NE-ONE FAMILY
IMPAIRMENTS AND TRANSITIONS GUIDE

NE ONE By Calnex

WHAT'S INSIDE

1. Overview
2. Linkspeed, Bandwidth & Queuing
3. Latency & Jitter
4. Loss
5. Cloud Network
6. Bit Error
7. Duplicate
8. Out Of Order
9. Fragment
10. Pause
11. General Link Impairments
12. Scenario Builder Network Transitions
13. Glossary

OVERVIEW

The NE-ONE family provides the most complete and realistic network impairment library allowing you to easily mimic what happens in real-world networks. With more than 100 parameters to choose from NE-ONE combines realism and accuracy in a Software Defined Test Network that allows you to test applications over a range of controllable and repeatable scenarios. Furthermore, each impairment function has multiple parameters allowing you to customize its behavior for your specific testing needs. NE-ONE's impairments are frequently updated based on evolving networks and customer needs.

LINKSPEED, BANDWIDTH & QUEUING

Linkspeed and FIFO Queue Bytes
Description: Controls the Linkspeed, Congestion and Queuing on a link. It is primarily used for link bandwidth and throughput control.

Parameters: Linkspeed, Queue Length, Overhead (Layer 2), Congestion Percentage, TTL Cost

Summary: This is the fundamental Linkspeed/Bandwidth function. Setting the Linkspeed also implicitly creates a delay called the Latency of Serialization to the link (this is only very apparent when bit rates are low and/or packet sizes large) as would happen in real world links. You can set the queue size to allow for data surges, simulate the overhead of different layer 2 header sizes and use simulated congestion.

Linkspeed with Variable Congestion
Description: Controls the Linkspeed and Queuing on a link and features a variable congestion model to simulate links where bandwidth availability varies greatly over time.

Parameters: Linkspeed, Queue Length, Overhead (Layer 2), Congestion Percentage, TTL Cost, Table of Congestions (Min and Max) and Durations (Min and Max)

Summary: This function is intended to make is very easy to represent links which, while having a constant linkspeed, have a highly varying congestion, and therefore variable bandwidth & throughput. No scripting is required to achieve this.

QoS Class Bandwidth
Description: This function allows an overall Linkspeed to be defined and then split into QoS Classes. A built in tabular classifier allows selection of what traffic constitutes each class. Traffic shaping (Peak or Average) is also featured.

Parameters: Linkspeed, Circuit Queue Size, Circuit Overhead, Shaping Type (Peak, Average), CIR, Bc, Be, Queue Size, Table of Qos Classes (each featuring Shaping Type, CIR, Bc, Be, Queue Size and ability to use spare Bandwidth), Table of Traffic Classifiers (Source and Dest IP, Soure and Dest TCP/UDP Port, IP Protocol, Vlan Id)

Summary: This is very flexible function is designed to allow QoS control of Linkspeed by class using the parameters common in many modern routers e.g. Cisco, Juniper etc. You can define what traffic qualifies for each class and each class features traffic shaping in addition to the shaping of the entire link.

LINKSPEED, BANDWIDTH & QUEUING (CONT.)

QoS Class Bandwidth using Expressions
Description: This function is similar to its counterpart QoS Class Bandwidth in allowing an overall Linkspeed to be defined and then split into QoS Classes. However, it features an expression (similar to Wireshark selection syntax) classifier to define the QoS Classes. Traffic shaping (Peak or Average) is also featured.

Parameters: Linkspeed, Circuit Queue Size, Circuit Overhead, Shaping Type (Peak, Average), CIR, Bc, Be, Queue Size, Table of QoS Classes (each featuring Shaping Type, CIR, Bc, Be, Queue Size and ability to use spare Bandwidth), Table of Traffic Classifiers using "Wireshark like" expressions.

Summary: This is a very flexible function designed to allow QoS control of Linkspeed by class using the parameters common in many modern routers e.g. Cisco, Juniper etc. You can define what traffic qualifies for each class using "Wireshark like expressions" and even add your own protocol definitions to allow custom definition of QoS classes. Each class features traffic shaping in addition to the shaping of the entire link.

