# Sliceable Routers from Multi-Core Processors

Mic Bowman and Rita H. Wouhaybi
Intel Research
Intel Corporation, Hillsboro OR
Email: {mic.bowman, rita.h.wouhaybi}@intel.com

May 23, 2007

## 1 Introduction

Network virtualization, the basis for isolating experiments in GENI and a proposed method for re-architecting the Internet, offers an interesting opportunity to re-examine the role of general-purpose hardware in high-performance packet processing. With the anticipated high-performance hardware of the near future, virtualization is a direction where many systems seem to be heading. Virtual routers, as a result, on shared multi-core hardware will become a necessity that researchers will require in large-scale testbeds, such as GENI. From a high-level perspective, the newly-introduced, as well as the perceived, multi-core systems seem to contain all the individual components and modules that made network processors such powerful and flexible routing devices. Yet, when comparing the performance of comparable systems with different architectures, a gap of several orders of magnitude remains. However, in order to deliver high-performance flexible virtual routers, researchers have to identify and implement modifications spanning the entire stack.

## 2 Challenges

The main challenges facing a scalable virtual router based on general-purpose computer are summarized as follows:

- Performance: In order to be successful, a virtual router should deliver high and predictable performance, matching line speeds.

- Isolation: This problem is not new, and is mainly due to the fact that virtual machines (VMs) sharing a physical machine are not totally isolated and their respective loads can affect each other. However, a virtual router adds a dimension where isolation is required in the networking part. When conducting an experiment, or even when building a virtual network using virtual routers, very often constraints are present on the connections of the different components. These constraints are challenging to maintain since the connections are shared with other traffic that is often unpredictable.

- Flexibility: Virtual routers have to provide their users with lots of flexibility in their configuration as well as monitoring their behavior. In other words, they have to mirror current router-in-a-box where a user with modest requirements can get them up and running with minimal effort and time, while at the same time, an experienced user can use for more complicated tasks and add modules seamlessly.

We target the ambitious goal of addressing all three challenges but start our solution by relaxing the requirement on the first two.

## 3 Existing Work and Approach

Looking at the existing work, this problem seems to be at the intersection of several fields and existing pieces of work. Researchers have proposed models and abstraction for routing on a general purpose computer as XORP in [3]. In addition, Click [4] has been used as a forwarding module that XORP as well as other routing platforms have used. Despite the flexibility of XORP and Click, their performance on top of existing OSes remain unsatisfactory, as they often lack way behind line speeds. Other abstractions, such as extending PacLang [5] for a general-purpose architecture will also be beneficial in achieving flexibility.

However, Click and XORP seem to be a good candidate for a start on top of a virtual machine platform such as XEN. This acted as a starting point for VINI as described in [1], however further optimizations can be implemented in order to improve the performance as was presented in [2].

Last but not least, hardware changes might be essential to meet the different challenges and requirements that were presented in the previous section. These might involve processor architecture changes for future generations as well as better network support as was suggested in [6].

## 4 Different Roles of Routers

We divide the tasks of a packet processing device into 3 categories that will be included in our testbed and experimentation. The categories are summarized as follows:

- *termination*: the packets are headed to the computer for consumption and need to reach the destination process on the computer and will be consumed there.

- *simple forwarding*: the packets enter the box with the intention of being routed somewhere else and are not for "local consumption." In here, it can be either a simple routing table or other manipulations might be happening such as NATing.

- *processing*: this case is the one that puts the most strain on the three challenges presented in the previous section. In here, the packets are examined more closely and possibly manipulated before they get out of the box for their next hop.

## References


[1] Andy Bavier, Nick Feamster, Mark Huang, Larry Peterson, Jennifer Rexford. "In VINI Veritas: Realistic and Controlled Network Experimentation," SIGCOMM 2006.

[2] Aaron Kunze, Stephen Goglin, Erik Johnson. "Symerton - Using Virtualization to Accelerate Packet Processing," ANCS 2006.

[3] Mark Handley, Eddie Kohler, Atanu Ghosh, Orion Hodson, Pavlin Radoslavov. "Designing Extensible IP Router Software," In the proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI) 2005.

[4] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. "The Click modular router," ACM Transactions on Computer Systems 18(3), August 2000, pages 263-297.

[5] Robert Ennals, Richard Sharp, and Alan Mycroft. "Task Partitioning for Multi-Core Network Processors," International Conference on Compiler Construction (CC) 2005. Lecture Notes in Computer Science Volume 3443, Springer-Verlag.

[6] Van Jacobson. "Architecting a Network on a Die," Presented part of Intel MC-Net, Nov 30, 2006.