CHIANTI

Grant Agreement no.	**216714**
Project acronym	**CHIANTI**
Project full title	**Challenged Internet Access Network Technology Infrastructure**
Dissemination level	PU (public)
Document Date	2010-01-15
Deliverable Number	D5.2
Deliverable Name	**Final Trial Report**
Type	Report
Work package(s)	WP 5
Status & version	Public Report V1.0
WP / Task responsible	NDL
Main Author / Editor	Olaf Bergmann, Stefanie Gerdes
EC Project Officer	Paulo de Sousa
Keywords:	Lab tests, CHIANTI Proxy, Standardisation
Abstract	This report documents the results of the final user trials that have been carried out to evaluate the CHIANTI software for nomadic and mobile use. The trials did not only cover quantifiable tests in a controlled environment but also included the production environment on a commercial access platform that is operated by one of the project's participants. The standardisation and dissemination activities documented here include contributions to the Eiffel Think Tank as well as invited talks at the Keio University.

 © The CHIANTI consortium – 2008-2009

Contents

 © The CHIANTI consortium – 2008-2009

Acronyms and Abbreviations

ACRONYMS AND ABBREVIATIONS	Description
3G	3rd generation of mobile phone standards – including UMTS
CCTV	Closed Circuit Television; often used for surveillance purposes.
CHIANTI	Challenged Internet Access Network Technology Infrastructure
CHIANTI-P	CHIANTI specific protocol (here used as “place holder” prior to the completion of the CHIANTI architecture design phase)
CSP	CHIANTI Service Provider (Sphere)
DHCP	Dynamic Host Configuration Protocol
DP	Delay-tolerant Protocol – the CHIANTI protocol suite is being developed in three stages: *DP-Basic*, *DP-Enhanced* and *DP-DTN*.
GPRS	General Packet Radio Service – a packet oriented mobile data service, typically allowing data rates between 56 and 114 kbit/s.
HTTP	Hypertext Transfer Protocol
HTTP+X	HTTP protocol plus extended CHIANTI HTTP functionality
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
Sphere	Topological region of a network
TCP	Transmission Control Protocol
TV	Television
NCP	Network Control Protocol – the predecessor of today’s IP protocol
PEP	Performance Enhancing Proxy
UMTS	Universal Mobile Telecommunication System of the third generation of mobile phone standards
VPN	Virtual Private Network
WAN	Wide Area Network
WiMAX	Worldwide Interoperability for Microwave Access, based on IEEE 802.16, a standards-based technology enabling the delivery of last mile wireless broadband access
WLAN	Wireless LAN
WP	Work Package – refer to Annex I of the Grant Agreement for a description of the work package with the given number.

© The CHIANTI consortium – 2008-2009

List of Figures

 © The CHIANTI consortium – 2008-2009

Applicable Documents

Reference	Document Title, Version
[1]	CHIANTI Annex 1 – Description of Work FP7-ICT-2007-1-216714
[2]	CHIANTI D1.1 – Description of Use Cases and Scenarios
[3]	CHIANTI D1.2 – Operational and User Requirements
[4]	CHIANTI D2.1 – Protocol Analysis Report
[5]	CHIANTI D2.4 – Protocol Specification
[6]	CHIANTI D3.2 – System Architecture
[7]	CHIANTI D4.3 – Prototype III Implementation
[8]	CHIANTI D5.1 – Report on First Trial Results and Interim Report on Standardisation and Dissemination
[9]	J. J. Garrett: *Ajax: A New Approach to Web Applications*, 2005. http://www.adaptivepath.com/ideas/essays/archives/000385.php (last visited 2009-12-23).
[10]	Wei Li, A. W. Moore, M. Canini: *Classifying HTTP Traffic in the New Age*, in: ACM SIGCOMM 2008 (Poster session).
[11]	J. Nagle: *Congestion control in IP/TCP internetworks*, in: ACM SIGCOMM Computer Communication Review, 4 (14), 11–17, 1984.
[12]	Average web page size triples since 2003. Available at http://www.websiteoptimization.com/speed/tweak/average-web-page/

© The CHIANTI consortium – 2008-2009

1 Introduction

To evaluate the impact of CHIANTI on data communications in the mobile Internet and demonstrate its potential benefit to mobile users, several prototypes were developed in this research project. Every prototype implementation has been tested extensively in the lab and validated against the user requirements listed in deliverable 1.2 [3]. Prototype III eventually has been evaluated under real-world conditions in distinct usage scenarios, with a focus on nomadic use and the vehicular use case for mobile users.

This deliverable documents the results of Activity 5.3 (see Annex I, Section 1.3.5.6, of the Grant Agreement [1]) which is targeted on an empirical evaluation of the software system created in the CHIANTI project. Besides the in controlled environments to validate the quantifiable objectives specified in deliverable 1.2, the final prototype has been evaluated in a user trial on several high-speed trains in North England.

While most of the initial lab-tests that have been run against all delivered prototypes already have been documented in deliverable 5.1 [8], the results of the key tests that have been run with the final prototype in a controlled environment are included here for completeness. **Section 2** gives a brief summary of the lab tests and the validation against the key requirements the implementation must fulfil.

Section 3 then shows the results of the evaluation process. Here, appropriate evaluation methodologies have been selected according to the specific needs of the respective usage scenario that is targeted. A report on the final user trial including an analysis of the advantages that CHIANTI has in specific use cases concludes the final trial report.

The outcome of Activities 5.1 and 5.2 is summarised in **Section 4** on dissemination and standardisation.

Appendix A concludes this deliverable with a brief summary of the results of validating CHIANTI prototype III against the core requirements that have been specified in Appendix B of deliverable 1.2 [3].

2 Lab Tests and Quantifiable Objectives

To validate the basic functionality of the CHIANTI software, the delivered prototypes have been tested for robustness and stability in the lab and evaluated against the requirements specified in Appendix B of deliverable 1.2 [3]. To prepare for the final user trial, the basic disruption tests and the validation against the quantifiable objectives has been repeated with the final CHIANTI software. Special attention has been laid on the FlexProxy and its DP-Basic implementation to ensure proper operation during the final user trials in a production environment where the functioning was crucial for the revenue-generating Internet access service it was combined with.

The basic disruption tests have been performed as documented for prototype I in deliverable 5.1 [8]. For these tests and the following validation of quantifiable objectives, it is important to distinguish three fundamental types of disruptions an application can face:

Direct Access Network disruption: A mobile node has wireless Internet access using a local WiFi or 3G interface. An outage of the (wireless) access link may cause existing TCP connections to be terminated by the operating system (e.g., Microsoft Windows XP) because the interface is being removed from the internal network stack. In this case, even short disruptions of not more than 1 second will cause teardown of existing connections.

Network Hopping: When moving to another wireless hotspot or roaming between distinct mobile network providers, the node’s IP address may change and thus cause TCP connections to be terminated notwithstanding the continuous presence of the network interface on the operating system level. Technologies like mobile IP can be used to preserve the mobile node’s IP address while roaming through different networks with the overhead of additional communication between the user’s home network.

Disruptions between intermediate routers: If not affected by outages of the access link, data communication still can suffer from disrupted connectivity. The impact of this scenario often is less severe as local interfaces remain in service and thus TCP connection state will persist until TCP timeouts occur. As a result, small disruptions often are tolerated by the vanilla TCP stack. Yet, longer disruptions will cause, e.g., TCP connections to terminate. The concrete maximum duration of a disruption that is tolerated by a TCP stack depends on the actual implementation, timer settings, and even the connection state. For example, an idle TCP connection typically tolerates quite large disruptions, while a TCP connection waiting for an outstanding ACK will be terminated fast (e.g., within one minute). ICMP messages generated by (intermediate) routers also may cause fast termination of TCP connections.

While various technologies (e.g. mobile IP) exist to address the issues outlined here, none of them is suited to meet all requirements listed in deliverable 1.2 [3]. As the CHIANTI system was designed specifically with these requirements in mind, the prototype implementation has been validated against them. The following subsections show the key requirements R-4 and R-5 that are essential for the overall project goals. In addition, a brief description of the UDP tests is included here as UDP is not covered by the field trials.

2.1 Undisrupted Service Operation Across Disconnections

Requirement 4 targets at the uninterrupted service operation across disconnections that last more than five minutes. A user-initiated transaction such as sending or receiving email shall not be aborted by the local application or operating system. Changes that occur on the transport layer such as changed IP addresses after re-connect or different link characteristics induced by switching access technologies should not interfere with ongoing transactions.

 © The CHIANTI consortium – 2008-2009

The test setup is depicted in Figure 1. Here, the local user is equipped with Microsoft Windows XP and Outlook Express as email client. The SMTP and POP3 traffic is tunnelled through the CHIANTI service and captured on the server side using Wireshark.[1] Outages are emulated on the network link between the Chianti Client and Chianti Proxy as foreseen by the CHIANTI system architecture.

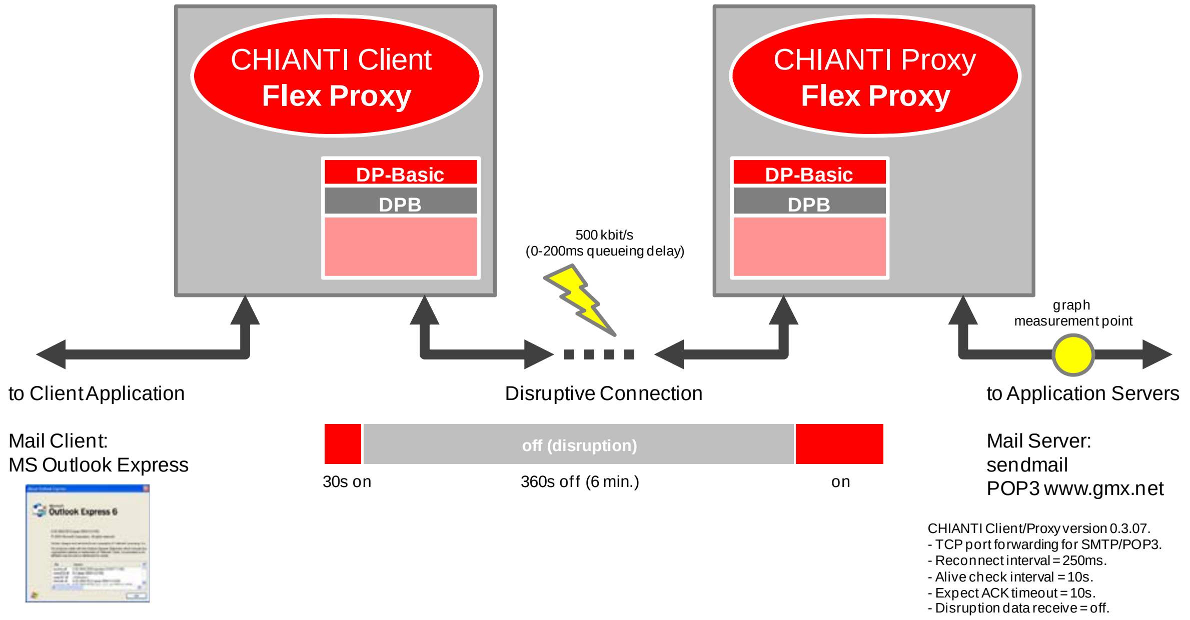

Figure 1: Test measurement setup – undisrupted service operation across disconnections.

