

Design and Implementation of the NORNET CORE Research Testbed for Multi-Homed Systems

Thomas Dreibholz, Ernst Gunnar Gran
Simula Research Laboratory, Network Systems Group
Martin Linges vei 17, 1364 Fornebu, Norway
{dreibh,ernstgr}@simula.no

Abstract—The Internet has made it possible to communicate and to use services over large geographical distances. While it has originally been built for less critical services like e-mail and file transfer, it is nowadays also increasingly often used for availability-critical services like e.g. e-commerce or healthcare. Clearly, the reachability of such services must be ensured by so-called multi-homing of endpoints. That is, endpoints are simultaneously connected to multiple Internet Service Providers (ISP) to provide redundancy. If one ISP has problems, it is intended that the connection to another one still works. However, such assumptions have never been verified in real, large-scale setups.

The intention of the NORNET project is to build up a realistic Internet testbed for multi-homing. In this paper, we describe the design of NORNET with focus on the implementation of its fixed-line part: NORNET CORE. This paper is intended to give researchers an overview of its mode of operation, its capabilities as well as its interesting feature realisations. The knowledge about these items is very useful to plan own experiments in the NORNET testbed.¹²

Keywords: NORNET, Testbed, Multi-Homing, Design, Implementation

I. INTRODUCTION

The Internet has quickly become a critical infrastructure that our modern society heavily relies on. No longer is the Internet only a carrier of less time- and availability-critical services like e-mail and file transfer, but it now also carries services like e.g. e-commerce and healthcare, where even a temporary disruption of service could result in huge economical losses, reduced quality of healthcare or even death in the extreme cases. Failures, however, are unavoidable, and our current Internet needs to be improved to ensure the availability of critical services even when the underlying network fails. One way of doing this is to increase resilience through the use of *multi-homing*. A multi-homed node is connected to more than one *Internet Service Provider* (ISP), in order to improve availability through the use of multiple Internet connections. As long as at least one ISP is operational, the node and its services should remain available.

The size and complexity of the Internet, together with the increased use of the network as a carrier for critical services, is a barrier for doing changes to the current infrastructure of the

network. New ideas, like new protocols and algorithms related to multi-homing, need to be thoroughly tested and verified before they can be deployed. Furthermore, the testing and verification needs to be conducted in a realistic environment for the results to be valid. To satisfy this need in the context of multi-homing, we are realising the distributed multi-homed testbed NORNET. Particularly, it will be the first testbed platform that provides the possibility to perform realistic and large-scale multi-homing experiments in the Internet.

The goal of this paper is to introduce the NORNET testbed to the reader. Therefore, the paper will give a general overview of the testbed, and a detailed description of the design and implementation of the fixed-line part of it, NORNET CORE. By sharing the implementation details of NORNET CORE and informing about the capabilities of the testbed, we would like to invite researchers in multi-homing to become future users of the NORNET testbed.

II. THE DESIGN

The NORNET testbed is made up of two parts, NORNET EDGE and NORNET CORE. NORNET EDGE is the wireless part of the NORNET infrastructure. It consists of more than 500 multi-homed measurement nodes that are geographically distributed in Norway, each node being connected to five wireless broadband ISPs. NORNET CORE, on the other hand, which is the main subject of this paper, consists of a set of multi-homed, programmable sites being interconnected by wired Internet connections.

Figure 1 shows the location of the ten first NORNET CORE sites. Each site consists of four servers and a switch, where one of the servers acts as a so-called *tunnelbox*. While the switch connects all the servers, or *nodes*, locally at a site, the tunnelbox is responsible for connecting its own site to other sites in NORNET CORE, by using the available set of ISPs (see Figure 2). Note that a tunnelbox creates connections, or more precisely IPv4 and IPv6 *tunnels*, by using all possible combinations of local ISPs and remote ISPs at all the other sites' tunnelboxes. That is, the topology of the NORNET CORE is a fully connected mesh. At each site, UNINETT (the ISP for the Norwegian research institutions) is used as the default ISP for all administrative purposes.

The additional servers at the sites, that is, the ones not being tunnelboxes, host the virtual resources that are made

¹Parts of this work have been funded by the Research Council of Norway (Forskningsrådet), prosjektnummer 208798/F50.

²The authors would like to thank the reviewers for their helpful comments.

