

August 2007

Using Xen For Router Virtualisation

Adam Greenhalgh, Mark Handley, Tim Schooley
University College of London

Norbert Egi, *Mickaël Hoerd*t, Laurent Mathy
University of Lancaster

Outline

- Motivations.
- Xen network internals overview.
- Experiments.
- Results.
- Conclusions and further work.

Motivations

■ Virtual Routers: Why?

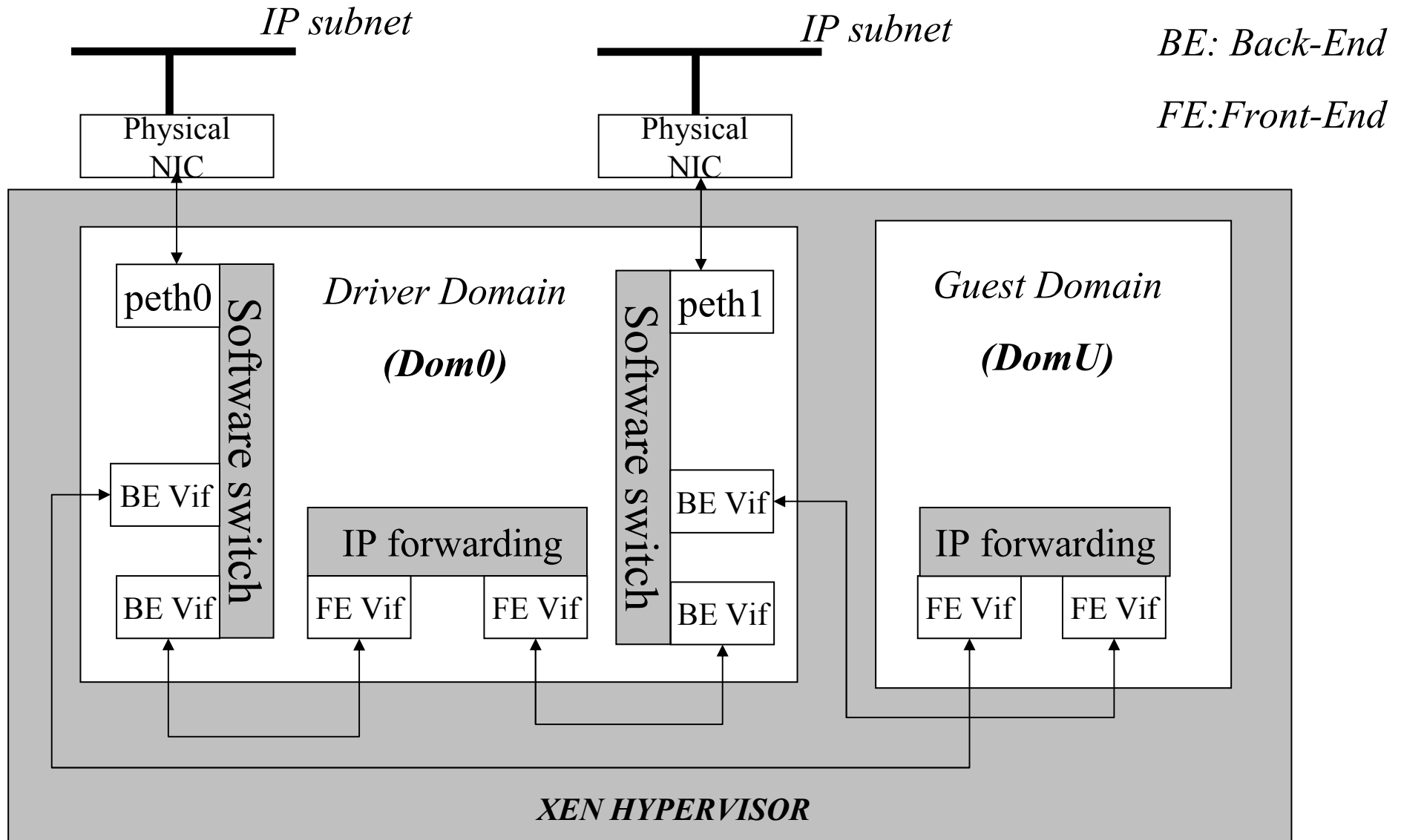
- One box can play the role of multiple independent routers.
 - Resources sharing, management flexibility
- Multiple organizations sharing a single physical router.
 - Small businesses within one building.
 - Internet eXchange Points.
- Entire physical network can be shared.
 - Virtualize the routers.
 - Tunnel between virtual routers over shared IP infrastructure.
 - And then you have an Inter-domain overlay
- Great platform for experimentation (cf. VINI).

