Simulate Real-World IP Networks

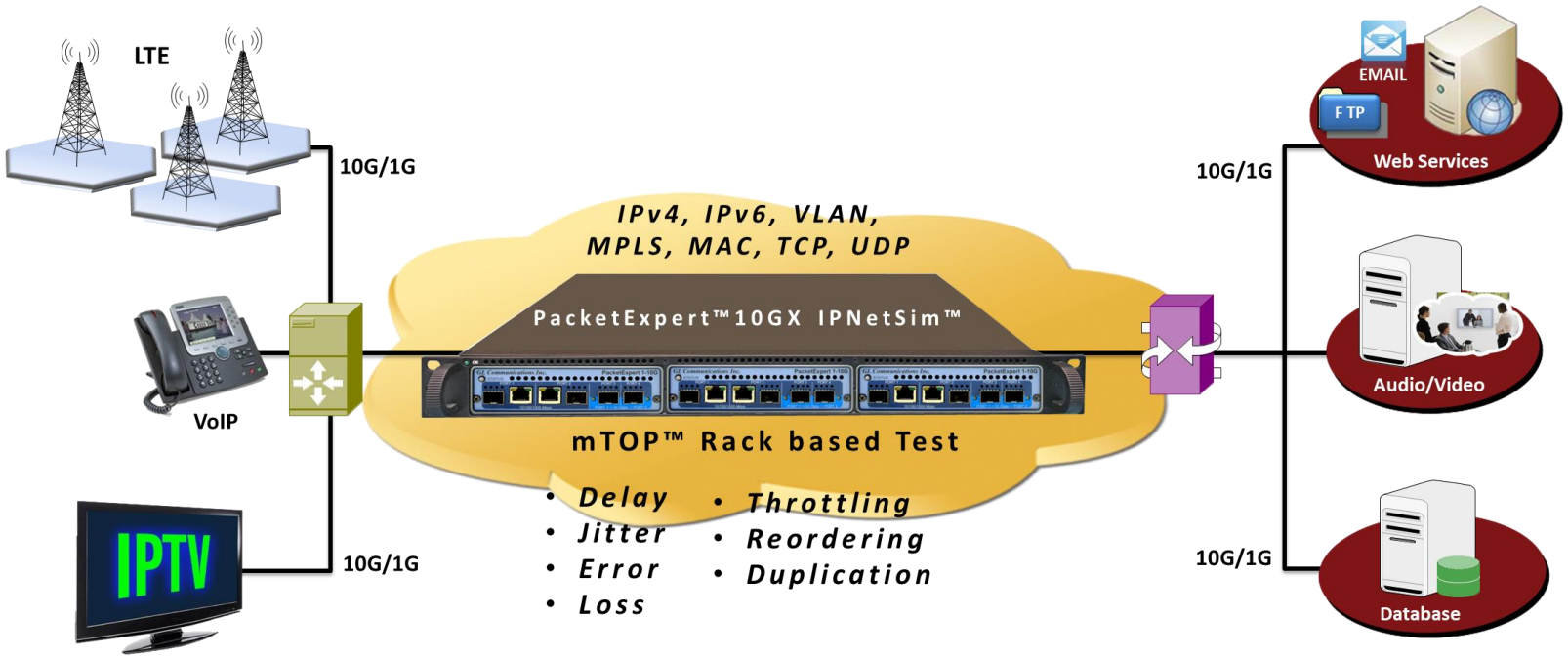
Impairments, Delay, Errors, Loss, Optical, Electrical

GL Communications Inc.
818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878
Phone: (301) 670-4784 Fax: (301) 670-9187 Email: info@gl.com
Website: https://www.gl.com

Overview

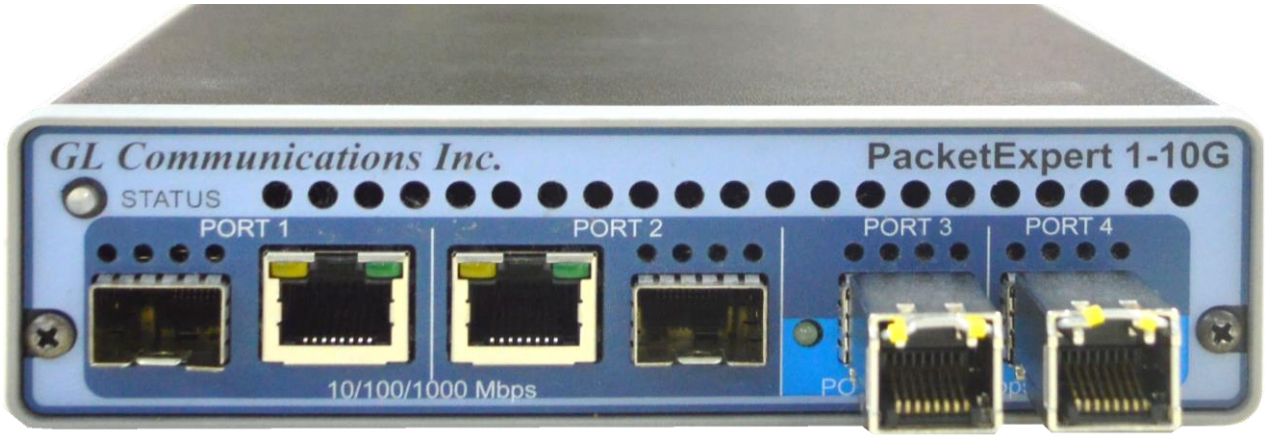
How does GL simulate real-world IP Networks? What is GL's IPNetSim?

- Lab Testing Solution - application and automation
- Emulate Full Duplex 1 Gbps and 10 Gbps networks
- Real-world network conditions by imposing impairments
- Multiple streams independently configured



GL Communications

Portable Unit



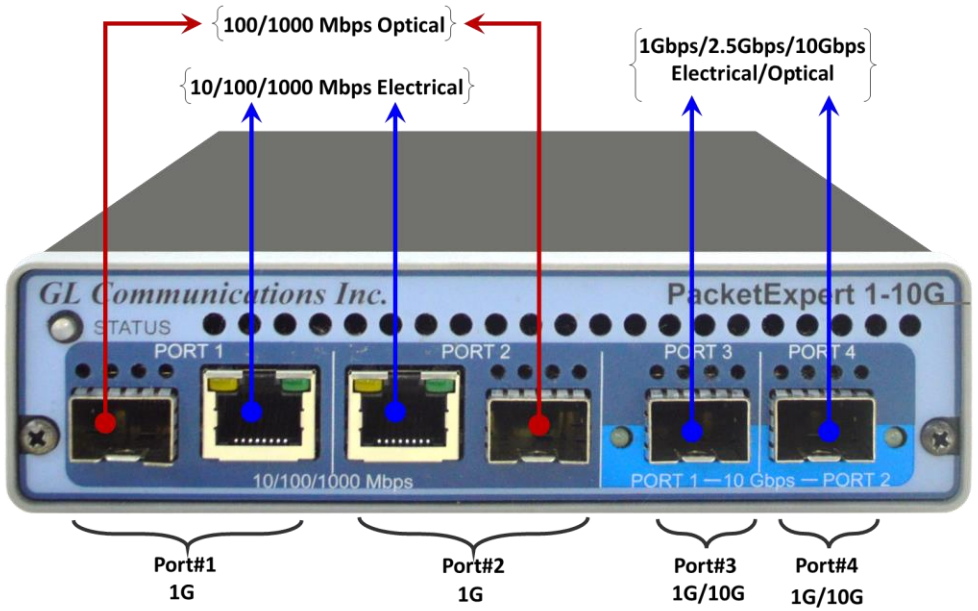
PacketExpert™ 10GX Standalone

- 4 x 1 Gbps Optical or Electrical
- 2 x 10 Gbps /2.5 Gbps Optical or Electrical

GL Communications

Hardware Overview

Hardware Overview + Highlights

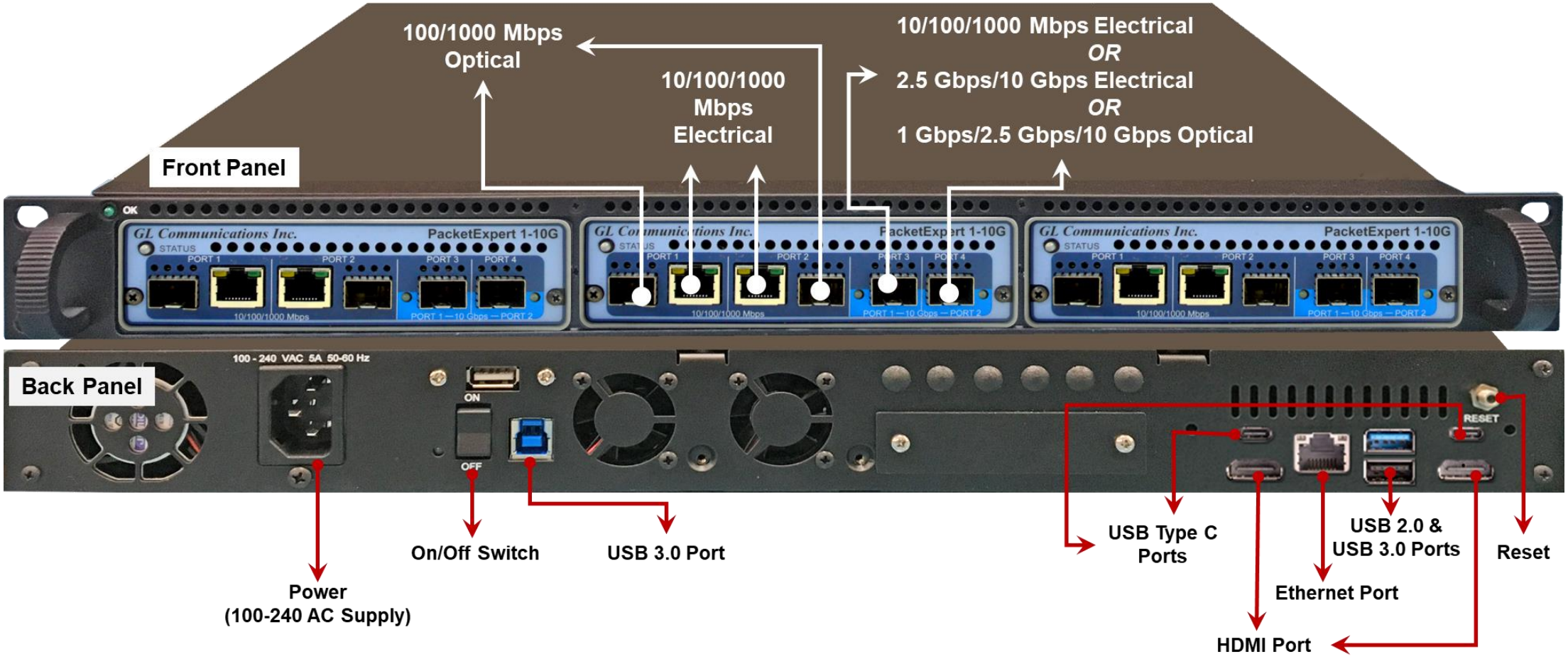


IPNetSim™ is an optional application available within PacketExpert™ platform (PXN100).

- IPNetSim operates in both multi-stream and single stream mode.
- IPNetSim acts as a bridge between two network segments. As long as the hardware has power it allows frames to flow freely.
- IPNetSim allows users to define up to 16 different streams of traffic. Each of these streams can have its own independent set of impairments applied to them. More to come on streams and exactly how GL defines them.
- IPNetSim is hardware-based...meaning all impairments and timing controls happen at the hardware level.

GL Communications

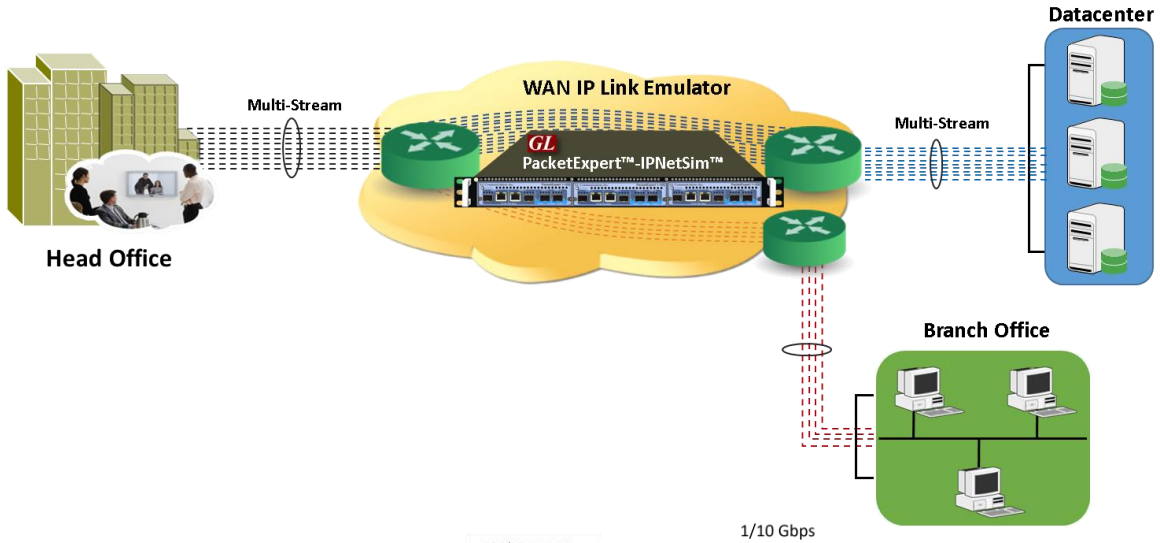
Hardware Interfaces

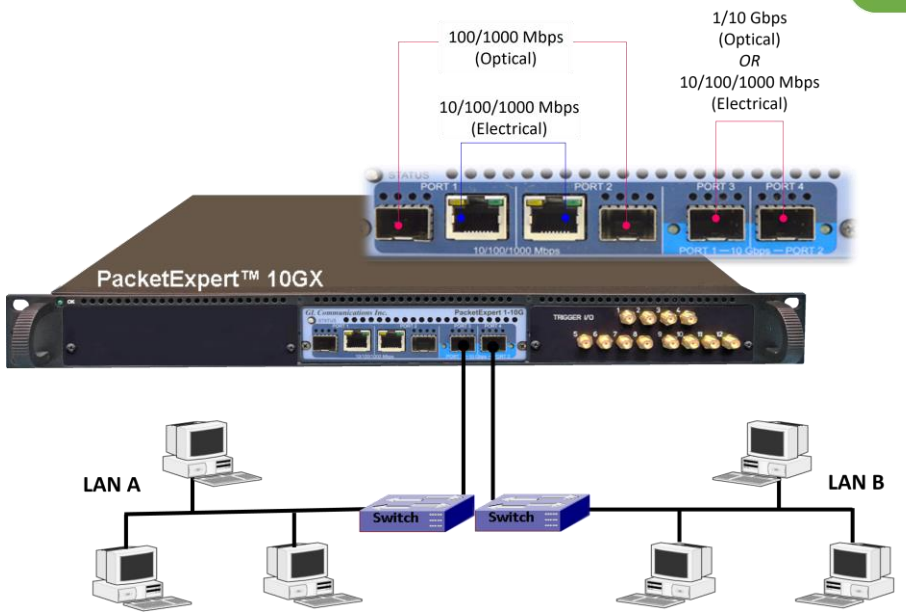


Application Overview

Application Overview

- Test Enterprise and Individual-level applications
  - Audio and video streaming (VoIP, IMS, HDT, IPTV)
  - Storage services (Critical Data Access)
  - Cloud and web services
  - FTP / HTTP
- Simulate backhaul network
  - Static and dynamic networks
  - Satellite + other long delay networks
- Test Quality of Service (QoS) and Quality of Experience (QoE)
- Evaluate the stability of network devices (switches, VoIP Phones, VoIP PBXs, Set-top boxes and VoD Servers.

