GÉANT
Networks • Services • People

Open Source Routers

Best Practice Document

Produced by the Portuguese CBP working group (DBPC-303)

Authors: Jorge Matias (IST, Lisbon), Israel Lugo (IST, Lisbon), Rui Ribeiro (ISCTE Business School), Tiago Sousa (Univ. de Évora), Carlos Friaças (FCCN)

November 2015

Document No: GN3-DBPC-303
Version / date: Version 1.3; November 2015
Original language : English
Original title: “Open Source Routers”
Original version / date: Version 1.0; December 2014
Contact: cfriacas@fccn.pt

FCT-FCCN is responsible for the contents of this document. The document was developed by the Portuguese CBP working group (DBPC-303).

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

Table of Contents

Table of Figures

Executive Summary

Open source routers are a viable alternative to traditional vendor-based routing platforms that can be used by Campus Network Managers. This option involves investment in developing local knowledge about routing platforms, and in the medium/long run it could turn out to be a good decision cost-wise. This approach was chosen by the Technical University of Lisbon (UTL), one of the largest members of RCTS (the R&E network in Portugal), after seeing traffic levels increasing and low performance on existing vendor-based routing equipment.

Architecture is a key factor in deciding if this is the correct approach, together with the capacity to deal with high levels of traffic in a single physical interface. While routing can be completely managed through static routes, the support of both interior and external routing protocols is an obvious improvement.

Most of this document's content focuses on UTL's experience, but other members of RCTS have also started to use open source routers, namely the University of Évora, ISCTE-IUL, Lisbon's Polytechnic Institute and the Nursing School of Coimbra, among others. Not all of these are using 10G interfaces, because their networking needs are less than 1 Gbps of capacity/bandwidth.

1 Introduction

The basis for this work is the routing ecosystem's evolution at the Technical University of Lisbon (UTL), which was merged in 2013 with the Classical University of Lisbon, forming the University of Lisbon (ULISBOA). As of April 2015, the full network merge, which may result in some extra global improvements, is still to be completed.

The decision to move to open source routers was taken in 2003, due to poor performance on vendor routers and the need for IPv6 support at that time. The Technical University of Lisbon originally established its IPv6 connection to the NREN (RCTS) in June 2003, two months after the NREN deployed its native IPv6 connection to GÉANT (the pan-european research and education network). There was also some previous experience with open source border and wireless routers at Instituto Superior Técnico (IST), a unit of UTL.

The poor manageability of security (with access control lists, stateless firewalls) was one of the strong arguments to evolve towards a different setup. Performance was also one of the most important issues, given that only 22.5 Mbps of traffic were enough to take the processor's occupancy to 70% on the vendor routing platform.

2 Hardware and Software

In 2003, UTL's vendor routing platform was replaced by two HP DL360G3 servers, with two 1Gbps ports each.

After the last upgrade, the platform is now composed by two HP DL360G6 servers, with:

- Two 64-bit quad-core CPUs @ 2.66GHz.
- 12GB RAM.
- Two 10G SFP+ Ethernet ports.
- SolarFlare SFN5162F (SFC9020 chip).

2.1 Pros

- Lower cost (acquisition and maintenance).
- Flexibility.
- Ability to create tunnels of any type.
- Creating a transparent HTTP Proxy.
- Firewall capacities.
- Quicker diagnosis and correction of OS bugs.
- Quicker routing software update.
- Diversity of routing software (Quagga [1], OpenBGPd [2], BIRD [3], XoRP [4], Vyatta).
- Trafic analysis tools (tcpdump, tshark, ifstat, iftop, nettop, nethogs).
- Usage of OpenFlow, sFlow.

2.2 Cons

- Slower installation and configuration time.
- Requires choosing the best components.
- Significant optimisation is needed in order to obtain the hardware's maximum performance.
- Resistance to DDoS attacks.
- It's not a turnkey solution.

2.3 Hardware

2.3.1 10Gbps

One CPU (one core) is unable to process 10Gbps, thus several cores are necessary. We should not expect full 10Gbps capacity from any previously untested 10Gbps Network Interface Cards (NICs) [5]. The major issue in very high bandwidth networks often is not the I/O capacity nor CPU power, but interrupts and the card's capacity to offload interrupts from the system's CPU [6].

Thus, there are several aspects that need to be analysed in order to validate the capacity of any 10 Gbps NIC being considered:

- Receive-side scaling.
- SMP IRQ to CPU affinity.
- Internal NIC latency.
- Drivers' quality.

