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: **http://www.gl.com**

Company Overview – GL Communications Inc.

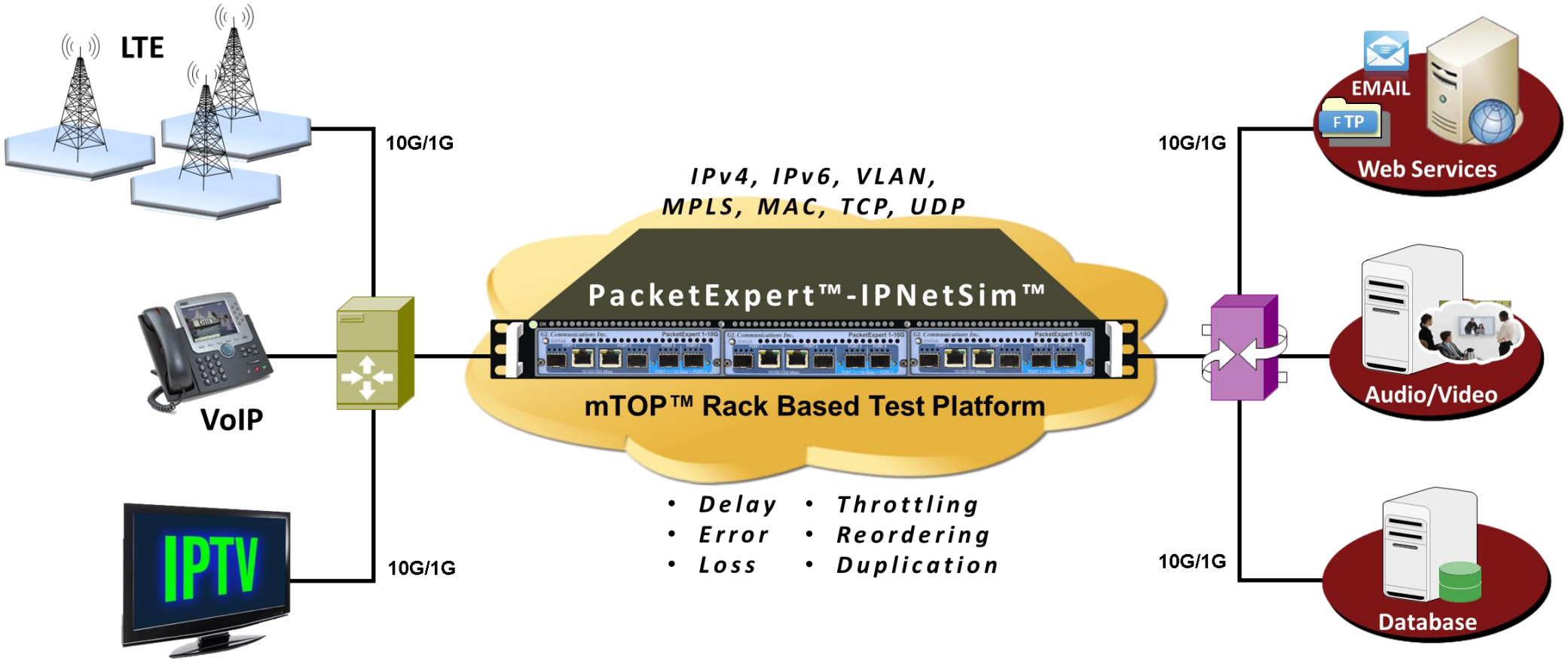
- Headquarters: Gaithersburg, Maryland USA
  - Multiple GL Branch Offices and Worldwide Representatives
- Founded in 1986
- Test & Measurement Equipment
  - IP, VoIP, SONET-SDH, TDM & Wireless Solutions
  - Visualization, Capture, Storage, Portability, Cost-Effectiveness
  - Endpoints Devices & Core Network Elements
- Engineering Consulting Services (mainly for Transportation and Government agencies)

GL Communications

Overview

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

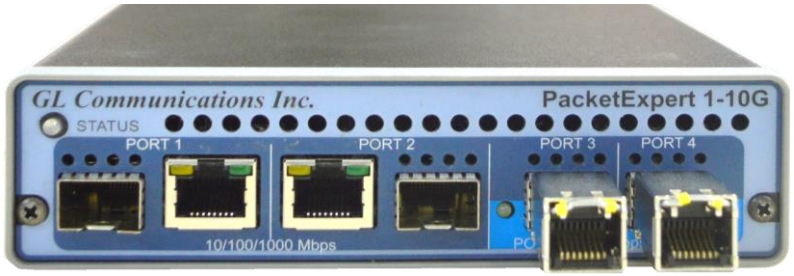
- ❖Lab Testing Solution - application and automation
- ❖Emulate Full Duplex 1 Gbps and 10 Gbps networks
- ❖Real-world network conditions by imposing impairments
- ❖Multiple streams independently configured



GL Communications

Hardware Overview

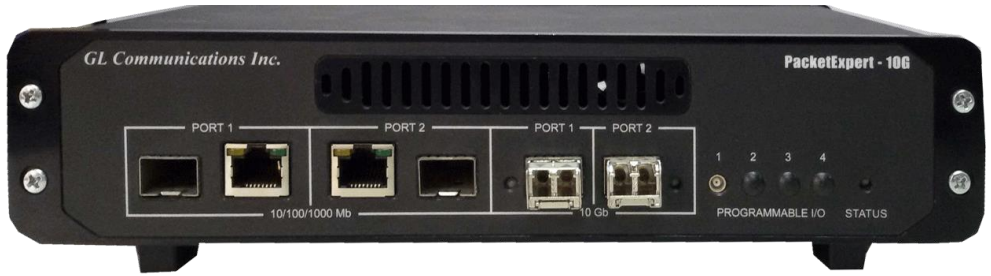
Portable Units



PacketExpert™ 10GX Standalone

- 4 x 1 Gbps Optical OR Electrical
- 2 x 10 Gbps Optical only

PacketExpert™ 1G (4 Port)

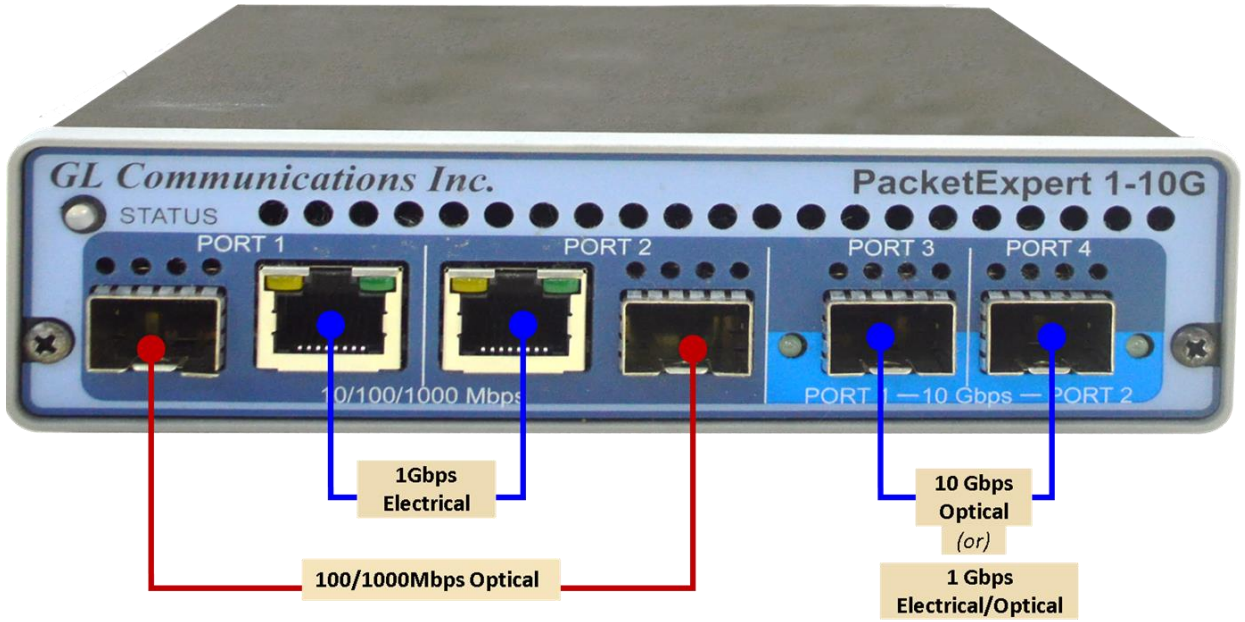


PacketExpert™ 10G Standalone

- 2 x 1 Gbps Optical OR Electrical
- 2 x 10 Gbps Optical only

GL Communications

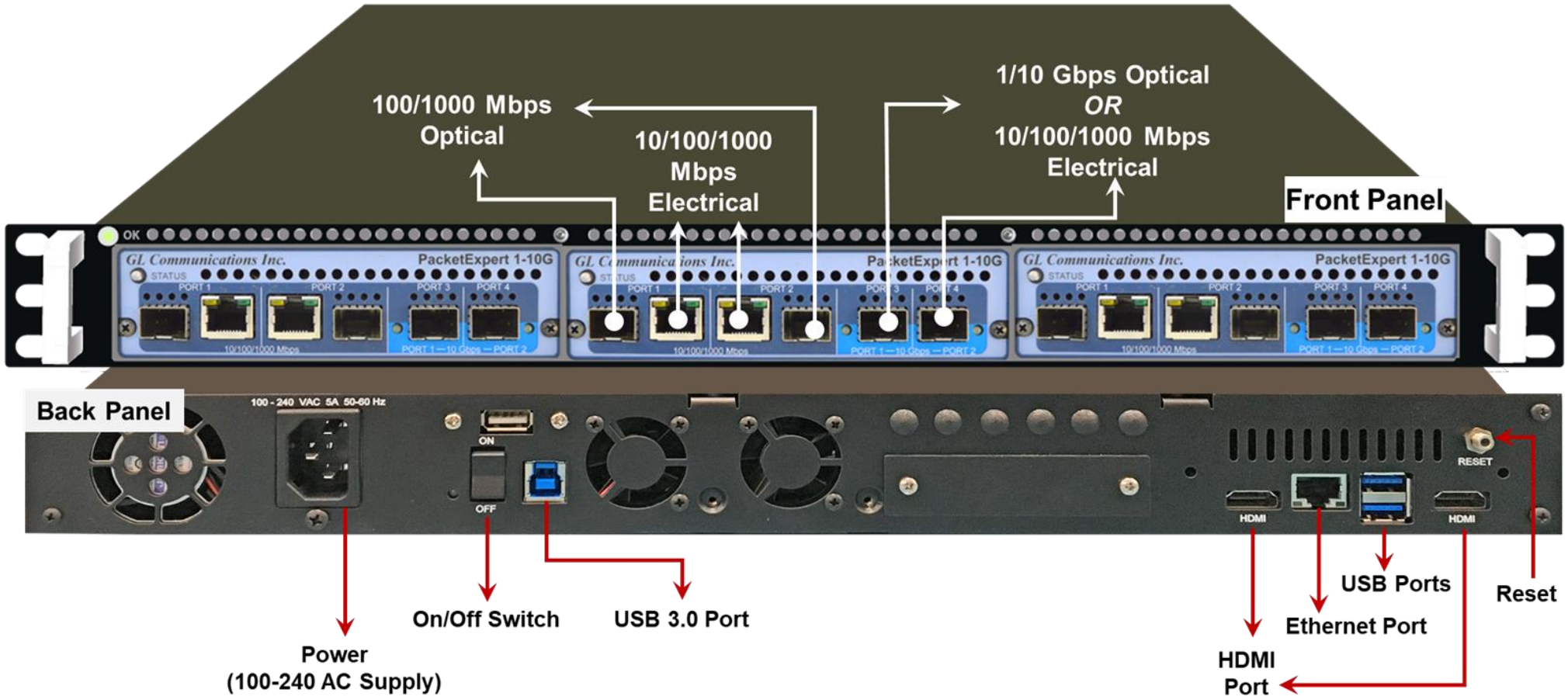
Hardware Overview + Highlights



IPNetSim™ is an optional application available within PacketExpert™ platforms (PXG100 and 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.

Hardware Interfaces



Application Overview

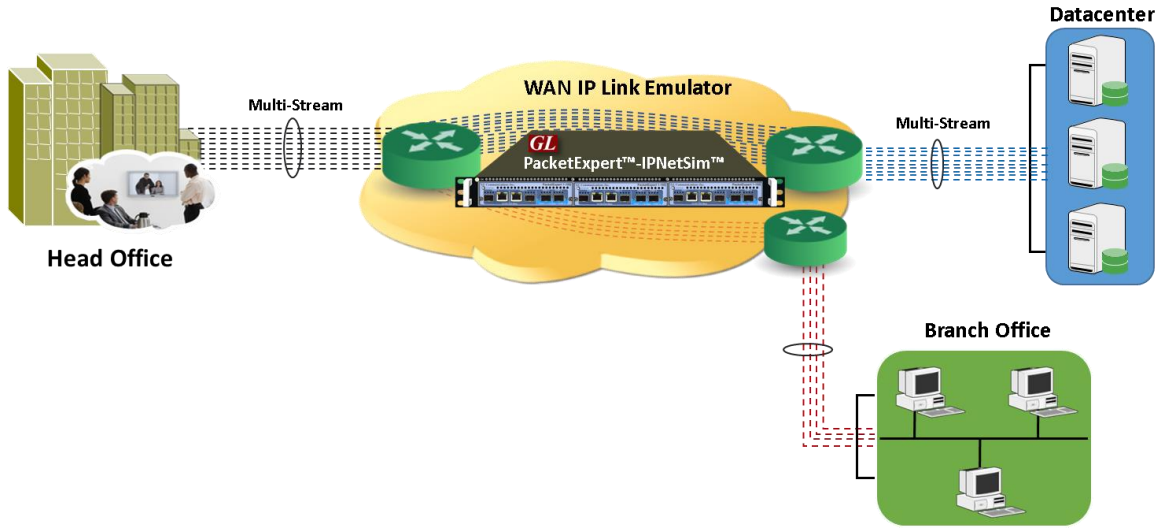
Application Overview

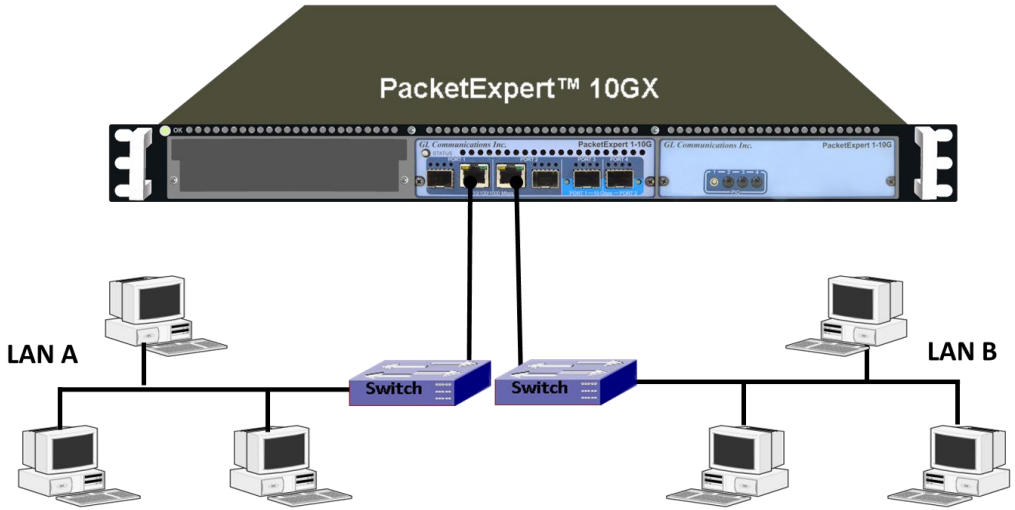
- Test Enterprise and Individual-level applications…
  - Audio and video streaming (VoIP, IMS, HDT, IPTV)
  - Storage services (Critical Data Access)
  - Cloud and web services
  - FTP / HTTP

- 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.

