Open Source Routing in High-Speed Production Use

Robert Olsson, Hans Wassen, Emil Pedersen
Uppsala Universitet
P.O. Box 887
751 08 Uppsala
Sweden
Robert.Olsson@its.uu.se
Hans.Wassen@its.uu.se
Emil.Pedersen@its.uu.se

1 Abstract

In almost 10 years we have used Open Source Routers in mission critical networking to bring Internet connectivity to many tens of thousands users. Needless to say Uppsala University is one of the largest universities in Sweden and it is a well-connected university. It is currently using four Gigabit (including two for its student network UpUnet-S) connections towards our ISP (SUNET), and a production for 10G connection is planned. Uptime for our users is close 100% due to the testing and verification efforts and also due to the redundancy of the dual access. This makes it possible to, without loss of connectivity, replace and upgrade core routers. Reporting this success does not mean it is without effort or simple. It requires skill and planning, and the network managers must understand issues like packet budget and bandwidth needs, and be able to match them to the used equipment/routers. It is also important to understand traffic patterns and routing protocols, and of course how to operate and monitor the routers.

2 Introduction

Open Source networking is in fast progress due to worldwide use and contributions of numerous organizations. Companies, universities and governments are putting their effort into various software projects. Many governments and the European Union are supporting these efforts to increase knowledge and be a part of the development loop. This opens for new research and development areas to challenge the industry, universities etc to produce new products and services that ultimately results in increased efficiency and economic growth. In this paper we would like to bring our experiences in the area of open source routing, an area which have received an increased interest during the last years. We see an upcoming commercial interest where companies are packaging, tailoring and selling software and support just like for ordinary Linux distributions, but now with focus on infrastructure as networks and routing, firewall solutions etc. We also see a breakthrough in technical areas with high-speed buses, multicore processors and new interesting models of interface cards, which are beginning to include features to make use of multicore CPU's etc. Hardware classifiers on interface cards is bringing interesting challenges and possibilities which will impact the design of operating systems. Needless to say network functionality and performance is crucial for a modern operating system.

3 Our Open Source history in short

Experiences gained in an early 90:s project (the ATM-pilot) by Uppsala University (UU) in collaboration with the Swedish University of Agricultural Sciences (SLU), Telia, Chalmers and KTH (the Royal Institute of Technology) showed that good PC-based hardware and early days Linux OS was outperforming many commercial workstations with proprietary OS'es in network server performance. The encouraging server results inspired us to test for pure router performance, and the idea of an Open Source Router based on available PC-components was born. This being the mid 90:s, we faced lots of challenges trying to avoid movable parts such as a hard disk in order to increase MTBF. We found the first generation of flash cards to solve this and managed to make a Linux distribution small enough for and bootable from them. Chassis was another challenge as PC's was not in server room at this time. Anyway some 19 inch chassis were found and could even be equipped with redundant power supplies. So we were in the process of solving problems and gaining new experiences in a sort of successive approximation motivated by interest and curiosity. Testing and verification is a crucial part of the work, see the separate chapter. Finally in the mid 1999 we were ready for the real challenge, to verify and trust our work in the most demanding task. This was to handle all of Uppsala University's Internet traffic, which includes staff, researchers and some ten thousand of students, by connecting us to our ISP (SUNET) using our two LINUX PC-routers with full BGP peering. We used an improved and patched gated (from Merit Gated Consortium) routing daemon, BGP peering with SUNET and a default-route conditionally created and redistributed into OSPF. The routing was designed so that when one router was out of service all traffic was handled by the other router. We could of course also control the exterior gateway use with BGP metrics. This design is still in use.

4 Relation between Testing/Verification and Development/Research

Networking including routing are complex tasks. Regardless of what type of equipment is used, a serious network manager must understand, test and verify it. Specifications are wrong or missing or does not work as expected, there may be incompatibility problems etc. This is not only crucial for succesful deployment of the equipment, the testing loop will also bring skill and competence to the tester, which is very important when managing and troubleshooting the network. With Open Source we generally get much better possibilities to test as we can review, add own debugging, monitor and profile the used code. This gives a much better understanding and control. To start with you can change parameters or definitions in the code to match your own needs, and you can study or pinpoint some error or behavior. After a while you have some idea how to address and (in the end) solve the problem. The lesson we learned is that the step from advanced testing to development and research can be very small.