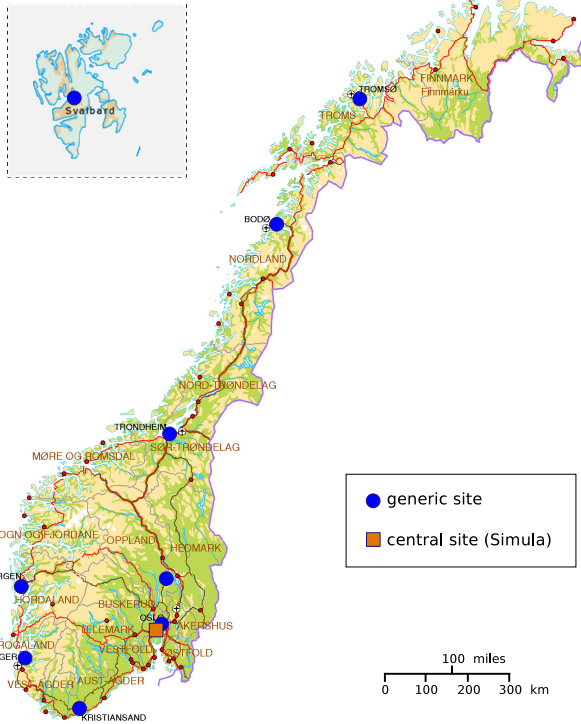


Figure 1. The NORNET CORE Sites Map

available for researchers to perform experiments in NORNET CORE. While there are currently three such servers at each site, it is of course possible to extend their number if there is need for increased research resources. The administration of resources and users is realised by using a private PLANET-LAB [1] installation, with a central server located at the Simula Research Laboratory [2] site. Resources are then requested and provided to the researchers as *slices*. A key challenge of the original PLANETLAB testbed is the low availability of nodes. At the time this paper was written, only about 36% of the PLANETLAB member nodes were actually available³ for usage. To avoid a similar situation in NORNET CORE, a close monitoring and administrative scheme is enforced throughout the testbed.

III. THE IMPLEMENTATION

In the following, we describe the implementation of the fixed-line NORNET CORE testbed.

A. The Testbed Management

NORNET CORE intends to reuse the PLANETLAB [1] software whenever possible. This software is available in a couple of distributions, most notably the original PLANETLAB software itself as well as ONELAB⁴. All variants have in

³We have associated all 1,035 registered nodes to a test slice but were able to login to only 367, i.e. about 36%, of them. Test made on June 28, 2012.

⁴ONELAB: <http://www.onelab.eu/>.

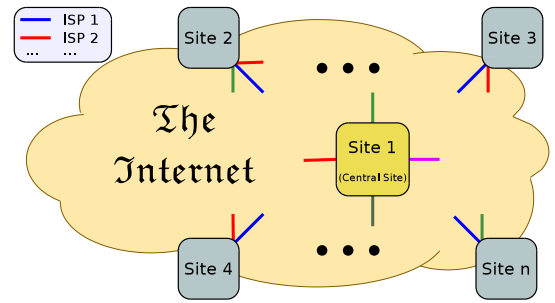


Figure 2. The NORNET CORE Network Schema

common that there is a central administration component, denoted as *PlanetLab Central* (PLC) [3]. The PLC manages a database with the configurations of sites, nodes, users and slices. Also, it provides a web-based administration interface as well as an XMLRPC-based API – called PLCAPI [4] – for software-based access to the configuration database. A particularly useful feature of the PLC software is to allow for custom extensions by adding so-called *tags* to the database. Tags are additional information to be managed, like e.g. testbed-specific configuration parameters.

For NORNET CORE, it has been straightforward to reuse the PLC as a basis. Custom, NORNET-specific configuration is simply realised by appropriate tags. In particular, tags are used to store additional information about the tunnelbox configurations (ISPs, addresses, etc.) and provider networks available at each site. The PLCAPI allows the NORNET components – which will be introduced later – to access their configuration information and apply it as needed.

