Computer Networks 61 (2014) 75–87

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

NorNet Core – A multi-homed research testbed ☆

Ernst Gunnar Gran *, Thomas Dreibholz, Amund Kvalbein

Simula Research Laboratory, Network Systems Group, Martin Linges vei 17, 1364 Fornebu, Norway

ARTICLE INFO

Article history:
Received 1 July 2012
Received in revised form 18 October 2013
Accepted 27 December 2013
Available online 3 January 2014

Keywords:
NorNet Core
Testbed
Multi-homing
Routing
Transport
Applications

ABSTRACT

Over the last decade, the Internet has grown at a tremendous speed in both size and complexity. Nowadays, a large number of important services – for instance e-commerce, healthcare and many others – depend on the availability of the underlying network. Clearly, service interruptions due to network problems may have a severe impact. On the long way towards the Future Internet, the complexity will grow even further. Therefore, new ideas and concepts must be evaluated thoroughly, and particularly in realistic, real-world Internet scenarios, before they can be deployed for production networks. For this purpose, various testbeds – for instance PlanetLab, GpENI or G-Lab – have been established and are intensively used for research. However, all of these testbeds lack the support for so-called multi-homing.

Multi-homing denotes the connection of a site to multiple Internet service providers, in order to achieve redundancy. Clearly, with the need for network availability, there is a steadily growing demand for multi-homing. The idea of the NorNet Core project is to establish a Future Internet research testbed with multi-homed sites, in order to allow researchers to perform experiments with multi-homed systems. Particular use cases for this testbed include realistic experiments in the areas of multi-path routing, load balancing, multi-path transport protocols, overlay networks and network resilience. In this paper, we introduce the NorNet Core testbed as well as its architecture.

1. Introduction

The Internet has become a critical infrastructure in our modern society. Individuals, organisations and governments rely on the algorithms, protocols, services and applications that constitute the Internet for conducting their business. Failures or unavailability of central components in the network immediately transforms to irritation, monetary loss and sometimes also breakdown in public services. Add to this the enormous scale of the Internet, and it becomes evident that the barrier for making changes to this infrastructure is high. New ideas must be thoroughly tested and validated before they can be deployed in production networks. It has long been clear that such testing must be done in a setting that transcends the traditional lab environment, in order to capture the complexity of scale, traffic and network heterogeneity that exists in a real network. Such tests can, however, often not be done in existing production networks, since they can potentially influence the stability of the network. This has led to an increased interest in recent years for large-scale distributed network testbeds to support experimentation with Future Internet technologies. The characteristics of these testbeds vary. Some offer a large number of nodes and are well suited for testing scalability [1], others target particular technologies such as optical [2] or wireless [3] networks, while

☆ Parts of this work have been funded by the Research Council of Norway (Forskingsrådet), prosjektnummer 208798/F50. The authors would like to thank Martin Becke for his friendly support.

* Corresponding author. Tel.: +47 99644916.
E-mail addresses: ernstgr@simula.no (E.G. Gran), dreibh@simula.no (T. Dreibholz), amundk@simula.no (A. Kvalbein).

1389-1286
http://dx.doi.org/10.1016/j.bjp.2013.12.035

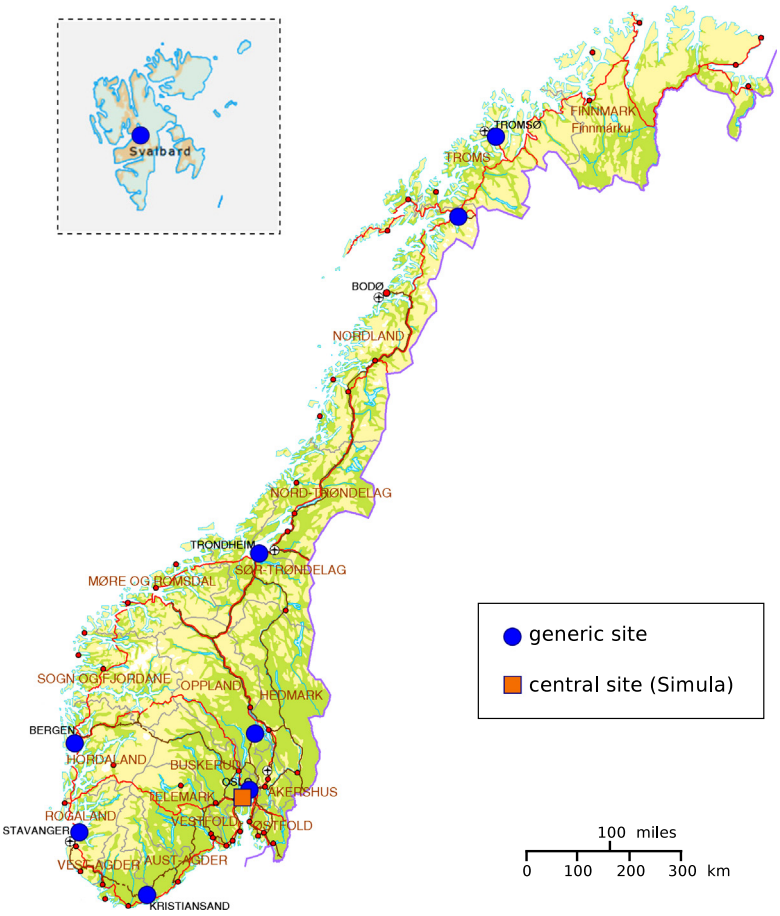

Fig. 1. The current NorNet sites map.

yet others offer the compute power that is needed to test the functionality of heavily distributed applications [4]. The common goal for these testbeds is to provide a realistic environment for testing out an idea, beyond what can be achieved at a single location.

This paper presents NorNet Core,[1] a distributed, wired testbed for experimental networking research that is currently being constructed in Norway. The NorNet Core is built in the context of the NorNet project, a project that also builds the complementary test-bed NorNet Edge [5,6]; a flexible research infrastructure for conducting measurements and experimentation with mobile broadband networks. Initially, the NorNet Core consists of 10 programmable sites that are geographically spread across most of Norway, mainly at universities and other research institutions, as shown in Fig. 1. In addition, two international sites in Essen, Germany, and Haikou, China, provide a view on the Norwegian network from abroad. Further international sites will be deployed in the future. The defining characteristic of NorNet Core is a strong focus on supporting experiments that exploit multi-connectivity. Each site will be connected to at least two wired Internet Service Providers (ISPs), and all these connections will be exposed and available for use. Furthermore, all the sites being part of NorNet Core will be interconnected as a fully connected mesh, including all possible combinations of available ISPs as the individual sites. This allows the use of multiple (potentially partly overlapping) paths between any pair of sites in the testbed. This in turn opens up the possibility for a range of experiments in the areas of multi-path routing, load balancing, multi-path transport protocols, overlay networks or network resilience, just to mention a few.

NorNet Core is built on the MyPLC software developed by the PlanetLab[2] [1] consortium. This has the advantage of a large and well-maintained code base and user community, and eases federation with other similar testbeds. NorNet Core extends the functionality offered by MyPLC by giving experimenters access to multiple network connections. Anybody can apply for a user account in NorNet Core, but access will be regulated in order to guarantee that each experiment receives sufficient resources.[3]

The rest of this paper is organised as follows: In Section 2, we give an overview of relevant Internet testbeds.

[1] NorNet: http://www.nntb.no.

[2] PlanetLab: http://www.planet-lab.org/.

[3] In case of contention, priority will be given to experiments with the involvement of a Norwegian research group.

In Section 3, we discuss design choices related to NorNet Core, before we give a thorough description of the NorNet Core architecture in Section 4. In Section 5 we highlight some experiments where the multi-homing capabilities of NorNet Core should be of particular value. We shortly present the current status of the testbed deployment in Section 6, before we provide a short tutorial overview of an experiment based on NorNet Core in Section 7. We conclude and discuss the future direction of NorNet Core in Section 8.

2. Related work

There is currently a significant focus in the research community on building large and realistic testbeds as key enablers for the Future Internet. These testbeds are intended to provide a flexible environment for performing measurements and testing. Examples of large initiatives supporting such testbeds are FIRE[4] in the European Union and GENI[5] [7] in the United States. These and other initiatives have led to the establishment of several large distributed testbeds, with different goals and characteristics.

