Application Server Availability; Measurement and Analysis of the NorNet Core

Sune Jakobsson
Department of Telematics
NTNU
Trondheim, Norway
e-mail: sune.jakobsson@telenor.com

Abstract— **This article investigates the availability of applications servers running on the NorNet Core test-bed. NorNet Core is the world's first, open, large-scale Internet test-bed for multi-homed systems and applications. Particularly, it is currently used for research on topics like multi-path transport and resilience. The NorNet Core test-bed provides access to worldwide distributed nodes, connected with multiple interfaces over a set of ISPs (Internet Service Providers), providing independent transport paths between them. Each node has a set of programmable nodes that can be used for network experiments. The objective of this paper is to describe a practical approach to assess how suitable this test-bed is for distributed computing, and application servers. In conclusion, the approach taken in this paper shows that by using a simple client to check all communication links, and then create plots of the data, one can observe changes over time, due to network failures, or changes to the network between the sites.**

Keywords- Test-bed; Java virtual machines; application servers; availability; tunnelling.

I. Introduction

This paper is an extended version of a conference paper by the author [1]. This paper addresses the behaviour and availability of distributed computing resources in a “virtual” network built on top of academic and commercial networks, like the NorNet test-bed [2]. The main goal is to use simple tools and mechanism already available, to assess the properties of the NorNet test-bed. It is often necessary either do this assessment, before the test-bed is taken into use, or run the assessment in parallel with the trials. The NorNet test-bed provides a multi-homed IP infrastructure with nodes distributed over the world. The conference paper [1], where it is demonstrated that simple monitoring, using HTTP requests and monitoring memory usage is sufficient to access the availability of a server. Using this technique over months of time, one gets a reasonable view of the availability for the measured system, including long-term memory leakages. A previous paper discusses the availability of web servers in commercial settings using providers (e.g., Amazon, Google and other providers) hosting the computing resources, that use the Internet as the transport network [3]. In such setting, you as a customer, have little or no control over the computing resources. In that case it is hard to assess to what extent the computing resources are shared or virtualized, but one can assume that the Internet itself is a reasonably stable platform for transport. However, as a customer of the cloud computing resources, one has little or no control of the instance of deployment, or how the instance is configured, and how well it is monitored. This is due to the fact that commercial providers do not disclose any or very little information regarding their internal infrastructure, keeping this as company secrets. In the NorNet Core setting, one has near complete control over and information about the computing resources, but limited control over the point-to-point tunnels running between the sites. For additional publications regarding the NorNet Core, please visit [4].

The objective of this paper is to assess the behaviour and availability of servers serving Web pages and running in the NorNet Core network and the transport of IP packets between the sites. It describes a series of simple experiments at the application level, i.e., invocation of Web servers and how to capture their continuous operation and long term behaviour.

In Section II, we describe the NorNet test-bed in detail and how the experiments were carried out. The goal of these measurements was to detect network changes and service quality in the test-bed over periods of days, weeks and months, due to issues that can be traced back to the inter-site communication links or the software (SW) running at the sites and how these issues affect application servers running on the test-bed. The NorNet Core test-bed was continually updated and upgraded and the packet route the tunnels use was entirely up to the ISP, so there was a number of factors that impacted the availability.

Section III addresses the details of the monitoring, and what tools were used to collect the data, and validate the measurements.

Section IV provides some highlights of selected subsets of the results, and discusses the visualisation used in the plots.

Finally, Section V discusses the shortcomings of a best effort worldwide distributed test-bed and provides recommendations for test-beds in general and areas for improvement of the NorNet Core.

