

Sponsored by:



## NETWORKWORLD

This story appeared on Network World at

<http://www.networkworld.com/reviews/2006/100906-quagga-router-test-how.html>

# How we tested Quagga router

[Clear Choice Tests](#) By John Bass, Network World, 10/09/06

**CLEAR CHOICE TEST** We tested Quagga 0.99.4 on Gentoo [Linux](#) (kernel v2.6.15) running on both a [Dell](#) 2850 and a Dell 1550. The Dell 2850 has two [Intel](#) Xeon 3.6GHz processors with hyperthreading and 1024MB of RAM and two 10/100/1000Mbps Ethernet ports.

The Dell 1550 has a single PIII 933MHz processor with 256MB of RAM and two 10/100Mbps Ethernet ports. We also ran the tests on a [Cisco](#) 2621 with two 10/100Mbps Ethernet ports and a [Nortel](#) 5520 with 48 1000Mbps Ethernet ports for comparison.

Register or sign in now to view the entire article and access to Network World's Buyer's Guides.

[">Register/Sign in](#)

We used a Spirent Communications Smartbits 6000 chassis load with 3301A Terametrics 10/100/1000Base-T cards for all test runs. We used Smartflow for most tests and TeraRouting Test v5.0 for all the routing convergence tests.

We measured Layer 3 throughput by finding the maximum throughput without loss in a bidirectional two port-pair with all ports in full duplex mode at the fastest supported link speed. The test is repeated for seven packet sizes (64 bytes, 128 bytes, 256 bytes, 512 bytes, 1,024 bytes, 1,280 bytes and 1,518 bytes).

We measured Layer 3 packet loss by sending bidirectional traffic streams through a two-port pair with all ports in full-duplex mode at the fastest supported link speed. The traffic stream rates are increased from 10% load to 100% load in 10% increments. The test is repeated for seven packet sizes (64 bytes, 128 bytes, 256 bytes, 512 bytes, 1,024 bytes, 1,280 bytes and 1,518 bytes).

We measured Layer 3 latency at the maximum throughput with no loss across a bidirectional two port-pair in full-duplex mode at the fastest supported link speed. The test is repeated for seven packet sizes (64 bytes, 128 bytes, 256 bytes, 512 bytes, 1,024 bytes, 1,280 bytes and 1,518 bytes). The latency values referenced in the article are for 10% load with 1,280-byte packets.

The maximum number of [OSPF](#) routes was found by loading the routing table with advertised routes until the routing table could not hold any more.

Route convergence was tested by the following procedure:

- \* Send 80% of max number of supported routes.
- \* Send traffic over those routes to verify the routes.
- \* Withdraw 20% of the advertised routes to measure the amount of time for the router to adjust and to confirm that the router adjusted properly.
- \* Readvertise the withdrawn routes to verify that the routes are added to the route table, how long it

takes to add the routes, and if the router can properly handle the route operations.

---

[< Return to main Quagga story](#)

All contents copyright 1995-2006 Network World, Inc. <http://www.networkworld.com>

Sponsored by:



## NETWORKWORLD

This story appeared on Network World at

<http://www.networkworld.com/reviews/2006/100906-quagga-router-test.html>

# Open source Quagga router sets an acceptable pace

[Clear Choice Tests](#) By John Bass, Network World Lab Alliance, Network World, 10/09/06

**CLEAR CHOICE TEST** Open source routing software projects have been receiving attention lately as viable, inexpensive platforms for midlevel routing deployments. But are they practical for enterprise network managers used to the performance levels and feature bells and whistles served up by commercial router vendors?

That is the question we explore in this initial test of open source router software. Most open source router software spawns from one of two projects - Zebra and eXtensible Open Router Platform (XORP). Because the Zebra project has been dormant since mid-2004, we tested its descendant, Quagga, which is updated about every quarter. (We plan to test a XORP-based product later this year.)

Register or sign in now to view the entire article and access to Network World's Buyer's Guides.

[">Register/Sign in](#)

Overall, we found running Quagga on modest [Dell](#) servers yields performance numbers that would make the combination a formidable platform for a purpose-built routing application. If you're looking to run Quagga as a router in your production network, however, you may require a higher port density. This requires a server with more horsepower, which is more expensive. This additional cost may offset somewhat Quagga's low-cost appeal.

---

[How we tested the Quagga open source router](#)

[Archive of Network World tests](#)

[Subscribe to the Network Product Test Results newsletter](#)

---

Quagga runs on GNU/[Linux](#) 2.4.x and higher, FreeBSD 4.x and higher, NetBSD 1.6 and higher, OpenBSD 2.5 and higher, and Solaris 8 and higher. It supports Routing Information Protocol and RIPv2, RIPv6, Open Shortest Path First (OSPF), OSPFv2, OSPFv3, IPv6, Internet Group Management Protocol and IGMPv2, and Border Gateway Protocol 4.