Lab testing is cheap, efficient and a good start to deeper understanding. It also offers reproducibility in a controlled environment. For router hardware and software tests this means injection of traffic and to study behavior and counters, code profiles etc. A successful testbench can be followed by deployment in less critical parts of the network, next to be proven in more critical parts. The development at Uppsala University have an unique advantage of the close cooperation with the daily operation in being able to test new features and ideas with no or minimal impact on users. Of course the results and experiences are shared with the network managers, and we gain as much indirect experiences as possible, meaning experience through others, including mailing lists and information available on Internet.

Lab testing might sound expensive and unachievable but for the matter of our lab it only consists or three PC's, so visitor expecting very expensive and exclusive equipment might be somewhat disappointed. One of the challenges we faced was to do innovative testing with the equipment available, our selected PC-hardware. We often had to create our own testing tools, for example pktgen [3] which is now used worldwide via the the Linux kernel for the benefit of many others.

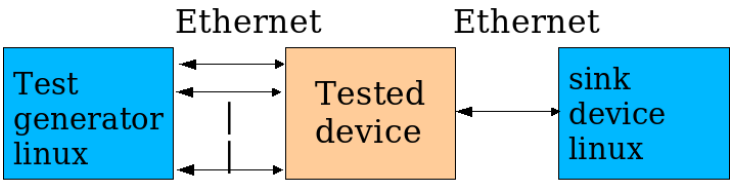

Figure 1: UU router lab setup

5 Production Use

Today Uppsala University uses Open Source Routing in three major installations, based on the same concept;

- Uppsala University Network
- Uppsala University Student Network
- The SUNET Archive

5.1 Uppsala University

The internally owned and administrated router network is glued together with OSPF and has many Cisco routers. There are external connections to our ISP (SUNET) and to a local DMZ, to share local traffic (see Figure 2). BGP is used for this traffic exchange with both ipv4 and ipv6, but only ipv4 is used over the local peering point. Protocol compatibility is important so Linux routers can coexist with proprietary routers like Juniper and Cisco

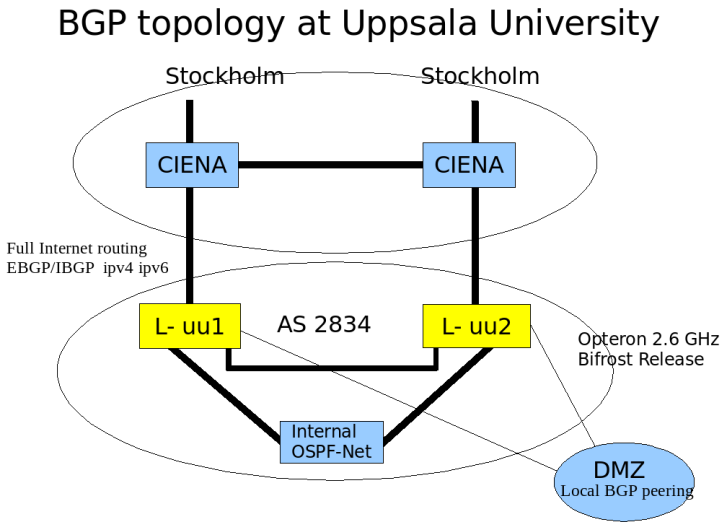

Figure 2: UpUnet BGP topology

5.2 Student Network

Uppsala university has a large number of students, around 30.000 in total of which about 12.000 are living on different student campuses. The university was among the pioneers to give Internet access to

the student campuses, it meant that their own infrastructure with fiber etc had to be installed. One crucial issue was not giving anonymous access or access to none-students, in order to conform with university and ISP polices.

In early 1998 a scheme for authenticated access to the network was worked out. Using Open Source and Linux it was easy to use an authentication service (TACAS in the beginning) to control the kernel netfilter rules and thereby to control user network access. To detect if a computer was shutdown or disconnected without logging off the logged on host was probed with arping, and missing replies meant the host should be logged off. Uppsala University and SLU provided a software package for this purpose which was used by many Swedish universities and many others. It has since evolved and is still in use today, making the student network a mostly Open Source based network.