2.4 Software

The distribution and the chosen kernel version might have a decisive impact on the overall solution. We strongly recommend not using old versions (3+ years). At UTL, after choosing Linux over *BSD, the selected distribution was Debian/Wheezy, with a v3.16 Kernel (amd64/x86_64).

This kernel version was ported from Debian/Jessie, from the Debian backports repository. This addresses the enormous time distance between two "stable" releases. Kernel versions from 3.6.0 are particularly interesting for routing purposes, as they no longer have a routing cache [7]. The routing cache hinders parallelism, is vulnerable to poisoning by in-band traffic and can be thrashed by legitimate traffic from varying endpoints [8].

Even for 1Gbps routing, testing at IST showed a noticeable improvement in routing performance by going from Linux 3.2.0 to Linux 3.16.0. The latter was capable of forwarding at line rate (1.4 Mpps) in modest hardware without much system impact, even with highly variable host addresses.

Organisations installing a new solution from scratch should consider using the latest version of Debian (at this time Jessie). If your new setup is being designed to support significant instances of IPSec tunnels or to apply traffic shaping [9] policies, you might also want to consider using FreeBSD [10] instead of Linux.

Other routing software, which was found to be essential, was:

- KeepAlived [11], UCARP – VRRPv2.
- Quagga - BGP, OSPFv2 and OSPFv3.
- XoRP- PIMSMv2.
- Firewall – iptables / ip6tables.

When deciding whether to implement stateful firewalling, one must consider that connection tracking will seriously damage parallelism, by introducing contention and locking. For very high packet rates, stateful firewalling is not recommended.

2.4.1 BGP through open source software

The network routing software suite Quagga was the open source software chosen to deploy the new routing architecture at UTL. Quagga provides implementations of OSPFv2, OSPFv3, RIP v1 and v2, RIPng and BGP4 for UNIX platforms. Quagga is a fork of GNU Zebra, which was developed by Kunihiro Ishiguro. Quagga's official website is located at http://www.nongnu.org/quagga.

Quagga uses text configuration files, and allows Cisco-like command line interface, either locally (vtysh) or remotely (using 1 tcp port for each routing module/protocol). One important feature is the support of 4-byte ASN (Autonomous System Numbers), along with an extensive IPv6 support (through the protocols mentioned above – OSPFv3, RIPng and multiprotocol BGP). The main traffic engineering functions, which are commonly available on Cisco routers for the BGP protocol are also supported. More details about Quagga and routing protocols can be found at Campus Best Practice document "Dynamic Routing Protocols for Campuses" [12].

3 Architecture

3.1 University of Lisbon

The following pictures describe the former and current network architecture at the Technical University of Lisbon and its faculties, including IST.