Quagga's basic architecture consists of a Zebra daemon that handles updates to the [Unix](#)-based routing tables, and additional daemons for OSPF, RIP and BGP. Quagga has a unified configuration file for all daemons, which is easier to maintain than the original Zebra implementation that has separate configuration files for each daemon. The syntax of the Quagga configuration files is very much [Cisco](#)-esque. Quagga lets the user change IP addresses from the configuration file and command line; there is no need to configure IP addresses outside the router. The command-line interface is easy to use and intuitive (assuming some Cisco experience).

[Click to see: NetResults graphic for Quagga](#)

**OPEN SOURCE ROUTING SOFTWARE**

QUAGGA ROUTING SOFTWARE

## Quagga

- Pros:** It's inexpensive; has good performance; is easy to use if you are familiar with Cisco user interface.
- Cons:** Open source community is only source of support; requires a hefty machine to handle an enterprise level of ports; weak on features when compared with Cisco IOS.

### Access Router Buyer's Guide

Shop around for commercial competitors to the [Quagga open source router project](#).

## Measuring performance

To characterize IP packet-forwarding capabilities of the Quagga software running on two Dell machines, we ran throughput, packet loss and latency tests using a Spirent Smartbits 6000 results with 3301A 10/100/1000Mbps Ethernet interfaces interfaces (see [How we tested Quagga open source router](#)).

We compared a Cisco 2621 router (running IOS v12.1) with Quagga running on a rack-mounted Dell 1550 server (with single 933MHz Pentium III CPU, 256MB of RAM and two 10/100 Ethernet NICs). We also compared a [Nortel](#) 5520 switch with packet-forwarding capability with Quagga running on a Dell 2850 server (with dual 3.6GHz Xeon CPUs with 1GB of RAM and two 10/100/1000 Ethernet NICs).

Quagga running on the Dell 1550 performed at line rate (100Mbps) with packets larger than 256 bytes. For 64-byte packets, the Quagga box was able to forward 31Mbps of traffic. The Cisco 2621 didn't do very well in the forwarding tests, but we expected that, because it's an older entry-level router with a small processor (an MPC860 processor clocked at 66MHz), one that an open source router might be targeted to replace. The Cisco 2621 was able to forward 30Mbps worth of 1,518-byte packets and fell to 4.3Mbps with 64-byte packets.

The Cisco router did a better job of getting its packets through the router with less delay than the Quagga router. We saw a 630-microsec latency measurement for the Cisco 2621, compared with a 1,615-microsec latency for Quagga running on a Dell 1550. Keep in mind that increased latency results in decreased TCP throughput overall.

The Nortel box forwarded packets flawlessly at line rate for all packet sizes. Quagga running on the Dell 2850 combination hit close to line rate - 915Mbps - with 1,518-byte packets but could only forward 100Mbps with 64-byte packets. The packet-loss tests reflected the results of the throughput tests for the Nortel and Quagga systems. The Nortel router's latency was very low - 4 microsec. Quagga on the Dell 2850 had a 224 microsec average latency.

## Routing capabilities

To see how our Quagga/Dell installation fared in maintaining its routing table, we ran an OSPF-based route-flapping test in order to stress the router. The router had to change the routing table after receiving a huge number of changes at once from the routing protocol. We ran this test against both Quagga combinations and the Cisco 2621. The Nortel 5520 doesn't support OSPF in the configuration we had in the lab.

The Quagga platform accepts about 500 routes via OSPF. The Cisco 2621 was able to accept more than 2,000 routes. But as the lowest common denominator, we used a maximum of 500 routes for all our route-flap tests. All

the routers were able to recover from the flapped routes, which is worth noting. We've seen some routing implementations crash or begin dropping a significant number of packets after a route-flap event. That being said, the Cisco 2621 handled the route changes quicker. The Quagga installation took five to 10 seconds compared with Cisco's three seconds.

If you are interested in using Quagga as a platform to develop new routing features - such as a [firewall](#) or [intrusion detection](#) - go for it. Quagga performed respectably in all areas of our tests and should be a stable development platform in terms of function and performance. We didn't evaluate the extensibility of the Quagga code, however, so you'll have to determine whether Quagga is right for your development environment.

If you are interested in running Quagga in your production network, you'll need to consider its performance limitations. Quagga probably will perform better with faster hardware than we used in our tests, but that also raises the price bar, possibly negating a very important open source advantage. You must also consider the necessary port density required for your network, as performance will degrade at least linearly as you add more ports, and most servers have a limited number of NIC slots available.

*Bass is technical director and Zabala a network engineer at the Centennial Networking Lab of North Carolina State University in Raleigh. CNL tests network equipment and network-attached devices for interoperability and performance. They can be reached at [john\\_bass@ncsu.edu](mailto:john_bass@ncsu.edu) and [jzabala@cnl.ncsu.edu](mailto:jzabala@cnl.ncsu.edu), respectively.*



Bass and Zabala are also members of the Network World Lab Alliance, a cooperative of the premier reviewers in the network industry, each bringing to bear years of practical experience on every review. For more Lab Alliance information, including what it takes to become a member, go to [www.networkworld.com/alliance](http://www.networkworld.com/alliance).

All contents copyright 1995-2006 Network World, Inc. <http://www.networkworld.com>