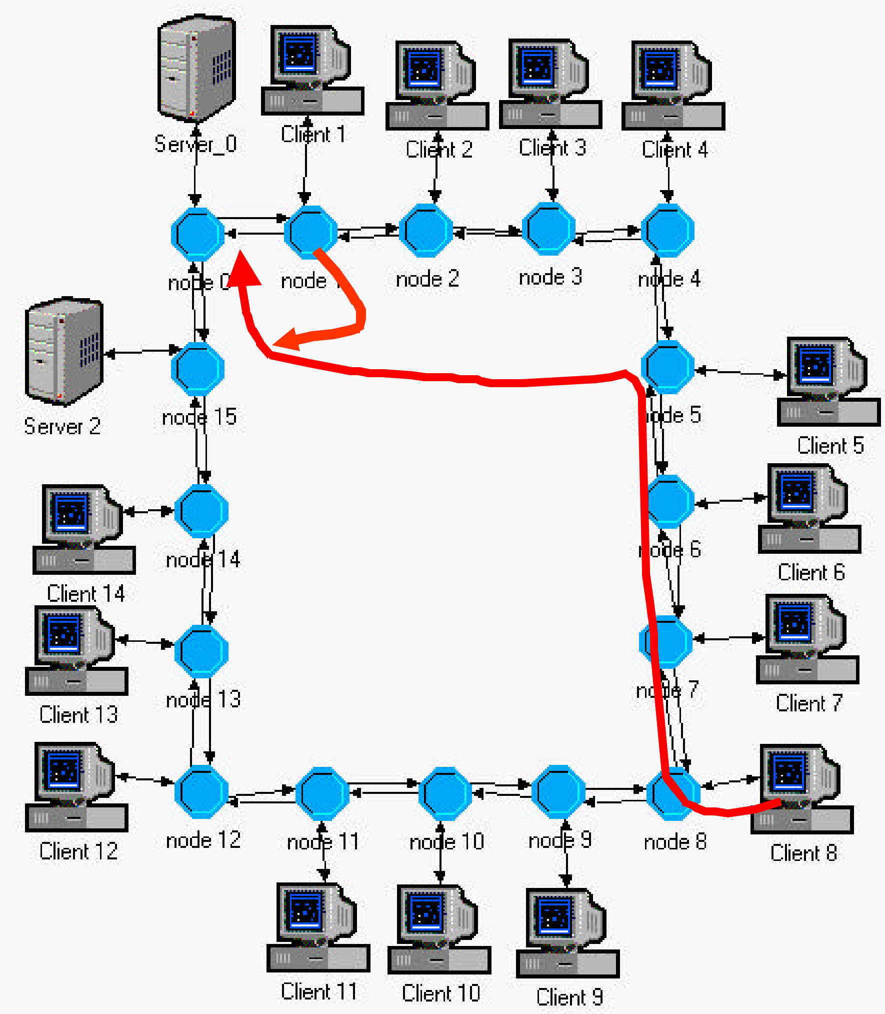




Delay Value and Histogram: Node 1→Node 0

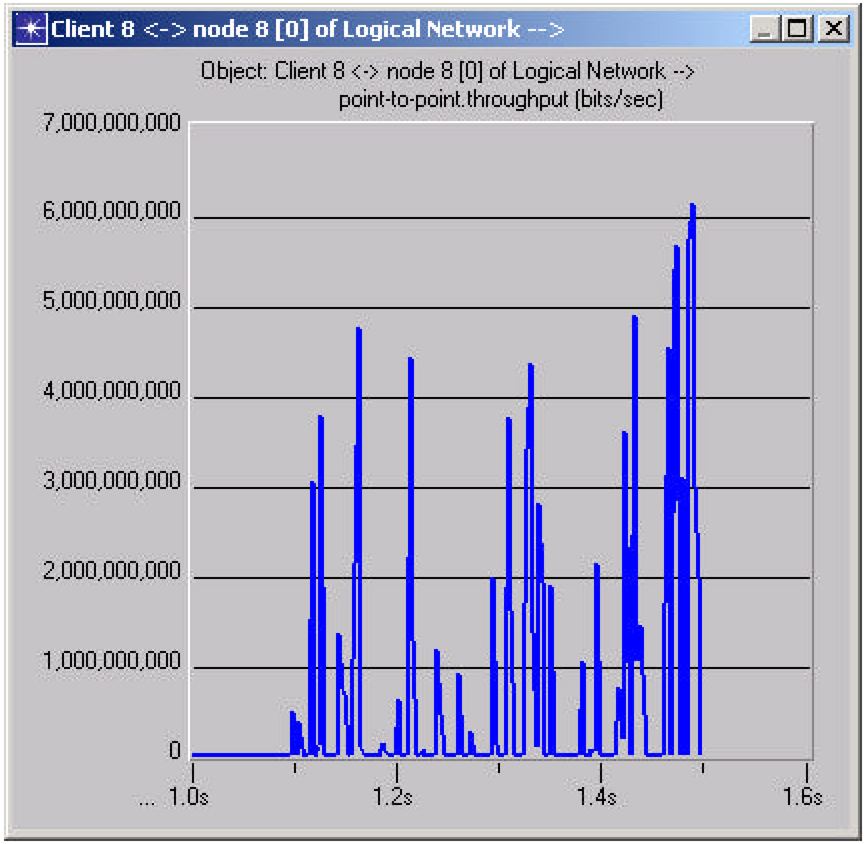
Pathological Scenario (Bursty)

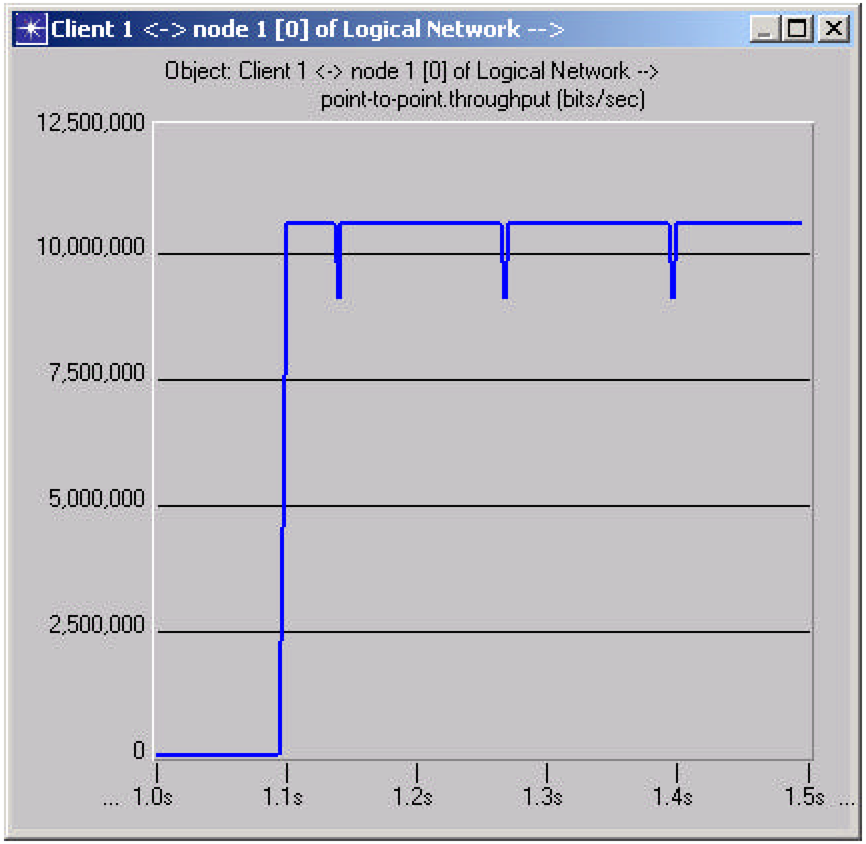
Hub Scenario



- OC12, 100km
- Client 1 sends 10Mbps CBR HP to Server 0
- Client 8 sends 750Mbps Bursty LP to Server 0
- Clients are connected to RPR nodes via 10GE
- No reserved BW
- All weights equal to 1
- No rate shaping (Darwin)