As a result Linux was used in the network infrastructure, and these servers also served as routers to form the internal student network. The student traffic exceeds by far the ordinary university traffic and the network has recently been upgraded to have its own dual BGP peering with our ISP and local peering for university trafic. It is of course based on our own successful Linux/Bifrost [7] concept. The current network topologi is seen in Figure 3. Throughout the years there have been many problems to solve, such as DOS-attacks, scalability, fairness and bandwith issues but over the years the network has provided high availability and high bandwidth.

An excerpt from MRTG (5 min average) from one of the Uppsala University student border routers (reglus) 2008-04-13 is in Figure 4 & 5. We run virtually at Gigabit wire rate at both input and output with no or very few packet drops, and from rstat we see that about 300 kpps hits the warm cache and there is about 10 k new connection per second. It does full BGP against two peers as well as some local peering. This router runs Bifrost release 5.19 on hardware as specified in Appendix B

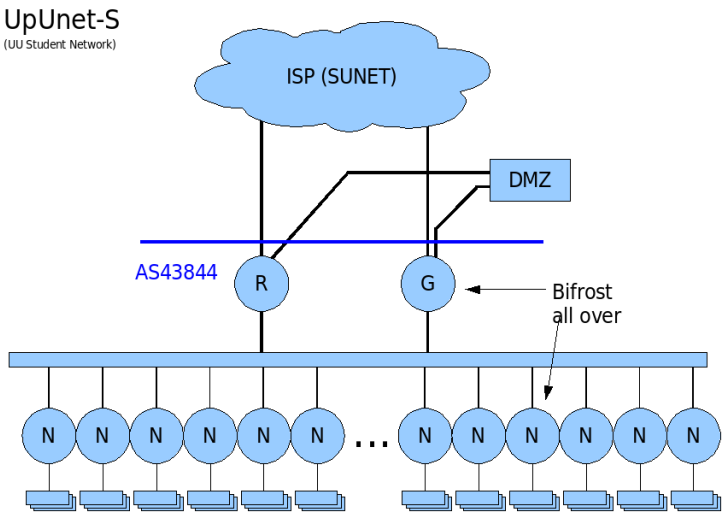

Figure 3: UpUnet-S topology

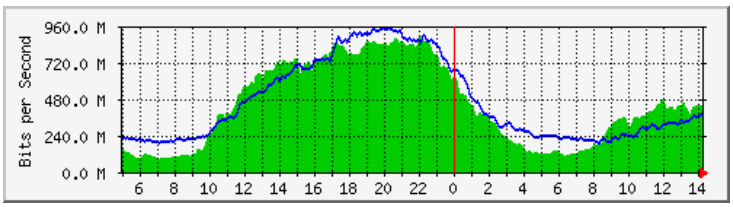

Figure 4: Reglus 24h BW

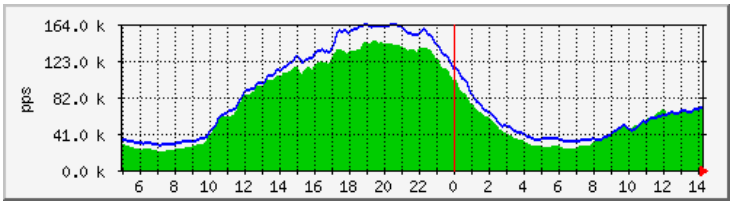

Figure 5: Reglus 24h pps

5.3 The SUNET Archive

The Sunet Archive has over the years been one of the major traffic sources in Sweden and Europe. It has from the very beginning been connected to Internet via SUNET with Linux Routers.

During many the years it has been one of the most popular and widely used achieves in the world. It started at SLU as a pioneering service in the early 1990's, but was later moved (in 2000) to Uppsala University.

