Radboud University Nijmegen

LTE-WiFi Handover Strangelove

Or: How I Started Worrying and Fear the OSI model

Gerdriaan Mulder

31 January 2014

Bachelor thesis

Radboud University Nijmegen
Institute for Computing and Information Sciences

Supervisor
Dr. Ir. Erik Poll
Radboud University Nijmegen
e.poll@cs.ru.nl

Supervisor
Fabian van den Broek, MSc
Radboud University Nijmegen
f.vandenbroek@cs.ru.nl

Abstract. This thesis examines the current techniques in LTE-WiFi data handover. Handovers take place when a mobile device switches from one network to another. It is interesting to look at methods to offload the rather expensive mobile data connections to the cheaper WiFi (home) networks. This transition is usually not seamless. A good example is when you start a streaming video whilst on mobile data and a known WiFi network appears. Your mobile device automatically connects to the WiFi network and the streaming video stops. These so-called *vertical* handovers have not been made seamless yet. This thesis compares several techniques that operate on different layers of the OSI model. To facilitate vertical handover, it is useful to know how *horizontal* handovers work. This kind of handover occurs when, for example, a mobile phone switches from one cell tower to another. Contrary to vertical handover, horizontal handover occurs practically seamless. Horizontal handovers in both LTE and WiFi networks are discussed, to give a heads up for the problems that arise for vertical handovers. Vertical handovers can be done at different points in the OSI model. This thesis covers solutions that have been devised on a few of these layers. The solutions include *Mosh*, *Multipath TCP* and *Mobile IP / Proxy Mobile IPv6*. All of these solutions try to solve the big problem in vertical handover: the change of the client's IP address. Not all of these solutions work as well as expected. Mosh works very well and Multipath TCP looks very promising, although it did not work out in the experiment that is done in this thesis.

Contents

1. Introduction

Mobile devices can access the internet through several technologies. This evolved in cellphones from the rather slow, *modem-speed* GPRS to the high-speed LTE technology. On other mobile devices, such as tablets or notebook computers, access to the internet is done via WiFi. Nowadays, practically every mobile device has a combination of these technologies built-in.

Data subscriptions for cellphones are rather expensive. They occasionally have a data cap and the price per megabyte beyond the data cap can be quite high. A regular LTE subscription with a data cap of 50 gigabytes per month can be consumed in about 5.5 hours when downloading at a constant speed of 20 megabits per second.[1] It is, therefore, interesting to use a cheaper WiFi connection when it is available. It is rather unpleasant, however, when the connection drops during the switch from LTE to WiFi and, for example, a streaming video needs to be restarted.

When a mobile device switches from LTE to WiFi, this is called *vertical handover*[15]. Its counterpart, *horizontal handover*, happens regularly when traveling with, for example, a cellphone. When one cell tower gets out of range and another one is in range, a handover occurs between these cell towers, such that an existing call stays connected. Usually the phrase *roaming* is used for horizontal handover.[2]

In our current society, everybody wants to check their e-mail, browse the web, watch (streaming / live) videos at any given time. This causes problems when a lot of people want to do this in a highly-crowded area. It is much cheaper for the user to use a nearby, free WiFi network than their data plan. To facilitate a vertical handover between WiFi and LTE, several aspects need to be looked into:

- What (technical) infrastructure needs to be setup to facilitate vertical handover?
- What (security) issues arise when moving from one network to another?

There are solutions available and in use, such as *Multipath TCP*. This technology adds a few parameters to a connection in order to be able to use another *internet path* from client to server when the initially established connection is interrupted or breaks down. Multipath TCP has already been deployed on Apple's iOS 7 platform, especially when Siri is used[8][28].

In the Netherlands, several parties have experimented with LTE and convergence with existing university networks[21]. SURFnet, the Radboud University Nijmegen and Tele2 have studied vertical handovers between the WiFi network *eduroam* and a local test LTE network. Their results varied from "working with a couple of seconds delay" to "does not work"[10].

This thesis first shows some theory behind LTE and WiFi networks, in section 3 and 4. In these sections, first, the network topology of the technology is explained, after

[1] This is the current, average speed for LTE in The Netherlands.

[2] This term is also used in another context when a mobile subscriber is using a GSM network different from its own *home* network, for instance when he is abroad.

which the horizontal handover procedure for that technology is explained. In sections 5 and 6, vertical handovers are introduced and a comparison of several vertical handover techniques is discussed. As a practical experiment, this thesis examines how Multipath TCP can be deployed in section 7, and what the results of that experiment are. Finally, related work is briefly addressed and a couple of topics for future research are given.

2. Definitions

This thesis addresses several subjects, each with their own set of definitions. They are listed below. Some definitions are synonymous to each other and are summarized in a single, universal term below in 2.1.

LTE: *Long Term Evolution*, or *4G*, used for transmitting data at high speed over cellular networks.

WiFi: *Wireless Fidelity*, the commonly used term for IEEE 802.11[5] networks. Another common term is *WLAN*: Wireless Local Area Network.

Non-3GPP: This term indicates networks other than LTE or 3G networks, which are not specified by the 3rd Generation Partnership Project such as WiFi and *WiMAX*[13].

Handover: The process of switching from one radio transceiver to another

Horizontal handover: Handover within a wireless technology, such as the switch from access point to access point in WiFi, or the switch from one cell tower to another in LTE.

Vertical handover: Handover between two different wireless technologies, such as LTE and WiFi.

ME: *Mobile Equipment*, such as a cellphone, a wirelessly connected laptop, tablet, etc.

UE: *User Equipment*, synonymous to `ME`.

MN: *Mobile Node*, used in Mobile IP. Synonymous to `ME`.

HA: *Home Agent*, used in Mobile IP. This indicates the device that is situated in the *home network* of the `MN`.

FA: *Foreign Agent*, used in Mobile IP. This indicates the device that the `MN` is connected to when it is not in its home network.

RADIUS: *Remote Authentication Dial In User Service*. This service is used to authenticate users to a network.

2.1. Recurring definitions.

Mobile device: This term is used to indicate an `ME`, `UE` or `MN`.

Network operator: This term is used to indicate an Internet Service Provider or Mobile Network Operator.