Hub Scenario –Bursty



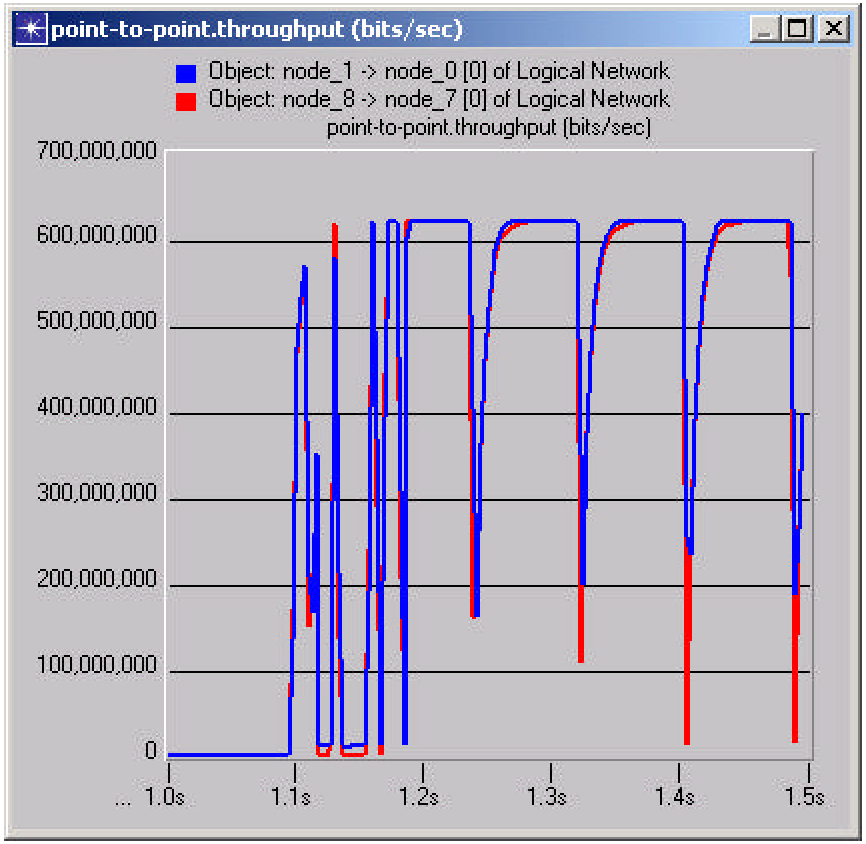


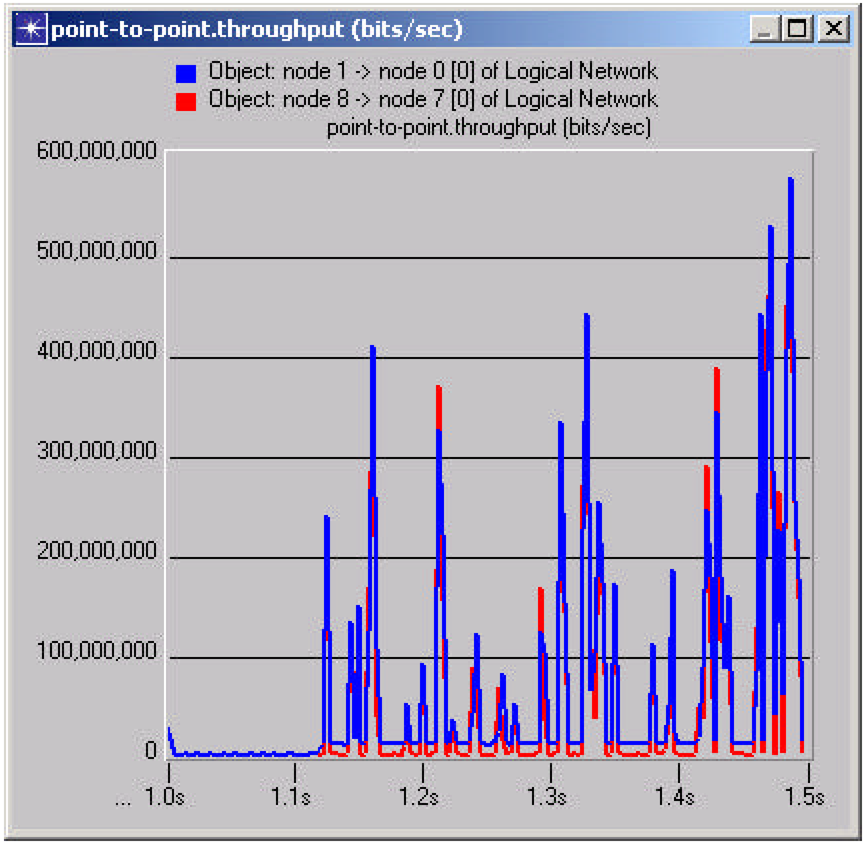
Traffic from Client to RPR Node

Hub Scenario –Bursty

Darwin

Alladin



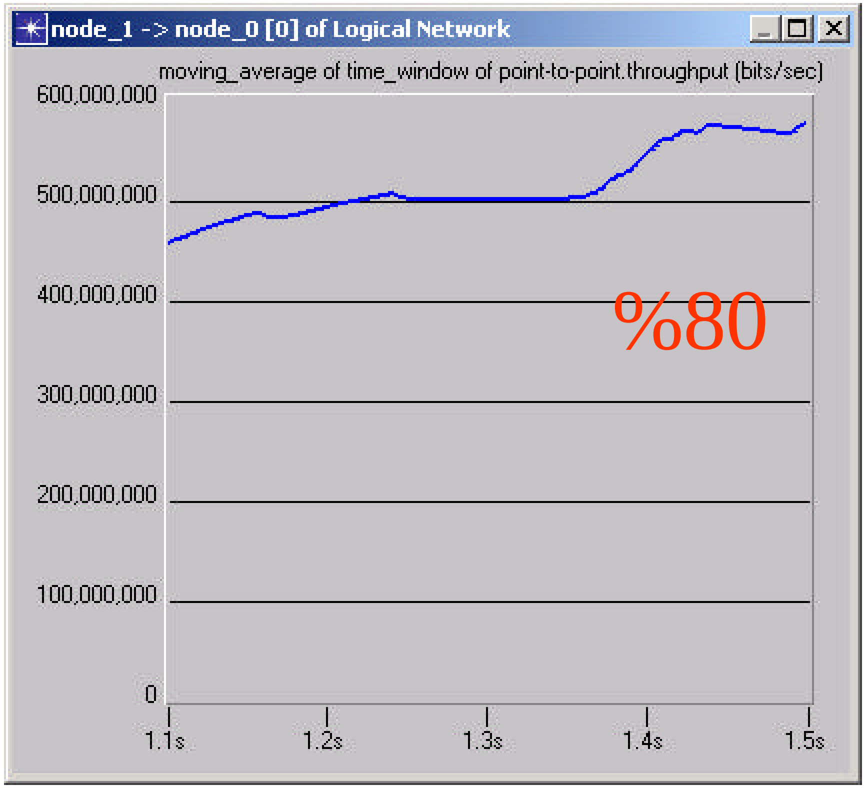


Throughput

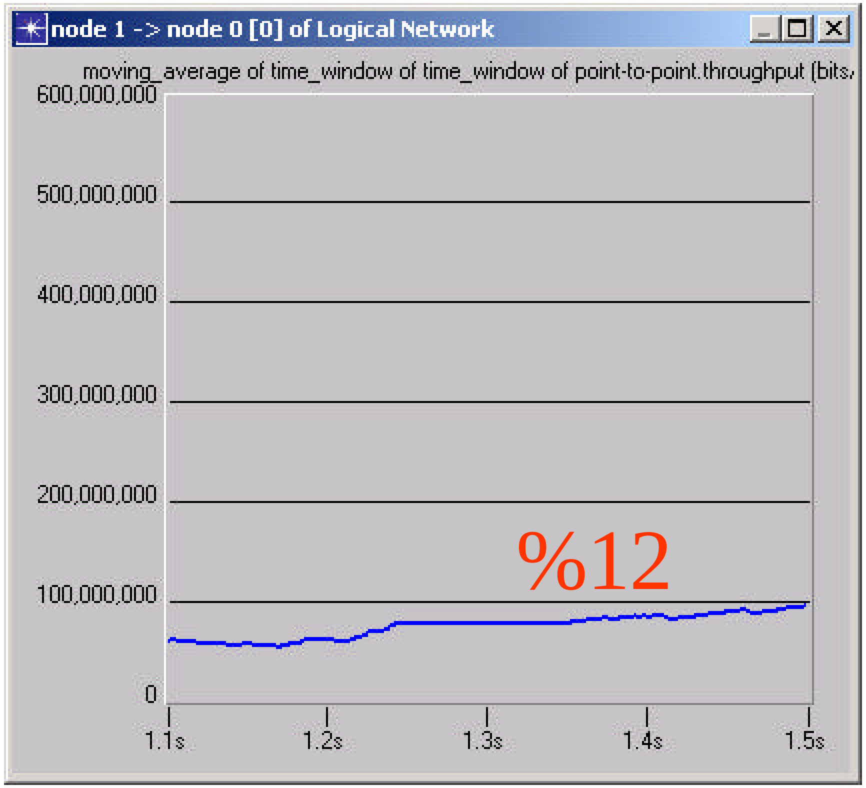
Node 8→Node 7 and Node 1→Node 0

Hub Scenario –Bursty

Darwin



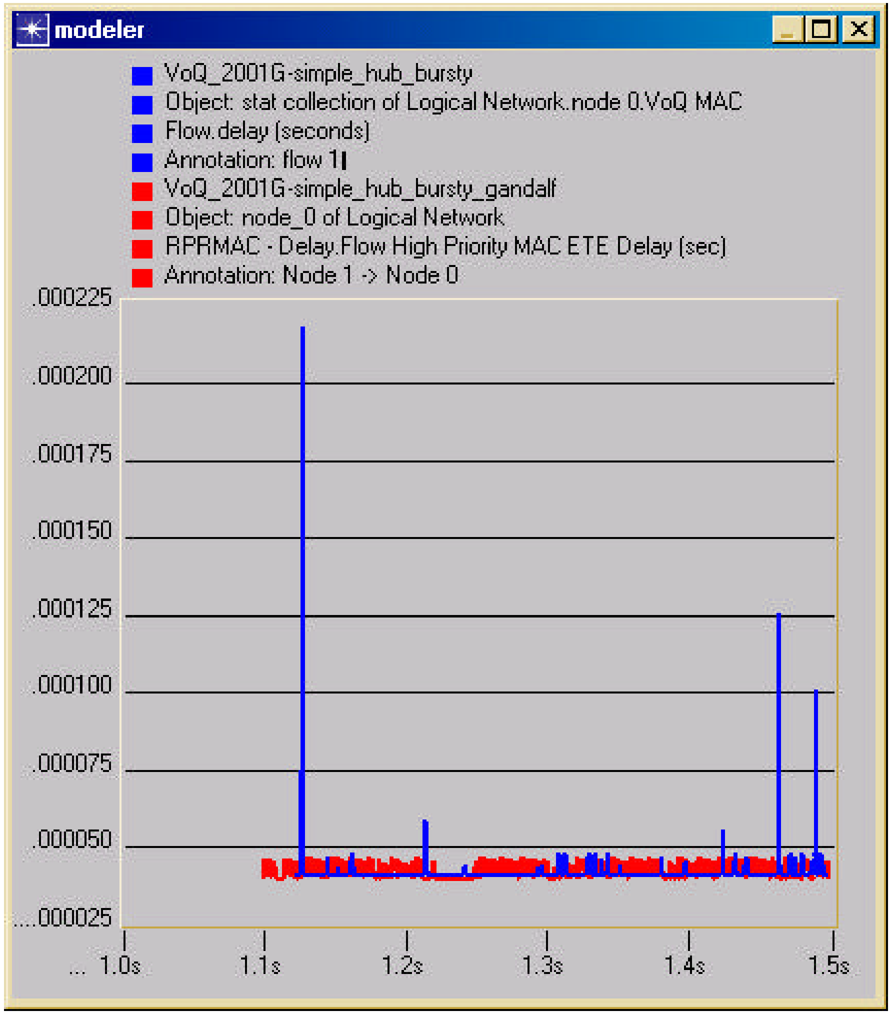
Alladin

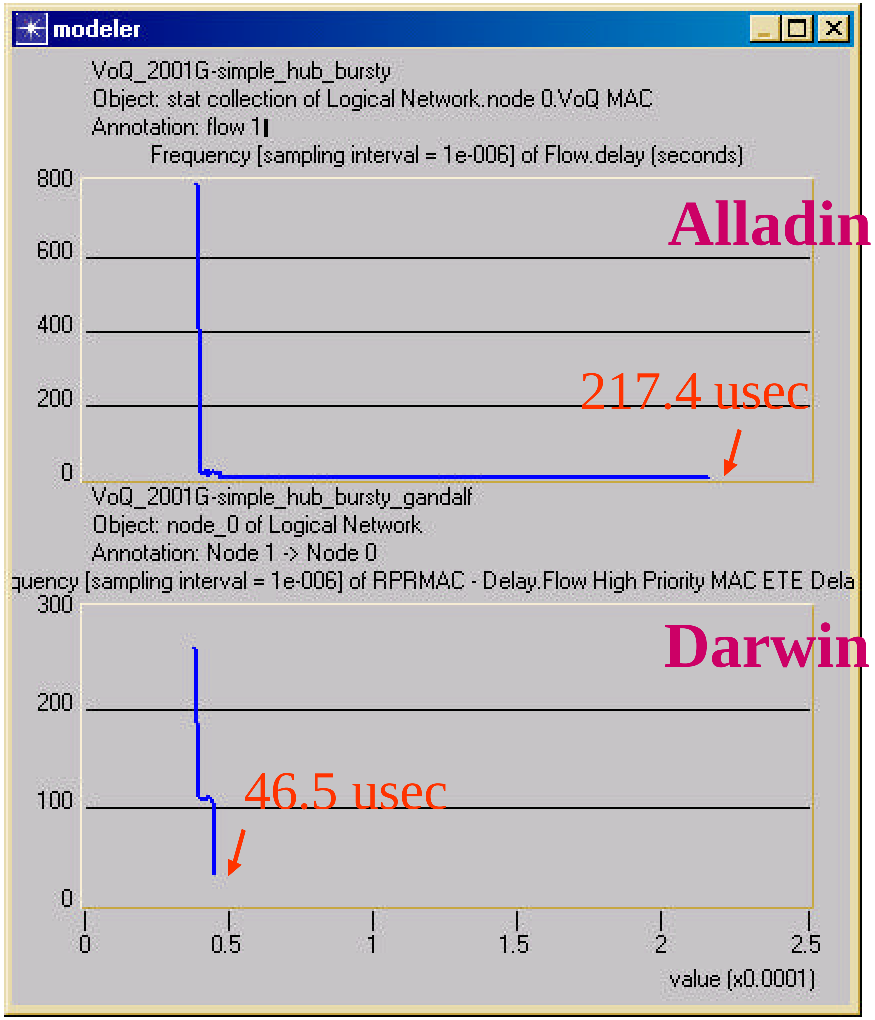