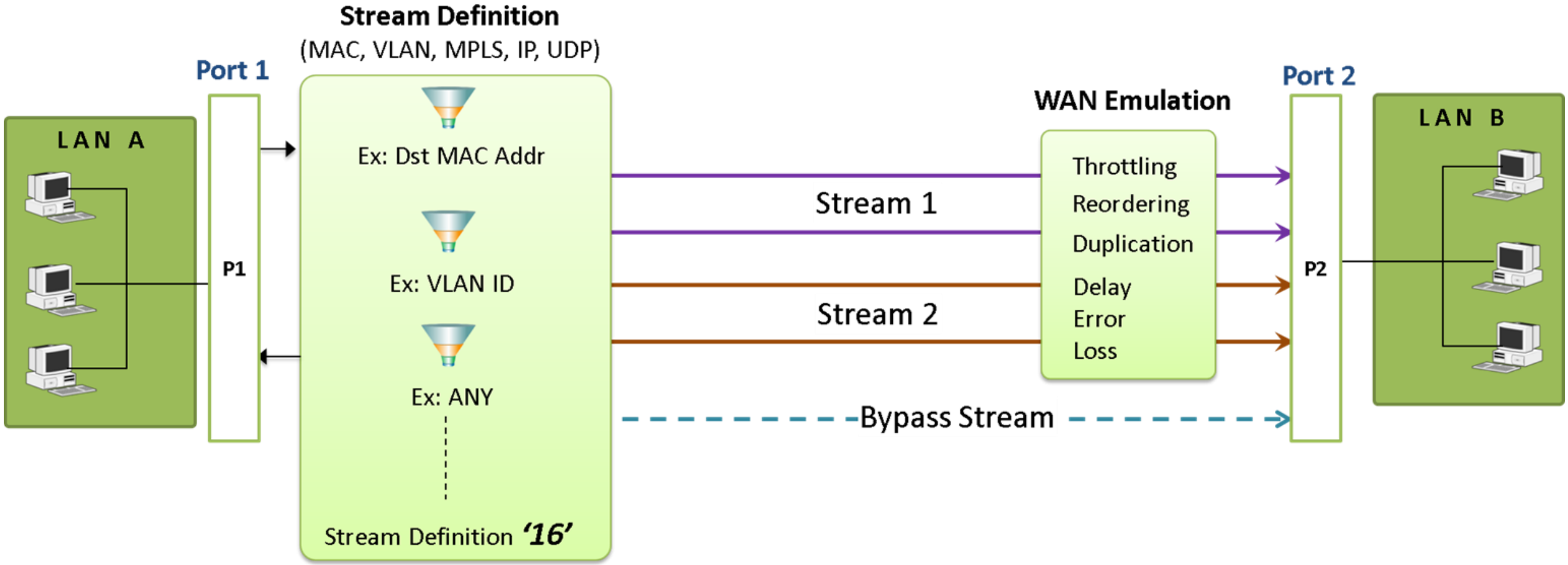




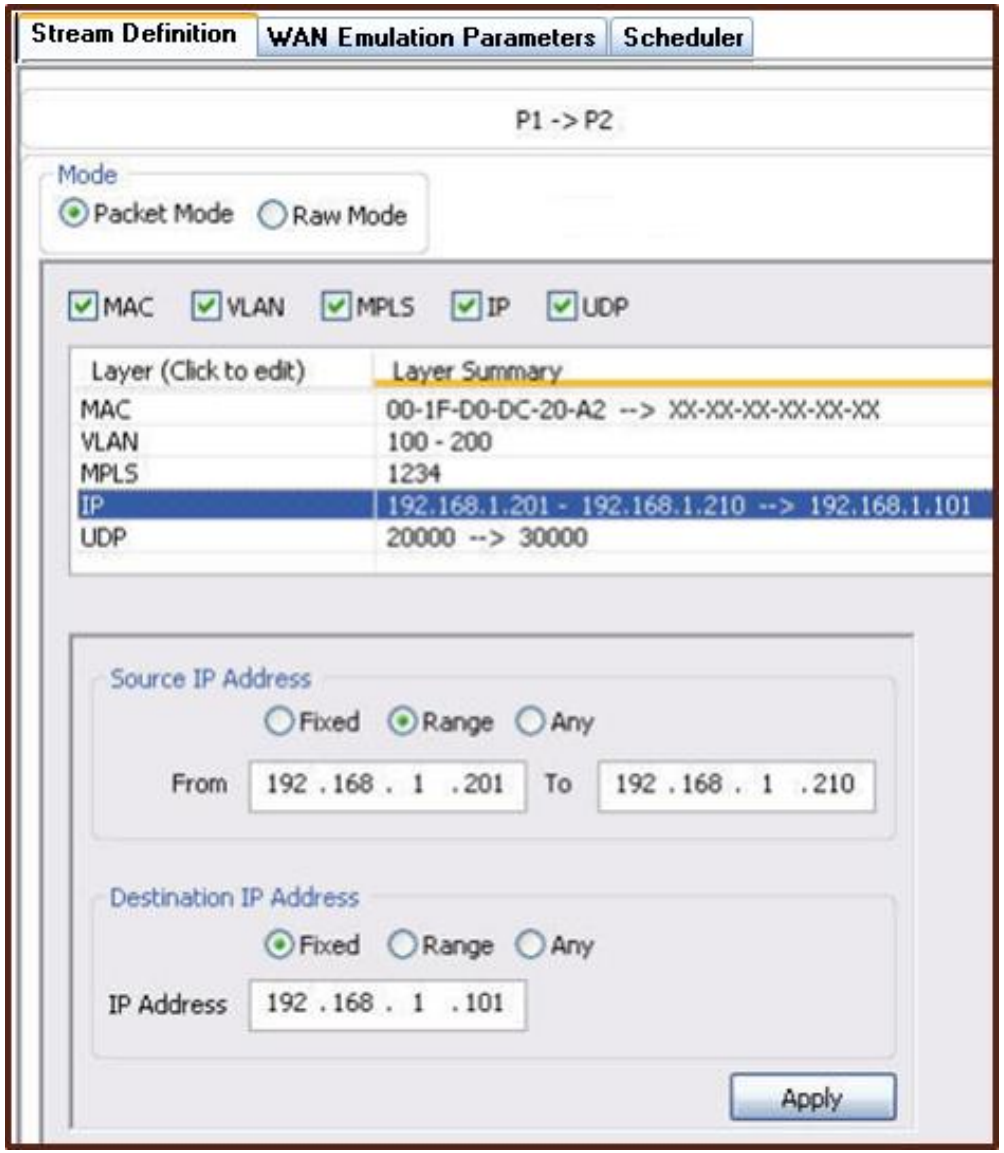
GL Communications

“Stream” Overview

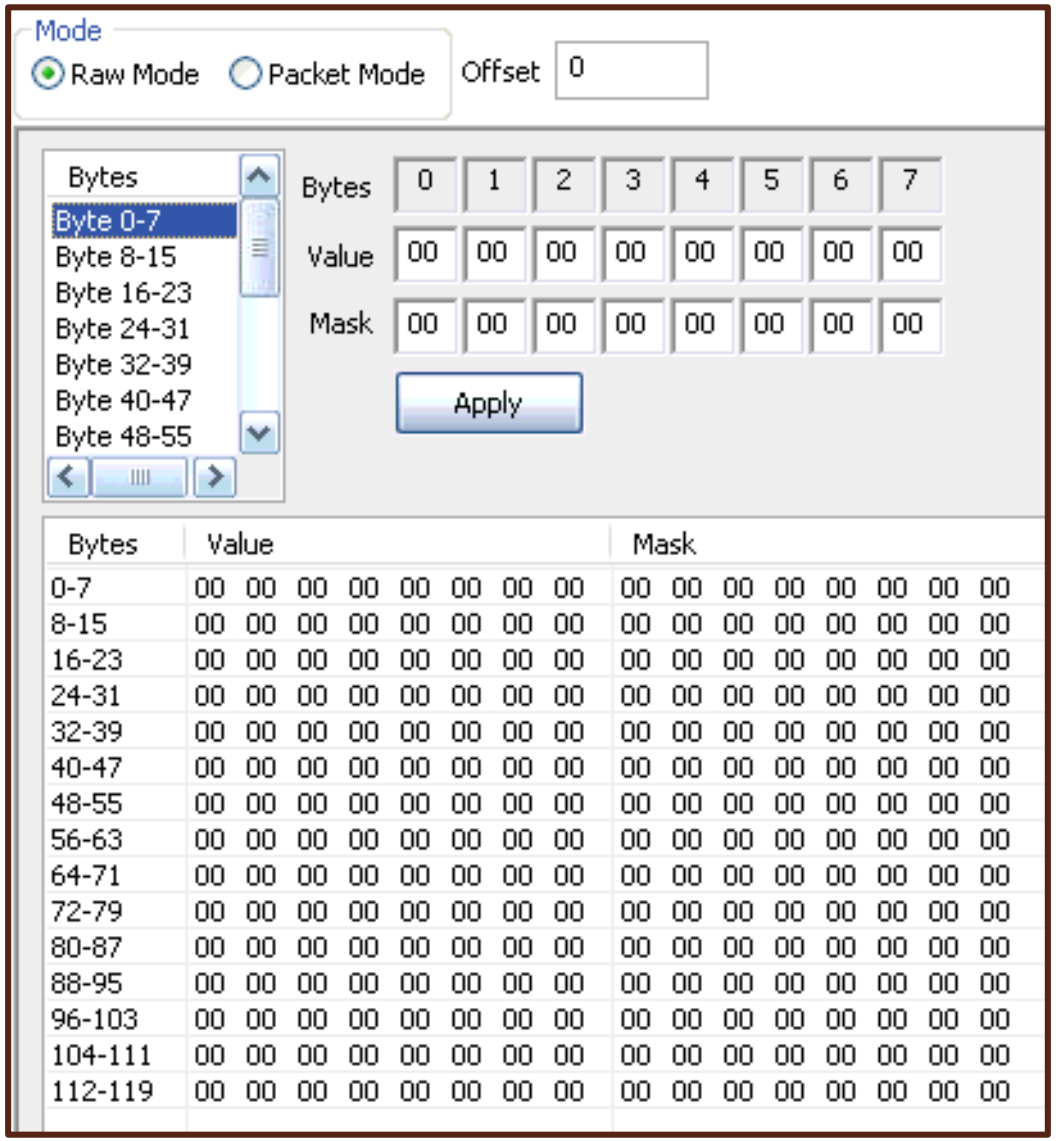
“Stream” Overview



Define Streams in Packet Mode



Define Streams in Raw Mode

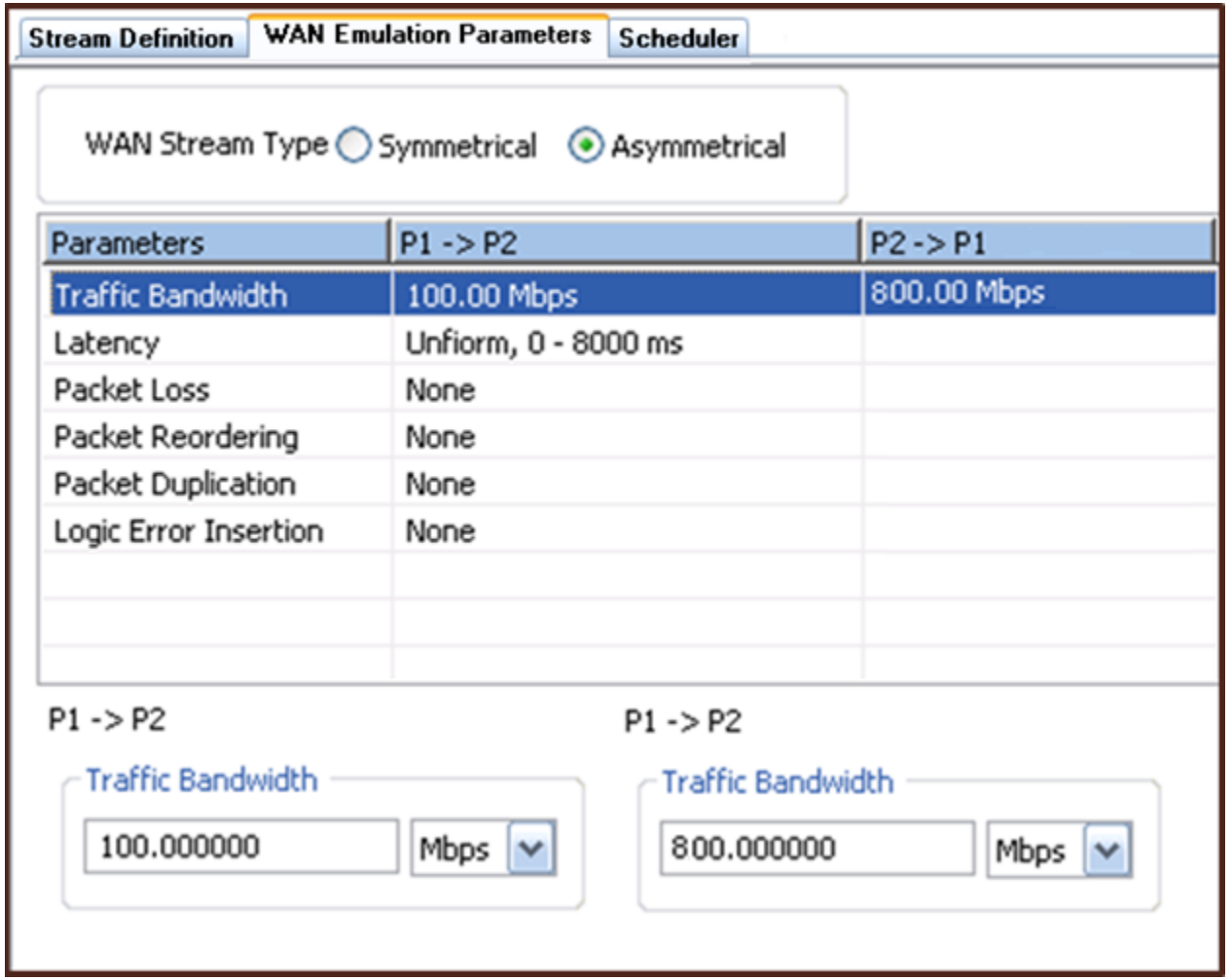


GL Communications

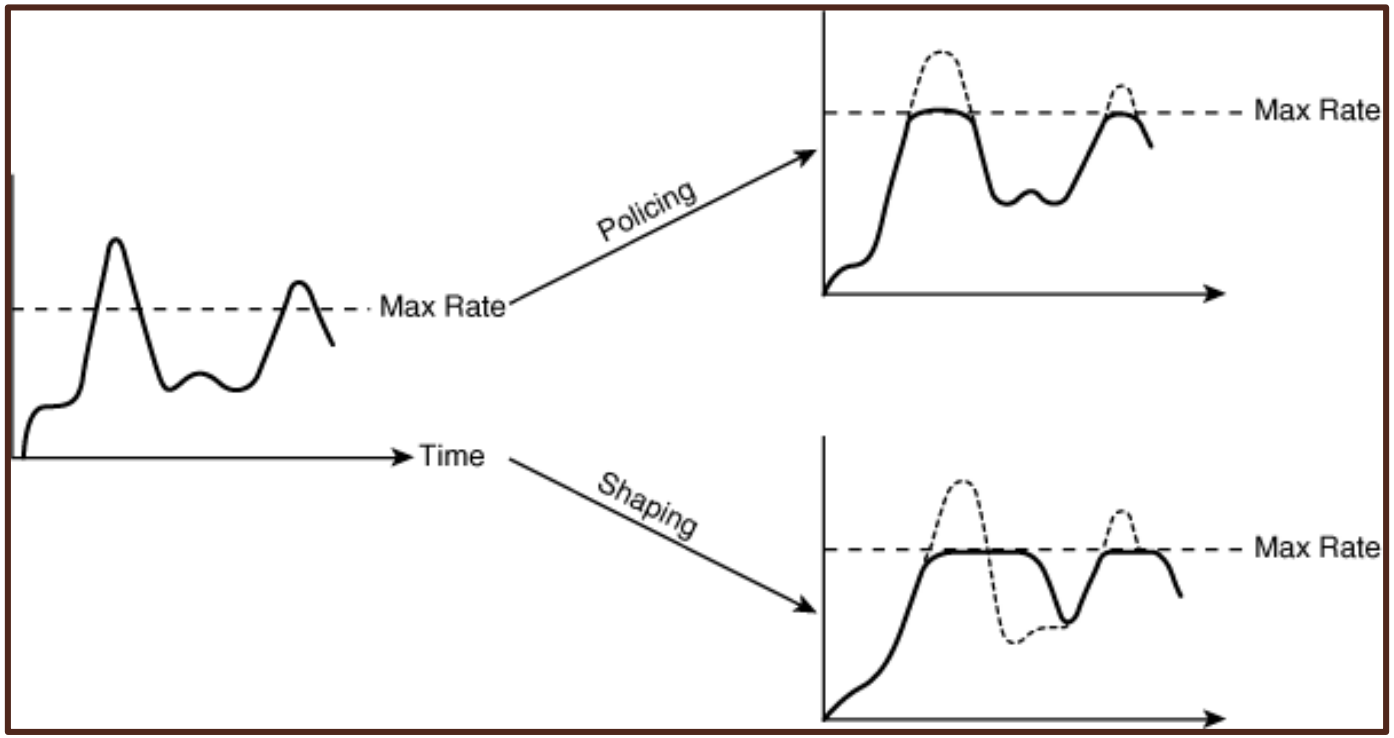
Impairments

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**

