# AUTHOR QUERY FORM

| | | |
|---|---|---|
| ELSEVIER | **Journal: COMPNW**<br><br>**Article Number: 5193** | **Please e-mail or fax your responses and any corrections to:**<br><br>**E-mail: corrections.esch@elsevier.sps.co.in**<br><br>**Fax: +31 2048 52799** |

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

| Location in article | Query / Remark: click on the Q link to go<br>Please insert your reply or correction at the corresponding line in the proof |
|---|---|
| Q1 | Please confirm that given name(s) and surname(s) have been identified correctly. |
| Q2 | The country name has been inserted for the affiliation. Please check, and correct if necessary. And also provide more details. |
| Q3 | Please check whether the designated corresponding author, the telephone number, and the e-mail address are correct, and amend if necessary. |
| Q4 | Please check the hierarchy of the section headings. |
| Q5 | Please note that Fig. 10 will appear in B/W in print and color in the web version. Based on this, please approve the footnote 12 which explains this. |
| | Please check this box if you have no corrections to make to the PDF file ☐ |

Thank you for your assistance.

Computer Networks xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

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

# The Nornet Edge platform for mobile broadband measurements ☆

Amund Kvalbein *, Džiugas Baltrūnas, Kristian Evensen, Jie Xiang, Ahmed Elmokashfi, Simone Ferlin-Oliveira

*Simula Research Laboratory, Norway*

A R T I C L E I N F O

*Article history:*
Received 2 July 2012
Received in revised form 26 November 2013
Accepted 20 December 2013
Available online xxxx

*Keywords:*
Nornet Edge
Mobile broadband
Testbed
Measurements
Multi-homing

A B S T R A C T

We present Nornet Edge (NNE), a dedicated infrastructure for measurements and experimentation in mobile broadband networks. NNE is unprecedented in size, consisting of more than 400 measurement nodes geographically distributed all over Norway. Each measurement node is a Linux-based embedded computer, and is connected to multiple mobile broadband providers. In addition, NNE includes an extensive backend system for deploying and managing experiments and collecting data. NNE makes it possible to run long-term measurement experiments to assess and compare quality and performance across different network operators on a national scale. Particular focus is put on allowing experiments to run in parallel on multiple network connections, and on collecting rich context information related to the experiments. In this paper we give a detailed presentation of NNE, and describe three different measurement experiments that illustrate how the infrastructure can be used. We also provide a roadmap for further development of NNE.



## 1. Introduction

Mobile broadband (MBB) networks are arguably becoming the most important communications infrastructure in the world. The immense popularity of mobile devices like smartphones and tablets, combined with the availability of high-capacity 3G and 4G mobile networks, has radically changed the way most people access and use the Internet. The use of MBB networks has exploded over the last few years, and global mobile traffic in 2012 was nearly 12 times the total Internet traffic in 2000 [2]. MBB traffic is estimated to keep growing at a compound annual growth rate of 66% towards 2017.

Given the importance of MBB networks, there is a strong need for objective information about their stability and performance. In particular, it is important to measure and understand the *quality as experienced by the end user*. Experienced quality is a complex notion, which manifests itself in many different performance metrics. Can a data connection be established when it is needed? Can a data connection be maintained without disruption for a long time? What is the capacity a user can expect? And what latency? Can the network support a VOIP session? Is it possible to stream a football match without interruptions?

Such performance information is sought by many parties. Regulators need it to control how operators fulfill their obligations, and as a basis for regulatory policies. Operators can augment their monitoring data with user-side measurements, in order to identify problems in their networks and reduce customer complaints. Organizations and businesses can use such information to build robust services and products on top of MBB networks. Application developers need to know the characteristics of the underlying network in order to build efficient applications. And finally,



* Corresponding author. Tel.: +47 95779674.
*E-mail address:* amundk@simula.no (A. Kvalbein).

1389-1286/$ - see front matter © 2013 The Authors. Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bjp.2013.12.036

consumers can use such information to make informed choices on which network provider to choose. There is, however, very little information publicly available today about the performance of MBB networks. For consumers, the only source of information is typically coverage maps based mostly on theoretical calculations, and reports on the providers' web pages when there are particular problems. Even regulators are often left with a posteriori event reports from the operators after large failure events, but lack a more detailed view of the stability of MBB networks.

The quality experienced by end users can only be assessed through systematic end-to-end measurements. Monitoring from within the network can reveal parts of the picture, but can hardly capture the complex interplay between user equipment, network, protocols and applications. A working base station is not the same as a stable Internet connection, good signal strength is not the same as high throughput, light system load is not the same as stable delay and so on.

There are several possible approaches to performing systematic measurements of MBB performance. Operators or independent agencies sometimes perform drive-by tests to identify coverage holes or performance problems. These tests are, however, expensive and do not scale well. Another approach is to rely on end users to run performance tests by visiting a website or running a special measurement application. The main advantage of this approach is scalability: it can collect millions of measurements from different regions, networks and user equipment. There are, however, several challenges. One challenge is privacy: data collected from real users must be carefully anonymized to avoid revealing private information. Another challenge is that users perform measurements at their own will, which might give a bias towards situations when the user experiences performance problems. It is also difficult to collect data on stability and availability with user-initiated measurements, since this typically requires long and uninterrupted measurement sessions. Finally, the context information around this type of measurements is often lacking. Location, type of user equipment, type of subscription, and connection mode (2G/3G/4G) are examples of useful information when analyzing the results.

The approach presented in this paper is based on building a dedicated infrastructure for measuring and experimenting in MBB networks. A dedicated measurement infrastructure can complement other approaches by overcoming many of the challenges listed above. It can be used to run measurement experiments at regular intervals over long time periods under similar conditions. It also gives full information about the context in which the measurements are collected, and allows targeted experiments triggered by the observed behavior.

This paper describes Nornet Edge (NNE), a dedicated infrastructure for measurements and experimentation in MBB networks. NNE is built in the context of the Nornet project, partly funded by the Research Council of Norway.[1] The Nornet project also builds the complementary test-bed Nornet Core [7], a Future Internet research infrastructure with multi-homed sites. The main features of NNE are:

*Unprecedented scale.*
NNE currently consists of more than 400 measurement nodes. The large number of nodes makes it possible to give a representative view of the characteristics of an entire network.

*Nationwide geographical coverage.*
The measurement nodes are widely distributed across Norway (including one node at Svalbard). The challenging Norwegian topography makes it possible to collect measurements under diverse conditions, from major cities to remote islands. At the same time, there is a dense deployment of nodes in a few main cities, giving a more detailed view of network conditions in urban areas.

*Fully programmable measurement nodes.*
The measurement nodes are embedded computers running a standard Linux distribution. They are flexible and powerful enough to run most measurement tasks, including video experiments.

*Multi-homed measurement nodes.*
All NNE nodes are connected to at least two MBB providers, and often also to fixed or wireless networks. This makes NNE particularly well suited for experimentation with methods that exploit multiple communication links.

*Rich context information.*
In addition to information about network, time and location for a measurement experiment, NNE nodes have built-in support for collecting information such as cell ID, signal strength, and connection mode.

*Advanced system for experiment management.*
NNE makes it easy to deploy experiments on all or a selected subset of the nodes. NNE also supports transferring and storing measurement results in a central sever with minimal effort from the experiment developer.

This paper shows some of the capabilities of NNE through three measurement experiments. The experiments reveal important characteristics of the investigated MBB networks, by measuring their Radio Resource Control (RRC) state machine, the stability of connections, and the relationship between download speed and connection parameters.