Moving Average of Throughput

Node 1→Node 0

Hub Scenario –Bursty

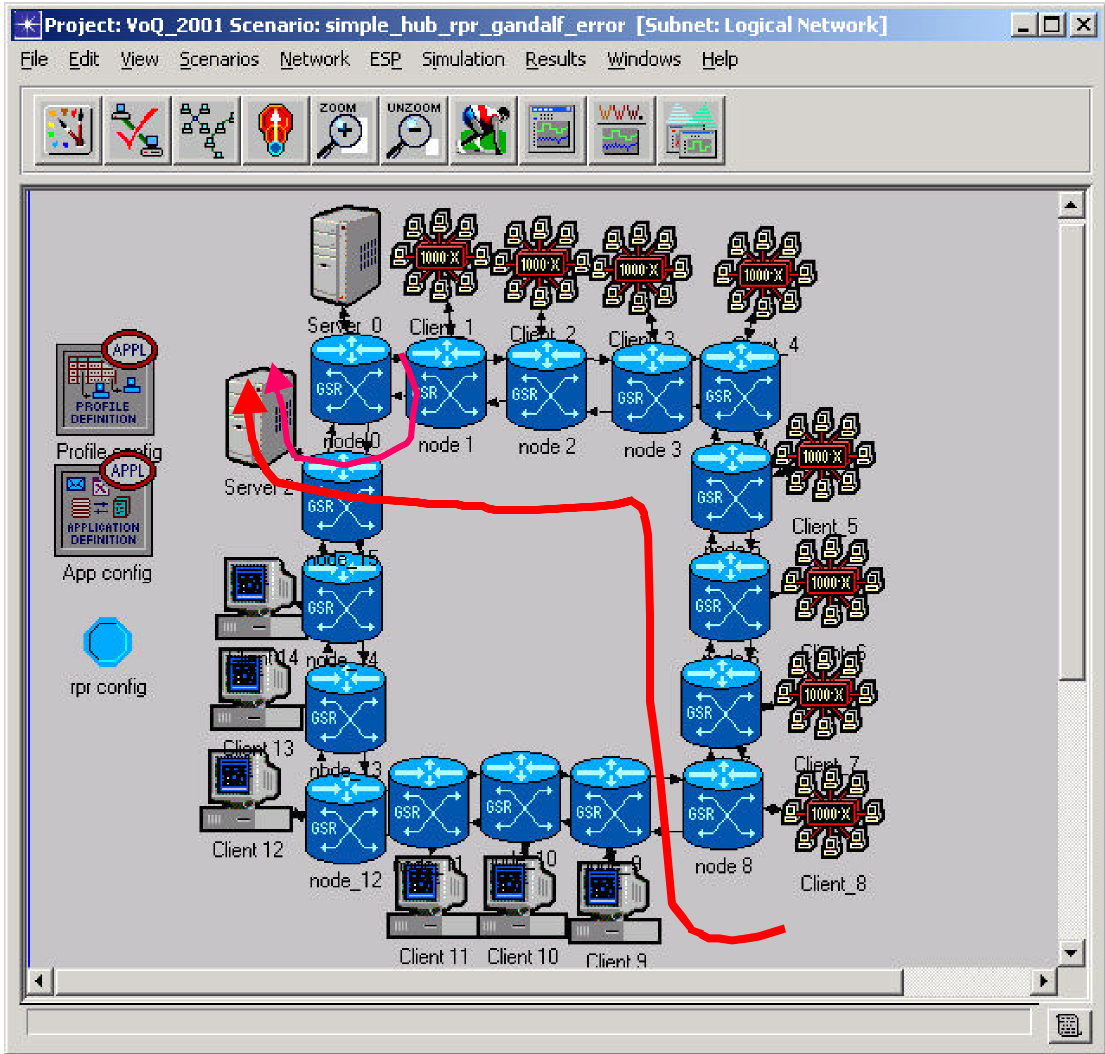




Delay Value and Histogram: Node 1→Node 0

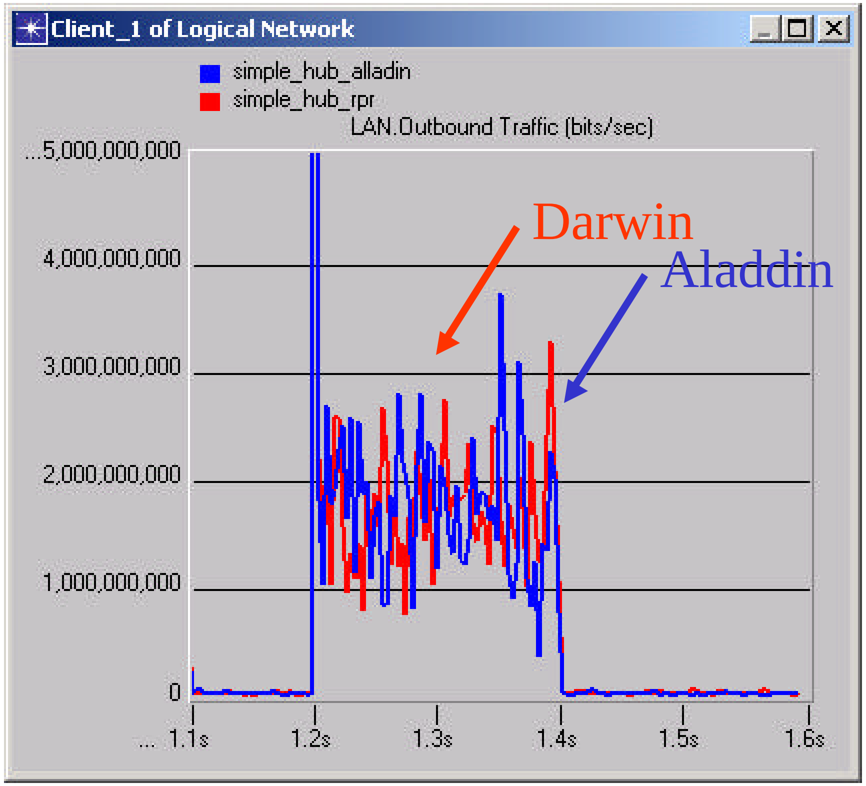
Small Traffic Node
in
Hub Configuration

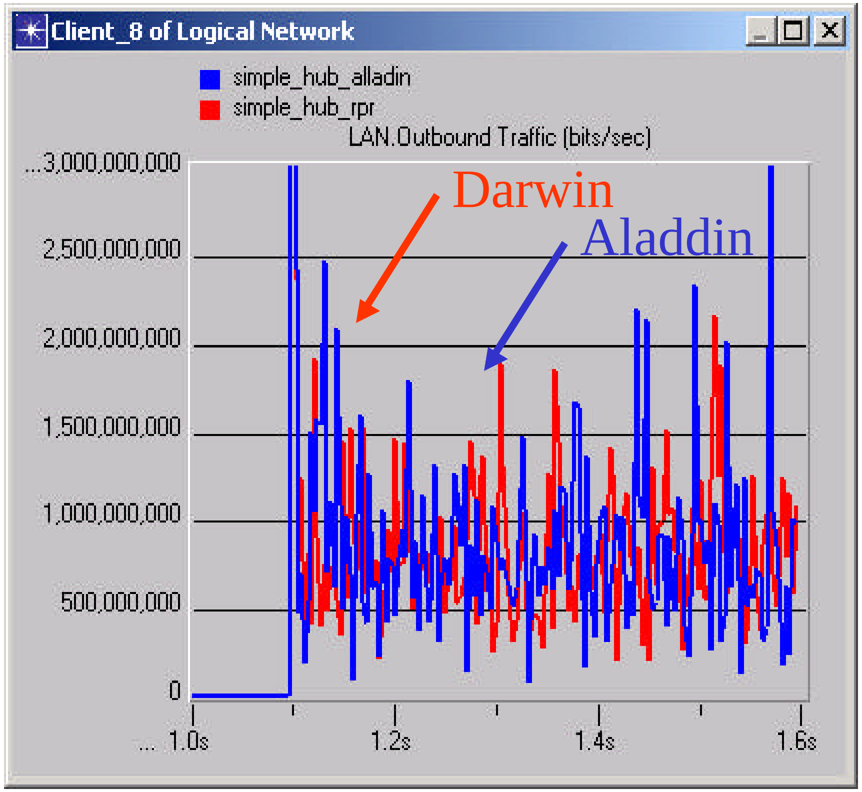
A OC12 ring with FTP/UDP Server (hub)



- Client_8 transmits 800Mbps ftp puts traffic to Server_0 along outer ring
- ftp response traffic returns from Server_0 by outer ring as well