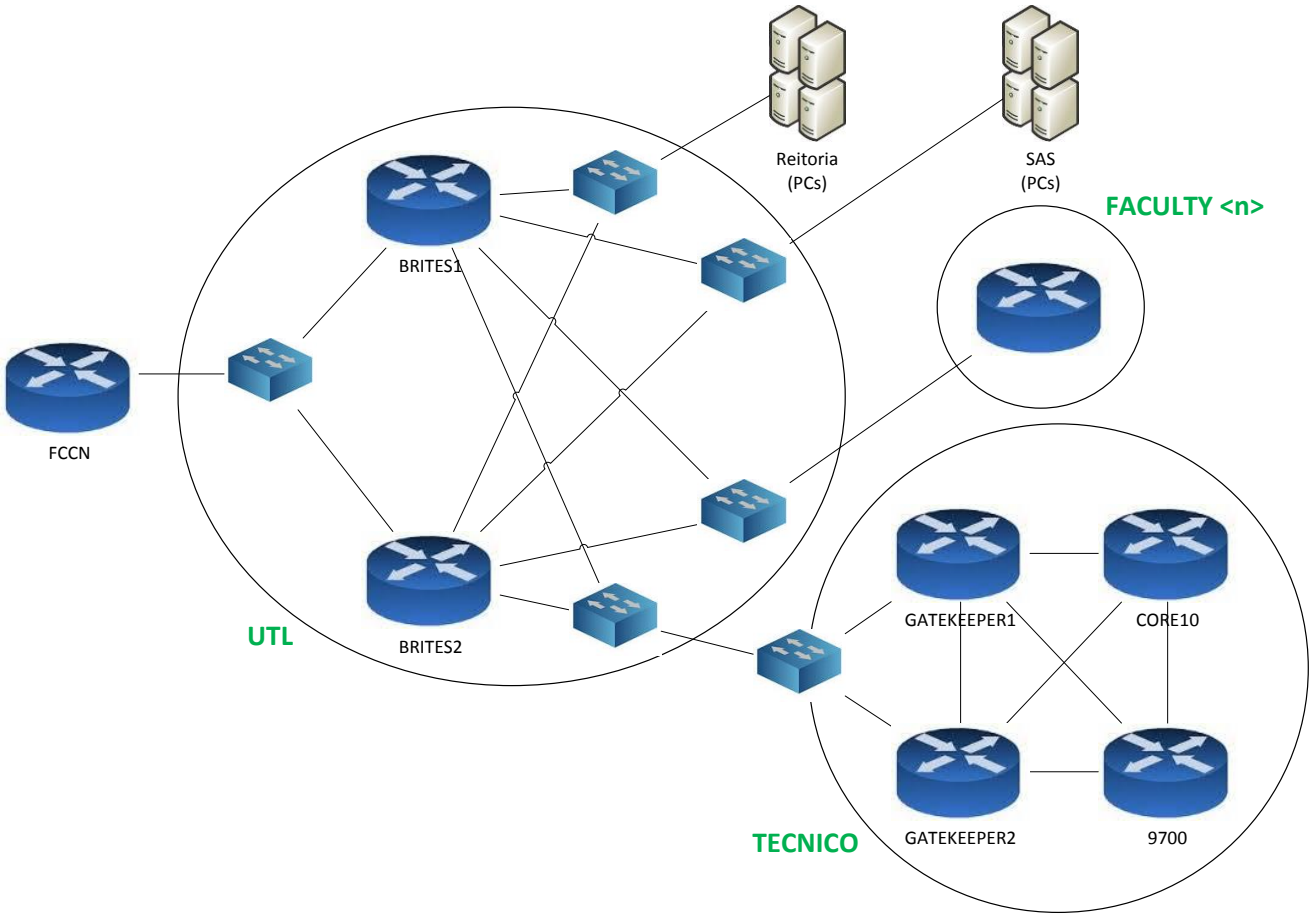

Figure 3.1: Former network architecture at the University of Lisbon

Communication between the single NREN router and UTL was based on VRRP configured on UTL's side, which was simple to deploy and also benefited from a very quick failover (3 ~ 6 secs). However, this architecture didn't deal with the split-brain issue or with network interface cards which keep on transmitting but stop being capable of receiving – this behaviour is invisible to VRRP and stops the MASTER re-election. The setup also didn't allow routing through multiple border routers simultaneously, which inhibits exploration of multiple links in order to perform traffic optimisation. But probably the most negative characteristic is the requirement to deploy static routes.