The rest of this paper is structured as follows. Section 2 gives an overview of NNE and the current deployment. Section 3 discusses related work in the area of MBB measurements. Section 4 introduces the measurement node used in NNE, including the hardware configuration and software tools. Section 5 describes the backend part of the NNE infrastructure. Section 6 describes the three different experiments that illustrate some of the capabilities of NNE. Section 7 describes plans for the future evolution of NNE. Finally, Section 8 summarizes and concludes the paper.

[1] Project number 208798/F50.

## 2. System overview

The main goal of NNE is to measure end-to-end performance as seen by the end user. This performance is influenced by properties of the operating system, the MBB modems, the Radio Access Network (RAN), the underlying transport network, and the mobile core network. NNE allows direct comparison of different MBB networks, by using the same hardware and operating system for all providers.

Fig. 1 shows a schematic overview of the NNE infrastructure. It consists of two main components: a large set of NNE nodes, and a central backend system. The NNE nodes run a standard Linux distribution (currently Debian Wheezy with a 3.0.8 kernel), giving great flexibility in the types of measurements that can be supported. The NNE nodes are described in detail in Section 4. The backend system consists of a number of servers for monitoring and controlling the nodes, deploying and managing measurement experiments, and processing, storing and visualizing measurement data. The backend contains an SSH proxy server that enables remote login to NNE nodes. The backend system is described in detail in Section 5.

Measurement data often becomes much more valuable when combined with additional information about the context in which they were collected. For example, it is often interesting to know the cell ID of the base station that the node is connected to, the signal strength, the mode of the connection (e.g., 2G, 3G or 4G), the submode of the connection (e.g, GPRS, EDGE, WCDMA or HSPA+), the state of the Radio Resource Controller (RRC) on the node, etc. NNE nodes contain a built-in tool that collects this information from the MBB modems, and makes it available for measurement applications in a publish-subscribe fashion.

### 2.1. Deployment of nodes

NNE nodes are deployed in collaboration with Norwegian communes and the Ministry for Local Government and Regional affairs. Their interest in NNE is related to electronic voting. Voter registration is done electronically, and all voting centrals must have a working Internet connection. NNE nodes are used to monitor the existing Internet connection in these voting locations, and to provide a robust backup connection in case the primary connection fails. Hence, most NNE nodes are placed in voting locations in Norwegian communes. Voting locations are often schools, city halls or other community houses. 5 large (by Norwegian standards) cities have NNE nodes placed in all voting locations. In total 289 of our 443 nodes are placed in these 5 cities, with 136 of these in the largest city Oslo. 127 of the remaining nodes are spread across 85 other communes, with 1–6 nodes per commune. Fig. 2 shows an overview of the geographical location of nodes.

The collaboration with the communes has proven to be a good way to manage the physical distribution of NNE nodes. It gives a good geographical spread of the nodes, which is representative for the population in Norway. Local IT support at the communes is responsible for installing the nodes, and also helps with on-site management such as physically rebooting a node or replacing faulty equipment when needed. Most management of NNE nodes are done remotely (see Section 5), but occasionally there is need for physical intervention, e.g., when contact with a node is lost.

### 2.2. Measured MBB networks

A particular focus in NNE is to support multi-homing, enabling multiple connections to be used in parallel. Each NNE node is connected to 2–5 MBB providers, using standard subscriptions. In addition, the nodes have a fixed Internet connection when this is possible. Multihoming opens up many opportunities for novel types of experiments. At the infrastructure layer, NNE can be used to discover correlations in the performance characteristics of the different MBB networks. Such correlations are important for applications that rely on multi-homing for increased robustness or performance. At the transport layer, NNE is well suited for experimentation with novel protocols such as SCTP [18] and MPTCP [5] which allow concurrent transmission over multiple paths.

There are 4 MBB operators in Norway with their own RAN. Telenor and Netcom operate nationwide 2G/3G/4G networks in the GSM family. Tele2 is building the third 2G/3G GSM network, which will soon also have nationwide coverage. For historical reasons, there are 2 separate networks operating in Tele2's RAN (Tele2 and Network Norway). When customers of these operators are outside the coverage area for Tele 2's network, they do national roaming to Netcom and Telenor respectively. This arrangement

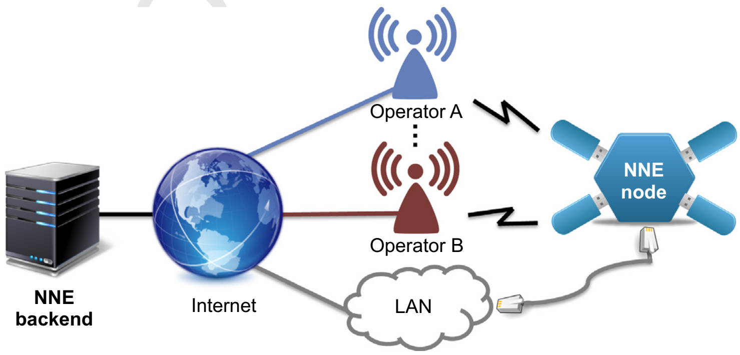



**Fig. 1.** NNE overview.

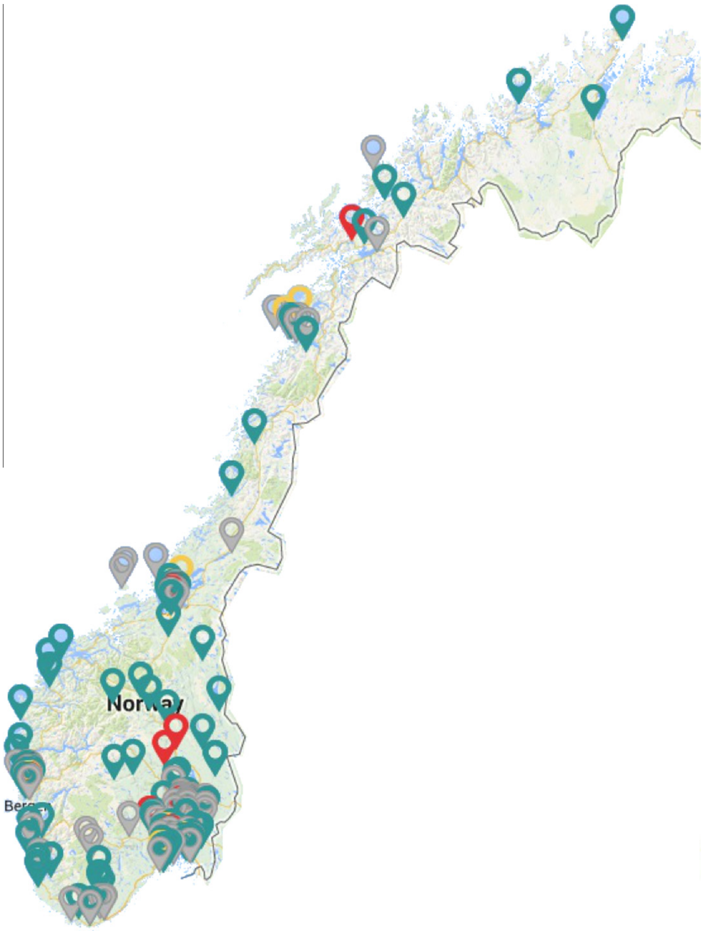


**Fig. 2.** Geographical distribution of NNE nodes.

opens up interesting possibilities for comparing performance between connections that are on the same RAN but with different core networks, or connections that are on the different RANs but uses the same core network. The fourth operator, Ice, runs a data-only CDMA network in the 450 MHz frequency band. This network is qualitatively different than the GSM-based networks, since it uses a different technology, operates on a much lower frequency, and uses different end user equipment.