The most well-known large distributed testbeds are PlanetLab [1] and its European sibling OneLab,[6] which give users access to processing and network resources on more than thousand nodes in all regions of the world. These testbeds are very well suited for evaluating large distributed systems like peer-to-peer networks. However, the limited resources and large user base makes it difficult to guarantee sufficient resources to each experiment.

Other testbeds are smaller in size, but offer extended functionalities or more powerful resources. FEDERICA[7] [8] offers a high-capacity network testbed based on dedicated channels in European research and education networks. Emanicslab[8] provides a distributed testbed based on the MyPLC framework for use by Emanics partners. G-Lab[9] [9], which is also partly based on MyPLC software, provides access to both wired and wireless nodes at tens of sites across Germany. GpENI[10] [10], which has been a source of inspiration for NorNet, adds flexibility by interconnecting sites by Data Link Layer tunnels (or optical channels at some sites). GpENI is a global infrastructure, with sites in the United States, Europe and Asia. Finally, PanLab[11] provides a platform for integrating testbeds located at different institutions. In contrast to NorNet Core, however, none of these testbeds have a particular focus on multi-homed sites.

In addition to the dedicated experimental facilities mentioned above, most national research and education networks are also used to support network experiments. The advantage of these networks over dedicated testbeds is that they carry real user traffic, and can therefore provide a more realistic environment. This is, however, also their biggest drawback: since they are production networks, they cannot be used for experiments that jeopardise normal operations.

3. Design choices

When using a set of ISPs to provide multi-connectivity between sites, a central decision to make is whether to establish the needed site-to-site connections – the *tunnels* over the Internet – at the Data Link Layer or at the Network Layer. In the case of NorNet Core, this decision translates into a question of whether to bridge the LANs that constitute each site into an amalgamated NorNet Core LAN (e.g. by using L2TPv3 [11] or a similar protocol), or to run the LANs as autonomous entities interconnected by routers (e.g. by using Virtual Private Network (VPN) software or IP tunnels between the sites).

Creating a distributed testbed as an amalgamated LAN has several advantages. The connectivity between all the nodes will be provided by the LAN technology itself, while LAN-targeted management tools could be used for managing and monitoring the whole testbed infrastructure. The GpENI network is an example of a testbed that successfully utilises L2TPv3 capable Cisco routers to connect the different sites that constitute the testbed.

While NorNet Core is inspired by GpENI, there is however one major difference between these two testbeds. This difference is the support of NorNet Core for multi-homed sites. The GpENI network is built as a star-shaped topology. In particular, this means that the topology does not contain any loops. In contrast, the NorNet Core topology with its mesh of interconnected ISPs will contain a multitude of loops. Using the de facto LAN standard Ethernet [12] to internally connect nodes at each site, creating an amalgamated NorNet Core LAN would then easily result in broadcast storms and MAC address table instability. Broadcast storms and MAC address table instability are typically avoided in LANs containing loops by enabling the Spanning Tree Protocol (STP) [13]. The STP, however, breaks a loop in the topology, the cause of the before mentioned problems, by removing a link from the loop, i.e. leaving the link idle. In the case of the NorNet Core testbed, this kind of behaviour is not acceptable, as it will have an adverse effect on the multi-homing characteristics of each site. When a new ISP is added to a NorNet Core site, it creates several loops in the NorNet Core topology as new tunnels are created from the newly added ISP to the ISPs at other sites. In such a scenario, the STP would detect these newly created loops and remove them by leaving the new tunnels idle (or alternatively idle some of the old tunnels being part of the same set of loops). In other words, enabling the STP would effectively remove the multi-homing capabilities of the sites in the NorNet Core testbed, which is of course unacceptable.

It is possible to imagine a configuration of Data Link Layer tunnelling for the NorNet Core testbed where each tunnel between the ISPs is configured as a separate VLAN [14]. The STP is then not needed, as each separate VLAN no longer contains any loops in the topology. Such a configuration would, however, introduce an unwanted level of

[4] FIRE: http://cordis.europa.eu/fp7/ict/fire/.
[5] GENI: http://www.geni.net/.
[6] OneLab: http://www.onelab.eu/.
[7] FEDERICA: http://www.fp7-federica.eu/.
[8] Emanicslab: http://www.emanicslab.org/.
[9] G-Lab: http://www.german-lab.de/.
[10] GpENI: http://wiki.ittc.ku.edu/gpeni/.
[11] PanLab: http://www.panlab.net/.

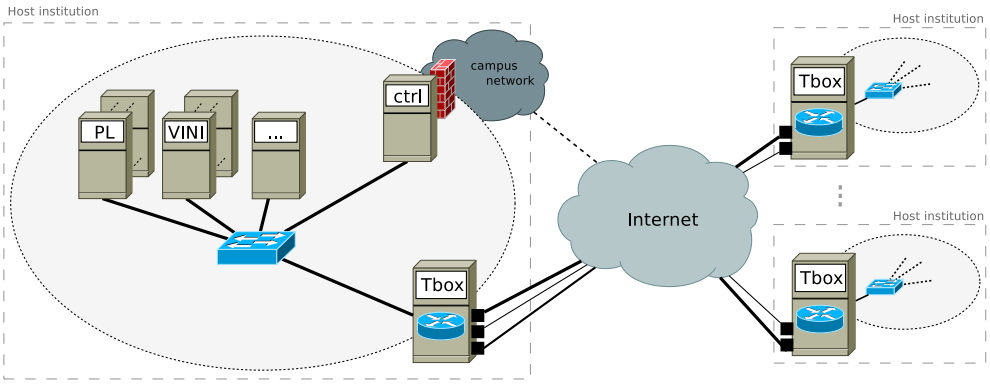

Fig. 2. The NorNet Core architecture overview.

complexity to the testbed, and at the same time remove some of the benefits of having an amalgamated NorNet Core LAN. A decision was therefore made to establish the tunnels between the NorNet Core sites at the Network Layer.

Tunnels between sites at the Network Layer are commonly created by using third-party VPN software. Such software, like tinc,[12] typically include encryption, compression and self-routing techniques to facilitate the user requirement of creating a private and secure network. The multi-homing characteristics of NorNet Core could, however, again pose a challenge. Each NorNet Core site needs to handle a large set of partly overlapping tunnels,[13] potentially in the range of about 100 tunnels[14] per site for our initial 12 site NorNet Core topology. Such a number of encrypted tunnels could pose a scalability issue. It is of great importance that the computational requirements to handle the tunnels at a site do not introduce a load to the system that interferes with the experiments running in the testbed. Furthermore, it is of the utmost importance, that the VPN software itself does not interfere with the routing between the NorNet Core site nodes where the researcher using the testbed should be able to choose exactly which tunnels to use for a given experiment. We need to avoid any unfortunate side-effect or hidden routing caused by behind-the-scenes intelligence in the VPN software itself.

The scalability and routing concerns considered, adding the fact that the NorNet Core testbed per se has no need for encrypted tunnels,[15] we decided to establish the tunnels between the NorNet Core sites using static IP tunnels. More specifically, and as further detailed in Section 4, the static IP tunnels are realised by using the Generic Routing Encapsulation (GRE) protocol [15] over IPv4 and IPv6-over-IPv6 tunnels, as implemented by the Linux operating system.

[12] tinc: http://www.tinc-vpn.org/.

[13] As explained further in Section 4, between any two sites there will exist several tunnels to take advantage of all possible combinations of ISPs at the two sites.

[14] For instance, 2 local ISPs at a site S having 11 other sites as peers with 2 ISPs each and 2 IP protocols – i.e. IPv4 and IPv6 – result already in $2 \times 11 \times 2 \times 2 = 88$ tunnel endpoints at S.

[15] Note that this does not put any restrictions on the users of the testbed. They may freely include encryptions in their experiments if wanted. Encryption will just not be provided as a service by the testbed itself.

4. The NorNet Core architecture

In the following, we describe the NorNet Core architecture that has been developed in accordance with the design choices explained above.

4.1. Overview

An overview of the NorNet Core architecture is presented in Fig. 2. It consists of multiple sites at different locations (see also Fig. 1), where each site consists of a set of nodes: the *research nodes* (PL, VINI,...) constitutes the nodes where researchers will actually run their experiments, a *control node* (ctrl) provides the institution hosting a site with local access to the site for e.g. local monitoring, while a *tunnelbox* (Tbox) manages all the tunnels that connect this site to other NorNet Core sites, using the available ISPs.