Two tests have been performed: First, a 2,9 MByte mail attachment was sent to a remote mail server that was ensured to keep the connection state for at least six minutes. The second step was to download this mail via POP3 from a large German free-of-charge mail provider (gmx.net).

Sending Mail – SMTP: Reference measurement undisrupted

To ensure a valid test setup, a reference measurement without any link disruption was done before the actual tests were started. The transferred volume over time for the send transaction is shown in Figure 2.

[1] The capturing has to be done on the server side since Wireshark is not able to intercept traffic on the Windows loopback interface.

© The CHIANTI consortium – 2008-2009

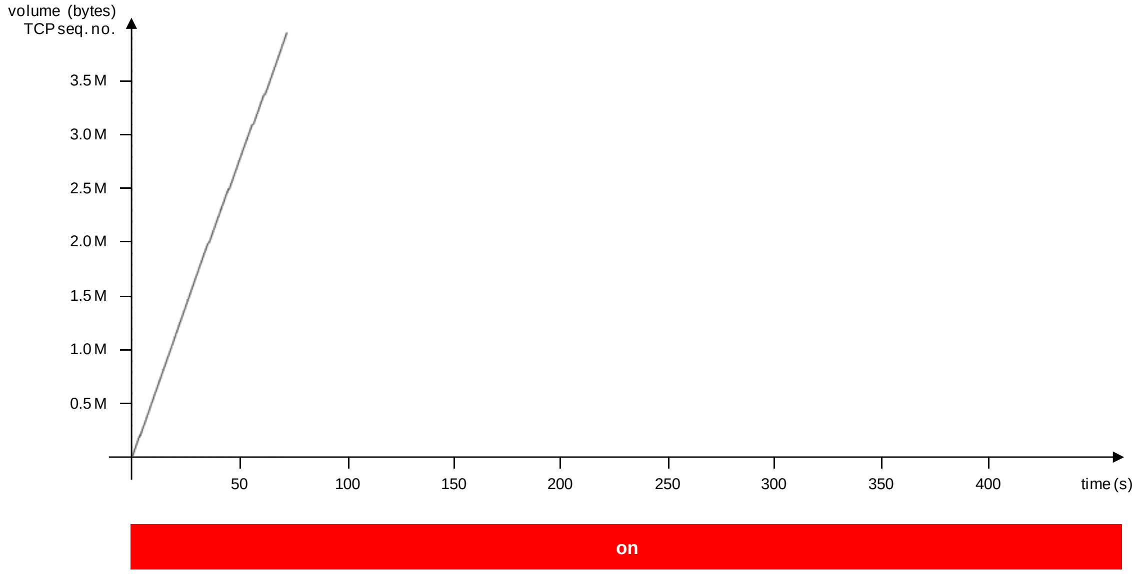

Figure 2: Sending email with SMTP – reference measurement undisrupted

Sending Mail – SMTP: With disruption direct without CHIANTI DP-Basic

The disruption test for SMTP over a plain TCP connection is shown in Figure 3. After a short period of seamless connectivity used to initiate the transmission of the mail, the link has been switched off for 360 seconds. After 96 seconds, the mail client has considered the send operation to have failed and displayed an alert box with an error message to the user. The client also has terminated the TCP connection as it has not received ACK messages from the SMTP server.

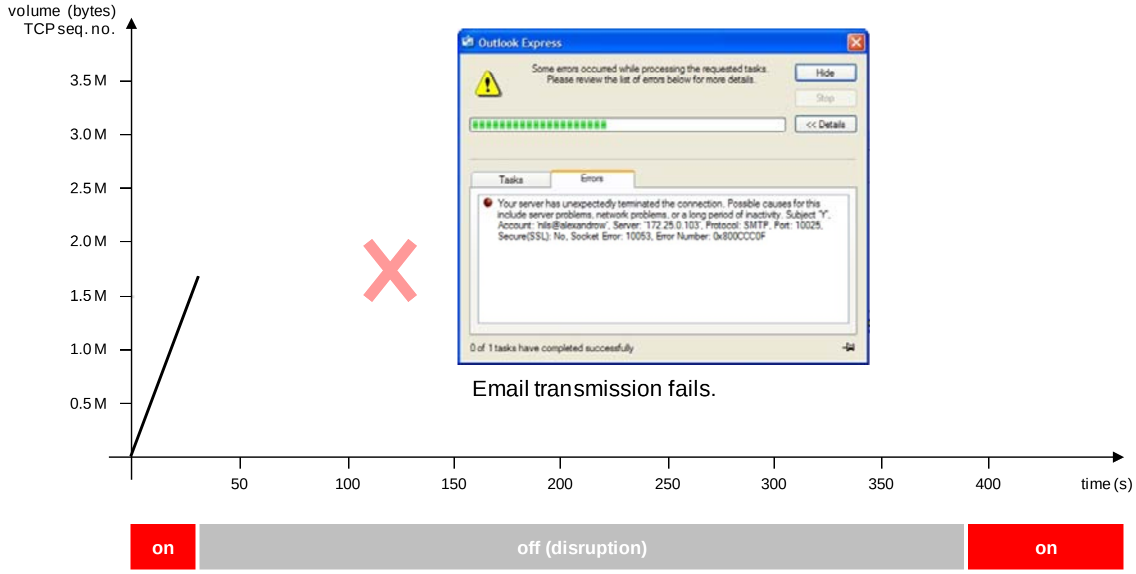

Figure 3: Sending email with SMTP – 6 min. disruption direct (without CHIANTI support)

Sending Mail – SMTP: With disruption via CHIANTI DP-Basic

Finally, the disruption test has been repeated with the CHIANTI service bridging the disrupted link. Figure 4 shows the continued data transmission after the outage. To inform the user of the outstanding acknowledgments, the mail client displays an informational message after a configurable timeout. The operation will continue unless the user decides to abort using this dialog box.

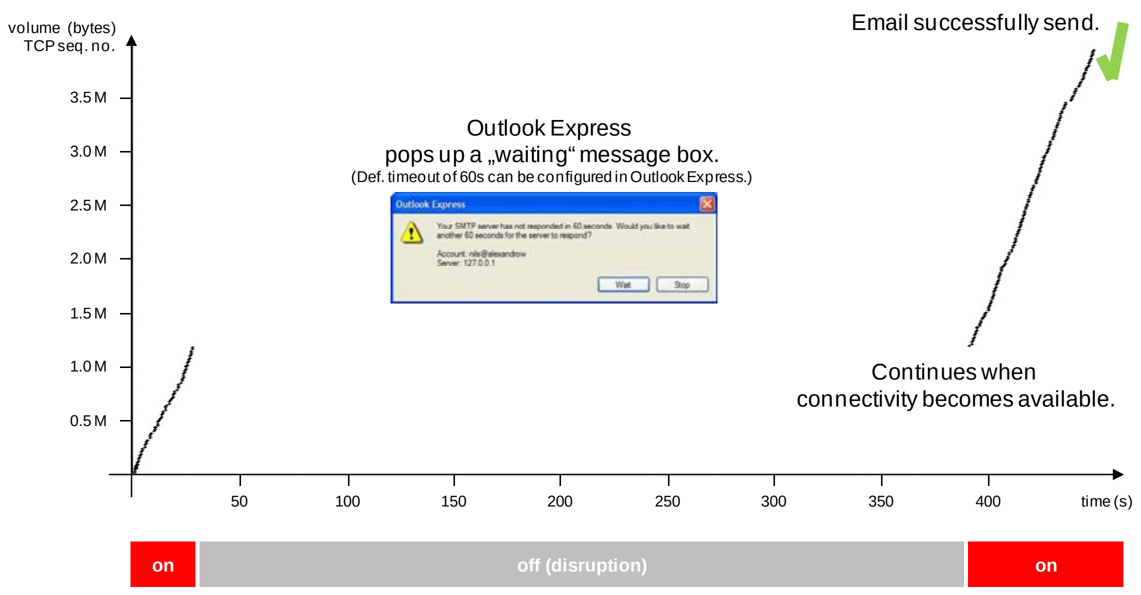

Figure 4: Sending email with SMTP – 6 min. disruption with CHIANTI DP-Basic support

Receiving Mail – POP3: Reference measurement undisrupted

Mail reception via POP3 was validated similar to the send transaction. First, a reference measurement has been performed without disruption to check the test setup for errors (see Figure 5 for the results).

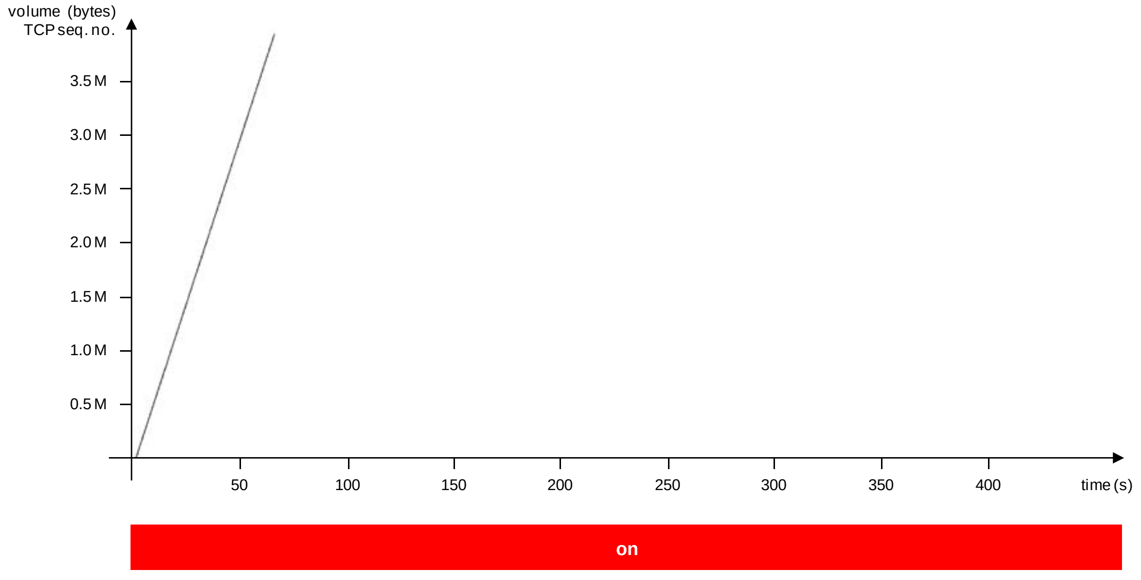

Figure 5: Receiving email with POP3 – reference measurement without disruption

Receiving Mail – POP3: With disruption direct (without CHIANTI support)

Without CHIANTI support, the mail client displays a notice that the download operation was deferred. In this case, the configurable timeout value was set to 300 seconds, the default is 60 seconds. The mail client does not notice that the connection was terminated by the remote POP3 server after 140 seconds and therefore continues waiting for data. As the mail client did not continue downloading the mail for some time after connectivity has come back, the test was stopped after 20 minutes (see Figure 6).

