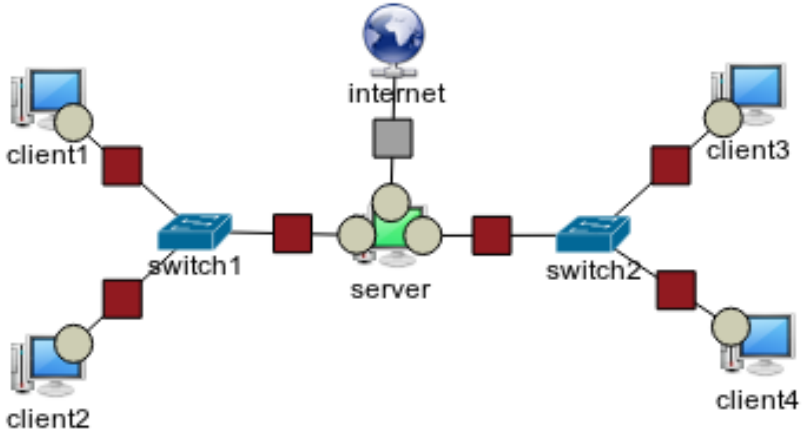
# ToMaTo - Link Emulation

# ToMaTo, Topology Management Tool

ToMaTo is topology-oriented, i.e. users build topologies for their experiments.




## Devices

- produce and consume data
- can run software

## Connectors

- forward and manipulate data
- connect devices

## Three kinds of devices

- KVM devices
- OpenVZ devices
- Programmable devices

## Two kinds of connectors

- VPN networks
- External networks

# Link emulation

## Definition

Link emulation is a term that defines methods to apply artificial characteristics or network links and path to a communication link.

## Emulatable characteristics

ToMaTo is able to emulate the following link characteristics for all VPN links in both directions (seperately):

- Delay
- Jitter
- Statistical delay distribution
- Bandwidth limitation
- Packet loss
- Packet duplication
- Packet corruption

# Delay and jitter

## Delay

Delay can be added to a link by buffering each packet for a certain time before processing it. Delay can be configured in milliseconds but also fractions can be specified. The resolution depends on the timer resolution of the kernel but is very accurate in general.

## Jitter

Jitter is a deviation applied to the per packet delay. Like the delay, jitter can also be specified in milliseconds but also fractions will work. The meaning of the jitter value depends on the selected statistical delay distribution but the jitter is applied in an average-neutral way.

## Packet reordering

Packet reordering can be a result of high jitter values. If a packet gets a high delay value and the next packet a low value due to jitter, the later packet could leave the buffer first.

# Statistical delay distribution

## Uniform distribution

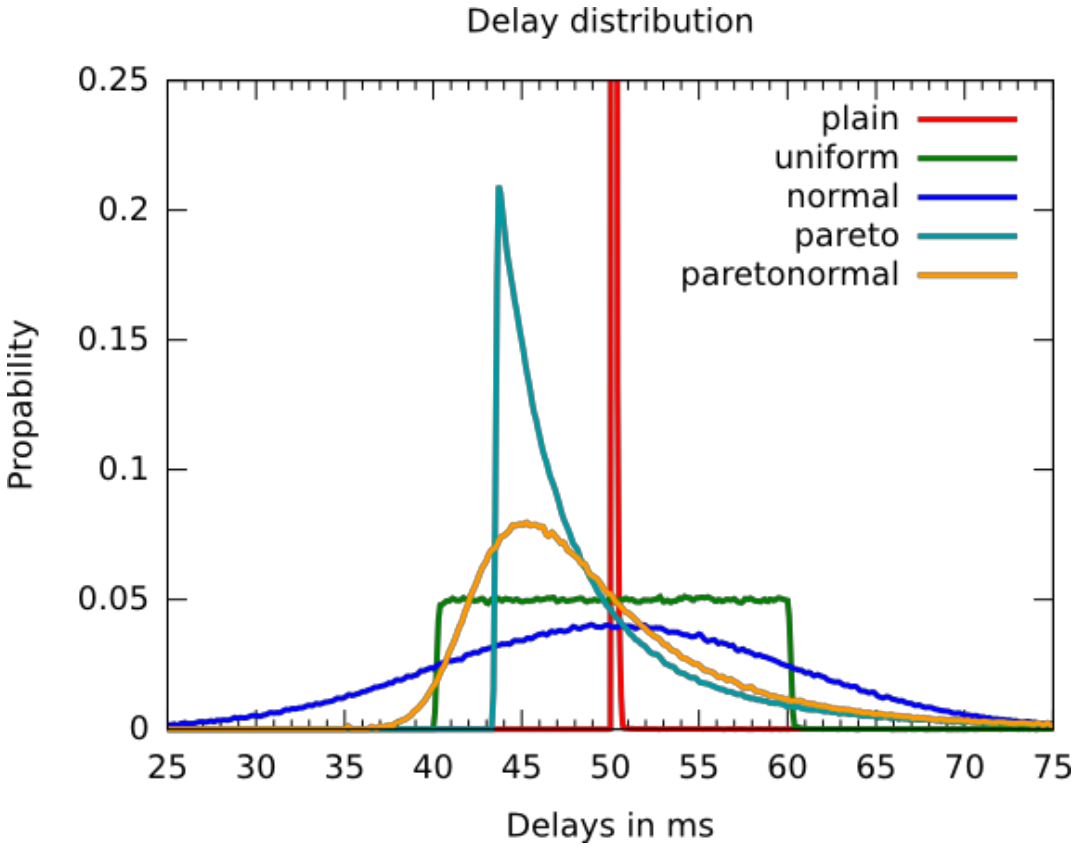
- Default distribution
- Values evenly distributed within `±jitter`