Each site is connected to at least two ISPs. For simplicity, we have allocated a unique identification number $P_i \in [1,255] \subset \mathbb{N}$ – which is denoted as NorNet *Provider Index* – for each ISP i used in the NorNet setup. Having a site a – identified by a unique identification number $S_a \in [1,255] \subset \mathbb{N}$ denoted as NorNet *Site Index* – connected to the ISPs $\hat{P_a} = \{P_{a_1}, P_{a_2}\}$ and a site S_b connected to ISPs $\hat{P_b} = \{P_{b_1}, P_{b_2}, P_{b_3}\}$, there are $\left|\hat{P_a}\right| \times \left|\hat{P_b}\right| = 2 \times 3$ paths from S_a to S_b possible, as illustrated in Fig. 3:

$$P_{a_1} \rightarrow P_{b_1}; P_{a_1} \rightarrow P_{b_2}; P_{a_1} \rightarrow P_{b_3};$$
$$P_{a_2} \rightarrow P_{b_1}; P_{a_2} \rightarrow P_{b_2}; P_{a_2} \rightarrow P_{b_3}.$$

That is, traffic from site S_a can use the two outgoing providers $\hat{P_a}$; traffic received at site S_b can come in from the three incoming providers S_b. All six possible paths from site S_a to site S_b are represented by static tunnels among the corresponding sites' provider endpoints. Note, that the reverse direction (i.e. site S_b to site S_a) works in the same way; it has been omitted here for simplification. Also, it has to be noted that the tunnel setup is separate for each Network Layer protocol (i.e. IPv4 and IPv6).

At each site, the tunnels are terminated at the tunnelbox. The tunnelboxes are routers that form a fully-connected mesh of tunnels among the NorNet Core sites. They also connect the research nodes at the sites as well as the management infrastructure. All nodes i within a site

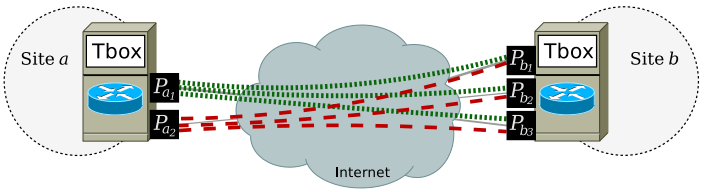

Fig. 3. A NorNet Core tunnel example.

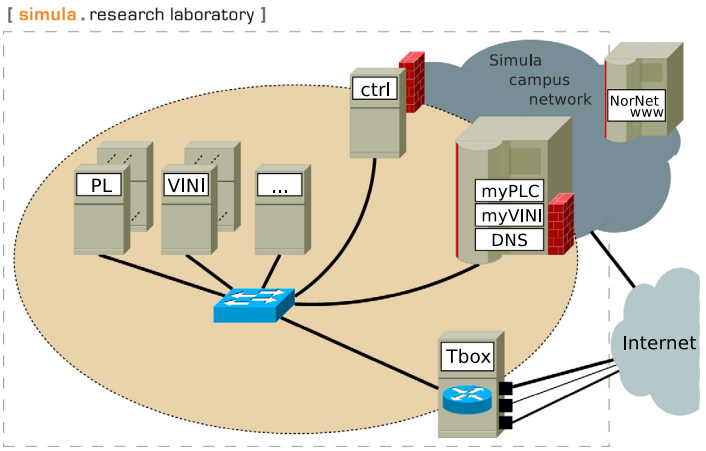

Fig. 4. The NorNet Core central site at the Simula research laboratory.

are identified by a site-unique NorNet *Node Index* $N_i \in [1, 255] \subset \mathbb{N}$.

Site #1 is the Simula central site; an overview of this site is provided in Fig. 4. Beside the research nodes and the tunnelbox, it also contains the management and monitoring infrastructure for the NorNet Core testbed. The following sections describe the NorNet Core parts in more detail.

4.2. Site address space layout

The tunnelling scheme is particularly applied to allow for a clean and systematic NorNet-internal addressing of all components. Since IPv4 addresses are scarce, it has been considered as being practically impossible to allocate a *single* consecutive address space for the whole per-provider NorNet Core network from each of the used ISPs. For instance, this would have meant that an ISP P_1 would have had to provide an address space that is large enough to subdivide it into a subnet for *each* existing and future site – also taking future growth of each site into consideration.

Therefore, we have decided to use private address spaces within the sites, with routing over the public networks through the tunnels among the tunnelboxes. To keep the addressing scheme simple and clean, we have used the subnetting scheme described in the following.

For IPv4, the devices on each site are addressed by:

$$\underbrace{10}_{/8\ \text{NorNet}}.\langle\text{Provider Index}\rangle.\langle\text{Site Index}\rangle.\langle\text{Node Index}\rangle\,/24.$$

/16 Provider Network

/24 Site Network

That is, a node N_8 on site S_1 connected to the ISPs P_1 and ISP P_2 has the address $10.P_1.S_1.N_8$ in ISP P_1's network as well as the address $10.P_2.S_1.N_8$ in ISP P_2's network. Also, e.g. the whole network of ISP P_1 is simply $10.P_1.0.0/16$. Since the address space for each ISP is contiguous, the routing tables remain small and the routing process simple and efficient.

For IPv6, we have adapted the IPv4 approach but received the public address space 2001:700:4100::/48:

$$\underbrace{2001{:}700{:}4100}_{/48\ \text{NorNet}}:\langle PP\rangle\langle SS\rangle::\langle XXXX\rangle:\langle NN\rangle\,/64.$$

/56 Provider Network

/64 Site Network

/112 Node-Internal Network

Here, PP denotes the two-digit hexadecimal Provider Index, SS the two-digit hexadecimal Site Index and NN the two-digit hexadecimal Node Index. Since IPv6 provides plenty of address space, it is furthermore possible to use node-internal addressing denoted by XXXX (a four-digit hexadecimal index). This is e.g. applied for allocating address space to virtual machines on a physical node. Note, that the remaining unused 40 bits could easily be used for such purposes as well.

Instead of using a public IPv6 address space, it would also be possible to *uniquely* enumerate future NorNet variants by using IPv6 Unique Local addresses [16], i.e. realising a private but – in contrast to private IPv4 addresses – globally unique addressing scheme. This allows for connecting such networks without a need for address changes to make all addresses in the resulting combined network unique.

As a further simplification, we define that the tunnelbox on each site always has a Node Index of 1, i.e. it will have the lowest possible host addresses in each of a site's NorNet subnets.

4.3. Access providers and tunnels

Each of the NorNet Core sites within Norway will be connected to the provider UNINETT,[16] which manages the Norwegian national research and education network. Particularly, all universities in Norway are connected by UNINETT. Since it provides a fast and reliable network, it will also be the network to be used for all administrative and monitoring communication purposes among the national sites.

For the additional ISP connections, we prefer to have a mix of different connection types. That is, while we will add further high-speed connections, it is also desired to add the type of interconnection that is provided to "regular" customers, i.e. consumer-type broadband Internet access. Particularly, such interconnections should get the same kind of "best effort" Quality of Service (QoS) as regular consumers experience for their everyday Internet usage. This will allow for representative network evaluation experiments.

While UNINETT supports native IPv6 Internet access (i.e. without the need for tunnelling all packets over IPv4), the availability of IPv6 from consumer ISPs is – despite the exhaustion of the IPv4 address space – still quite limited. As one of our design goals is IPv6 support, NorNet Core makes use of IPv6 between the tunnelboxes of two sites if the corresponding providers of a path support it. For a UNINETT ↔ UNINETT relation, this is of course mostly[17] the case. Then, the IPv6 packets are tunnelled over a separate IPv6-over-IPv6 tunnel between the two tunnelboxes. However, if one side only supports IPv4, IPv6 traffic will be transported over the existing IPv4 tunnel along with normal IPv4 traffic. Clearly, IPv6 experiments have to keep this fact in mind.

Currently, NorNet Core is only intended for best effort traffic experiments. At the moment, only UNINETT would be able to provide certain QoS guarantees. However, if QoS functionalities become more widespread – in particular also for consumer Internet connections – in the future, it would be possible to add appropriate functionalities (e.g. bandwidth reservations, etc.) into the tunnelboxes as well.