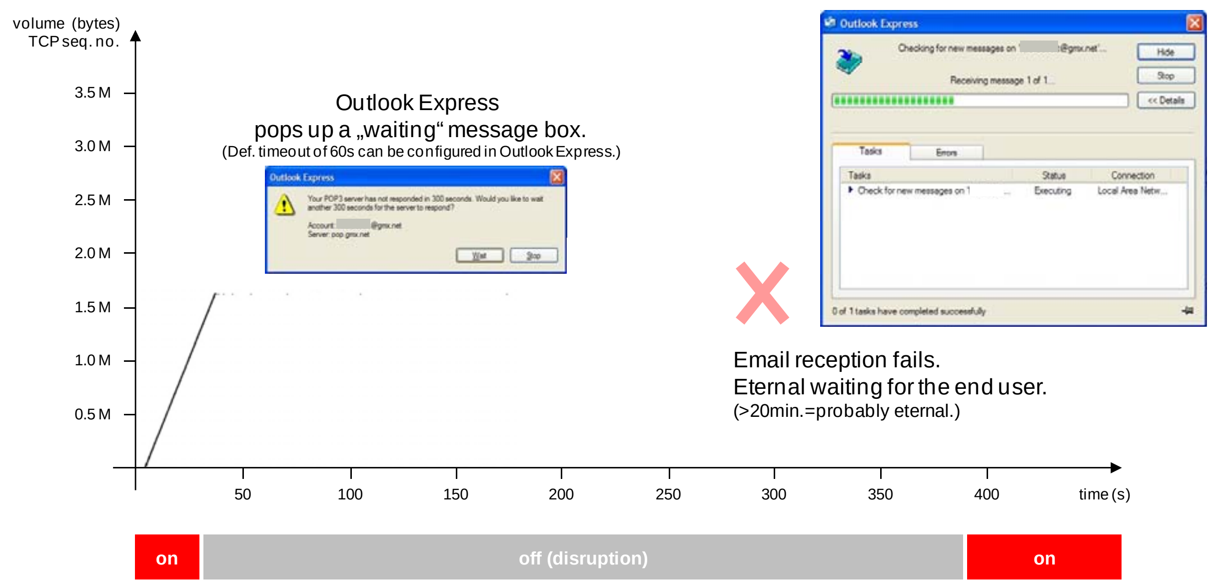

Figure 6: Receiving email with POP3 – 6 min. disruption direct (without CHIANTI support)

Receiving Mail – POP3: With disruption with CHIANTI DP-Basic support

Unlike the plain TCP test, the transmission continued after the 360 seconds link outage with the CHIANTI-enabled mail client. Figure 7 shows the same warning message box that was displayed by the mail client in the failed test without CHIANTI, but here, the client manages to recover and continue downloading the mail body as soon as network connectivity has come back.

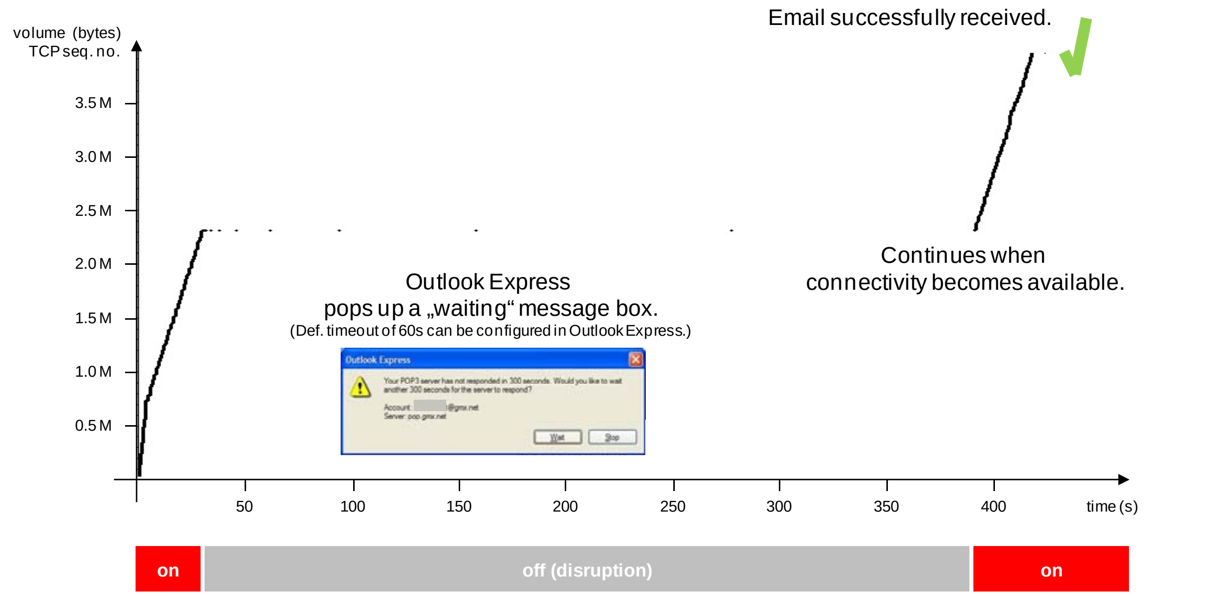

Figure 7: Receiving email with POP3 – 6 min. disruption with CHIANTI DB-Basic support

For completeness, these tests have been repeated successfully in a lab setup with the stand-alone DP-Enhanced reference implementation and a python POP3 implementation on a Linux host. As the plain TCP connection has recovered from the outage as well, an additional IP address change has been introduced in the test setup to demonstrate CHIANTI's extended capabilities.

The major issue in this setup is the use of a free-of-charge POP3 server that has an idle timeout of 60 seconds on the application layer. A reference test using a private POP3 server with a higher timeout value revealed that DP-Enhanced also meets Requirement 4 while the plain TCP connection was terminated when the IP address changed.

2.2 Throughput Optimisation for Disrupted Connectivity

Requirement 5 aims at the optimisation of throughput during intermittent connectivity. While the CHIANTI cross-layer signalling that is needed to detect link availability and to synchronise session state comes with a small overhead causing a slight performance degradation when network conditions are good, the throughput of user data is expected to increase by more than 30 % when all CHIANTI optimisations can be applied.

The test setup for this objective is shown in Figure 8. Again, Microsoft Outlook Express running on a Microsoft Windows XP notebook is being used on the mobile side. The test procedure comprises sending mail via SMTP and retrieval from a public free-of-charge mail server (gmx.net) using POP3. Fetching the mails from the server will take at least 5.5 minutes. During this period, the network link is periodically available for 30 seconds, followed by 30 seconds outage.

The traffic is intercepted for measurement at the server side of the CHIANTI system using Wireshark as described previously. The reference measurement to check the test setup is identical to the reference test without disruptions described in section 2.1.

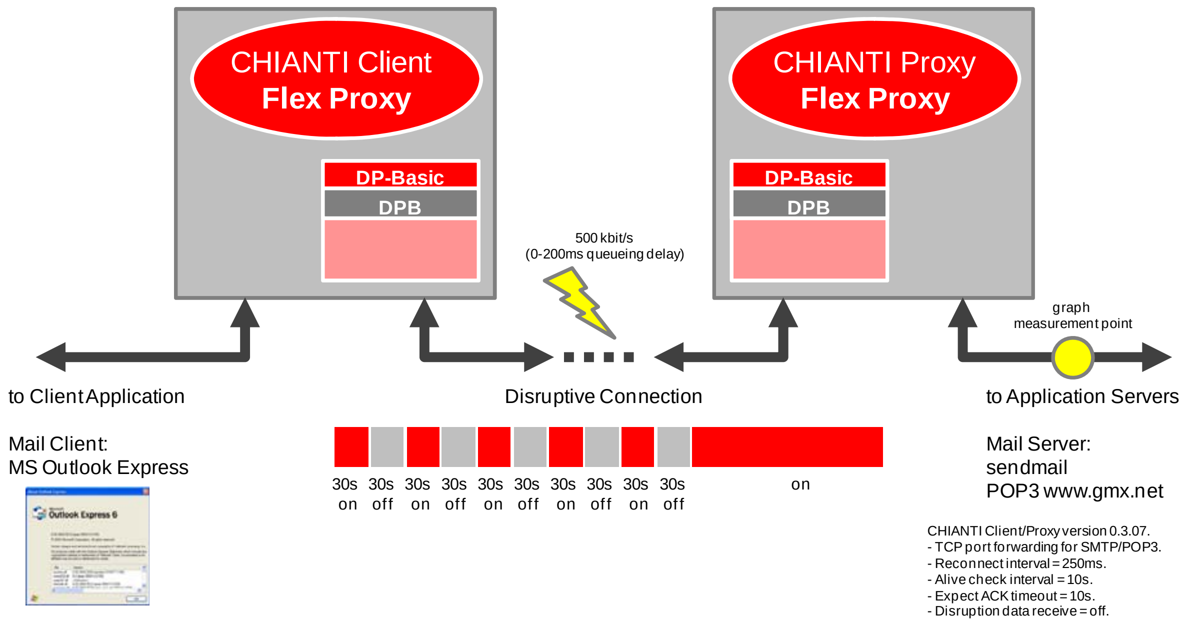

Figure 8: Test measurement setup – throughput optimisation for disrupted connectivity.

Sending Mail – SMTP: With disruption direct without CHIANTI DP-Basic:

The result of SMTP transmission over plain TCP on a disrupted link is depicted in Figure 9. After 107 seconds, the user is notified of a failure during transmission of the mail. The client has terminated the TCP connection as no ACKs have been received for some time.

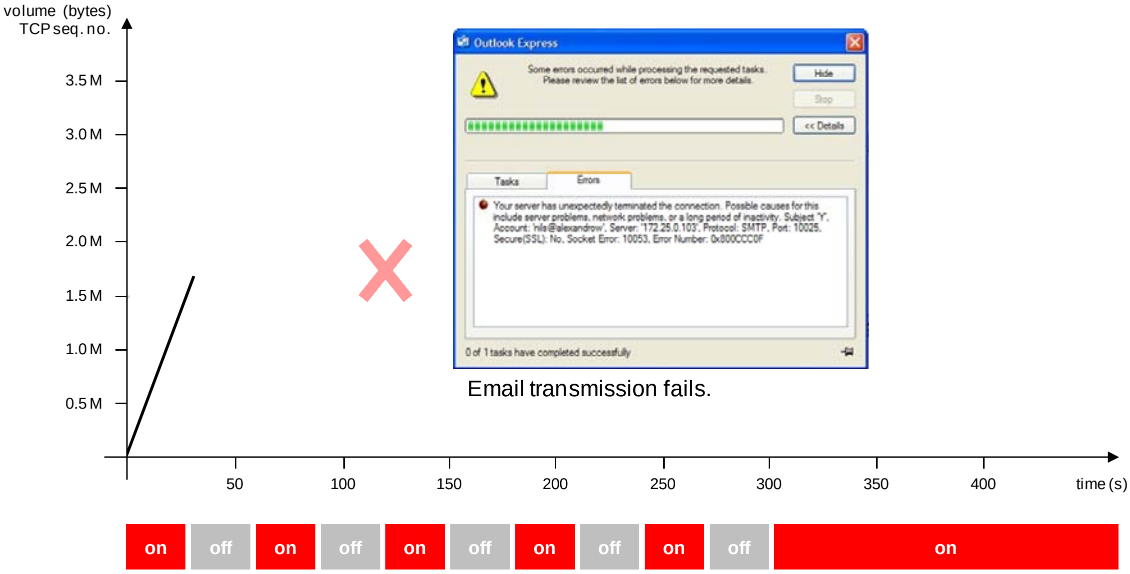

Figure 9: Sending email with SMTP – 30s on/off direct (without CHIANTI support)

Sending Mail – SMTP: With disruption with CHIANTI DP-Basic:

With CHIANTI bridging the disrupted link, the transmission continues instantly after the end-to-end path is reconnected after the 30 second outages. Since the reconnection is fast enough to utilise every 30 second slot where the link is up, the alert box that has been observed during the disruption tests in section 2.1 is not shown. A graphical overview of the transmission process is given in Figure 10.