- Client_1 sends 40Mbps first, then 1.5Gbps puts traffic to server_0
- Both Client_1 and Client_8 are 200 FTP users aggregated on a 10GE LAN
- FTP request interarrival times and file sizes are exponential
- Each ring link delay 32us

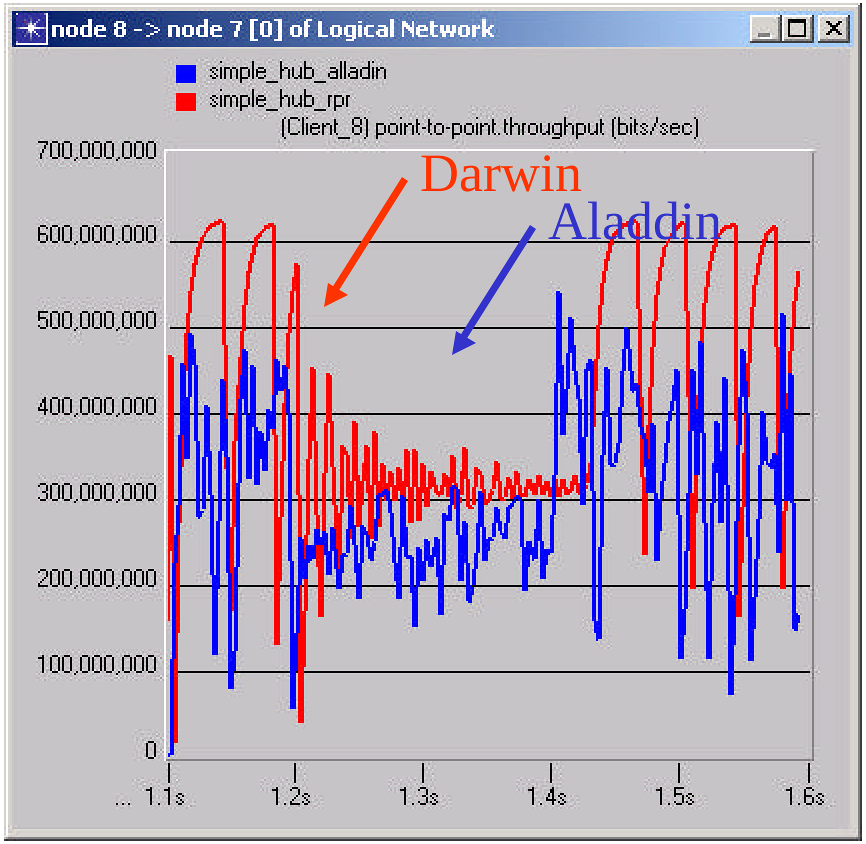
Client_1 and Client_8 LAN Traffic Sources





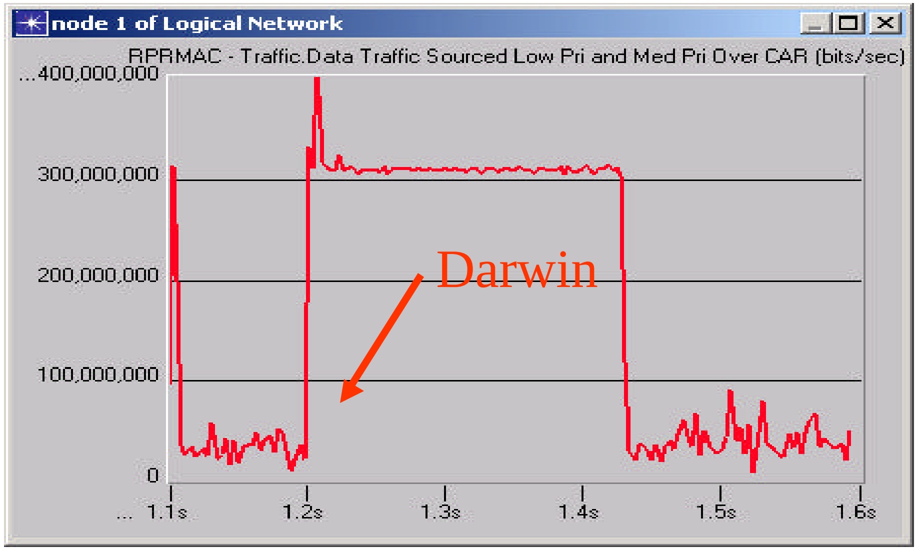
- **Aggregated LAN FTP put traffic (UDP) source**