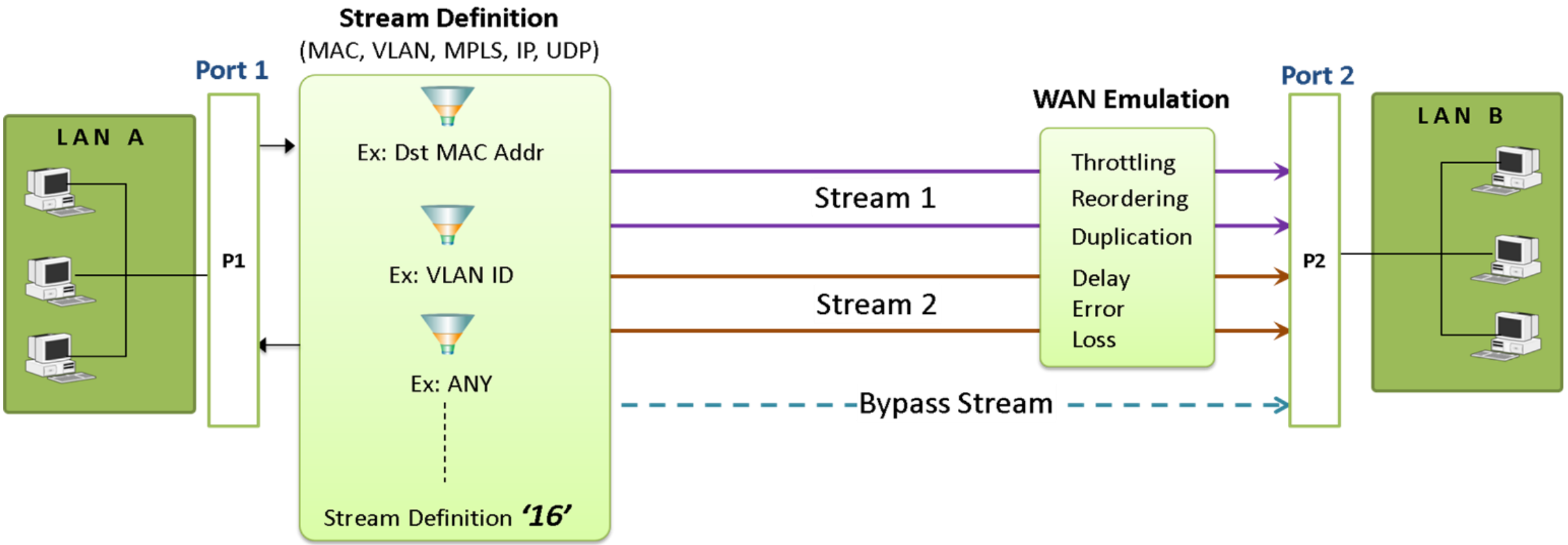




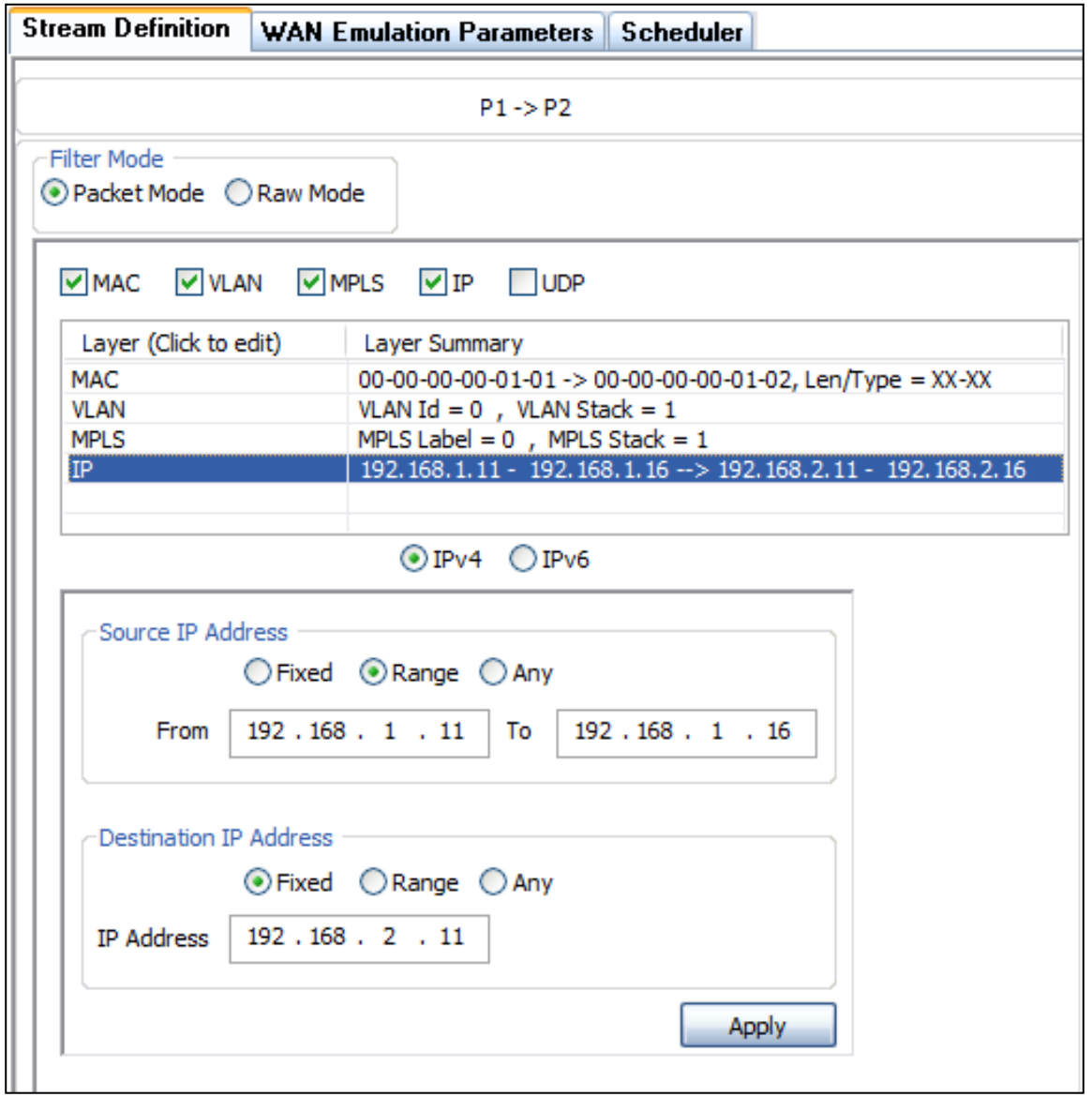
GL Communications

Stream Overview

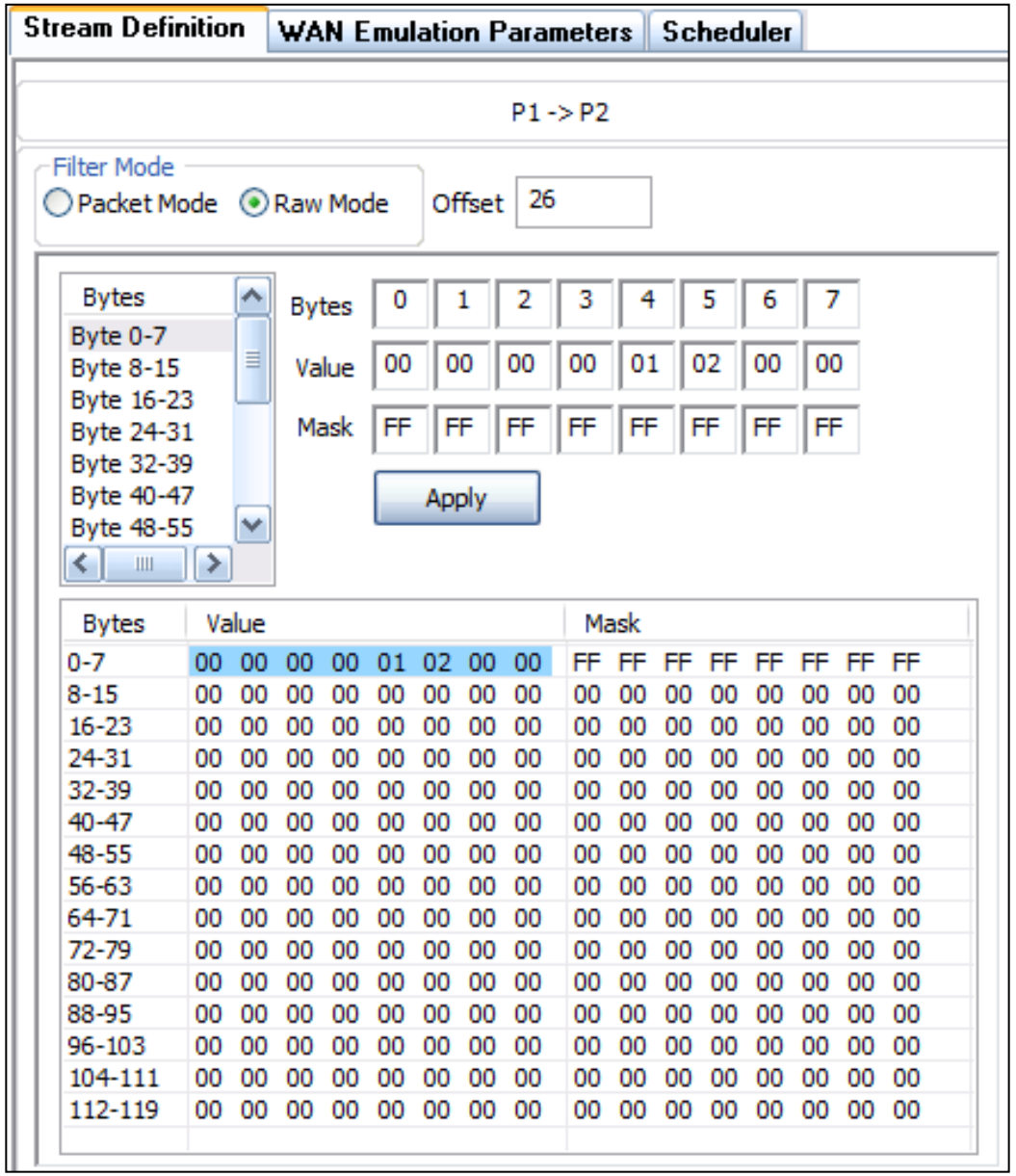
Stream Overview



Define Streams in Packet Mode



Define Streams in Raw Mode

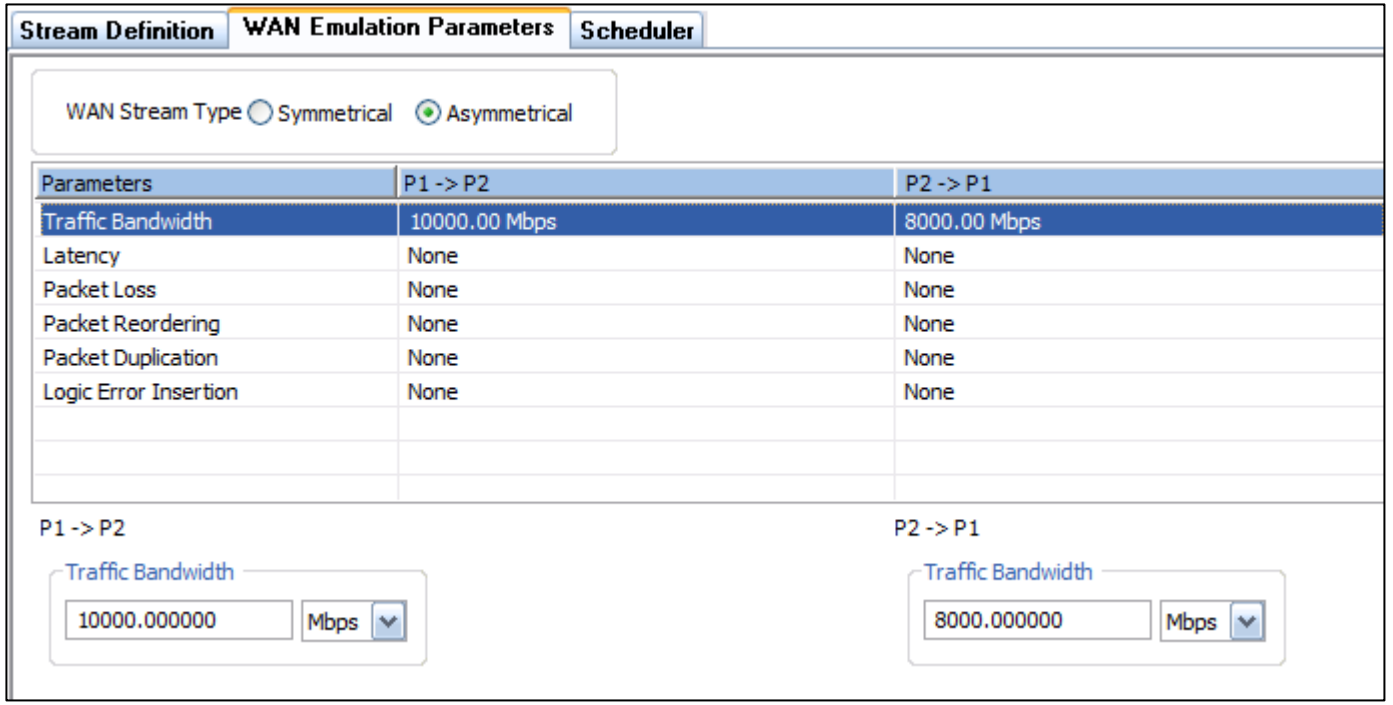


GL Communications

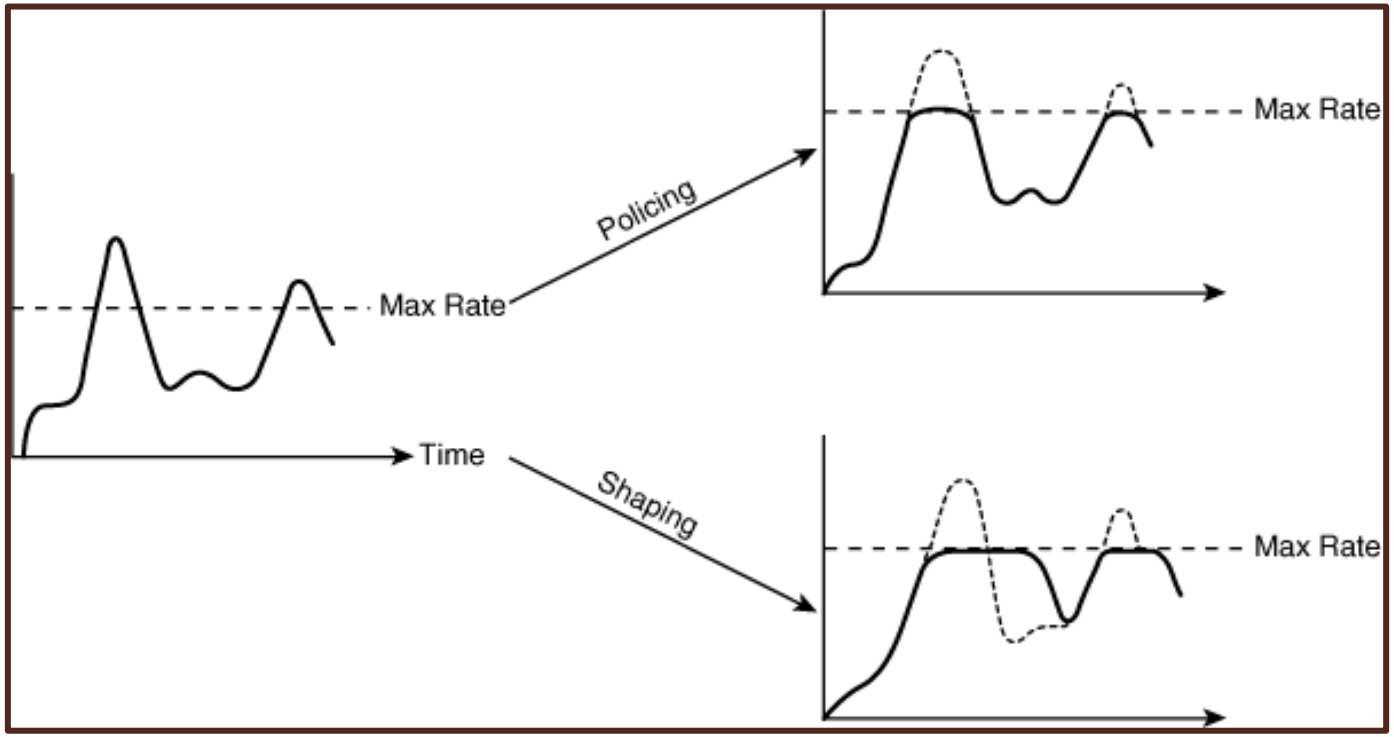
Impairments

Traffic Bandwidth

- Traffic which exceeds the stated rate is **silently dropped**
- UDP Applications will experience **data loss**
- TCP Applications should adapt via **congestion-avoidance algorithms**