Motivations

- Enabling technologies:
 - Click Modular Router
 - Provides dynamically reprogrammable forwarding plane
 - Xorp Extensible Open Router platform
 - Provides an extensible open source control plane
 - Xen Virtual Machine Monitor
 - Hardware support on commodity PCs
- What could we build with that ?
- Can we achieve good performances ?

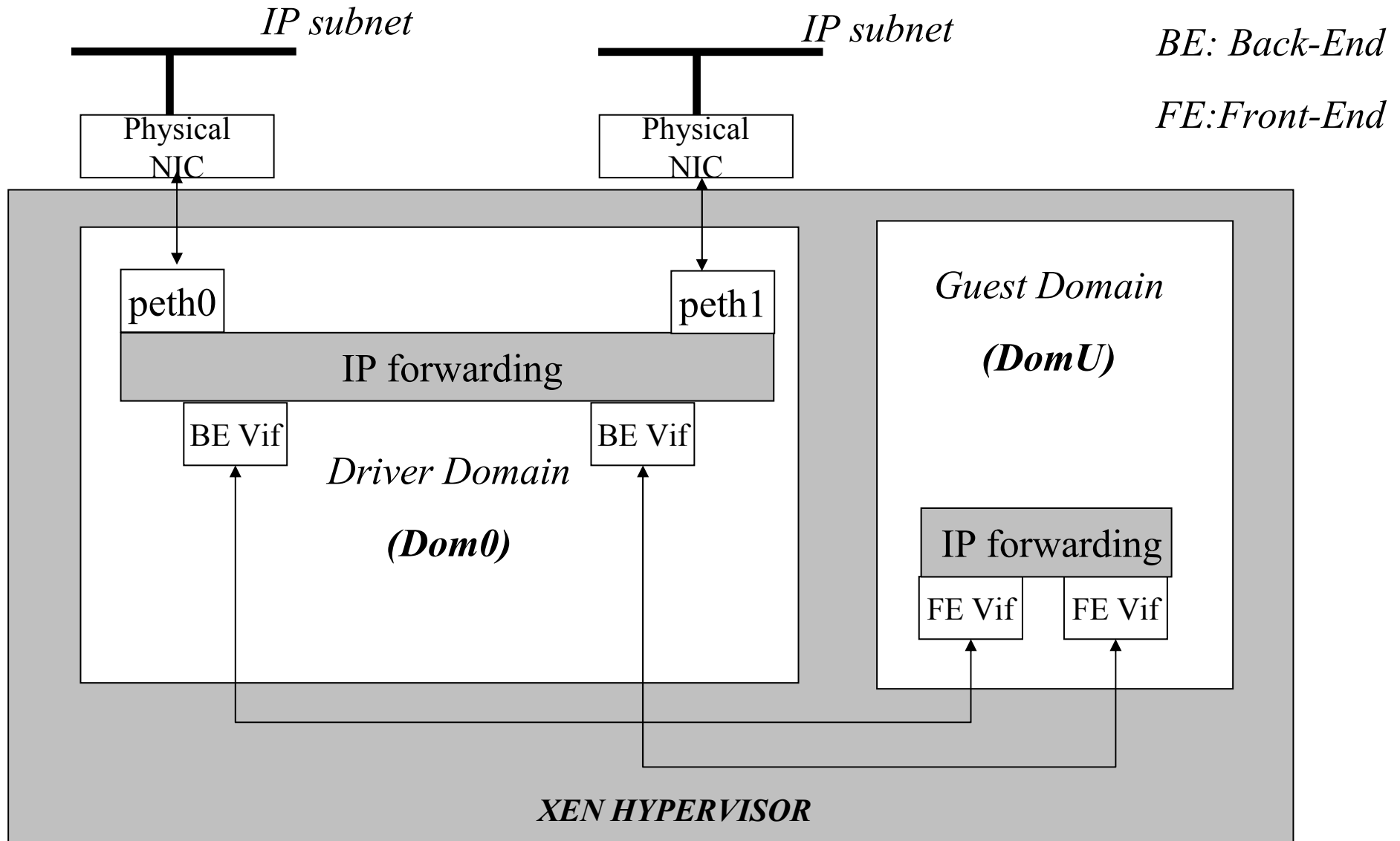
Xen network internal overview

- Xen classical bridged setup



Xen network internal overview

- Xen classical routed setup



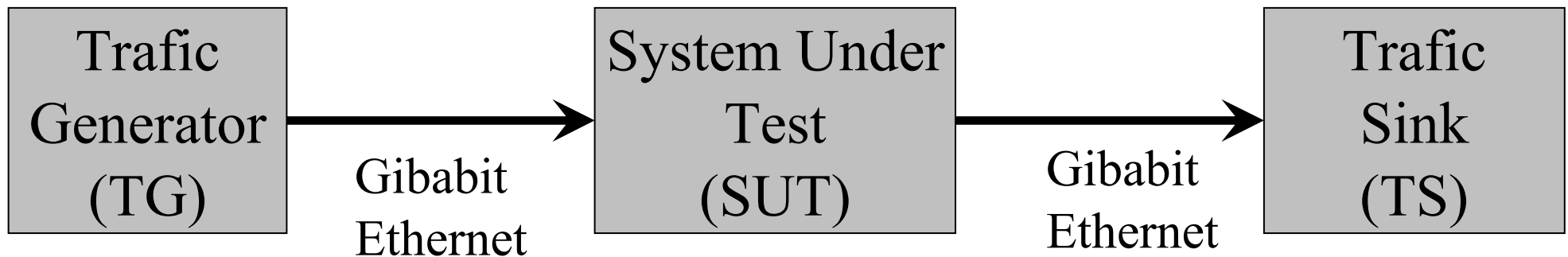
Opened questions

- Question : Is it worth pushing the forwarding plane of a domU down to dom0 ?
 - How does dom0 forwarding performances compare with native linux performances ?
 - What is the impact of increasing the number of domU's on dom0 forwarding performances ?
 - What is the impact of the classical routed and bridged Xen setup on the forwarding performances ?
 - How does the forwarding performances of dom0 compare to the forwarding performances of a domU ?

Experimental setup

- A three node topology :

- Vanilla Linux 2.6.16.33
- **OR**
- Xen Linux + Xen 3.0.4-1



TG : Generate a 10s CBR packet stream at rate ranging from 100 kpps to 1000 kpps, packet size 64 bytes, granularity 100kpps