All 4 network operators were invited to take part in the establishment of NNE, and 2 of them entered into a collaboration on building NNE. These 2 operators (Tele2 and Ice) offer free MBB subscriptions to NNE nodes. For the remaining 2 operators, we buy subscriptions. The operators' main motivation for taking part in NNE is access to real-time data on the current status of all MBB networks as seen by the end users (see Section 5 for a description of how such data is presented). This information is seen as useful by the operators to complement their own network monitoring.

## 3. Related work

Active measurements in MBB networks can be performed either from dedicated nodes, or by instrumenting smartphones or other mobile devices to run measurements. An approach in the first category is WiScape [16]. This test-bed consists of a handful of nodes mounted on public buses, each equipped with MBB connections from multiple operators. The infrastructure is used to characterize and compare the performance of these networks in a metropolitan area. Compared to WiScape, NNE has a significantly larger scale, uses stationary nodes, and uses identical modems for all connections of the same type.

Another example of MBB measurements with a dedicated infrastructure, is the measurements undertaken by the telecommunications authorities in Lithuania [3]. They perform regular performance measurements of all MBB operators, using test nodes that are rotated between post-offices across the country. Several performance metrics are collected, including connection availability, HTTP GET, and FTP throughput.

A different approach to active MBB measurements is to crowd-source results from a large number of MBB users. By distributing dedicated measurement software in the form of easily installable apps, researchers can harvest millions of performance tests from a large user base. A prominent example in this category is Mobiperf [1], which can run a range of tests related to DNS, HTTP and TCP performance among others. Approaches based on user-initiated tests

Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036

can complement measurements from dedicated infrastructures such as NNE. It is difficult, however, to rely on such tests for continuous monitoring of networks for assessing their stability.

A related approach is to piggyback on popular applications for gathering data on various MBB properties. This approach has been used for geo-locating IP addresses in MBB networks [20].

MBB performance can also be evaluated through network-side monitoring and logging, and such data has been exploited in several measurement studies (e.g., [8,17]). Network-side logs can give a very detailed insight into the causes of observed behavior. A main challenge is, of course, that such data is generally only available to operators.

A related approach is Meddle [14], which proposes to use a combination of middle boxes and VPNs for measuring mobile traffic. Users are envisioned to subscribe to services like ad-blocking and hence allow their traffic to go through Meddle infrastructure. Meddle is an interesting approach; however, it would potentially face a myriad of privacy challenges. It does not accommodate active measurements.

MITATE [6] was recently proposed as a collaborative platform for mobile network experimentation. MITATE depends on leveraging participants devices for running experiments requested by other participants in a tit for tat fashion.

Dedicated infrastructures for monitoring network performance has recently gained popularity in residential broadband evaluations. Telecom regulators in many regions of the world deploy test nodes from SamKnows[2] for measuring the quality of home broadband. The main goal of these tests is to see if consumers receive the capacity that they pay for, but the tests also collect several other performance and functionality metrics [15]. Dedicated measurement nodes are also used in the Bismark [19] project, which looks at a wide range of metrics for evaluating home broadband performance.

## 4. The Nornet Edge node

This section discusses the requirements that led to the selection of the current hardware platform, and gives a description of the hardware and software functions that the node offers to support experiments.

NNE nodes are small embedded computers running a standard Linux distribution, and the software and tools that run on the nodes are all based on standard Linux libraries. Any computer that can run Linux can therefore in theory be used as an NNE node. We have, however, chosen to standardize on a single hardware configuration for our nodes. This makes management and administration of the nodes easier. It is not unlikely that the situation with a single hardware platform will change in the future, when new generations of NNE nodes will be deployed.

Several requirements were taken into consideration when selecting the hardware and software configuration for NNE nodes, including price, performance, availability and user community. Based on these requirements, several different single-board computers were tested, including the BeagleBoard,[3] the BeagleBone,[4] the PandaBoard[5] and the Raspberry Pi.[6] After extensive testing, it was found that none of these offer the stability and performance required. With most of these boards, the USB controller is not able to cope with the number of USB modems connected to an NNE node. In addition, none of the tested devices had a sufficient number of built-in USB ports. This required the use of an external USB hub, with its own power supply in order to give enough power to the connected USB modems.

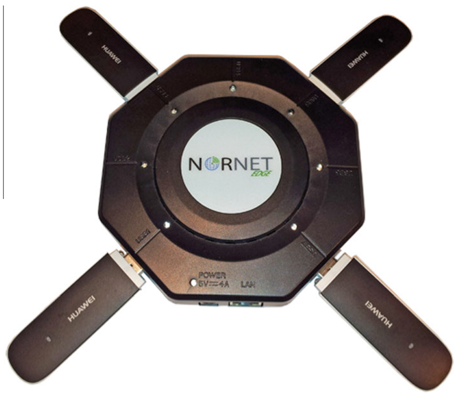



**Fig. 3.** NNE node with 4 USB modems.

Based on this experience, it was decided to develop a new single-board computer, specifically made to be a hardware platform for NNE. This may sound like a radical decision, but it was deemed necessary in order to have a sufficiently stable node. Hence, the *UFO-board* (named after its shape) was designed and produced in collaboration with Dynsense,[7] a Hong Kong based hardware development company.

The main drawback of designing a board specially for this project is the lack of an existing user community. Note, however, that most of the components on the UFO-board (CPU, USB controller, Ethernet port, etc.) are off-the-shelf components that are supported by the mainline Linux kernel. Only a small patch is required to make the UFO-board run. The advantage of having a custom designed board is the ability to select components that fit our needs. This includes a CPU that is powerful enough to support multiple parallel experiments on different network interfaces, enough USB ports to support a large number of MBB modems and other peripherals, and a USB controller and power supply that can handle many connected devices. So far, the UFO-board has proven to be well suited for its purpose, and the stability has been far better than for any other tested board.

2 http://www.samknows.com/.

3 http://beagleboard.org/.
4 http://beagleboard.org/bone.
5 http://pandaboard.org/.
6 http://www.raspberrypi.org/.
7 http://dynsense.com/.

### 4.1. Technical specifications

Fig. 3 shows a picture of an NNE node. In its current configuration, it consists of the following components:

#### 4.1.1. UFO-board single-board computer

The UFO-board has a Samsung S5PV210 Cortex A8 1 GHz processor. It has 512 MB RAM and 512 MB NAND Flash memory, and a 16 GB SD card for storage. 7 USB ports and 1 Fast Ethernet port is embedded in the device.

#### 4.1.2. Linux operating system

The UFO-board runs a standard Linux distribution, currently with a 3.0.8 kernel, with a planned upgrade to 3.8.8. The root file system includes standard Debian Wheezy components as well as the software described below. The image including the root file system is written to the SD card during the assembly process. During this process, each UFO-board is also given a unique node ID.

#### 4.1.3. 1–4 3G (UMTS) modems

Huawei E353-u2 3G USB modem is used for all UMTS operators. This modem supports GSM technologies up to HSPA+ ("3.5G"). A main motivation for selecting this model is that it is sold to end-users by some of the operators that the nodes connect to, which ensures that our measurements are comparable to a typical user experience. Another advantage of this modem is that it reports about events such as change of network connection mode and submode, cell ID, signal strength, and RRC state.

#### 4.1.4. 1 CDMA (1× EV-DO) modem

This modem is used to connect to Ice, which operates a MBB (data only) network in the 450 MHz frequency band. Two different modem models are used, supporting the Rev. A and Rev. B versions of the 1× EV-DO standard respectively.