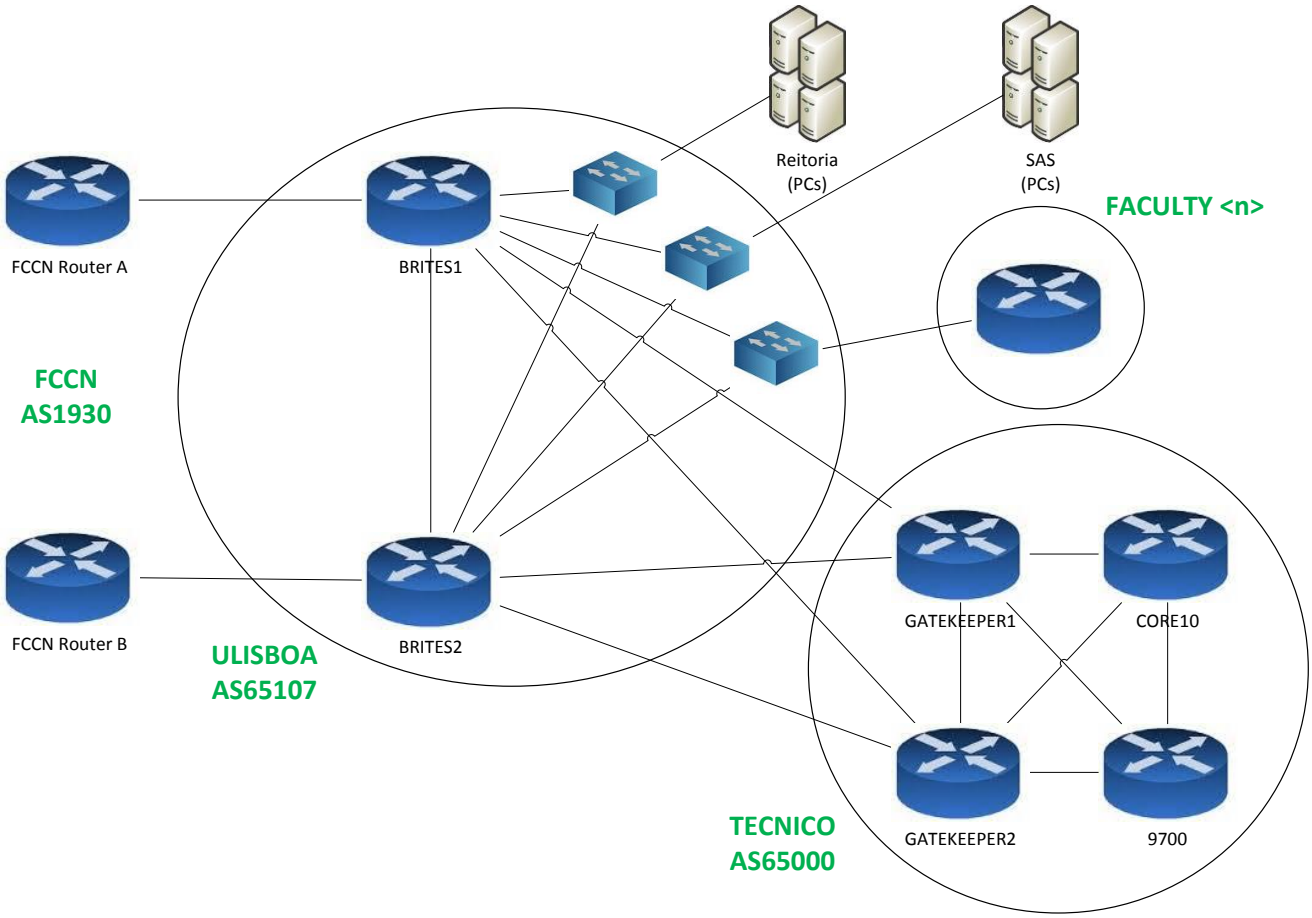

Figure 3.2: Current network architecture at the University of Lisbon

Using BGP as a basis for UTL and NREN router communication made the architecture more tolerant to more failure scenarios. The split-brain issues on either network stopped being a problem for these routes, and at the same time started to allow traffic engineering enabling the usage of multiple links in order to optimise traffic. However, it also requires some training/knowledge about BGP and failover is slower (20 secs ~ one minute) even with optimised BGP timers, when compared to the previous architecture.

3.2 University of Évora

University of Évora's experience with Quagga, using BGP and OSPF has been perfectly stable and flexible. The solution is built over Debian Wheezy.

In the beginning, there were some difficulties configuring route redistribution from BGP to OSPF. Some commands only worked in OSPFv2 and not OSPFv3, or vice-versa, such as "default-information originate" and "network". After some workarounds and fine-tuning, everything was solved.

Quagga only supports one OSPF process/zone, which might cause some constraints depending on specific scenarios – this was not a problem for University of Évora, given that Quagga is only used on the border with RCTS. Due to this reason, among others, BIRD should be an alternative to Quagga for someone which is still starting from scratch.

University of Évora's solution is built over Intel 82599 10 Gbps NICs, mainly because Intel hardware is perceived to have better support with open-source software. The current connection with RCTS is, however, still established at 1 Gbps. For this reason, NIC tuning hasn't been necessary yet.

3.3 ISCTE-IUL

ISCTE-IUL's experience with Quagga comes from providing anycast/failover support for DNS servers using OSPF. The solution has been used since 2012 and nowadays is built over Debian Jessie.

DNS servers' IP addresses configured over the organisation's infrastructure are virtual. They are mapped to their real IP addresses via Quagga routers running OSPF on each server. One part of the node acts as a router/firewall on the campus backbone.

Employing a judicious use of costs allows the mapping of each VIP DNS server to each of the DNS servers, thus redistributing the load evenly upon a failure of one of them. Using monit, the service's health is also monitored, and if something unexpected happens the Quagga service is stopped.

