Cisco 8000 Hardware Emulator Datasheet

Software Release

Last updated: 05/18/2021

Contents

Product Overview

The Cisco 8000 Hardware Emulator portfolio (here after referred to as 8000e) provides one for one equivalent simulation of the 8xxx Series routers. Unlike other Virtualized Network Operating System solutions, 8000e provides both accurate hardware chassis and forwarding engine emulation. This enables 8000e to run the same production IOS-XR images as hardware. Secondly, it can run third party Operating Systems such as SONIC which have been ported to the 8000 series routers.

Benefits

Key benefit for customers:

- Early access to virtual hardware for testing and integration
- Integration with network emulation environments
- Integration into CI/CD development pipelines
- Evaluation of next generation IOS-XR7

Emulated Platforms

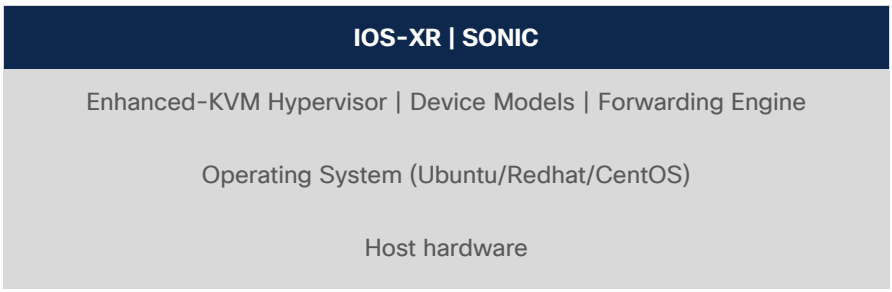
Currently Supported Platforms:

Hardware Emulator	Port Configuration	OS support
8201	20x400GE + 12x100GE	IOS-XR 7.x SONIC
8201-32FH	32x400GE	IOS-XR 7.x
8202	12x400GE + 60x100GE	IOS-XR 7.x
8101-32H	32x100GE	IOS-XR 7.x
8102-64H	64x100GE	IOS-XR 7.x

Note 1: Additional emulated platforms will be released coinciding with the introduction of new 8000 routing hardware.

8000 Emulator Architecture

The 8000e product line leverages type 2 hardware accelerated hypervisor technology and benefits from running on the host operating system. Being a hardware emulator, the 8000e is a distinct software product and is independent of the operating system it runs.



While single board/CPU platforms can be represented with an instance of the hypervisor, modular chassis such as the 8808 require multiple instances. For this reason, the computational requirement for a populated modular chassis grows with the number route processors, Line cards, and Fabric cards simulated in the system.

Deploying the Emulator

The software package comes with complete toolset required to create and run topologies of emulated routers. Using our solution, user can launch flavors of the 8000 emulator, other virtual routers, traffic generators, and interconnect them. The topology is specified in a YAML notation. The toolset includes a python library to manage the simulation lifecycle.

Users can also deploy an instance of the emulated router within their own framework. To deploy as a component, the 8000 instance is wrapped in a docker container. It can also be wrapped in a single VM.

Deployment options are:

Deployment	Base System	OS	What is provided	Note
Server installation Linux	16+ cores 64G+ Mem	Ubuntu18 CentOS/RH8	Linux packages + install scripts	
Docker env	16+ cores 64G+ Mem	Ubuntu18 CentOS/RH8	Docker file to create environment	
AWS	Bare metal instance	Ubuntu18	Automation scripts to create AMI images	
Azure	16+ cores 64G+ Mem	Ubuntu18	Linux packages + install scripts	Requires nested VM
ESXI	16+ cores	Ubuntu18	Linux packages + install	Requires nested VM

	64G+ Mem		scripts	
Windows* Hyper-V	8+ cores 24G+ Mem	Windows Ubuntu VM	Linux packages + install scripts	Requires nested VM
Windows* VMware	8+ cores 24G+ Mem	Windows10 Ubuntu VM	Linux packages + install scripts	Requires nested VM
Apple* Fusion	8+ cores 24G+ Mem	MacOS Ubuntu VM	Linux packages + install scripts	

Note 1: Resource requirements for the last three rows cover ability to run single instances in nested virtualization mode.

Note 2: Integration with CML2 and GNS3 are forthcoming.

Runtime Requirements

Emulator	Operating System	CPU	Memory	Min Memory	Disk	Comment
8201	IOS-XR	4	32G	12G	30G	
8802	IOS-XR	4	32G	12G	30G	
8808		8*	64G*	12G*	30G	Per board in system

Limitations

While 8000 emulator attempts to match actual hardware router, there are limitations:

Feature	Emulator limitation
Traffic shaping and rate limiting	Functionality is limited and does not match hardware.
Data Throughput	Emulator throughput is in the thousands of packets a second and cannot be used to handle production traffic.
Counters	Some hardware counters are not supported in the emulator.