LATENCY & JITTER

Gaussian (Normal Distribution) Delay
Description: Applies a latency to a packet based on the Mean Delay and the Standard Deviation, determined by the Gaussian (Normal) distribution. Latency will never go below Min Delay.

Parameters: Min Delay; Max Delay; Mean Delay; Standard Deviation.

Summary: This is probably the most real-world delay impairment but the minimum and maximum delay parameters ensure a base network latency (which is realistic) and rejects low probability but ridiculously high latencies. Creates Gaussian Jitter.

Step Delay Periodic
Description: Applies a step latency beginning at Min Delay and going up to Max Delay in milliseconds, changing steps every specified amount of ms has elapsed. When Max is reached the steps come down to Min Delay and the process is repeated.

Parameters: Min Delay; Max Delay; Step Delay; Step Duration.

Summary: Allows efficient stress testing of applications from a latency point of view to establish limits.

Step Delay Packet
Description: Applies a step latency beginning at Min Delay and going up to Max Delay in nanoseconds changing steps every packet. When Max is reached the steps come down to Min Delay and the process repeated.

Parameters: Min Delay; Max Delay; Step Delay.

Summary: Exposes any vulnerabilities to high latency and allows efficient stress testing to establish limits.

Fixed Delay
Description: Applies a fixed latency in nanoseconds.

Parameters: Delay.

Summary: Allows you to be exact about your delay (+- 10 microseconds in practice) which is useful when precision is required.

LATENCY & JITTER (CONT.)

Fixed Delay with Jitter
Description: Applies a fixed base delay (latency) and random jitter (PDV).

Parameters: Base Latency; Max Jitter.

Summary: Allows you to separately control base delay and maximum jitter.

Random Delay
Description: Applies a random latency between Min Delay and Max Delay in milliseconds.

Parameters: Min Delay; Max Delay.

Summary: Allows you to easily and quickly set random latency causing Jitter.

Delay Sequence
Description: Vary Latency and Jitter over time with no requirement to script using a tabular definition.

Parameters: Table of: Base Delay, Jitter, Duration entries and Repeat (checkbox)

Summary: This function allows you to quickly set up a varying set of Base Delays and Jitter values setting their duration before moving onto the next set of values in the table when the table is complete you can optionally start again from the beginning. Its purpose is to make sequencing latencies and jitters simple with no programming required.

Delay Scenarios
Description: Provides a very simple and rapid way of selecting some delay scenarios based on simple concepts e.g. LAN in a Campus, WAN to Nearby Cities etc. This is not to be confused with NE-ONE's city-to-city database or Network Type databases.

Parameters: Network Type (LAN in Building, LAN in Campus, WAN Nearby Cities etc).

Summary: Designed to make it easy to choose a representative latency with basic knowledge. This has been augmented by the on board city-to-city database which allows nodes to have their cities defined and then computes appropriate circuit latency. Further augmentation comes from the extensive onboard network type database.

Inter Packet Gap Delay
Description: Applies a random inter-packet gap between Min Gap and Max Gap milliseconds.

Parameters: Min Gap; Max Gap; Delay.

Summary: Aimed at creating Jitter in high speed packet streams by specifying a definite inter-packet gap. Ideal for simulating flows being held up and then bunching which can be very useful for certain streaming tests.

LOSS

Random Drop with Burst
Description: Drop a sequence of packets on a Random (percentage) basis. The number of packets to drop is randomly chosen between Min and Max.

Parameters: Loss Percent; Minimum Packets to Drop; Maximum Packets to Drop.

Summary: A simple way (without programming) to create networks that suffer periodic noise or high losses to determine the impact on application performance.

Poisson Drop
Description: Drop on average Lambda packets in each Interval specified, using the Poisson Distribution (with mean Lambda).

Parameters: Lambda; Interval; Duration.

Summary: Drops packets according to the Poisson distribution to mimic patterns often seen in the real world.

Packet Error 1 in X Bits
Description: Drops packet if packet errored. This is based on bit errors. Specify loss as 1 in every x bits.

Parameters: Loss x.

Summary: This is common for networks that create bit errors. The drop simulates other equipment dropping the broken packet due to a checksum mismatch and is very realistic for mimicking satellite and certain wireless transmissions.

Burst Loss
Description: This function allows you to specify two loss levels— regular loss and a burst (usually higher) loss and choose how frequently and long to burst.