4.4. The tunnelboxes

The main purpose of the tunnelboxes is the routing among the NorNet Core sites through the tunnels that represent the different combinations of outgoing and incoming ISPs. Classic Internet routing for a packet is just based on its destination address. Then, the appropriate output port of the router is chosen by the longest prefix match [12] in the router's global routing table. However, for a multi-homed site, this simple procedure is not sufficient any more. Here, all packets to the same destination would just take the same path – which is clearly not the intended behaviour. Instead, for instance, a packet originating from the ISP I_1 address space of a site (i.e. identified by the packet's *source* address) should be routed through an appropriate tunnel (i.e. chosen by the packet's *destination* address) over the access of ISP I_1. Likewise, a packet having an ISP I_2 source address should be routed over an access of ISP I_2. This functionality requires separate routing tables, with a selection of the routing table based on a packet's source address.

A feature of the Linux networking stack is policy-based routing [17]. It provides the capability of selecting a separate routing table based on conditions like the packet source address and the value of the Type of Service field (TOS, for IPv4, see [18]) or Traffic Class field (for IPv6, [19]). Therefore, our tunnelboxes are realised by Linux-based systems which are configured with appropriate routing tables for IPv4 and IPv6, as well as with policies to select one of the tables based on a packet's source address.

As an additional feature, we also make use of the Type of Service/Traffic Class field support of the policy-based routing functionality in Linux by using it to explicitly allow a sender to select a specific outgoing provider. That is, a research node at a site could e.g. send a packet with a source address from ISP I_1, but request it to be routed through a tunnel over a different (local) ISP, ISP I_2, instead. This allows for experiments with asymmetric packet routes, i.e. an application's packets from a site S_a to a site S_b may explicitly take a different path than it is used for the answer back from site S_b to site S_a. For this purpose, three bits in the DiffServ Code Point (DSCP) field of Type of Service or Traffic Class (defined in [20]) are used as an index for the outgoing provider in the Provider Index list of all providers the site is connected to. That is, for instance, if the site S_a is connected to the ISPs $\hat{P}_a = \{1, 8, 42\}$, a DSCP-based index of 3 will then choose the third provider in this list (i.e. provider #42 here). The outgoing packet is then routed through a tunnel via this provider.

The DSCP index of 0 means to just use the default provider, i.e. the provider given by the packet's source address. Then, the three bits allow an explicit output specification for the first seven (i.e. $2^8 - 1$) providers only. The limitation to three bits instead of six – as reserved for the DSCP [20] – is a limitation of the routing policy implementation in Linux. If necessary, this limitation could be removed by adapting the kernel implementation to use all six bits (then allowing specification of $2^6 - 1 = 63$ separate outgoing providers in the DSCP). However, as for now, NorNet sites connected to more than seven local ISPs seem to be unlikely. Note, that the explicit provider choice by DSCP index only affects the *outgoing* provider chosen a site. The incoming provider of the remote site is identified by the destination address of a packet.

At each site, the local tunnelbox connects all the devices within the site's provider networks (see Subsection 4.2). Physically, however, a site's internal networks are just realised by a single Gigabit Ethernet topology. The different provider address spaces are just a logical configuration and can be realised as Virtual LANs within the Ethernet. This allows for an inexpensive setup of a site. Adding a

[16] UNINETT: https://www.uninett.no/.

[17] A few sites have not yet fully deployed IPv6 connectivity from their UNINETT endpoint to their NorNet Core setup.

new provider just means to physically connect the tunnelbox; any further site configuration is just being performed in software.

In addition to provide the routing, each tunnelbox also hosts a Network Time Protocol (NTP) [21] service for time synchronisation, as well as a Domain Name System (DNS) [22] service for name lookups. The latter is particularly used to map the names of all systems within NorNet Core to addresses (see also Subsection 4.2) as well as to provide a reverse lookup of addresses back to names. A detailed overview of the technical realisation of the tunnelboxes is provided in [23].

4.5. Research nodes

The research nodes of each site are – of course – the reason for setting up NorNet Core. They allow researchers to run experiments on them which make use of the multi-homed topology. Conceptually, NorNet Core with tunnels and tunnelboxes is independent of a specific research platform. However, since node management tasks like configuration, resource provisioning and sharing are recurring tasks, it is useful to have at least some common and generic platforms. Therefore, all NorNet Core sites will provide at least some nodes that are based on the PlanetLab software platform.

4.5.1. PlanetLab/OneLab

PlanetLab[18] [1] is the oldest and most widespread network research testbed platform; its core software is also reused in various adapted forms for other testbeds. Particularly, OneLab[19] is a European testbed initiative that provides its own code branch[20] of the original PlanetLab software. The OneLab branch is particularly interesting in the context of NorNet Core due to its out-of-the-box kernel support for the multi-homed Transport Layer protocol SCTP [24,25].

Nodes based on the PlanetLab software are Linux-based physical machines that run virtual machines. The nodes are centrally administrated by a *Planet Lab Control* (PLC) server [26]. The PLC takes care of managing user accounts and so-called *slices*. A slice is a reserved set of resources in the testbed used to conduct an experiment with certain attributes (e.g. access permissions, bandwidth restrictions, etc.). A node can be mapped to one or more slices. For each slice, the node will then instantiate its own virtual machine – denoted as *sliver* – with the given permissions. That is, a node is shared among all researchers running slivers on it. The virtualisation software ensures that different slivers on the same machine do not interfere with each other (although some interaction may be explicitly permitted by setting certain permissions).

The PlanetLab node software is based on Linux-VServers,[21] an operating-system-based virtualisation approach for Linux. Linux-VServers itself is not a part of the standard Linux kernel; it therefore requires a patched kernel with corresponding userland tools. However, the current mainline Linux kernel development prefers the approach of linux containers[22] (LXC), providing relatively similar functionalities. The current development direction of the PlanetLab/OneLab software therefore goes in the same direction. LXC-based OneLab builds[23] are available now and used in the NorNet Core deployment. NorNet Core is one of the first experimental users of this software and we are also in contact with the developers in order to contribute improvements.

Besides the advantage that the LXC-based OneLab builds provide a much easier possibility to use state-of-the-art Linux kernels and software, it also provides a significantly improved network handling in comparison to the original PlanetLab software. Particularly, it uses Open vSwitch[24] [27] to provide a virtual switch that is used to connect the slivers. This virtual switch is then bridged into the site's NorNet Ethernet. This provides the possibility to use separate addresses for each sliver, i.e. a researcher can use its "own" addresses, without a need to share them with other slivers. In contrast, the original PlanetLab software shared a single IPv4 address per node among all slivers, which resulted in a restriction to TCP and UDP as transport protocols (plus SCTP in the case of OneLab) and a mapping of ports to slivers. Also, the new LXC-based software provides IPv6 support as well.

4.5.2. Other platforms

Research nodes based on other testbed platforms – like VINI [28], ToMaTo [29], etc. – can also be installed as required. For these nodes, a site's tunnelbox is just a regular IPv4/IPv6 router that has to be appropriately configured into the components.

4.6. Management infrastructure

Clearly, NorNet Core needs a management infrastructure to maintain and distribute the configurations of the tunnelboxes at different sites. Our intention here has been to reuse as much of the existing testbed infrastructure as possible. Since we deploy a PlanetLab-based infrastructure [26] for all sites, we have decided to integrate the tunnelbox management into this framework. The PlanetLab software takes care of general testbed management tasks by providing a database, a web-based configuration interface, a cryptographically secured XMLRPC interface [30] to access and modify the configuration, as well as user, node and site management. Therefore, we have just added special attributes to sites and nodes records for holding the NorNet-specific configuration information. A tunnelbox, based on a lightweight Linux setup, then uses the XMLRPC API to obtain the configuration data for setting up interfaces, routing policies and routes, as well as for dynamically providing information about any changes to the list of locally connected ISPs.

[18] PlanetLab: http://www.planet-lab.org/.
[19] OneLab: http://www.onelab.eu/.
[20] OneLab source code repository: http://git.onelab.eu/.
[21] Linux-VServers: http://www.linux-vserver.org/.
[22] linux containers: http://lxc.sourceforge.net/.
[23] OneLab LXC builds: http://build.onelab.eu/lxc/.
[24] Open vSwitch: http://www.openvswitch.org/.