- **There are 200 FTP users in each LAN, same traffic source configuration**
- **10GE link to ring access node**

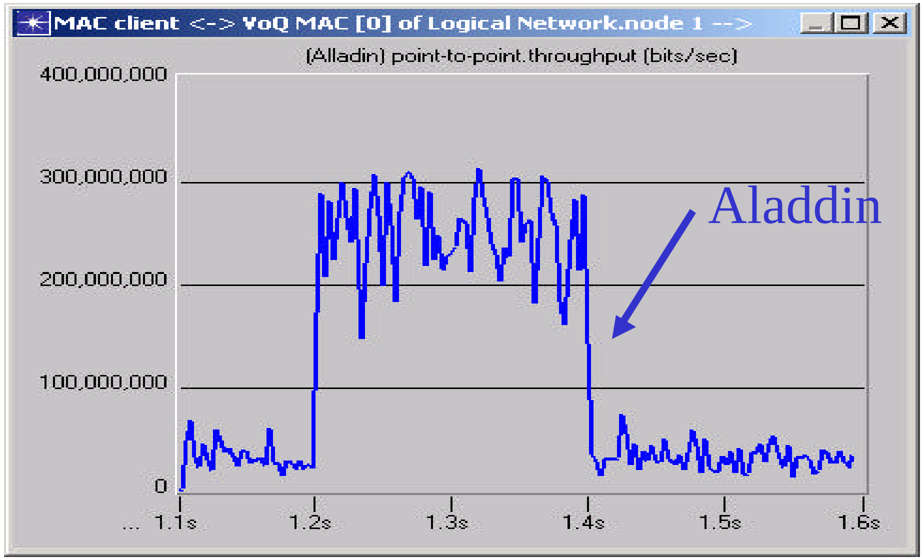
Client 8 Throughput to Ring



- **When both Client_1 and Client_8 oversubscribe the ring**
  - **Client_8 gets its fair share in Darwin**
  - **Client_8 gets 88% of its fair share in Alladin**
- **When Client_1 transmits only 40Mbps**
  - **Darwin gives Client_8 90% fair share**
  - **Alladin gives Client_8 64% fair share**