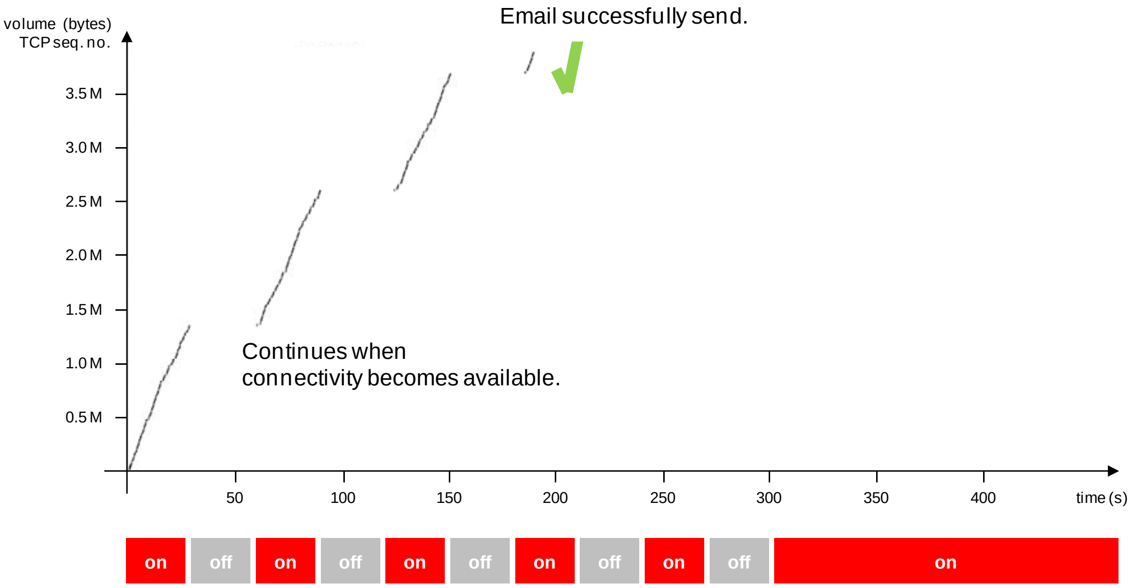

Figure 10: Sending email with SMTP – 30s on/off with CHIANTI DP-Basic support

Receiving Mail – POP3: With disruption direct without CHIANTI DP-Basic

For mail retrieval over POP3, the notebook has been equipped with a local WiFi interface resembling the nomadic user scenario (optionally, a 3G network card could have been used as well). Here, the interface is removed from the internal network stack when the operating system has detected a link failure (cf. Deliverable 5.1 [8]).

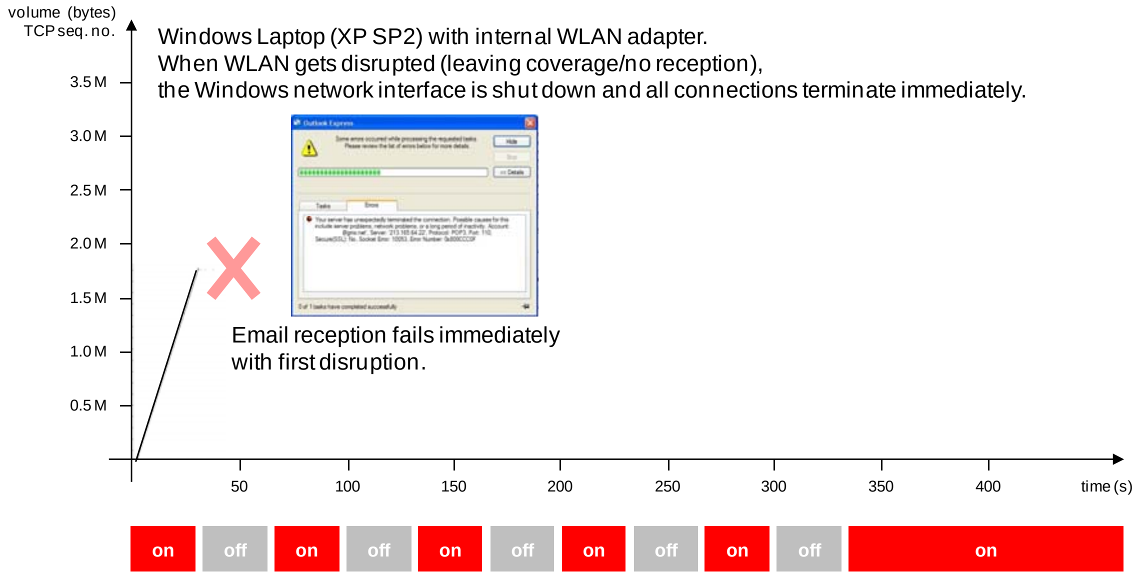

Figure 11: Receiving email with POP3 via laptop-WLAN – 30s on/off direct (without CHIANTI)

The results of the test using POP3 over plain TCP are shown in Figure 11. The mail retrieval stops immediately when the link goes down for the first time and does not recover as the network interface has been removed by the operating system. An error message is being displayed by the mail reader to inform the user of the aborted transmission.

In cases where the disruptions do not occur on the access link but somewhere else on the path between the mobile user and the remote POP3 server, the local WiFi interface will not get removed and thus the mail reader manages downloading the entire mail as shown in Figure 12. An alert box is displayed after some time to inform the user that the server is not responding. When not aborted by the user, the operation likely succeeds after some time.

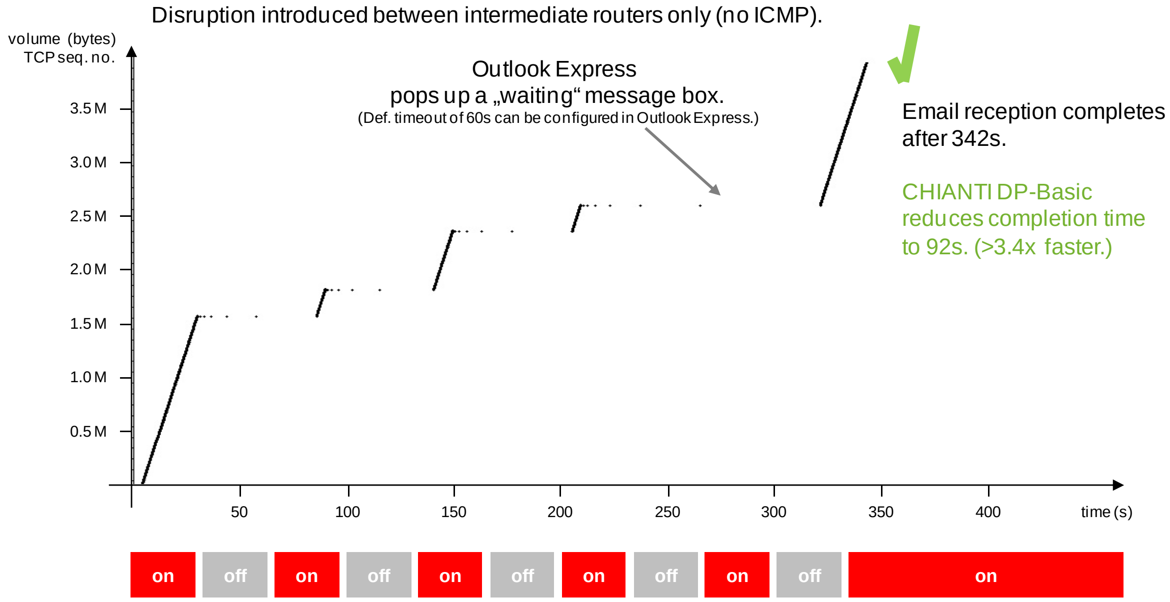

Figure 12: Receiving email with POP3 disruption routers – 30s on/off direct (without CHIANTI)

Note that the disruptions cause high round-trip times that induce higher TCP retransmission timers. As a result, the downloading process does not continue immediately after connectivity has come back, leading to a bad utilisation of available bandwidth. In this reference test, mail retrieval was complete after 342 seconds.

Receiving Mail – POP3: With disruption with CHIANTI DP-Basic

The advantage of fast link detection is obvious in the disruption test using CHIANTI, see Figure 13. Here, the data transmission starts immediately after the first outage has ended, resulting in a completion time of 92 seconds.

Despite missing statistical significance, these measurements indicate a decent performance gain due to optimised link detection. Because of TCP's conservative calculation of retransmission timers to avoid overloading of congested link segments, the CHIANTI service can achieve the targeted 30 % throughput enhancement.

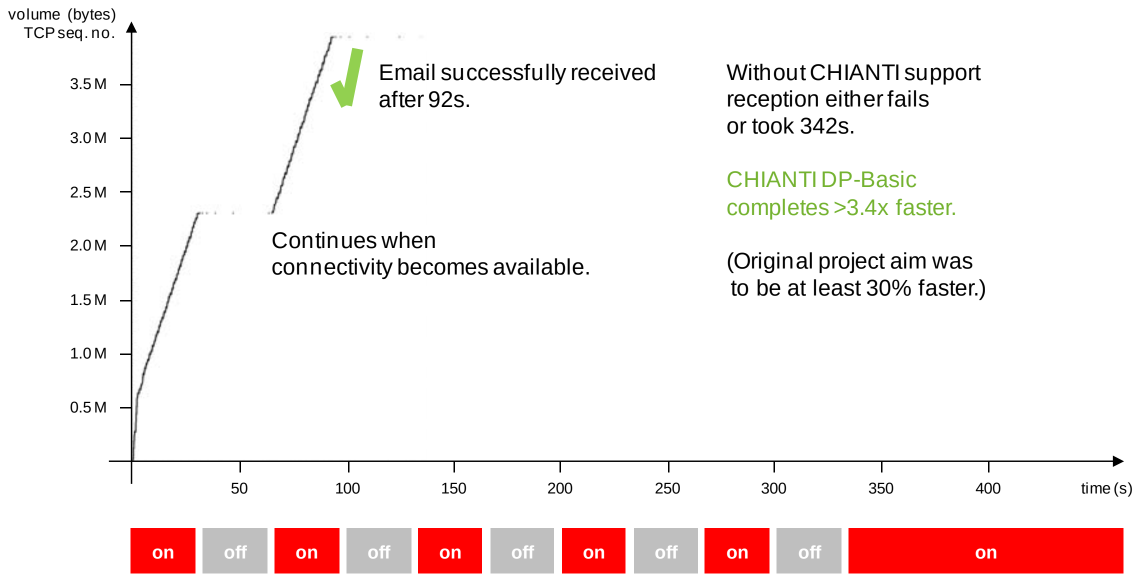

Figure 13: Receiving email with POP3 – 30s on/off with CHIANTI DB-Basic support

2.3 Maintaining Lossless Data Streaming

Requirement 6 targets at specific media streaming applications such as CCTV where real-time requirements are somewhat relaxed compared to interactive applications. Unlike those, CCTV relies on loss less streaming of video data. The formal requirement therefore states that the CHIANTI service has to enable bridging of disconnections that can last up to five minutes for video streaming. This requirement is checked for DP-Basic only, as the release version of DP-Enhanced does not yet support UDP. DP-DTN comes with support for delay-tolerant streaming, i.e. the streaming module provides support for caching of streamed media data and local play back during network outages.