4.7. Network monitoring

An important lesson learned from PlanetLab usage is that a tight monitoring of the nodes is necessary, in order to make sure that the whole testbed is available and usable by the researchers. This is actually a major issue for PlanetLab: currently, only 575 nodes of 1042 nodes[25] are up and running, i.e. the availability is just about 55%. Clearly, NorNet Core intends to do significantly better and targets a research node availability of at least 90%. To reach this availability, two goals have to be met:

1. Quick detection of node failures and problems.
2. Fast reaction to the detected issues.

To achieve the first goal, i.e. a quick detection of issues, we are going to utilise the network management tool Nagios [32]. A monitoring station at the Simula central site will continuously observe the status of all components and trigger actions in case of problems. Further details on the monitoring of NorNet Core can be found in [23].

Hardware failures, of course, need on-site actions at specific sites, like replacing a broken harddrive or network cable. A quick detection of an issue therefore ensures that such actions can be triggered as fast as possible. The NorNet Core administration will strictly require reasonably fast reaction times from all of its member sites, where all hardware used are bought with an on-site service agreement.

Due to the experimental characteristic of NorNet Core, a node unavailability will in many cases just be caused by a software failure – e.g. a kernel deadlock or system crash. In such cases, a node reset (e.g. by power-cycling) is necessary. E.g. for PlanetLab, the usage of a *power control unit* (PCU), i.e. a remotely-controllable device that can switch the power for connected components, is just optional. Since there is also no standardised PCU API, many PCUs may need manual operation by a human operator (e.g. login on a special web interface, etc.). For NorNet Core, it is intended to make the availability of an automatically controllable PCU for all devices mandatory. That is, the network control at the Simula central site will be able to remotely power-cycle devices in order to try to make them work again quickly and without further on-site interaction.

5. Applications

NorNet Core is a flexible network testbed that can facilitate a wide range of network experiments. These experiments can focus on mechanisms at the networking layer or above. In the following, we highlight some types of experiments where NorNet Core with its strong focus on site multi-homing can be particularly well suited.

5.1. Network layer

Multi-path routing is an old topic in the networking literature. A multitude of routing algorithms exist that can provide more than one next-hop for a given network destination (for an overview, see [33]). Given the availability of multiple paths between two end hosts, one of the main challenges becomes how traffic should be distributed across the different paths. This is a challenge both at the network edge and in the core of the network. At the edge, a multi-homed stub network can employ different strategies to distribute traffic load based on price or performance [34]. In the core of a network, different strategies can be used to split traffic on available paths. Traditional multi-path methods such as Equal Cost Multi-Path (ECMP) will split traffic equally on the available paths. Better performance can be achieved by unequal traffic splitting over paths with different costs, as done in e.g., DEFT [35]. In this approach, relatively more traffic is sent on the shorter paths. In a related line of work, several proposals have been made for dynamic load balancing, where the amount of traffic sent on each path is adjusted based on the current load situation in the network [36–38]. NorNet Core is very well suited to support experiments with network-layer techniques for more efficient load balancing.

5.2. Transport layer

Beside multi-homed routing, NorNet will also be a useful experimental platform for multi-homed Transport Layer protocols. Currently, two protocol extensions are very actively discussed in the context of the IETF Transport Services Working Group (TSVWG): the Concurrent Multipath Transfer extension for SCTP (CMT-SCTP; [39,40]) as well as the Multi-Path extension for TCP (MPTCP; [41,42]). Also, both protocol extensions are now available in experimental implementations, allowing for their larger-scale test within Internet setups. For example, [43] show some interesting – and relevant for the IETF discussion – measurement results on CMT-SCTP performance in a two-site Internet setup. [44,40] describe the details, and difficulties of a custom multi-site setup and suggest the creation of a generic, multi-homed testbed as experimental platform for further research. Clearly, also a larger-scale experimental evaluation of multi-path congestion control strategies for CMT-SCTP and MPTCP in realistic, multi-homed Internet setups – as suggested by [45,46,40] – could easily be realised as a NorNet Core experiment.

NetPerfMeter [44] is a Transport Layer protocol performance evaluation tool for SCTP, TCP and UDP. It is the tool that has been used for the CMT-SCTP experiments mentioned above and has also been applied for single-homed SCTP tests within the G-Lab project. We have already successfully applied NetPerfMeter for initial functionality tests in the currently deployed NorNet Core research nodes. Particularly, it provides out-of-the-box multi-homing support and also makes use of the SCTP support that is provided by the node software (see Subsubsection 4.5.1).

Particularly useful in the context of multi-homed Transport Layer protocol evaluation could also be the deployment of research node platforms like ToMaTo [29]. Nodes based on ToMaTo provide the possibility to boot custom operating system images. That is, unlike operating-system-based virtualisation approaches like PlanetLab, a researcher could evaluate the performance of

[25] Test made on September 9, 2013 with scripts from [31].

Table 1
The NorNet Core sites, September 2013.

Site Index	Site name	Location	First ISP	Second ISP
1	Simula Research Laboratory	Fornebu, Akershus/Norway	UNINETT (1)	Kvantel (2)
2	Universitetet i Oslo	Oslo, Oslo/Norway	UNINETT (1)	–[d]
3	Høgskolen i Gjøvik	Gjøvik, Oppland/Norway	UNINETT (1)	–[d]
4	Universitetet i Tromsø,	Tromsø, Troms/Norway	UNINETT (1)[a]	–[d]
5	Universitetet i Stavanger	Stavanger, Rogaland/Norway	UNINETT (1)[a]	–[d]
6	Universitetet i Bergen	Bergen, Hordaland/Norway	UNINETT (1) [a]	–[d]
7	Universitetet i Agder	Kristiansand, Vest-Agder/Norway	UNINETT (1)	–[d]
8	Universitetet pa Svalbard	Longyearbyen, Svalbard/Norway	UNINETT (1)[a]	–[d]
9	NTNU Trondheim	Trondheim, Sør-Trøndelag/Norway	UNINETT (1)	–[d]
10	Høgskolen i Narvik	Narvik, Nordland/Norway	UNINETT (1)	– [d]
42	Universität Duisburg-Essen	Essen, Nordrhein-Westfalen/Germany	DFN (30)	Versatel (31)[b,c]
88	Hainan University	Haikou, Hainan/China	CERNET (80) [a]	Unicom (81) [a]

[a] IPv6 available from ISP but not yet deployed to NorNet Core site.
[b] IPv6 not yet available from ISP.
[c] Consumer-grade ADSL connection.
[d] Negotiations with ISPs are in progress.

specially-adapted, kernel-based network stacks. Clearly, this is very interesting [6] for IETF-related research – like the ongoing activities on MPTCP and CMT-SCTP – that has a strong focus on "running code" in real Internet setups.

5.3. Higher layers

While the research on multi-homed transport is currently focused on just a few approaches (i.e. mainly SCTP and MPTCP), there is a large number of applications that can benefit from an underlying multi-homing infrastructure.

Applications with need for network resilience are a major use case. An interesting approach to unify a set of server redundancy functionalities – like server pool management and session handling – in combination with multi-homed SCTP-based transport is the Reliable Server Pooling (RSerPool) framework [47,48]. The core of RSerPool has been standardised by the IETF [49]. However, there are still active Internet Drafts that need further evaluation, particularly in the context of realistic, multi-homed Internet setups. [50,31] show PlanetLab-based results on RSerPool performance in a single-homed, large-scale Internet setup. By using NorNet Core, RSerPool research – and particularly the performance implications of an underlying, multi-homed infrastructure – becomes feasible. Currently, functional tests of NorNet Core are performed with the RSerPool demonstration platform introduced in [51].

A further, highly interesting research topic on applications is the transport of real-time multimedia data among multi-homed endpoints [52–54]. The challenge here is that data – like a video or audio stream – have to be split up among paths and recombined at the receiver while maintaining timing constraints. Unlike for a lab setup with heterogeneous high-speed links, however, distinct paths in the Internet may have very different QoS characteristics (i.e. bandwidth, delay, jitter, packet loss). A very interesting experimental application for such scenarios is the Homer framework,[26] a multimedia conferencing system with multi-path transport support based on CMT–SCTP. The CMT–SCTP-based multi-path transport is currently examined in lab setups and used for proof-of-concept demonstration purposes [55]. NorNet Core will allow performing realistic experiments with this application.