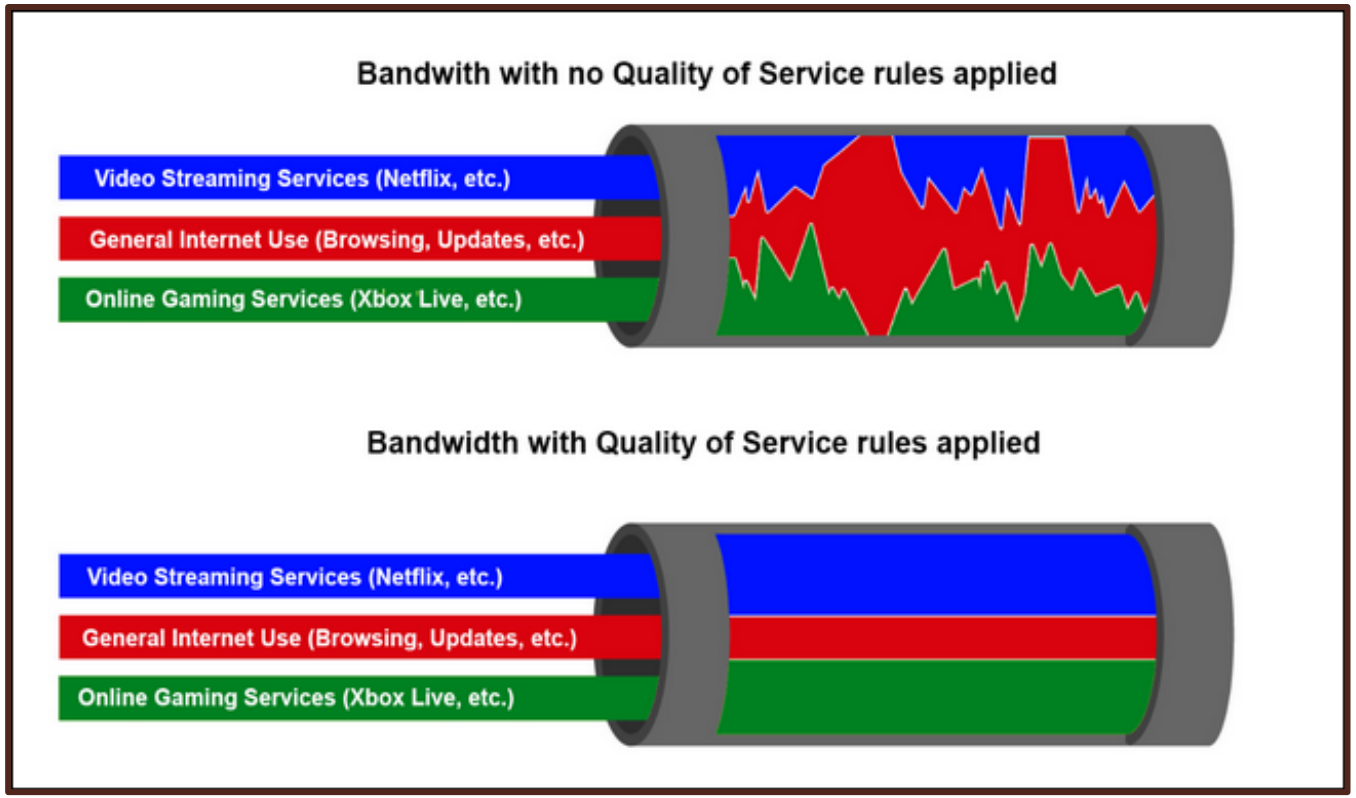


Traffic Bandwidth (Contd.)



- Simulate WAN Applications where **Traffic Policing Policies** may be in effect, **ie Service Level Agreements** between Provider and Customer

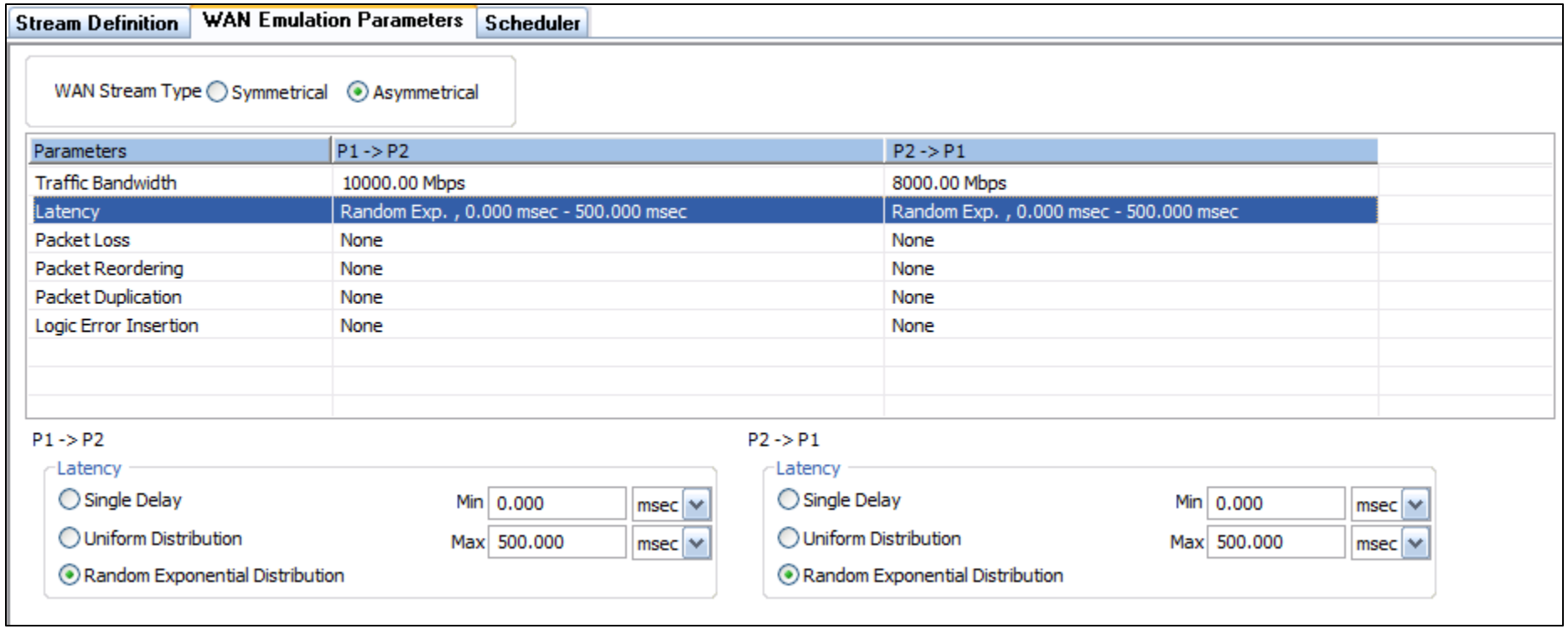
Traffic Bandwidth (Contd.)



- Simulate QoS settings by setting different bandwidth caps on different ports (or port ranges)

Latency / Jitter

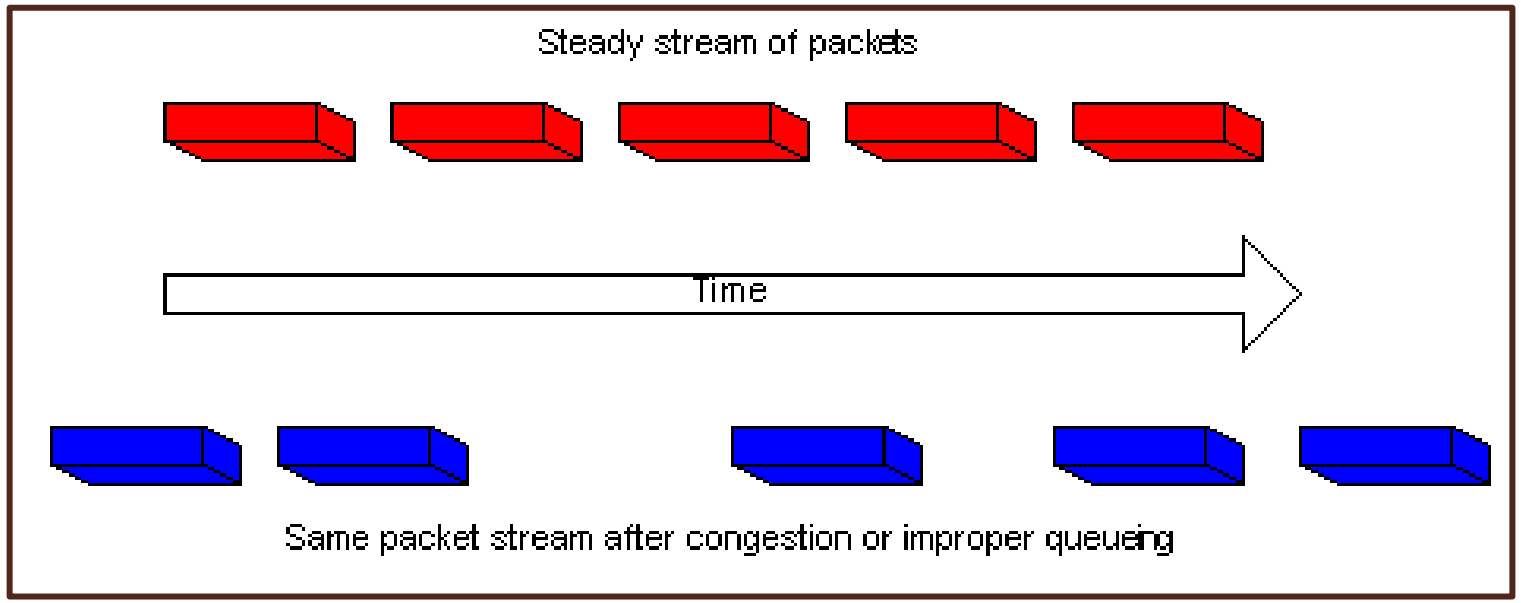
- Apply Static Delay, or a Uniform or Exponential distribution between a minimum and maximum
- Delay a packet up to 500 ms in 1ms increments for 10G ports



Latency / Jitter

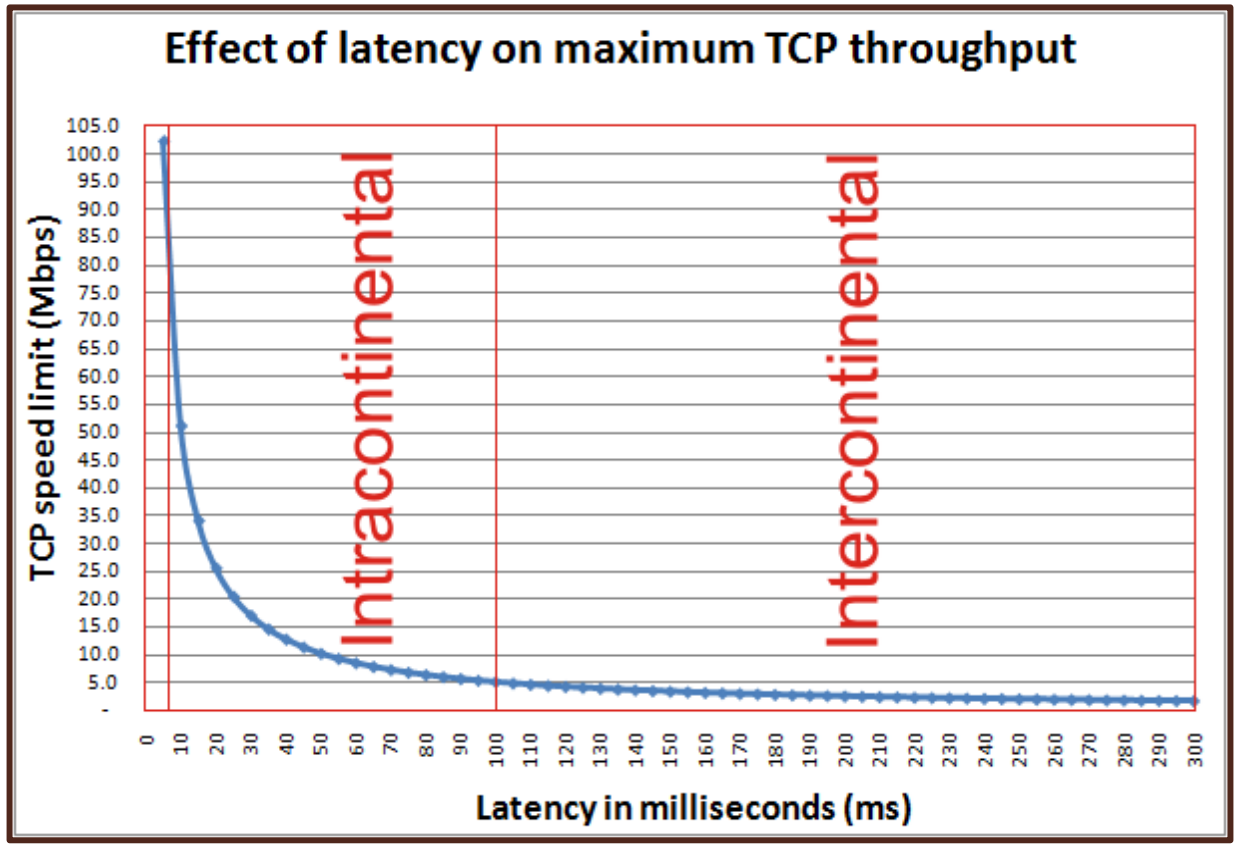
- Apply a large static delay to simulate **backhaul communication (satellite hops, etc.)**

Latency / Jitter (Contd.)