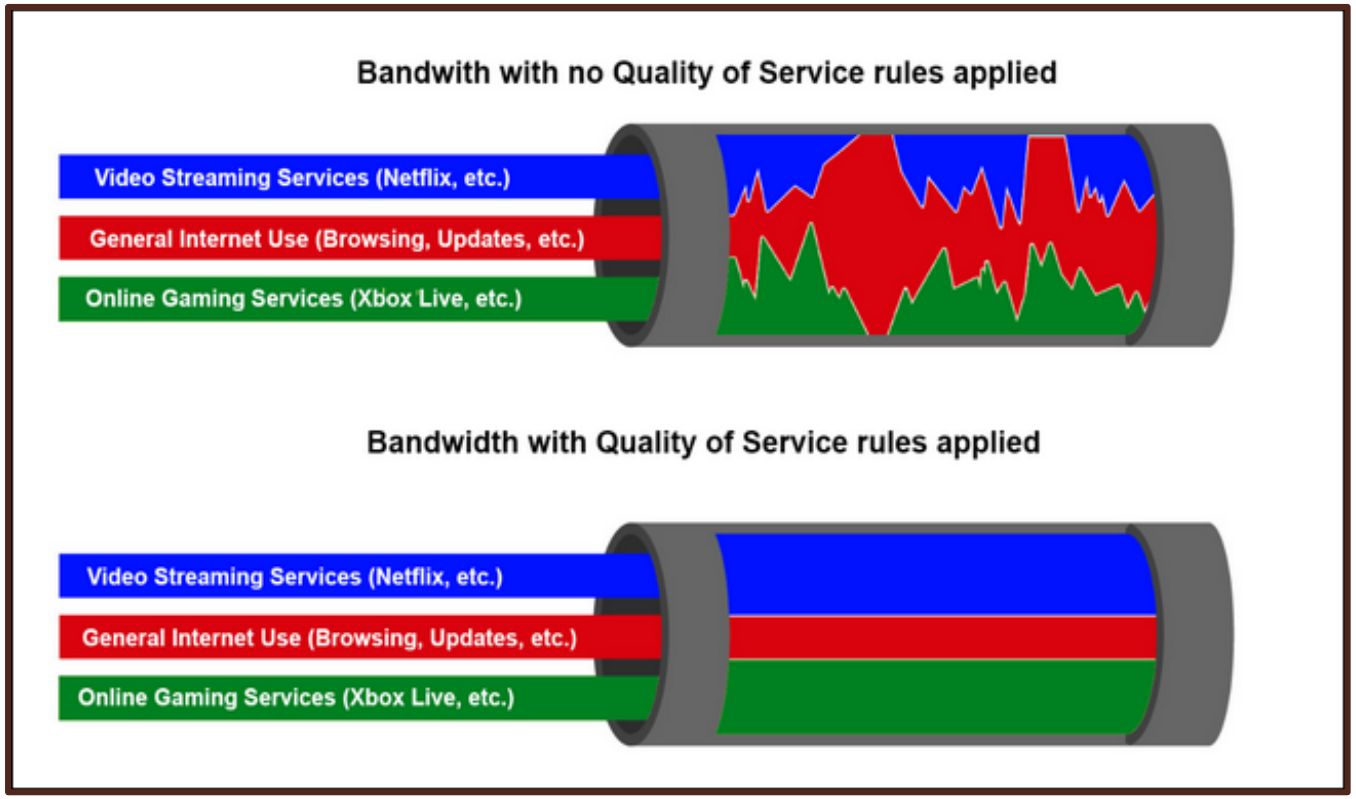


Traffic Bandwidth



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

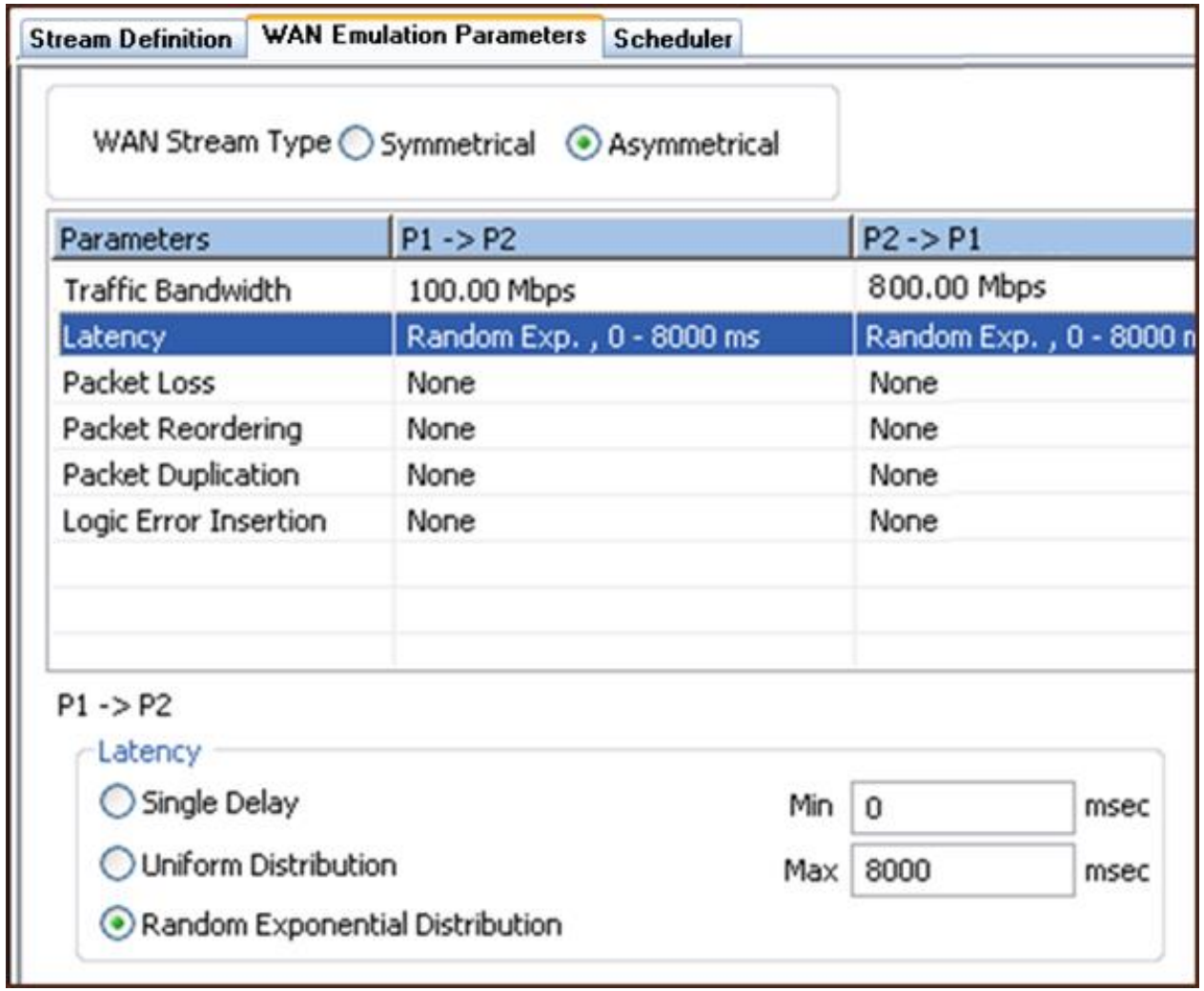
Traffic Bandwidth



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

Latency / Jitter

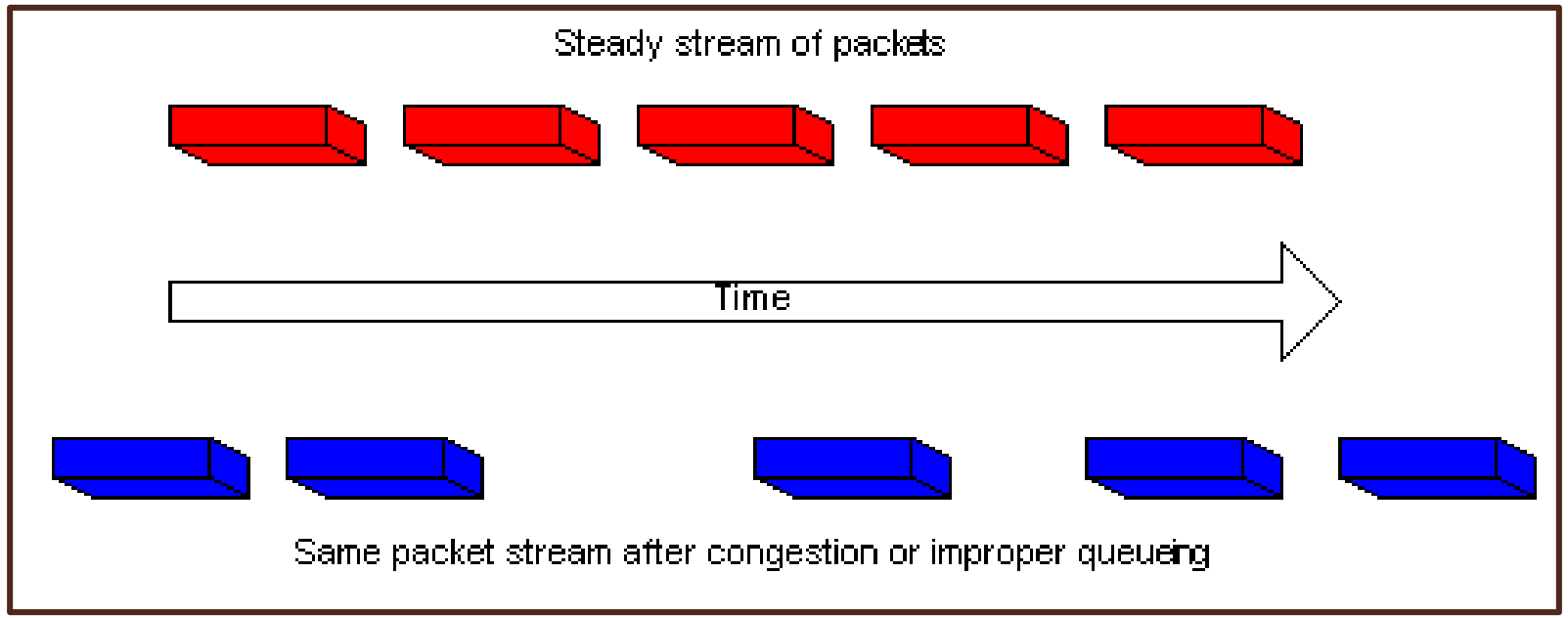
- Apply ***Static*** Delay, or a ***Uniform*** or ***Exponential*** distribution between a minimum and maximum
- Delay a packet **up to 8000 ms** in 1ms increments