6. Current network setup

Table 1 presents the sites of the NorNet Core as of September 2013. Currently, it consists of 10 sites in Norway (see Fig. 1 for their geographic location). All of these sites use UNINETT (Provider Index 1) as their primary ISP, as explained in Subsection 4.3. While in fact IPv6 should be available at all of these sites, some of them have not yet deployed IPv6 to their NorNet Core site setup itself. Also, currently only the Simula central site is connected to Kvantel[27] (Provider Index 2) as the second ISP. Negotiations to add further ISPs are in progress.

The first international NorNet Core site (Site Index 42) has been deployed at the Institute for Experimental Mathematics of the University of Duisburg-Essen in Essen, Germany. It is connected to the Deutsches Forschungsnetz (DFN, Provider Index 30) – the German research network that corresponds to UNINETT in Norway – as the primary ISP, as well as a consumer-grade Asymmetric Digital Subscriber Line (DSL) connection from Versatel[28] (Provider Index 31) as the second ISP. Unlike the other ISP connections, which are fibre-based with a symmetric speed of at least 100 Mbit/s, the ADSL link is asymmetric with a downstream of 16 Mbit/s and an upstream of just 1 Mbit/s.

The second NorNet Core site outside of Norway (Site Index 88) is hosted at the College of Information Science and Technology at the Hainan University in Haikou, China. It is connected to the China Education and Research Network (CERNET[29], Provider Index 80) – the Chinese research network – as the primary ISP, as well as to China Unicom[30] (Provider Index 81) as the second ISP.

All machines at the 10 Norwegian sites are HP ProLiant DL320 G6 servers equipped with a 4-core Intel Xeon E5606

[26] Homer: http://www.homer-conferencing.com/.

[27] Kvantel: http://www.kvantel.no/; formerly Hafslund Telekom.
[28] Versatel: http://www.versatel.de/.
[29] CERNET: http://www.edu.cn/.
[30] China Unicom: http://www.chinaunicom.com/.

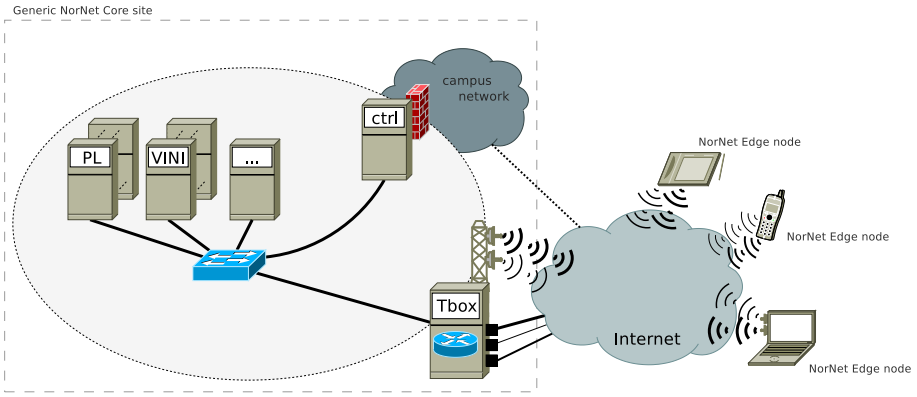

Fig. 5. The NorNet Core extension with wireless broadband access.

CPU at 2.13 GHz, 8 GiB of memory and a 450 GiB harddisk. These servers also contain HP Integrated Lights-Out (iLO). iLO provides a management instance that runs completely separate from the operating system. Its main use case is to provide PCU functionality, i.e. it allows to remotely reset the system in case of problems. Some more information on the hardware setup can be found in [23].

7. Experiment tutorial

In general, the steps to perform an experiment in the NorNet Core testbed are as follows: first a user account for the PLC server is necessary. Particularly, the new user also has to store a Secure Shell (SSH) [56] public key on the PLC server. This key will later be used to authenticate the user when accessing slivers. A new slice can then be created for the user, or the user could be mapped to an existing slice, by an administrator. The administrator also has the possibility to allocate "own" IP addresses on each node that get mapped to the slice (i.e. on nodes that are going to run a sliver of this slice). This will probably be the usual procedure for multi-homing experiments, since the user gets control over the IP addresses. Otherwise, sliver addresses are shared as with the original, non-LXC PlanetLab software (see also Subsubsection 4.5.1). Beside this address allocation process that is automated by a script, the rest of the PLC-based user and slice maintenance is the same as for PlanetLab/OneLab; details on that can e.g. be found in [26,30].

To access the NorNet Core network, the user simply connects a computer to the NorNet Ethernet of its local site. For external users, it is intended to make the network also accessible via a VPN to the Simula central site. By using the SSH private key, the user can now establish SSH connections to all of the slivers of a corresponding slice. Inside the slivers, the user finds a Linux environment that can be configured as needed, e.g. by installing additional software from the standard repository as well as custom software for research experiments. As for PlanetLab/OneLab, it is important to note that slivers should not be expected to be a reliable, permanent storage. In case of a problem with the node hosting the slivers, the standard procedure is just to reinstall the node. Such a reinstall also wipes all slivers. It is therefore strongly recommended for the user to take care of this fact by e.g. preparing scripts to easily recreate the needed configuration within the slivers.

The network configuration of each sliver will show[31] an Ethernet interface *eth0* that has one or more logical IPv4 and IPv6 networks configured – one for each local ISP. For a sliver at the Simula central site, this could e.g. be 10.1.1.120/24, 10.2.1.120/24 (i.e. Node Index 120 at site 1 for providers 1 and 2; see also Subsection 4.2). For a simple test, the user could e.g. choose a peer sliver at the international site in Essen. Let's say it has the IPv4 configuration 10.30.42.133/24, 10.31.42.133/24 (i.e. Node Index 133 at site 42 for providers 30 and 31). Then, traceroute could be used[32] on the Simula site's sliver for testing the four possible combinations of outgoing and incoming ISPs to reach the Essen site's sliver:

- Provider 1 (Simula) to provider 30 (Essen).
- Provider 1 (Simula) to provider 31 (Essen).
- Provider 2 (Simula) to provider 30 (Essen).
- Provider 2 (Simula) to provider 31 (Essen).

This is performed by choosing the right source address (i.e. either 10.1.1.120 of provider 1 or 10.2.1.120 of provider 2 at the Simula site 1) as well as the destination address (i.e. either 10.30.42.133 of provider 30 or 10.31.42.133 of provider 31 at the Essen site 42). Note, that the answer packets in all these cases take the reverse path backwards (i.e. back to the specified source address). That is, the routing is symmetric.

In order to make use of asymmetric routing, the packet TOS[33] can be set. The relevant bits are bits 2–4 (counted from 0), i.e. possible TOS settings are 0×00 (default provider), 0×04 (the first provider), 0×08 (the second provider), etc. if the two lowest bits – which are used for Explicit Congestion Notification (ECN) [57] – are set to 0. Note, that the TOS specifies the number of a provider at a site (first, second, third, etc.) and not its index (e.g. 2, 30, 31, etc.). That is, sending a packet from a provider 1 address at Simula to a provider 31 address in Essen, but setting the TOS to 0×08 (choosing the second provider, here: provider 2), will lead to sending out the packet with a source address in provider

[31] E.g., `ip -4 addr show dev eth0`.
[32] E.g. `traceroute ⟨destination⟩ -s ⟨source⟩`.
[33] Bits in the TOS field: D D D **D D D** C C.
DSCP ECN

1's network via provider 2. Since the source address is relevant for the response, the peer side will send its reply to the provider 1 address, i.e. it comes back via provider 1.

In the same way, this simple test can also be repeated by using IPv6 instead of IPv4. The Traffic Class field has the same format as the TOS for IPv4. Also, in order to keep an overview of the used addresses during the TRACEROUTE runs, it is practical for the user that the DNS service for NorNet Core [23,22] provides reverse lookup for the NorNet Core addresses.

8. Conclusions and future work

The steady growth and reliance on availability-critical services in the Internet leads to a growing interest in multi-homed systems; multi-homing will become an important property of the Future Internet. Therefore, it is necessary to test and evaluate new ideas and approaches – particularly in the areas of multi-path routing, load balancing, multi-path transport protocols, overlay networks and network resilience – in real-world, multi-homed Internet setups. The NorNet Core testbed platform, which has been presented in this paper, provides an environment to make such experiments possible. The testbed is currently under deployment [58], with a number of sites distributed all over Norway, and with a future – also international – extension in the planning stage.