#### 4.1.5. Wi-Fi module

Some NNE nodes are equipped with a TP-LINK TL-WN822N USB Wi-Fi module. This module serves two purposes. First, it allows the NNE node to connect to an available Wi-Fi network. Second, it can be used to turn the NNE node into a Wi-Fi access point, providing Internet access through the MBB connections.

### 4.2. Management and support functions

NNE nodes contain a set of tools and services that are useful for many measurement experiments. These support functions give experiment developers easy access to information about the state of the MBB connections, and perform standard operations like uploading measurement results. Fig. 4 shows a schematic overview of the most central components running on an NNE node.

#### 4.2.1. SSH connection

NNE nodes can be reached remotely via SSH. Since most nodes do not have a fixed IP address, the SSH connection is initiated from the node. This is done using `autossh`, which automatically (re-) establishes a reverse SSH connection to our server on a port deduced from the unique node ID.

The SSH connection will normally be established over the default route on the node (i.e., the route with the lowest metric). The default route is set individually on each node, based on the observed quality of the different connections (the fixed LAN connection is always preferred if it is present). `autossh` will retry a limited number of times to re-establish the connection over the default route if it goes down. If not successful, it will give up and exit. The `autossh` process will then be restarted and will establish the connection over the next available connection with the lowest metric.

#### 4.2.2. Usbmodem-listener

The `usbmodem-listener` is a daemon written in Python that incorporates three main functions: modem management, cellular connection management and metadata collection.

The MBB modems used in NNE nodes are mostly USB based. These modems expose multiple serial devices to the operating system. Before the modem can be used for communication, a series of AT-commands must be sent on the correct serial port in order to identify the SIM card and establish the connection to the network. `usbmodem-listener` listens to `udev` events from the Linux kernel, and performs the necessary configuration as soon as a new USB modem is plugged in. It spawns the *pppd* process that creates the PPP connection to the modem. `usbmodem-listener` keeps monitoring the state of the forked *pppd* process and restarts it when it exists or becomes stale. It also makes sure that all occupied resources are freed when the modem is disconnected.

Finally, `usbmodem-listener` collects metadata information about each active network connection and publishes it to subscribers. NNE nodes use Huawei modems with a Qualcomm chipset. It is possible to extract a large number of status variables from these modems, using AT commands and special tools developed for Qualcomm-based modems [13,10]. Examples of collected information is the cell ID, signal strength, connection mode and submode and various radio resource state parameters. Access to this information is limited to a single process at a time. The `usbmodem-listener` therefore acts as a publisher of the status variables it receives from the modems, and makes it available to local or remote subscribers through a socket interface.

#### 4.2.3. MULTI

NNE nodes use MULTI [4] to manage the network connections. MULTI is a command line network manager for Linux with support for multihoming. The currently available network managers for Linux do not properly configure routing tables with multiple active interfaces. Without special consideration, multiple interfaces cannot be used in parallel.

MULTI automatically detects when interfaces are connected/removed, and then obtains an IP address (if needed) and configures the routing tables correctly. After routes have been added or removed, MULTI broadcasts a notification to a MULTI Netlink group. Netlink is a socket-based,

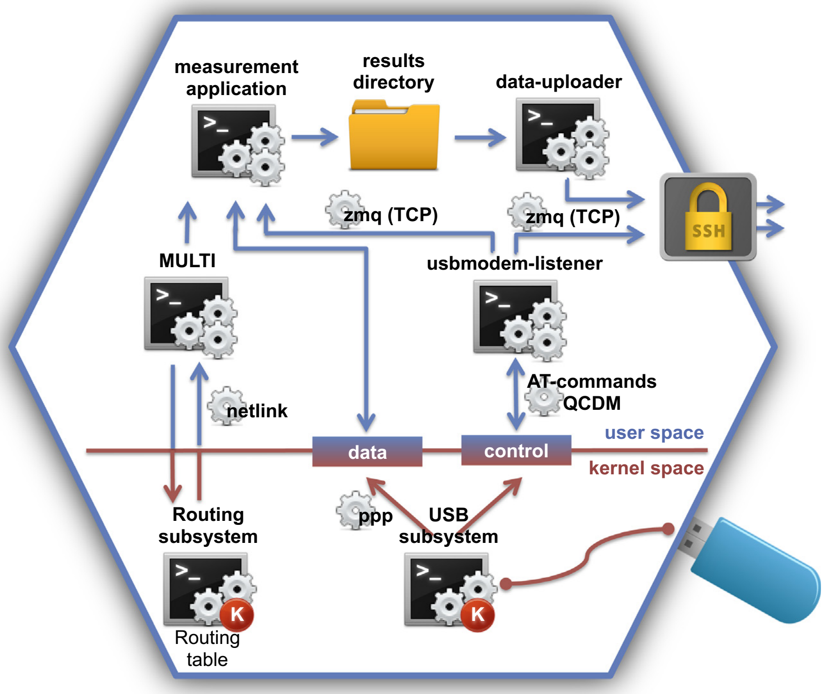



**Fig. 4.** The most important functions on the NNE node.

IPC-mechanism for Linux. Applications can subscribe to these messages and adapt based on changes in network state. For example, a measurement application can be configured to automatically start measuring on interfaces when they become available.

#### 4.2.4. Data-uploader

NNE offers functionality to ease the process of uploading measurement data to a server. Measurement applications write their output in a predefined format to a special output directory on the local file system. The `data-uploader` will periodically scan this directory and transfer log files to a corresponding directory at the server-side data collector described in Section 5. This approach makes it easier to make semi-realtime visualization of measurement results, without having to write special upload and encoding/decoding procedures and importing data into a database in each measurement experiment.

## 5. Backend system

The NNE backend system consists of a number of servers that collect, store, process and visualize measurement data, and perform various monitoring and control tasks for the NNE nodes. The NNE backend is based on a VMWare cluster, with an attached RAID shelf for storage. Fig. 5 shows an overview of the NNE backend with its various functions.

### 5.1. Node control and management

#### 5.1.1. SSH proxy

Each NNE node maintains an SSH connection to a server acting as an SSH proxy, which makes it possible to access the nodes remotely. The SSH proxy accommodates both local and remote port forwarding to the nodes. Local port forwarding is used on a node if a restrictive firewall policy is enforced by a LAN connection, so that the node cannot reach the backend servers directly. Remote port forwarding is used so in order to make the SSH server and the updates produced by the `usbmodem-listener` on the NNE nodes accessible remotely. The SSH proxy itself is secured, and remotely forwarded port ranges are limited only to the NNE servers that need to reach the nodes.

#### 5.1.2. Puppet for node management

Puppet[8] is used for managing and updating NNE nodes. The main purpose of Puppet is to push changes and updates in measurement or management software to NNE nodes automatically. The Puppet infrastructure consists of Puppet agents running on each node, that connect to a central Puppet master. NNE nodes are preconfigured to start the Puppet agent on boot, which then connects to the Puppet master. The master compiles the configuration for the node based on manifest files, and sends it to the agent. The agent in turn checks the received configuration against the current system state, and takes the necessary action by creating or updating

[8] https://puppetlabs.com/.

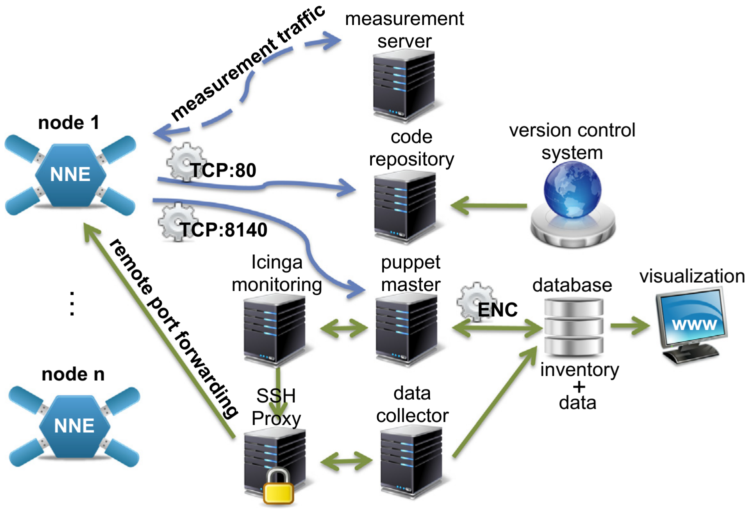



**Fig. 5.** NNE backend.

resources like files, Debian packages or configurations. It then reports the results back to the master and repeats the whole procedure every hour.

*5.1.3. Icinga for monitoring*

Icinga,[9] a branch of the well-known Nagios monitoring system, is used to perform periodic availability and service checking of NNE nodes. Each check is done by connecting to the node over SSH via the SSH proxy server, and reporting the status of parameters like disk space, processor load, etc.

*5.1.4. Build server and package repository*

NNE has a simple process for deploying and updating measurement applications and other software. When a new version of an application is ready, it is pulled from the version control system and placed on the package repository server. Most measurement applications and other software that is installed on NNE nodes are packaged as Debian packages for ease of management, but they can also be distributed as tar balls, etc. Once a new package (or version) is available in the code repository, the Puppet master will instruct the relevant nodes to download and install the update.

*5.1.5. Measurement server*

Some measurement experiments require active participation from the server side. The NNE backend contains a measurement server that can be used as an endpoint for these experiments. Experiments can also be run agains any other server on the Internet.

*5.1.6. Data collector*

Measurement data collected at the nodes is transferred to the backend by the `data-uploader` described in the previous section. At the backend, the files are received and processed by the data collector, which inserts data into a relational database. The data collector can insert the data directly, or perform some processing before the data is inserted, like calculating averages or extracting selected values.

*5.2. Database*

The database is a central component in the NNE backend, that serves two main purposes. First, it keeps records for the NNE inventory, consisting of nodes and their associated components like modems, subscriptions and other hardware. For each node, there is also information about location, address, contact person, administrative messages, etc. Second, the database is used to store data from the measurement experiments. By using a single database to store both inventory information and measurement result, it becomes easy to extract measurement results from a particular node, and to compare results across nodes.

Measurement data is usually stored in a generic database table, which can be used for any type of measurement. The table contains a timestamp, pointer to the measurement instance (e.g., the node, network and type of measurement), as well as the output from the measurement in an XML format. The format of the measurement data is specified in a separate table, and can for example be an RTT value for a single data packet, the setup time for a TCP connection, or the result from a throughput test. The use of a generic measurement data table makes it easier to deploy new types of measurement experiments.

With a large number of nodes running continuous measurements on up to 5 MBB interfaces in parallel, the database must be organized in a scalable way. After considering several options for scalable data storage, such as Apache HBase, we finally decided to use MySQL and stay with a relational database model. This decision was made because we have a fixed scale deployment, where the number of nodes, networks and approximate number of daily database records is known in advance and will not grow uncontrolled. To improve scalability, the dataset is split into smaller chunks using the MySQL partitioning feature. Daily

[9] https://www.icinga.org.



Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036

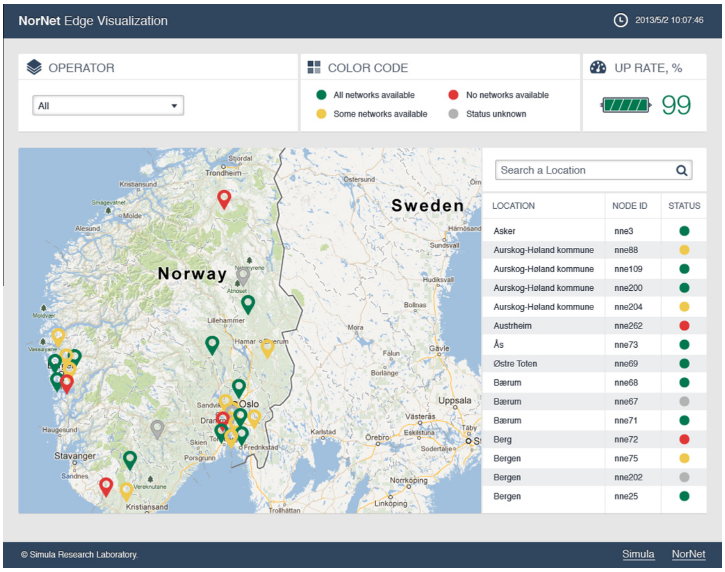



**Fig. 6.** Screenshot from the visualization website.

partitions are created for the tables that contain large amounts of measurement results. Partitioning also allows to fetch a range of data fast, since it only involves scanning partitions that fall into a given interval of dates, so for example in order to select data for past 15 min, a scan of only one partition is made.

### 5.3. Visualization

The status of NNE nodes and results from selected measurement experiments are visualized on a website associated with the Nornet project.[10] The website shows a real-time view of the status of all NNE nodes, including the status of each MBB connection. The status of the different connections are visualized as color codes on a map, and status can be shown for individual operators or all operators combined. A screenshot from the preliminary website is shown in Fig. 6. The status information is collected by the `usbmodem-listener` as described in the previous section, and status updates are published to a status collector running in the backend in real time, so that the website is updated within a few seconds.

By clicking on the nodes in the map, more detailed information about the current status of each connection is shown, as well as graphical representations of various measurement results. Results from some measurement applications are also shown in near-realtime, with a delay of approximately 1 min. The liveness of such data is determined by how frequently the application writes its output to the results directory on the node, and how frequently the `data-uploader` scans this folder and transfers the data to the server. In addition to (near-) realtime data, the website also shows historical data on how different metrics have evolved over time at the different nodes.

The visualization website serves several purposes. First, it is an efficient way for researchers to navigate in the collected data. Second, the website serves as a point of entry for media and the general public who are interested in assessing and comparing the performance of the different MBB operators, and in seeing data on how various failures and incidents in the network affect performance. Last, the website serves as an easy way for our collaborators (operators and node hosts) to get a real-time overview of the status of their network and measurement nodes. The operators use a tailored version of this website to get a view of the customer-experienced performance of their networks, to complement their own network monitoring.

## 6. Example measurement experiments

NNE is built as a flexible platform for measurements and experimentation in MBB networks. NNE can be used to evaluate both properties of the network itself and the interaction between the network and transport/application layer protocols. In particular, NNE is well suited for experiments that require a large number of geographically distributed measurement nodes, simultaneous connections to multiple operators, information about the context in which measurements are taken, and/or long-running continuous measurements. Examples of such experiments can be characterization of how network performance evolve over time, robustness characterization, comparison between different operators, etc.

In this section, we present three examples of measurement experiments that in different ways exploit the capabilities of NNE. The purpose of this section is not to give in-depth analysis of particular measurement results. Rather, we seek to highlight some of the possibilities that NNE gives for augmenting data plane measurements with various metadata and information from the network and MBB modem, and to point out some directions for measurement

[10] http://robustenett.no.



Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036

studies that can be done on this infrastructure. We focus in this section only on the UMTS MBB connections, and disregard the CDMA operator.