## Normal distribution

- Higher chance for values near delay
- 68% of values within `±jitter`
- 95% of values within `±2*jitter`
- 99% of values within `±3*jitter`

## Pareto distribution

- Power law
- High chance for small delays, small chance for high delays
- Average is configured delay, median is below




## Paretonormal distribution

- Combination of normal and pareto distribution
- Good emulation of most existing links

# Other characteristics

## Bandwidth limitation

- limit the maximal data rate
- configured in kilobits per second
- default bandwidth limit is 10 MBits/sec

## Packet duplication

- chance for each packet to be duplicated
- only one duplicate
- duplicate counts a traffic
- delay and jitter is applied seperately

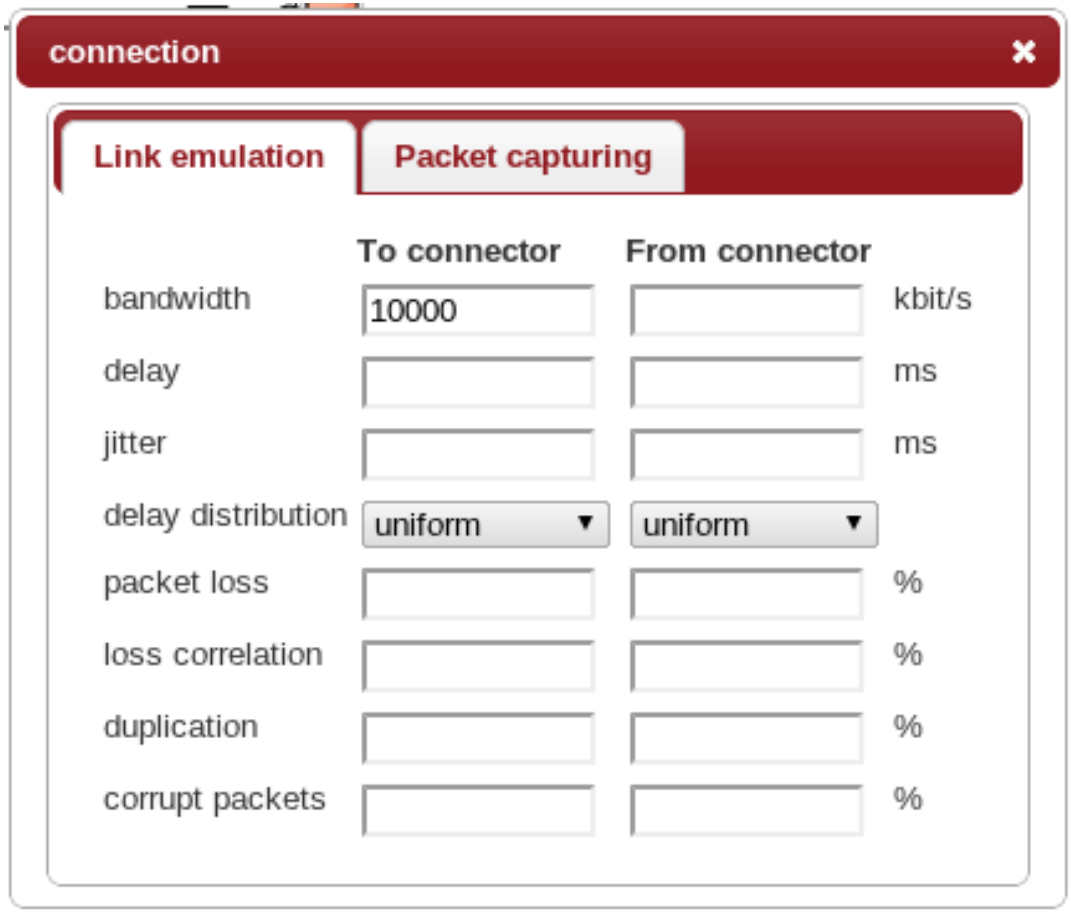
## Packet loss

- chance for each packet to be lost (dropped)
- configured in percent (5 means 5%)
- optionally correlated

## Packet corruption

- chance for each packet to be corrupted
- one random bit is flipped

# Characteristics configuration




The link emulation properties can be changed on the link properties page for each link. The link characteristics can even be changed for running connections.

# Characteristics are additive

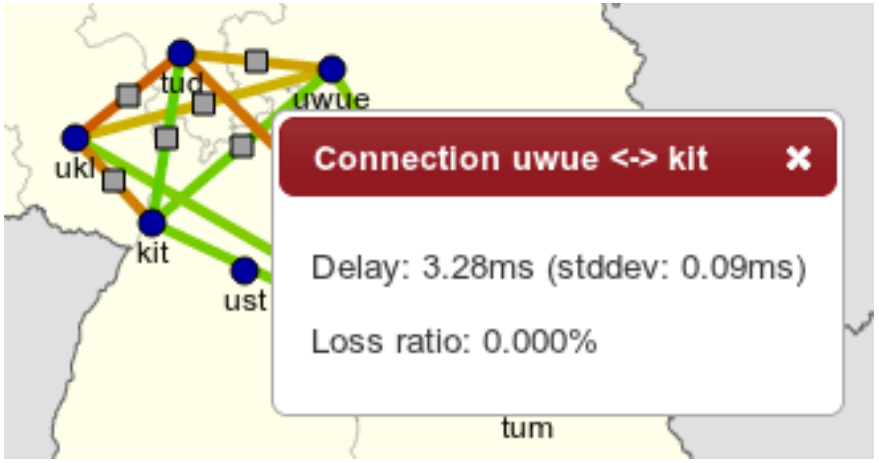
## Characteristics are additive

The configured link emulation settings are applied additionally to the existing characteristics of the underlying link.

- Delay is just added to the underlaying delay
- Jitter values and distributions are combined to more complex distributions
- Resulting bandwidth limit is the minimum of the existing and the configured limit