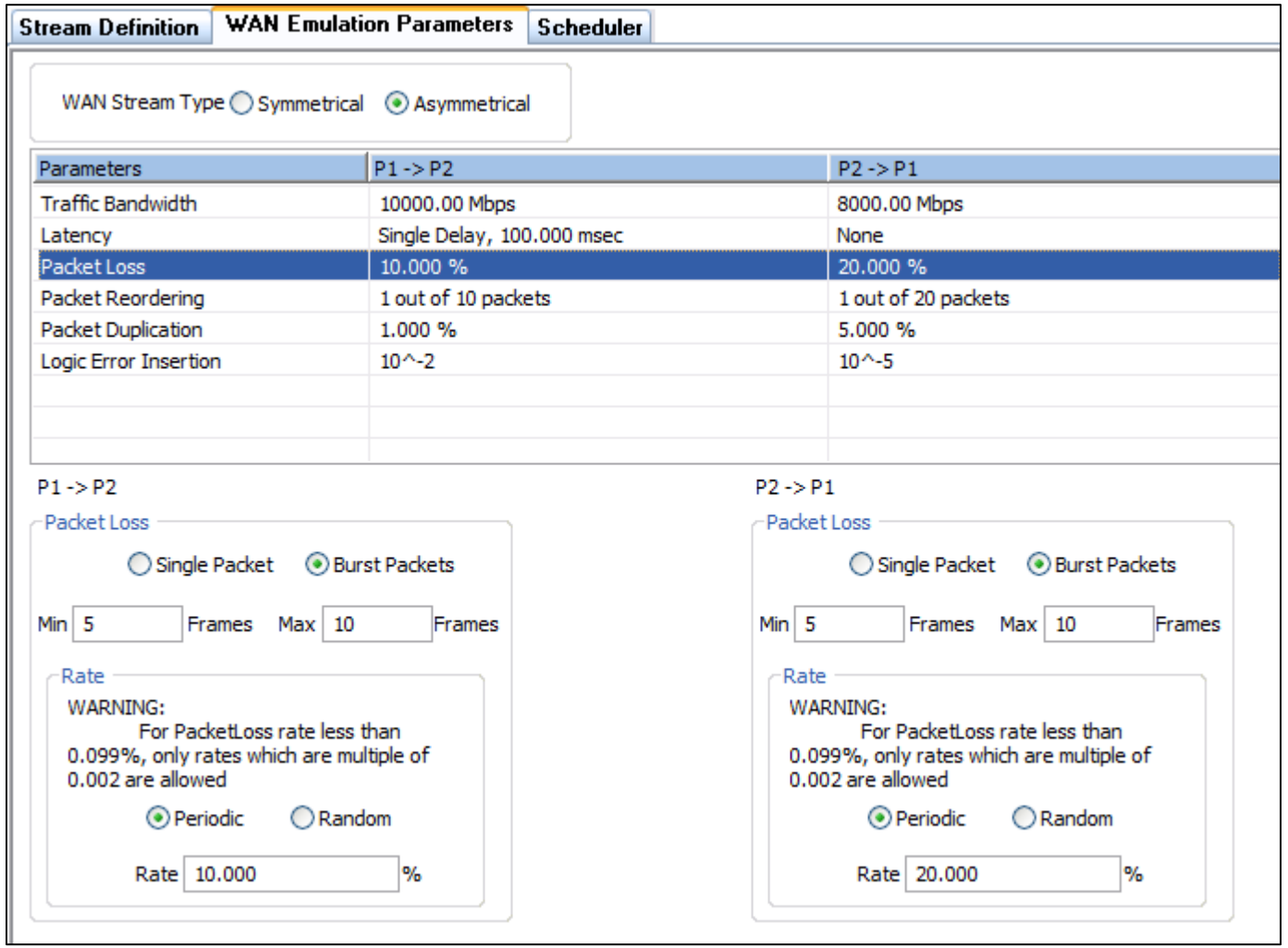
- Apply Variable delay (ie, Jitter) to simulate Traffic Shaping policies and/or Network Congestion
- Jitter leads to packet discard (and therefore data loss) in Real Time UDP Applications

Latency / Jitter (Contd.)



- Increased Latency causes TCP applications to spend increasing amounts of time idling while waiting for ACKs from the far side, thereby throttling throughput

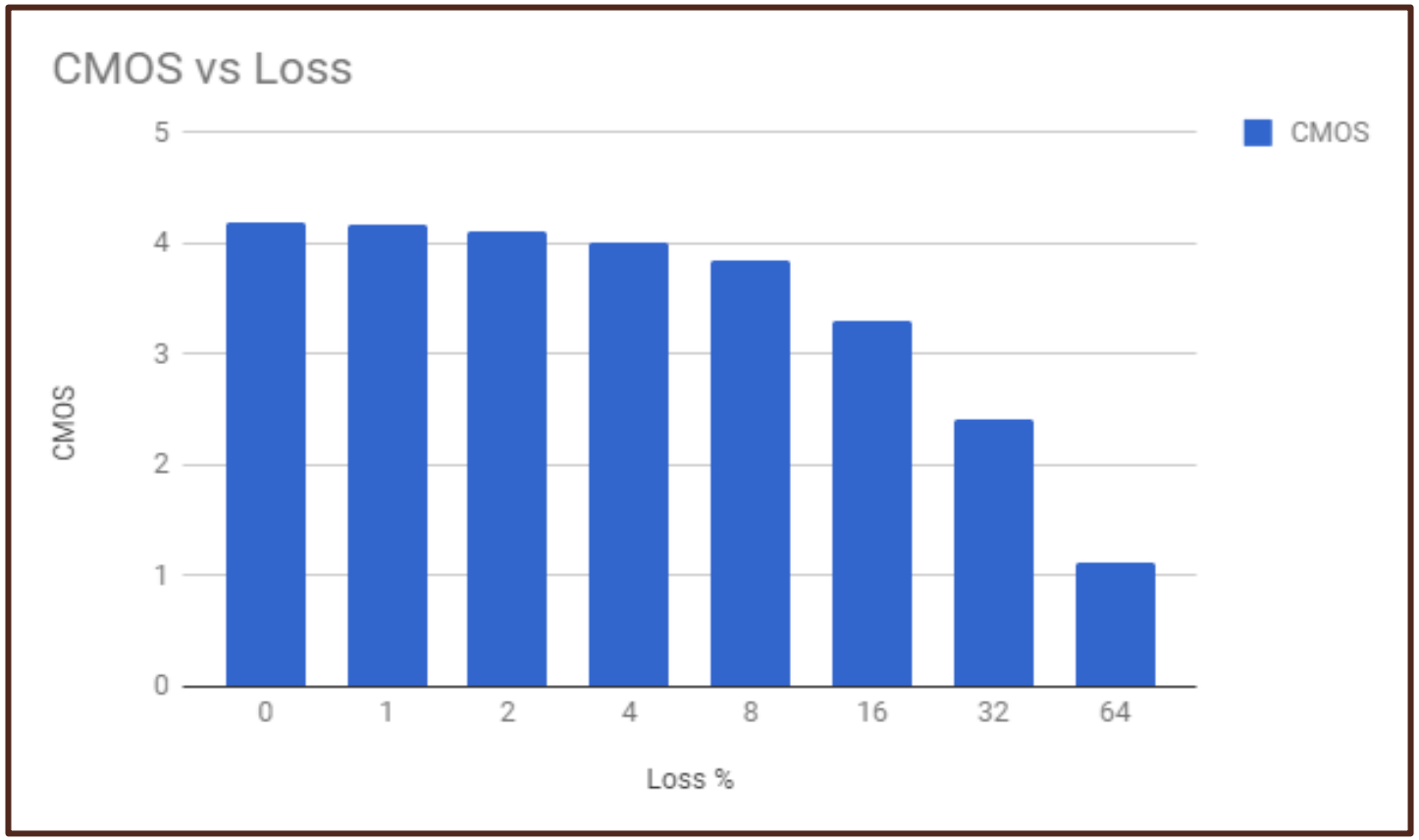
Packet Loss



- Randomly drop from 0.01% to 100% of all Packets in the stream

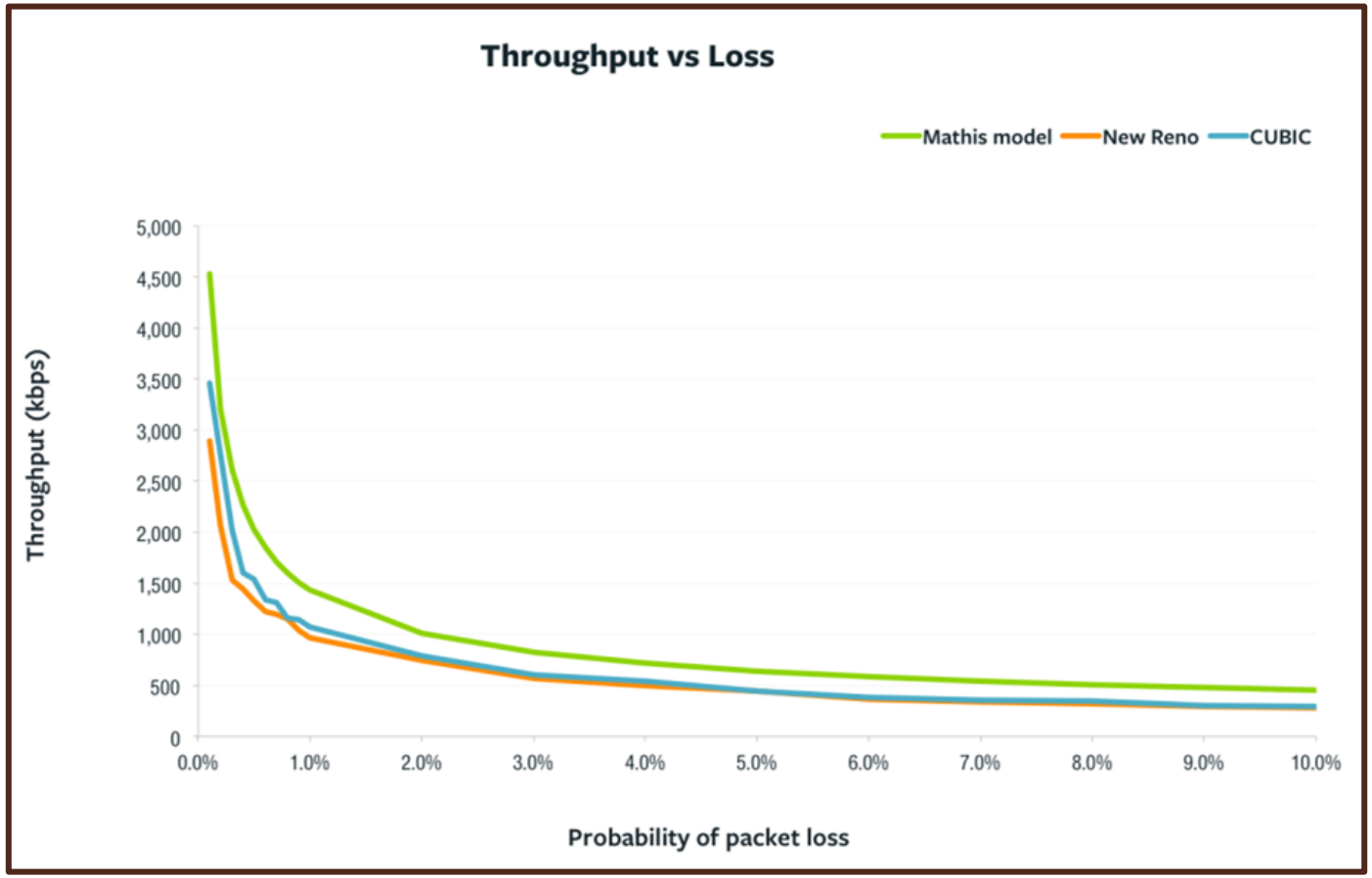
GL Communications

Packet Loss (Contd.)



- Real Time UDP Applications are resilient to minor loss, but vulnerable to heavy loss

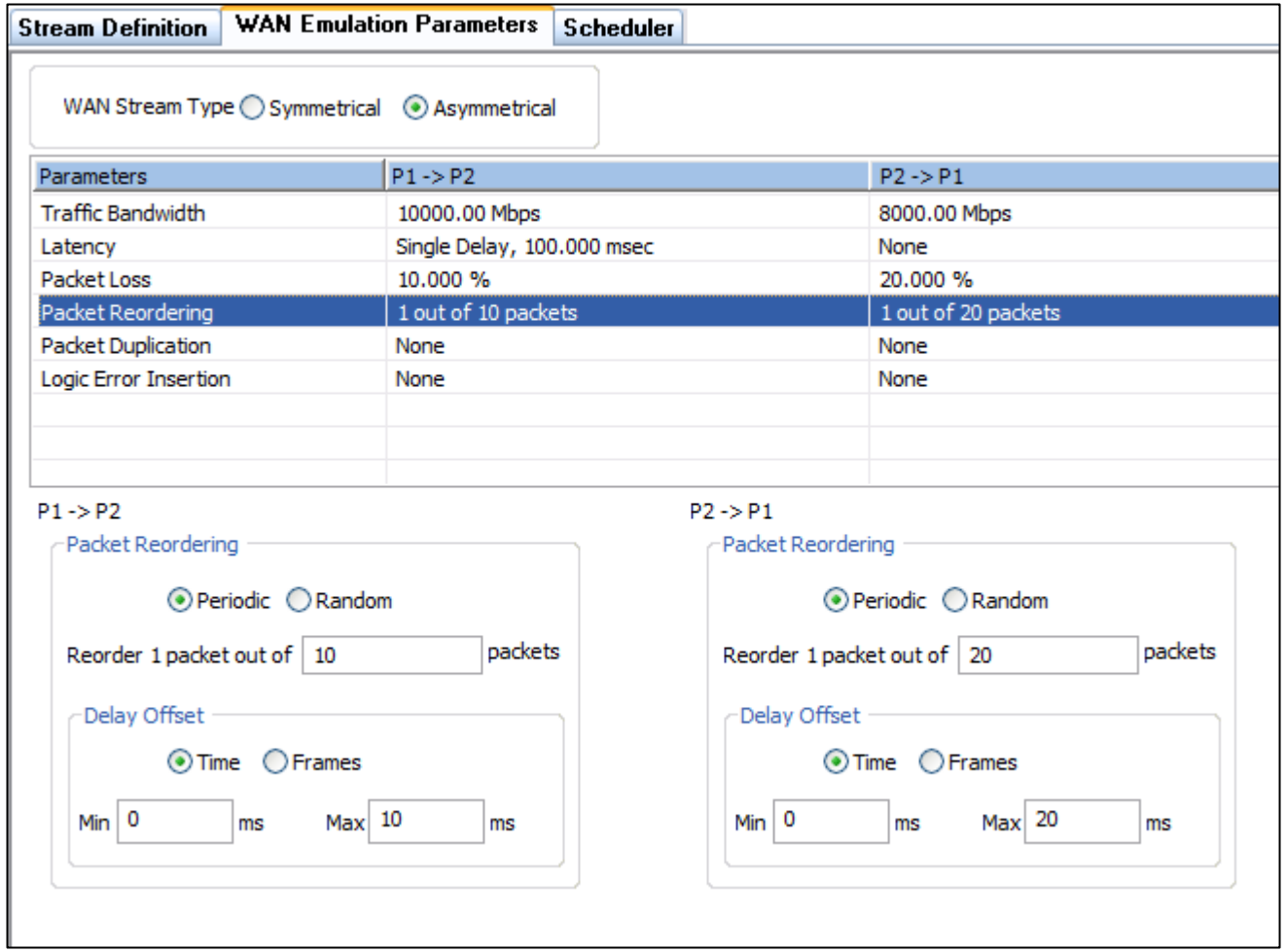
Packet Loss (Contd.)