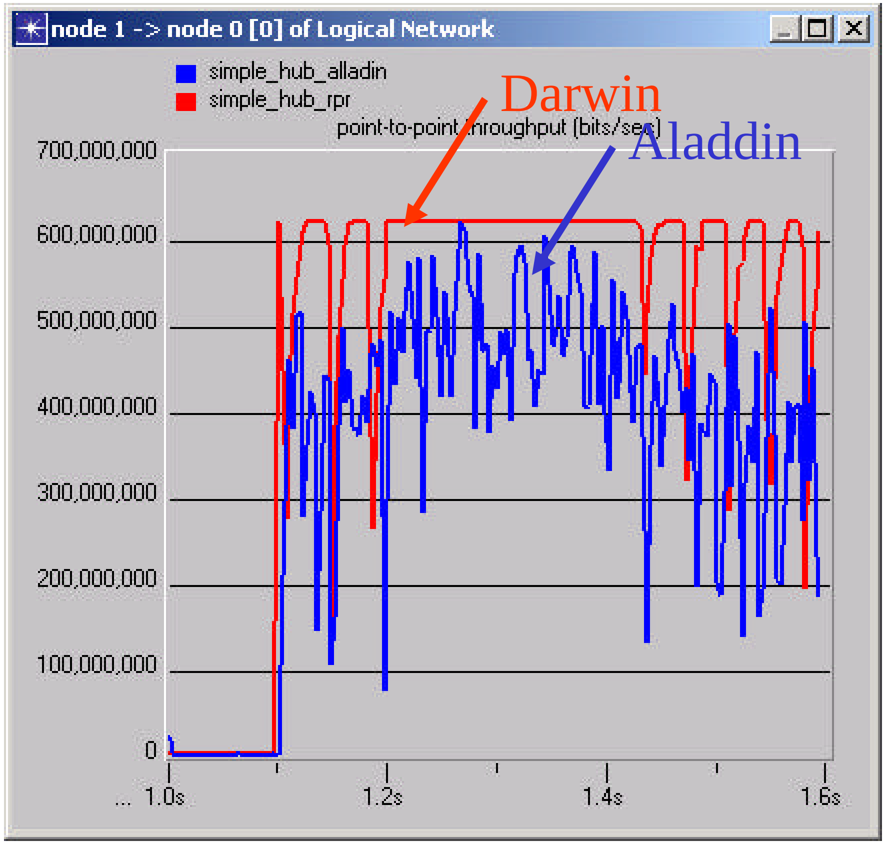
Client_1 Throughput to Ring





- **When both Client_1 and Client_8 oversubscribe the ring**
  - **Client_1 gets its fair share in Darwin**
  - **Client_1 gets 90% of its fair share in Alladin**
- **When Client_1 transmits only 40Mbps**
  - **Both Darwin and Alladin give what Client_1 wants**

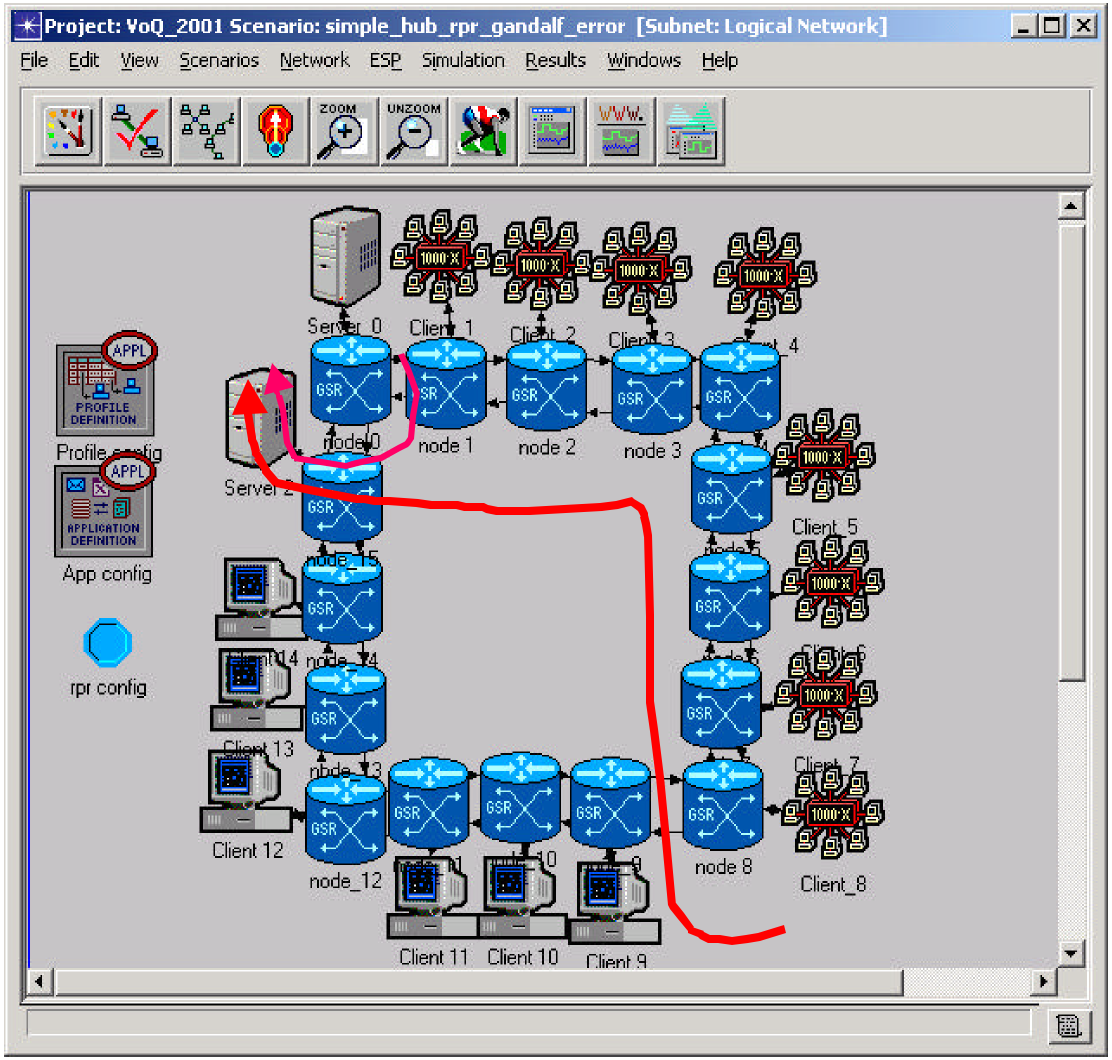
Ring Congested Link Throughput



- **When both Client_1 and Client_8 oversubscribe the ring**
  - **Darwin achieves 100% utilization**
  - **Alladin achieves less than 90% utilization**
- **When Client_1 transmits only 40Mbps**
  - **Darwin utilization is 90%**
  - **Alladin utilization is 64% or even less**

