



JAR:emulate

Network Emulation



WE LIVE IN A DIGITAL WORLD
AND RELY ON APPLICATIONS TO
DRIVE BUSINESS SUCCESS.

Risk of business interruption and pressure to increase speed to market means network and application failure is not an option, they need to work first time, every time, all the time.

Testing before deployment is critical. Simulate networks and emulate real world conditions with JAR:Emulate to get [predictable infrastructure performance](#).

Whether you want to emulate point to point links with latency and bandwidth throttles, or simulate complex data centres with multiple gateways and interconnecting links, with JAR:Emulate you can [build networks in seconds](#).

Using a visual design process to build emulations, take control by linking together impairments in the order you require whilst directly controlling the flow of packets around your simulated network.

Application Performance: Determine how your software will perform for the end user on their network before deployment; trouble shoot and resolve issues before the need for re-work, release delay or failure

Optimization: Adjust application and software settings under replicated network characteristics to optimize performance for different user groups.

Trouble Shoot: Investigate reported problems and test resolutions without disrupting production traffic

Network Design: 'What if' scenario building to evaluate network changes - evaluate different topologies and technologies before expensive implementation. For example evaluate WAN Acceleration Technology: Determine how different WAN optimisation products will perform under the best/average/worst conditions of your network.

Accurately Simulate

VoIP	Satellite	Cloud	ADSL
Microwave	OC-3	3G	4G
T1	E1	T3	E3
2G	WiMAX	DSL	XDSL
WiFi	RoIP	GPRS	Dial Up

Under Real Network Conditions

Busy Peak Times	Limited Bandwidth
Long Distance	Information Corruption
Jittery Connections	De-sequencing of Data
Duplication of Data	Network Traffic Bursts
Fragmentation of Data	Partial or Total Outage
Traffic Re-routing	Back-up Switch-overs

FEATURES

Build Networks in seconds

JAR:Emulate enables you to quickly build networks in seconds. Whether you are interested in emulating point to point links with latency and bandwidth throttles, or simulating complex data-centres with multiple gateways and interconnecting links. Build these emulations through a visual design process.

Filtering

It's very important to be able to identify and target certain streams of data on an emulated network.

JAR:Emulate provides a host of built-in filters to identify packets for impairment or analysis.

Furthermore you can join filters together to create more complex filters (for example, you could find HTTP traffic going to Port 8080 on VLAN ID 90 with only 2 filters).

Switch over and back-up circuits

Built on top of our fantastic filtering system, you can easily simulate multiple paths in any network and switch between them in real-time (or using an external trigger). This allows you to model a "good" and "poor" network and switch between them in real-time, furthermore you can create an unlimited number of paths.

Virtual Routers

Routers effectively sit at the edge of most networks; they provide your ADSL connection in your business or the accessibility to a cloud server through an ISP backbone.

JAR:Emulate contains full Virtual Router simulation, providing a method for simulating multi-hop WAN networks. This Virtual Routers provide DHCP or Static IP address assignment, automatic and manual routing table, multiple virtual interfaces - they even respond to both local and WAN pings and detailed analysis on packets received.

They also include OSPF to allow for the simulation of complex ring, double ring and double star topology networks with the ability to self-heal when injected faults are detected.

Impairments

Shipping with all the standard impairments, you'll be amazed that there is an additional 45+ tools available to allow complete design and control over your emulations. Whether it's filtering, traffic generation, virtual routers or statistical analysis you will have complete control over their operation.

Load Generation

Very few networks are silent; there is always a certain amount of background traffic. Whether it's file servers, printers or other devices generating traffic it's important to model this correctly.

JAR:Emulate provides a number of ways of generating this traffic including background traffic generation (contention), TCP load generators, PCAP file replay, etc.

Comprehensive settings

Not only do we include 45+ impairments with our emulators, but each one of those impairments has extensive options and settings to uniquely control their operation. This gives you a unparalleled level of fine tuning; such as buffer and burst management on bandwidth throttles or affecting only B-Frames in H.264 video streams.

Need more than 2 ports?

JAR:Emulate ship with true "Any Port to Any Port"[™] capabilities. Whether your using 4, 8 or 24 ports they will all communicate with each other.

Others emulate products may state you are buying a '4 port emulator' but often what you actually get is 2 'emulators' with 2 ports inside a single box. There may be 4 ports on the outside, but what you have got is 2 independent emulators and you won't be able to send packets between each. So effectively you have got 2 emulators with 2 ports, not the 4 ports you need.

Wire Rate

All core models provide full wire-rate but we understand that you might need 10Gb/s ports but only want to put 1Gb/s through the emulator. JAR:Emulate can be tailored to your throughput needs, helping you to secure that high return on investment.

Enterprise Options (as standard!)

Our emulators often form the backbone of complex "always on" testing environments. Therefore you can expect our kit to include all those enterprise necessary features - including SNMP Traps, NTP and PTPv2 time synchronisation, execute emulation on power-up, trigger packets for emulation events, etc.

Flexible UI with Wizards and Easy Licensing

Our user interfaces run on Windows, Linux or Mac and provides a fast and responsive environment for you to build emulations. We've included some fantastic wizards to quickly create example networks and don't lock you down to restrictive licensing (install it on as many machines as you want!).

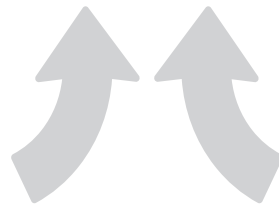
Automation Ready

We understand that everyone's automation environments are different. JAR:Emulate can be controlled externally through a network of methods. You are provided with a CLI application (Windows, Linux or Mac) for controlling execution from the command line. Whilst developers can directly issue XML requests to the emulator to gain deeper and full control of its operation.



TEST APPLICATIONS

or new technologies before deployment,
under the exact conditions of the end user network:



Injection of:

Packet loss
Delay
Jitter
Corruption
Congestion
Other Network Impairments

Analysis of:

Application Performance
Quality of Service
Proposed Networked Technologies
WAN Optimisation Technologies
Data Centre Relocation / Consolidations
'What If' Network Planning

Emulating (for example)