SUT: Forward packets using native linux or dom0 or/and domUs as routers

TS: Receive packets and measure the rate

Results: Dom0 only

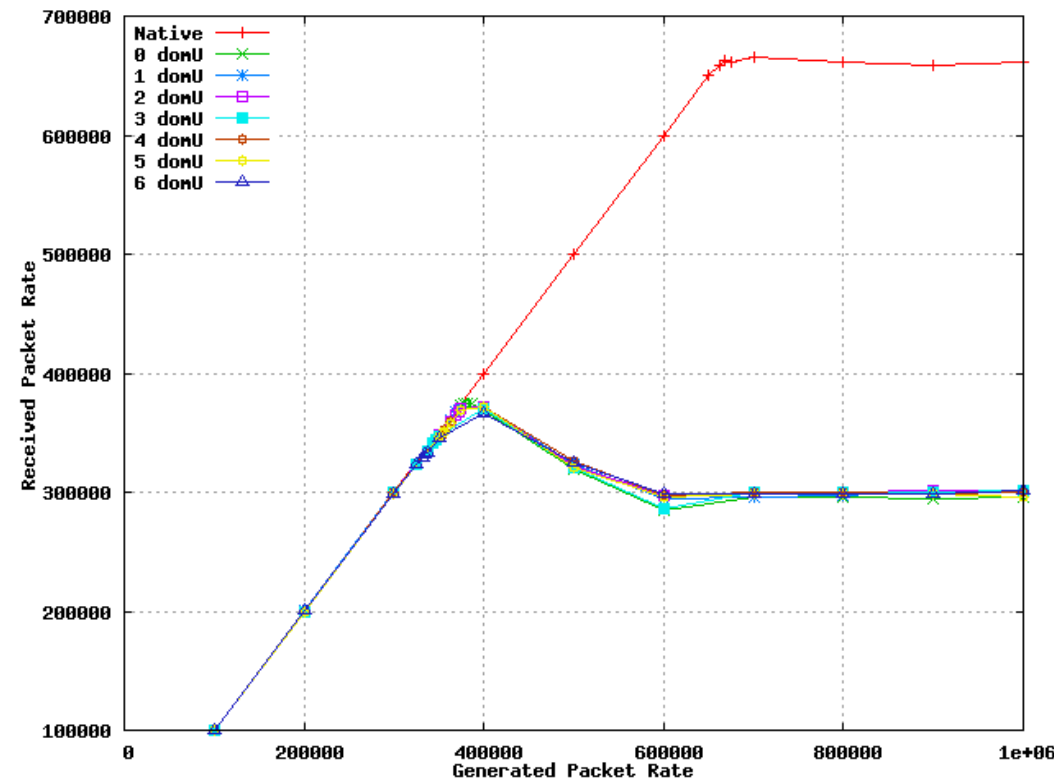
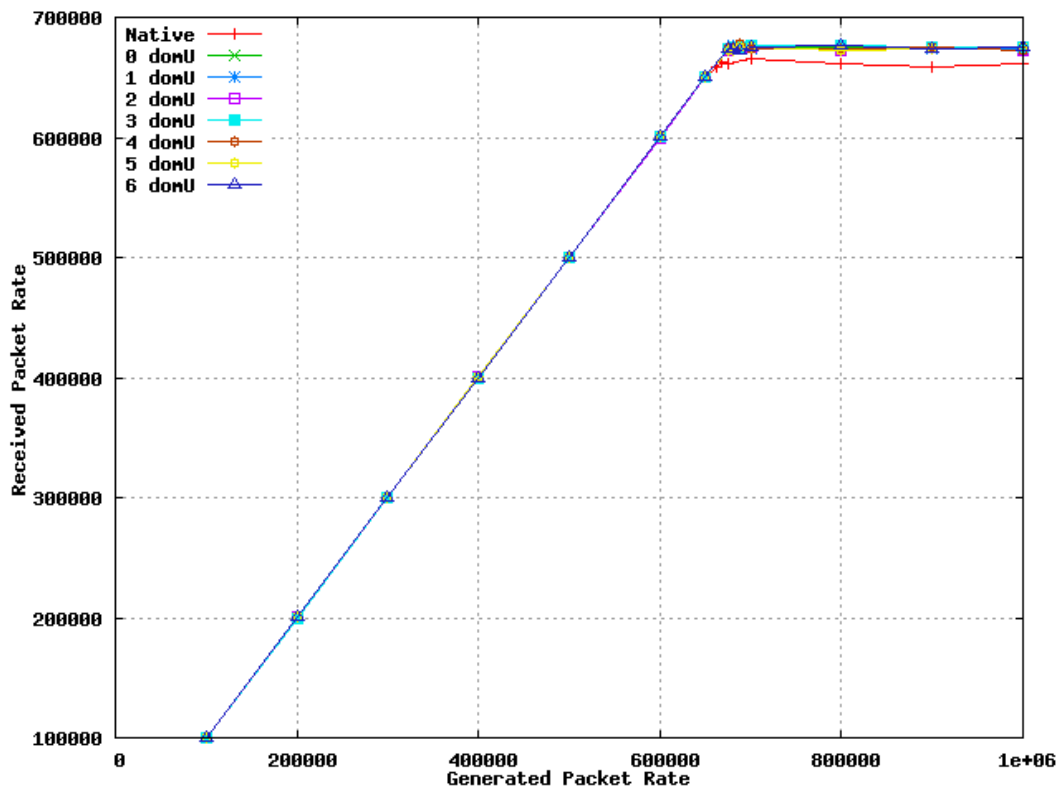
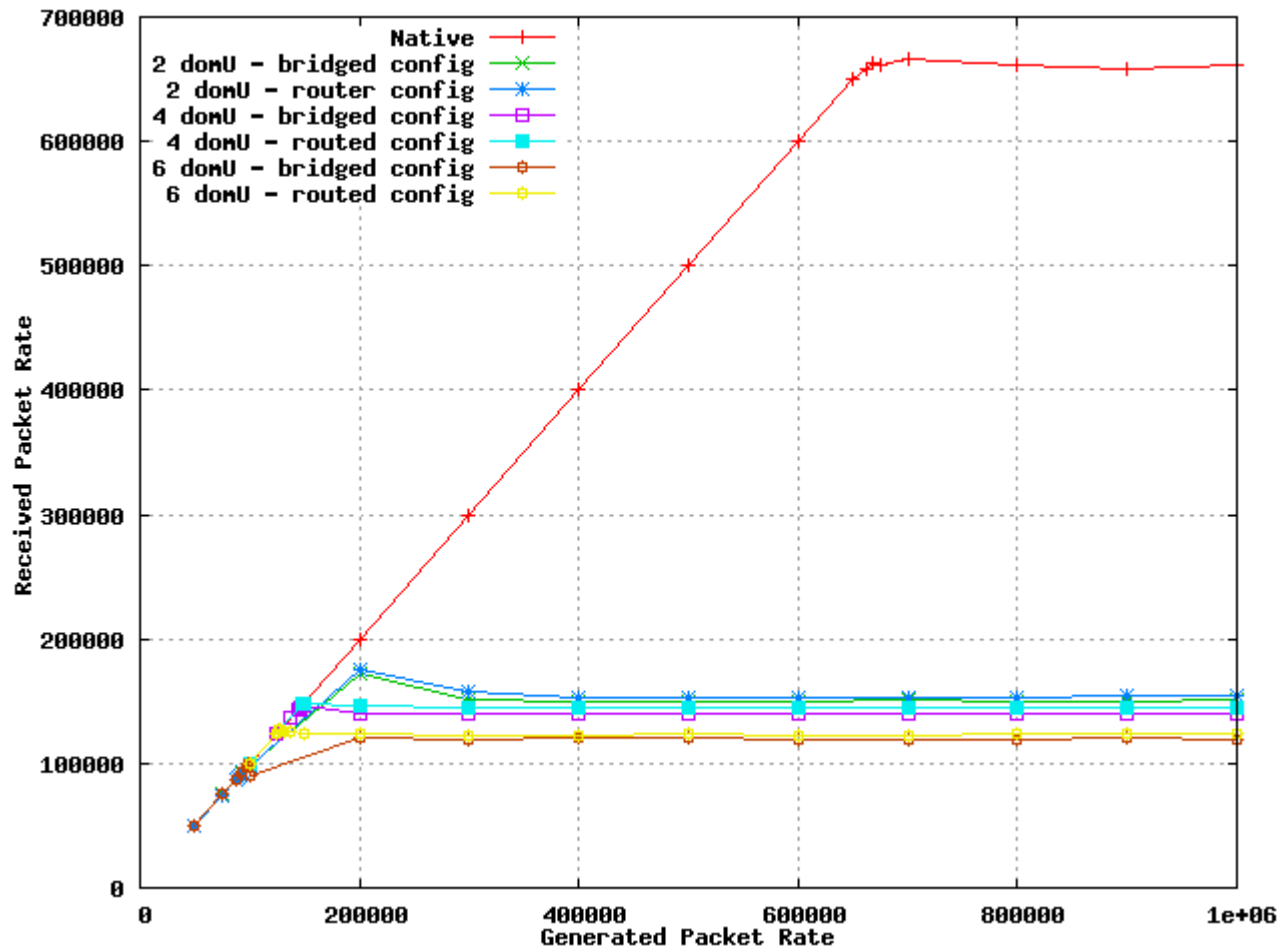