II. The NorNet Core test-bed

A. *Test-bed structure*

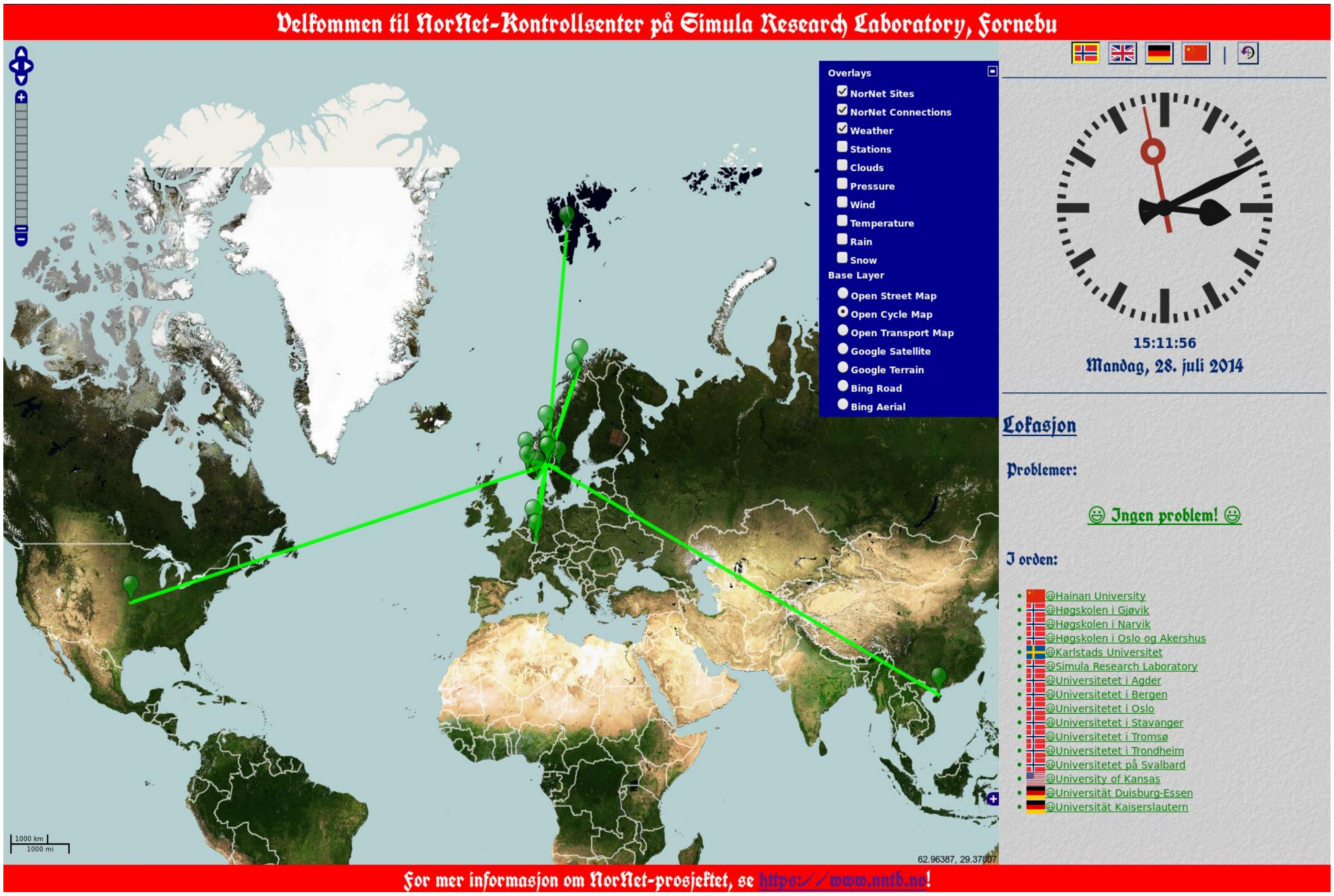
As of writing May 2016, the NorNet Core test-bed [2] is deployed on 20 sites physically distributed across the world and interconnected with tunnels over 15 different ISP’s networks. The majority of the sites are at the major universities in Norway, at Simula Research Laboratory in

Oslo and the rest are at universities in Sweden, Germany, China, Korea Australia and USA. The 16 ISP's provide connectivity across the sites, so that roughly half of the sites are connected using tunnels with more than one ISP involved. As an example, the University in Trondheim, is connected to Simula Research Laboratory by Uninett and PowerTech, and hence provides two independent communication tunnels with their respective interfaces, hence providing a true multi-homing configuration.

A screen shot of the testbed console is shown in Figure 1. The colour of the links between the sites, indicate their status, and a similarly the pins on the sites indicate the site status. Green indicates that the site or link is available. The site names and their ISPs are shown in Table I. We do not know how the ISPs route the packets between the sites, but we can observe when the underlying packet routing changes over time.

Each site has a set of research systems running virtual machines for experiments, a control-box for management functions, and a tunnel-box that terminates the tunnel end-points between the sites. The tunnel-box terminates all the tunnels at a site, and also provides the assignment of the local IP addresses, which allows an experimenter his own IP range. The NorNet Core runs its own domain name service (DNS), and the tunnels provide both IPv4 and IPv6 connectivity between the sites. The tunnels provide site to site connectivity over academic and commercial IP networks. The overall structure of a node in the NorNet Core is illustrated in Figure 2.

The thin red line from the control box is the connection to the central management system in Oslo. The primary ISP is used for the configuration from the central Simula site.