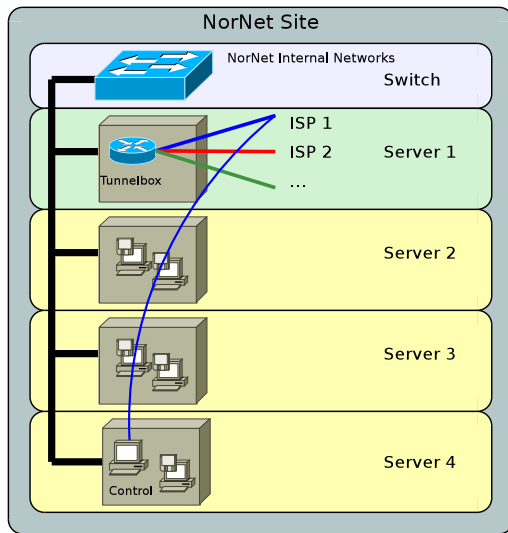
B. The Sites

Each NORNET CORE site needs appropriate hardware to realise routing (i.e. the tunnelbox functionality) as well as to provide resources for researchers. Therefore, we have decided to deploy the setup depicted in Figure 3 at each site; Subfigure 3(a) shows a schematic illustration of a site, while subfigure 3(b) shows a picture of a real setup⁵. A site physically consists of one managed switch and four HP DL320 servers. All servers have an identical configuration (4-core x86_64-CPU, 8 GiB RAM, ca. 500 GiB harddisk, and two 1000BaseT Ethernet interfaces). Since server 1 (figure 3(a)) is intended to work as a tunnelbox, it also contains an additional network interface card with four 1000BaseT Ethernet ports. The first of these ports is connected to UNINETT; the other ports are used for further providers (not yet connected in the depicted setup). The switch connects the first built-in Ethernet port of the servers 1 to 4. This Ethernet hosts the NORNET-internal, provider-specific networks (which are realised by Ethernet VLANs).

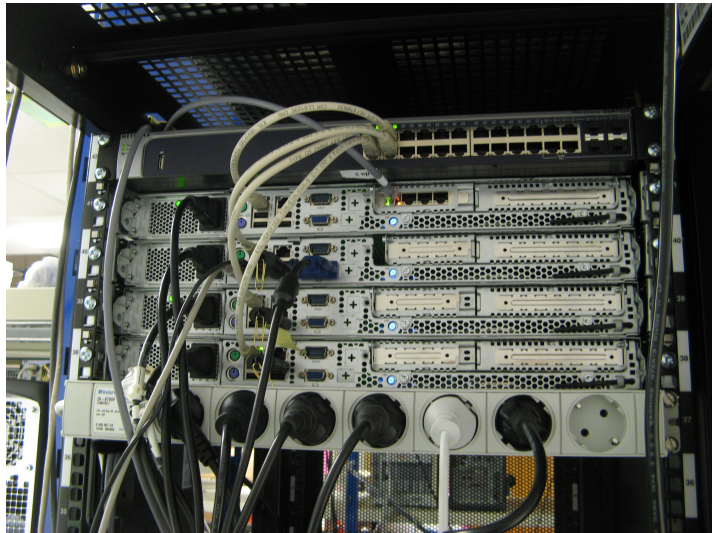
The operating system installed on the servers is UBUNTU LINUX⁶ 12.04 LTS, a version with five years support, realised as a quite minimal server installation (to be explained in more

⁵Setup at Universitetet i Tromsø (UiT), Troms/Nord-Norge.

⁶UBUNTU LINUX: <http://www.ubuntu.com/>.



(a) Schematic View



(b) A Real Deployment

Figure 3. The Physical Site Setup

detail in Subsection III-C). The purpose of this installation is to allow for basic configuration tasks, remote login by using *Secure Shell* (SSH) [5] as well as to run VIRTUALBOX [6] to host virtual machines for all further tasks. Particularly, this means that the tunnelbox, a control system (allowing remote debugging and configuration tasks) and all research systems are realised as virtual machines. This allows for a good utilisation of the available hardware resources, and it is furthermore possible to add and remove systems on demand (e.g. for testing, for special kernel-based experiments not being possible with the PLANETLAB software, etc.). Also, it allows to easily backup and restore the virtual systems, without requiring physical access to the hardware itself. This should make it possible to resolve most problems *quickly*, in order to achieve minimal downtime for the testbed users.

To provide access to a site even when server 1 (hosting the tunnelbox and therefore handling routing to other NORNET CORE sites and the Internet) is not working properly, the control system on the fourth server is also equipped with a global IP address in the UNINETT network (not yet connected in Subfigure 3(b)).

For fast deployment, the UBUNTU LINUX setup for the machines has been preseeded on an installation USB-stick. Preseeding [7] denotes to automatically apply a pre-defined configuration, i.e. partitioning the harddisk, selecting software packages and performing post-installation configuration tasks. That is, to install a server, the only manual tasks are to set up the network and to confirm repartitioning. Once server 1 is able to perform routing, the three other servers can run the installation procedure in parallel. Network access during installation is necessary to ensure up-to-date package versions (including security updates).

While all sites host virtual machines with one tunnelbox, a control system and some research systems (to be explained in detail in Subsection III-G), the Central Site – at the

Simula Research Laboratory (see the map in Figure 1) – also hosts the PLC server (see Subsection III-A) as well as the network monitoring infrastructure (to be introduced in Subsection III-H). Also, a file server is available to store virtual machine image backups.