### 6.1. RRC state experiment

In the first example, we exploit the Qualcomm diagnostic port that is available on the USB modems to investigate an important property of the different radio networks that our nodes are attached to, namely the Radio Resource Control (RRC) state machine. In NNE, information about the RRC state is available to measurement applications by subscribing to updates from the `usbmodem-listener` as described in Section 4.

Before a modem can send data, the network must first allocate a RRC state to the connection [9]. The RRC state machine is used by the network to ensure that radio resources are managed efficiently. Mobile units that send or receive much data are promoted to a high RRC state (Cell_DCH). In this state the unit can send data at the highest rate, but also consumes the highest amount of power and spectrum resources. Units with less or no data to send are kept at a lower RRC state (Cell_FACH, Cell_PCH, URA_PCH or IDLE) where they use less power and radio resources. A promotion to a higher RRC state is triggered by sending or receiving data, while the demotion to the lower states is controlled by different timers.

Different networks implement and configure the RRC state machine differently in terms of which states are used, the data rate that is required to achieve/sustain a high RRC state, and the timers that are used to fall back to a lower RRC state. These factors have a direct impact on both the performance and energy efficiency experienced in the network. Several previous studies have looked at how the RRC state parameters can be inferred by active measurements [11], how they can be optimized to save energy [12], and how they influence network performance [8].

The information available from the `usbmodem-listener` makes it very easy to write measurement applications that react to changes in RRC states, without the need to look at indirect indications such as changes in packet delays. Here, we use a very simple test to infer the RRC state machine parameters of the 3 UMTS Radio Access Networks in Norway. We are interested in the time it takes to acquire the Cell_DCH state, and the timers that are used before reverting to Cell_FACH and then Cell_PCH or IDLE. Two of the three networks tested implement either Cell_PCH or URA_PCH state, while the third does not and instead goes to IDLE.

In most networks (including those under test here), sending a single large data packet is enough to trigger promotion to Cell_DCH. Hence, the parameters we are interested in can be collected with the following very simple algorithm (starting from IDLE state): (1) Send a 1 KB data packet once per second until `usbmodem-listener` reports that the connection has reached Cell_DCH, then stop sending immediately. This data rate is enough to go beyond the limits of the Cell_FACH channel. (2) Wait until `usbmodem-listener` reports that the connection has been demoted to Cell_FACH. (3) Keep waiting until `usbmodem-listener` reports that the connection has been demoted

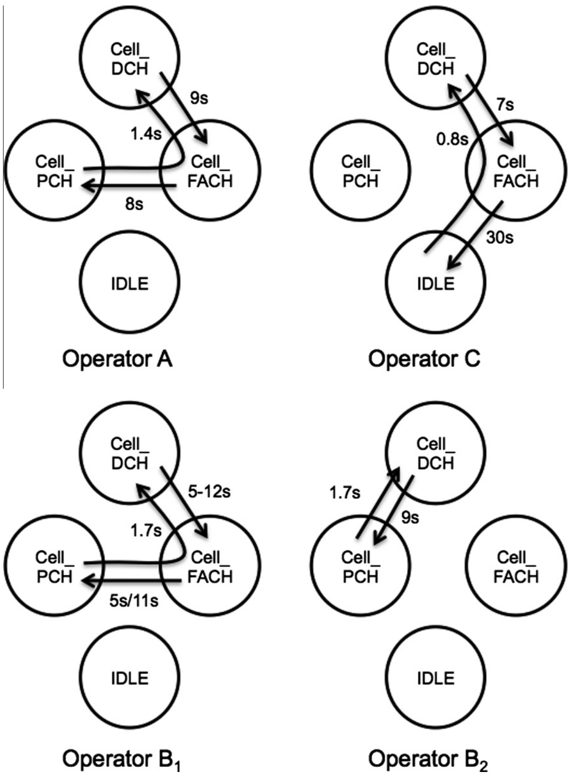



**Fig. 7.** RRC state machines.

to Cell_PCH or IDLE. We record the time used in each step, and repeat 100 times at 5 different nodes for each network.

Fig. 7 shows the states used in each network we measured. The annotated arrows show state transitions and the time it takes to get promoted/demoted. The values shown are averages over all runs. The RRC state machine parameters are normally the same throughout a network, so it is sufficient to measure them at one or a few locations. In our case, operator B uses different parameters at different locations in their network. The state machine for operator B is therefore shown twice. In most areas, this operator uses three RRC states (Cell_DCH, Cell_FACH and Cell_PCH), although with slightly different timeout values for state demotions in different regions. In some areas, however, they use a very different state machine implementation, using only two RRC states (Cell_DCH and Cell_PCH). It is known that operator B buys equipment from two different vendors in different parts of their network, and we believe this can be why they use different parameters at different locations. There may also be operational reasons for having different RRC state machines, based on the differences in traffic patterns at different locations.

When discussing these figures with one of the operators, they informed us that the configured timer they use to demote a connection from Cell_DCH is 2s lower than what is reported here, which indicates that there may be some delay before the demotion is announced by the modem.

The most interesting observation from this experiment is the time it takes from the first packet is sent until the Cell_DCH state is achieved. This time will directly add to the delay experienced by a user when she first opens a con-



Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036

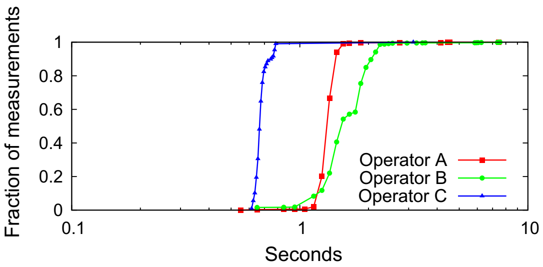



**Fig. 8.** CDF of Cell_DCH acquisition times for each network.

nection. The values shown in Fig. 7 are average values. We did, however, sometimes experience values that were significantly higher, as shown in the CDF in Fig. 8. It is also interesting to observe the low Cell_DCH acquisition times measured for operator C. The promotion from IDLE to Cell_DCH requires more signaling in the network than the corresponding promotion from Cell_PCH in the other networks, so this is surprising. We do not have an explanation for this behavior at this time.

### 6.2. Connection status

MULTI and `usbmodem-listener` described in Section 4 log the status of each connected USB modem. Recall that MULTI reports whether the USB modem is available (Up) or unavailable (Down) as seen from the operating system, while the `usbmodem-listener` reports the mode (2G/3G/4G) of the connection (amongst other parameters). Knowing the state and mode of a connection is important in order to give a correct interpretation of measurement results. The logs can also be used independently or in concert with active measurements to say something about the stability of each MBB connection and operator. Here, we use the logs to classify the state of an MBB connection into 5 different states, and discuss the stability of the connections using this classification.

The **Down** state indicates there is no active PPP connection to the modem. When an MBB modem looses the connection to the network, the `usbmodem-listener` will try to establish a new session. The Down state is normally a result of such connection losses, and is seen for short periods before a new session can be established.

In the **No Service** state, the PPP connection with the modem is established, but no IP address has been assigned and no connection to the network has been established.

The **Attached** state means that the PPP connection with the modem is established, but the modem did not report the mode/submode of the connection. The status in this state is therefore unknown, and data-plane information or network side logs are needed in order to determine whether there is a working connection.

The **2G** and **3G** states are self-explained, and indicate that the connection is working as it should.