The service today is dimensioned to handle more than 10 000 concurrent users and the network connection is specified to handle 2 Gbit/s via 2 GIGE interfaces. The archive has its own AS-number for clean network topology and for local data sharing over a dual access-point. The software used is Bifrost distribution and quagga for BGP supporting both IPv4 and IPv6

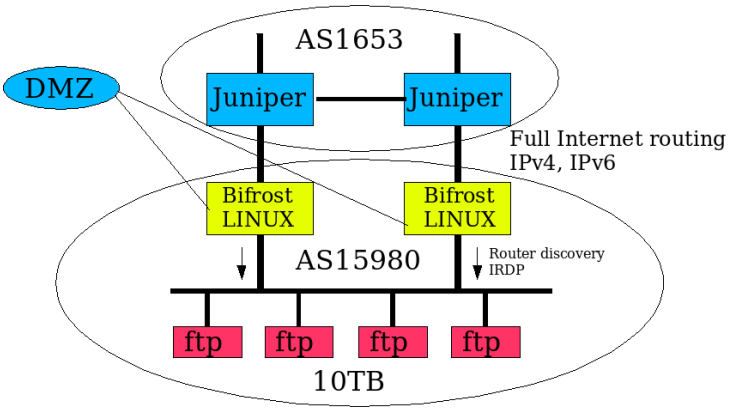

Figure 6: ftp.sunet.se network topology

5.3.1 IGMP Redundancy and Load Sharing

The archive itself is a server farm with IBM servers. The output traffic is load balanced and given redundancy in a very simple and straight- forward way – the Linux routers are injecting their default route via router discovery [8] to the server farm. The metric of the router discover message controls which of the outgoing routers should be used, and by using the same metric load sharing is achieved.

Throughout the years this has been a very convenient technique to operate this service, since router software or even hardware upgrades can been done without service interruption. (See Figure 6).

6 Software Selection

The platform used is our own Linux distribution Bifrost [7], which is distributed as a Unix tar archive and intended to boot and run Read-Only from an USB memory (or a Compact Flash using an IDE adapter).

It is very small and focused for networking purposes, and contains an improved, tuned and tested linux kernel where we try to bring our experiences.

To use it as a router we add our routing daemon of choise. Currently we use the Quagga Routing Software Suite (www.quagga.net) which is a GPL licensed IPv4/IPv6 routing software, but we test and make our own snapshots which we link statically. Fixes are, of course, submitted back to the developers, and over the years we have contributed with work in this area too. In the beginning gated was used and after that zebra/quagga. XORP is a new initiative (www.xorp.org), but we have no experience with it yet.

The Bifrost platform is also used as firewalls and for web based access control (Captive Portal) simply by choosing a different set of add-on packages.

7 Hardware Selection

Needless to say this a crucial process for stability and performance. All hardware must have good OS support via device drivers, so open documentation with specification of chipset etc is definitely an advantage. The selection process is very complex and time-consuming and in practical work this is integrated part of the testing and verification.

8 Open Source Router Performance

This a complex area and beyond the scope of this paper, but for mission critical and high performance routing understanding in this area is essential. Bandwidth is limited by hardware buses and interface cards while packet budget is usually limited by CPU power. The role of the CPU is to administer the DMA handling and to perform various lookups in cache, routing and other tables as well as any crypto handling if used. When dynamic routing is used the CPU also runs a routing daemon. CPU usage is small but provision must be taken so the routing daemon is given CPU time even at overload and Denial Of Service (DOS) attacks. The NAPI [5] was a work to improve this area. To give some idea about current performance we have included some unverified and unpublished test in Appendix A. It also show the variation in different setups and with different load.

9 Contributions (Areas of work)

During the years Uppsala University has been involved in collaboration, development/research and improvement of various parts of the network stack. This is more or less motivated by own needs. In the section below we mention areas were we have been involved and contributed to. See references for full list of authors. Also we should say that some work below was funded by SLU (the Swedish University of Agricultural Sciences) as one the authors has a background there.

NAPI NAPI [5] is an interrupt polling hybrid well integrated into the Linux kernel/softirq model. The network stack can perform at optimum and the system can behave well even if network load exceeds the system capacity, which is crucial for routers and high-end servers. Almost all high-speed network drivers uses NAPI today. This has recently been extended to support multiple queue by

Dave Miller, the current maintainer of Linux networking stack. Uppsala University had the first installation of NAPI in it's core as well as many of it forrunners (drivers based on HW-flowcontrol and tasklet-polling).

pktgen pktgen [3] is a module within the kernel for direct access to the device drivers sending routine (`hard_xmit()`), and for some type of tests the pktgen can even avoid memory allocations. This means pktgen can send packets at very high speeds and bandwidths and the reason why it's used for testing in many places.