Fig 1: Dom0 forwarding performances in **routed** setup with different number of domUs vs native linux

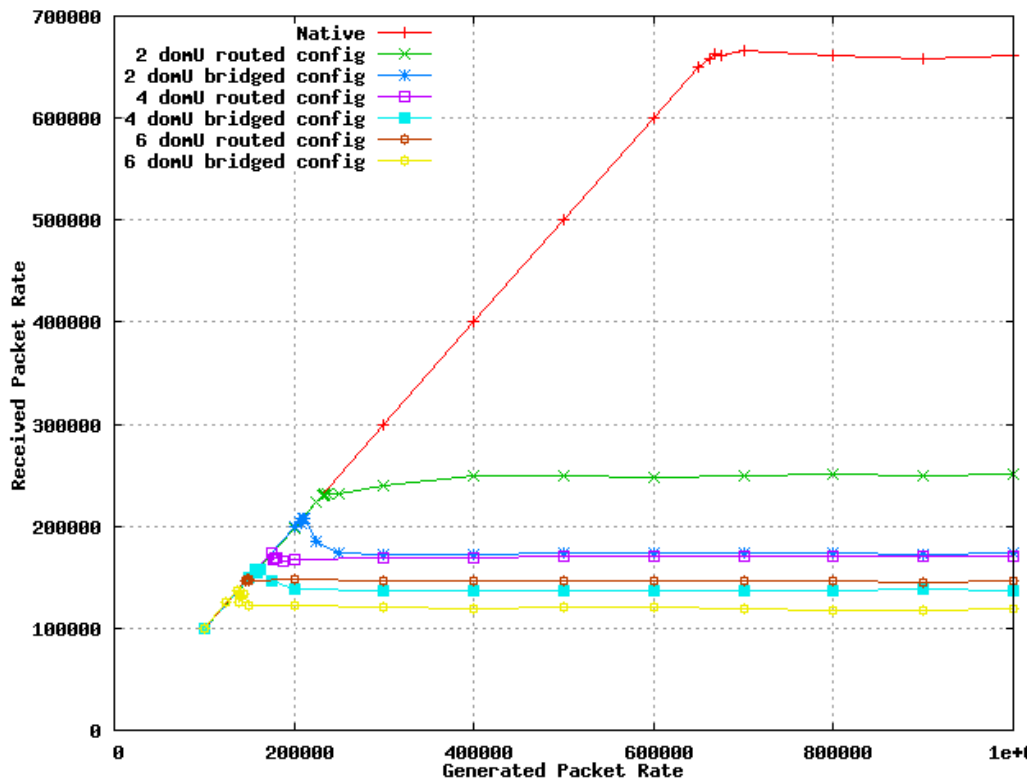
Fig 2: Dom0 forwarding performances in **bridged** setup with different numbers of domUs vs native linux