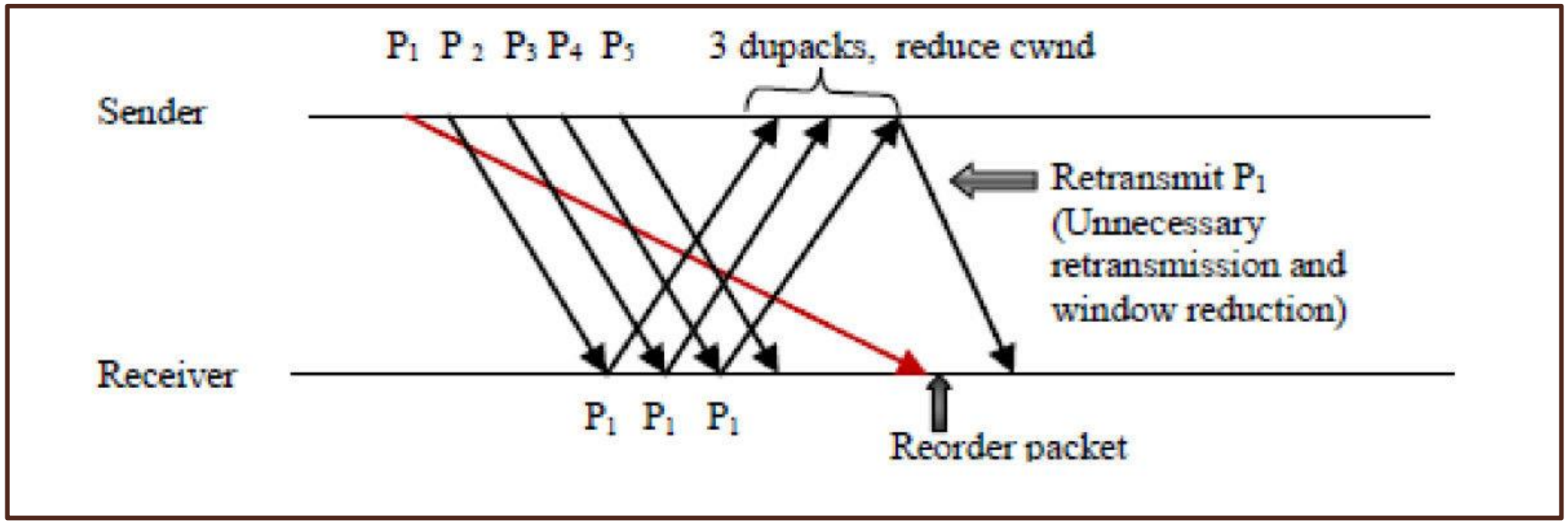
- TCP Applications are vulnerable to even very minor loss rates as every loss results in retransmissions and reduced window sizes.

Packet Reordering

- Reorder 1 out of every X packets.
- Set a minimum time in ms to hold the reordered packet
- Set a maximum time in ms to hold the reordered packet

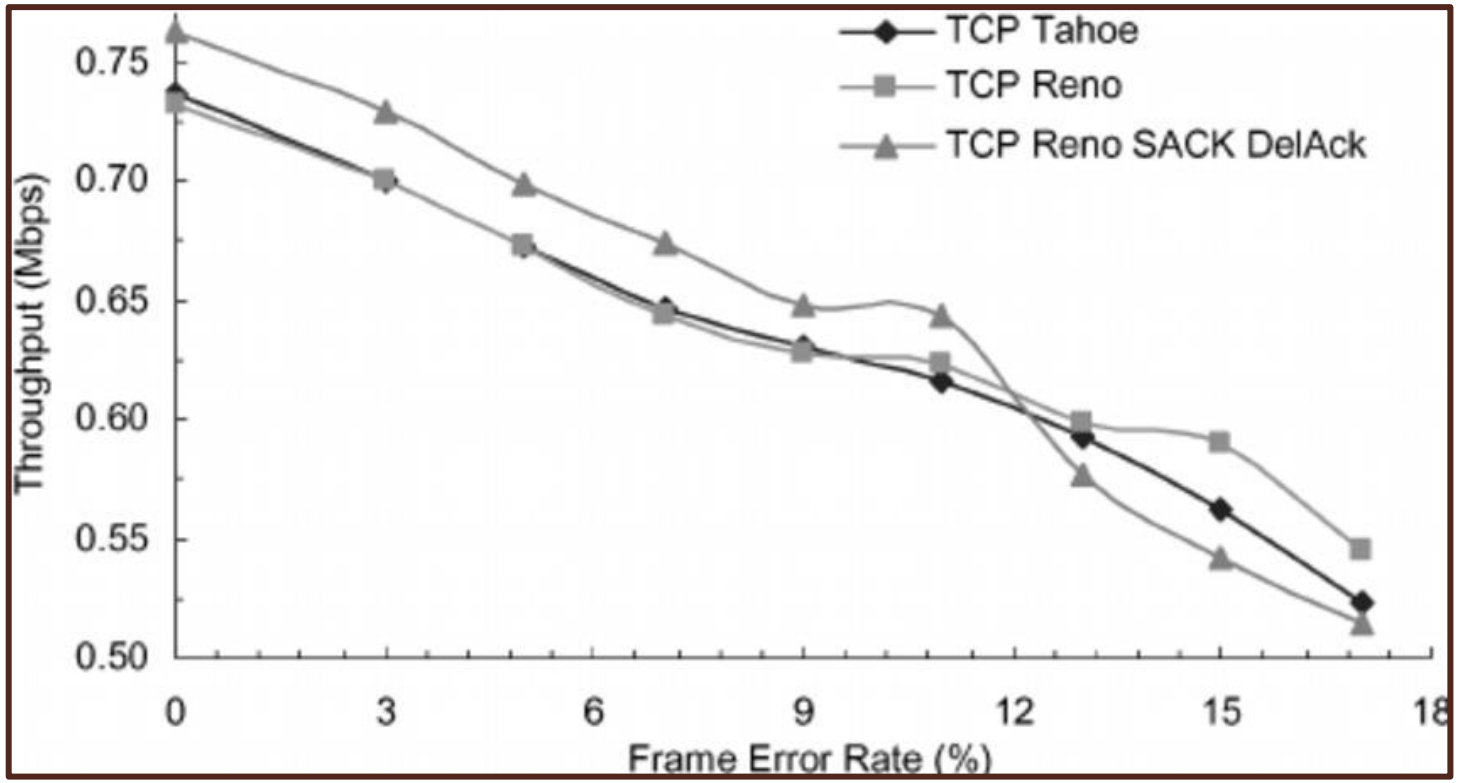


Packet Reordering (Contd.)



- When a packet is out-of-order, TCP behaves exactly as though the preceding packets are lost resulting in duplicate ACKs, retransmissions and window reduction

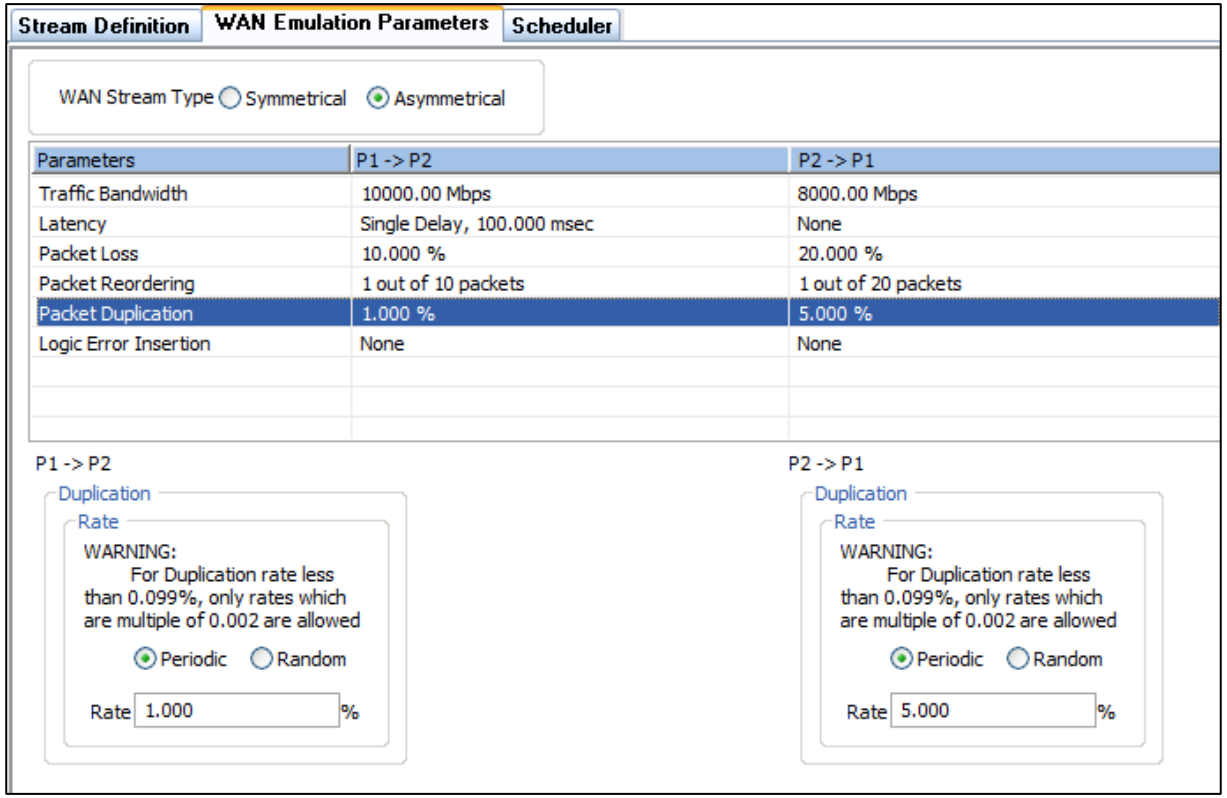
Packet Reordering (Contd.)



- TCP Selective Acknowledgement (SACK) can mitigate this issue by letting the receive side Acknowledge OOO packets.

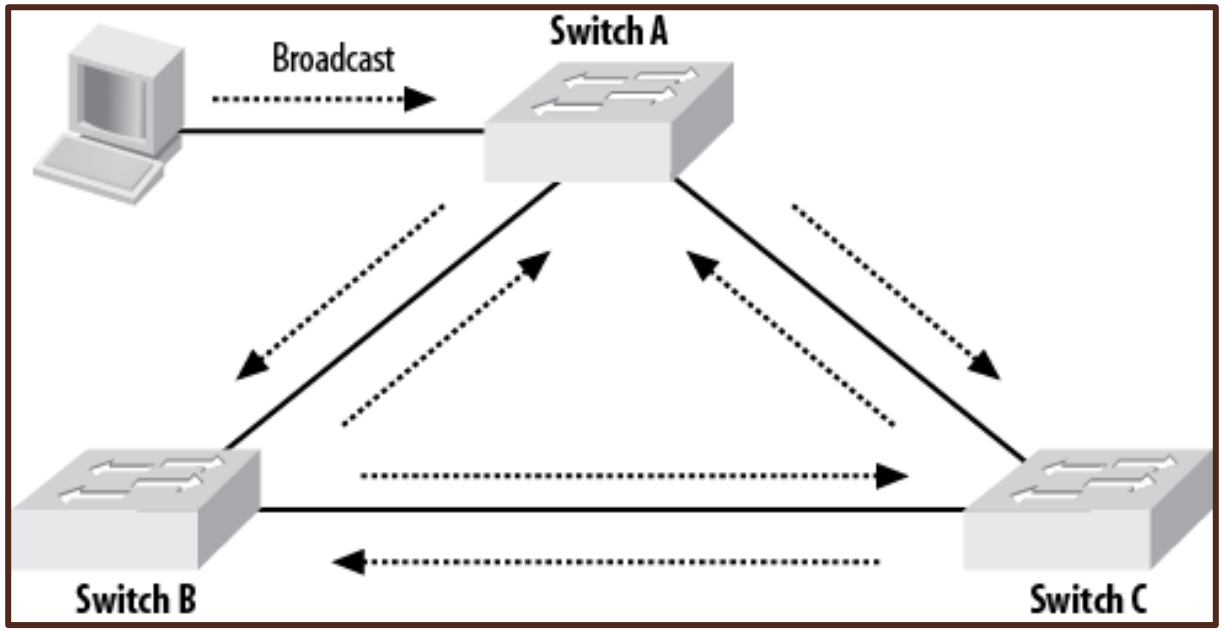
Packet Duplication

- Randomly duplicate from 0.01% to 100% of all Packets in the stream
- Emulate WAN applications where multiple paths are possible and Load Balancing may be present



GL Communications

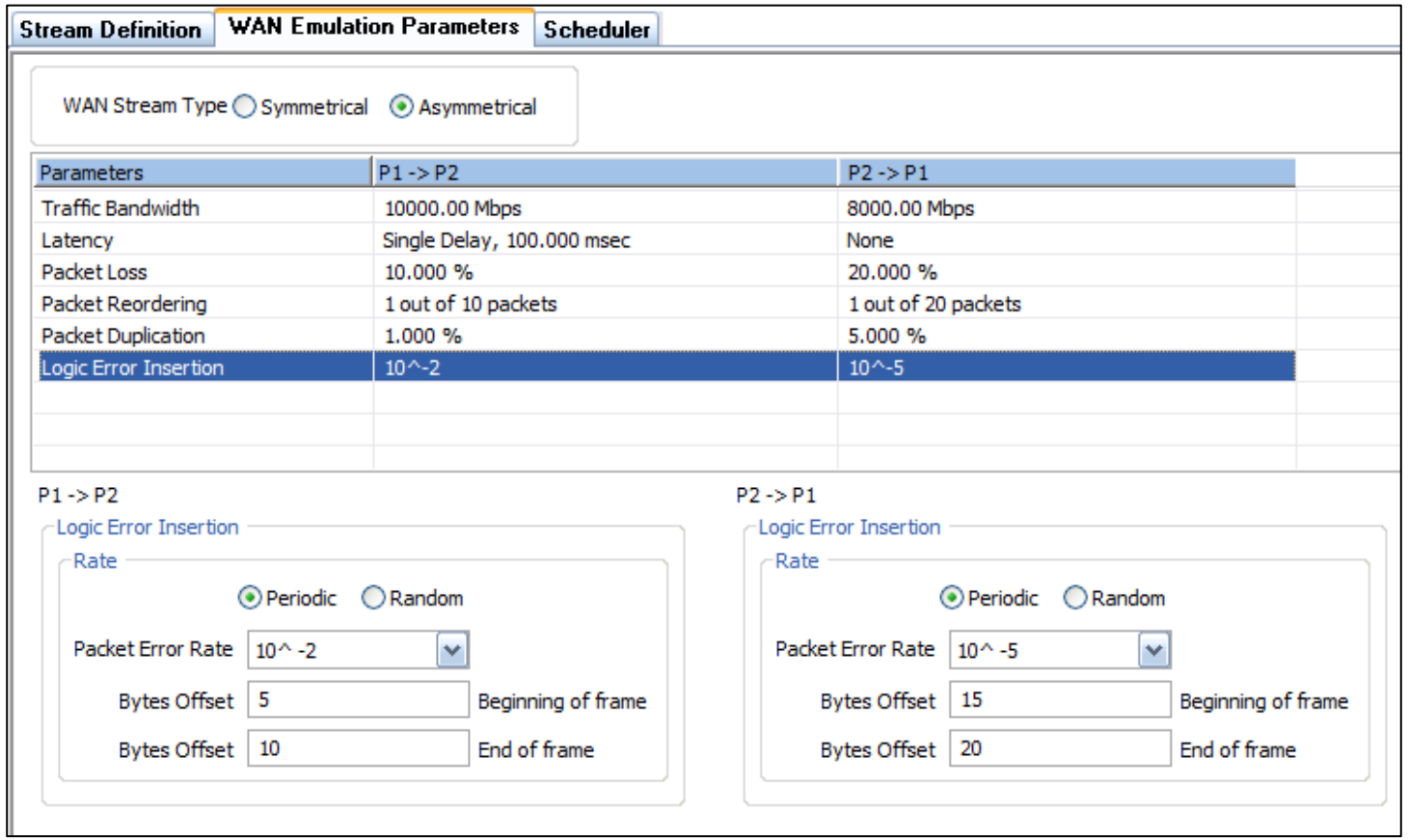
Packet Duplication (Contd.)