C. The Software

UBUNTU LINUX, as a variant of DEBIAN LINUX, is based on the *Advanced Packaging Tool* (APT) [8]. The DEBIAN tool-chain allows for easy creation, validation and building of software packages, including automatic handling of software dependencies, where the packages can be provided in a software repository for download. For UBUNTU, there is a cloud service provided for this task on LAUNCHPAD⁷: *Personal Package Archives* (PPA). Using PPA, it is just necessary to create a source package and upload it into this cloud service. It will then automatically build binary installation packages, which are afterwards provided in a package repository. Since this service has shown to be very convenient, it is also applied for NORNET CORE. All NORNET CORE systems just need to refer to the PPA for package installation, in order to obtain the up-to-date NORNET CORE software. Furthermore, the APT-based packaging with its support for automatic dependency resolution allows to efficiently break down a larger software system, like NORNET CORE, into multiple packages. The NORNET CORE packages are depicted in Figure 4. Please refer to this figure while reading about the packages in the following sections.

D. The Physical Servers

The installation of a physical server (see Subsection III-B) is described by the meta-package “Server”. It is automatically installed by the preseed procedure and particularly ensures the

⁷LAUNCHPAD: <http://www.launchpad.net/>.

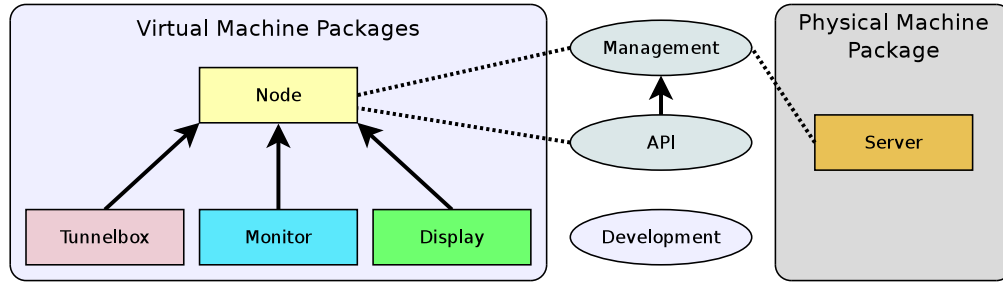


Figure 4. The Software Packages

installation of VIRTUALBOX [6] for the virtualisation, plus system management tools. These added system management tools include e.g. TRACEROUTE (routing debugging), WHOIS (IP ownership query tool), SUBNETCALC⁸ (IP network calculator), NETPERFMETER [9] (advanced network performance metering tool), RSPLIB [10] tools (RSerPool debug tools), OPENSSH (SSH server for remote login [5]), SNMP tools (for accessing network management information [11]) and GPM⁹ (mouse support for the console). Since these tools are not only useful for physical servers, they have been grouped in the meta-package “Management”. It will be installed as a dependency of the “Server” package.

E. The Virtual Nodes

All standard virtual NORNET CORE systems run UBUNTU LINUX and are based on the “Node” package. Similar to “Server”, it depends on “Management” to install the system management tools. Also, “Node”-based systems need to access the PLCAPI (see Subsection III-A). To make it easier to use the NORNET-specific extensions (stored as tags), the “API” package contains some PYTHON classes and scripts for that purpose.

In addition, the “Node” package installs some management scripts. Particularly, CRON-APT¹⁰ will be set up to automatically apply APT-based package updates, in order to keep the installation up-to-date. Also, a CRON-job is installed to regularly query the PLC database for configuration changes. Once there is a configuration update, it will be applied. The following configuration tasks are handled in this way:

- The hostname of the system is set.
- The IP configuration of the NORNET-internal networks is applied.
- *Domain Name System* (DNS) [12] server addresses are set.
- The *Network Time Protocol* (NTP) service for time synchronisation is configured (to be explained in Subsection III-F).
- The *Simple Network Management Protocol* (SNMP) [11] service is set up. It allows to query status information of the system within the NORNET-internal network. Partic-

ularly, it will be used for monitoring (to be explained in Subsection III-H).

Also, the “Node” package installation updates the configuration of the GRUB2 boot manager. Particularly, it sets a higher screen resolution. The to-be-booted Linux system then uses this resolution for its consoles (1024×768 instead of 640×480; this is the maximum possible for non-X11 consoles in VIRTUALBOX). This setting is much more suitable for current displays, making local configuration more convenient.

F. The Virtual Tunnelboxes

Clearly, the purpose of the “Tunnelbox” package is to realise the tunnelbox functionality. It depends on the “Node” package and just adds the tunnelbox features to the configuration tasks:

- The tunnels are set up.
- The NTP service is set up.
- The DNS service is configured.