Latency / Jitter

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

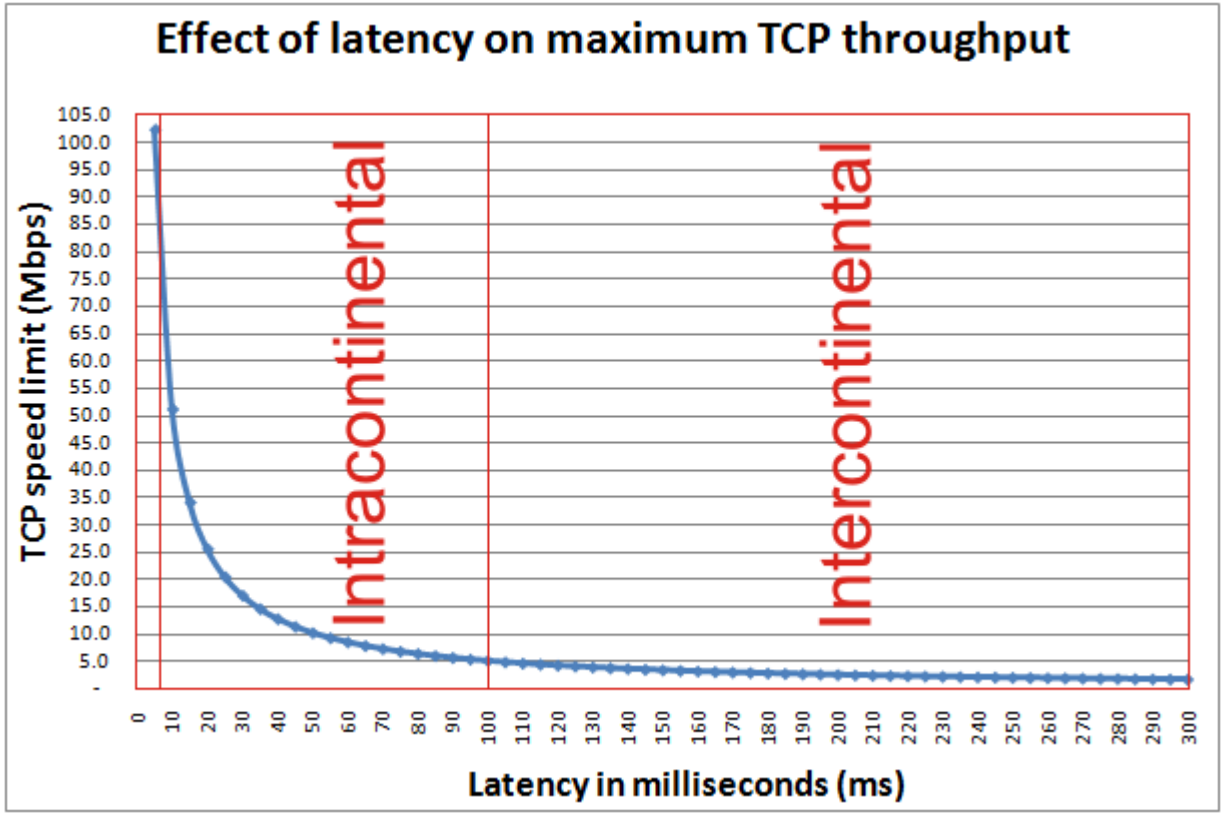
Latency / Jitter



- 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

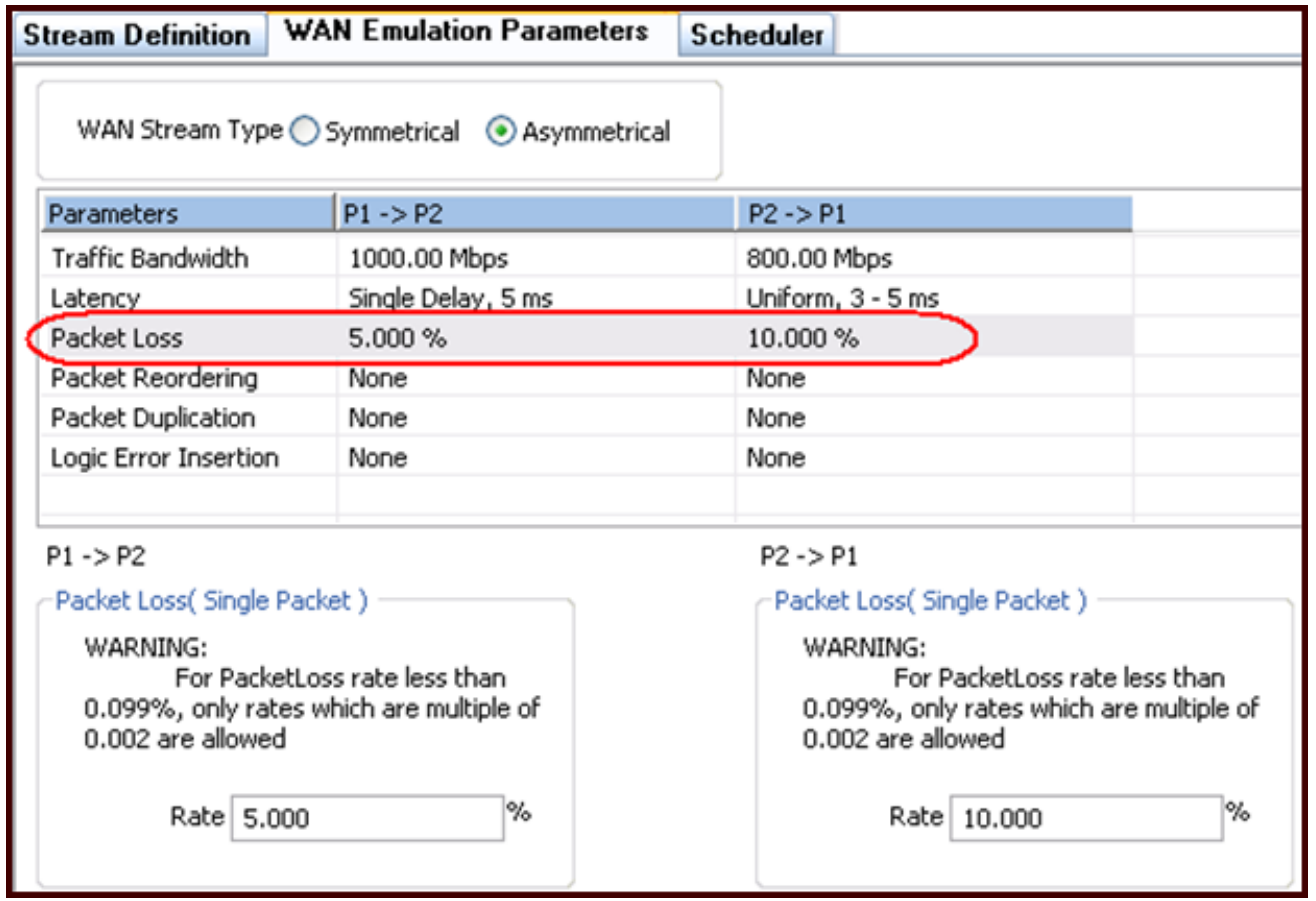
Latency / Jitter

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



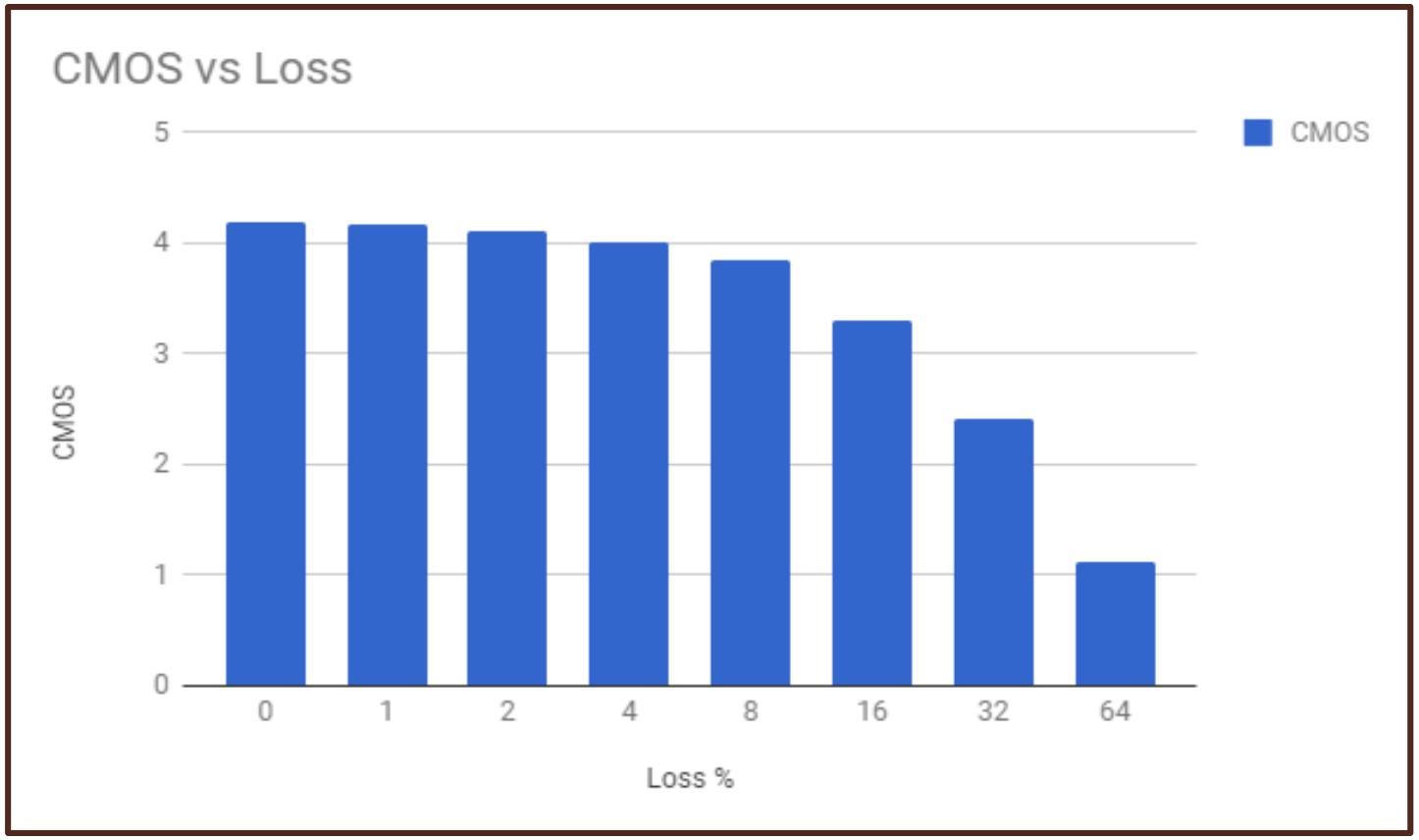
Packet Loss

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



Packet Loss

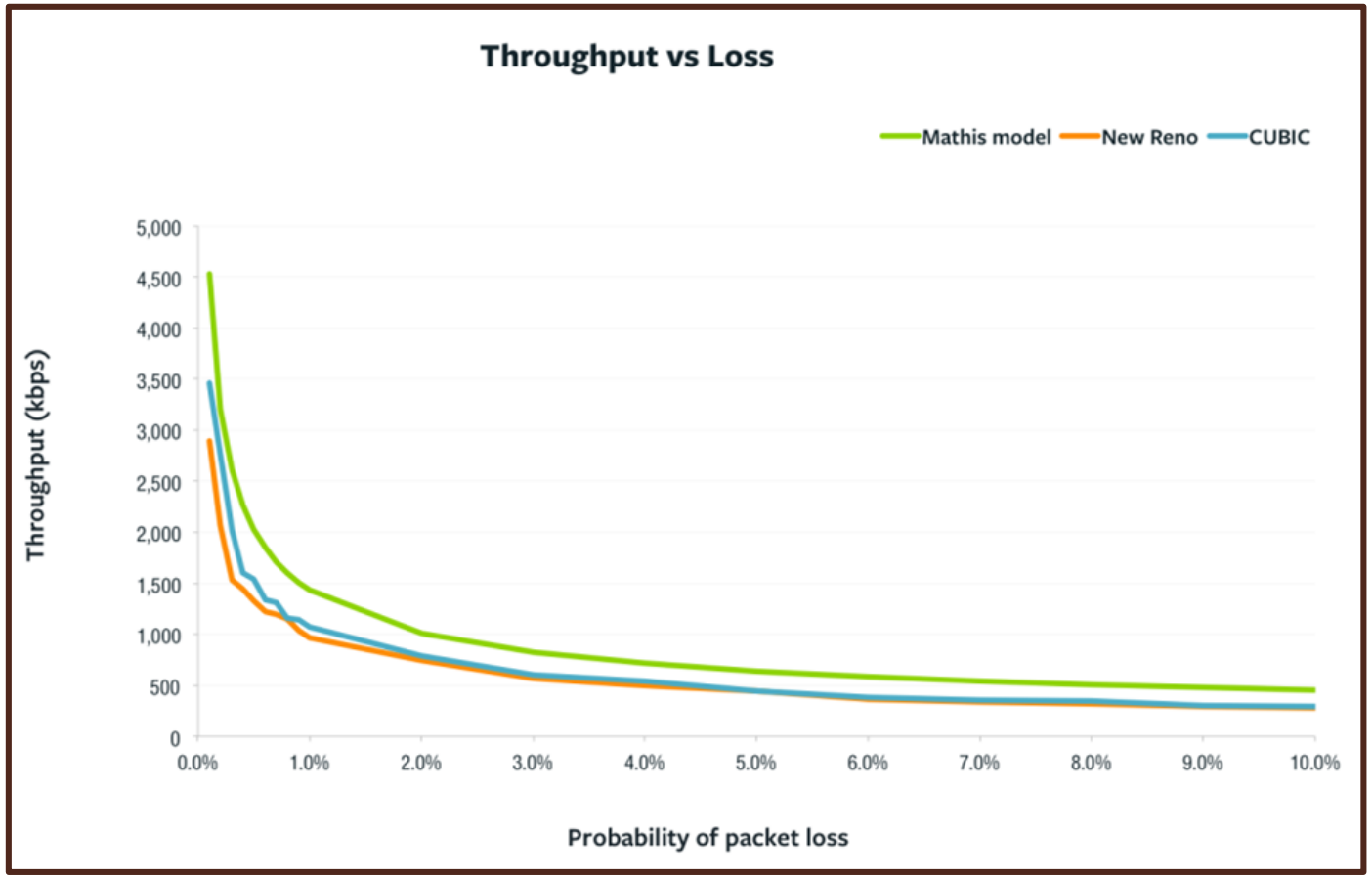
- Real Time UDP Applications are resilient to minor loss, but vulnerable to heavy loss