- Duplication can be fatal in broadcast situations (i.e, broadcast storm)
- Similarly dangerous in multicast applications where small network misconfigurations can have disproportionately large consequences
- Watch out for this in multipath Spanning Tree networks

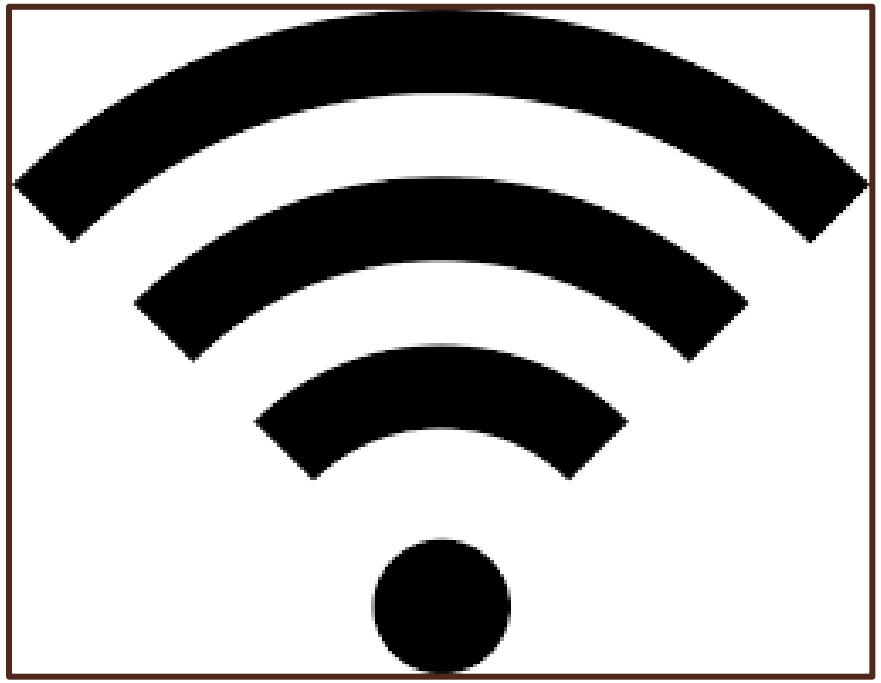
Logic Error Insertion

- Insert a single bit error every 10^-X frames (-1 <= X <= -9)
- Use byte offsets to target particular parts of a frame



GL Communications

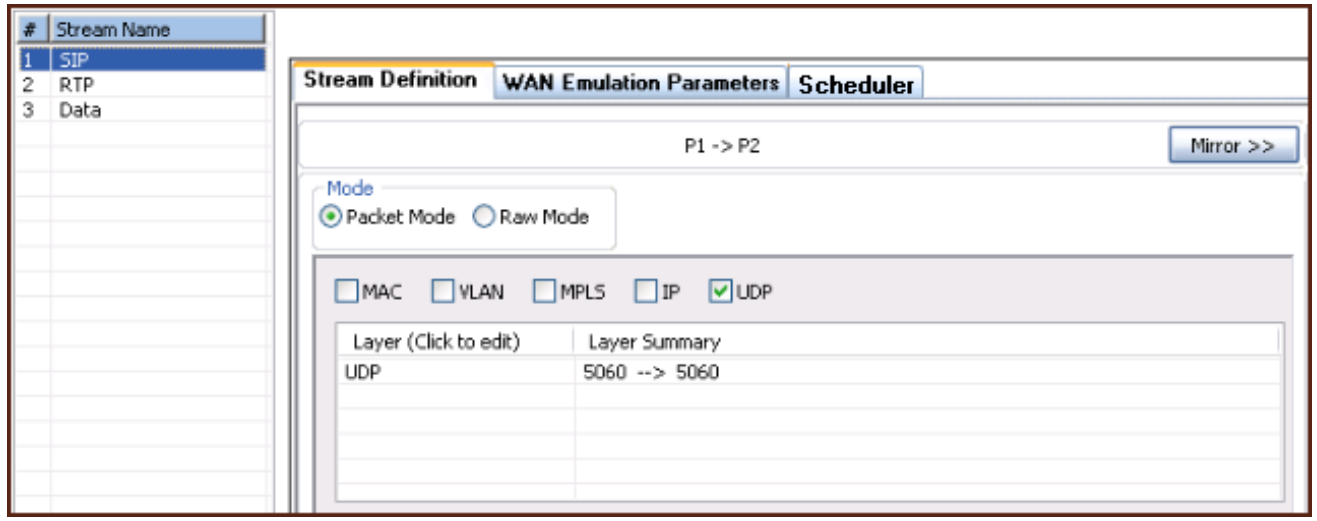
Logic Error Insertion (Contd.)



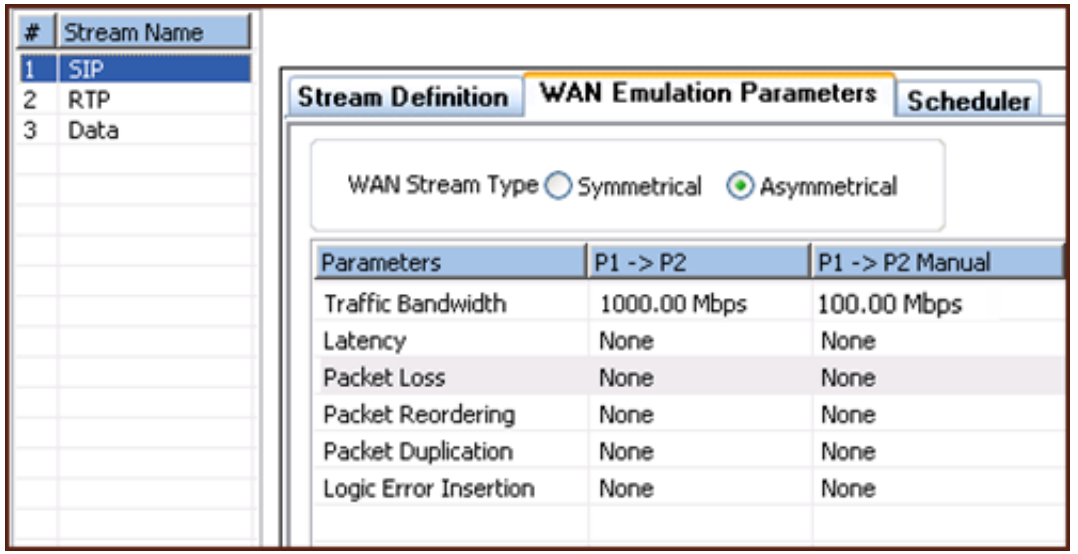
- Cellular and WiFi links are very prone to bit errors (as well as latency and bandwidth issues)

Application Examples

VoIP

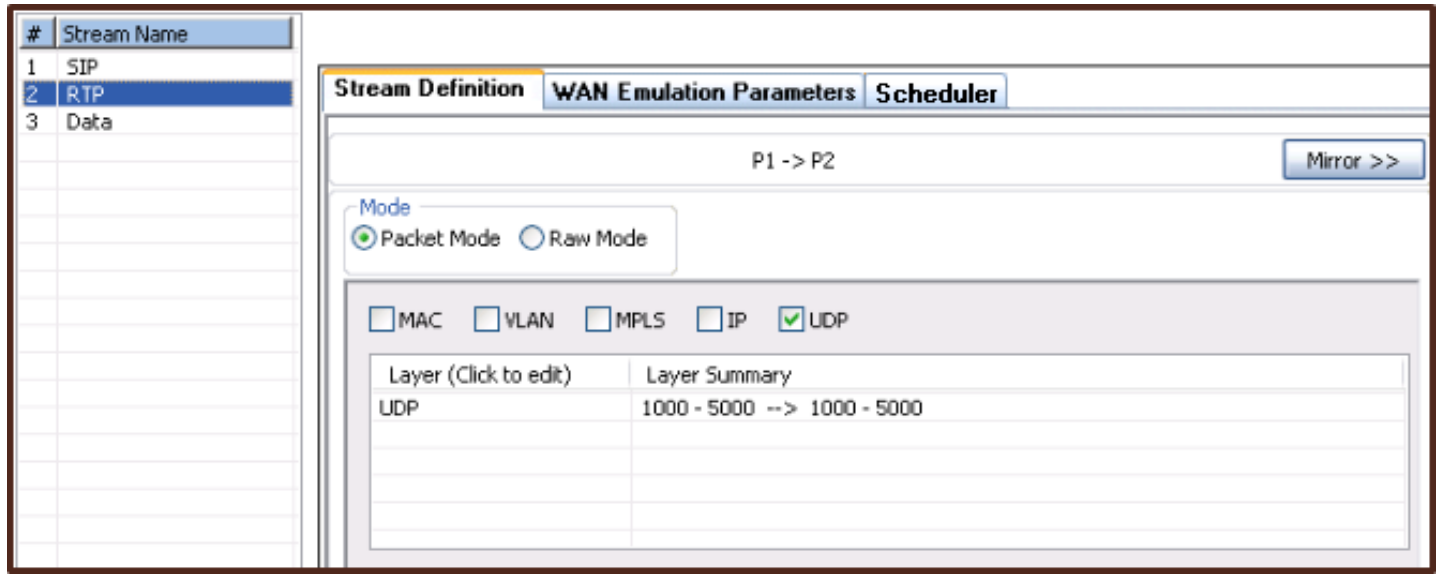


- Configure SIP packets to be completely unimpaired

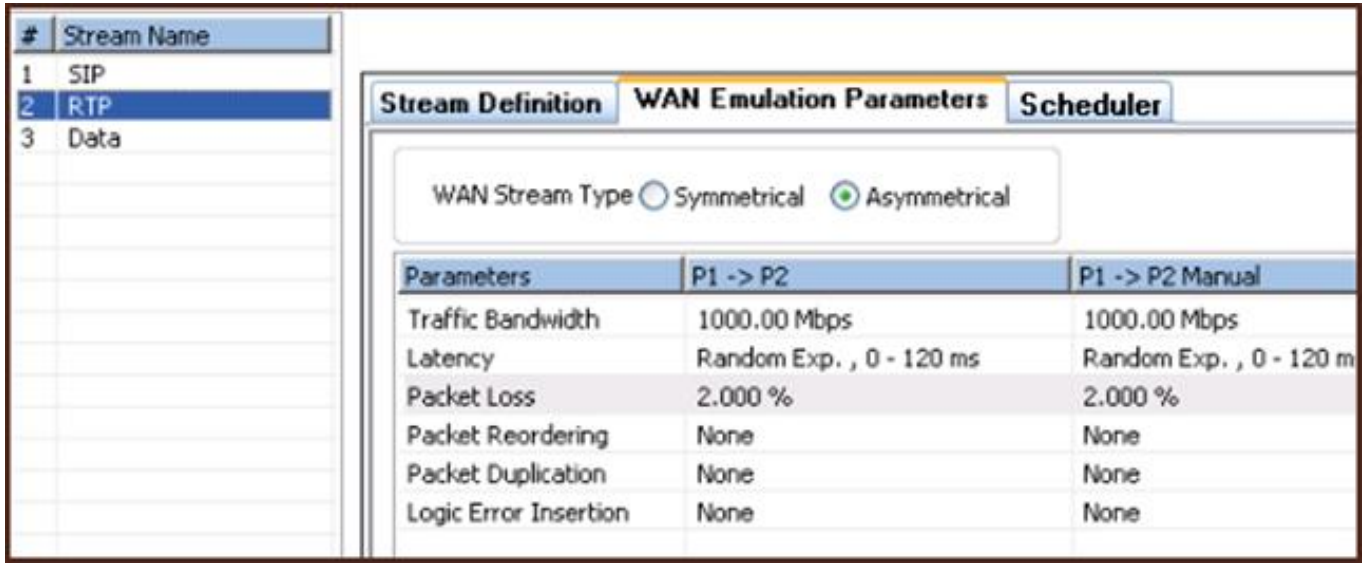


GL Communications

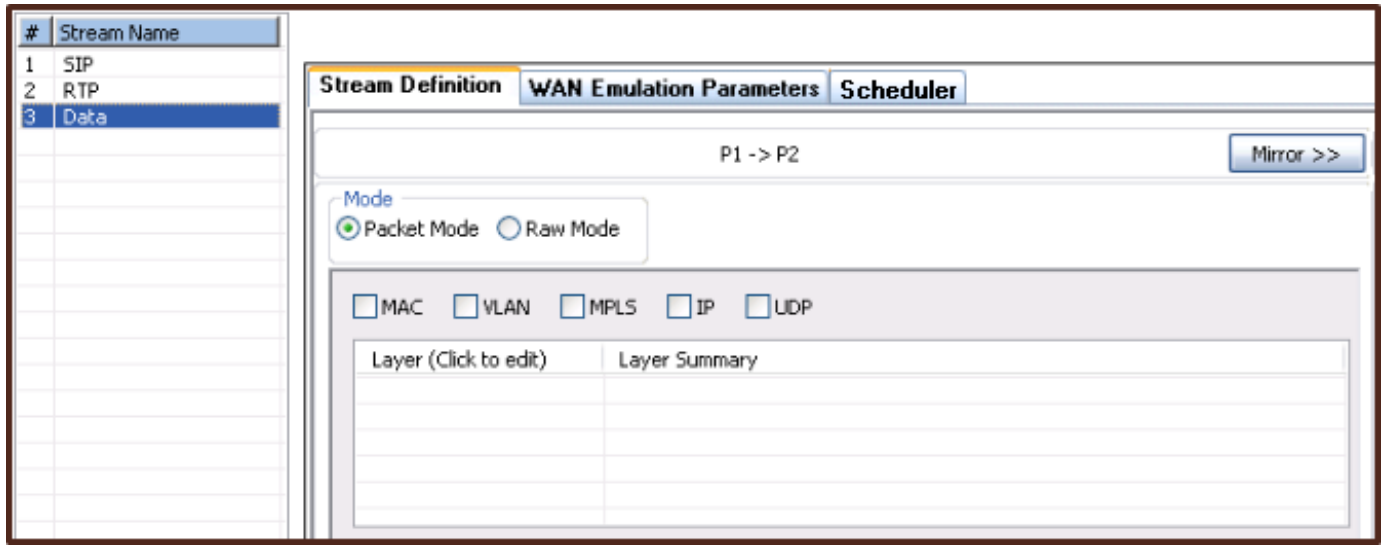
VoIP (Contd.)