1) *Tunnel Setup*: The first part of the tunnel setup obviously consists of configuring the tunnels themselves. For IPv4 tunnels, *Generic Route Encapsulation* (GRE) [13] is applied. This is a well-known standard for tunnelling over IPv4; it is therefore assumed that, in case of future extensions, it can also be handled by firewalls/middleboxes at the gateway to an ISP. IPv6 on the other hand, directly uses tunnelling over IPv6. If there is no IPv6 interconnection provided by the ISPs of two tunnelboxes, IPv6 will just be multiplexed, together with IPv4, over the GRE tunnel.

After setting up the tunnels, appropriate routing has to be configured. By default, routing is based on the longest prefix match for the destination address as well as on the routing metric. For multi-homed systems, however, this is not sufficient. That is, all packets for the same destination IP address would take the same route (via the same ISP, of course). To support multi-homing, routing also needs to consider the source address. For example, if a packet has a source IP address of the ISP UNINETT, the packet should go over the interface to UNINETT. On the other hand, a packet having a source IP from the ISP TELENOR should be sent out via TELENOR. This functionality is realised by applying IP-rules [14]: an own routing table is configured for each ISP. Then, IP rules select the routing table for a packet based on its source address.

In order to give the researcher even more control over the outgoing interface, IP rules realise another feature: by setting

⁸SUBNETCALC: <http://www.iem.uni-due.de/~dreibh/subnetcalc/>.

⁹GPM: <http://www.nico.schottelius.org/software/gpm/>.

¹⁰CRON-APT: <http://inguza.com/software/cron-apt>.

the *Type of Service* (TOS) field of IPv4 packets/Traffic Class of IPv6 packets to certain values, the sender can “override” the outgoing ISP. That is, even if a packet has a source address in the TELENOR network, the researcher can decide to send it via the UNINETT interface. Due to a limitation in the TOS/Traffic Class handling within the Linux kernel, it is only possible to use seven combinations of values. For the ISP selection based on TOS/Traffic Class, this limits the number of outgoing ISPs per site to seven. This is assumed to be sufficient, and – with a patched kernel – it could also be increased to 63, if really needed. An alternative approach for realising the routing could be to apply OPENFLOW [15], in form of the OPEN VSWITCH [16] extension for Linux.

2) *Time Synchronisation*: For many network measurement tasks, it is quite handy to e.g. measure the one-way delay. For that purpose, however, exact time synchronisation between the sender side and the receiver side of a packet flow is necessary. NTP [17] is the IETF standard for time synchronisation. Since a tunnelbox is available locally at each site, and it is running on a dedicated machine, the NTP daemon of a tunnelbox is chosen to act as a local server for the site. That is, the systems within the local network can use it for synchronisation. Furthermore, the NTP daemons of all sites act as peers. That is, they form a NORNET-internal network for time synchronisation. In addition, the NTP daemon at the Central Site uses the external NTP servers of the Physikalisch-Technische Bundesanstalt¹¹ (PTB) in Braunschweig/Germany for time synchronisation. Based on atomic clocks, the PTB provides the official time reference for Central Europe. It can therefore be considered to be a highly reliable and trustworthy time source.

3) *Name Resolution*: A tunnelbox also acts as a DNS server [12] – by using the BIND [18] package – for its local site, providing caching for external DNS resolutions. Furthermore, in order to help researchers working with the NORNET CORE private network addresses, all interfaces also have a DNS name below the private `nor.net` top-level domain (TLD). Each site has its own second-level domain below this TLD, e.g. `simula.nor.net` is the domain of the Central Site as the Simula Research Laboratory. Each site’s tunnelbox is the master DNS of its second-level domain; the Central Site’s DNS acts as slave. For the Central Site (managing the TLD and the `simula.nor.net` second-level domain), the other sites’ tunnelboxes act as slaves. This should ensure a sufficient resilience and load distribution.

A further feature of the NORNET CORE DNS service is that it provides LOC resource records [19] for all NORNET-internal DNS names. These records provide the geographic location of the names, allowing researchers to easily access this information and make use of it.

G. The Research Systems

The research systems are the virtual machines that researchers use for their experiments. Clearly, they are the purpose NORNET CORE has been built for. In many cases, it is – with extensions to make use of multi-homing – just

sufficient to use PLANETLAB-based virtual systems here. Systems based on this software [1] obtain their configurations from the PLC and install automatically. It is just sufficient to obtain a bootstrap CD-ROM image (from the PLC) and boot a virtual PLANETLAB-based research system with this image.