Fig. 9 shows the fraction of time spent in different states for a large number of connections in two different operators (note the logarithmic scale on the y-axis). The plots are based on 90 connections from operator A and 47 connections for operator B. For each operator, the connections are sorted based on the fraction of time they are unavail-

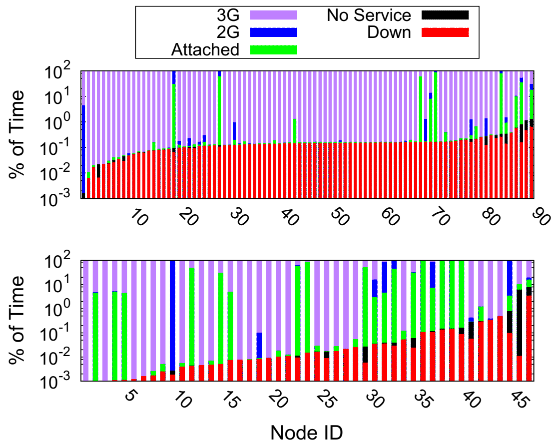



**Fig. 9.** Status for connections in operator A and operator B.

able (Down + No Service states). This way of plotting the data allows us to look at the general trend in the availability for each operator, and to highlight differences between operators. The measurements were collected over 4 weeks in April 2013. The measurement setup (location,[11] modem model, measurement period, software, etc.) is identical for both operators, so it is reasonable to assume that any observed differences are caused by the network.

We can make several interesting observations from the plots. First, we note that the time spent in the Down state is relatively stable across both operators and connections. Averaging across all connections, operator A is Down 0.14% of the time, while the corresponding figure for operator B is 0.12%. If we also include the time spent in the No Service state, the figures are 0.17% and 0.35% respectively. There is, in other words, a clear difference in the time spent in the No Service state between the two operators.

There is also a clear difference between the operators regarding the time spent in the Attached state. While connections in operator A spends on average 4.3% of their time in this state, the number for operator B is 16.5%. There are, however, large differences between individual connections at both operators, as can be seen from the plots. A closer look at the data indicates that connections that spend much time in this state often experience poor signal strength and frequent reconnects. This is, however, not always the case, and cannot explain the full difference. Active data plane measurements and/or network side logs are needed in order to tell the actual operational state of the connections.

We have also looked at the time spent in different states for the other 2 UMTS operators that our nodes are connected to. Also for these operators, we see clear distinctions. Both of these operators spend less time in the Down state (0.02% and 0.06% respectively), while the time spent in the Attached state is 0.1% and 14.5% respectively.

The observations in this subsection show that there are clear structural differences in the results between operators. Using an identical setup for all operators allows us

[11] We have more observation points for operator A then for operator B. The locations for operator B are a subset of the locations for operator A.



Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036

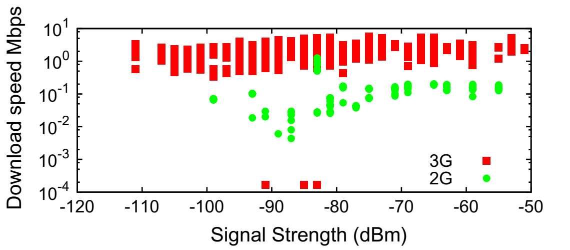



**Fig. 10.** Download speed vs signal strength. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

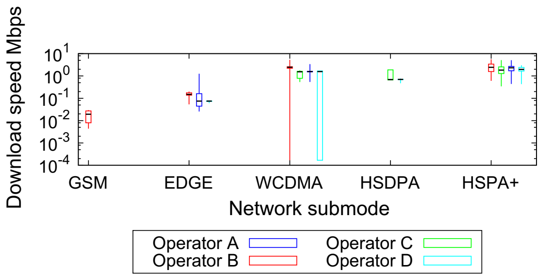



**Fig. 11.** Download speed vs connection mode.

to compare their stability and performance directly. In future studies, we plan to perform several large measurement studies that directly compare and correlate between operators, and also to work with some of the operators in order to explain the observed behavior. Such studies will require strict control of the hardware setup, rich metadata from the measurement nodes, and flexibility in how measurements are implemented.

### 6.3. Download time analysis

As a third and final example, we illustrate how the metadata collected on the NNE nodes allows the analysis of measurement results from multiple angles. We perform a very simple experiment, where we record the time it takes to download a 1 MB file over a MBB connection using `curl`. Note that this experiment is not an attempt to systematically measure the achievable throughput in MBB networks, but rather serves to illustrate the importance of collecting context information.

The file download is performed over 148 different connections, distributed over 77 locations and 4 different GSM operators. For each connection, we repeat the download 10 times, so in total we have 1480 download samples. At the end of each download, we record the signal strength and the mode/submode of the connection.

The scatterplot in Fig. 10 shows the relation between the measured signal strength and the achieved download speed. We distinguish in the plot between downloads over 2G connections (green)[12] and 3G connections (red). Not surprisingly, there is a clear difference in download speed depending on the connection mode. The figure also shows that the signal strength has a very limited effect on the achieved download speed, in particular for 3G connections. Download speed in 3G networks is largely determined by the downlink transmission rate set by the Radio Network Controller. This figure shows that the selected transmission rate is not very sensitive to the received signal strength.

Fig. 11 shows the download speeds broken down on different connection submodes and operators. The box plot shows median, quartile and max/min values. Again we observe clear differences between 2G (GPRS/EDGE) and 3G (WCDMA/HSDPA/HSPA+) connections, while the difference between 3G submodes is less profound. Note that the large majority (80%) of the measurements were taken on HSPA + connections. The large variation seen for two operators in the WCDMA case is due to a few outliers which are also visible in Fig. 10. On closer inspection, we find that these measurements were taken on a location with a very congested network, and that the low throughput is seen because the connection was never promoted to the DCH state as discussed in Section 6.1.

## 7. Roadmap

NNE is funded from public and private sources with a long-term interest in MBB research, and will be maintained as long as it is useful for researchers, network operators and policy makers. In this section, we report on our plans and ambitions for the future development on NNE as a research testbed.

The current deployment consisting of over 400 nodes all over Norway is mostly homogenous, with only small differences in hardware and software capabilities between nodes. This gives advantages both in terms of management and for comparing results across a large number of identical nodes. It is likely, however, that we will see more heterogeneity among nodes in the future. In particular, the modems attached to the NNE nodes need to be updated as MBB networks evolve. The first steps in this direction have already been taken, and a few tens of nodes are now being equipped with 4G (LTE) modems to keep up with the increasing 4G coverage in Norway.

Today almost all NNE nodes are stationary, with only a few experimental mobile nodes. A clear ambition for NNE is to incorporate a significant number of mobile nodes. We are already working with the Norwegian railway operator NSB and several other potential partners in order to make this happen. Mobile nodes will open up for a large number of new and interesting experiments regarding availability, performance and multi-link operation in challenging environments.

Another clear ambition for NNE is to make it easier for other researchers to access and use the infrastructure. Access to NNE nodes will never be made completely open to anyone interested, because of the limited hardware resources on the nodes, the limited (and costly) subscription plans, and the possibility for malicious use. Instead, we aim to develop a simple application procedure for researchers who are interested in running experiments on NNE, and

[12] For interpretation of color in Fig. 10, the reader is referred to the web version of this article.

Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036

tools and guidelines that explain how these experiments can be deployed.

NNE in its current form is deployed only in a single country. While MBB networks are built using the same standards in most parts of the world, there are still interesting differences between networks, as illustrated in the experimental section of this paper. It is therefore interesting to compare measurement results over a wider range of networks than those found in Norway. This can be done by placing NNE nodes in other regions. It is also a goal to make NNE interoperable with other testbed deployments, so that tests and experiment code can be reused between them. The first efforts in this direction is currently underway in collaboration with researchers at the University of Wisconsin–Madison.