GL Communications

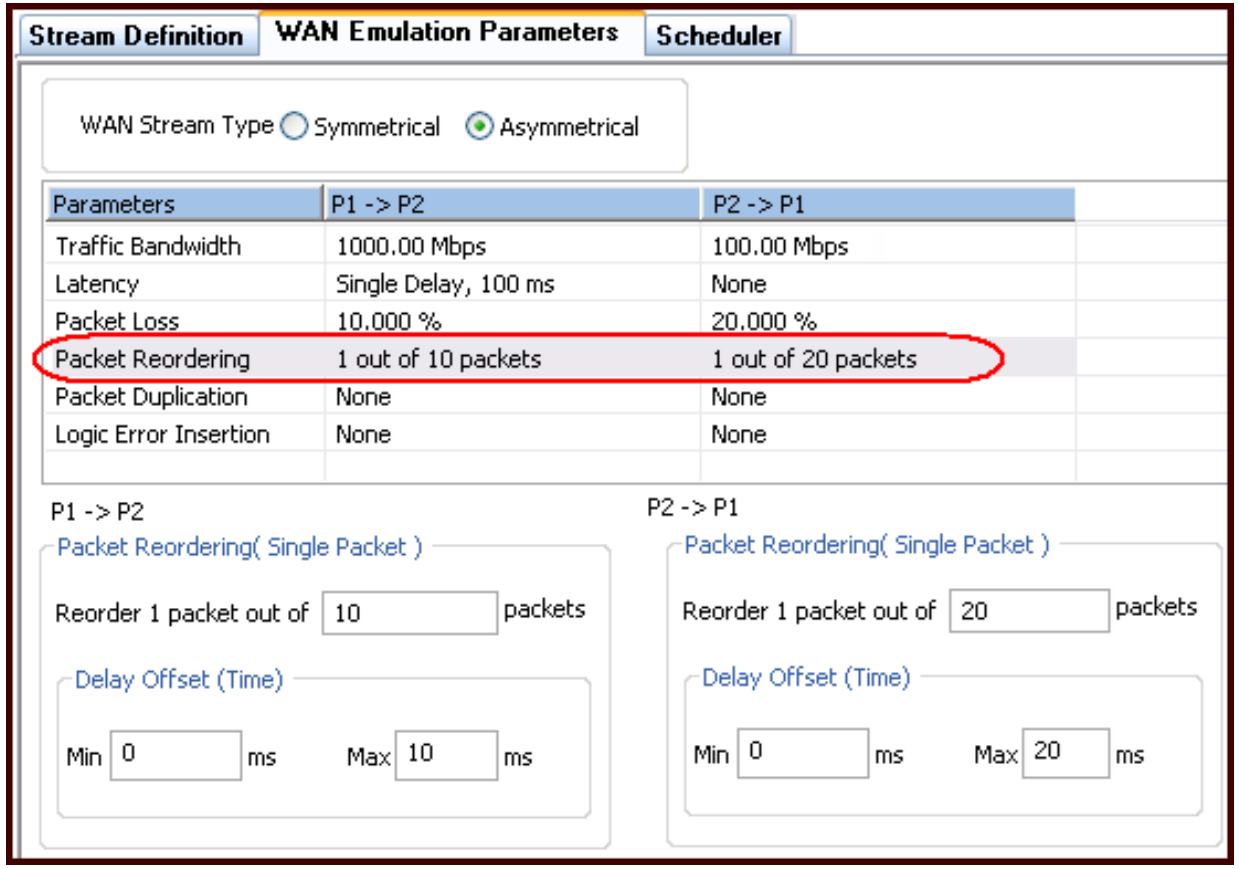
Packet Loss

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



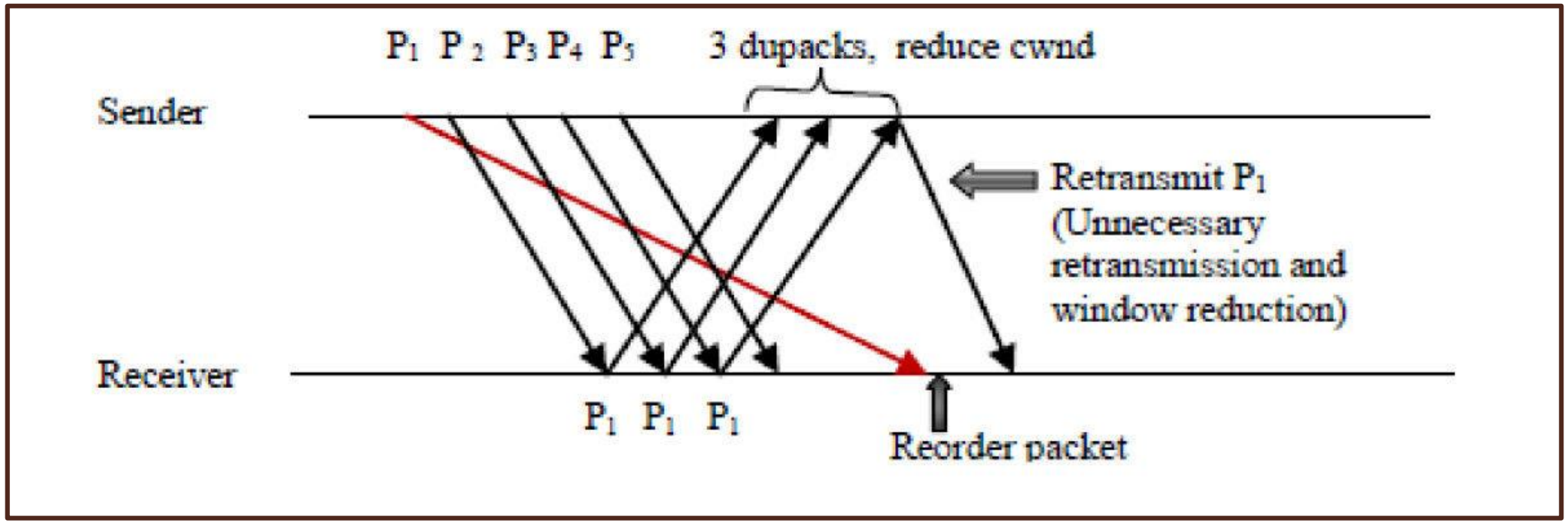
Packet Reordering

- 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



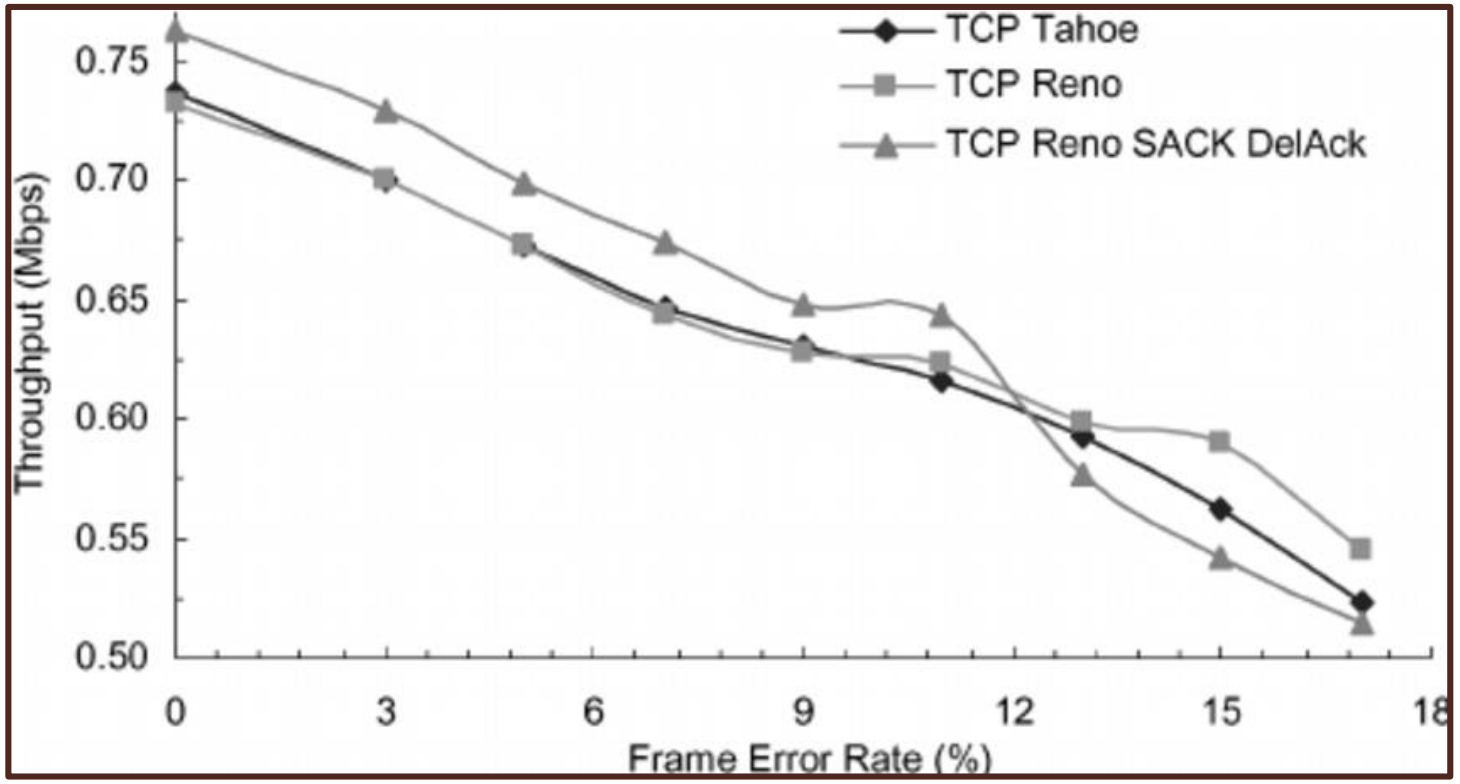
GL Communications

Packet Reordering



- 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

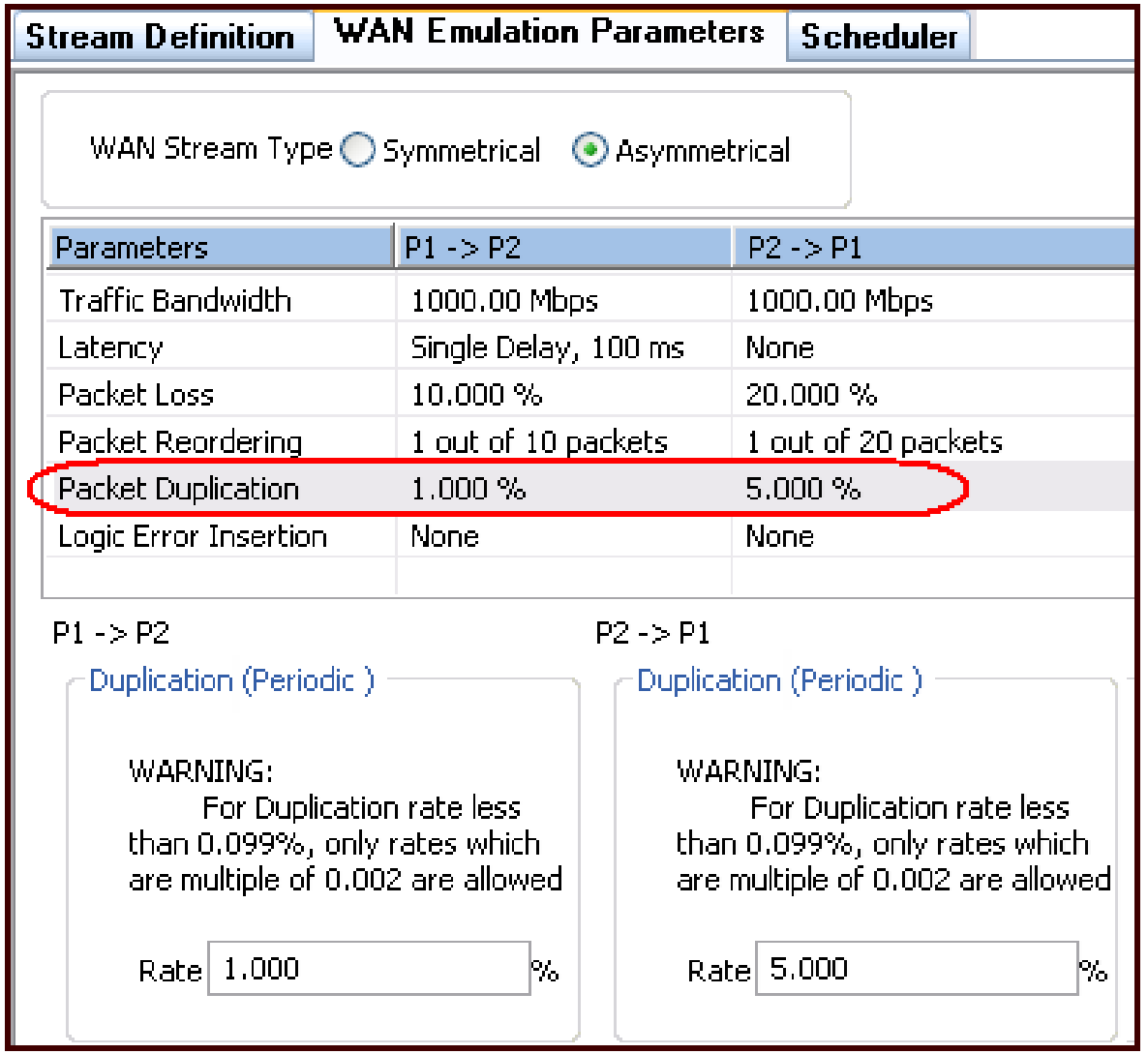
Packet Reordering



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

Packet Duplication

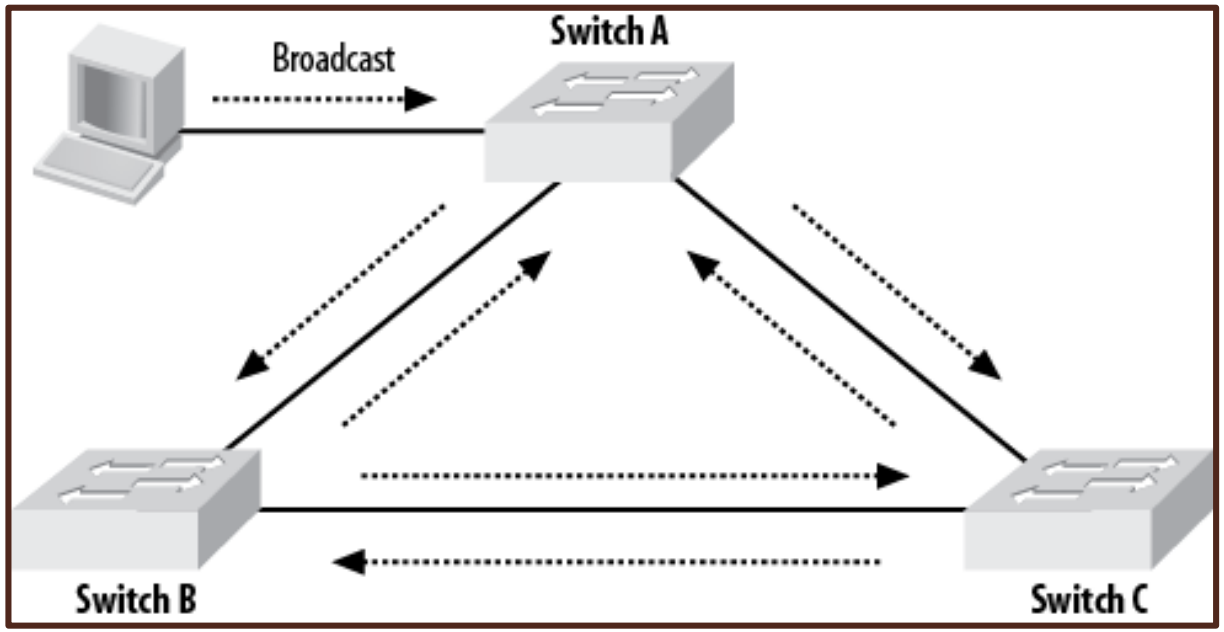
- Randomly duplicate from 0.01% to 100% of all Packets in the stream
- Emulate WAN applications where multiple paths are possible and Load Balancing may be present



GL Communications

Packet Duplication

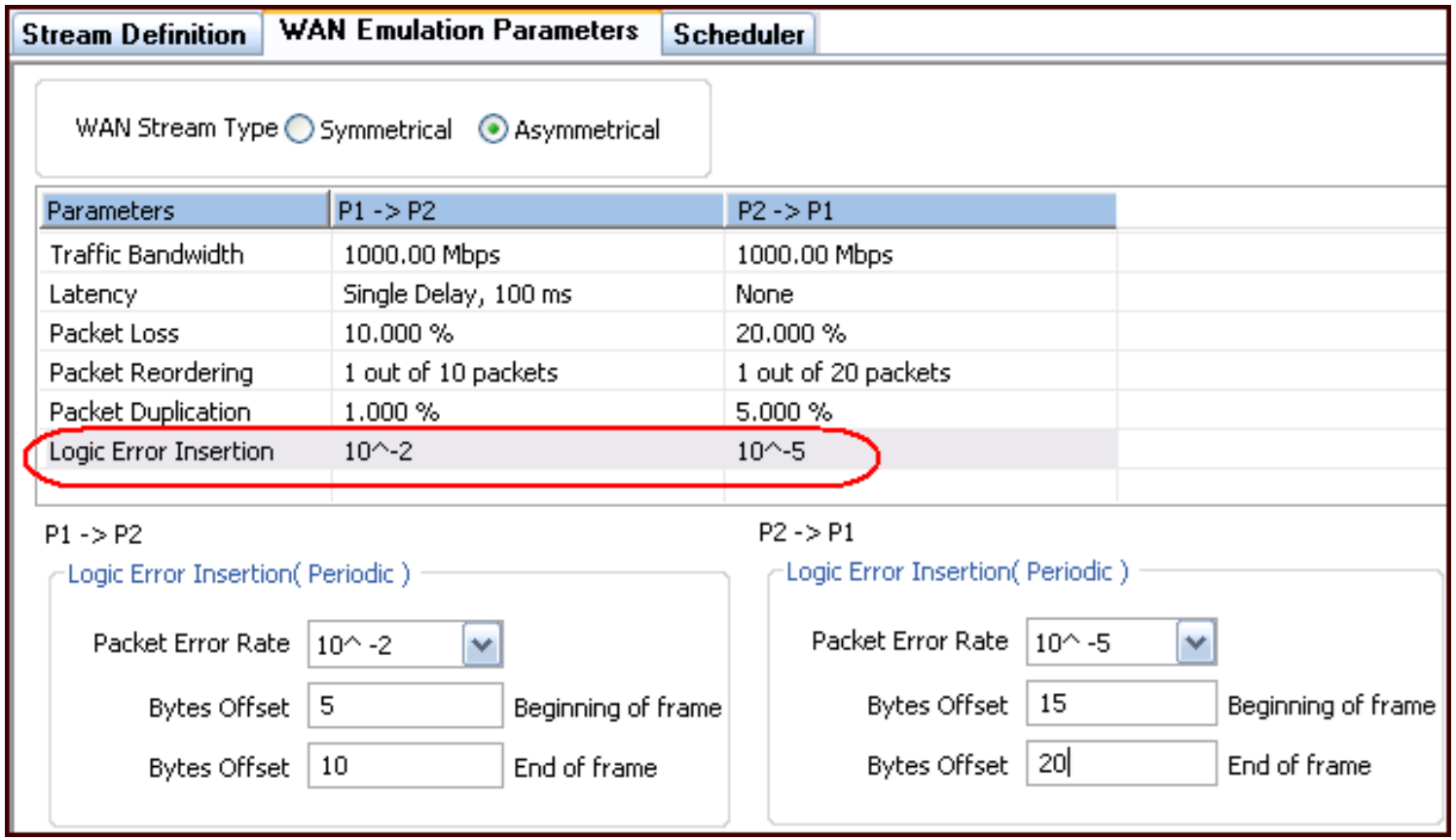
- Duplication can be fatal in broadcast situations (ie broadcast storm)
- Similarly dangerous in multicast applications where small network misconfigurations can have disproportionately large consequences
- Watch out for this in multipath Spanning Tree networks