When the monit service is stopped, either forcibly or due to a crash or shutdown of one DNS server, the VIP floats to another server. This functionality also allows for stopping a server intentionally, taking it out of the cluster, for maintenance operations. Doing it on turns allows effective maintenance (and even rebooting) when there are new kernel versions to be deployed, without interrupting the DNS service.

Each server has dummy interfaces with all the DNS VIP addresses. The one IP address that answers to the requests on each one is selected via OSPF and is injected on the backbone's routing tables.

It is ISCTE-IUL's experience that processes over Linux spend a small amount of resources. Quagga, for instance, is only using 5MB of RAM, on Debian-based servers (running DNS+Quagga) with a total memory between 512MB and 1GB.

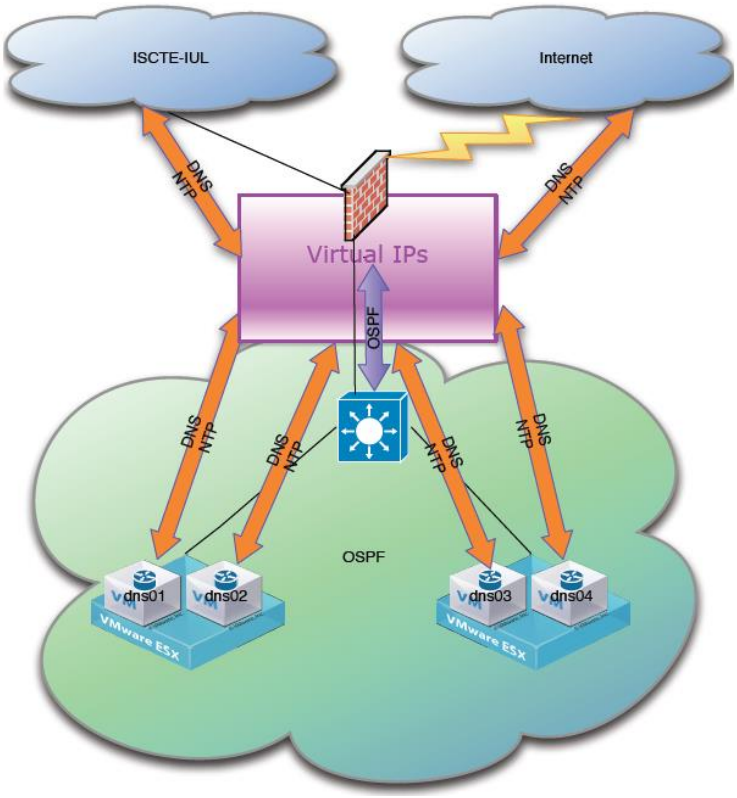

Figure 3.3: Virtual network architecture at ISCTE-IUL

4 Performance

Tools like NTOP [13] or IFSTAT [14] can be used to evaluate performance, occasionally. ETHTOOL [15] is also an important tool to retrieve a physical NIC's status. The NETHOGS [16] tool shows how much each process is using in terms of traffic, and can be used, mostly to optimise/detect any optional functions, given that most of the traffic within a router is dealt by the kernel itself. DTRACE [17] should also be useful to track at the kernel level what exactly is being used.

Performance on a routing platform solution can be degraded by the amount of services embedded. If choosing to build a firewalling solution together with your routing platform, you must be aware of the impact. Using iptables, especially with conntrack can surface several issues, of which dealing with packet fragmentation can be a real challenge.

Conntrack, associated with iptables, is a key component if you need to perform IP accounting, but it will have an impact on performance. So if you really need this feature, you will need to evaluate how the global system behaves with a significant load in terms of traffic.

Latency will typically increase when several services are combined with routing on the same platform. If the impact is visible, instead of performing any eavesdropping and traffic analysis on the hardware which is doing the routing function, one option would be to use port mirroring -- if a switch is available before the traffic reaches the routing hardware. A different option would be to use CPU pinning, which should provide more control to auxiliary functions/processes, and reduce possible damages to performance.

5 Additional Usages

An open-source routing solution or an open-source routing architecture may contain other capacities apart from the routing function. In this chapter we plan to briefly perform some considerations about several possible additional usages.

5.1 Anycast

Open-source routers in general (and Quagga in particular) are very versatile to create anycast configurations. This is useful to create a cluster of DNS servers, which, by making use of OSPF and BGP metrics allows anyone to define which node starts to answer queries when a fellow node dies.