As "the road to hell is paved with unused testbeds" [59], great effort has been made to ensure that the NorNet testbed actually will be used by researchers. That is, "NorNet wants to be a building block of the railroad to heaven".[34] We are currently in contact with several research groups in the area of multi-homed systems, and are also very interested in establishing new contacts. Initial experiments with multi-homed systems in the context of multi-path transport and resilient applications have already started, with further experiments in preparation.

An important future development step of NorNet Core, will be a tighter coupling with the NorNet Edge project on multi-homing with *wireless* broadband providers, offering 3G and 4G access (i.e. UMTS, LTE, etc.). As presented in Fig. 5, the tunnelboxes will be equipped with wireless devices, along with the existing cable-based Internet connections. Furthermore, NorNet Edge distributes a large set of mobile nodes over the whole country of Norway. Management of, and access to these nodes, is intended to be integrated into the infrastructure provided by NorNet Core. This will provide researchers with a unique, novel and realistic testbed for research on multi-homed systems with both wired and wireless access links that have very heterogeneous QoS characteristics. That is, experiments can be conducted in an environment similar to the world experienced by the "regular", real-world Internet users of today.

References

[1] L. Peterson, T. Roscoe, The design principles of PlanetLab, Operat. Syst. Rev. 40 (1) (2006) 11–16, http://dx.doi.org/10.1145/1113361.1113367. ISSN 0163-5980.

[2] K.-I. Kitayama, M. Koga, H. Morikawa, S. Hara, M. Kawai, Optical burst switching network testbed in Japan, in: Proceedings of the IEEE Optical Fiber Communication Conference (OFC) 3. ISBN 1-55752-783-0. http://dx.doi.org/10.1109/OFC.2005.192713.

[3] I. Broustis, J. Eriksson, S.V. Krishnamurthy, M. Faloutsos, A blueprint for a Manageable and affordable wireless testbed: design, pitfalls and lessons learned, in: Proceedings of the 7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), Orlando, Florida/USA, 2007, pp. 1–6. http://dx.doi.org/10.1109/TRIDENTCOM.2007.4444684.

[4] L. Peterson, A. Bavier, S. Bhatia, VICCI: A Programmable Cloud-Computing Research Testbed, Tech. Rep. TR-912-11, Department of Computer Science, Princeton University, 2011.

[5] A. Kvalbein, D. Baltrūnas, K. Evensen, J. Xiang, A. Elmokashfi, S. Ferlin-Oliveira, The Nornet Edge Platform for Mobile Broadband Measurements, Comput. Netw. 61 (2014) 88–101.

[6] T. Dreibholz, The NorNet testbed: a platform for evaluating multi-path transport in the real-world internet, in: Proceedings of the 87th IETF Meeting, Berlin/Germany, 2013.

[7] M. Berman, J.S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, I. Seskar, GENI: a federated testbed for innovative network experiments, Comput. Netw. 61 (2014) 5–23.

[8] M. Campanella, F. Farina, The FEDERICA infrastructure and experience, Comput. Netw. 61 (2014) 176–183.

[9] P. Müller, D. Schwerdel, B. Reuther, T. Zinner, P. Tran-Gia, Future internet research and experimentation: the G-Lab approach, Comput. Netw. 61 (2014) 102–117.

[10] D. Medhi, B. Ramamurthy, C. Scoglio, J.P. Rohrer, E.K. Çetinkaya, R. Cherukuri, X. Liu, P. Angu, A. Bavier, C. Buffington, J.P.G. Sterbenz, The GpENI testbed: network infrastructure, implementation experience, and experimentation, Comput. Netw. 61 (2014) 51–74.

[11] J. Lau, M. Townsley, I. Goyret, Layer Two Tunneling Protocol – Version 3 (L2TPv3), Standards Track RFC 3931, IETF, 2005, ISSN 2070-1721.

[12] A.S. Tanenbaum, Computer Networks, Prentice Hall, Upper Saddle River, New Jersey/USA, 1996. ISBN 0-13-349945-6.

[13] IEEE, Media Access Control (MAC) Bridges, Tech. Rep. IEEE 802.1D, LAN/MAN Standards Committee of the IEEE Computer Society, 2004.

[14] IEEE, Virtual Bridged Local Area Networks, Tech. Rep. IEEE 802.1Q, LAN/MAN Standards Committee of the IEEE Computer Society, 2006.

[15] D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, Generic Routing Encapsulation (GRE), Standards Track RFC 2784, IETF, 2000. ISSN 2070-1721.

[16] R.M. Hinden, B. Haberman, Unique Local IPv6 Unicast Addresses, Standards Track RFC 4193, IETF, 2005. ISSN 2070-1721.

[17] M.G. Marsh, Policy Routing With Linux, SAMS Publishing, 2001. ISBN 978-0672320521.

[18] J.B. Postel, Internet Protocol, Standards Track RFC 791, IETF, 1981. ISSN 2070-1721.

[19] S.E. Deering, R.M. Hinden, Internet Protocol, Version 6 (IPv6), Standards Track RFC 2460, IETF, 1998. ISSN 2070-1721.

[20] K. Nichols, S. Blake, F. Baker, D.L. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, Tech. Rep. 2474, IETF, 1998. ISSN 2070-1721.

[21] D.L. Mills, J. Martin, J. Burbank, W. Kasch, Network Time Protocol Version 4: Protocol and Algorithms, Standards Track RFC 5905, IETF, 2010. ISSN 2070-1721.

[22] P.V. Mockapetris, Domain Names – Implementation and Specification, Standards Track RFC 1035, IETF, 1987. ISSN 2070-1721.

[23] T. Dreibholz, E.G. Gran, Design and implementation of the NorNet core research testbed for multi-homed systems, in: Proceedings of the 3nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS), Barcelona, Catalonia/Spain, 2013, pp. 1094–1100. ISBN 978-0-7695-4952-1.

[24] R.R. Stewart, Stream Control Transmission Protocol, Standards Track RFC 4960, IETF, 2007. ISSN 2070-1721.

[25] T. Dreibholz, I. Rüngeler, R. Seggelmann, M. Tüxen, E.P. Rathgeb, R.R. Stewart, Stream control transmission protocol: past, current, and future standardization activities, IEEE Commun. Mag. 49 (4) (2011) 82–88, http://dx.doi.org/10.1109/MCOM.2011.5741151. ISSN 0163-6804.

[26] M. Huang, MyPLC User's Guide, 2006.

[27] J. Pettit, J. Gross, B. Pfaff, M. Casado, 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, 2010.

[34] Thomas Dreibholz, 87th IETF Meeting, MPTCP Session.

