A Fully-Programmable and High-Speed Virtual Network Platform

Yong Liao
ECE Department
University of Massachusetts
Amherst, MA 01003, USA
yliao@rio.ecs.umass.edu

Dong Yin
Automation Department
Northwestern Polytech University
Xi'an, ShanXi 710072, China
yindong@mail.nwpu.edu.cn

Lixin Gao
ECE Department
University of Massachusetts
Amherst, MA 01003, USA
lgao@ecs.umass.edu

Network virtualization provides a powerful way to facilitate designing, testing, and deploying network innovations over a shared substrate. As a first step toward the long term goal of providing a global infrastructure in which multiple virtual networks could run concurrently, each customized to a specific purpose, currently the network research community is focusing on building a shared, wide-area experimental virtual network platform to support a broad range of research in networking and distributed systems. To that end, the virtual network platform must have four key properties: (1) good *isolation* to minimize the interference between virtual networks; (2) enough *flexibility* so that a virtual network can be highly customized, this is particularly important for supporting new network innovations as they often require controlled network environment to conduct tests; (3) good forwarding *performance* so realistic experiments can be conducted and the platform can attract long term applications; and (4) decent *cost* in building that platform so as to make it easier for wide-area deployment.

The challenge of building such a virtual network platform is that those four key properties, i.e., *isolation*, *flexibility*, *performance*, and *cost*, are often tightly coupled issues in system design so that we usually have to compromise one of them in order to improve the other one. For example, the VINI platform [3] is successful in terms of isolation, flexibility, and cost, but suffers poor forwarding performance. The Trellis [4] system is not as flexible as VINI but its forwarding speed can be much faster. The VRouter system [2] reports a forwarding speed which matches the native software router forwarding speed but it loses much of the isolation in order to achieve that performance. The supercharging planetlab platform [5] provides superior forwarding performance but the cost of using special hardware (network processor) would be an issue for large scale deployment.

We have designed a new virtual network platform using cost-efficient commodity hardware and open source software. In this platform, we decouple the control plane and data forwarding plane of a virtual network and have different machines to perform those two tasks. More specifically, one "node" in this virtual network platform is actually a cluster of machines, with one of them working as the *management host* and the others are *forwarding engines*. Those machines are sliced by using operating system level virtualization such as OpenVZ [1], in order to share resources such as CPU and storage. The *forwarding engines* machines are virtualized to perform the actual packet processing tasks for the *management host*. A virtual network creator will feel that it is the *management host* which forwards the data in its network, but in fact the *management host* does only trivial multiplex/demultiplex work and all packet processing and forwarding functions, such as lookup and traffic shaping, are done by user mode software routers running on forwarding engines. Running user mode software router provides tremendous freedom to virtual network creators in implementing their own customized forwarding functions without jeopardizing isolation between virtual networks, i.e., a buggy forwarding plane of one virtual network will not impact other virtual networks. Having multiple forwarding engines compensates the poor forwarding performance of user mode software router so that the aggregate forwarding speed of the forwarding engines can match the native forwarding speed of commodity hardware.

In our poster, we will demonstrate the architecture of virtual network platform and show the design of the two entities, namely, the *management host* and *forwarding engine*. We have build a proof-of-concept prototype of this virtual network platform and we will provide some preliminary but promising packet forwarding performance measurement results in our poster.

References

[1] Openvz. http://www.openvz.org/.

[2] Virtual router. http://nrg.cs.ucl.ac.uk/vrouter/.

[3] A. Bavier and et al. In VINI veritas: realistic and controlled network experimentation. In *SIGCOMM '06*.

[4] S. Bhatia and et al. Hosting virtual networks on commodity hardware. Technical report, Princeton University, 2007.

[5] J. S. Turner and et al. Supercharging planetlab: a high performance, multi-application, overlay network platform. In *SIGCOMM '07*, 2007.