Results: DomUs only

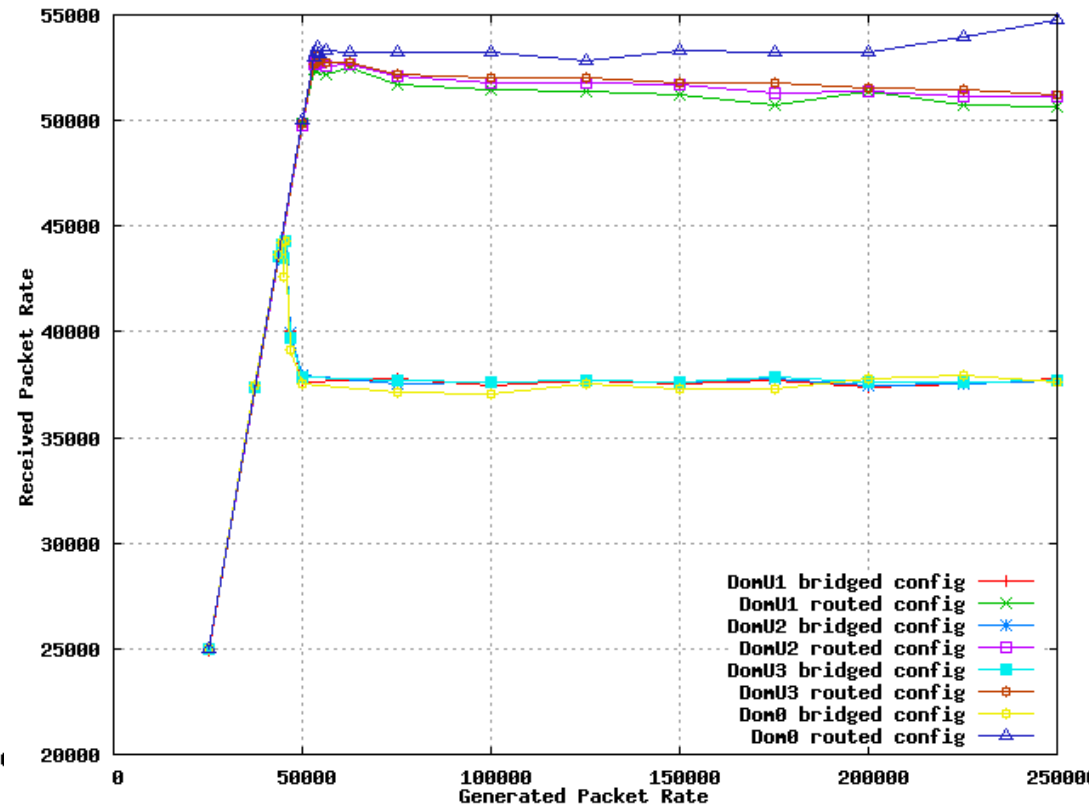


DomUs only aggregated forwarding performances in **bridged** and **routed** setup with different number of domUs vs native linux

Results: Dom0 and DomUs



Dom0 and DomU aggregated forwarding performances in **bridged** and **routed** setup with different numbers of domUs vs native linux



Dom0 and DomUs flow breakdown forwarding performance in **bridged** and **routed** setup with 3 domUs

Conclusions and Further work

- Conclusions:
 - Dom0 forwarding performances are good compared to native Linux!
 - *But* DomU's forwarding performances aren't.
 - *And* DomU's forwarding impact Dom0 performances badly!
- Further work
 - Use Click and Xorp to design a programmable shared forwarding plane running in dom0 on the behalf of the domUs