Parameters: Loss Percent; Burst Loss Percent; Burst Frequency; Burst Duration.

Summary: Developed for customers that need a random burst rather than period burst for testing dropped packets randomly followed by dropping several in a row - a random burst. Provides a very realistic test network for random bursts of noise, for example in wireless networks.

Drop 1 in X
Description: Drops 1 in every x packets

Parameters: Loss x

Summary: Provides a steady drop of 1 in x packets which is good for early testing of protocols and whether they can recover from certain amounts of loss.

Random Drop
Description: Applies the percentage packet drop

Parameters: Loss Percent

Summary: The most commonly used drop function to simulate real world drops due to failure to queue etc. Provides a realistic symptoms of what happens when network links are busy and queues are full.

CLOUD NETWORK (REPRESENT A CORE NETWORK IN A SINGLE OBJECT)

Cloud functions control multiple impairments at the same time - they are effectively complete networks in one icon.

Description: This allows a single Node (internal router) to act as though it was a complete network by having multiple (potentially thousands) of internal network paths each with different characteristics. It allows complex networks such as Cellular, MPLS, Internet and Wireless to be easily represented.

Parameters: Table of Bandwidth, Latency, Jitter, Queue Size, Loss and TTL Cost together Traffic Classifiers (Source and Dest IP, Source and Dest TCP/UDP Port, IP Protocol, Vlan Id).

Summary: This powerful feature allows a complex fully or partially meshed network to be represented as one item, as it often is in network diagram but with different characteristics being applied to different traffic based on your own classification rules.

BIT ERROR

Error with Burst
Description: Applies bit errors— at the rate (and deviation interval) given. It has the ability to burst to a higher error if burst parameters defined.

Parameters: Error Rate; Error Rate Deviation; Burst Error Rate; Interval Between Bursts; Interval Deviation; Burst Duration.

Summary: Mimics a steady bit error rate and then a random burst on top providing realistic packet corruption when noise hits a network. Particularly good for wireless and satellite circuit testing.

Poisson Error
Description: Error on average Lambda bits in each Interval specified, using the Poisson Distribution (with mean Lambda)

Parameters: Lambda; Interval; Duration.

Summary: In the real world, when not bursty, bit errors occur according to the Poisson distribution as opposed to normal or uniform distribution.

Random Packet Error
Description: Select a packet for error on a random % basis (uniform distribution) - then randomly choose which bit to error.

Parameters: Packet Error Percent

Summary: Ensures a certain proportion of packets are corrupted and is as likely to corrupt a small packet as a big one. Good for stress testing in packet error situations to determine if the protocol or application can cope.

Random Packet Corrupt
Description: The purpose of this function is to allow specific parts of a packet to be modified (corrupted) in a specific manner (REPLACEing, ANDing, ORing, Flipping (XORing) bits & bytes etc). This is valuable when probing for weakness in protocols or applications.

Parameters: Packet Corruption Percent, Table of Corruptions (Byte Offset, Corruption Type [XOR, OR AND, ADD, SUB, OVERWRITE], Corruption Value)

Summary: This function corrupts packets on a randomly chosen basis specified by a Percentage value. There is a table of corruptions to apply which include adding, subtracting, anding, or-ing, xor-ing or overwriting any specified bytes in the packet.

DUPLICATE

Packet Duplicate & Move
Description: Select a packet in the current stream on a random % basis (uniform distribution) then duplicate it and optionally move it between a minimum and maximum packet positions. A move of 0 packets means the duplicate occurs immediately after the original.

Parameters: Selection Percent; Minimum Move; Maximum Move.

Summary: Allows more realistic testing than simple duplication because the duplicated packet can be held and inserted randomly in the traffic flow and not behind the original. Good for certain cyber/crypto testing.

OUT OF ORDER

Random Packet Time Reorder
Description: Takes a packet out of the current stream on a random % basis (uniform distribution) - and holds it between a Minimum and Maximum time in milliseconds.

Parameters: Move Percent; Minimum Time; Maximum Time.

Summary: Used to test protocol resilience and exposes application inadequacies e.g. Jitter buffer ability to efficiently put packets back into order. It is likely that you would have to wait a long time to see this in the real world.