routing stats Kernel routing cache statistics and the userland application rtstat was added to help monitoring and understanding of the linux routing cache. The routing cache is crucial for network performance and it is also very useful for monitoring network traffic characteristics and load. If for example we are under a DOS-attack we will see this using rtstat.

routing lookup fib_trie [4] is the implementation and rework of LC-trie [1] [2] for the Linux kernel. The algorithm is well-known and well-studied in the scientific literature. It builds a very efficient and flat search tree even for a very large number of routes. The implementation was given Intel Academic Award 2005

flow lookup TRASH [6] Is an effort to add unified lookup and improve stateful networking. Bifrost has an experimental TRASH-enabled kernel, which is in use in UpUnet-S (the UU student network). Here the destination cache is replaced with a TRASH data-structure which always does a full flow lookup. The major application now is flow logging without connection tracking, but more work is needed to explore the full capabilities.

routing daemon Some work on routing daemons has also been done. A first take of PIM-SM (Sparse-Mode) IPv4 multicast daemon was implemented with the zebra framework. Other contributions to zebra/quagga was the first code for zebra/MBGP. zebra/IRDP [8] was also implemented.

10 Acknowledgement

Many people have contributed over the years, especially we like to mention the crucial support during the very first years who's support and collaboration that has made this work possible;
Alexey Kuznetsov – a Linux networking guru and author of many parts of the Linux networking code and even some hacking on routing daemons and more.
Tom Johans, SLU – developed and managed the Bifrost distro from the very start.
Olof Welin-Berger – a former colleague and a network manager at SLU, open for new ideas and willing to take the risk to deploy Bifrost in his own critical infrastructure.

References

[1] Stefan Nilsson and Gunnar Karlsson. *IP routing with LC-trie* Selected Areas in Communications, 17(6):1083-1092, 1999.

[2] Stefan Nilsson and Matti Tikkanen. *An experimental study of dynamic tries* 1998. Submitted to Algorithmica.

[3] Robert Olsson. *pktgen the linux packet generator* Proceedings of the Linux Symposium, Ottawa, Canada, volume 2, pages 11-24, 2005.

[4] Robert Olsson, Jens Låås, and Hans Liss. *LC-trie implementation in linux* Linux v2.6.16.27, net/ipv4/fib trie.c.

[5] Jamal Hadi Salim, Robert Olsson, Alexey Kuznetsov. *Beyond Softnet, aka NAPI* USENIX, Oakland California, 2001, Nov 5-10

[6] Robert Olsson, Stefan Nilsson. *TRASH A dynamic LC-trie and hash data structure* IEEE, High Performance Switching and Routing. Brooklyn, NY, May 30 2007-June 1 2007. ISBN: 1-4244-1206-4

[7] More information and downloads at http://bifrost.slu.se/index.en.html

[8] Router Discovery http://www.ietf.org/rfc/rfc1256.txt

Appendix A

Unpublished and unverified numbers from bifrost release test. Note that the routing performance varies from 2 Mpps to about 200 kpps depending on load and configuration.

```
                    Linux router packet budget

                   Robert Olsson/Emil Pedersen
                       Uppsala Universitet

Preliminary version: 07115
Aggregated Routing Performance results in PPS (packets per second),
the total throughput is the sum of the number of packets hitting the
routing cache and number of packets missing it (Slow path):

                Flow load               Dual single flows       Test
Small rtable    900+140 = 1040          2010+0 = 2010           (1)
Full rtable     810+140 =  950          2020+0 = 2020           (2)
IPT loaded      680+140 =  820          1670+0 = 1670           (3)
IPT + conntr     77+144 =  211           730+0 =  730           (4)

* Results taken from rtstat handy but not so accurate.
* Concurrent load (in->out) eth0->eth1 and eth2->eth3
* CPU affinity CPU0(eth0, eth1), CPU3(eth2, eth3)
* Kernel version is 2.6.24rc1-git
* Flow load 2 * (4096 concurrent flowlen 10 pkts)
* Dual flow, same dst (one for eth0, eth2 resp).
* Hardware 2 * Dual Opteron 2220(2.8 GHz), MB TYAN 2915
* In total for 4 CPU cores, two used in test (by use of affinity).
* NIC. Intel e1000 2 * Dual NIC (82571EB) PCIe
* NAPI w/o link HW-flowctrl
* Conntrack size 16k.
* No IPT rules loaded at all.
* pktgen sends 64 byte UDP packets.
* Aggregated results in pps. Cache_hit + New_flow = PPS Total

* Full routing table here (test 2) is 214394 prefixes, but
  we always match same two prefixes. So these results should
  definitely be taken with a pinch of salt. Performance here
  is dependent how deep in the trie the prefixes are stored.
  You have been warned.

Modules loaded for test 3 (for basic netfilter usage):
  e1000, ip6table_filter, ip6_tables, xt_tcpudp, ipt_LOG,
  iptable_filter, ip_tables, x_tables

Modules loaded for test 4 (test 3 plus connection tracking):
  e1000, ip6table_filter, ip6_tables, xt_tcpudp, ipt_LOG,
  iptable_filter, ip_tables, x_tables, xt_state,
  ipt_REDIRECT, xt_MARK, iptable_nat, nf_nat, nf_conntrack_ipv4,
  nf_conntrack, iptable_mangle
```