Velkommen til NorNet-Kontrollsenter på Simula Research Laboratory, Fornebu

Overlays
☑ NorNet Sites
☑ NorNet Connections
☑ Weather
□ Stations
□ Clouds
□ Pressure
□ Wind
□ Temperature
□ Rain
□ Snow

Base Layer
○ Open Street Map
● Open Cycle Map
○ Open Transport Map
○ Google Satellite
○ Google Terrain
○ Bing Road
○ Bing Aerial

15:11:56
Mandag, 28. juli 2014

Lokasjon

Problemer:

😃 Ingen problem! 😃

I orden:

- 😃Hainan University
- 😃Høgskolen i Gjøvik
- 😃Høgskolen i Narvik
- 😃Høgskolen i Oslo og Akershus
- 😃Karlstads Universitet
- 😃Simula Research Laboratory
- 😃Universitetet i Agder
- 😃Universitetet i Bergen
- 😃Universitetet i Oslo
- 😃Universitetet i Stavanger
- 😃Universitetet i Tromsø
- 😃Universitetet i Trondheim
- 😃Universitetet på Svalbard
- 😃University of Kansas
- 😃Universität Duisburg-Essen
- 😃Universität Kaiserslautern

For mer informasjon om NorNet-prosjektet, se https://www.nntb.no!

Figure 1. Screen shot of Control center for NorNet Core servers.

Table I. Locations of NorNet Core server and their respective ISP's

Index	Site	ISP1	ISP2	ISP3	ISP4
1	Simula Research Laboratory	Uninett	Kvantel	Telenor	PowerTech
2	Universitetet i Oslo	Uninett	Broadnet	PowerTech	-
3	Høyskolen i Gjøvik	Uninett	PowerTech	-	-
4	Universitetet i Tromsø	Uninett	Telenor	PowerTech	-
5	Universitetet i Stavanger	Uninett	Altibox	PowerTech	-
6	Universitetet i Bergen	Uninett	BKK	-	-
7	Universitetet i Agder	Uninett	PowerTech	-	-
8	Universitetet på Svalbard	Uninett	Telenor	-	-
9	Universitetet i Trondheim	Uninett	PowerTech	-	-
10	Høgskolen i Narvik	Uninett	Broadnet	PowerTech	-
11	Høgskolen i Oslo og Akershus	Uninett	-	-	-
30	Karlstad Universitet	SUNET	-	-	-
40	Universität Kaiserlauten	DFN	-	-	-
41	Hochschule Hamburg	DFN	-	-	-
42	Universität Duisenburg-Essen	DFN	Versatel	-	-
43	Universität Darmstadt	DFN		-	-
88	Hainan University	CERNET	CnUnicom	-	-
100	Univercitade Federal de São Carlos	RNP	-	-	-
160	Korea University	KREONET	-	-	-
200	National ICT Australia	AARNet	-	-	-

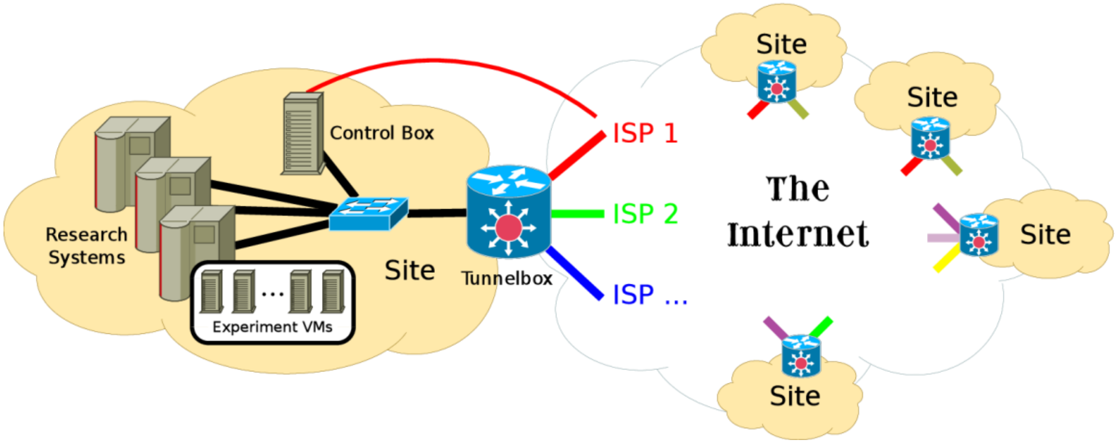


Figure 2. Overall structure of a single NorNet Core site.

Each site contains a set of physical servers that host individual virtual machines running instances of Planet lab software [5] for managing the sites. The virtual machines (VM) run the Fedora version 18 operating system [6], and connect to all available VPN tunnels at the site through the tunnel-box, and researchers can use them for multi-homed experiments as needed. Each site contains a number of VM instances, and they are all connected to all ISP's at the site. The experimenters are free to install SW on the VM's as needed. These VM instances are referred as slivers in the NorNet terminology. Please note that the term sliver in this paper refers to a running VM at a site. The term is also used by Fedora, but with a different meaning in their setting. The test-bed is configured so that the users get global access to all nodes, and they are able to do experiments on each node as needed, by accessing each virtual machine on an instance by instance basis. This allows individual users to get assigned VM's with private IP addresses, and do not need to consider sharing network interfaces with other users. There are some restrictions on what access rights a user is assigned to the operating systems on each site, and access to all operating system instances is done through the central site at Simula Research Laboratory in Oslo, Norway. These restrictions include tunnel configuration, and the underlying management of the research systems at a site. The entire NorNet Core infrastructure is managed from Simula Research Laboratory and their technical staff in Oslo, Norway.