GL Communications

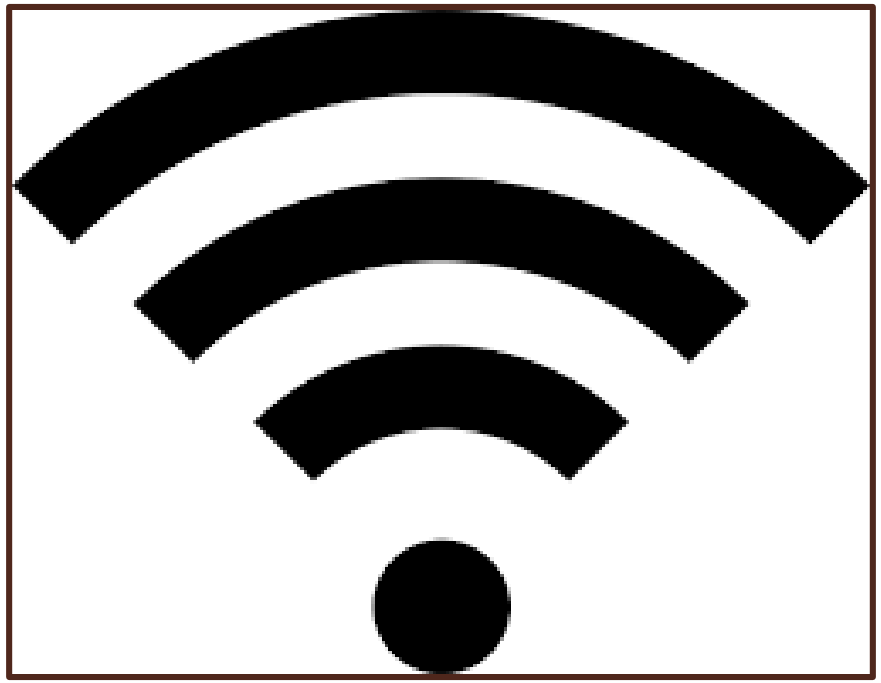
Logic Error Insertion

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



GL Communications

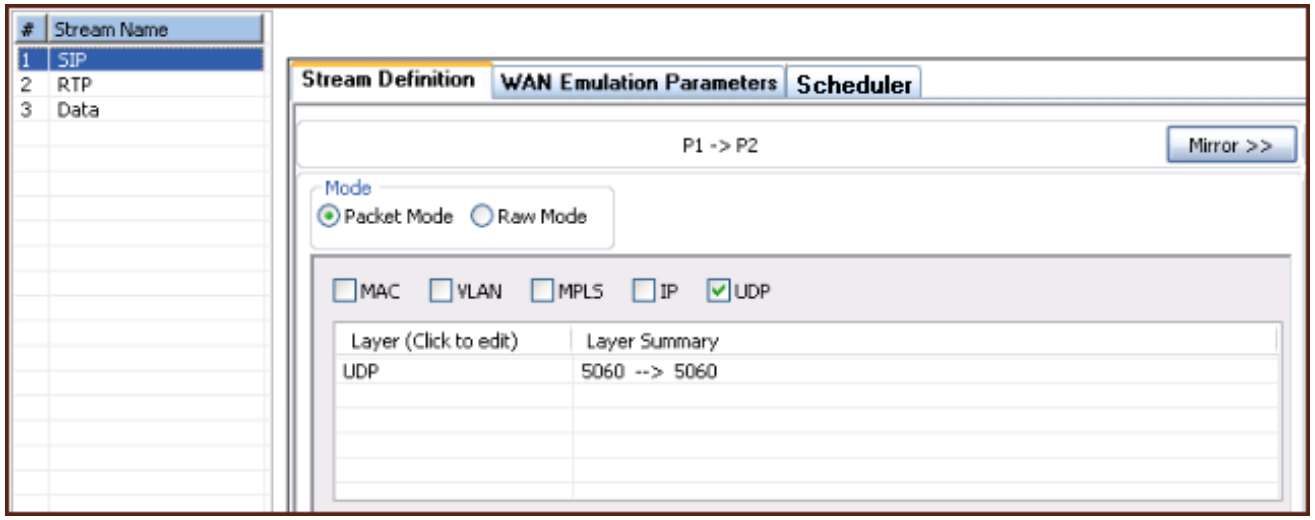
Logic Error Insertion



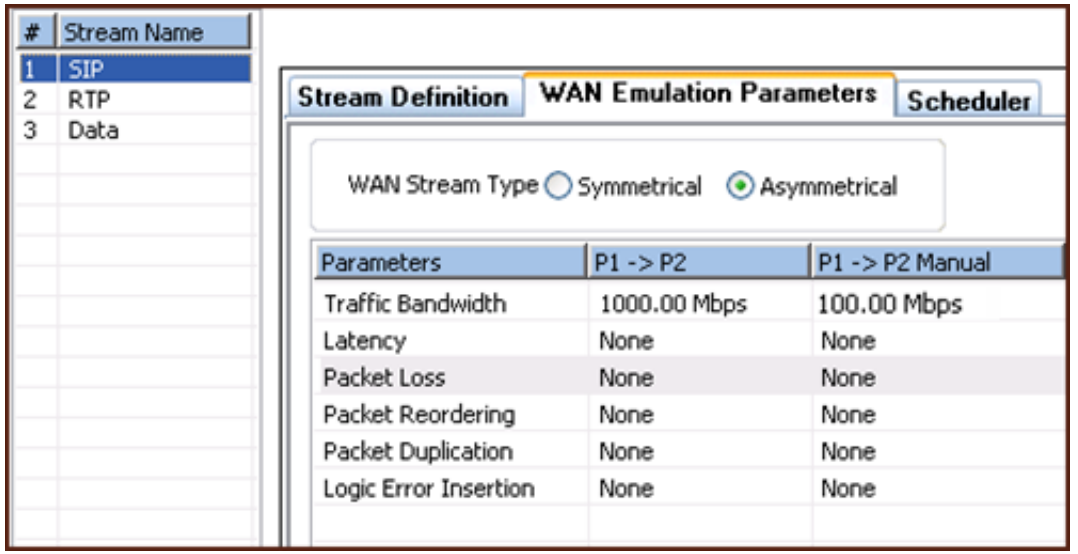
- Cellular and WiFi links are very prone to bit errors (as well as latency and bandwidth issues)