Results: 6 DomUs+Dom0 flow breakdown

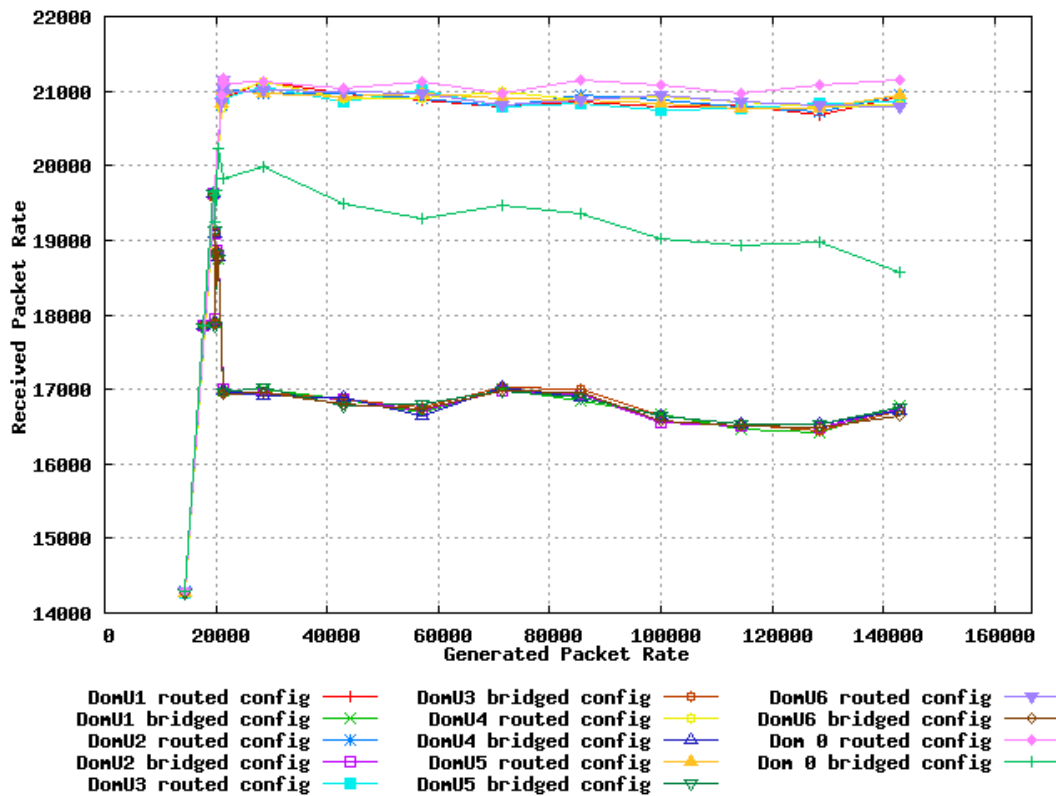


Fig 1: Sun Fire X4100 Dom0 and domU flow breakdown forwarding performances in **bridged** and **routed** setup with 6 domUs