Appendix B

Bifrost Hardware History

Pre-Era
AMD K2 @ 233Mhz

2000-11-03
Motherboard ASUS PII P2B / P3B-F 100 Mhz bus
19" Chassi, KI-P20WP
Budget chassies AOPEN HX-45 / HX-95
SanDisk 48 Mb Compact Flash with PCMCIA adapter
D-Link 4-port Tulip DFE-500TX (21143)
4-portars D-Link DFE-570TX
Ethernet-kort Netgear FA 310 TX
DEC Tulip chip:et. Rev C6.0 eller C6.1 is OK
Warning Newer Netgear does not use tulip
Watchdog-card Berkshire

2001-02-16
Motherboard ASUS CUBX FCPGA 100 Mhz bus
CPU 700 MHZ PIII Coppermine FCPGA

2001-08-31
Motherboard Supermicro 370DL3
Memory (ECC) 2x256MB
Chassies 19" P43X
CPU Pentium PIII 2 * 933/133 MHz
NIC GbE with Intel 82543GC Gigabit Controller

For some years
Motherboard Supermicro X5DL8-GG
2 * Processor Intel XEON 2.66 GHz

Next (date lost, probably 2005) ⇐ ***UU core routers***
TYAN 2882 AMD-8131/AMD-8111 Socket 940
2 * Opteron 252 2.6 GHz
NIC. GIGE Intel Dual NIC (82546EB) PCI-X

2008-01-03 ⇐ ***Archive & UpUnet-S core couters***
TYAN Thunder n5550W (S2915-E) NVIDA NPF3600/NPF-3050
256 MB Reg ECC PC3200 (400 MHz)
2 * Opteron dual core 2220 processor
19" 4U chassi
USB memory stick
Redundant Power Option

2008-09-05 ⇐ ***Next generation, in lab***
TYAN 2927GNR-E 1MB, NVIDA NFP3600
1 * Quad-Core Opteron.(Barcelona)
19" 2U or 4U chassies
NIC. GIGE Intel Dual NIC (82571EB) PCIe
NIC. 10g Intel Dual/Sinhle NIC (82598EB) PCIe
Option: Redundant power for 4U

Appendix C

Open Source Router History by Date at Uppsala University

```
981123  ftp.sunet.se at SLU traffic shaped to 28 Mb/s (using tbf qdisc
        and cron) during office hours otherwise full 34 Mbps

011206  Peering with ftp.sunet.se local DMZ
990318  L-uu1 SUNET-155 Mbps (2*155 Mbps)
000224  L-uu1 upgrade to 600Mhz CPU and chassimount.
010905  L-uu1 upgrade Supermicro MB och 2 X 1GHz PIII CPU's
020731  L-uu1 upgrade to GbE
020925  L-uu1 moved to GigaSUNET (2*1000 Mbps)
050918  L-uu1 now AMD Opteron.(Linux version 2.6.11.12_Bifrost)
070205  L-uu1 moved to OptoSUNET

071211  UpUnet-S moved from university network to their own router
        pair towards OptoSunet (with local peering for UU and others)
```