Application Examples

Application Example: VoIP

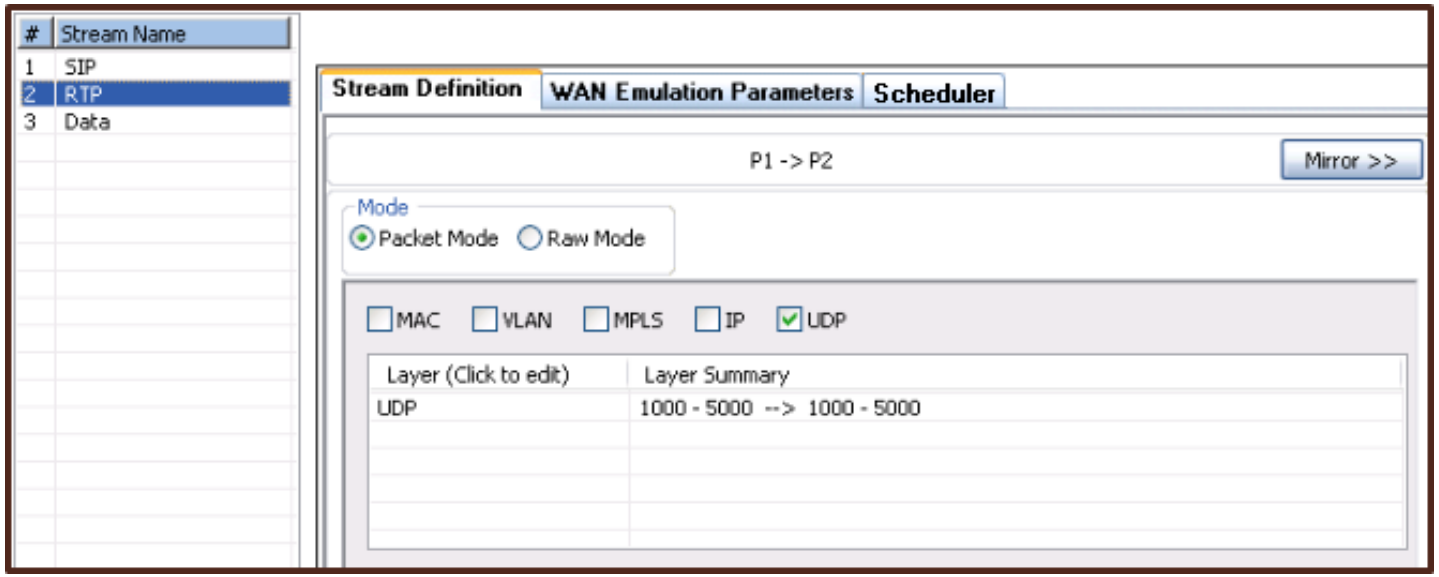


- Configure SIP packets to be completely unimpaired

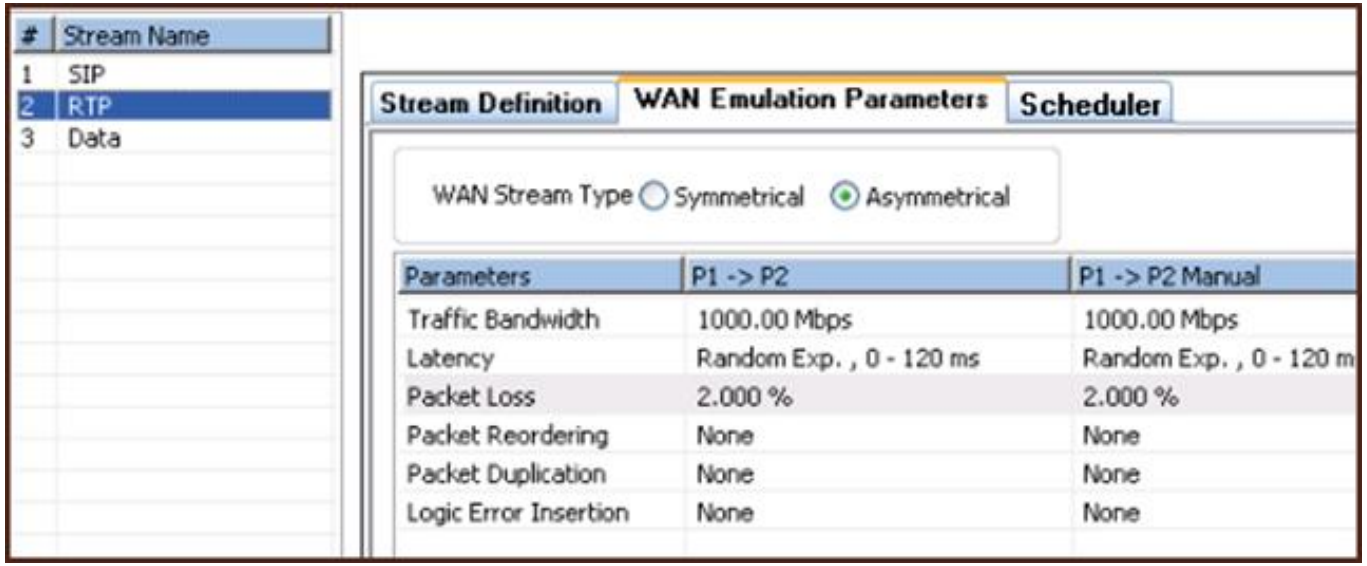


GL Communications

Application Example: VoIP

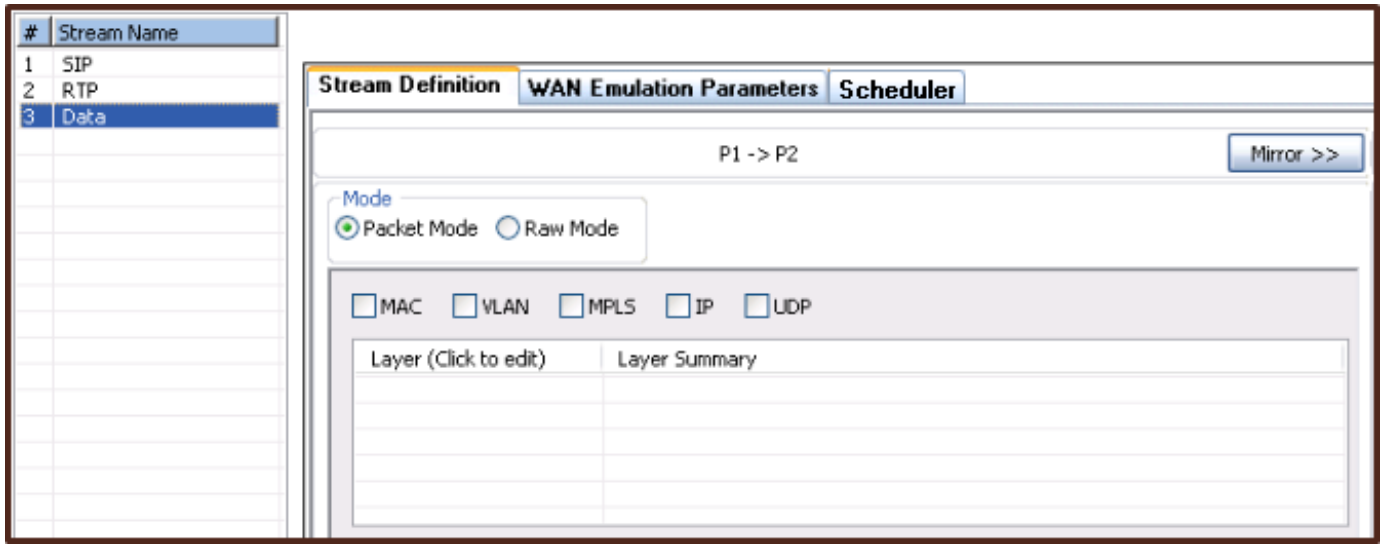


- Apply loss and jitter to RTP streams

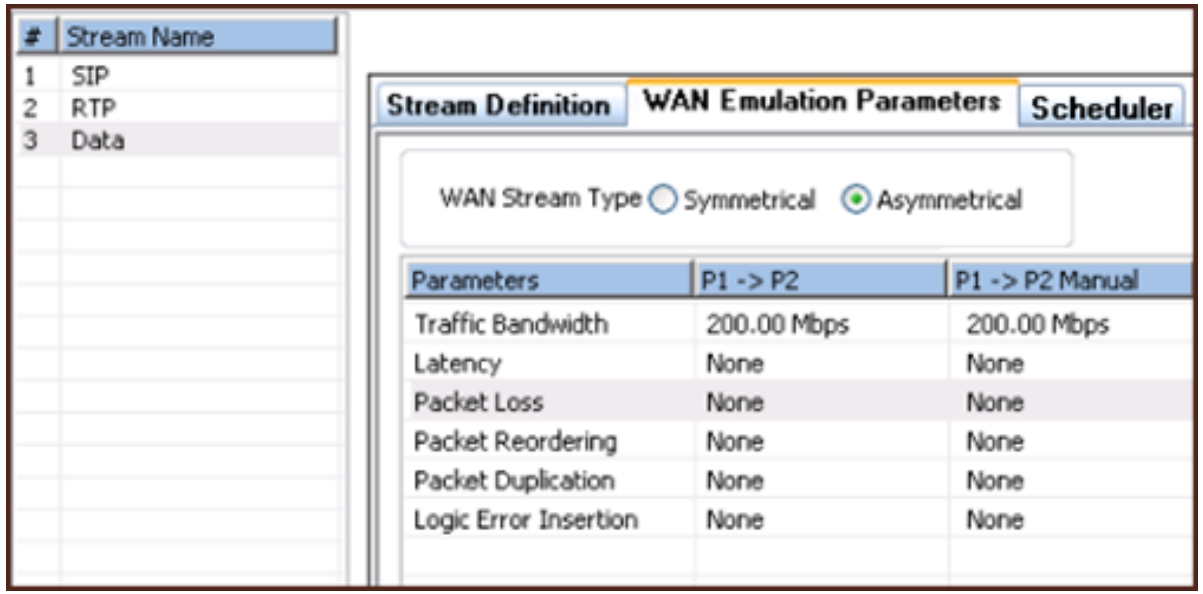


GL Communications

Application Example: VOIP

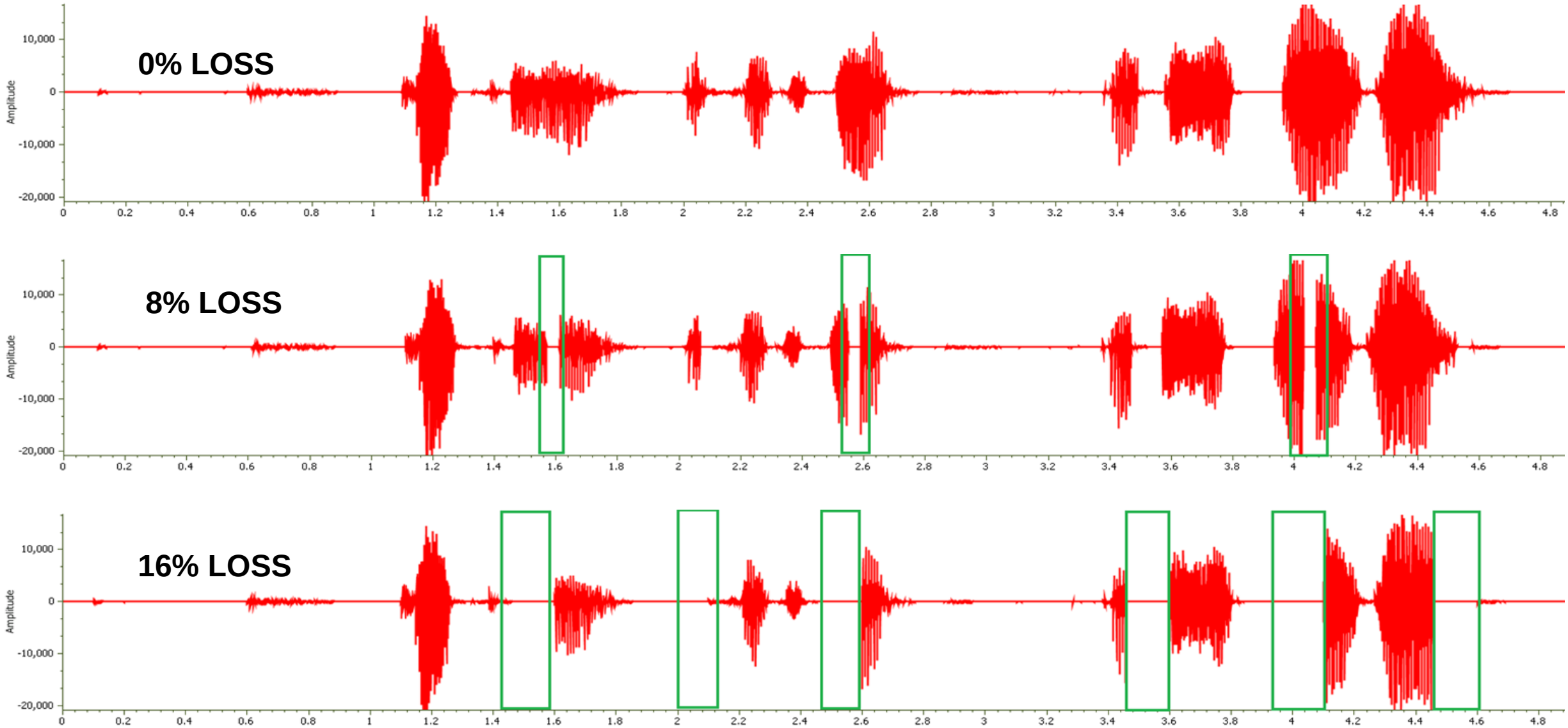


- Set an SLA style bandwidth cap on Data



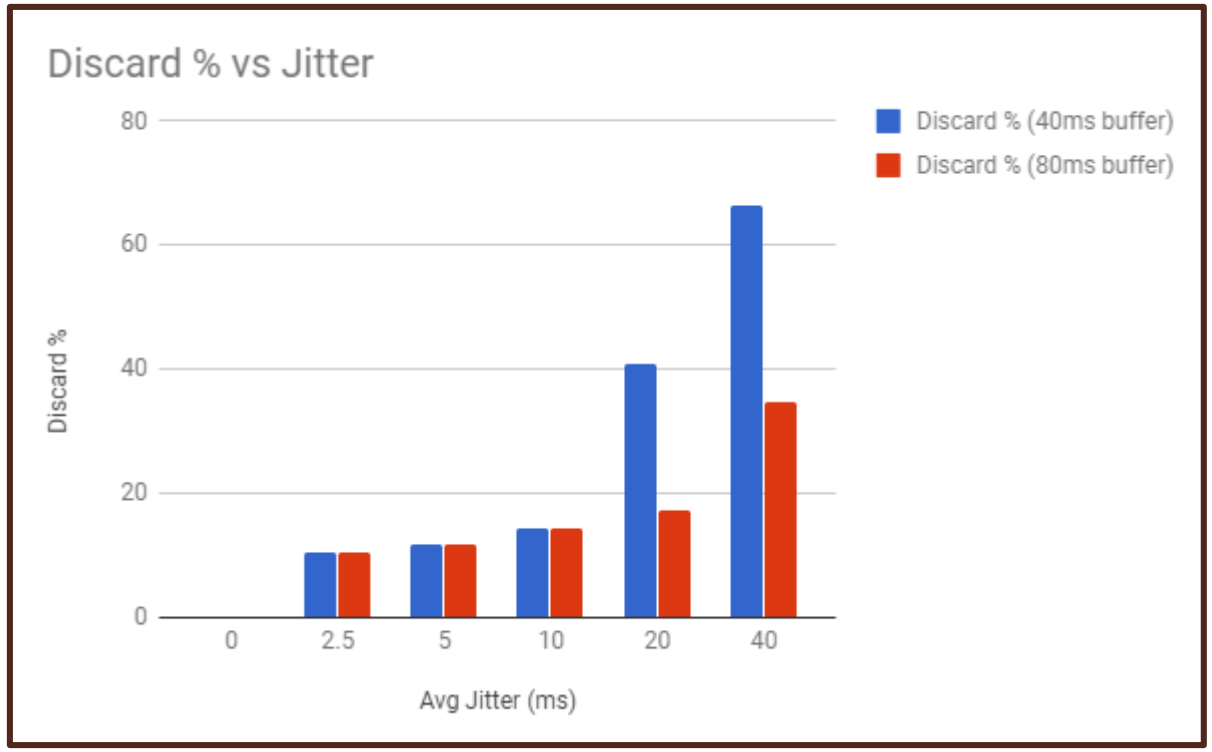
GL Communications

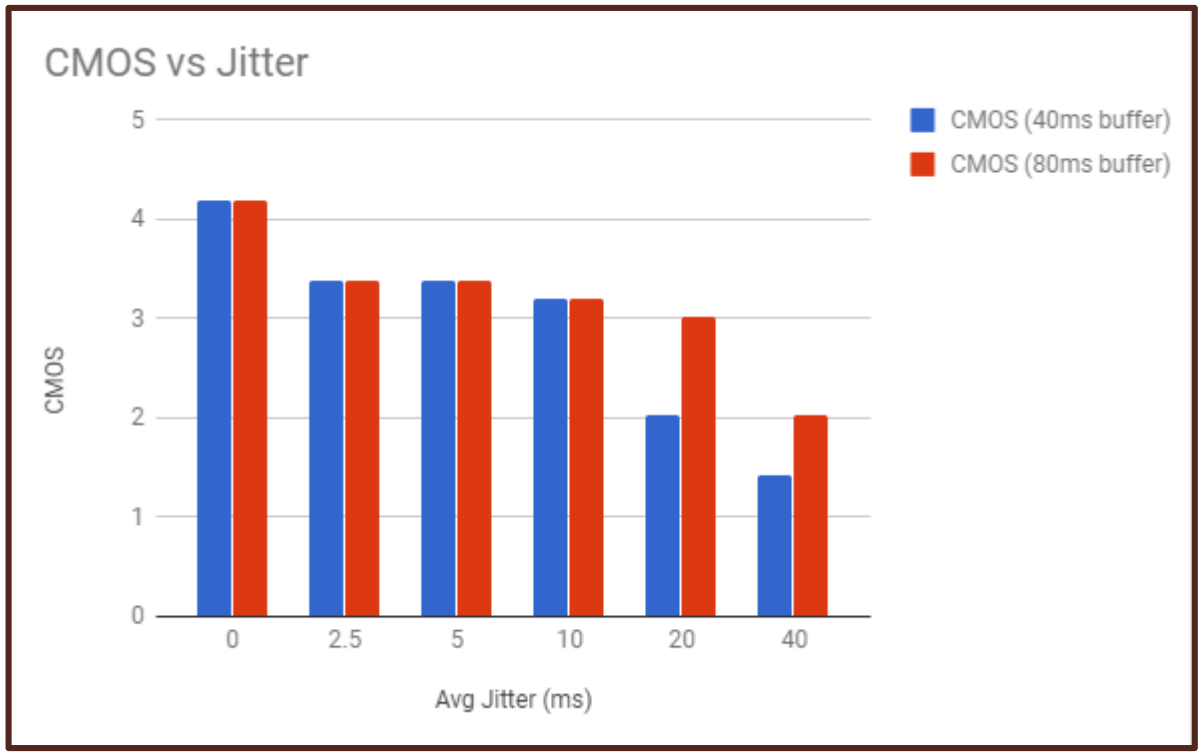
Application Example: VoIP



GL Communications

Application Example: VOIP

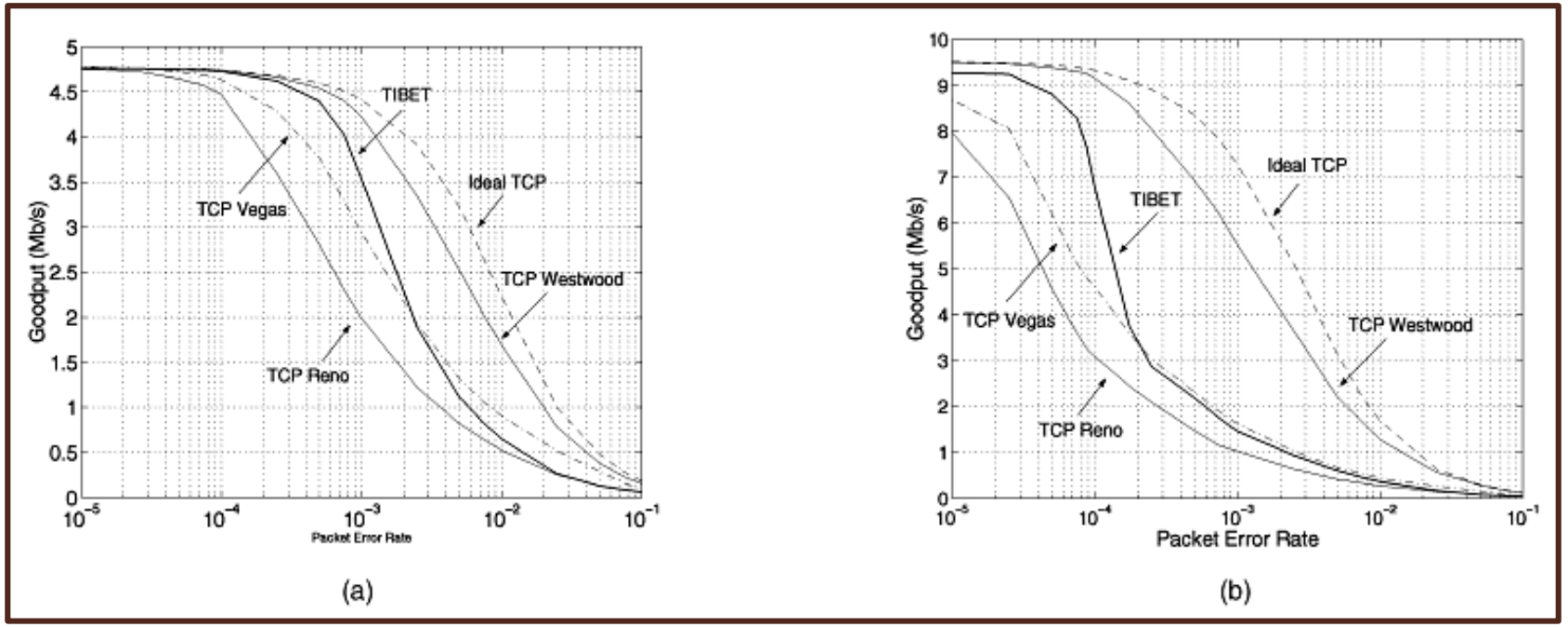




- Determine how your application will behave under expected (and unexpected) network conditions
- Determine what codecs you should use, what jitter buffers, etc.