 © The CHIANTI consortium – 2008-2009

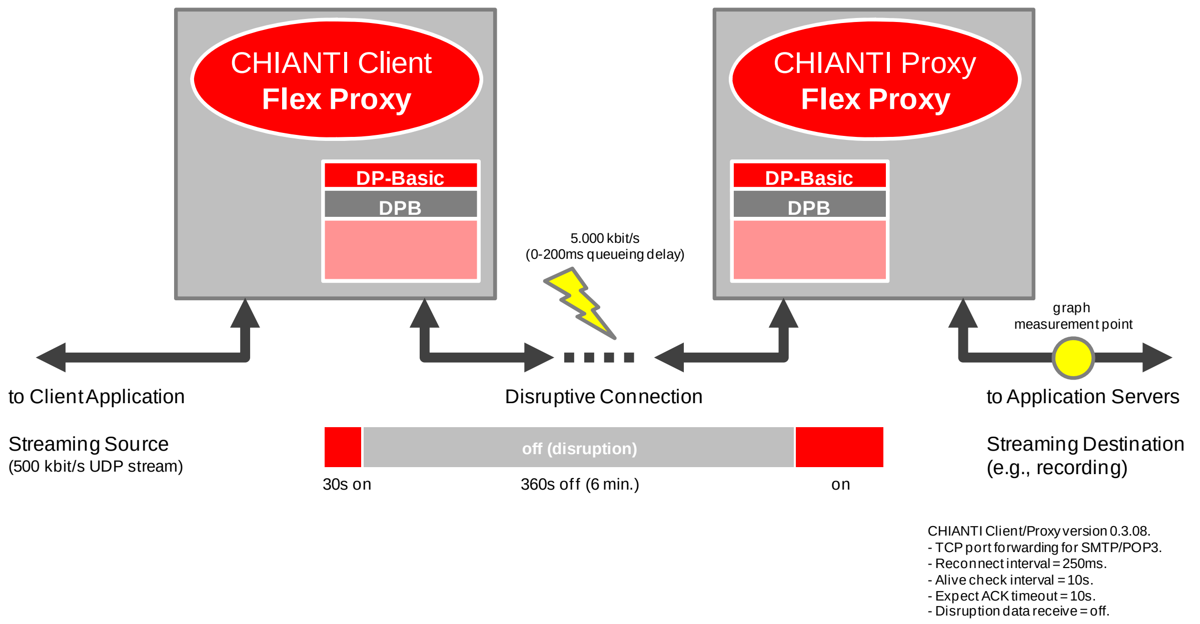

Figure 14: Test measurement setup – maintaining lossless data streaming.

The test setup for this objective is depicted in Figure 14. For a controlled test, a packet generator has been used to send UDP packets with a constant rate of 500 kbit/s. The bandwidth of the disrupted link has been set to 5 Mbit/s to speed up transmission of old packets that have been cached at the server during a network outage. Otherwise, the backlog at the server would grow with every disruption, eating up significant server memory.

The disruption test starts with a short period of connectivity to begin the streaming process. After 30 seconds, the link is shutdown for 360 seconds, and eventually turned on again.

UDP streaming data incoming at server: Reference Measurement undisrupted

The reference measurement is shown in Figure 15. The test stream is sent for 420 seconds. The graph shows a continuous packet flow of 500 kbit/s, with a small number of peaks that are related to timing issues at the sender.

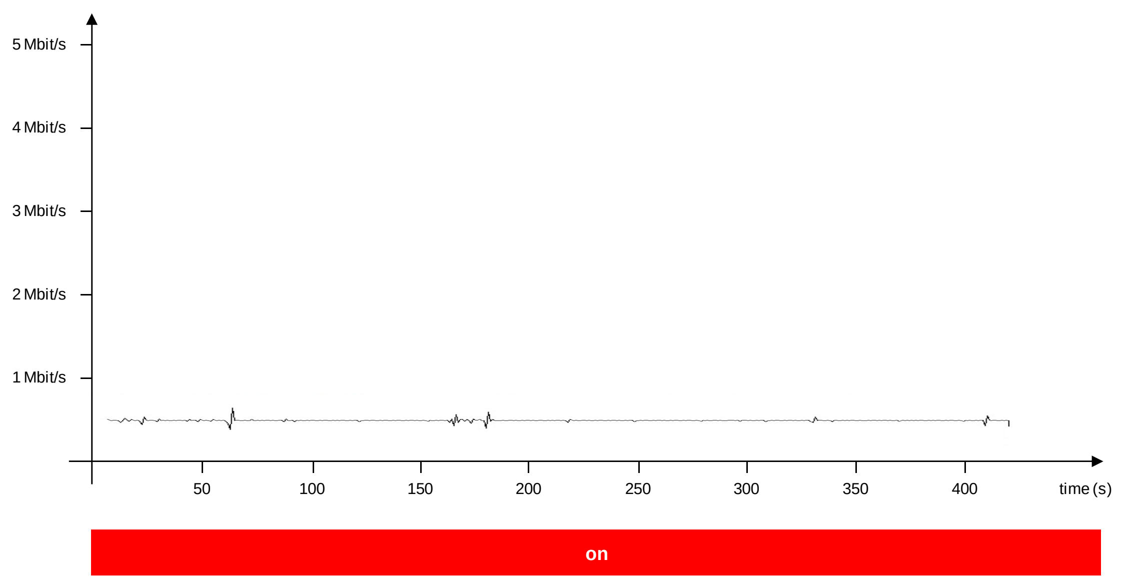

Figure 15: Streaming data UDP – reference measurement undisrupted

UDP streaming data incoming at server: With disruption direct without CHIANTI

As denoted by Figure 16, the direct packet flow between streaming server and mobile client stops entirely during the network outage as the UDP datagrams are discarded silently at the last hop before the disruption. When the link has come back, the stream again is received on the mobile client.

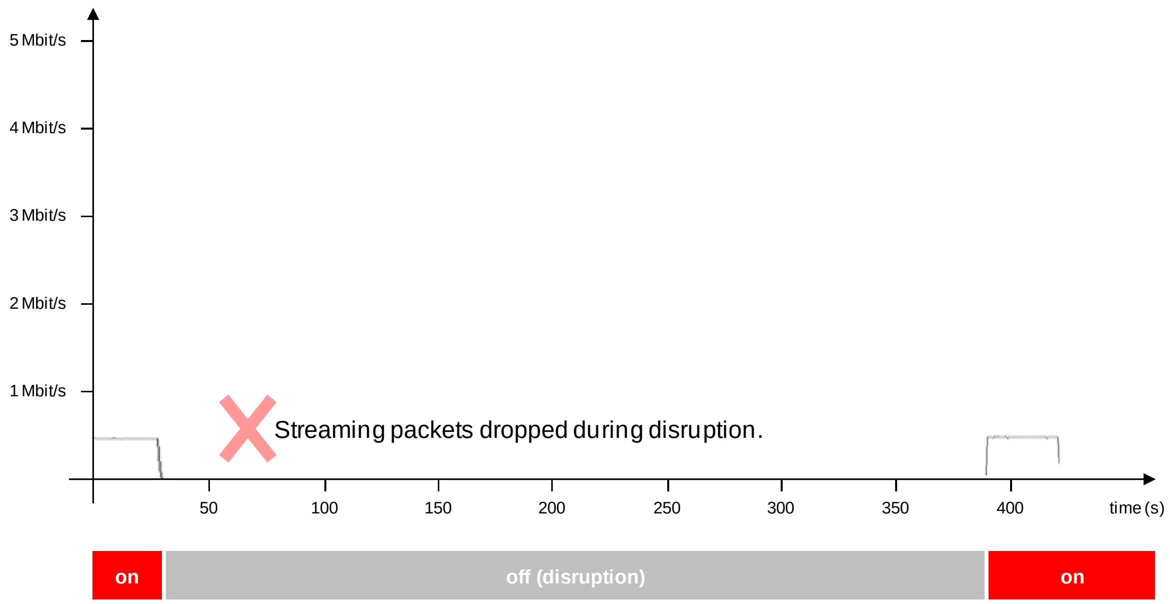

Figure 16: Streaming data UDP – 6 min. disruption direct (without CHIANTI support)

UDP streaming data incoming at server: With disruption with CHIANTI DP-Basic

With CHIANTI, the packet flow also stops at the mobile client during the outage. As the packets still arrive at the CHIANTI Proxy, they are buffered and transmitted are wire-speed when the link comes back. Figure 17 shows the effect of this bulk transmission after 390 seconds. Here, the sending of the cached UDP datagrams took about 50 seconds while new packets have been streamed for 30 seconds (as the entire stream took 420 seconds).

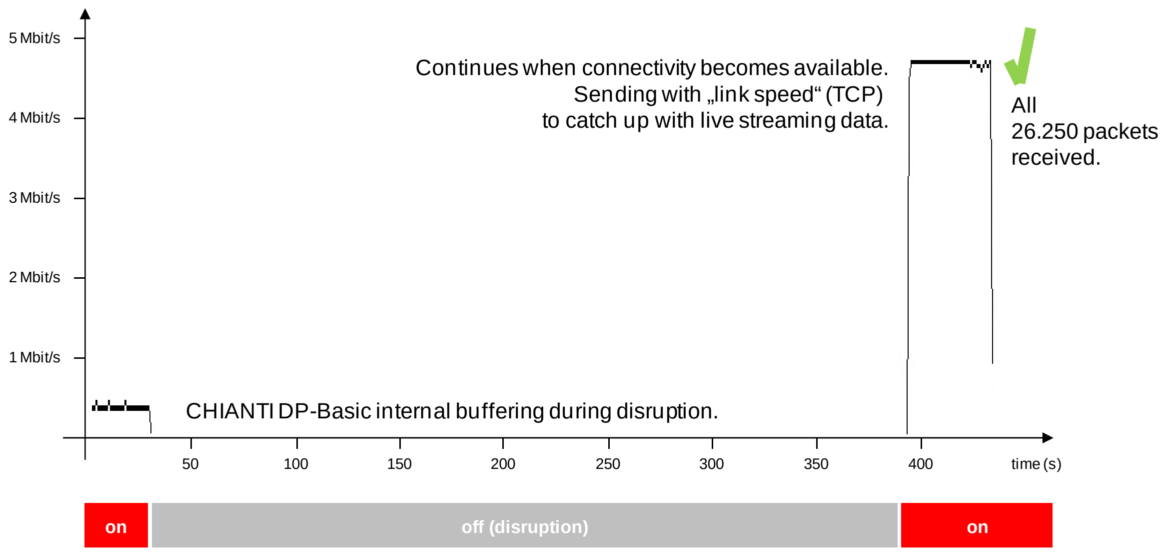

Figure 17: Streaming data UDP – 6 min. disruption with CHIANTI DP-Basic support

This bulk transfer also shows the major difference between the CCTV usage scenario and media streaming for interactive applications or entertainment. While CCTV acts more like a recording engine, real-time streaming requires accurate timing for the playout of packets. This is shown by the CHIANTI project as a proof-of-concept for delay-tolerant networking using DP-DTN.