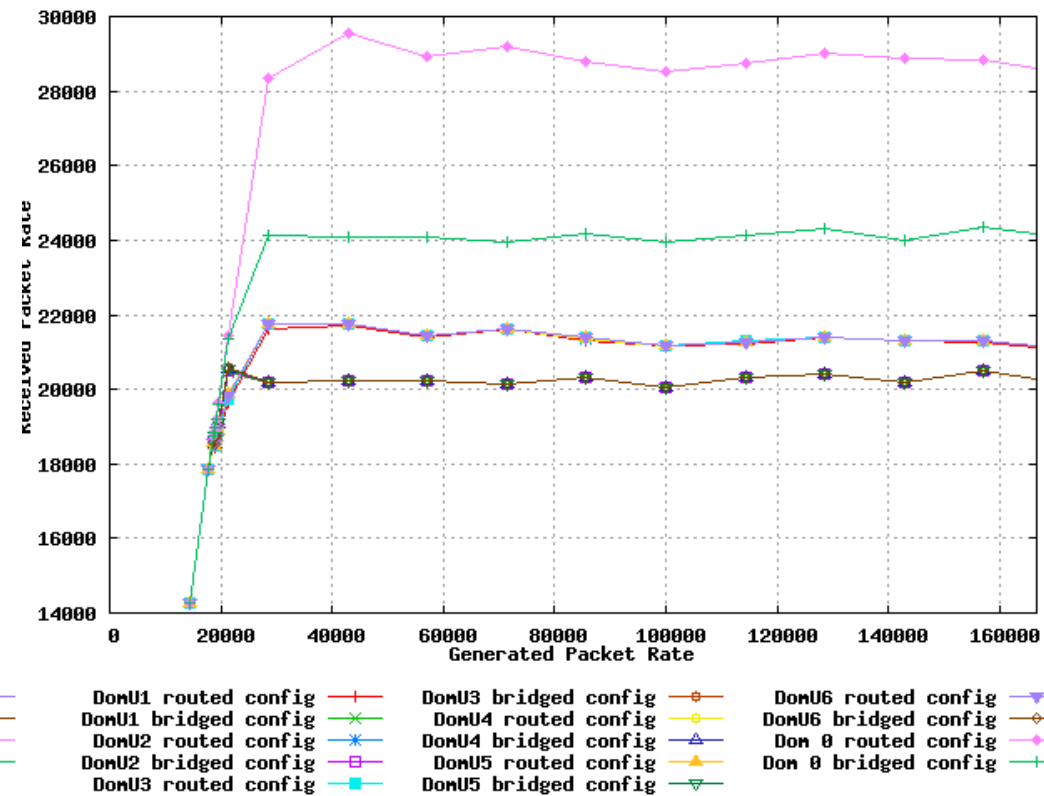
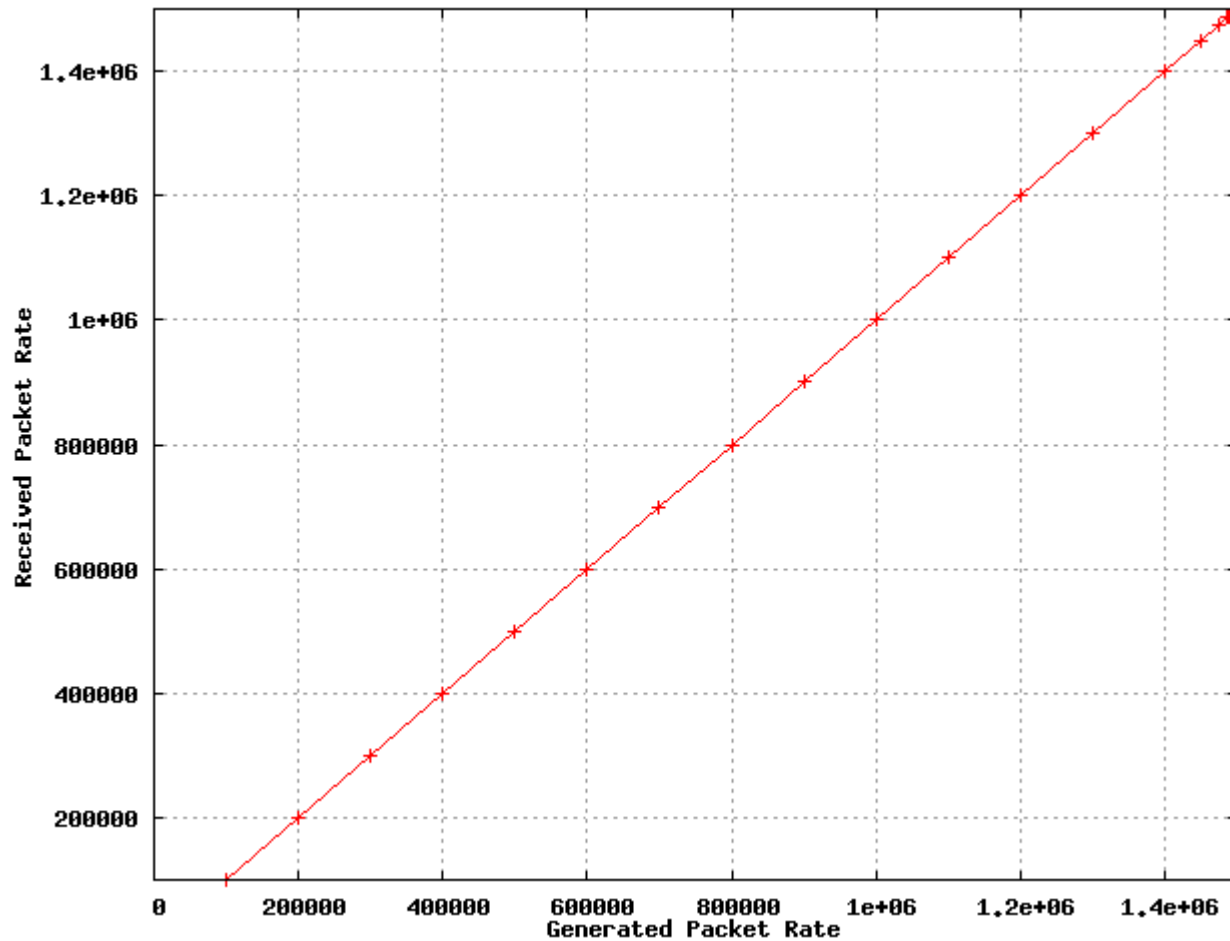


Fig 2: Dell 1950 Dom0 and domU flow breakdown forwarding performances in **bridged** and **routed** setup with 6 domUs

Experimental setup

- Checking TG to TS performances:



TG to TS IP link performances