GL Communications

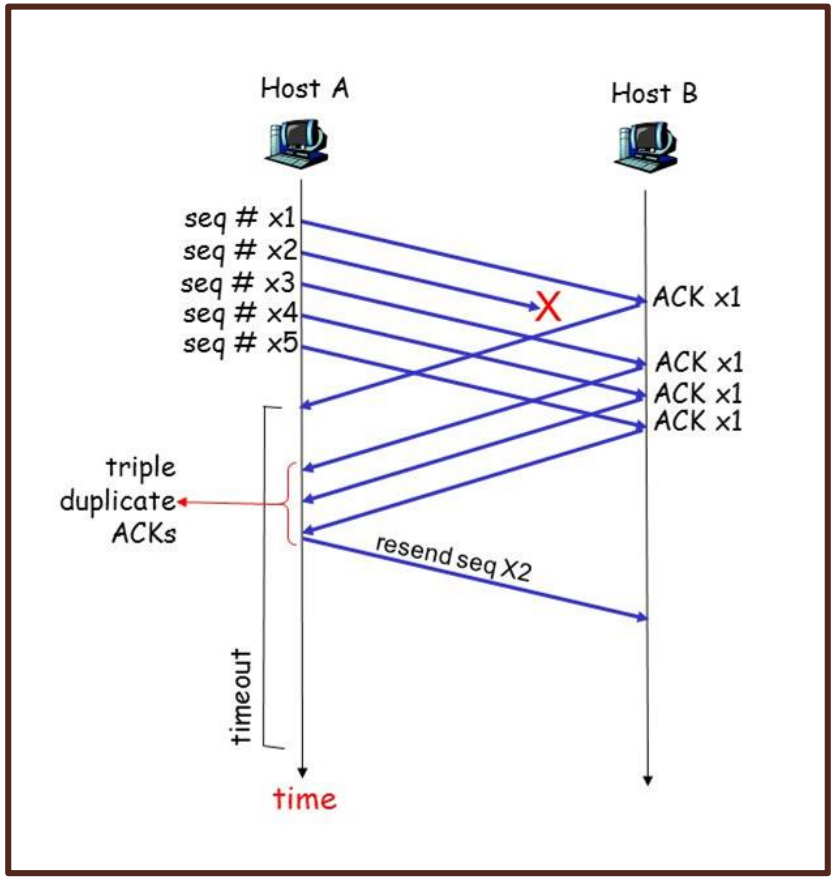
Application Example: TCP Over Wireless



- 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

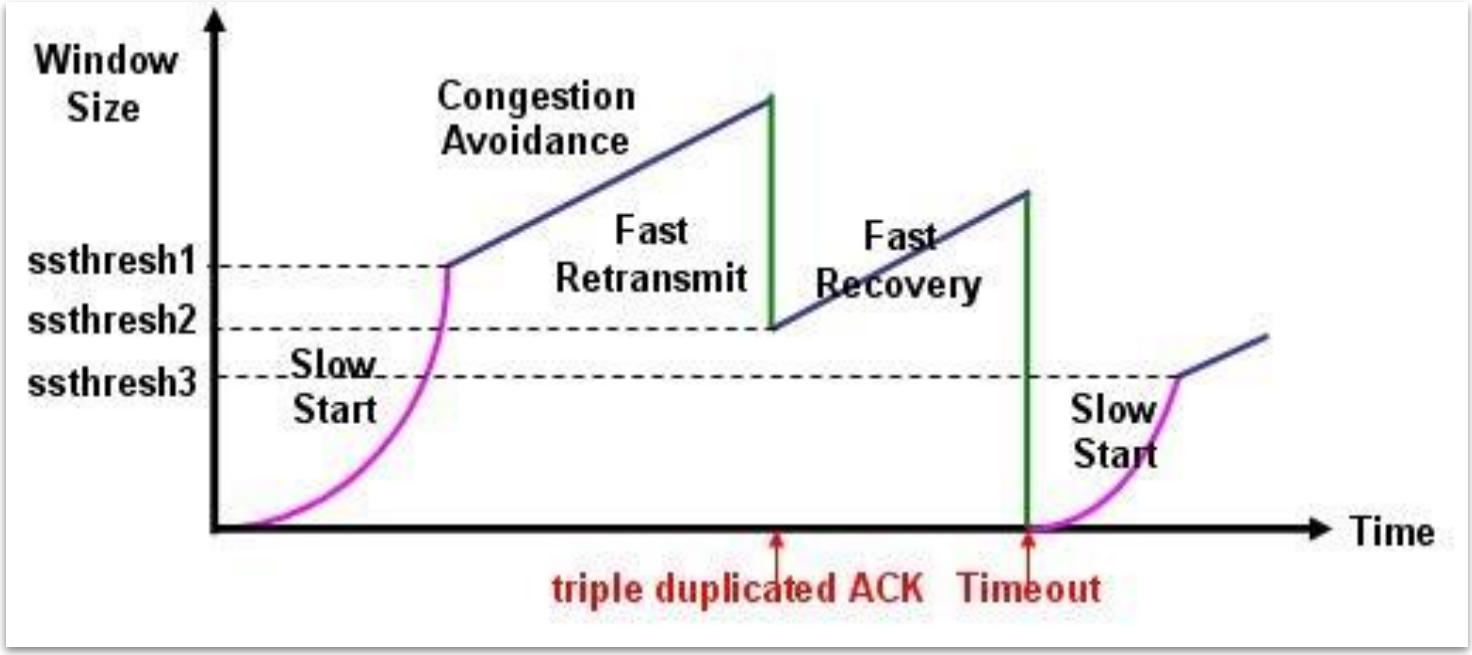
GL Communications

Application Example: TCP Over Wireless



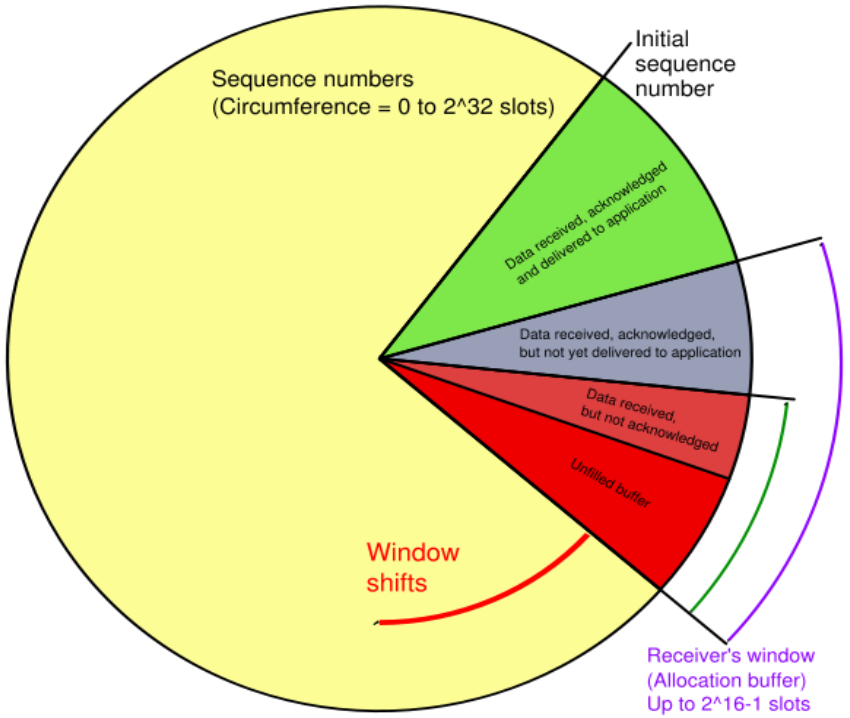
- Loss in a TCP application leads to duplicate ACKs, which lead to retransmissions

Application Example: TCP Over Wireless



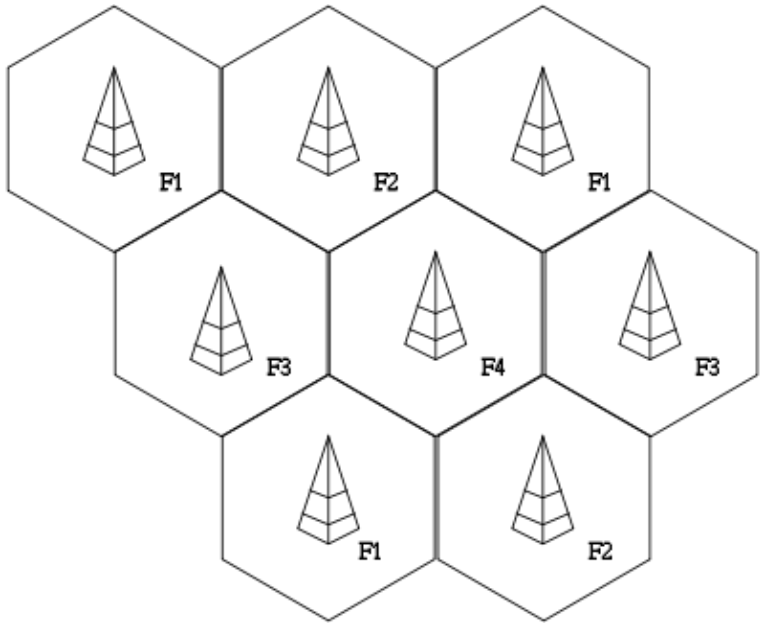
- TCP assumes that Loss is due to Congestion
- When Loss occurs TCP automatically cuts throughput to avoid congestion

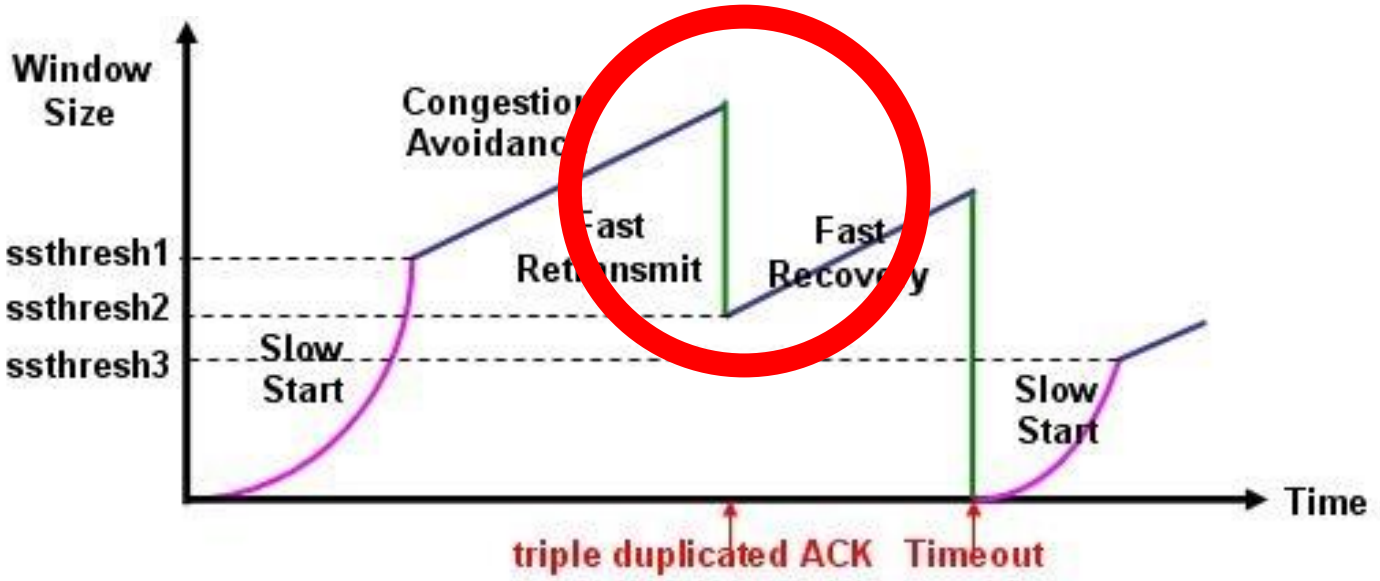
Application Example: TCP Over Wireless



- 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

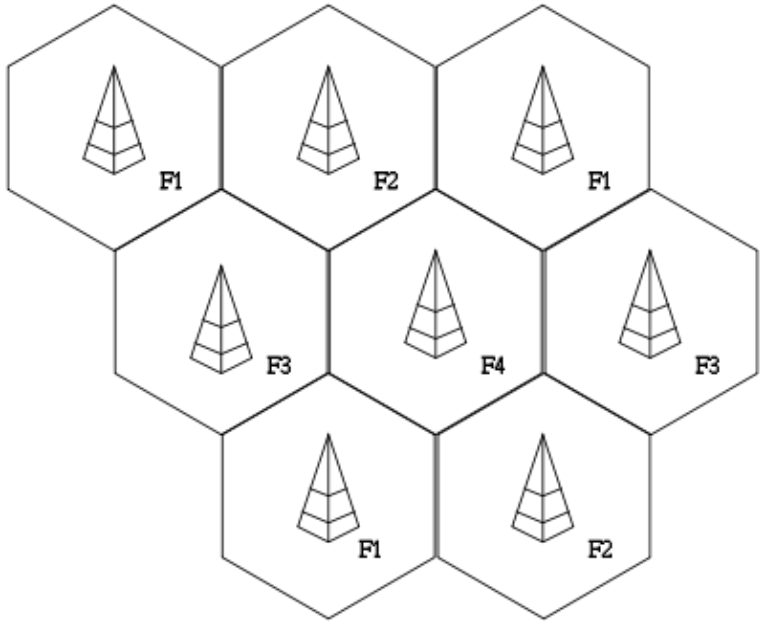
Application Example: TCP Over Wireless

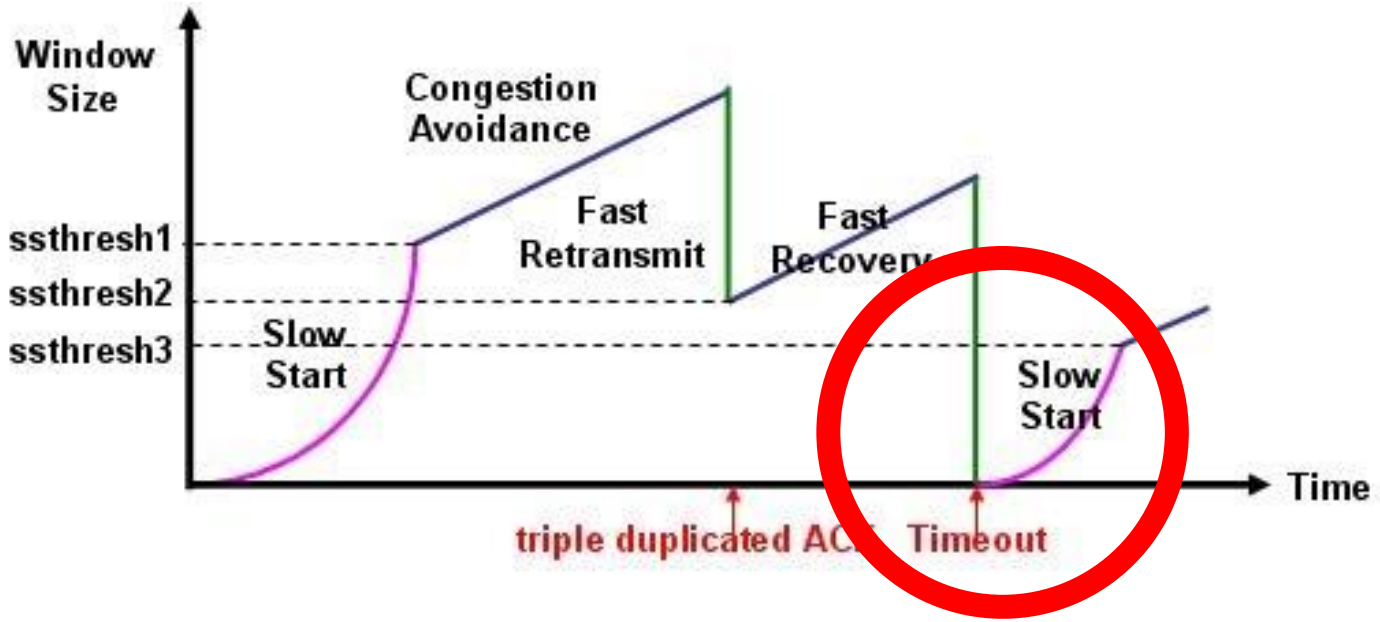




- When a client moves between cells handoff will cause Loss
- TCP will interpret this as Congestion and cut throughput even if the exact same amount of bandwidth is still available (even if more bandwidth is now available)!

Application Example: TCP Over Wireless





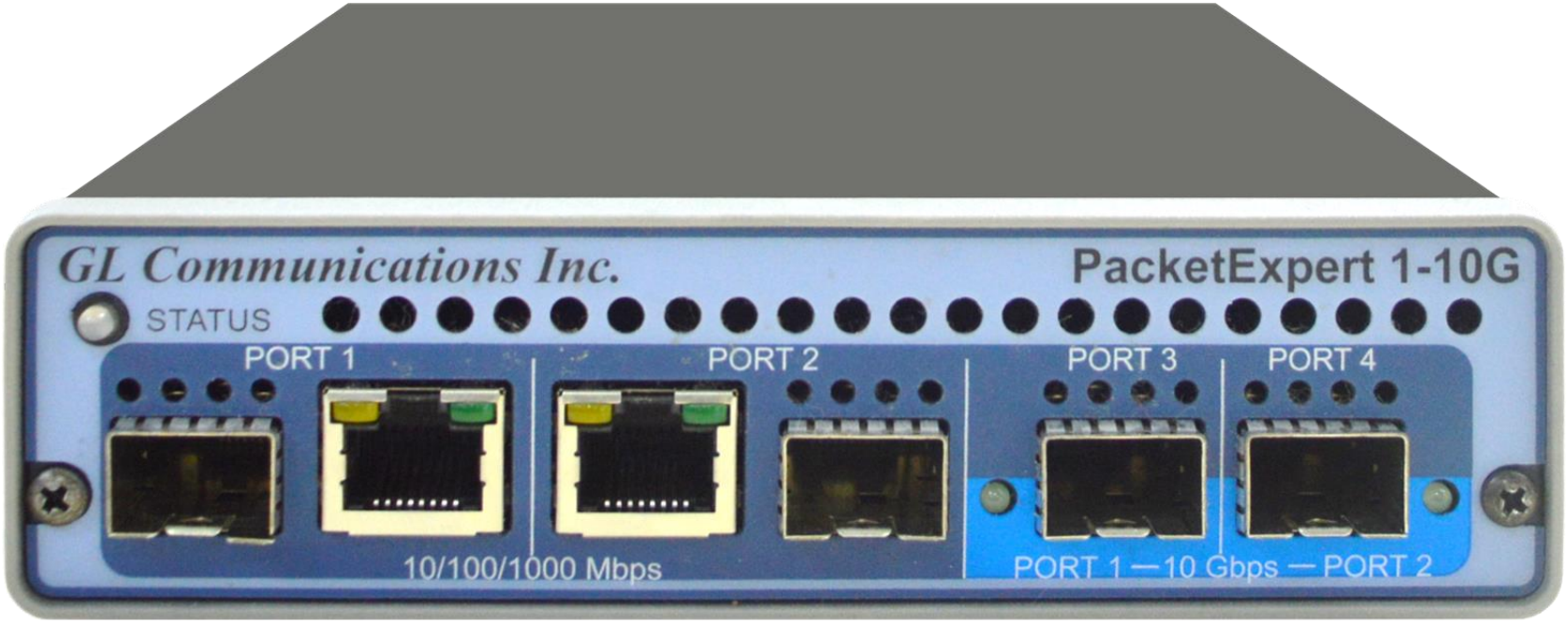
- TCP has Retransmission Timeout mechanism that attempts to track the RTT of the connection
- If Latency suddenly increases (ie cell handoff), this can easily cause Timeouts to trigger, immediately cutting throughput to the minimum!

Application Example: TCP Over Wireless

- Wireless communication is High Latency
- Wireless communication is prone to Packet Loss and Packet Corruption
- Applications that work perfectly in the lab can be crippled by a wireless link

Automation Options

Automation Options



- Automate with C/C#, TCL, Python and Java

THANK YOU - Any Questions?