3. LTE - *Long-Term Evolution*

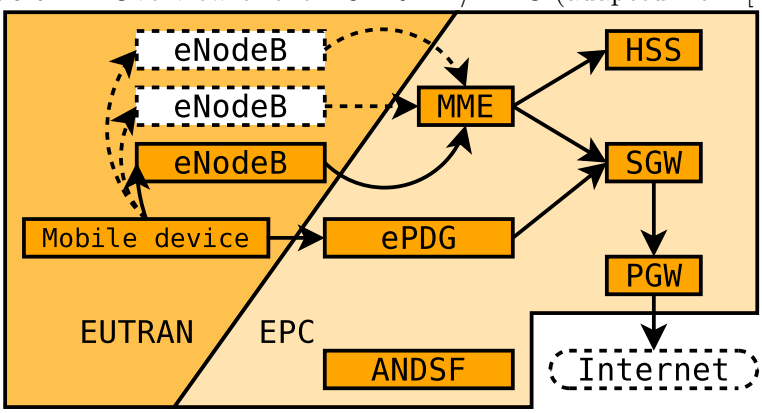
This section explains what LTE is and the components which it consists of, in order to get an understanding of the technology and how horizontal handovers are facilitated. A horizontal handover occurs when a mobile node switches from one cell tower to another cell tower, depicted by the dotted *eNodeB*'s in Figure 1

LTE (or 4G) is the successor of 3G and its predecessor GPRS. It is used for calling, sending text messages and (internet) data transfer over a cellular network to a mobile device. GPRS and 3G were rather limited in speed (about 250 kilobits per second and 15 megabits per second, respectively) and were built on circuit-switched and packet-switched network components for voice and text messages, and data transfer, respectively. LTE has been redesigned, such that it only uses packet-switched components for voice, text messages and data transfer[13]. This is usually referred to as being an *All-IP* network.

3.1. **Network topology.**

This section gives a short overview of the components involved in LTE: the *EUTRAN* and *Evolved Packet Core*, and how authentication takes place.

FIGURE 1. Overview of the EUTRAN / EPC (adapted from [17])



In short, the components in LTE can be summed up as follows. For readers familiar with GSM, the GSM equivalent of some LTE components are also given.

- `EUTRAN` (Evolved UMTS Terrestrial Radio Access Network), which consists of:
  - `UE`, User Equipment, a mobile phone, a tablet, etc.
  - `eNodeB`, Evolved Node B, a computer system that controls a (group of) cell tower(s).
- `EPC` (Evolved Packet Core), which consists of:

  - `MME`, Mobility Management Entity. This device tracks all cellular devices and is responsible for authentication, authorization and accounting of those devices. This device is comparable to the Visitor Location Register (VLR) in GSM.
  - `HSS`, Home Subscriber Server. This server keeps track of users and subscriptions. This device is comparable to the Home Location Register (HLR) in GSM.
  - `SGW`, Serving Gateway. This gateway connects to the `UE` and `PGW` and is used during horizontal handovers. This is comparable to the Serving GPRS Support Node (SGSN) in GSM.
  - `PGW`, PDN Gateway. This gateway is connected to all `SGW`'s and serves as a gateway to a *Public Data Network*, such as the internet. This is comparable to the Gateway GPRS Support Node (GGSN) in GSM.
- The following devices are also in the `EPC`, but they are not covered in this thesis.
  - `ANDSF`, Access Network Discovery and Selection Function. This system can be used to determine nearby access networks, such as WiFi networks, and provide policies for connecting to such networks.
  - `ePDG`, Evolved Packet Data Gateway. This gateway ensures secure transmission of data on untrusted networks, when some `UE` wants to connect to the `EPC`. This is used for connecting femtocells[27] across the (untrusted) internet, roaming clients on *non-3GPP* networks and lawful interception.

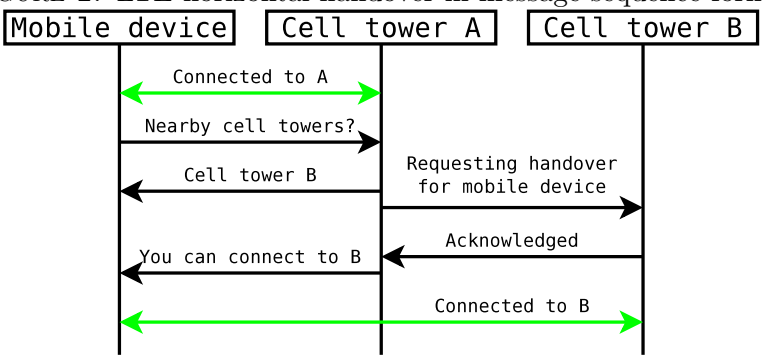
The mobile device has a *Subscriber Identity Module* card, or SIM card. This is a small smartcard that contains information about the network operator, the phone number and perhaps a couple of additional applications that have been stored on it by the network operator. Authentication is done mutually[22], that is, the network authenticates the mobile device and the mobile device authenticates the network. In this way, both parties know that they are legitimate. This authentication scheme is based on a symmetric key K, that is stored on the SIM card as well as in the `HSS`. K is used to generate new key material for securing future communication. The protocol used is called EPS-AKA, which is not discussed further.

3.2. **Horizontal handover.** With regards to *horizontal handover*, i.e. when a mobile device switches from one cell tower (the dotted *eNodeB*s in Figure 1) to the other, the following things happen:

(1) The mobile device gets out of range of one cell tower: A
(2) The mobile device gets in range of another cell tower[3]: B
(3) The mobile device and A exchange measurements regarding nearby cell towers and their signal strength

[3]The coverage of these cell towers may overlap.

FIGURE 2. LTE horizontal handover in message sequence format



(4) A decides that a handover is needed to B and sends a `HANDOVER REQUEST` to B

(5) B prepares for the handover and, if this is likely to succeed, sends a `HANDOVER REQUEST ACKNOWLEDGE`

(6) A sends reconfiguration parameters to the mobile device

(7) *The handover occurs*

(8) The mobile device disconnects from A and reconnects with B

In Figure 2 this is shown in message sequence format. After the handover is complete, the mobile device has the same IP address as before the handover.

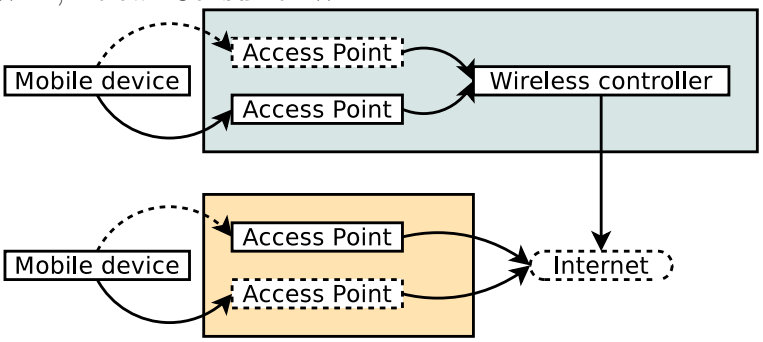
4. WiFi - *802.11 networks*

This section gives an overview of WiFi networks, the difference between consumer WiFi and enterprise WiFi and how horizontal handovers work in both cases. First, the network topology for both cases is explained. After that, horizontal handover is discussed for consumer and enterprise WiFi.