[28] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford, In VINI veritas: realistic and controlled network experimentation, ACM SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 3–14, http://dx.doi.org/10.1145/1151659.1159916. ISSN 0146-4833.
[29] D. Schwerdel, D. Hock, D. Günther, B. Reuther, P. Müller, 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, 2011.
[30] OneLab, PlanetLab Central API Documentation, 2012.
[31] T. Dreibholz, E.P. Rathgeb, On improving the performance of reliable server pooling systems for distance-sensitive distributed applications, in: Proceedings of the 15. ITG/GI Fachtagung Kommunikation in Verteilten Systemen (KiVS), Informatik aktuell, Springer, Bern/Switzerland, 2007, pp. 39–50. ISBN 978-3-540-69962-0. http://dx.doi.org/10.1007/978-3-540-69962-0_4.
[32] E. Galstad, Nagios Core Documentation, 2010.
[33] J. He, J. Rexford, Towards Internet-wide Multipath Routing, IEEE Netw. Mag. 22 (2008) 16–21, http://dx.doi.org/10.1109/MNET.2008.4476066.
[34] A. Akella, B. Maggs, S. Seshan, A. Shaikh, R. Sitaraman, On the performance benefits of multihoming route control, IEEE/ACM Trans. Network. 16 (1) (2008) 91–104, http://dx.doi.org/10.1109/TNET.2007.899068.
[35] D. Xu, M. Chiang, J. Rexford, DEFT: distributed exponentially-weighted flow splitting, in: Proceedings of the IEEE INFOCOM, Anchorage, Alaska/USA, 2007, pp. 71–79. http://dx.doi.org/10.1109/INFCOM.2007.17.
[36] A. Elwalid, C. Jin, S.H. Low, I. Widjaja, MATE: MPLS adaptive traffic engineering, in: Proceedings of the IEEE INFOCOM, Anchorage, Alaska/USA, 2001, pp. 1300–1309.
[37] S. Fischer, N. Kammenhuber, A. Feldmann, REPLEX – dynamic traffic engineering based on wardrop routing policies, in: Proceedings of the ACM CoNEXT Conference, Lisboa/Portugal, 2006. http://dx.doi.org/10.1109/NOMS.2004.1317807.
[38] A. Kvalbein, C. Dovrolis, C. Muthu, Multipath load-adaptive routing: putting the emphasis on robustness and simplicity, in: Proceedings of the 17th annual IEEE International Conference on Network Protocols (ICNP), 2009, pp. 203–212. ISBN 978-1-4244-4634-6. http://dx.doi.org/10.1109/ICNP.2009.5339682.
[39] P.D. Amer, M. Becke, T. Dreibholz, N. Ekiz, J.R. Iyengar, P. Natarajan, R.R. Stewart, M. Tüxen, Load Sharing for the Stream Control Transmission Protocol (SCTP), Internet Draft Version 06, IETF, Network Working Group, draft-tuexen-tsvwg-sctp-multipath-06.txt, work in progress, 2013.
[40] 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, 2012.
[41] A. Ford, C. Raiciu, M. Handley, S. Barré, J.R. Iyengar, Architectural Guidelines for Multipath TCP Development, Informational RFC 6182, IETF, 2011, ISSN 2070-1721.
[42] S. Barré, C. Paasch, O. Bonaventure, MultiPath TCP: from theory to practice, in: Proceedings of the 10th International IFIP Networking Conference, Valencia/Spain, 2011, pp. 444–457. ISBN 978-3-642-20756-3. http://dx.doi.org/10.1109/JPROC.2010.2093850.
[43] H. Adhari, T. Dreibholz, M. Becke, E.P. Rathgeb, M. Tüxen, Evaluation of concurrent multipath transfer over dissimilar paths, in: Proceedings of the 1st International Workshop on Protocols and Applications with Multi-Homing Support (PAMS), Singapore, 2011, pp. 708–714. ISBN 978-0-7695-4338-3. http://dx.doi.org/10.1109/WAINA.2011.92.
[44] T. Dreibholz, M. Becke, H. Adhari, E.P. Rathgeb, Evaluation of a new multipath congestion control scheme using the NetPerfMeter toolchain, in: Proceedings of the 19th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Hvar/Croatia, 2011, pp. 1–6. ISBN 978-953-290-027-9.
[45] T. Dreibholz, H. Adhari, M. Becke, 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, 2012. ISBN 978-0-7695-4652-0. http://dx.doi.org/10.1109/WAINA.2012.186.
[46] M. Becke, T. Dreibholz, H. Adhari, 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, 2012, pp. 2666–2672. http://dx.doi.org/10.1109/ICC.2012.6363695.
[47] T. Dreibholz, Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture, Ph.D. thesis, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, 2007.
[48] T. Dreibholz, E.P. Rathgeb, On the performance of reliable server pooling systems, in: Proceedings of the IEEE Conference on Local Computer Networks (LCN) 30th Anniversary, Sydney, New South Wales/Australia, 2005, pp. 200–208. ISBN 0-7695-2421-4. http://dx.doi.org/10.1109/LCN.2005.98.
[49] P. Lei, L. Ong, M. Tüxen, T. Dreibholz, An overview of reliable server pooling protocols, Informational RFC 5351, IETF, 2008. ISSN 2070-1721.
[50] T. Dreibholz, X. Zhou, M. Becke, J. Pulinthanath, E.P. Rathgeb, W. Du, On the security of reliable server pooling systems, Int. J. Intell. Inform. Database Syst. (IJIIDS) 4 (6) (2010) 552–578, http://dx.doi.org/10.1504/IJIIDS.2010.036894. ISSN 1751-5858.
[51] T. Dreibholz, M. Becke, The RSPLIB Project – From Research to Application, Demo Presentation at the IEEE Global Communications Conference (GLOBECOM), 2010.
[52] M. Becke, T. Dreibholz, A. Bayer, M. Packeiser, E.P. Rathgeb, Alternative transmission strategies for multipath transport of multimedia streams over wireless networks, in: Proceedings of the 12th IEEE International Conference on Telecommunications (ConTEL), Zagreb/Croatia, 2013, pp. 147–153. ISBN 978-953-184-175-7.
[53] T. Volkert, M. Becke, M. Osdoba, A. Mitschele-Thiel, Multipath video streaming based on hierarchical routing management, in: Proceedings of the 3nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS), Barcelona, Catalonia/Spain, 2013, pp. 1107–1112. ISBN 978-0-7695-4952-1. http://dx.doi.org/10.1109/WAINA.2013.161.
[54] T. Zinner, K. Tutschku, A. Nakao, P. Tran-Gia, Using concurrent multipath transmission for transport virtualization: analyzing path selection, in: Proceedings of the 22nd International Teletraffic Congress (ITC), Amsterdam, Noord-Holland/Netherlands, 2010, pp. 348–349. ISBN 978-1-4244-8837-7. http://dx.doi.org/10.1109/ITC.2010.5608710.
[55] T. Volkert, F. Liers, M. Becke, H. Adhari, Requirements-oriented path selection for multipath transmission, in: Proceedings of the Joint ITG and Euro-NF Workshop on Visions of Future Generation Networks (EuroView), Würzburg, Bayern/Germany, 2012.
[56] T. Ylonen, C. Lonvick, The Secure Shell (SSH) Connection Protocol, Standards Track RFC 4254, IETF, 2006. ISSN 2070-1721.
[57] K.K. Ramakrishnan, S. Floyd, D.L. Black, The Addition of Explicit Congestion Notification (ECN) to IP, Standards Track RFC 3168, IETF, 2001. ISSN 2070-1721.
[58] T. Dreibholz, S. Ferlin-Oliveira, The NorNet research testbed, in: Proceedings of the CHANGE Bootcamp Workshop, Louvain-la-Neuve/Belgium, 2013.
[59] E.K. Çetinkaya, J.P.G. Sterbenz, Programmable Networking with GpENI, Presentation, University of Kansas, Communication Networks Laboratory, 2011.

Ernst Gunnar Gran received his Candidatus Scientiarum degree in computer science from the Department of Informatics at the University of Oslo in 2007. Beside his studies, Ernst worked full time for several years as a System Administrator and as a Scientific Programmer, first at the Department of Informatics, University of Oslo, and then later at the Simula Research Laboratory. In December 2007, Ernst started his Ph.D. studies in the ICON research group of the Networks and Distributed Systems department at Simula. Today he holds a position as a Research Engineer associated with the Resilient Networks project of the NetSys department at Simula.

As part of the ICON research group, Ernst works as a researcher focusing on high performance interconnection networks in general, and congestion management in such networks in particular. As part of the Resilient Networks project, his main responsibility is the development, deployment and management of the NorNet Core Multi-Homed Research Testbed.

Thomas Dreibholz has received his Diplom (Dipl.-Inform.) degree in Computer Science from the University of Bonn in Bonn, Germany in 2001. Furthermore, he has received his Ph.D. degree (Dr. rer. nat.) in 2007 as well as his Habilitation (Priv.-Doz.) degree in 2012 from the University of Duisburg-Essen in Essen, Germany. Now, he works as a researcher on computer networks in the Network Systems Group of the Simula Research Laboratory in Fornebu, Norway.

He has published and presented more than 45 research contributions at international conferences and in journals, on the topics of Reliable Server Pooling (RSerPool), the Stream Control Transmission Protocol (SCTP) and Quality of Service (QoS). Furthermore, he has contributed multiple Working Group and Individual Submission Drafts to the IETF standardisation processes of RSerPool and SCTP. He is also co-author of multiple RFC documents published by the IETF. In addition, he has written the RSerPool reference implementation.

Amund Kvalbein is a Senior Research Scientist at Simula Research Laboratory in Oslo, Norway. He holds a Ph.D. degree from the University of Oslo (2007). After finishing his Ph.D., he spent one year as a post doc at Georgia Institute of Technology, before returning to Oslo and Simula. He is currently leader of the Resilient Networks project, focusing on methods for improving the user-experienced stability and reliability of fixed and cellular communication networks. His main research interest is in the robustness and performance of networks and networked services, with a particular focus on recovery and scalability at the routing layer.