© The CHIANTI consortium – 2008-2009

3 Final User Trial

This section documents the results of the prototype evaluation that has been done in several real-world trials, in particular a user trial on several high-speed train routes in North England in December 2009. The trials were targeting at the demonstration of the impact that the CHIANTI service has on the major user scenarios that were defined in the beginning of the project, i.e. the nomadic use case and the mobile usage scenario.

3.1 Nomadic Use

The Nomadic User support scenario was described in WP1 [3]. In this scenario, the CHIANTI components are implemented directly on at least one of the devices a user is carrying, e.g. a notebook or mobile phone. Thus the user is not dependent on the surrounding infrastructure to use CHIANTI.

Unlike the vehicle support scenario the Nomadic User scenario allows the user to move between different environments without losing the connection to the CHIANTI infrastructure. It is thereby possible to e.g. leave a train and enter a car while still being connected to the CHIANTI client.

On the other hand, the installation of the software on the user's device means some additional effort. Not every mobile device may be able to meet the requirements for that, e.g. provide for a SOCKS implementation. The use of an existing CHIANTI infrastructure circumvents this problem.

The behaviour of the CHIANTI software within the Nomadic User Scenario has been observed during various tests. According to Eurostat the main means of transport in Europe are passenger cars followed by busses and trains. Beyond that, a German study of the Bundesministerium für Verkehr states that another main form of transportation is walking. Thus, the trials for the Nomadic User scenario cover pedestrian usage as well as in-car and train tests.

3.1.1 Pedestrian Usage

The pedestrian nomadic usage scenario addresses two aspects:

1. Benchmarking the performance of the CHIANTI software systematically under different controlled nomadic conditions to understand how the CHIANTI system compares to plain (i.e., non-enhanced) Internet access. This investigation is carried out in different settings using different object sizes for download.
2. Running CHIANTI in a nomadic environment utilising various WLAN hot-spots in the city of Espoo and moving between them in random patterns with short online access in each of them. This investigation is carried out using web access scripting for representative object size download from a local web server.

All these evaluations are carried out with DP-Basic. The evaluation of DP-Enhanced and DP-DTN is limited to a functional validation and thus does not involve close-to-real-world evaluation.

3.1.1.1 Trial Setup

The nomadic trial setup for pedestrian use is depicted in Figure 18. Being one representative for nomadic scenarios, the CHIANTI client and the user software run on the same machine: an Apple MacBook with an Intel Core 2 Due 2.4 GHz processor and 4 GB RAM running MacOS 10.5.8. The CHIANTI proxy runs on a SuperMicro rack server with two AMD quad core processors and 8 GB RAM, running Linux 2.6 connected to the TKK Comnet experimental

© The CHIANTI consortium – 2008-2009

laboratory network (openly accessible from the outside). The resources downloaded by the client are served from the TKK Comnet production web server supplied with a set of files dedicated to the trial (see below), connected to the TKK Comnet production office network and firewalled from the open Internet.

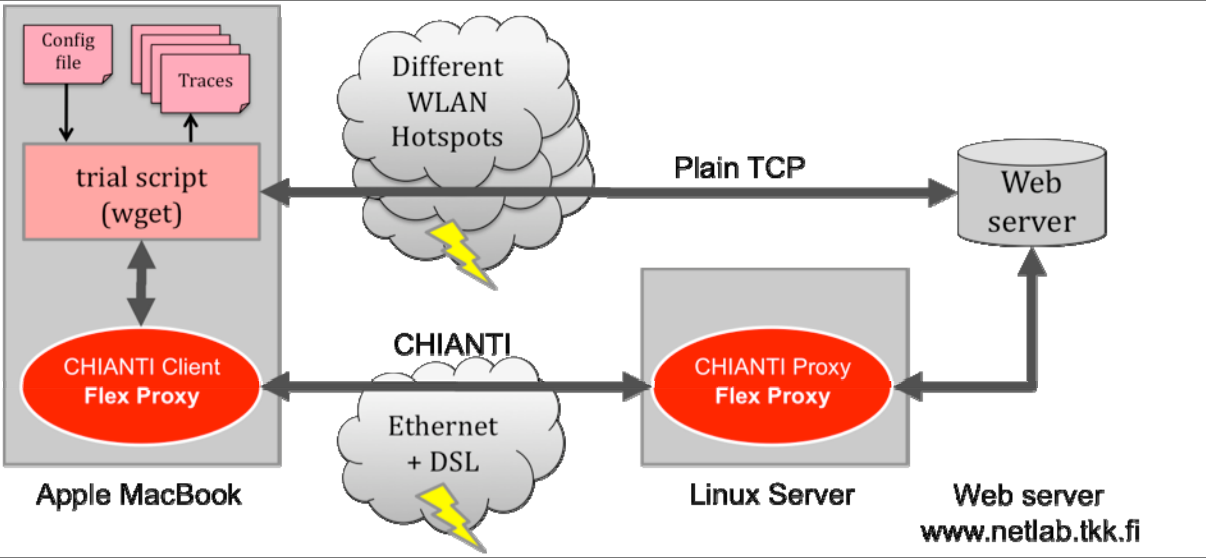

Figure 18: Nomadic trial setup

The client uses `wget` to emulate HTTP requests. The `wget` requests can be either directed to use the CHIANTI client (“CHIANTI”) – by means of setting the “http_proxy” environment variable to point to the local Flex Proxy – or to connect directly to the web server (“Plain TCP”) by resetting the variable; note that the setting “-proxy=[on/off]” alone is insufficient. `wget` is called from a measurement script shown below:

```
#!/bin/sh
# Usage: do_wget <resource-file> [<testname>]
rm -rf download.plain/*
rm -rf download.chianti/*

resource_file=$1.res
shift
if [ $# -eq 1 ]
then
	test=$1
fi

WGET_OPTS="-t inf -N -e robots=off -U 'moz' --delete-after --timeout=300 -retry-connrefused"

unset http_proxy
wget -proxy=off -o log/wget.plain.$test.log -i $resource_file -P download.plain $WGET_OPTS &

export http_proxy=127.0.0.1:58000
wget -proxy=on -o log/wget.chianti.$test.log -i $resource_file -P download.chianti $WGET_OPTS

wait
```

`wget` is configured to act as a Mozilla browser (-U 'moz'), to retry downloading any object indefinitely (-t inf), to provide timestamps for later evaluation (-N), and to delete files after downloading (--delete-after). Logging is turned on to provide trace files (-o) and different download directories (that are emptied prior to every run) are defined for Plain TCP and CHIANTI-based operation (-P).

Finally, a resource file is supplied from which `wget` reads the resources in the order they are to be download (-i) and the timeout after which `wget` aborts a download can be various (--timeout). We enforce `wget` to not give up if temporarily no path can be found to the destination (--retry-connrefused).

The measurements for *Plain TCP* and *CHIANTI* can be run in parallel in which case the two commands get synchronised (using wait) before the script completes so that the script can be started from within a loop iterating through a number of measurements. This case is shown above. This mode of operation is referred to as "parallel". Alternative, the two commands can be run in sequence so that the CPU and network capacity are exclusively available. This mode is referred to as "serial".

We use an enhanced variant of this script to control the routing table of the laptop and alternating remove and add the *default route*, thus emulating network disruptions in a controlled way. (We use this approach instead enabling/disabling the respective network interface using `ifconfig` because the latter turned out to cause system instabilities when issued frequently.)

The `wget` command each generates a log file with timestamps, shows the time to complete each download, and provides a calculation of the mean observed download bit rate. This information is extracted from the log files by offline processing. For this offline processing, number of shell, *sed(1)*, and *awk(1)* scripts are being used to extract and evaluate the data. For visualization, we use *gnuplot(1)* 4.5 patchlevel 2.

3.1.1.2 Test Methodology

As noted above, we perform two types of measures: We conduct a controlled study of protocol capabilities as the nomadic user would see them, i.e., a setup with fairly stable wired/wireless connectivity alternating with turned off devices and no connectivity. And we then take the CHIANTI software into a real-world nomadic scenario.

<u>Controlled nomadic CHIANTI experiments</u>

The CHIANTI system undoubtedly introduces overhead into the communication of an end user as two more entities – the Flex Proxies – are added to the data path. This addition means that additional entities need to handle the data. Furthermore, we may see suboptimal routing, no longer following the shortest IP path. Moreover, the CHIANTI system uses a minimal additional protocol that leads to some overhead in terms of communication capacity and one additional handshake. Finally, the extra components require CPU cycles for processing and thus may introduce additional limitations. All these aspects potentially lead to increased latency and thus reduced performance.

The biggest gain of the CHIANTI system is a more agile and reactive operation with respect to disruptions: preventing applications from noticing them, bridging disconnections efficiently, and recovering faster than TCP would do.

The **primary metric** of interest for the CHIANTI system operation is thus the observed **delay** that are incurred when completing (interactive) operations with a remote peer, e.g., when accessing a web page.

An indirect measure for delay is the achievable **mean application throughput** put when interacting with a remote party.

 © The CHIANTI consortium – 2008-2009

In our controlled nomadic experiments, we investigate the impact of multiple variables on these two performance metrics to be able to characterise the operation of the CHIANTI system under relatively static conditions as they prevail in nomadic scenarios:

a) We investigate the impact of different on/off times, i.e., periods of connectivity and disconnection. We use permanently connected systems as one reference to determine the achievable performance. In addition, we vary the connection/disconnection periods between 30s and 600s.

b) We examine the impact of the object size to be retrieved. For this purpose, we choose permanent connectivity and vary the object size exponentially between 1KB and 64MB.

c) We determine the impact of the CHIANTI and Plain TCP measurements running in parallel and sequentially for different object sizes as per b).

For the measurements in a), we use a file size distribution with a mean roughly equivalent to what has been found as mean page size on the web for 2007 [12]: slightly more than 300 KB. We generated file sizes using an exponential distribution with a mean of 273 KB and then rounded the resulting value up to the next larger file out of a set of 17 static files available at the web server. These files have exponentially increasing sizes: 1KB, 2KB, 4KB, …, 16MB, 32MB, 64MB. Sizes larger than 64 MB are mapped to the largest file. This simplified process intentionally results in a bias toward slightly larger downloads, yielding mean of 388 KB file size 10,000 objects (slightly more than the 2007 size since the trend towards larger objects appears to continue).