Sample Configuration:

```
!
! Zebra configuration saved from vty
!   2011/03/24 15:42:46
!
hostname ospfd
password 8 xxxxxxxxxxxxxxxxxx
enable password 8 xxxxxxxxxx
log stdout
service password-encryption
!
!interface dummy0
 ip ospf cost 900
!
interface dummy1
 ip ospf cost 1000
!
interface dummy2
 ip ospf cost 100
!
interface dummy3
 ip ospf cost 500
!
interface eth0
 ip ospf authentication message-digest
 ip ospf message-digest-key 2 md5 xxxxxxx
 ip ospf cost 1000
!
interface eth1
 ip ospf cost 1000
!
interface lo
!
router ospf
 ospf router-id 10.10.32.37
! Important: ensure reference bandwidth is consistent across all routers
 auto-cost reference-bandwidth 10000
 network 10.10.32.0/22 [10.10.32.0] area 0.0.0.0
 network 10.19.90.11/32 [10.19.90.11] area 0.0.0.0
 network 192.168.188.249/32 [192.168.188.249] area 0.0.0.0
 network 192.168.188.1/32 [192.168.188.1] area 0.0.0.0
```

```
 network 192.168.188.4/32 [192.168.188.4] area 0.0.0.0
 area 0 filter-list prefix AREA_1_IN in
 area 0 filter-list prefix AREA_1_OUT out
!
ip prefix-list AREA_1_IN seq 5 deny any
ip prefix-list AREA_1_OUT seq 5 permit 10.19.90.11/32 [10.19.90.11]
ip prefix-list AREA_1_OUT seq 10 permit 192.168.188.249/32 [192.168.188.249]
ip prefix-list AREA_1_OUT seq 15 permit 192.168.188.1/32 [192.168.188.1]
ip prefix-list AREA_1_OUT seq 20 permit 192.168.188.4/32 [192.168.188.4]
ip prefix-list AREA_1_OUT seq 25 deny any
!
line vty
!
```

5.2 Traffic Shaping

Traffic shaping is the process of limiting the speed of certain data transfers on a network. Dummynet [18] can be used to obtain this functionality over FreeBSD. You have, however, to recompile a kernel in order to optimise the system. A pipe definition is used, and a bandwidth parameter is used to define the amount of data associated with each pipe. Bandwidths and queue management parameters should be optimised according to the traffic patterns identified.

In Linux you can also perform traffic shaping with TC HTB, however, this is a less flexible solution than FreeBSD.

5.3 Virtual Private Networks (VPNs)

An open source router can also be used as a VPN concentrator, by using appropriate software. OpenVPN [19] and SoftEther VPN [20] and FreeLAN [21] are just three free open source examples. The rollout of such a complex feature will need a deep analysis to see if the hardware and OS have suitable characteristics in order to deal with general traffic and at the same time deal with the encryption that this additional service implies.

IPSec [22] is also supported at the kernel level, both by FreeBSD and Linux. In that case, you also usually use raccoon to automatically key the IPSec connections.

5.4 Accounting

Performing accounting through Netflow, in which the router itself does summarisation work, should be the preferred option. Normally getting accounting information in Linux involves developing software to take the individual IP traffic from the iptables' tables; however that is CPU intensive, wasting CPU cycles while the recollection is happening.

The overall traffic of the router can be acquired via scripting ifconfig / ip command, via /proc tables or via SNMP.

5.5 Firewalling

There is a significant advantage in building firewalls over Linux or FreeBSD. In today's market it's unusual to find vendor-based firewalls based on bridging instead of IP. Deaggregating firewalling and NAT functions into distinct hardware may improve performance. A different option is to resort to CPU affinity (or CPU pinning), which can be used to limit the firewall's process to using only one CPU.

A FreeBSD-based firewall with Packet Filter and also running CARP [23] (Common Address Redundancy Protocol), or a Linux-based firewall with iptables, conntrackd [24] and Keepalived for VRRP, allow for redundant firewalling, with two or more boxes sharing state tables. If one of the nodes stops working, the other one(s) should assume all the workload.

5.6 Load Balancing

By using routing or load balancing with LVS [25] it is possible to create redundant solutions to scale or provide redundancy to other services, such as Web, VoIP or DNS servers. LVS services are provided by the kernel, and are a simple and very scalable solution operating at the OSI layer 4.