- Chances for loss, duplication and corruption are inversely multiplied, i.e. $p'=1-(1-p_1)*(1-p_2)$

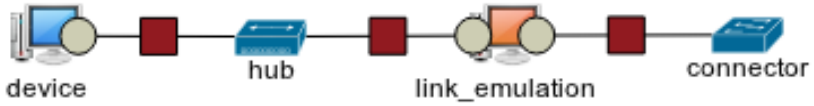
## Continous measurements

To help users estimate the resulting link characteristics the links between all sites are continously measured and long-time statistics are provided.



# Complex link emulation

Sometimes a more complicated link emulation or traffic shaping is needed on links. This can be done by using a link with a device in the middle.




## OpenVZ

- filtering using the Linux kernel firewall

## KVM

- can run any custom filtering and link emulation software

## Programmable devices

- easy link emulation as a program
- easy filtering as a program

# Obtaining and contributing

## How to get ToMaTo

ToMaTo is Open-Source! It can be simply downloaded from the Github page. There is also a step-by-step tutorial on how to setup ToMaTo in a testbed. ToMaTo includes some nice features that make it pretty easy to install it in an experimental facility:

- All components packaged for debian (updates come automatically)
- Multiple authentication plugins: LDAP, htaccess, SQL-Database, Planet-Lab, ...
- Automatic checks and problem reports for the ToMaTo hosts

## How to contribute to ToMaTo

As an Open-Source project ToMaTo is open for hints and contributions.

- Github offers an easy way to fork the project and offer contributions as pull requests
- The wiki is publically editable so everyone can help by adding to the documentation
- The issue tracking system can be used for bug reports and feature requests