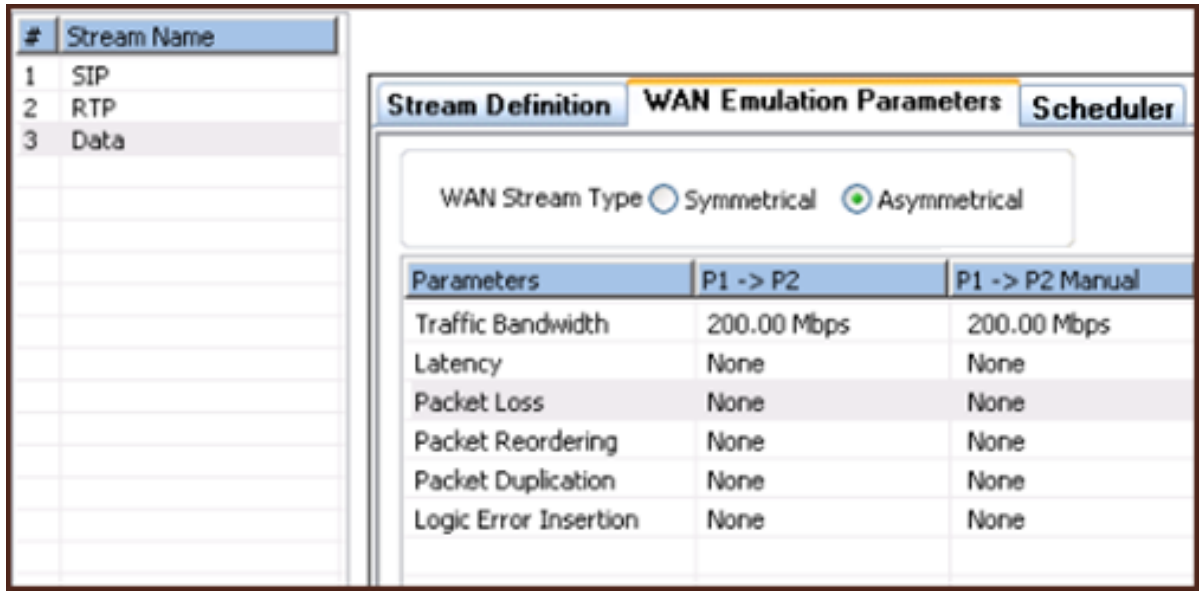
- Apply loss and jitter to RTP streams



VoIP (Contd.)

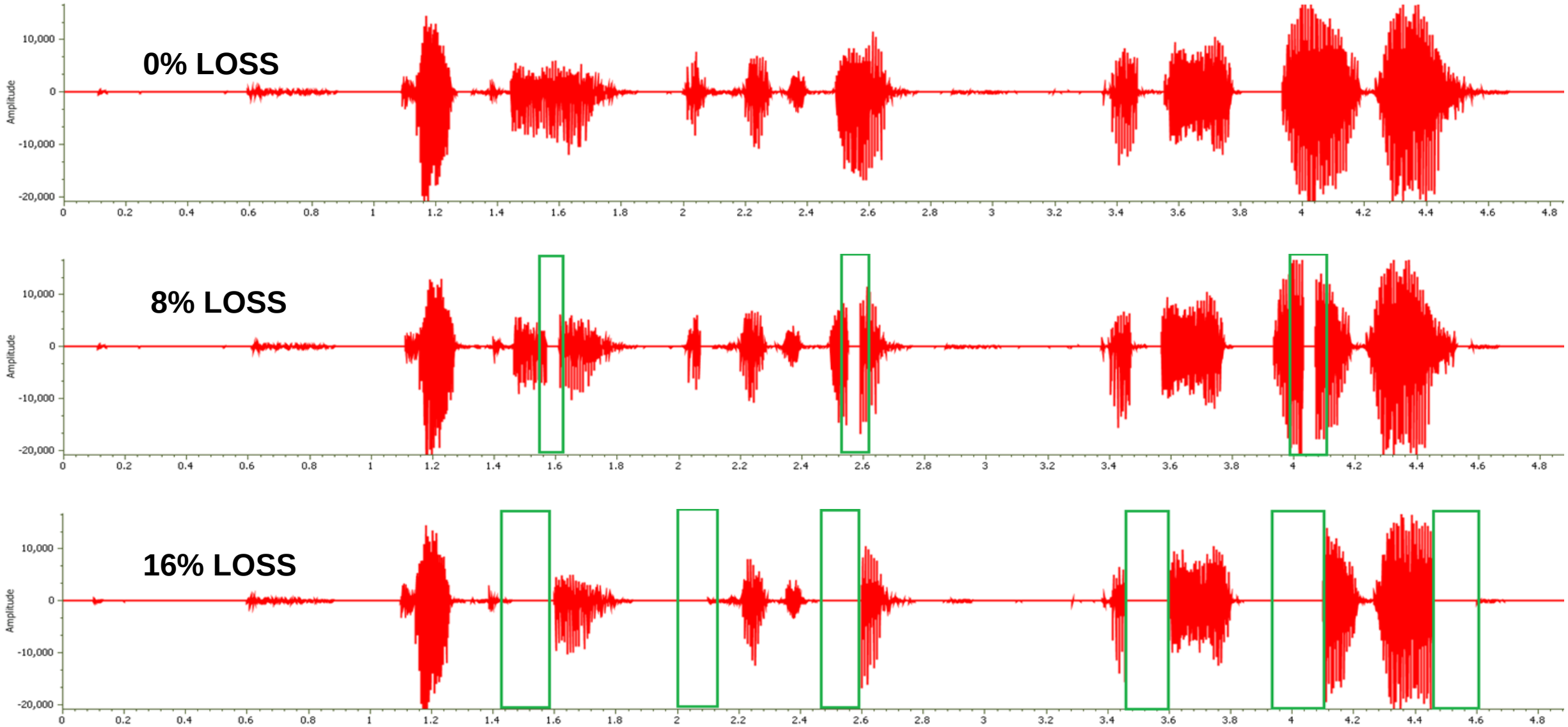


- Set an SLA style bandwidth cap on Data



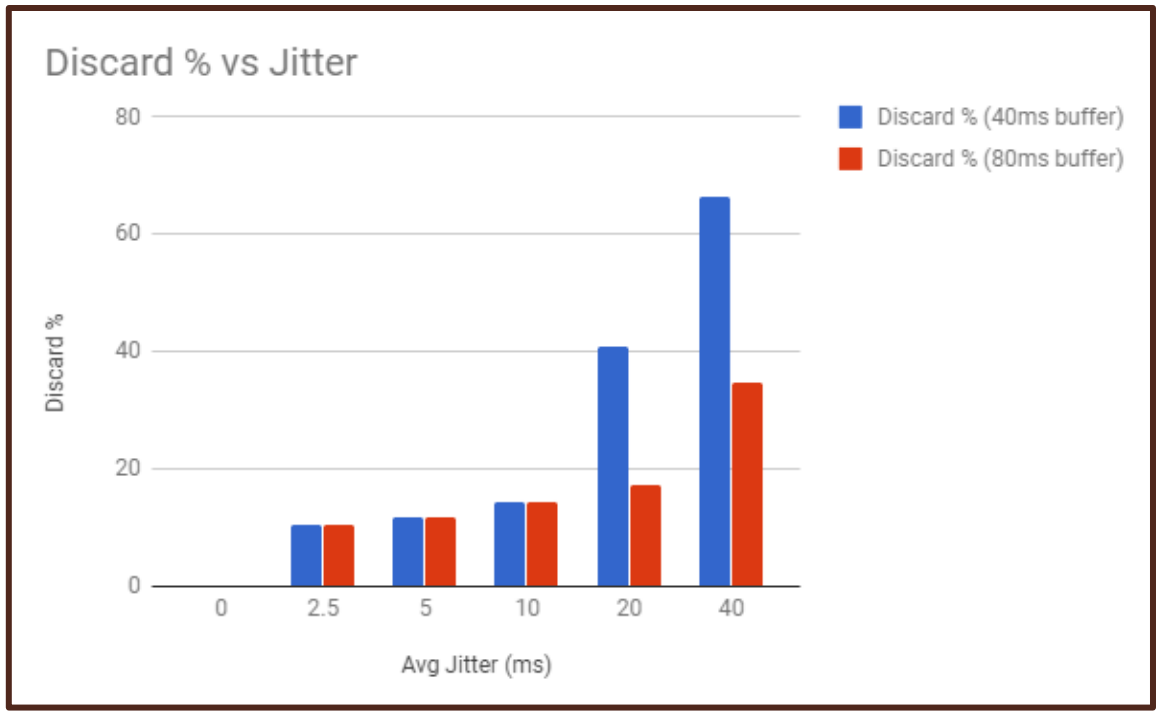
GL Communications

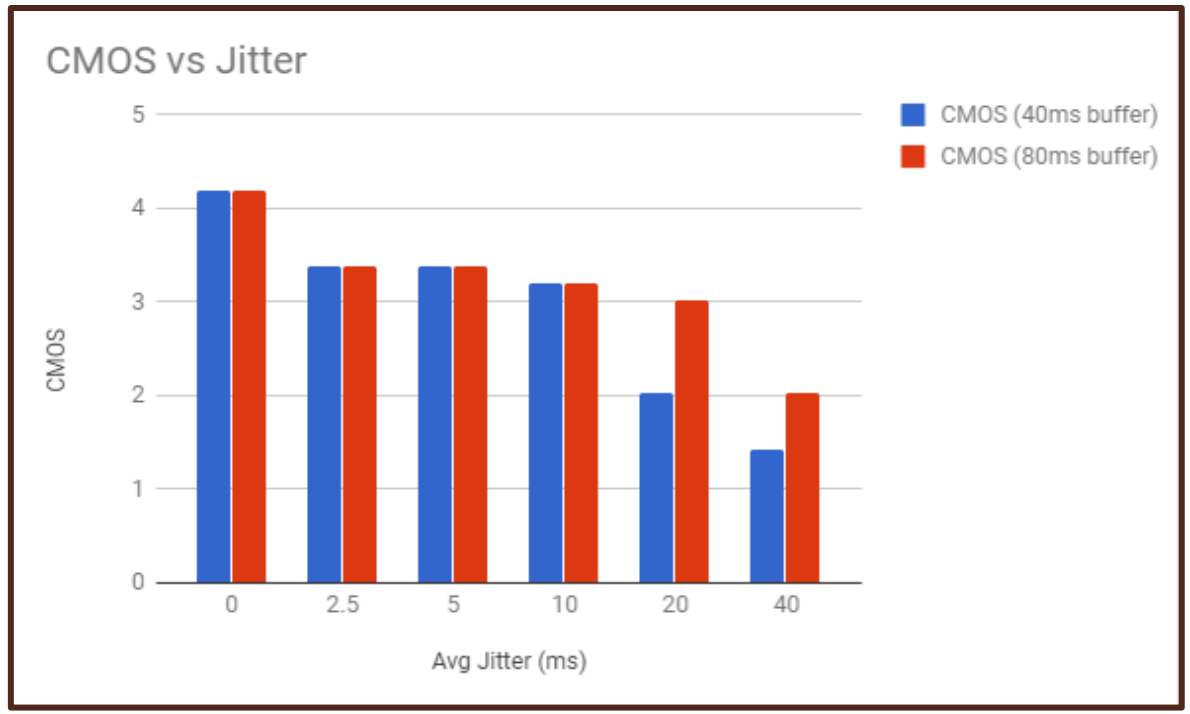
VoIP (Contd.)



GL Communications

VoIP (Contd.)

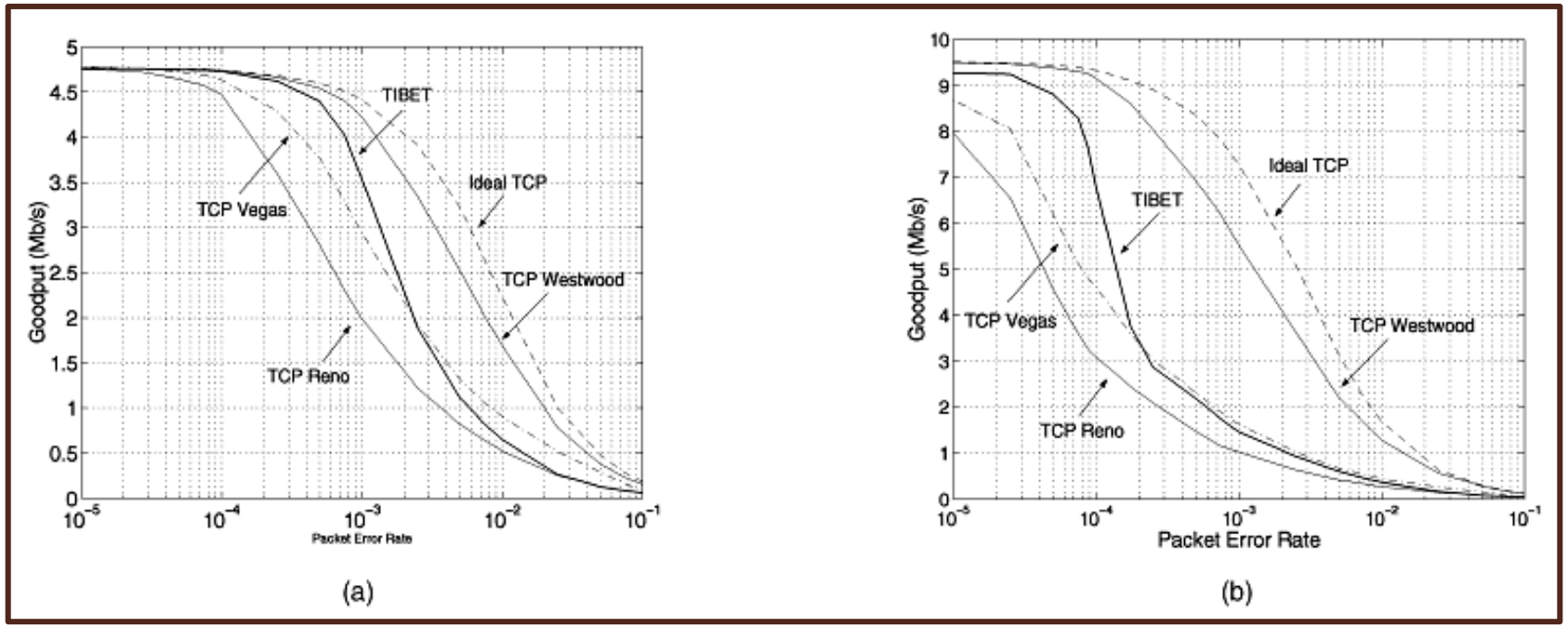




- Determine how your application will behave under expected (and unexpected) network conditions
- Determine what codecs you should use, what jitter buffers, etc.