B. The mesurement setup

The NorNet Core test-bed at Simula Research Laboratory has kindly provided an additional central node in Oslo for measurement purposes, which has direct access to the network interface at the site, and is able to do packet capture on the wire, so that the behaviour of the network can be captured and studied in retrospect. As shown in Figure 3, when site A invokes the Web server at site B, using all three networks in between, this traffic is captured by the measurement node at Simula Research Laboratory. This means that each server has a dedicated interface for each of the tunnels, and is able to send and receive IP traffic independently on each ISP. This implies that each site has a separate IP address for each tunnel, and one can therefore us IP addresses to select which tunnel to use.

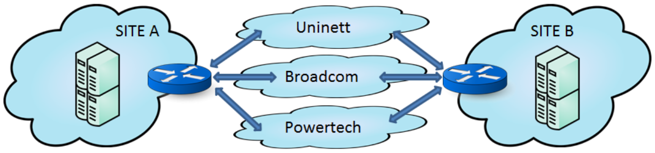


Figure 3. Tunnels between two sites

The measurement server at the Simula site is able to record the entire communication between the Simula site and any arbitrary chosen site, allowing investigation at the packet level of IP. The detailed measurements are done on HTTP calls issued from the Simula site in Oslo, where the calls are issued at fixed intervals to a selected set of sites, making calls over all tunnels, and thereby using the infrastructure provided by all involved ISP's. Since this is also the node that manages all other tunnels and nodes, and has other usages within Simula Research Laboratory, it is fair to assume that any operational issues are observed and rectified within reasonable time. This central measurement node runs the Ubuntu operating system and is directly connected to four ISP's. The measurement node is hosted in an Experiment VM, as shown in Figure 2, with access to the site traffic. The HTTP calls from the measurement node are issued by a shell script using the *curl* command [7] and the *crontab* [8] mechanism to schedule the commands every minute, and the results are captured in a log file, by the Web server (Jetty) as shown in Figure 4. The results from two different days without down-time and invocation errors on a particular ISP (Broadnet) and site (University in Oslo) are shown in Figure 6 and Figure 7, where the status (200 OK) is shown in a horizontal pink line and the black lines are the T_{dns} (DNS lookup times), the blue lines are the T_c (connection setup times), and the green lines are the T_t (total invocation time). Note that in most invocations the black DNS lookup time is so small that it is nearly invisible, and hidden at the bottom. However, some of the invocations might exceed the acceptable time constraints, but this depends on the actual application, and its requirements. The amount of free memory is shown in bytes, and highlights the staircase pattern, that is typical for reoccurring garbage collection. The regular pattern is due to the fixed interval of the invocations, which allocate the same amount of buffers for each call.

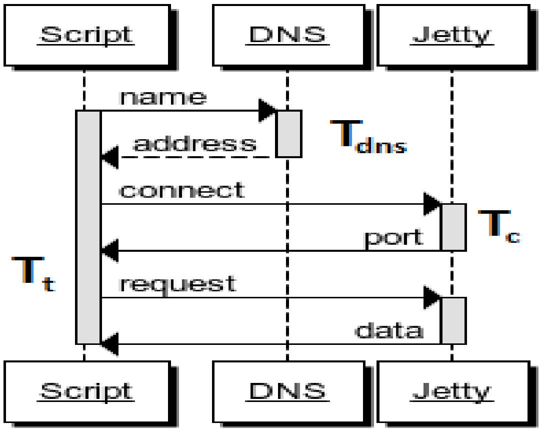


Figure 4. Invocation sequence diagram for the measurement script.

C. Experiment VM at sites

For the availability experiment, an instance of an Embedded Jetty Server [9] runs as a Web Server and listens to HTTP requests on all interfaces, connected to the individual ISP tunnels at all participating sites. The HTTP requests are issued with the `curl` command, and scheduled with `crontab`. The invocations are scheduled at one minute intervals, and each ISP tunnel is used in 1440 measurements per day. The number "1440" comes from the number of minutes in a day, and is also the shortest interval one can define in "crontab". The measurement period is over several months, so using a granularity of one minute invocations, is sufficient to detect irregularities in the long run. The measurements are also considered to be independent events. Since the network is virtualized and each sliver has its own unique IP addresses on each of the ISP tunnels, the HTTP requests are tagged with time-stamps and also logged locally on each Web server. The Web Server on each site logs locally the incoming request and their unique invocation tags, and responds with a short response containing the amount of available memory on its Java instance. The triplet of IP source and destination addresses and time-stamp provides a unique identifier in the local logs. This also eases the identification of the packets captured on the wire between servers and clients, and makes it possible to observe the network behaviour at the packet level, with the packet sniffing tools.

III. Monitoring of the test-bed

There are multiple issues that one wants to observe in order to determine the stability of a test-bed. One is the nodes themselves, their ability to communicate, and what changes over time are observable. Given that the test-bed should be able to run Internet scale experiments, observing them from an application point of view will give a real life view of its abilities. The physical layout of the NorNet Core needs a set of experiments carried out in parallel in order to pinpoint possible performance or availability issues that can occur, whether they occur on a single sliver, on an entire site, or on the NorNet Core as a whole.

The measuring node runs two distinct sets of measurements in parallel:

1: The first set of measurements runs towards physically distributed nodes, to detect communication issues between the sites, the tunnel SW and configuration issues towards the other sites.

2: The other set of measurements runs towards the slivers residing on one physical location, to detect local issues and if the slivers are communication, and in good health. The reasoning behind this is to be able to detect internal issues on a node, i.e., if there are issues that can be traced back to the tunnel-box and the installation at that site vs. general operation issues in the test-bed as a whole. Since all requests are issued on all ISP's available for transport at that particular site, one can determine if an issue is related to a site or to transport.

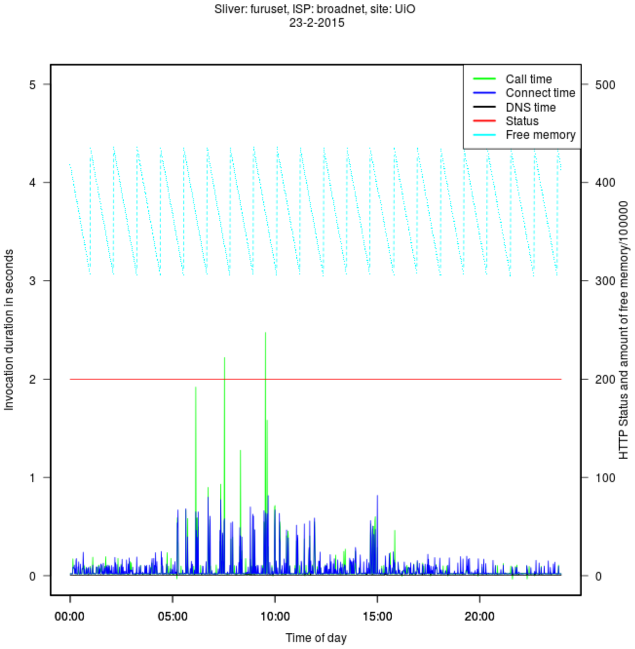


Figure 5. Invocation times and status for 24 hours

Each tunnel on each ISP can then be plotted every day as a graph shown in Figure 5 or Figure 6, to give a visual overview of the behaviour from day to day.

.

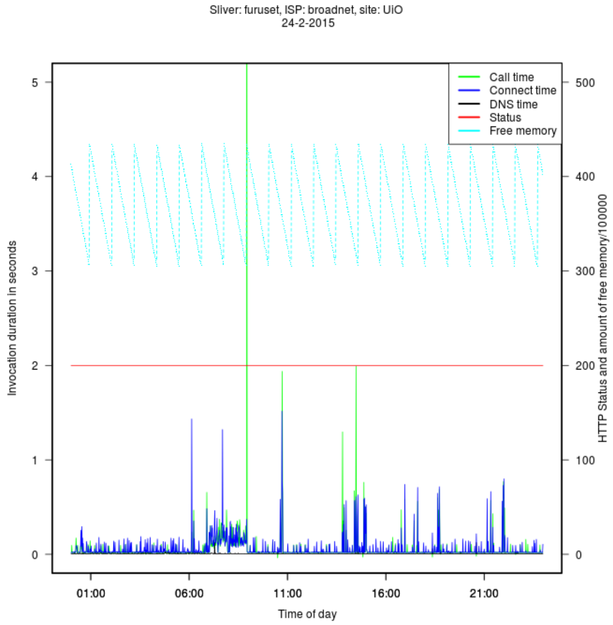


Figure 6. Invocation times and status for 24 hour

The "`curl`" command gives the connection time, DNS lookup time and connection time for each invocation. In addition each invocation is tagged with a time-stamp so that it is possible to explore the network behaviour at the packet level, using tools like "`Wireshark`" [10] and to do post-mortem inspection of unusual or odd behaviour in the HTTP communication between the sites. Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. For the users of the testbed the packet capture is only available at the Simula site. In addition the local logs are available at each node that is invoked.

With the redundant transport between the nodes it is easy to determine the overall condition of an individual tunnel and an individual sliver between the measuring node and the sliver. By automating the plot generation, daily plots are easily generated like the ones shown in Figure 5 and Figure 6. This, however, results in great numbers of plots and checking them all for abnormalities can be a daunting task. By enforcing a limit on the invocation time T_t on tunnels or sites, when this limit is exceeded, issues are easily identified and can then be inspected further. By visually comparing plots between different physical sites, it is straight forward to identify global issues or particular issues only manifesting themselves at one site or on one single tunnel (ISP) providing connectivity to that site.

It is also desirable to assess the network characteristics of the tunnels on a daily basis by a statistical analysis. Since the tunnels are tunnelled over Internet or some local transport, their characteristics varies over time. The connection times T_c for a particular tunnel (Broadnet) and site pair (University in Oslo) for two days, are shown as a density plot in Figure 7 or as a visual plot of an empirical cumulative distribution as shown in Figure 8. Given the shape of the distributions and the number of samples per day, the Kolmogorov-Smirnov test [11] is chosen to be the most suitable test to compare the daily connection time data.

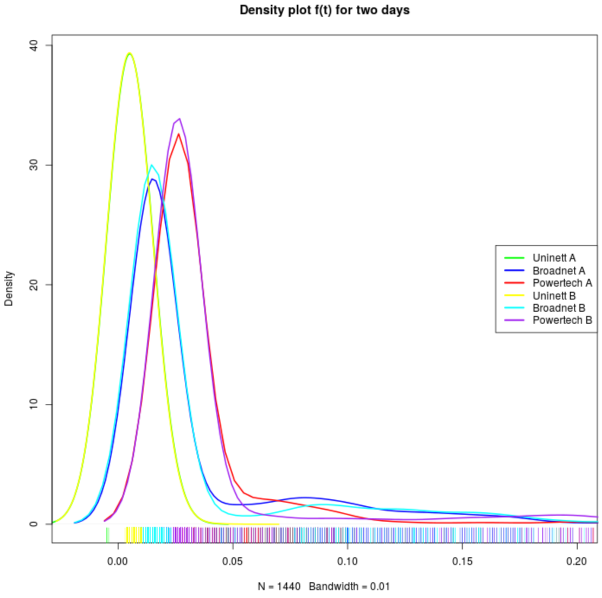


Figure 7. Density plot of connection times.

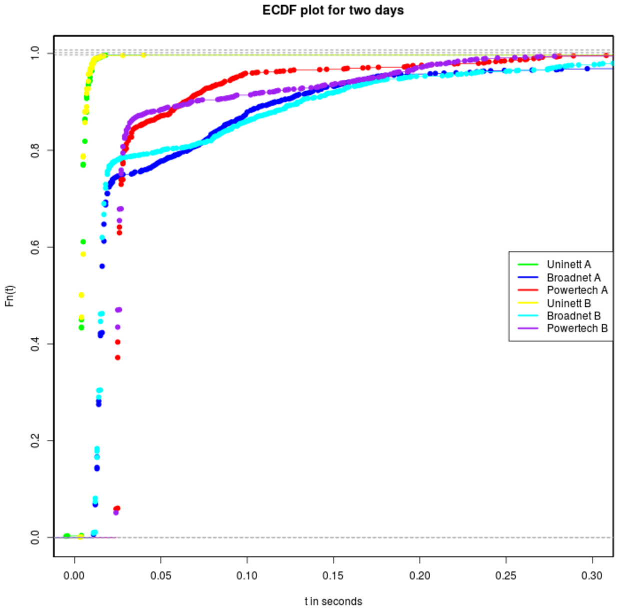


Figure 8. Empirical Cumulative Distribution Function.

The daily connection time distribution can be determined for each tunnel and sliver pair and the result gives an indication if there are changes in the communication between the measuring node and a particular sliver. Uninett is the ISP for the universities in Norway, whereas Telenor, PowerTech, Kvantel, Altibox, BKK and Broadnet are commercial ISP providers in Norway.

IV. Results

By assessing daily measurements first at HTTP invocation level, and by defining an acceptable maximum invocation time, depending on the application and the usage, and comparing connection-time distributions and slivers memory usage patterns, enables detection of changes in all involved parts of the test-bed. For a typical Web site used by a human one would expect a maximum tolerable page loading time of 5 second. After that a user would either abandon the site or try to reload the Web page in order to get a response, and hence reissue the original request. We therefore consider invocations that take more than 5 seconds as lost or missing, even if the request would succeed at a later point. By overlying the daily HTTP invocation and status plots one can identify “global” issues affecting the entire test-bed, and as well as “local” issues affecting one site, or one ISP tunnel between the measuring node and the site or sliver. By “global” issues we mean events that impact the entire NorNet Core network, like the DNS or the management functions, whereas “local” issues are issues that affect only one site. As shown in the sub-plots for each ISP in Figure 9. They show the maximum connection time for each ISP, spanning over several weeks, and also the average connection time as a black line. We have a period where the entire test-bed was down, shown as lack of data, and also show the long-term average difference between the individual tunnels provided by the individual ISPs, indicating a “global” issue. The long term connection time also changes significantly from the time before the entire test-bed was upgraded to afterwards, indicating that the new test-bed SW impacts the connection time in a negative fashion. There are also some observable similarities between two of the ISPs (PowerTech and Broadcom) that could indicate that their respective tunnels are indeed sharing some IP transport links between the physical sites. Whereas in Figure 7 between time 7:00 and 9:00, there is a two hour period, where the packets on that particular ISP used on average an extra 100 milliseconds in transport. This indicates a rerouting issue for that particular ISP. Even though there are variations the connection-time the distribution is stable, unless there is a change in the IP packet route or a change in the tunnel-box SW. By viewing the daily density plots of two consecutive days (Figure 7) or daily empirical cumulative distribution functions (ECDP) (Figure 8) on tunnel and sliver pairs, the tunnels repeat the same plots, unless there is some underlying issue. Day one in the plot is referred to as A, and day two as B. In addition the Kolmogorov-Smirnov tests have been run to show that the days without changes or downtime give the same distribution. It turns out that the visual presentation of two consecutive days is better in the

ECDP plot, than in the density plot, when there are substantial differences between two days.

In addition the memory usage on the slivers has also been monitored, and the slivers do not appear to be disturbed by other processes on each sliver. They all show a regular pattern in the amount of free available memory for the virtual machine running the Web server, and are hence not disturbed or affected by external factors, as shown in Figure 5 and Figure 6.

The Web Server SW and the scripts used to run the experiments are all available at github [12].

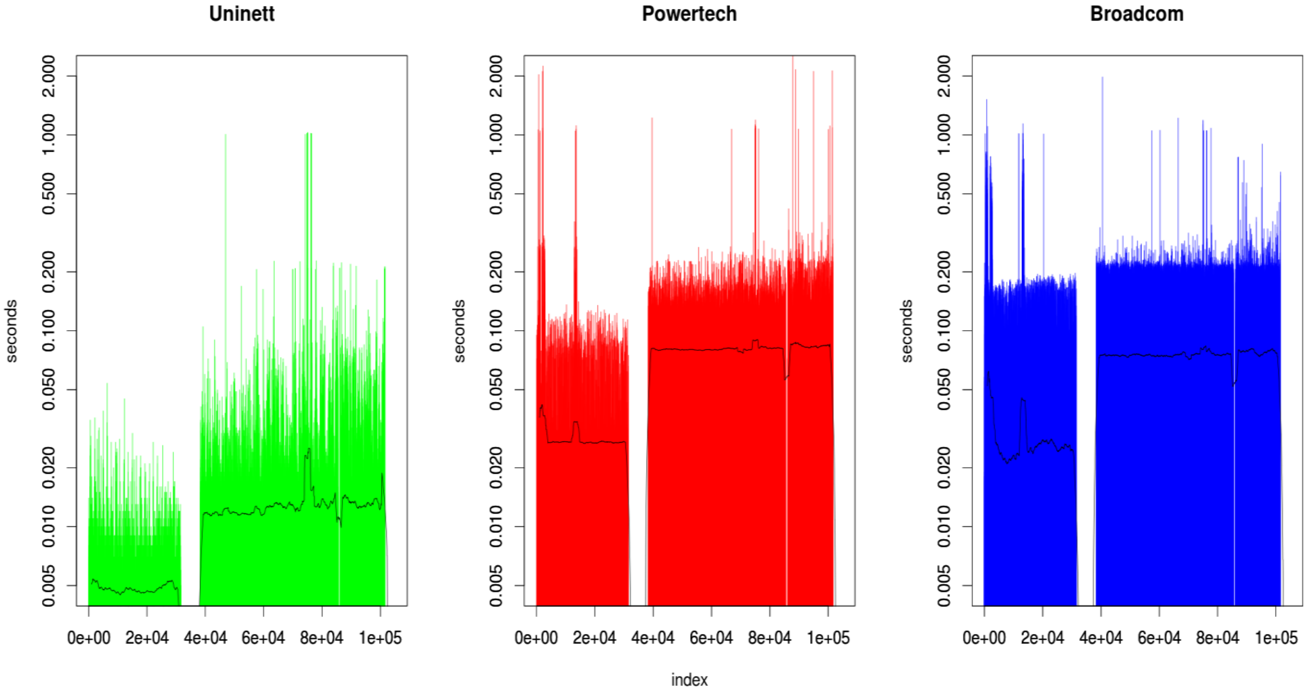


Figure 9. Longterm connection time performance (seconds) over several weeks.

V. Recommendation

The NorNet test-bed provides a multi-homed environment for large scale Internet experiments, but it has unfortunately focused the technical aspects of such a test-bed, and primarily at the transport level between the slivers. Most of the experiments published address multi-home transport and their protocols, and are not addressing Internet style client-server usage [13].

The physical distribution of sites adds some transport time between them, and occasionally the routing changes between sites add a constant to the transport time.

The NorNet Core test-bed provides monitoring tools with graphical interfaces. The monitoring provides a colour coded overview of the sites and the links, but does not address the individual slivers at the site. However, this does not give a detailed picture of the communication between the sites nor the status or quality of the tunnels between the sites. When a site goes off-line there is limited support for bringing the site back on-line other than contacting the personnel at the site. This has some grave implications on availability if parts of the test-bed run into issues or go down outside office hours or vacation times. Being a research network NorNet Core does not provide a service level agreement (SLA) for their users, so you do not get your money back when there are failures [14].

The lack of 24/7 operation management, it is hard to plan and carry out long term experiments. It is necessary with more than only best effort guarantees on a test-bed, to provide repeatable experiments, without adding the operational uncertainties.

The NorNet Core should add some rudimental monitoring SW for each node and each tunnel, and provide this information on the NorNet Core web page. This information could also be used internally at Simula Research Laboratory to alert the personnel in charge to quicker respond to failures or errors that are bound to happen at some point. A SLA for the users of NorNet Core could be beneficial for all parties involved.

VI. Conclusions

The approach described in the paper shows that simple monitoring at the application level is sufficient to pinpoint issues connected to tunnels, site SW and operation of the test-bed as a whole, by only collecting the invocation times, and amount of memory on the servers at the sites. This has been accomplished by invoking simple HTTP services between the involved sites, and collecting data about call setup, response times and memory usage. Using graphics enables one to quickly get an overview, and to pinpoint irregularities. One can then zoom down to interesting abnormalities, to post analyse the root cause of the issue. However, to fully automate the process and to use it to predict future behaviour is still work in progress. The lack of 24/7 operational management limits the possibility to run stable long terms experiments, and should be addressed for the success of the NorNet Core test-bed.

ACKNOWLEDGMENT

I would like to thank Professor Rolv Bræk and Professor Bjarne Helvik at Department of Telecommunication at NTNU, for their advice and guidance in my research work, and my wife for proof reading my papers.

REFERENCES

[1] Sune Jakobsson, “Estimation of Performance and Availability of Cloud Application Servers through External Clients”, in DEPEND 2013, 6th International Conference on Dependability, pp. 1-5, ISBN: 978-1-61208-301-8

[2] NorNet Core, https://www.nntb.no/nornet-core, Last seen May 2016

[3] Sune Jakobsson, “Measuring Application Server Availability on the NorNet Core” in DEPEND 2015, 8th International Conference on Dependability, pp. 1-4, ISBN: 978-1-61208-429-9

[4] NorNet publications, https://www.nntb.no/publications, Last seen January 2016

[5] Planet lab, https://www.planet-lab.org, Last seen January 2016

[6] Fedora 18 (Spherical Cow), https://docs.fedoraproject.org/en-US/Fedora/18/html/Release_Notes/index.html, Last seen January 2016

[7] Curl, tool, http://curl.haxx.se/docs/manual.html, Last seen January 2016

[8] Crontab, tool, http://www.unix.com/manpage/linux/5/crontab, Last seen January 2016

[9] Jetty, application server, http://eclipse.org/jetty, Last seen January 2016

[10] Wireshark, tool, https://www.wireshark.org, Last seen January 2016

[11] Vito Ricci, “Fitting distributions with R”, http://cran.r-project.org/doc/contrib/Ricci-distributions-en.pdf , Last seen January 2016

[12] GitHub, code base, https://github.com/sunejak/EmbeddedJetty, Last seen January 2016

[13] Thomas Dreibholz, Jarle Bjørgeengen, and Jonas Werme: “Monitoring and Maintaining the Infrastructure of the NorNet Testbed for Multi-Homed Systems”, in 5th International Workshop on Protocols and Applications with Multi-Homing Support (PAMS) 2015, pp. 611–616, ISBN 978-1-4799-1775-4, DOI 10.1109/WAINA.2015.76

[14] Brian Harry, “How do you measure quality of a service?”, http://blogs.msdn.com/b/bharry/archive/2013/10/14/how-do-you-measure-quality-of-a-service.aspx, Last seen January 2016