4.1. **Network topology.**

This section explains how a WiFi network is constructed. In Figure 3, two situations are shown. Both situations are further explained below and how authentication takes place.

FIGURE 3. Overview of two types of WiFi infrastructure. Above: Enterprise WiFi, Below: Consumer WiFi



Consumer WiFi networks consist of at least one *access point*. In this situation, this access point also functions as gateway, or router, to the internet. A wireless network is commonly identified using the human-readable *SSID* (Service Set IDentification). Security (and in this case, also authentication) of the wireless connection is obtained by configuring a password, called a *pre-shared key* (PSK) and entering this password on each device that wants to connect to the wireless network. The access point makes sure each connected client gets an IP address and further information to connect to the internet.

These WiFi access points are considered as *stand-alone*. It is possible to extend the range of the network by configuring more access points and making sure each device has the same SSID and PSK. This is rather inconvenient, from the user's perspective, as well as the maintainer's perspective, as can be seen next.

From the user's perspective, each access point is a separate device, which may or may not hand out the same IP address and therefore break services like video streaming. Another inconvenience is that the user may be in range of both access points, such that his connection constantly switches from one access point to the other.

From the maintainer's perspective, the decentralized nature of the setup means each access point needs to be reconfigured when either the SSID or the PSK needs to be

changed. Another nuisance is that load balancing of connected devices (i.e. evenly distribute the devices across access points in the neighbourhood) is practically impossible. Each access point operates as an island, without knowledge about its surroundings.

In larger companies, universities and conferences, people currently expect to have a wireless connection everywhere they go. One of the fine examples is the Chaos Communication Congress, which takes place every year just between Christmas and New Year. Their 30th conference had about 9000 visitors and their peak usage was 5000 unique devices on their wireless network[11]. This is unmanageable when each access point is on its own.

4.2. **Enterprise WiFi.** Thankfully, there is a solution, often called *enterprise* WiFi. Each access point is connected to a *wireless controller*, through which all traffic to and from the access points is going. One of the protocols that is used for this is the *Lightweight Access Point Protocol*[9][4], which will not be discussed further, though only used as leading example.

A wireless controller is able to control all traffic to and from its connected access points. This controller has a couple of tasks to perform. It needs to configure the access points in terms of transmitting power, SSID, encryption scheme (if any), etc. Next, it needs to be able to control interference between access points, facilitate in load balancing and, if needed, control the transmitting power of each access point to control gaps in coverage. Wireless controllers also facilitate in *AAA*: *Authentication*, *Authorization* and *Accounting*. Authentication is usually done using a standard called 802.1X. The user enters a username and password as authentication to the network. The authentication scheme is shown below, in Figure 4. As explained further on, authorization is not necessary, per se, although it is possible to put clients in a separate *VLAN* (Virtual Local Area Network). A practical example is to have employees and guests connect to the same SSID, but to put employees in the company network VLAN and guests in a guest VLAN. This makes sure guests are able to, for example, access the internet, but are not able to access the company's private data. Finally, accounting can be used when offering a paid WiFi service, or just to create statistics of the network usage.

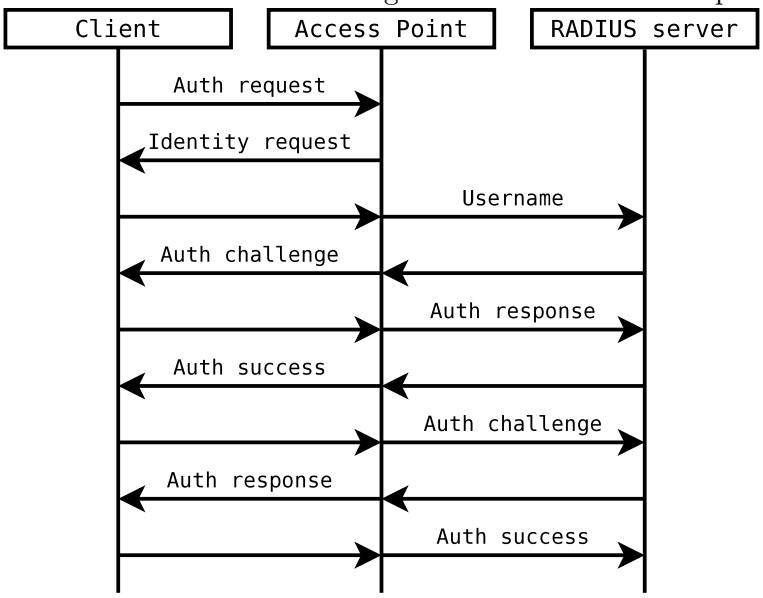
4.3. **Horizontal handover.** This section discusses horizontal handover in consumer and enterprise environments.

Horizontal handovers "consumer style" are not very difficult, although it is technically not a handover. In short: the mobile device disconnects from one access point and connects to another one. The IP address may stay the same, although it depends on how the access points are connected.

In enterprise environments, authentication plays a key role in the network. The standard scheme for authentication uses a *RADIUS* server, which takes care of the previously mentioned *AAA*. Authentication through an access point is done as follows (adapted from [25]):

[4]Several companies have devised their own (proprietary) standard for configuring and managing multiple access points. LWAPP was devised by Cisco Systems, but later accepted as IETF standard.

FIGURE 4. Authentication using RADIUS and an access point



This is a rather lengthy process, especially when a mobile device regularly switches from access point to access point. Several vendors have developed techniques, varying from Cisco's Centralized Key Management (CCKM)[25], to Zero-Handoff[26] by Ubiquiti Networks. The former uses an intermediate authentication server which handles reassociation requests from mobile devices that are CCKM-capable. This technique reduces the number of exchanged packets to about 4 [25, Figure 11-2]. The latter technique let multiple access points be seen as one, which makes the mobile device think it is connected to a single access point[26].

5. Vertical handover between LTE and WiFi

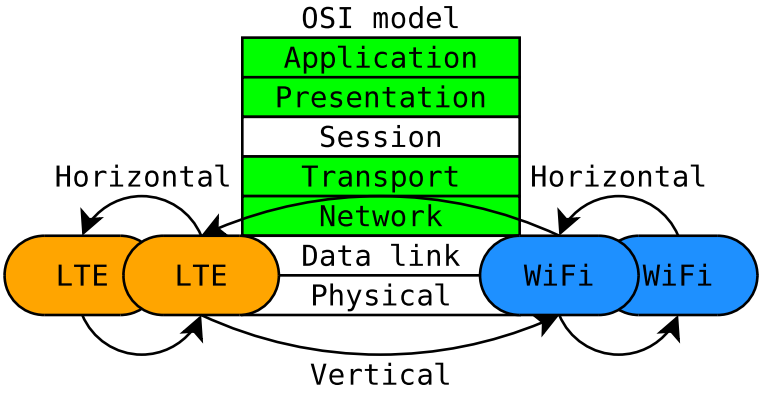
This section will explain what problems arise when doing a vertical handover. First, a small recap on the distinction between horizontal and vertical handover is given. Next, the problem with addressing is looked into. Finally, a few organizational aspects are explained.