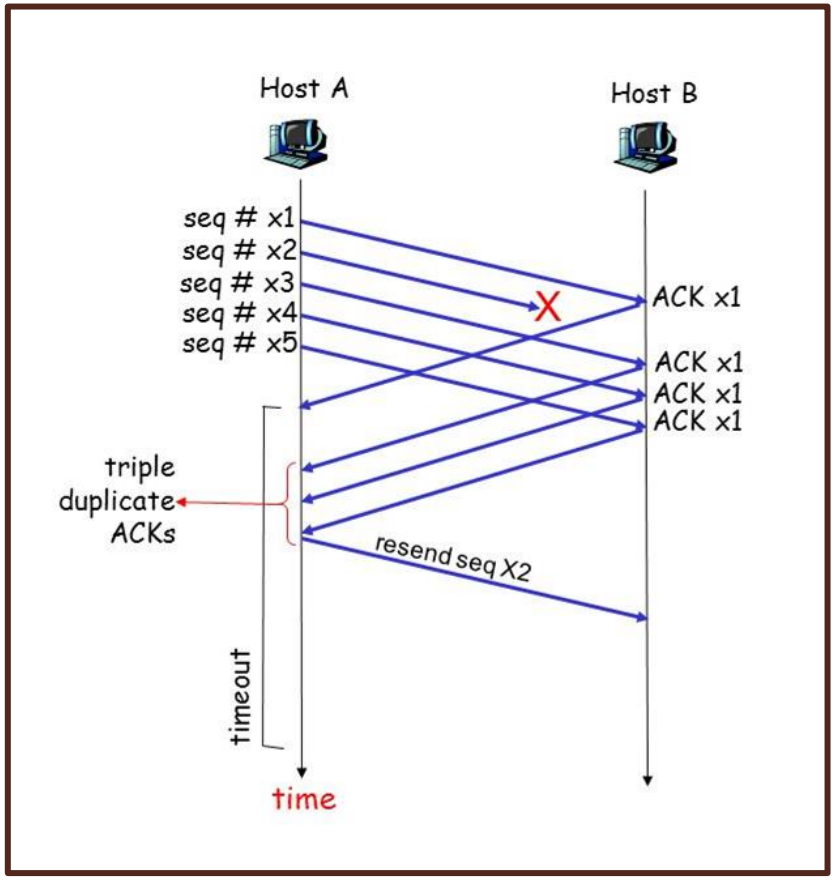
GL Communications

TCP Over Wireless



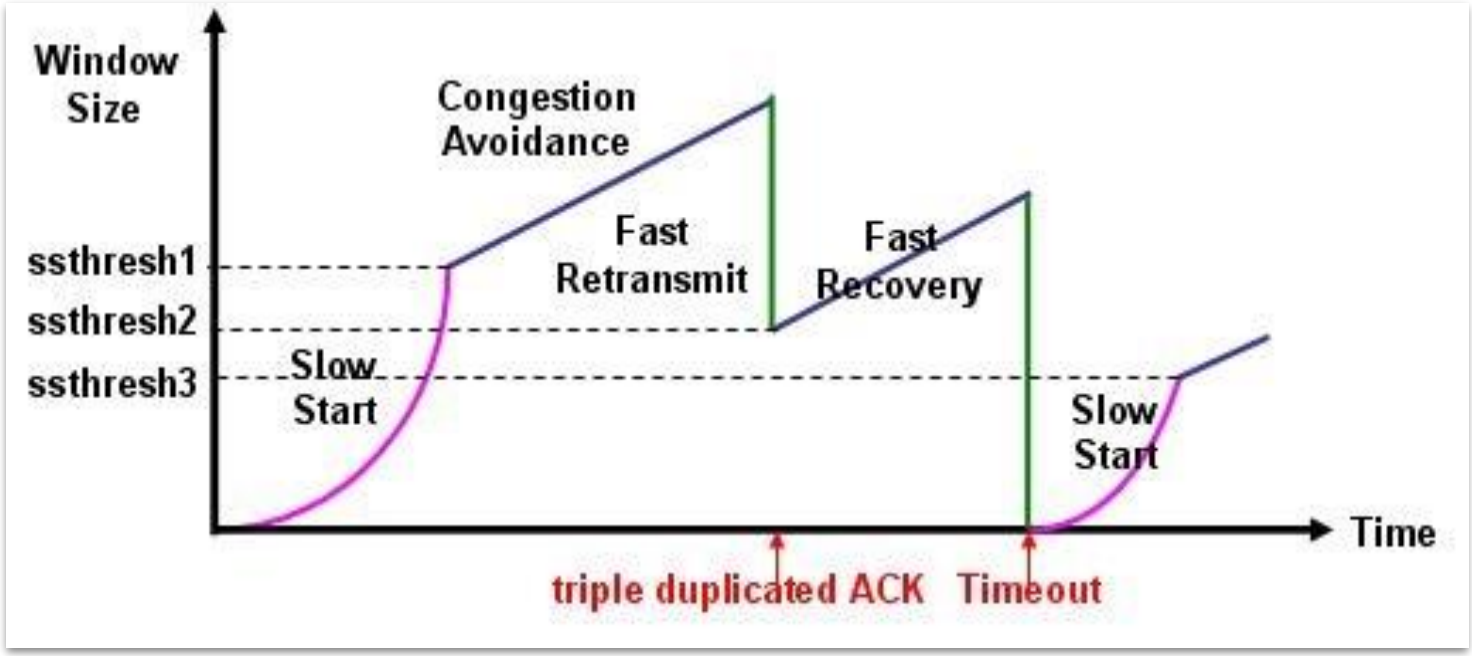
- TCP was first described in an IEEE paper written in 1974
- Ethernet was first standardized by the IEEE in 1983
- The first GSM Network went online in 1991

TCP Over Wireless (Contd.)



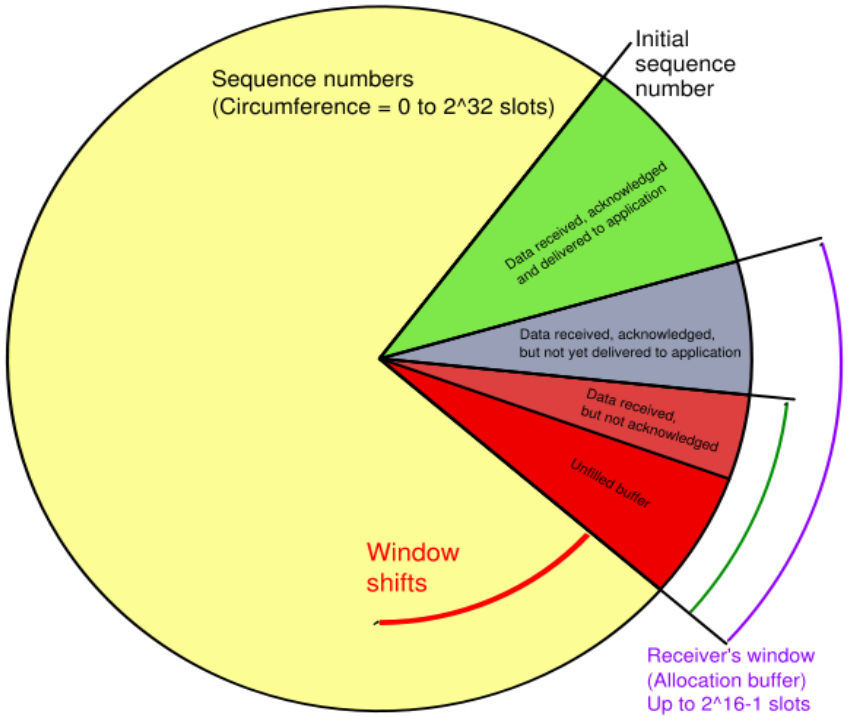
- Loss in a TCP application leads to duplicate ACKs, which lead to retransmissions

TCP Over Wireless (Contd.)



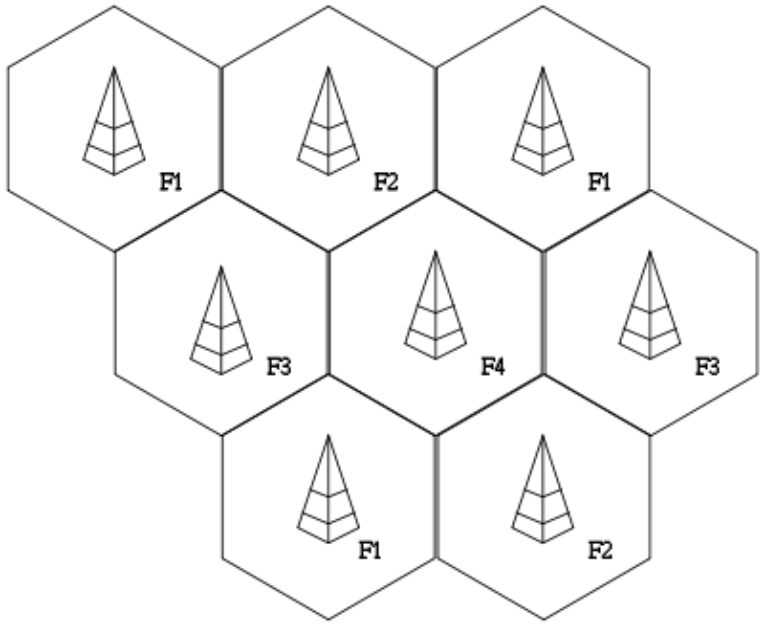
- TCP assumes that Loss is due to Congestion
- When Loss occurs TCP automatically cuts throughput to avoid congestion

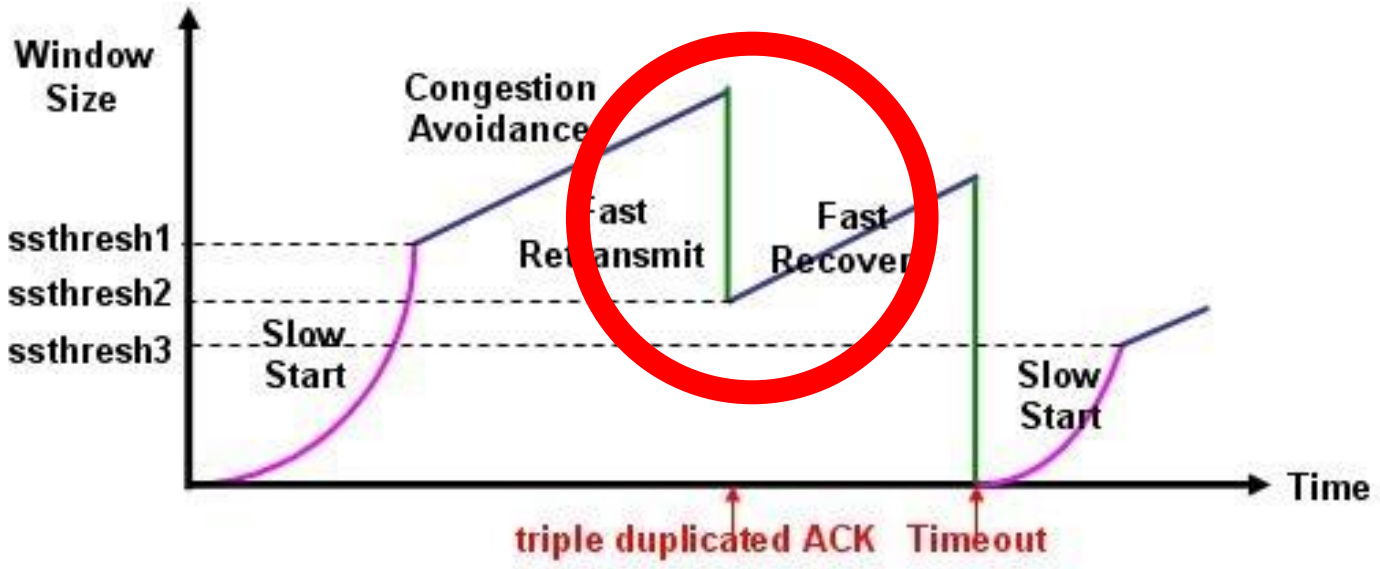
TCP Over Wireless (Contd.)



- TCP uses a Sliding Window mechanism to limit how many unacknowledged bytes can be transmitted before the sender is forced to idle
- High Latency links slow ACKs, cause forced idle, and limit throughput

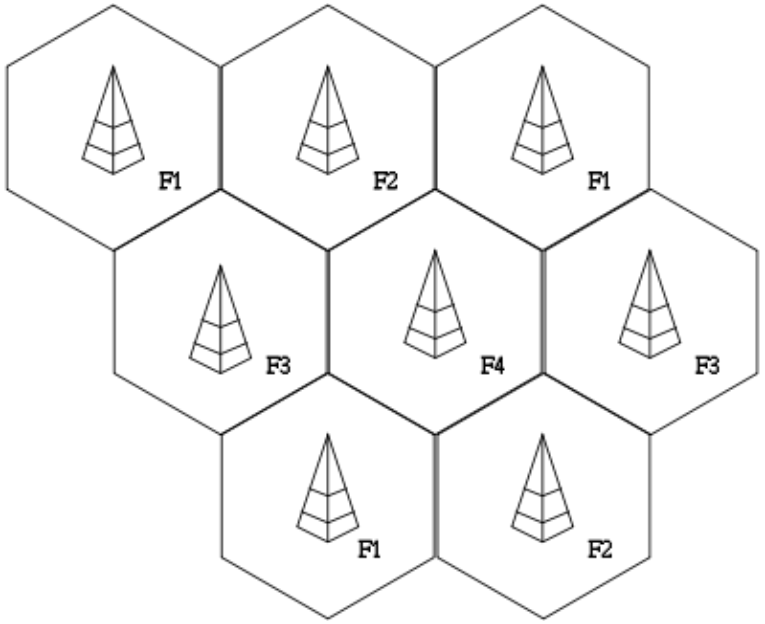
TCP Over Wireless (Contd.)

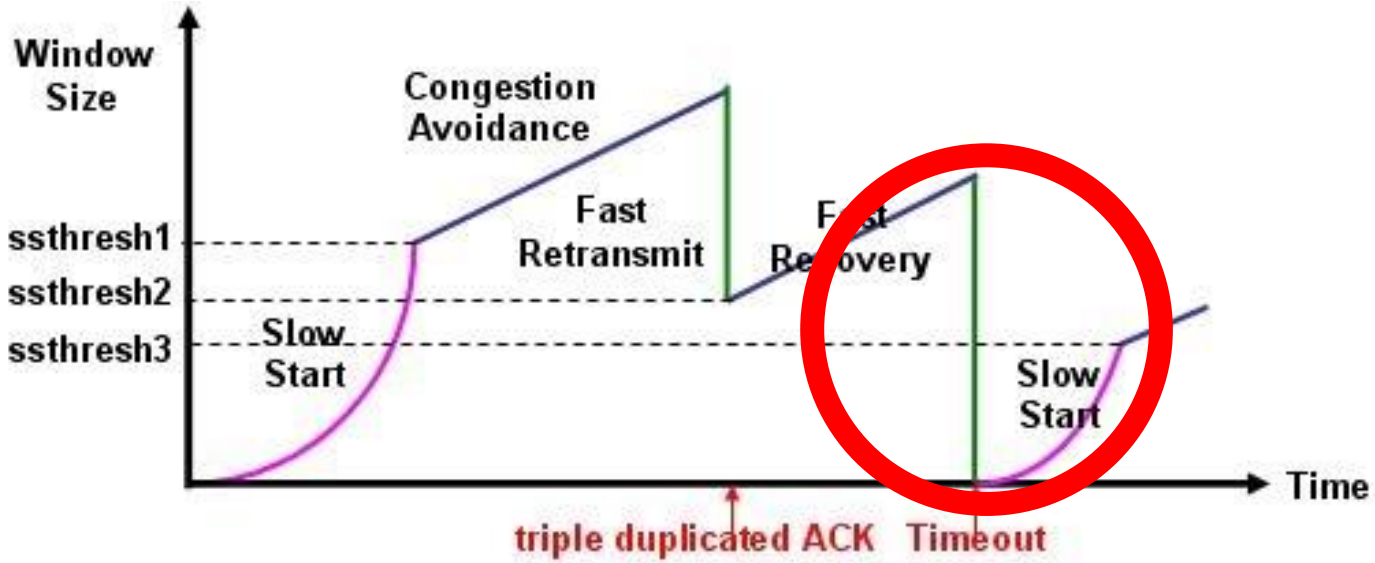




- When a client moves between cells handoff will cause Loss
- TCP will interpret this as Congestion and cut throughput even if the exact same amount of bandwidth is still available (even if more bandwidth is now available)!

TCP Over Wireless (Contd.)





- TCP has Retransmission Timeout mechanism that attempts to track the RTT of the connection
- If Latency suddenly increases (ie cell handoff), this can easily cause Timeouts to trigger, immediately cutting throughput to the minimum!

THANK YOU!