A Flexible and Performant Virtual Router

N. Egi[1], A. Greenhalgh[2], M. Hoerdt[1], M. Handley[2], and L. Mathy[1]

[1] Computing Dept., Lancaster University, UK and [2] Dept. of Computer Science, University College London, UK

Abstract

We explore the building of a flexible and performant virtual router architecture constructed from off-the-shelf hardware components. We aim to show that it is possible to design an isolated and fair router system used by multiple users who share the same physical hardware platform by allocating each a virtual control plane and a virtual forwarding plane.

Using virtualization techniques we gain the ability to isolate different users' systems from one another whilst enabling each user to have a significant amount of flexibility in what they run and how they configure it, to the extent where different layer 3 implementations can run concurrently.

In earlier work [1] we showed that using entire system virtualization and undertaking forwarding in the virtual system gave rise to poor performance because of context switches amongst the entities. So whilst a virtual system is the ideal place for a router's control plane it is not an ideal location for virtualizing the forwarding plane where we avoid the context switches but maintain the isolation and parallelization of the virtual forwarding planes.

Motivation

Concurrent routers on a single platform
- Shared resources, more efficient resource exploitation
- Reduce overall cost
    - E.g. Small businesses in the same building sharing the architecture

Platform for experimentation
- Rolling out new and unstable solutions without risk
- Increasing the rate of innovation in the Internet core
- Isolation provided by the virtual router platform
- No physical change to the network is needed

The whole physical network can be shared
- Tunnels between virtual routers over shared network infrastructure
- Router migration

Generic router architecture

Evaluating Xen for Router Virtualization

(1) Evaluated and compared the forwarding performance of two identical Linux software router configurations run either above the Xen Hypervisor or within vanilla Linux.

(2) Even with minimal sized packets, we showed that the Xen Dom0 privileged domain offers near native forwarding, whereas Xen unprivileged domains offer very poor performance.

(3) This shows that an important design principle for virtual router platforms must be to handle all forwarding, for all virtual routers, in a single aggregated forwarding engine, in order to avoid much detrimental per-packet context switching.
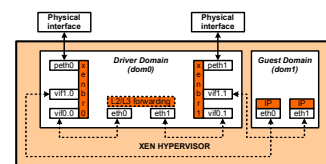

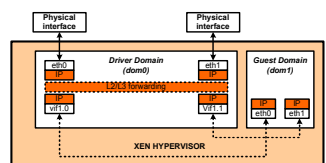FIGURE 1: Xen's classical network internals
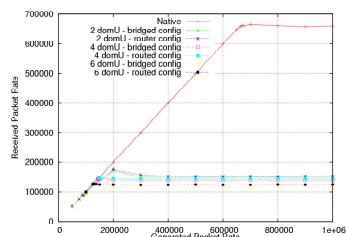

FIGURE 2: Xen's routed network internals

Enabling Technologies

Click Modular Router [2]
- Provides a dynamically reprogrammable forwarding plane
- Fine-grained components, which support fine-grained extensions
- Hot-swap, dynamic reloading

XORP Extensible Open Router Platform [3]
- Provide the control plane and routing protocols
- Enables extensibility by the use of XRLs

XEN Virtual Machine Monitor [4]
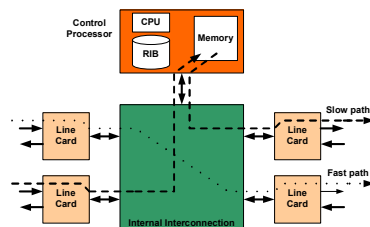- Securely executes multiple virtual machines on a single physical system

FIGURE 3: DomUs aggregated forwarding performance in bridged and routed setup with different numbers on domUs VS native Linux's performance
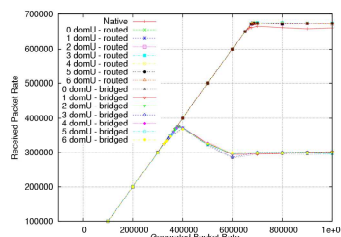

FIGURE 4: Dom0 bridged and routed forwarding performance with different numbers on domUs switched on (but not forwarding whatsoever) VS native Linux's performance

Design requirements

- Flexibility, modularity and programmability
- Automatic analysis of the virtual forwarding paths
- Performance
- Fairness
- Resource isolation

Conceptual Virtual Router

Realization of the Virtual Router Forwarding plane
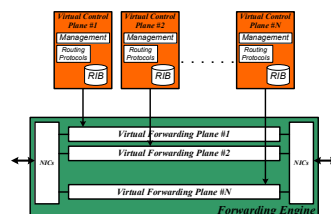
Forwarding Plane

Packet classification is an important issue with respect to the scheduler in the context of a virtual forwarding plane :

- Packets are classified according to a fixed rule (the MAC address in our design)

In consequence, the scheduler is in control of the flows to be processed as it isn't driven by the arrival pattern of packets.

Scheduling with different Forwarding path scenarios :
- **Static virtual forwarding plane**
    - Identical FPs approved and provided by the system
    - FPs' costs are know and are very similar
    - Basic Round-Robin FP scheduling is sufficient
    - **Scheduler:** Click's original (Stride scheduling based) CPU scheduler

- **Configurable virtual forwarding plane**
    - Custom made FPs, but only from elements approved and provided by the system
    - FPs' costs are unknown (hardly predictable) and can differ significantly
    - FPs' costs need to be measured and a weighted CPU scheduler is needed
    - **Scheduler:** Click's original CPU scheduler has to be extended with an FP cost measurement tool and a dynamic ticket modifier

- **Customizable virtual forwarding plane**
    - Custom made elements are allowed to be included in the FP, but in an isolated environment only, provided by the virtualization technique
    - Interface(s) to the isolated elements need to be provided
    - FPs' costs are unknown (hardly predictable), can differ significantly, and is consisting of two separate parts (shared and isolated parts of the FP)
    - FPs' costs need to be measured and a weighted CPU scheduler is needed in the shared forwarding engine, while the CPU scheduler of the virtualization technique has to be tuned too
    - **Scheduler:** Click's original CPU scheduler has to be extended with an FP cost measurement tool and a dynamic ticket modifier and Xen's scheduler has to adjusted for providing sufficient resources to domUs

References

[1] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, L. Mathy, and T. Schooley. *Evaluating Xen for Virtual Routers*. In *PMECT'07*, Honolulu, HI, USA, August 2007.

[2] E. Kohler, R. Morris, B. Chen, J. Jahnotti, and M. R. Kasshoek. *The Click Modular Router*. In *ACM Transaction on Computer Systems*, 18(3):263-297, 2000.

[3] M. Handley, O. Hodson, and E. Kohler. *XORP: An Open Platform for Network Research*. In *HOTNETS'02*, October 2002.

[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. *Xen and the Art of Virtualization*. In *19th ACM Symposium on Operating Systems Principles*. ACM Press, October 2003.