A system based on the PLANETLAB software runs slivers (i.e. virtual machines) for the experiments. Originally, the PLANETLAB software – as well as the different variants of it – have applied LINUX VSERVER¹² for this purpose. However, LINUX VSERVER is not in the mainline kernel, which makes kernel updates difficult. Particularly, this has been the reason why the stable PLANETLAB distributions are based on quite out-of-date variants of FEDORA CORE LINUX (e.g. version 14 for ONELAB; version 12 for original PLANETLAB). The currently ongoing development direction of the PLANETLAB software is therefore to replace LINUX VSERVER with LINUX CONTAINERS¹³ (LXC). Particularly, LXC is fully included in the mainline Linux kernel, making it easy to deploy state-of-the-art kernels as well as underlying Linux distributions.

At the moment, there are already experimental builds of the LXC-based PLANETLAB software available¹⁴. Besides the fact that it will support to assign an own IPv4 address per sliver (which avoids complicated separation of flows among slivers), it will also support IPv6. Due to these advantages, in combination with the very active development, we are intending to base NORNET CORE on the LXC-based software. Also, we are actively involved within the development community, by contributing tests of and improvements to the new software.

Slices – regardless of whether they are based on LXC or LINUX VSERVER – provide lightweight virtual Linux systems for the researchers. Particularly, there is only a single Linux kernel running. All slices running on the system share it; changes to the kernel would therefore affect *all* slices of that system. However, in some cases, it might be useful to run full virtual systems, e.g. in order to deploy custom kernels to perform evaluations of CMT-SCTP under FreeBSD [20], [21] or kernel-based Linux MPTCP [22]. In this case, further VIRTUALBOX-based full virtual machines could be configured on the physical servers. In the future, the instantiation of virtual machines to boot custom kernels could e.g. be automatised by a system like TOMATO [23].

H. The Monitoring Setup

In order to achieve the testbed availability described in Section II, tight monitoring is necessary to detect problems early and to trigger their resolution as soon as possible. The basis of our monitoring infrastructure is a NAGIOS [24] setup, which is deployed in a virtual machine at the Central Site. The “Monitor” package provides this functionality, based on the “Node” package (see Subsection III-E). NAGIOS actively obtains the status of all components, e.g. the systems themselves with PING tests, and services like the PLC’s web server by HTTP checks. Particularly, NAGIOS allows a fine-granular configuration of warning and error conditions, as well as the

¹¹Physikalisch-Technische Bundesanstalt: <http://www.ptb.de/>.

¹²LINUX VSERVER: <http://linux-vserver.org/>.

¹³LINUX CONTAINERS: <http://lxc.sourceforge.net/>.

¹⁴LXC-based PLANETLAB build: <http://build.onelab.eu/lxc/>.



Figure 5. “NorNet-Kontrollsenster” – A Screenshot of the Control Center Display

extension to custom services (e.g. to check certain system-dependent status values by using SNMP) in form of plug-ins. Such plug-ins have been written to perform checks of sites and tunnels. The monitoring results are recorded by NAGIOS; a web interface provides access for administrators.

While the web interface presents a very detailed and fine-grained view of NORNET CORE, a more illustrative view was intended to give a quick overview of the site statuses for both administrators and users of the testbed (i.e. the researchers). Therefore, we – as a first step – applied the NAGMAP¹⁵ plug-in for NAGIOS, which provided a map view of the NORNET CORE testbed. However, NAGMAP just produced a quite simple projection of the sites and their connections (i.e. the tunnels) to a GOOGLE MAPS¹⁶ satellite map. Particularly due to the fully-connected mesh topology of the tunnels (see Section II), the link view, however, became relatively confusing. Therefore, we have developed our own plug-in named NORNET-KONTROLLSENTER.

NORNET-KONTROLLSENTER uses APACHE/PHP [25] to access the recorded NAGIOS status information. It then provides an HTML page with the results. That is, the plug-in runs on the monitoring system itself. The results can then be accessed from a web server running on the same system, while a web browser on another machine can display the corresponding webpage (as shown in Figure 5). The returned web page with the results makes use of JavaScript, AJAX (Asynchronous JavaScript and XML) as well as SVG (Scalable

Vector Graphics) with JavaScript to provide dynamic updates without having to reload the whole page. Furthermore, instead of tying the map display to a certain map service, we apply OPENLAYERS [26]. It provides a service-independent mapping and layering library in JavaScript; the display can dynamically change between map services (currently, we apply OPEN STREET MAP¹⁷, GOOGLE MAPS and BING MAPS¹⁸). Furthermore, overlaying functionality is provided. That is, overlays like the OPEN WEATHER MAP¹⁹ overlay can add additional information to the map (in Figure 5: the current weather as well as clouds and precipitation forecasts). In the future, we plan to use this feature for more detailed network status information. Map services and the visibility of overlays can be changed interactively.