We created resource files for 10, 100, 1000, and 10 000 objects, yielding a total download volume between 3.6 MB and 3.88 GB. Below is an excerpt of a resource file used with `wget` with 1,000 object references:

```
…
http://www.netlab.hut.fi/~jo/misc/512K
http://www.netlab.hut.fi/~jo/misc/128K
http://www.netlab.hut.fi/~jo/misc/32K
http://www.netlab.hut.fi/~jo/misc/256K
http://www.netlab.hut.fi/~jo/misc/32K
http://www.netlab.hut.fi/~jo/misc/1M
http://www.netlab.hut.fi/~jo/misc/256K
http://www.netlab.hut.fi/~jo/misc/256K
http://www.netlab.hut.fi/~jo/misc/512K
http://www.netlab.hut.fi/~jo/misc/512K
…
```

For the measurements in b) and c), we use repeated downloads of individual of the 17 files mentioned above. For this purpose, we created a resource file per file size with 10 entries, yielding download volumes between 10 KB and 640 MB.

Nomadic Hot-spot CHIANTI experiments

For the nomadic hot-spot measurements, we use the above setting with 10,000 entries to make sure that the retrieval process does not complete prematurely before all hot spots are visited. We used four different hot-spots:

a) the public Aalto wireless network at TKK,

b) a protected lab network at TKK Comnet,

c) the public WLAN network at Helsinki Institute for Information Technologies (HIIT), and

d) a private WLAN network connected via the Finnish ISP elisa.

Access networks a), b), and c) are connected via the university network and Finnish university backbone (for c) at high data rates. Access network d) uses DSL with 8 Mbit/s downlink and 1 Mbit/s uplink.

Some measurements were carried out only involving switching between a), b), and c). For those measurements where d) was involved, measurements either started or ended at d) and the other networks were visited in between.

In all cases, the laptop was suspended while connected to one network, then moved to the next network where operation was resumed. Moving between networks a) and b) involved moving along the corridors inside the TKK ECE building in Otakaari 5 in Espoo, Finland. Moving towards c) and d) required driving by car or bus.

These experiments are carried out only a limited number of times and repeated runs did not follow a strict protocol. The goal was not to obtain statistically significant data, but rather to obtain qualitative insights into the operation of the system.

3.1.1.3 Results

This section documents the results of the evaluation of the CHIANTI service for the nomadic use case. The tests in a controlled environment are documented in section 3.1.1.3.1, followed by the findings of the exemplary roaming between distinct hotspots in section 3.1.1.3.2.

3.1.1.3.1 Controlled Experiments

To characterise the communication capabilities of the CHIANTI system, we carried out a set of experiments with controlled up/down times (also referred to as “on/off” intervals). The first set of measurements is carried out via a DSL link. The results are shown in Figure 19 through Figure 24. For each experiment, we downloaded 1000 files of random size (see above) and observed how quickly the download completes. In the following figures, we plot the elapsed time on the X axis and the degree of completion, i.e., the cumulative fraction of files already received. The figures show the mean results over ten runs – and therefore, there are not always plateaus clearly visible (and if so, they are not strictly horizontal) during disconnection periods.