Random Packet Move Offset
Description: Takes a packet out of the current stream on a random % basis (uniform distribution) - and moves it between a Minimum and Maximum packet position.

Parameters: Move Percent; Minimum Move; Maximum Move.

Summary: Used to test protocol resilience and exposes application inadequacies e.g. Jitter buffer ability to efficiently put packets back into order. It is likely that you would have to wait a long time to see this in the real world.

Packet Reorder 1 in X
Description: Takes a packet out of the stream on a 1 in X basis and moves it backwards by a random value chosen between a minimum and maximum amount.

Parameters: Reorder x (the x in 1 in x), Minimum Move, Maximum Move.

Summary: This allows packets to be removed from the packet stream and held until a certain number of packets has passed, which is randomly chosen between the Minimum and Maximum values supplied. This function, like other packet reorder functions, enables testing that applications and protocols are resilient to packets being out of order either for security purposes or performance purposes. Packets can naturally go out of order when more than one route is available and streams are not forced down one route.

FRAGMENT

Fragment MTU
Description: Fragments a packet into multiple smaller size packets, transmits the smaller packets and drops the original packet

Parameters: MTU Limit; Don't Fragment Flag Option

Summary: Emulates what happens in the real-world and allows applications' and protocols' ability to efficiently reassemble the fragmented packet to be tested.

PAUSE TRANSMISSION

Description: This function allows a link to pause transmission such as might occur if the next node was not available - packets will be queued until the link is un-paused. This is valuable in a number of situations including Store and Forward Routing and Beam Change in Satellite Constellations.

Parameters: Pause Time (can be infinite i.e. until unpaused)

Summary: This function was created to allow the creation of links where the next node was not necessarily available at that time. The current node would have to store the data until the next node becomes available. Particularly useful in Wireless and Satcoms (especially LEO constellations).

GENERAL LINK IMPAIRMENTS

Link Speed
Specify the network bandwidth from 1bps so that you can exactly match a given real world circuit. (Upper limit is model dependent)

Queue Length
Define the link buffer size so that packet drops happen when buffer is full just as happens to routers, traffic shapers or WAN optimizers in the real network.

Overhead
Define the Layer 2 header size so that you can model other network types, not just Ethernet. Provides more realistic testing for non-ethernet network emulation.

Congestion
Set Link congestion as a % of the link capacity. Allows you to simulate actual usage of the link because in the real world networks carry other traffic. Enables you to get a realistic test without needing to use traffic generators to make the link look busy.

TTL Cost
Represent the number of network hops the packets will pass through so that the TTL is the same as in the real network. External routers make routing decisions based on TTL therefore setting the TTL increases testing realism.

SCENARIO BUILDER NETWORK TRANSITIONS

Transitions define what happens when changing between test networks (elements) in the scenario, for example changing from a 2G to 3G network.

None
Provides an easy method of a hard change from the previous Element to the new Element's parameters. This can be used to stress an application by providing a hard change in parameters.

Graduating
This transition changes all emulation parameters Gradually between the values of the Element before and the Element after it. Allows a gradual ramping up or down of any parameter without using scripting or programming.

Variable
This transition changes all emulation parameters Variably (in a random manner) between the values of the Element before and the Element after it. Allows a very easy method of creating a fluctuating or flaky network, for example changing from 2G to 3G network.

Outage
This transition creates an increasingly lossy network, leading to total loss before returning quickly but not instantly. Such a transition is highly representative of mobile networks when losing signal from one source and coming back with another e.g. wifi - leave building, pick up 3G.

GLOSSARY

Milliseconds: Thousandth (10–3) of a second.
Microseconds: Millionth (10-6) of a second.
Nanoseconds: Billionth (10–9) of a second.

Lambda: The Poisson distribution is defined by one parameter:lambda (λ). This parameter equals the mean and variance. As lambda increases, the Poisson distribution approaches a normal distribution.

PDV: Packet Delay Variation is the difference in end-to-end one-way delay between selected packets in a flow with any lost packets being ignored. The effect is sometimes referred to as jitter.

ne-one.com • T: (USA): +1 888 448 4366 • T: (UK): +44 (0)1799 252 200 • E: ne-one@calnexsol.com