For improved visibility, we only display a connection to the Central Site (in green colour) if all tunnels of a site are working. Only non-working tunnels would result in yellow (warning level) or red (error level) coloured links between sites. Connections to not-yet-deployed sites are represented by grey, dashed lines.

Since the access to the monitoring results is browser-based, it is possible to run multiple browser instances simultaneously. We use this feature e.g. to display a map of the Norwegian mainland (see Figure 5) on a large screen as well as a map of Svalbard (more than 1,000 km north of the mainland) on a separate, small screen. The “Display” package extends the “Node” setup with configurations for such a display machine,

¹⁵NAGMAP: <https://github.com/hecko/nagmap/>.

¹⁶GOOGLE MAPS: <https://maps.google.no/>.

¹⁷OPEN STREET MAP: <http://www.openstreetmap.org/>.

¹⁸BING MAPS: <http://www.bing.com/maps/>.

¹⁹OPEN WEATHER MAP: <http://www.openweathermap.org/>.

by adding an X server and the FIREFOX web browser.

IV. TESTBED DEPLOYMENT AND RESEARCH IDEAS

Currently, we are in the deployment phase of the sites [27]. That is, the hardware setup described in Subsection III-B has been installed at 7 sites, with 3 further sites in the following weeks.

For the research part of NORNET CORE, there are obviously use cases for research on resilience (e.g. in the context of RSerPool [10], [28], [29], etc.). Furthermore, there is a strong interest in using the testbed in the context of multi-path transport performance with MPTCP [22] and CMT-SCTP [21], as well as for congestion control evaluations [30]–[32] in that context. Particularly, there is a growing interest in CMT-SCTP, since *Real-Time Collaboration on the World Wide Web* (RTCWEB) [33] is based on SCTP. RTCWEB is a framework for browser-to-browser multimedia communications (e.g. video conferencing) that is currently under standardisation in the IETF and therefore a very hot research topic. With SCTP being deployed by possibly billions of web browsers on frequently multi-homed endpoints (e.g. tablet PCs and smartphones with 3G and WLAN interfaces), the usage of multi-path transport might become very tempting.

V. CONCLUSIONS

NORNET CORE is a new Internet testbed with support for multi-homing. Therefore, it can particularly be applied for research topics in the areas of resilience and multi-path transport. Currently, it is in the deployment phase, with sites distributed all over Norway. There is already interest in performing research in the testbed for resilient networks as well as for multi-path transport – particularly in the context of RTCWEB. Therefore, even though “the road to hell is paved with unused testbeds” [34], there is good reason to assume that NORNET CORE will be utilised by researchers in order to advance the insights into multi-homed networks and systems for the benefits of all Internet users.