## 8. Conclusion

This paper has introduced Nornet Edge, a dedicated infrastructure for measurements and experimentation in MBB networks. NNE consists of a large number (currently over 400) of measurement nodes, placed in voting locations all over Norway, and a set of servers constituting the backend system. The NNE nodes consist of a special-made single-board computer running a standard Linux distribution, with attached MBB modems from up to 5 different operators. The backend system contains functionality for managing nodes, collecting and storing measurement data, and visualizing measurement results.

NNE can be used for a wide range of MBB experiments that target the performance of the MBB network itself, or the performance of protocols and applications running in the network. NNE is particularly well suited for experiments that require long-running or periodic measurements from a large number of identical and controllable measurement nodes, and a large geographical footprint. One of the unique features of NNE is the possibility to run experiments on multiple MBB networks in parallel and on identical hardware, which allows direct comparison of performance metrics across operators. NNE also automatically collects status information from the modem, so that measurement data can be enriched with information about the cell ID, connection mode, signal strength, etc.

NNE is designed to make it easy to deploy new measurement applications and start collecting data. The infrastructure will be made available to other researchers, with the aim of becoming a valuable resource for the network measurements community.

## Acknowledgements

Nornet (which consists of Nornet Edge and Nornet Core) is financed by the Research Council of Norway through the INFRASTRUKTUR program. Nornet Edge is realized with additional generous support from Tele2, Ice.-net and the Ministry of Local Government and Regional Affairs.

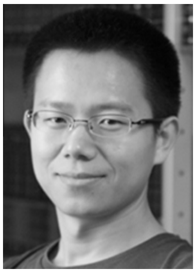
## References

[1] MobiPerf. 2013. <http://www.mobiperf.com>.
[2] Cisco Systems, Inc. Cisco visual networking index: global mobile data traffic forecast update, 2012–2017, 2013.
[3] Communications regulatory authority of the republic of Lithuania. Annual report of the communications regulatory authority (RTT) of the republic of Lithuania, 2011.
[4] K.R. Evensen, MULTI Network Manager (MNM), 2013. <http://github.com/kristrev/multi>.
[5] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, Architectural guidelines for multipath TCP development, 2011. <http://tools.ietf.org/html/rfc6182> (Internet RFCs, RFC 6182, ISSN 2070-1721).
[6] U. Goel, A. Miyyapuram, M.P. Wittie, and Q. Yang, MITATE: mobile internet testbed for application traffic experimentation, in: USENIX Symposium on Networked Systems Design and Implementation, NSDI '13, 2013.
[7] Ernst Gunnar Gran, Thomas Dreibholz, Amund Kvalbein, NorNet Core – A Multi-Homed Research Testbed, Els. Comput. Netw. (2013) (special issue on Future Internet Testbeds).
[8] E. Halepovic, J. Pang, O. Spatscheck, Can you GET me now?: estimating the time-to-first-byte of HTTP transactions with passive measurements, in: Proceedings of the 2012 ACM SIGCOMM Conference on Internet Measurement, IMC'12, 2012.
[9] H. Holma, A. Toskala, WCDMA for UMTS: HSPA Evolution and LTE, fouth ed., John Wiley & Sons Ltd., 2007.
[10] M. Trudel-Lapierre, ModemManager, 2013. <http://launchpad.net/modemmanager>.
[11] F. Qian, Z. Wang, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, Characterizing radio resource allocation for 3G networks, in: Proceedings of the 2010 ACM SIGCOMM Conference on Internet Measurement, IMC'10, 2010.
[12] F. Qian, Z. Wang, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, Profiling resource usage for mobile applications: a cross-layer approach, in: Proceedings of the 2011 International Conference on Mobile Systems, Applications, and Services, MobiSys'11, 2011.
[13] Qualcomm Inc. Qualcomm Interface Control Documents (QXDM), 2013. <http://www.qualcomm.com/solutions/testing/interface-control-documents>.
[14] A. Rao, J. Sherry, A. Legout, A. Krishnamurthy, W. Dabbous, D. Choffnes, Meddle: middleboxes for increased transparency and control of mobile traffic, in: Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop, CoNEXT Student'12, 2012.
[15] SamKnows™, Methodology and technical information relating to the SamKnows™ testing platform – SQ301-002-EN, 2012.
[16] S. Sen, J. Yoon, J. Hare, J. Ormont, S.Banerjee, Can they hear me now?: a case for a client-assisted approach to monitoring wide-area wireless networks, in: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement, IMC'11, 2011.
[17] M.Z. Shafiq, L. Ji, A.X. Liu, J. Pang, S. Venkataraman, J. Wang, A first look at cellular network performance during crowded events, in: Proceedings of the 2013 ACM/SIGMETRICS, SIGMETRICS'13, 2013.
[18] R. Stewart, Stream Control Transmission Protocol (SCTP), 2007. <http://tools.ietf.org/html/rfc4960> (Internet RFCs, RFC 4960, ISSN 2070-1721).
[19] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, A. Pescapè, Broadband internet performance: a view from the gateway, SIGCOMM Comput. Commun. Rev. 41 (2011) 134–145.
[20] S. Triukose, S. Ardon, A. Mahanti, A. Seth, Geolocating IP addresses in cellular data networks, in: Proceedings of the 2012 Passive and Active Measurements Conference, PAM'12, 2012.

**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.

**Džiugas Baltrūnas** is a Ph.D. Student and Research Engineer at Simula Research Laboratory, where he is currently responsible for the infrastructure part of the Nornet Edge project. He received his M.Sc. degree from Vilnius University in 2007. Before joining Simula, Džiugas has been working for a mobile network operator in Lithuania and was responsible for the service platforms and their integration into core and billing parts of the network. His Ph.D. thesis will focus on robustness in mobile broadband networks.

**Kristian Evensen** is currently working as CTO of the Norwegian startup Celerway Communication, developing innovative and disruptive multihome-solutions. Before this, he received his Ph.D. from the University of Oslo in 2012 and worked as Research Engineer in the Resilient Networks project at Simula Research Laboratory. The focus of his Ph.D. was to develop different techniques for aggregating the capacity of multiple, independent network connections. His main research interests are transport layer protocols, routing, operating systems and multihoming issues.



**Jie Xiang** is a research engineer at Simula Research Laboratory. He received his Ph.D. degree from University of Oslo in 2011, M.Eng. degree from Huazhong University of Science and Technology in 2006, and B.Eng. degree from Wuhan University in 2003. His research interests include real-time big data processing and visualization, network measurements, and wireless networks.

**Ahmed Elmokashfi** is a Post-Doc at Simula Research Laboratory. He received the B.Sc. degree in Electrical and Electronics Engineering from the University of Khartoum in 2003, the M.Sc. degree in Telecommunication form Blekinge Institute of Technology in 2007, and the Ph.D. degree from the University of Oslo in 2011. His current research focuses on the resilience, scalability, and evolution of the Internet infrastructure; the measurement and quantification of robustness in mobile broadband networks; and the understanding of dynamical complex systems.

**Simone Ferlin** received her Diplom (Dipl.-Ing.) degree in Information Technology from the University Friedrich-Alexander Erlangen-Nürnberg (FAU), Germany in 2010. Currently, she is doing her Ph.D. in the Network Systems Group at Simula Research Laboratory in Fornebu, Norway. Her current research interests involve mobile broadband networks, cross-layer optimization, network programming and multipath transmission with MPTCP and CMT-SCTP.



Please cite this article in press as: A. Kvalbein et al., The Nornet Edge platform for mobile broadband measurements, Comput. Netw. (2014), http://dx.doi.org/10.1016/j.bjp.2013.12.036