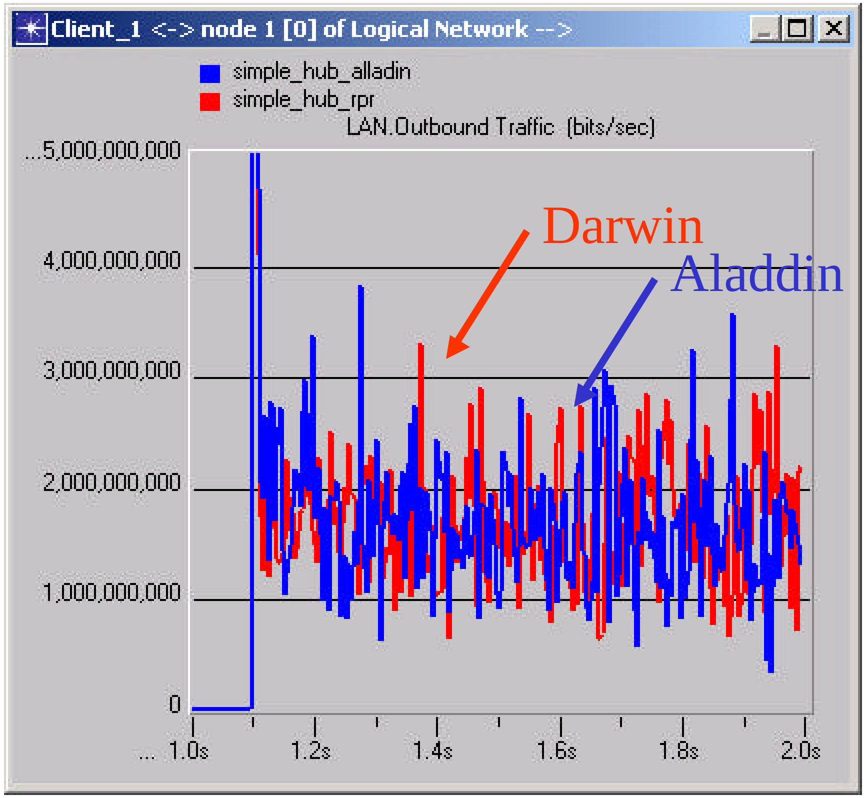
Different Oversubscription Ratios in Hub Configuration

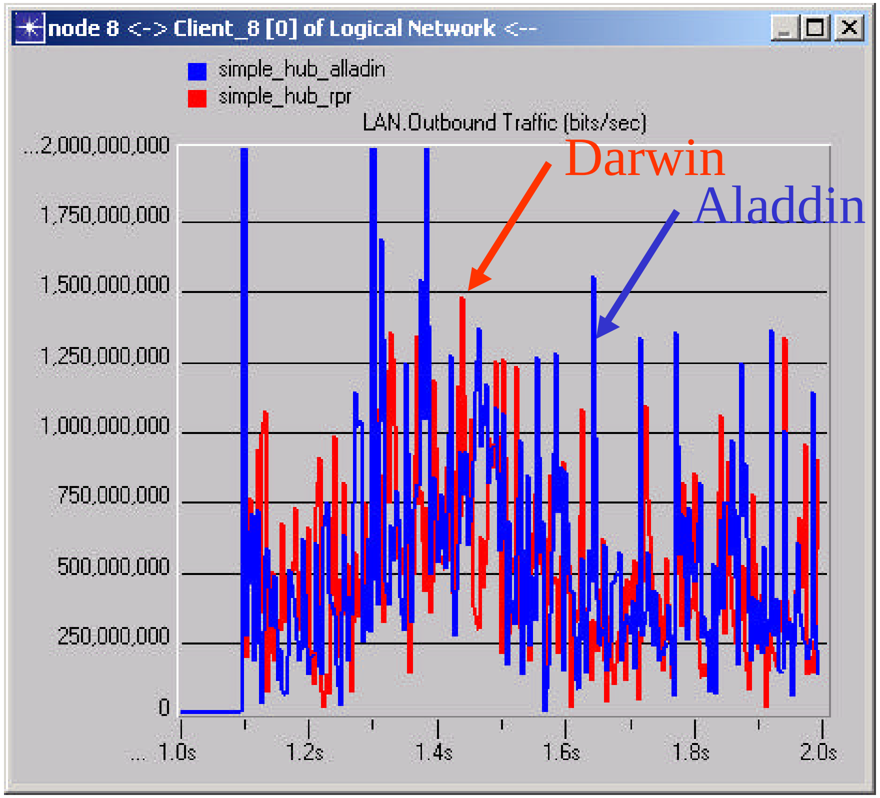
A OC12 ring with hub



- Client_8 transmits 350Mbps ~ 800Mbps ftp puts traffic to Server_0 along inner ring
- Client_1 sends 1.5Gbps puts traffic to server_0
- Both Client_1 and Client_8 are 200 FTP users aggregated on a 10GE LAN
- FTP request interarrival times and file sizes are exponential
- Each ring link delay 32us

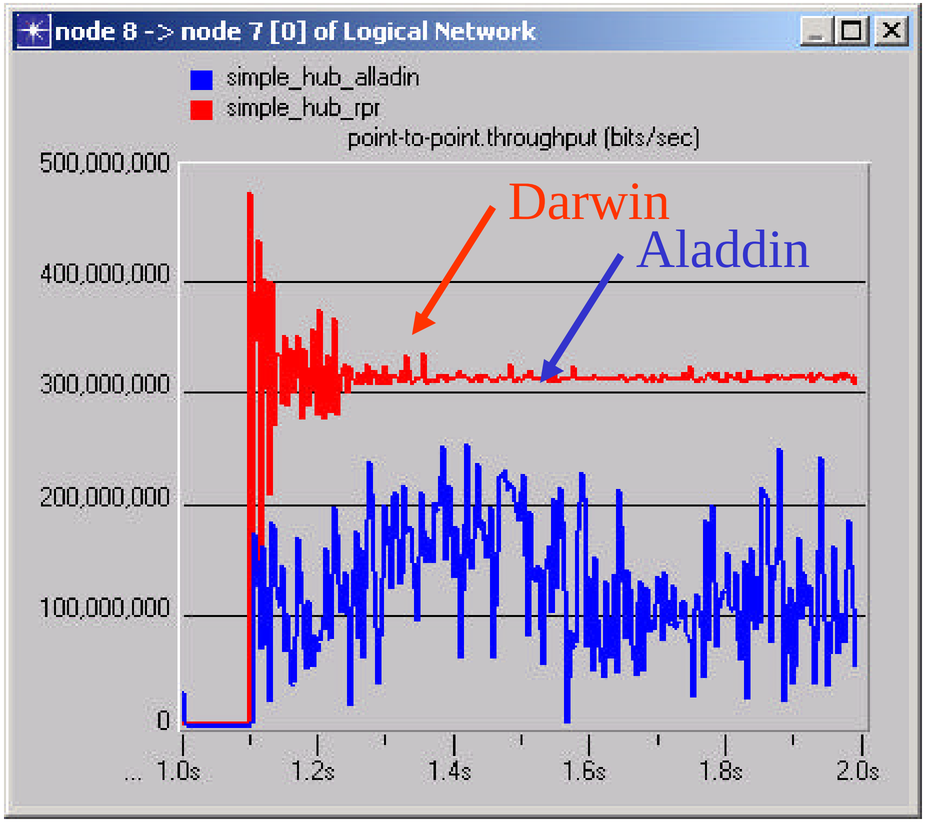
Client_1 and Client_8 LAN Traffic Sources