In section 6, several solutions for vertical handover are compared to each other.

5.1. **Small recap.** There are two types of handover: within one wireless system and between different wireless systems. The former is called *horizontal* handover, the latter is called *vertical* handover. Horizontal handovers have been discussed in section 3 and section 4 for LTE and WiFi, respectively.

Vertical handover has some more elements involved. For example, the mobile device needs to be able to connect to each wireless system. In the figure below, the OSI model is shown with the distinction between horizontal and vertical handovers. Furthermore the four layers which are discussed in section 6 are shown with a green background.

Figure 5. Overview of the OSI model and where LTE and WiFi handovers take place



5.2. **Addressing.** When a client connects to a server, the server assumes that the IP address of the client stays the same. This has been the case for fixed devices, such as personal computers, but with mobile devices, such as tablets, this is not always the case.

Applications such as `ssh`[5] can survive a handover or connection drop as long as the client's IP address stays the same (as is the case with horizontal handovers in LTE and enterprise WiFi). This situation changes in the case of vertical handover, when this assumption might no longer hold. Suppose there are two network operators, one manages a WiFi network, the other an LTE network. These network operators use their own address space to give each mobile device a unique IP address within that range. For example, my current IP address on a WiFi network is 10.87.118.138. When

[5] `ssh` is used to securely connect to a remote server: *Secure SHell*.

I want to switch to the LTE network, my wireless device asks for an IP address in order to communicate with the network. But when I switch, the IP address changes to 62.140.137.70, which is completely different. This means an application such as `ssh` does not work anymore.

This specific problem can be resolved, but it needs the support of each network operator that is involved in the vertical handover. Each network operator needs to be connected to each other and needs to accept all clients coming from another operator's network. Then, it can give out an IP address that is the same as where the wireless device initially came from. Applications like `ssh` will survive this handover, because the IP address stays the same. This is, however, easier said than done. Network operators have their own IP space which only they can operate on. It is likely that an *IP address sharing* solution is unfeasible in practice.

This IP address sharing solution also yields an organizational problem. Every operator needs to be connected to each other and be able to accept other clients. In the case of 4 operators, this means 6 connections need to be made. In the case of 8 operators, this rises to 29 connections.

5.3. **Organizational challenges.** In Nijmegen, a pilot study on LTE and handovers between LTE and WiFi was started in September 2012, with SURFnet, Tele2 and the Radboud University as operating parties. Next to the educational advantages[6] in terms of access to restricted sources within the university, the security and organizational aspects were to be researched. Furthermore, the pilot used *Proxy Mobile IP*, which is explained in section 6.1. SURFnet had already executed a pilot study in Utrecht in 2012[21], with cooperation of KPN, University of Utrecht, its medical center, and the University of Applied Sciences of Utrecht. The results of the pilot study in Nijmegen have not been published yet, although preliminary results indicate that only 2 of the 13 vertical handover tests[7] succeeded with a delay of about 3 seconds, which is probably the time it takes to find and connect to a WiFi network [14]. The other tests either indicated that the application stopped working or only sometimes survived a handover.

In the case of handover between LTE and WiFi, the organizations involved need to agree upon where, for example, authentication takes place. An LTE network operator has a database (HSS, see figure 1) which takes care of the authentication of its subscribers' devices. An enterprise WiFi operator, such as the Radboud University, has another way of authenticating its employees and students. This is usually done using RADIUS (see figure 4). Both networks authenticate their users, but these systems use different technologies. In order to connect these technologies, some sort of proxy needs to be used. The design document of the pilot study in Utrecht pictured a situation in which the authentication server of the LTE network operator was connected, via SURFnet, to the institute's authentication server (figure 6). In this case, the authentication servers are linked through RADIUS.

[6]The test was to take place around the botanic garden and greenhouses of the university.

[7]Handover from LTE to WiFi as well as WiFi to LTE were tested with 13 different applications, such as web radio and television, Skype and `sftp`.

FIGURE 6. "Institutional traffic"[21, Figure 2], showing the infrastructure used to link an LTE network with an institute's network (*note*: `eNB` is an abbreviation for *eNodeB*)

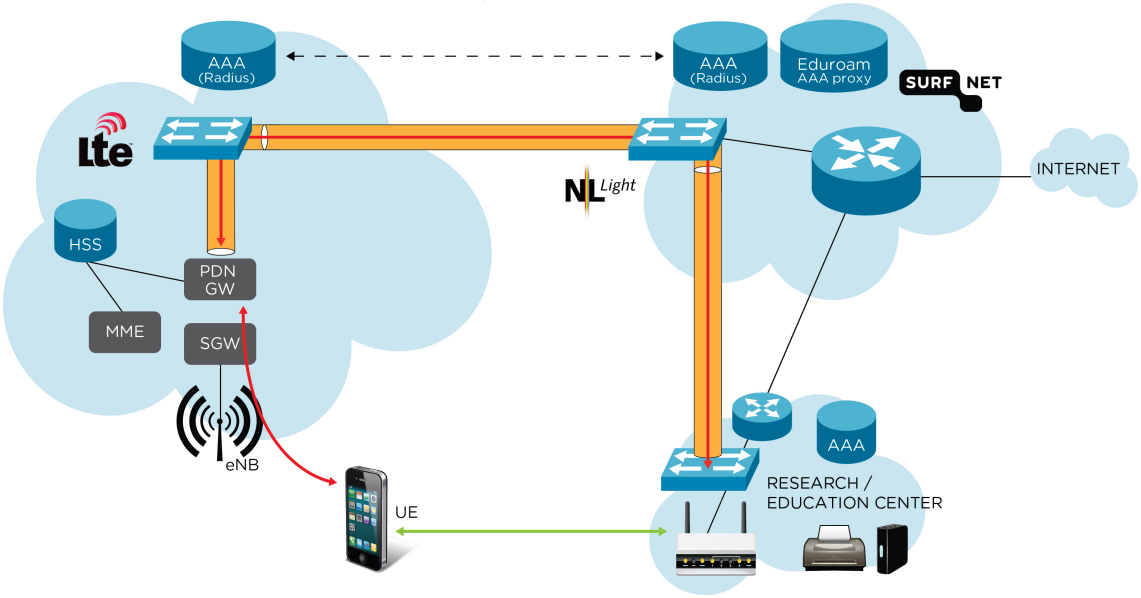


Figure 6 shows how so-called *institutional traffic* is transferred from the LTE network to the institute's network. The authentication is done using a RADIUS proxy in the network of SURFnet. The yellow connections indicate a *light path*, or direct connection, between network operators. In this way, the mobile device on the LTE network has the same network access to the institute's systems as if it were on the WiFi network.