6 Conclusion

The option for using open source routers is not an easy one, as there are also some disadvantages. We must not expect an uneventful deployment from day one. Like any other networking architecture component, routers' behaviour needs to be closely monitored. The experience from any other in-house engineering activities should be useful to help plan for a smooth migration from a vendor-based platform.

The main goal of this document is to offer proof that such approach works for several organisations, with a clear message that serious in-depth analysis and hard work needs to be in place to enable this choice and allow for medium/long term gain. An organisation which decides to go this way needs to be ready to spend a significant amount of engineering hours, fine-tuning all its components. The hardware choice is probably the most crucial, as (open-source) software options are more likely to be replaced without any extra cost, if needed. The wrong choice of hardware has the potential to degrade network availability and usability, and it will certainly have a negative impact on the budget side later.

References

[1] Quagga Routing Suite
http://www.nongnu.org/quagga

[2] OpenBGPD
http://www.openbgpd.org

[3] BIRD Routing Daemon
http://bird.network.cz

[4] XORP Routing Suite
http://www.xorp.org

[5] Choosing a 10G NIC for x86 server
Ruairi Carroll, December 2013
https://wpneteng.wordpress.com/2013/12/03/on-choosing-a-10g-nic-for-intel-servers/

[6] Interrupts and IRQ tunning
RedHat Inc.
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/s-cpu-irq.html

[7] Linux Kernel Source Tree – ipv4: Delete routing cache
David Miller, RedHat Inc., 2012
http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=89aef8921bfbac22f00e04f8450f6e447db13e42

[8] Removing the Linux Routing Cache
David Miller, RedHat Inc., 2012
http://vger.kernel.org/~davem/columbia2012.pdf

[9] Traffic Shaping
http://www.hyperois.com/members/knowledgebase/1/-using-dummynet-for-traffic-shaping-on-freebsd.html

[10] FreeBSD – The power To Serve
http://www.freebsd.org

[11] Keepalived - Loadbalancing and High-Availability
http://www.keepalived.org

[12] Dynamic Routing Protocols for Campuses
Carlos Friaças; Pedro Ribeiro; Paulo Costa; Rui Ribeiro; Israel Lugo
http://services.geant.net/cbp/Knowledge_Base/Physical_Infrastructure/Documents/CBP-15_Dynamic-Routing-Protocols-for-Campuses.pdf

[13] NTOP
http://www.ntop.org/

[14] NTOP
http://gael.roualland.free.fr/ifstat/

[15] ETHTOOL – Utility for controlling network drivers and hardware
https://www.kernel.org/pub/software/network/ethtool/

[16] NetHogs
http://nethogs.sourceforge.net/

[17] Dtrace for Linux
https://github.com/dtrace4linux/linux

[18] Using Dummynet for Traffic Shaping on FreeBSD
https://forum.ivorde.com/ipsec-vpn-between-iphone-and-linux-freebsd-racoon-daemon-t16301.html

[19] OpenVPN
http://www.unixwiz.net/techtips/iguide-ipsec.html

[20] SoftEther VPN
http://www.softether.org/

[21] FreeLAN
http://www.freelan.org

[22] Steve Friedl's Unixwiz.net Tech Tips – An Illustrated Guide to IPsec
http://www.unixwiz.net/techtips/iguide-ipsec.html

[23] CARP Firewall Failover
https://calomel.org/pf_carp.html

[24] Netfilter Conntrackd
http://conntrack-tools.netfilter.org/conntrackd.html

[25] Linux Virtual Server
http://www.linuxvirtualserver.org

Glossary

ASN	Autonomous System Number
BGP	Border Gateway Protocol
CARP	Common Address Redundancy Protocol
CPU	Central Processing Unit
DDOS	Distributed Denial of Service (attack)
DNS	Domain Name System
DoS	Denial of Service
GBPS	Gigabits per second
GHz	GigaHertz
HTTP	HyperText Transfer Protocol
I/O	Input/Output
IPv6	Internet Protocol version 6
IST	Instituto Superior Técnico – tecnico.ulisboa.pt
NAT	Network Address Translation
NIC	Network Interface Card
NREN	National Research and Education Network
OS	Operating System
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PIMSM	Protocol Independent Multicast – Sparse Mode
R&E	Research and Education
RCTS	Rede Ciência Tecnologia e Sociedade
RIP	Routing Information Protocol
ULISBOA	University of Lisbon – ulisboa.pt
UTL	Technical University of Lisbon
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol