Measuring Application Server Availability on the NorNet Core

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

***Abstract*— This paper 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. This paper describes a practical approach to assess how suitable this test-bed is for distributed computing, and application servers.**

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

I. INTRODUCTION

This paper addresses the behaviour and availability of distributed computing resources in a "virtual" network built on top of academic and commercial networks, afterwards referred as the NorNet test-bed [13]. In a previous paper discussing 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 [12]. In such setting, you as a customer have little or no control over the computing resources. 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 consumer of the computing resources one has little or no control of the instance of deployment. By this we mean that the commercial providers do not disclose any or very little information regarding their infrastructure. In the NorNet Core case, one has near complete control and information over the computing resources but limited control over the point-to-point tunnels running between the sites.

The objective here is to assess the behaviour and availability of web servers running in the NorNet Core network and the transport of packets between sites. It describes a series of simple experiments at 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 infrastructure in detail and how the experiments were carried out. The goal of these measurements was to detect changes in the test-bed over periods of days or weeks, due to issues that can be traced back to the communication 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 Section IV highlights a subset of the results. Section V provides recommendations.

II. THE NORNET CORE TEST-BED

A. Test-bed structure

As of writing March 2015, the NorNet Core test-bed [1] is deployed on 19 sites physically distributed across the world and interconnected with tunnels over 14 different ISP's networks. The majority of the sites are at the major universities in Norway, at Simula A/S in Oslo and the rest are at universities in Sweden, Germany, China, Korea and USA. The 14 ISP's provide connectivity across the sites, so that the majority of the sites are connected using tunnels with more than 1 ISP involved.

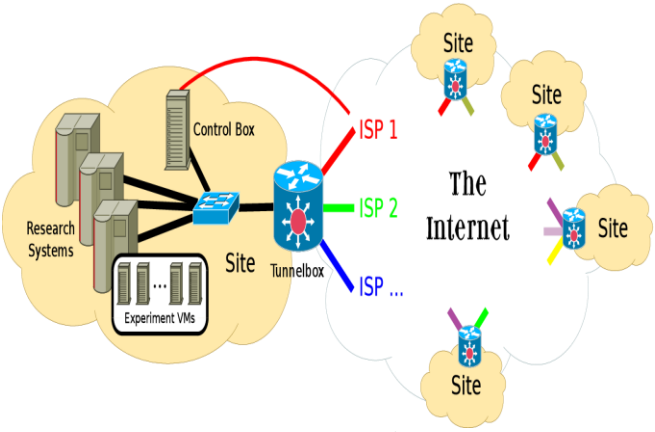

Figure 1. Overall structure of Nornet Core

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 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 the NorNet Core is illustrated in Figure 1.
The red line from the control box is the connection to the central management system in Oslo.

Each site contains a set of physical servers that host individual virtual machines running instances of Planet lab software [3] for managing the sites. The virtual machines

 ISBN: 978-1-61208-429-9

(VM) run the Fedora version 18 operating system [4], 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 A/S 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.

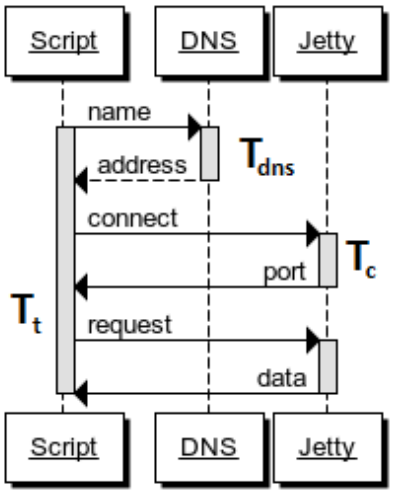

Figure 2. Invocation time sequence for the measurement script

B. The measurement setup

The NorNet Core test-bed at Simula A/S, has kindly provided a central node in Oslo for measurement purposes, which has direct access to the network interface, and is able to do packet capture on the wire, so that the behaviour of the network can be captured and studied in retrospect. The measurements are done on HTTP calls issued from the central node in Oslo, where the calls are issued at fixed intervals to a select set of sites, using 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 A/S, it is fair to assume that any operational issues are observed and rectifier within reasonable time. This central measurement node runs the Ubuntu operating system and is directly connected to four ISP's. The HTTP calls from the measurement node are issued in a shell script using the `curl` command [7] and `crontab` [8] to schedule the commands every minute, and the results are captured in a log file, as shown if Figure 2. The results from a day without down-time and invocation errors on a particular ISP and site are shown in Figure 3, where the status (200 OK) is shown in a horizontal pink line and the black lines are the DNS lookup times, the blue lines are the connection setup times, and the green lines are the total invocation time. However some of the invocations might exceed the time constraints, but this depends on the actual application, and its requirements. The time shown in green shows all the time elements, DNS lookup, connection time and data transfer time added together.

C. Experiment VM at sites

For the availability experiment, an instance of an Embedded Jetty Server [5] runs and listening to HTTP requests on all interfaces. 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 runs 1440 measurements per day. Since the network is virtualized each sliver has its own IP addresses on each of the ISP tunnels. The HTTP requests are tagged with time-stamps and also logged locally on each Web server.

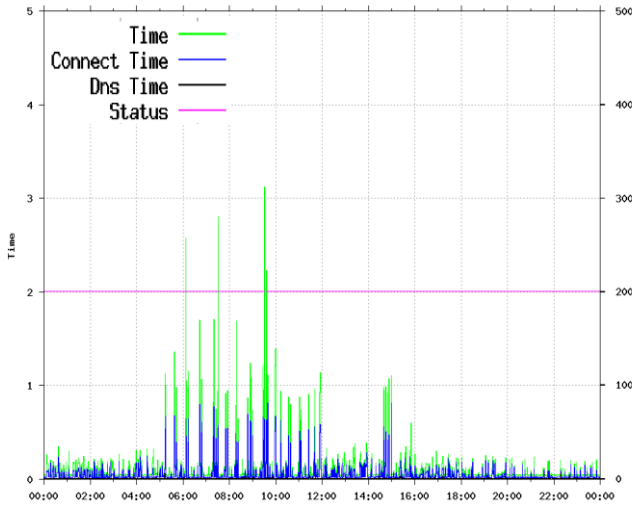

Figure 3. Invocation times (seconds) and status for 24 hours

The Web Server 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.

 ISBN: 978-1-61208-429-9

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 picture of its abilities. The setup of the NorNet Core needs a set of experiments carried out in parallel in order to pinpoint possible issues that can occur, whether they occur on a single sliver, on an entire site, or on NorNet Core as a whole.

The measuring node runs two distinct sets of measurements in parallel where the first set runs towards physically distributed nodes and the other set runs towards the slivers residing on one physical location. 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. Each tunnel on each ISP can then be plotted every day as a graph shown in Figure 3, and one has a visual overview of the behaviour over time from day to day.

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`" [9] to do retro inspection of unusual or odd behaviour in the HTTP communication between the sites. Unfortunately the packet capture is only available at the monitoring node. In addition the local logs are available at each sliver that is invoked.

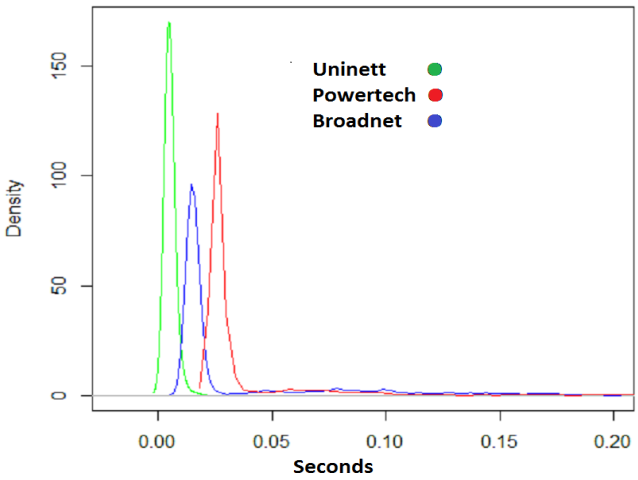

Figure 4. Density plot of connection times

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 one shown in Figure 3. This, however, results in great numbers of plots and checking them all for abnormalities can be a daunting task. By setting limits on the invocation time T_t on tunnels or slivers with 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 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 and sliver pair, are shown as a density plot in Figure 4 or as a visual plot of an empirical cumulative distribution as shown in Figure 5. Given the shape of the distributions and the number of samples per day, the Kolmogorov-Smirnov test [9] is chosen to be the most suitable test to compare the daily connection time data.

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 research network in Norway, whereas Powertech and Broadcom are commercial ISP providers in Norway.

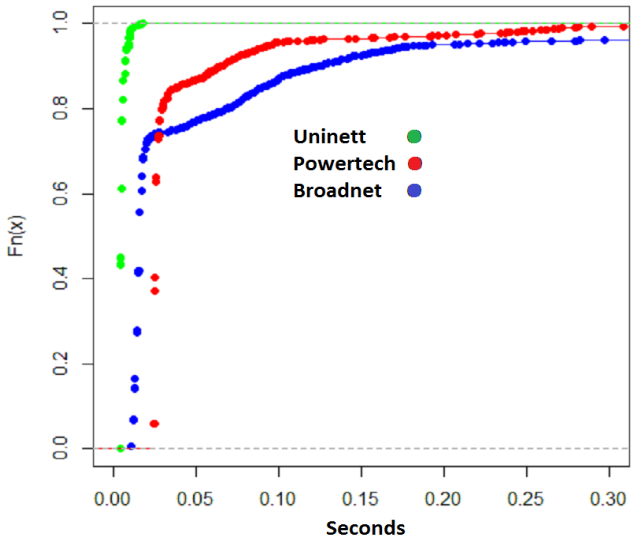

Figure 5. Empirical Cumulative Distribution Function

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. 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, where as "local" issues

 ISBN: 978-1-61208-429-9

are issues that affect only one site. Even though there are variations the connection-time distribution is stable, unless there is a change in the IP packet route or a change in the tunnel-box SW. By viewing the density plots (Figure 4) or empirical cumulative distribution functions (ECDP) (Figure 5) on tunnel and sliver pairs, the tunnels repeat the same plots. In addition the Kolmogorov-Smirnov tests have been run to show that the days without changes or downtime give the same distribution.

The memory usage on the slivers has also been checked, and the slivers do not appear to be disturbed by other processes on each server. They all show a regular pattern in the amount of free available memory, and are hence not disturbed or affected by external factors.

The Web Server SW and the scripts used to run the experiments are all available at github [6].

V. CONCLUSIONS

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 [2].

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. 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 [11]. To be able 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.

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 A/S 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.

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] NorNet Core, https://www.nntb.no/pub/nornet-configuration/NorNetCore-Sites.html , Last seen June 2015

[2] NorNet publications, https://www.nntb.no/publications/, Last seen June 2015

[3] Planet lab, https://www.planet-lab.org/ , Last seen June 2015

[4] Fedora 18 (Spherical Cow), http://docs.fedoraproject.org/en-US/Fedora/18/html/Release_Notes/index.html , Seen June 2015

[5] Jetty, application server, http://eclipse.org/jetty/ , Seen June 2015

[6] GitHub, code base, https://github.com/sunejak/EmbeddedJetty , Seen June 2015

[7] Curl, tool, http://curl.haxx.se/docs/manual.html , Seen June 2015

[8] Crontab, tool, http://www.unix.com/man-page/linux/5/crontab/ , Seen June 2015

[9] Wireshark, tool, https://www.wireshark.org , Seen June 2015

[10] Vito Ricci, "Fitting distributions with R", http://cran.r-project.org/doc/contrib/Ricci-distributions-en.pdf , Seen June 2015

[11] Brian Harry, "How do you measure quality of service?", http://blogs.msdn.com/b/bharry/archive/2013/10/14/how-do-you-measure-quality-of-a-service.aspx , Seen June 2015

Article in conference proceedings:

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

[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, Pages 611–616, ISBN 978-1-4799-1775-4, DOI 10.1109/WAINA.2015.76