6. Theoretical comparison between vertical handover techniques

There are various vertical handover techniques, each of which operate on a different layer within the seven-layered OSI model. In this section, a short introduction on the OSI model is given. Next, handover techniques from different layers are compared in terms of speed, usability and implementability.

A quick, simplified overview of the seven layers in the OSI model:

(1) *Physical* layer: Specification of the physical transport of bits

(2) *Data link* layer: Communication concerning "local" networks (Ethernet, switching)

(3) *Network* layer: Interconnection between "local" networks (IPv4/IPv6, routing)

(4) *Transport* layer: Provides an interface for applications to setup connections (TCP, UDP)

(5) *Session* layer: Provides an interface for using sessions (NetBIOS, PPTP)

(6) *Presentation* layer: Formats information such that it can be used in the application layer (data representation, XML)

(7) *Application* layer: Your web browser, e-mail client, etc.

These seven layers are often compacted into a five-layered model, in which layers 5 through 7 are considered as one *data* layer.

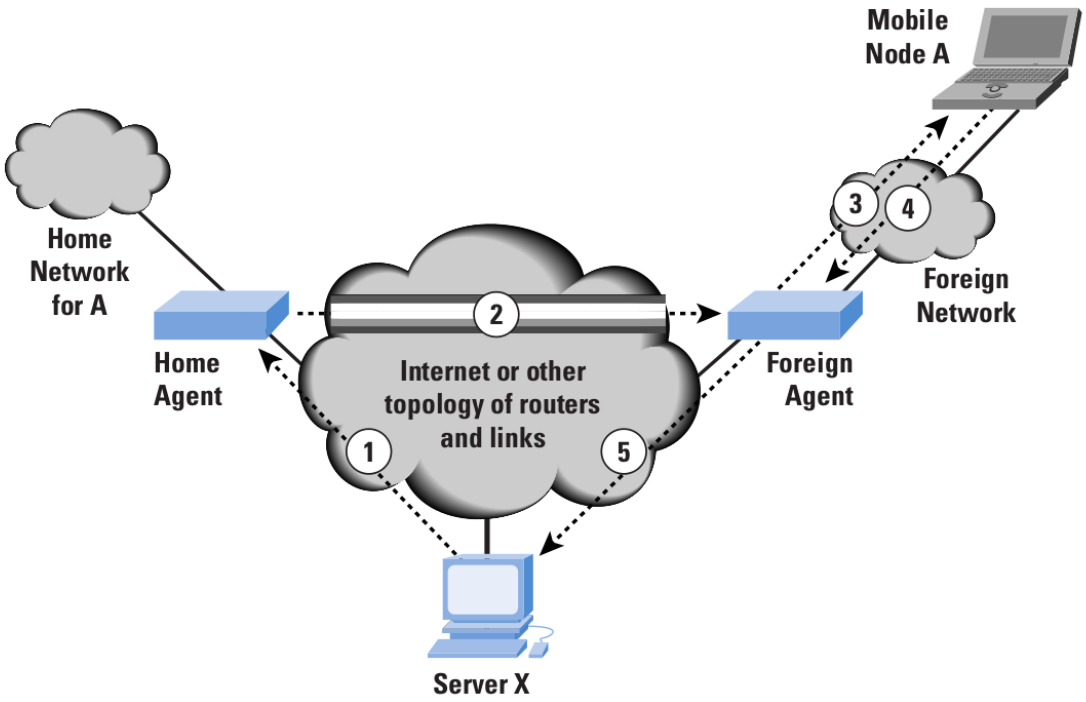
The OSI layers that are discussed in this section are shown in Figure 5, with a green background. In the same figure, the distinction between horizontal and vertical handover is made. Horizontal handovers take place in the first two layers; vertical handovers in the layers above. The OSI model was designed to create a level of abstraction. The higher layers should be agnostic of what is happening on lower layers. For example, the network layer is unaware of what exactly happens on the physical layer. This does not always work out. For example, most web applications (such as forums, social network sites) use the client's IP address to create a session. It creates a dependency on the network layer, which breaches the abstraction of the OSI model. In this way, a change of IP address automatically breaks the session in such (web) applications.

In the next sections, four vertical handover techniques are compared: *Mobile IP & Proxy Mobile IP(v6)*, *Multipath TCP*, *TLS Renegotiation*, and *Mosh*.

6.1. Mobile IP & Proxy Mobile IP(v6).

When it comes to mobility, it is possible that the client switches from network and thus of IP address. Mobile IP offers a solution to this problem. A *mobile node* has a certain *home network* with its associated static *home address*. When the mobile node switches to a *foreign network*, it contacts its *home agent*, indicating that its location has updated. The mobile node sends its *care-of address* which identifies its location to the home agent. The home agent connects through a tunnel to the *foreign agent* in order to correctly route the packets destined for the mobile node.

FIGURE 7. Mobile IP Scenario[24, p. 3]



When a server sends packets to the mobile node they are intercepted by the home agent and tunneled to the foreign agent. In fact, the home agent acts as a *Man-in-the-Middle*. This results in a situation where other devices in the network of the home agent are unaware of the location change of the mobile node. This solution was used in the Nijmegen LTE-WiFi project[10]

6.2. **Multipath TCP.** Multipath TCP (MPTCP) is a technique which can use multiple, separate TCP connections to the same server. These connections may use different paths on the internet. For example, a mobile device can use its 4G connection and its WiFi connection at the same time to stream a video from a server.

MPTCP is an extension on the Transmission Control Protocol, which adds a few options to the TCP header. This ensures backwards compatibility with regular TCP. If a device does not recognize the MPTCP options, they can be safely ignored[8].

An MPTCP connection consists of one or more *subflows*. A subflow is in fact the same as a regular TCP connection. When the initial MPTCP subflow is setup, the two hosts exchange a key which is used to construct or replace (additional) subflows.

The experiment on Multipath TCP and its results can be found in section 7.

[8]Some devices drop unknown TCP options, which usually breaks MPTCP connections.

6.3. **TLS Renegotiation.** TLS, or Transport Layer Security, is a generic protocol for securing other protocols. The most common use is TLS over HTTP, also known as HTTPS. TLS renegotiation is a technique used for changing the parameters of a TLS connection in the current TLS session. Furthermore, it has been shown that TLS renegotiation was vulnerable[12], such that an attacker could insert traffic into an existing TLS connection, while making it look like a legitimate request.