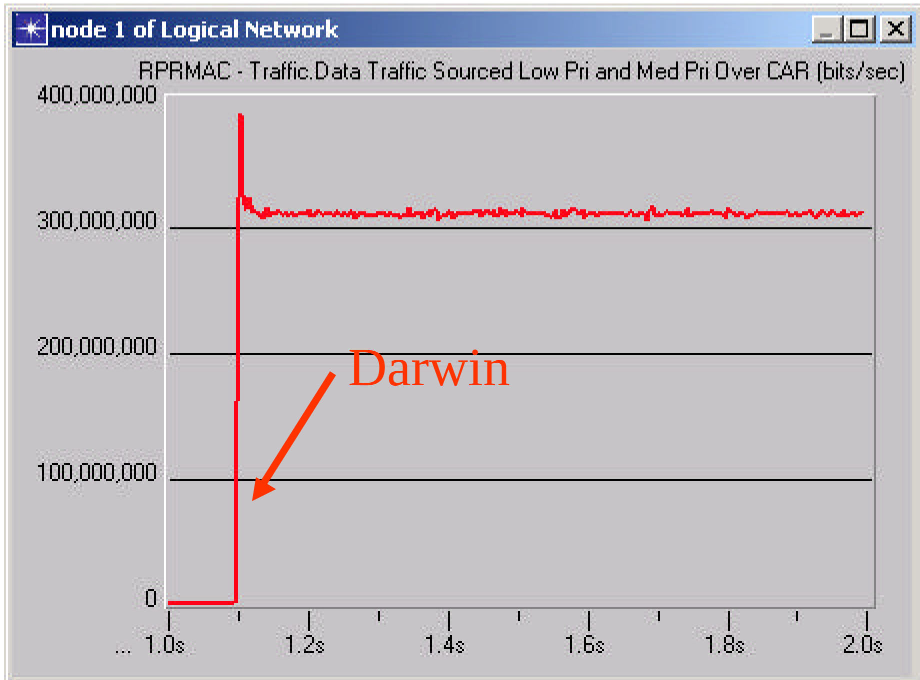
- **Aggregated LAN FTP put traffic (UDP) source**
- **There are 200 FTP users in each LAN**
- **Darwin and Alladin share the same traffic source configuration**

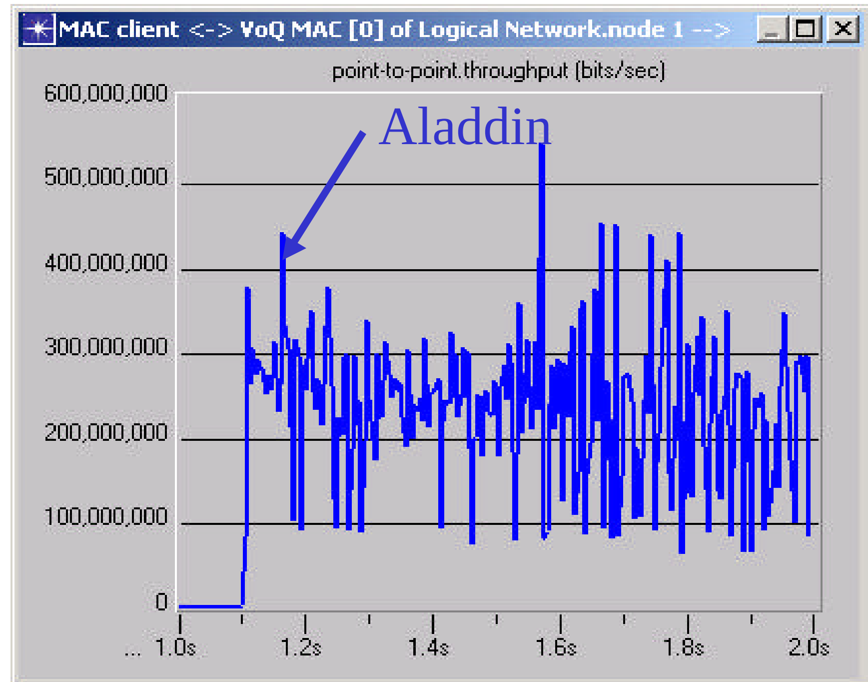
Client_8 Throughput to Ring



- **Both Client_1 and Client_8 oversubscribe the ring**
- **Darwin allows Client_8 to use up its fair share of ring bandwidth**
- **Alladin only gives 50% or even less of fair ring bandwidth to Client_8**

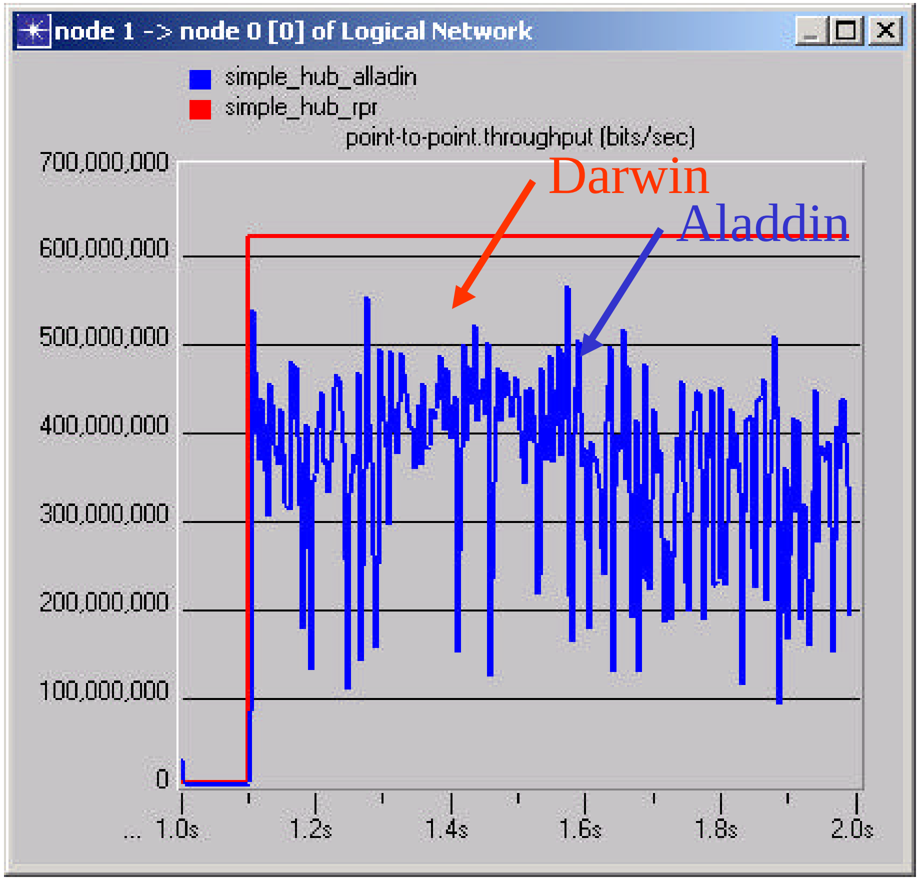
Client_1 Throughput to Ring





- **When both Client_1 and Client_8 oversubscribe the ring**
  - **Client_1 gets its fair share in Darwin**
  - **Client_1 gets 80% of its fair share in Alladin on average**
  - **There are no fairness in Alladin between Client_1 and Client_8**

Ring Congested Link Throughput



- **When both Client_1 and Client_8 oversubscribe the ring**
- **Darwin achieves 100% utilization**
- **Alladin achieves less than 65% or less utilization**

Things to consider

There will always be cases to cause closed-loop feedback algorithms to underperform

Conclusion

- Under bursty traffic (dominating characteristic of Internet), Darwin performed well
- Darwin exhibits superior performance while optimizing delay/jitter and throughput
- Darwin can react faster in bursty conditions