REFERENCES

- [1] L. Peterson, A. Bavier, M. Fluczynski, S. Muir, and T. Roscoe, “Towards a Comprehensive PlanetLab Architecture,” PlanetLab Consortium, Tech. Rep. PDN-05-030, June 2005.
- [2] A. Tveito, A. M. Bruaset, and O. Lysne, *Simula Research Laboratory – by thinking constantly about it*. Heidelberg, Baden-Württemberg/Germany: Springer, Nov. 2009, ISBN 978-3642011559.
- [3] M. Huang, *MyPLC User’s Guide*, Aug. 2006.
- [4] OneLab, *PlanetLab Central API Documentation*, July 2012.
- [5] T. Ylonen and C. Lonvick, “The Secure Shell (SSH) Connection Protocol,” IETF, Standards Track RFC 4254, Jan. 2006, ISSN 2070-1721.
- [6] Oracle Corporation, *Oracle VM VirtualBox User Manual*, Oct. 2012.
- [7] Ubuntu Documentation Team, *Ubuntu Server Guide for Ubuntu 12.04 LTS*, Apr. 2012.
- [8] A. Barth, A. D. Carlo, R. Hertzog, L. Nussbaum, C. Schwarz, and I. Jackson, *Debian Developer’s Reference*, Aug. 2012.
- [9] T. Dreibholz, M. Becke, H. Adhari, and E. P. Rathgeb, “Evaluation of A New Multipath Congestion Control Scheme using the NetPerfMeter Tool-Chain,” in *Proceedings of the 19th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Hvar/Croatia, Sept. 2011, pp. 1–6, ISBN 978-953-290-027-9.
- [10] T. Dreibholz, “Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture,” Ph.D. dissertation, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Mar. 2007.
- [11] D. Harrington, R. Presuhn, and B. Wijnen, “An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks,” IETF, Standards Track RFC 3411, Dec. 2002, ISSN 2070-1721.
- [12] P. V. Mockapetris, “Domain Names – Implementation and Specification,” IETF, Standards Track RFC 1035, Nov. 1987, ISSN 2070-1721.
- [13] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Generic Routing Encapsulation (GRE),” IETF, Standards Track RFC 2784, Mar. 2000, ISSN 2070-1721.
- [14] M. A. Brown, *Guide to IP Layer Network Administration with Linux*, Apr. 2003.
- [15] Open Networking Foundation, *OpenFlow Switch Specification – Version 1.2*, 5th ed., Dec. 2011.
- [16] J. Pettit, J. Gross, B. Pfaff, M. Casado, and S. Crosby, “Virtual Switching in an Era of Advanced Edges,” in *Proceedings of the 2nd IEEE Workshop on Data Center – Converged and Virtual Ethernet Switching (DC-CAVES)*, Niagara Falls, Ontario/Canada, Sept. 2010.
- [17] D. L. Mills, J. Martin, J. Burbank, and W. Kasch, “Network Time Protocol Version 4: Protocol and Algorithms,” IETF, Standards Track RFC 5905, June 2010, ISSN 2070-1721.
- [18] Internet Systems Consortium, *BIND 9.7.x Administrator Reference Manual*, Mar. 2010.
- [19] C. Davis, P. Vixie, T. Goodwin, and I. Dickinson, “A Means for Expressing Location Information in the Domain Name System,” IETF, RFC 1876, Jan. 1996, ISSN 2070-1721.
- [20] T. Dreibholz, “Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol,” Habilitation Treatise, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Mar. 2012.
- [21] T. Dreibholz, H. Adhari, M. Becke, and E. P. Rathgeb, “Simulation and Experimental Evaluation of Multipath Congestion Control Strategies,” in *Proceedings of the 2nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Fukuoka/Japan, Mar. 2012, ISBN 978-0-7695-4652-0.
- [22] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, “Improving Datacenter Performance and Robustness with Multipath TCP,” in *Proceedings of the ACM SIGCOMM*, Toronto/Canada, Aug. 2011.
- [23] D. Schwerdel, D. Hock, D. Günther, B. Reuther, P. Müller, and P. Tran-Gia, “ToMaTo – A Network Experimentation Tool,” in *Proceedings of the 7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Shanghai/People’s Republic of China, Apr. 2011.
- [24] E. Galstad, *Nagios Core Documentation*, Aug. 2010.
- [25] Apache Software Foundation, “Apache HTTP Server Version 2.4 Documentation,” 2012.
- [26] Open Source Geospatial Foundation, “OpenLayers Developer Documentation,” July 2012.
- [27] T. Dreibholz, “The NorNet Project – A Research Testbed for Multi-Homed Systems,” Invited Talk, Karlstad, Värmland/Sweden, Nov. 2012.
- [28] T. Dreibholz, X. Zhou, M. Becke, J. Pulinthanath, E. P. Rathgeb, and W. Du, “On the Security of Reliable Server Pooling Systems,” *International Journal on Intelligent Information and Database Systems (IJIIIDS)*, vol. 4, no. 6, pp. 552–578, Dec. 2010, ISSN 1751-5858.
- [29] P. Lei, L. Ong, M. Tüxen, and T. Dreibholz, “An Overview of Reliable Server Pooling Protocols,” IETF, Informational RFC 5351, Sept. 2008, ISSN 2070-1721.
- [30] H. Adhari, E. P. Rathgeb, and X. Zhou, “Throughput Analysis of Load-Sharing Protocols in an Inter-Continental Testbed Environment,” in *Proceedings of the 3rd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Barcelona, Catalonia/Spain, Mar. 2013.
- [31] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, “On the Fairness of Transport Protocols in a Multi-Path Environment,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Ottawa, Ontario/Canada, June 2012, pp. 2666–2672.
- [32] M. M. Yousaf, M. Welzl, and B. Yener, “Accurate Shared Bottleneck Detection Based On SVD and Outliers Detection,” University of Innsbruck, Institute of Computer Science, Innsbruck, Tirol/Austria, Tech. Rep. NSG-DPS-UIBK-01, Aug. 2008.
- [33] H. T. Alvestrand, “Overview: Real Time Protocols for Browser-based Applications,” IETF, Internet Draft Version 04, June 2012, draft-ietf-rtcweb-overview-04.txt, work in progress.
- [34] E. K. Çetinkaya and J. P. G. Sterbenz, “Programmable Networking with GpENI,” University of Kansas, Communication Networks Laboratory, Presentation, May 2011.