TLS renegotiation is used in OpenVPN[4], for example when a certain amount of time has passed. In terms of cryptography, it temporarily accepts old key material and new key material to make the renegotiation seamless. This technique can be used in vertical handovers, although it looks like the software using it needs to control the handover, as can be seen in the case of OpenVPN.

6.4. **Mosh.** Mosh is a terminal application that replaces `ssh` and provides local input prediction and seamless handover when the route to the server is changed (for instance, when switching from LTE to WiFi). It uses a newly developed protocol: State Synchronization Protocol, or SSP. This protocol is used to synchronize the screen state of a terminal between a server and a client. Both server and client keep a sequence number which indicates the current state of the terminal screen. When the client has a lower sequence number than the server, it can immediately update the screen without replaying all key presses or echoing all output that has been generated since.

The initial connection is setup through SSH. This provides authenticity, integrity and confidentiality through the SSH protocol. It then executes `mosh-server`, which creates a random shared encryption key and listens on a high UDP port. The SSH connection is then shut down and further communication is done through UDP.

In order to provide a roaming experience across switches of networks, SSP uses sequence numbers, which are strictly increasing. Since authentication and creation of the shared secret is done through SSH, a legit datagram from a *new* connection can be distinguished from a false datagram with the shared secret. When the sequence number of the datagram is higher than the last one the server received and the datagram is authentic, the server uses the IP address of that datagram to set the new target for the connection.

In a real-world test of about 40 hours, the researchers gathered around 10,000 keystrokes from six users. They then replayed the sessions over a commercial 3G network and a wired connection. Overall, Mosh showed a lower latency than SSH. Moreover, after a couple of peers were enthusiastic about Mosh, I decided to use it as well. I installed it on October 13, 2013 and it still makes me smile when the screen state updates automatically when I have switched networks. The usual switch involves suspending my laptop computer at home and resuming it at my university, and connecting to the university's wireless network.. My session is usually restored in about 30 seconds. This is the amount of time it takes to resume my computer and connect to the wireless network.

Although Mosh is a specific application with limited scope, it shows that it is possible to create a roaming environment for a single application, without altering current network infrastructure. Apart from the results gathered in an academical form, simply using it and being satisfied gives a good indication of its usefulness in practice.

6.5. **Comparison of the vertical handover techniques.** The table below sums up the four vertical handover techniques and gives a comparison in terms of speed, usability and implementability. Several comments on the comparison are also given below.

A handover in Mobile IP seems to have a moderate speed. It depends on the network latency in too many places to be able to do it fast. The LTE-WiFi project in Nijmegen used this technique in their pilot, so it looks usable, although the implementability might be subject to specific knowledge.

Multipath TCP is demonstrated[19] to be a fast vertical handover technique. Its usability and implementability seem to look fine, although section 7 concludes that it is difficult to make it really work.

TLS renegotiation is a technique that needs an application to implement it before it is usable. After it is embedded in an application, the application's user is (usually) unaware of its existence. TLS renegotiation is a usable, though vulnerable vertical handover technique.

The last vertical handover technique is Mosh. From my own experience, speed, usability as well as implementability are good or even very good.

TABLE 1. Overview of vertical handover techniques

Handover technique	Speed	Usability	Implementability
Mobile IP	Moderate	Seems usable	Specific knowledge on networks needed
Multipath TCP	Fast	Seems usable	Although there are instructions available, these do not seem to work very well
TLS Renegotiation	Fast, though insecure[12]	Does not need interaction of the user	Implementation is done through (browser) software
Mosh	Very fast	Easy	A single package needs to be installed on client and server

7. Practical experiment with Multipath TCP

This section shows an experimental setup with Multipath TCP. The experiment's target is to check whether it is easy to install, easily configured and roughly works as can be seen in a demo[19]. This section first gives an overview of the prerequisites. Next, it shows what configuration options are available. After that, the test setup is discussed and finally, the results of the experiment are given.

When searching for vertical handover techniques to experiment with, this extension to the Transport Control Protocol (TCP) came up. The reason to conduct an experiment with Multipath TCP was that it was available on Linux and reasonably well documented. Next to that, several papers have been written on this subject.

7.1. **Overview.** Multipath TCP has been implemented in the Linux kernel. In order to use MPTCP, it is required to use a kernel that supports it. This can be achieved by getting the source[18], configuring the kernel and compiling it. When you're on a standard Debian-based system, it can be installed using `apt-get`. On Debian Wheezy (assuming you're root[9]):

```
wget -q -O - http://multipath-tcp.org/mptcp.gpg.key | apt-key add -
echo "deb http://multipath-tcp.org/repos/apt/debian wheezy main" >> \
  /etc/apt/sources.list
apt-get update
apt-get install linux-mptcp
```

7.2. **Configuration.** There are several configuration options for MPTCP, as well as its path-manager. This can be controlled using `sysctl`. The options are described below.

- `net.mptcp.mptcp_enabled` Whether MPTCP is used, 0 or 1, defaults to 1.
- `net.mptcp.mptcp_checksum` Toggles the use of a DSS-checksum, 0 or 1, defaults to 1.
- `net.mptcp.mptcp_syn_retries` Number of SYN-retries that are sent, in order to evade network appliances that strip TCP options. Defaults to 3.
- `net.mptcp.mptcp_path_manager` Sets the path manager.
  - `default` Passively use MPTCP, do not initiate new subflows or announce different IP addresses.
  - `fullmesh` Applies the full-mesh tactic for every subflow.
  - `ndiffports` Use n subflows with a pair of IP addresses.

When the kernel has been installed, one way of checking whether MPTCP is enabled is to go to `http://amiusingmptcp.com/`. This website then responds with a yes or no, which indicates whether MPTCP is available. The repositories of MPTCP have a

[9] Also assuming the website is up and running, which at some point wasn't the case while writing this thesis.

patched version of `tcpdump` which can be used to check for the usage of MPTCP. An example of its output can be seen in section 9.

7.3. **Test setup.** The systems I have configured are a Virtualbox virtual machine called `bastogne`, with Debian Wheezy (7.2) using a standard configuration (1 CPU, 384 MB RAM, 2x Intel PRO/1000 MT Desktop in bridged mode) and an existing Xen virtual machine, called `rijksdaalder`, with Debian Wheezy (2 CPUs, 128 MB RAM, publicly connected).

Next to the installation of the kernel and some userspace tools, it is necessary to configure routing for each separate network that is used. More info about that can be found on `http://multipath-tcp.org/pmwiki.php/Users/ConfigureRouting`.

After this configuration stage is completed, two tests were conducted. The first test indicates whether Multipath TCP is correctly installed. This test uses a website (`http://amiusingmptcp.com/`) that is Multipath TCP capable to check whether each system (`bastogne` and `rijksdaalder`) is able to contact an external Multipath TCP capable system. The configuration options as seen above were left unchanged.

In the second test, the configuration options of the *path manager* were changed, and the network testing tool `iperf` [2] was used to generate traffic. To check which interface is used, the bandwidth monitoring tool `bwm-ng` [1] was used. The settings for the path manager are shown below:

- `net.mptcp.mptcp_path_manager`:
  - `default`
  - `fullmesh`
  - `ndiffports`

Furthermore, when a connection was running, one of the network interfaces was disabled to test whether the connection was actually sent over multiple paths.

7.4. **Results.** The intial setup was quite easy to accomplish. Apart from website unavailability, the installation went very smooth on both machines. A simple reboot was necessary to load the new kernel. In order to view what connections use MPTCP, `netstat -m` was used.

The first test, as explained above, succeeded: `bastogne` as well as `rijksdaalder` had been deployed with the MPTCP kernel, so the website responded with **yes**!

The second test did not go as expected. The initial connection was set up correctly and showed up in `netstat -m`. When one of the network interfaces was disabled, the traffic stopped completely. In `bwm-ng`, the transfer rate dropped to 0 KB/s for all interfaces. When re-enabling the disabled interface, the transfer did not continue. This behaviour did not change when changing the configuration options on either side. Due to time limitations, this has not been investigated further.

8. Related and future work

This section looks back on related work and poses a few topics that are useful to research in the future.

8.1. **Related work.** In [6], a network simulator tool was used to simulate horizontal handovers in WiFi using PMIPv6 and horizontal handovers in 3G using its built-in mechanism, with a moving wireless device. They devised a fitness function to determine when to perform a handover. [7] is similar to [6], but they used a Nokia N900 instead of a simulation tool. They devised a handover strategy based on the signal strength of available networks.

In another simulation study[16], a decision algorithm for vertical handovers between WiFi and LTE was devised. It proposed a system to link LTE and WiFi such that vertical handovers were supported by the network infrastructure. One of the factors that was proposed to help the handover decision was the speed of the mobile device.

LTE handovers were studied[27] in order to determine handover procedures in an LTE network which consists of many femtocells instead of one larger cell tower that covers a large area. It proposed a decision policy such that unnecessary handovers were prevented.

Multipath TCP[20] provides an extension to the Transmission Control Protocol, such that multiple connections can be used to connect to the same server. A demo[19] and viability study[23] confirm that it is an extension to TCP that works, although they do not describe the exact way they have set it up.

In [14] WiMAX to WiFi handovers were studied terms of delay when using SIP (internet telephony). It was discovered that DHCP, WiMAX authentication and probing for WiFi networks were the reasons for substantial delay in vertical handovers.

8.2. **Future work.** Multipath TCP is a relatively new technique, which should be investigated more thoroughly, in terms of deployment as well as security. As for deployment, the instructions given on their website do not automatically guarantee a working setup, as was shown in section 7. As for security purposes, it would be nice to check whether subflows can be hijacked, or whether new subflows can be initiated by a third party (e.g. eavesdropper).

Another subject to look into is handover decision strategies for mobile devices that can connect to LTE and WiFi. [7] shows an implementation for a specific mobile device. This could be extended to another study amongst multiple vendors and devices.

Furthermore, the network stack in mobile device operating systems, such as Android, make applications bind to a specific network interface[10]. This makes it difficult to have a transparent connection to either WiFi or LTE, regardless of which one is active. For example, an extra layer between the applications and the network stack can make applications agnostic of the physical data connection used.

Another issue that can be addressed is the usage of IPv4 versus IPv6. Mosh is known to *not* work with IPv6[3]. Next to vertical handover between LTE and WiFI, it is interesting to look at handovers between IPv4 and IPv6.

9. Conclusions

This thesis introduced the problem of switching from LTE to WiFi, with the issue that not every application is able to cope with this transition. Streaming video usually stops, applications such as `ssh` stop functioning and login-based websites break in terms of session persistence.

With the knowledge of the two technologies and their horizontal handover techniques, the concept of vertical handover was introduced. Vertical handover is not self-evident, which was explained in terms of IP addressing and organizational issues. Next, a theoretical comparison between vertical handover techniques was given. An experiment with one of the vertical handover techniques, namely Multipath TCP, was conducted and described.

In terms of (technical) infrastructure, this thesis has shown that there are several solutions available to facilitate vertical handover. Some of them require technical measures on both client and server, some of them only require the installation of an application on both client and server. Multipath TCP belongs to the first category, because a new kernel needs to be installed to support it. Furthermore, it is possible that, along the route from client to server, networking devices strip the multipath information from the packets, effectively breaking the solution. Mosh, on the other hand, belongs to the latter category. Apart from the installation of the application, there are no other requirements to use this solution.

In terms of (security) issues that arise when performing vertical handovers, a big issue has to do with IP addressing. Usually, a new connection is linked with the network interface used. When a mobile device switches from one network to the other, the network interface used switches and the IP address changes. Although the OSI model was set up to abstract from what happens at lower layers, various applications use information from lower layers, such as the initial IP address used. Web applications, such as forums, that depend on the IP address in terms of session handling, and `ssh` are examples of applications that break when the IP address changes. The second important issue that was identified concerned authentication in a multi-organizational network, as was seen in the pilot study in Utrecht. The solution that was shown involved the cooperation of each party participating in the network. This means that, in general, multi-organizational networks need the support of all parties involved. This may be difficult to accomplish.

In conclusion, the problem of vertical handover *can* be solved using different techniques. One of the solutions is Multipath TCP, which looks very promising, but may be difficult to deploy and get working properly. Another solution is Mosh, that works very good, although it (as of yet) lacks support for IPv6. Despite all these solutions, there are other problems that make vertical handover difficult, such as organizational challenges in terms of authentication and that applications bind to a specific network interface (and thus a specific IP address) when the initial connection is set up.

References

[1] bwm-ng (Bandwidth Monitor NG). http://www.gropp.org/?id=projects&sub=bwm-ng.

[2] Iperf - The TCP/UDP Bandwidth Measurement Tool. http://iperf.fr/.

[3] Missing IPv6 support - Issue # 81 - keithw/mosh. https://github.com/keithw/mosh/issues/81.

[4] Openvpn: Security overview. https://openvpn.net/index.php/open-source/documentation/security-overview.html.

[5] IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, 2012.

[6] Nickolay Amelichev and Kirill Krinkin. Simulation of 3G/WLAN offload: First steps. In *8th Conference of Open Innovations Framework Program FRUCT*.

[7] Nickolay Amelichev, Kirill Krinkin, SP Shiva Prakash, and TN Nagabhushan. Signal strength-based approach for 3G/WLAN handover on Nokia N900 devices. In *Proceedings of 10th Conference of Open Innovations Association FRUCT, Tampere, Finland*, pages 10–15, 2011.

[8] Olivier Bonaventure. Apple seems to also believe in Multipath TCP. http://perso.uclouvain.be/olivier.bonaventure/blog/html/2013/09/18/mptcp.html, sep 2013.

[9] P. Calhoun, R. Suri, N. Cam-Winget, M. Williams, S. Hares, B. O'Hara, and S. Kelly. Lightweight Access Point Protocol. RFC 5412 (Historic), February 2010.

[10] Personal communication with Fabian van den Broek, November 2013.

[11] Attilla de Groot. "5000 concurrent users on the #30c3 wifi.". Twitter, https://twitter.com/attilladegroot/status/417327876074393601, December 2013.

[12] EKR. Understanding the TLS Renegotiation Attack. http://www.educatedguesswork.org/2009/11/understanding_the_tls_renegoti.html, November 2009.

[13] Frédéric Firmin. The Evolved Packet Core. http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core.

[14] Youngbin Im, Hakyung Jung, Ji Hoon Lee, Wonjun Yoon, Ted "Taekyoung" Kwon, and Yanghee Choi. Vertical handovers in multiple heterogeneous wireless networks: a measurement study for the future internet. In *Proceedings of the 5th International Conference on Future Internet Technologies*, CFI '10, pages 10–13, New York, NY, USA, 2010. ACM.

[15] ITU. Considerations of horizontal handover and vertical handover. TSS COM19-C25-E, April 2007.

[16] Tae-sub Kim, Ryong Oh, Sang-Joon Lee, Suk-Ho Yoon, Choong-Ho Cho, and Seng-Wan Ryu. Vertical handover between LTE and Wireless LAN systems based on common resource management (CRRM) and generic link layer (GLL). In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, pages 1160–1166, New York, NY, USA, 2009. ACM.

[17] Jeffrey Michel. Evolved Packet Core diagram. Public domain, on http://en.wikipedia.org/wiki/File:Evolved_Packet_Core_Diagram.svg.

[18] Christoph Paasch. MultiPath TCP. GitHub, https://github.com/multipath-tcp/mptcp.

[19] Christoph Paasch. MultiPath TCP. YouTube, http://www.youtube.com/watch?v=VWN0ctPi5cw, apr 2012.

[20] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. Exploring mobile/WiFi handover with Multipath TCP. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, CellNet '12, pages 31–36, New York, NY, USA, 2012. ACM.

[21] Frans Panken. MOB-12-03 design document. http://www.surf.nl/kennis-en-innovatie/kennisbank/2013/rapport-design-document-mobility-between-4g-and-eduroam.html, July 2012.

[22] Masoumeh Purkhiabani and Ahmad Salahi. Enhanced authentication and key agreement procedure of next generation 3GPP mobile networks.

[23] Costin Raiciu, Christoph Paasch, Sébastien Barré, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. How hard can it be? Designing and Implementing a Deployable Multipath TCP. In *USENIX Symposium of Networked Systems Design and Implementation (NSDI'12), San Jose (CA)*, 2012.

[24] William Stallings. Mobile IP. *The Internet Protocol Journal*, 4(2):2–14, 2001.

[25] Cisco Systems. Configuring WDS, Fast Secure Roaming, and Radio Management. `http://www.cisco.com/en/US/docs/wireless/access_point/12.2_15_JA/configuration/guide/s15roamg.html#wp1035854`.

[26] Inc. Ubiquiti Networks. Technology Datasheet, Zero Handoff Roaming. `http://dl.ubnt.com/datasheets/unifi/UBNT_DS_Zero_Handoff_Roaming.pdf`.

[27] Ardian Ulvan, Robert Bestak, and Melvi Ulvan. Handover procedure and decision strategy in LTE-based femtocell network. *Telecommunication Systems*, 52(4):2733–2748, 2013.

[28] Iljitsch van Beijnum. Multipath TCP lets Siri seamlessly switch between Wi-Fi and 3G/LTE. `http://arstechnica.com/apple/2013/09/multipath-tcp-lets-siri-seamlessly-switch-between-wi-fi-and-3glte/`, sep 2013.

Appendix A: Sample trace of an MPTCP connection

A stripped[10] sample of `tcpdump -i eth0` with two standard MPTCP-enabled hosts:

```
16:24:35.858573 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp capable csum {0x444bd5b901d4f19e}]
16:24:35.858613 IP rijksdaalder.kassala.de.http > n138179.science.ru.nl.53878:
options [mptcp capable csum {0xdbbf1c6cf57fe93b}]
16:24:35.881812 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp capable csum {0x444bd5b901d4f19e,0xdbbf1c6cf57fe93b},
mptcp dss ack 2645686030]
16:24:35.882728 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss ack 2645686030 seq 2202377155 subseq 1 len 123 csum 0xed56]
16:24:35.882749 IP rijksdaalder.kassala.de.http > n138179.science.ru.nl.53878:
options [mptcp add-addr id 9 rijksdaalder.kassala.de,mptcp dss ack 2202377278]
16:24:35.882843 IP rijksdaalder.kassala.de.http > n138179.science.ru.nl.53878:
options [mptcp dss ack 2202377278 seq 2645686030 subseq 1 len 215 csum 0x6e47]
16:24:35.882910 IP rijksdaalder.kassala.de.http > n138179.science.ru.nl.53878:
options [mptcp dss ack 2202377278 seq 2645686245 subseq 216 len 231 csum 0x2e21]
16:24:35.906447 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss ack 2645686245]
16:24:35.906524 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss ack 2645686476]
16:24:36.052318 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss fin ack 2645686476 seq 2202377278 subseq 0 len 1 csum 0x621e]
16:24:36.052423 IP rijksdaalder.kassala.de.http > n138179.science.ru.nl.53878:
options [mptcp dss fin ack 2202377279 seq 2645686476 subseq 0 len 1 csum 0x1b5]
16:24:36.079364 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss ack 2645686476]
16:24:36.079456 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss ack 2645686477]
16:24:36.079472 IP rijksdaalder.kassala.de.http > n138179.science.ru.nl.53878:
options [mptcp dss ack 2202377279]
16:24:36.102142 IP n138179.science.ru.nl.53878 > rijksdaalder.kassala.de.http:
options [mptcp dss ack 2645686477]
```

[10]For readability's sake, a few standard TCP options have been stripped.