 © The CHIANTI consortium – 2008-2009

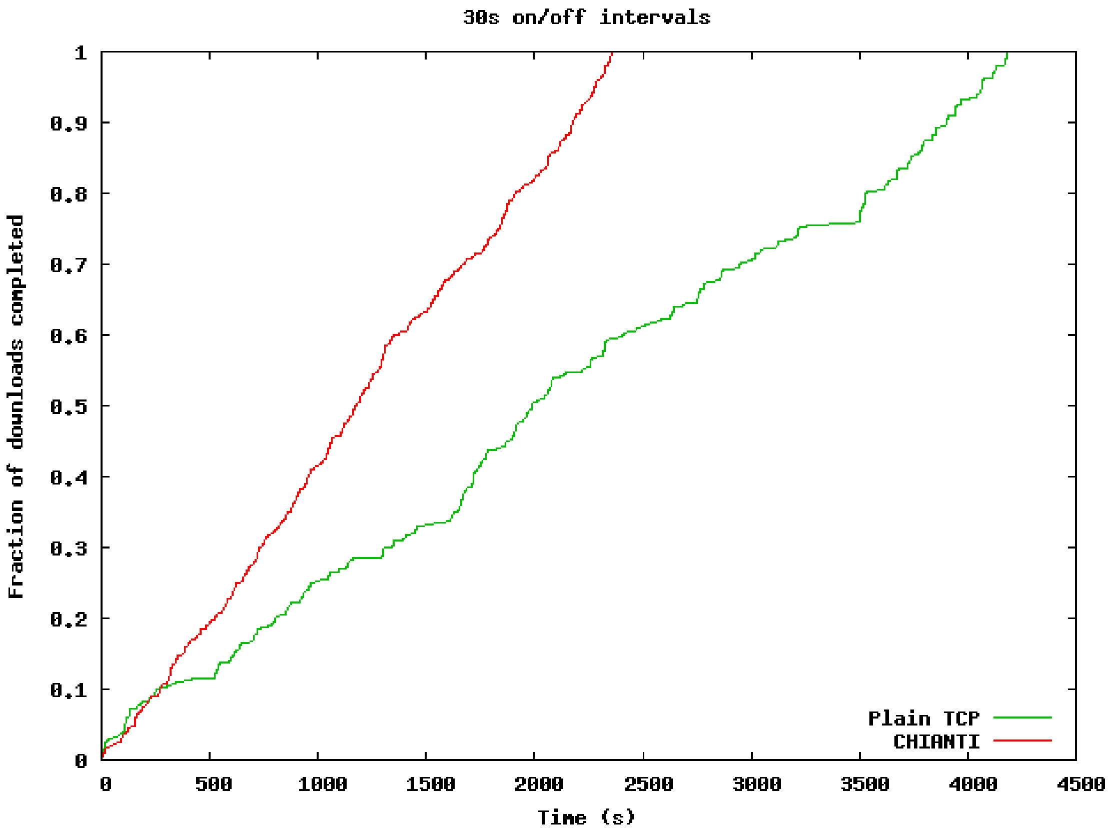

Figure 19: Parallel download, CHIANTI vs. Plain, on/off time 30s

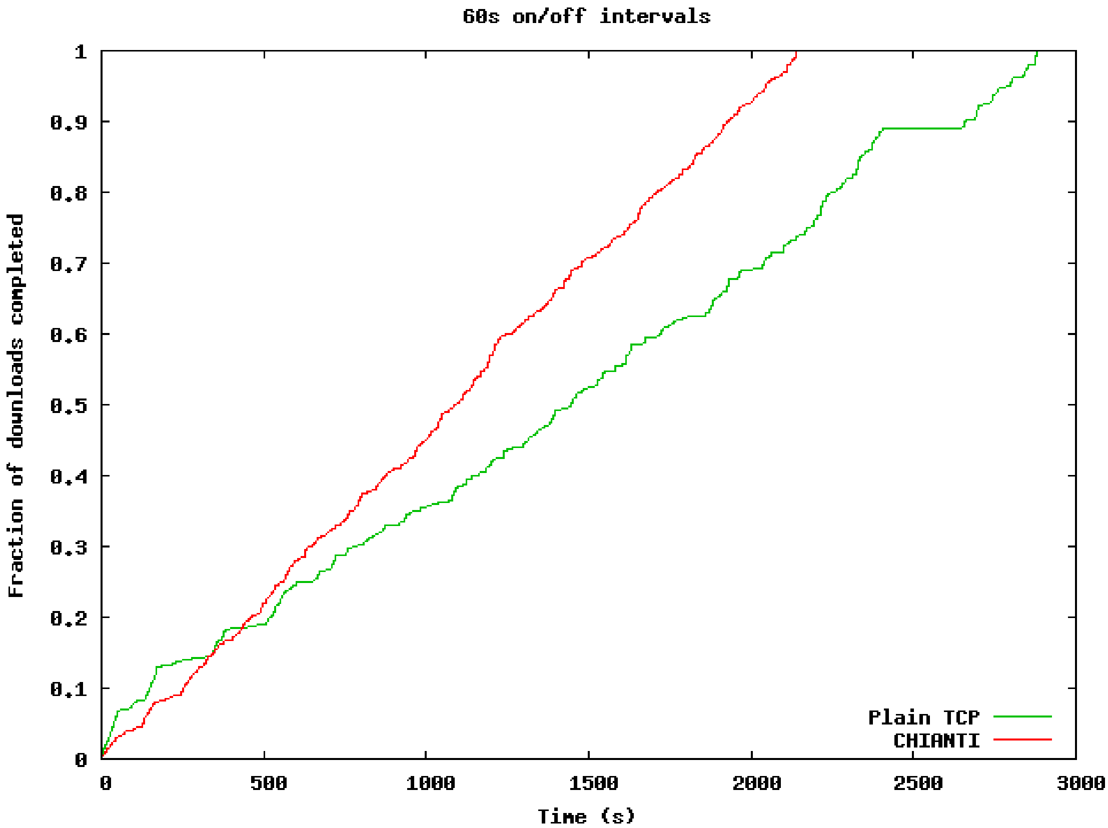

Figure 20: Parallel download, CHIANTI vs. Plain, on/off time 60s

© The CHIANTI consortium – 2008-2009

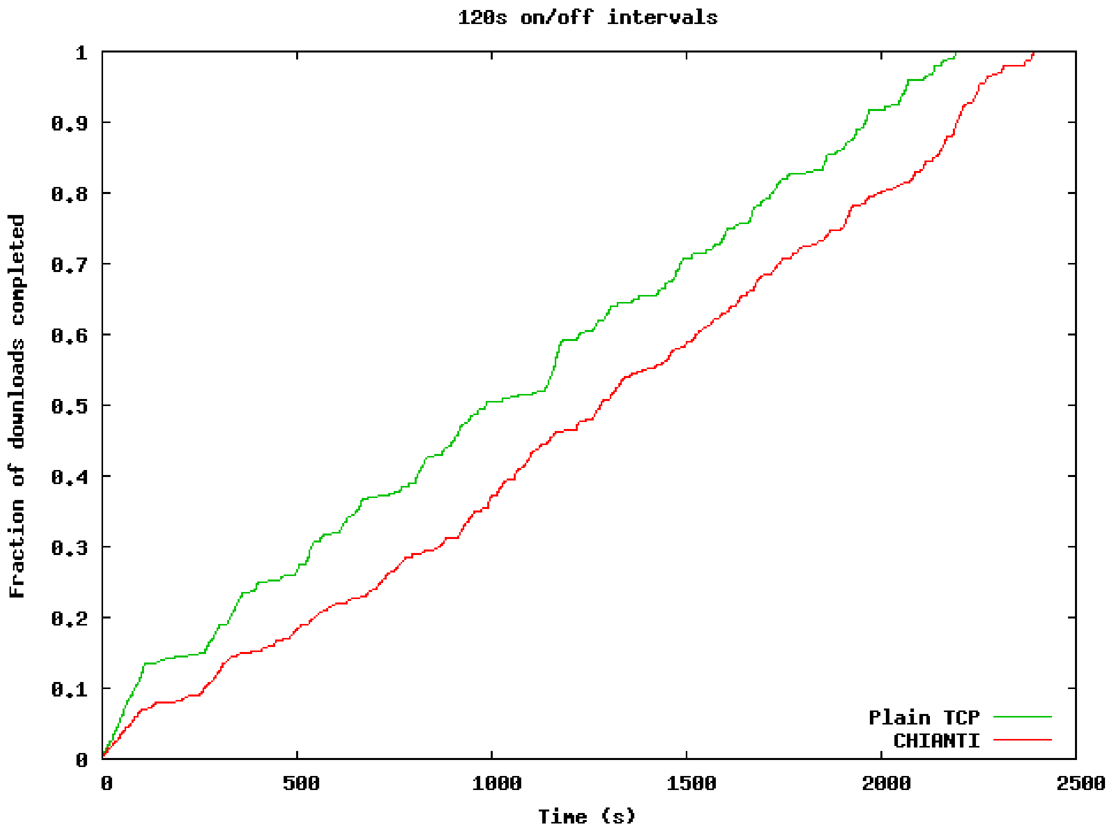

Figure 21: Parallel download, CHIANTI vs. Plain, on/off time 120s

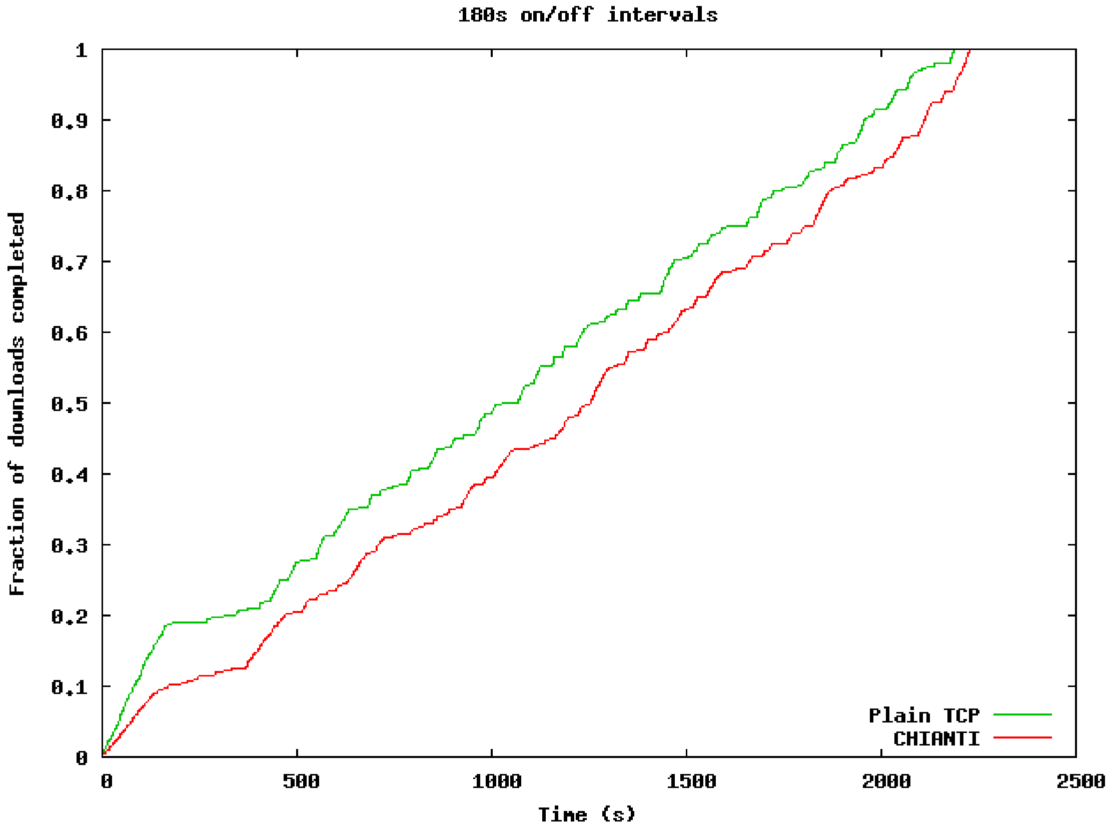

Figure 22: Parallel download, CHIANTI vs. Plain, on/off time 180s

 © The CHIANTI consortium – 2008-2009

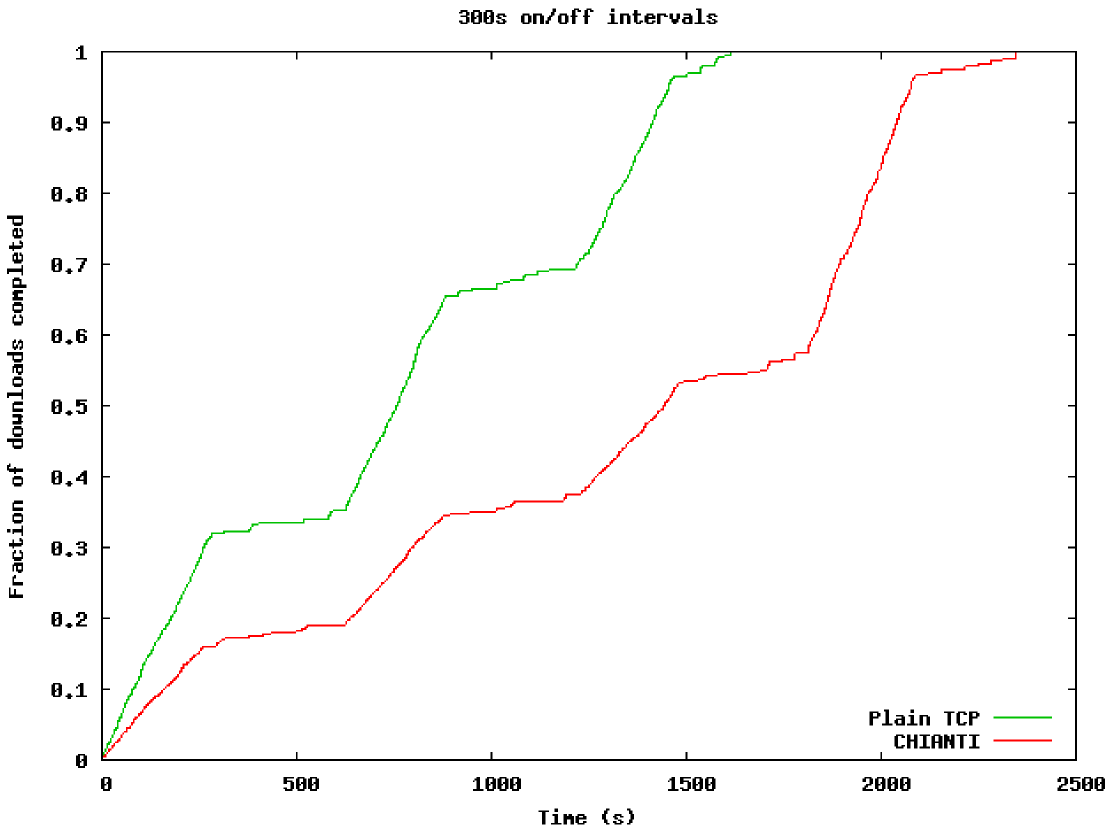

Figure 23: Parallel download, CHIANTI vs. Plain, on/off time 300s

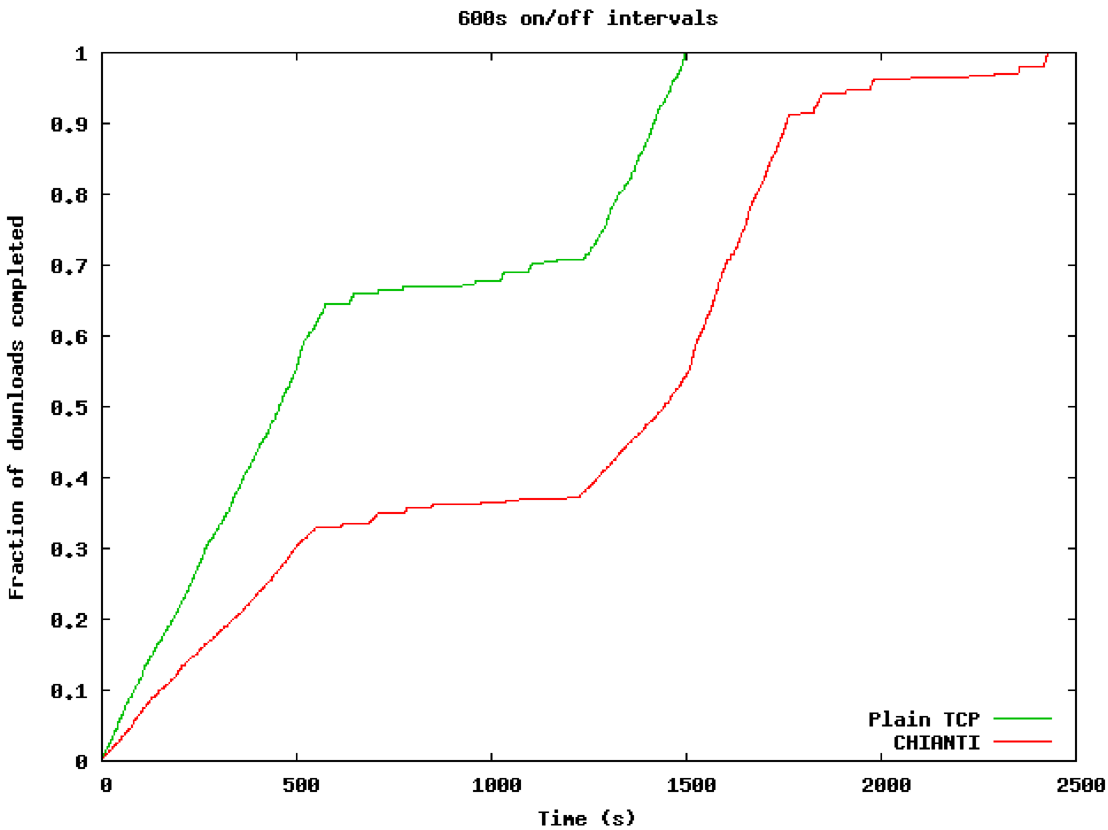

Figure 24: Parallel download, CHIANTI vs. Plain, on/off time 600s

Based upon these figures, we make the following major observations:

The relative performance of the CHIANTI system compared to plain TCP depends on the duration of the on/off periods. CHIANTI performs better for short connectivity periods (30s and 60s); the performances of both methods are roughly comparable for 120s and 180s; and plain TCP outperforms CHIANTI for longer connectivity durations such as 300s and 600s. We will analyse this further with reference to Figure 25 and Figure 26 below.

The latter is quite natural: as noted above, the CHIANTI system introduces two additional elements into the data path, the Flex Proxies, and thus systematically increases the latency of the transmission. Moreover, when – as in the nomadic scenario – the Flex Proxy also runs on the laptop, “twice” the amount of CPU processing is required: additional context switches are needed. We will revisit this when discussing Figure 27 below.

The CHIANTI system is designed for efficient recovery after network outages and to make effective use of available capacity. Therefore, especially the performance for short connectivity periods is relevant: How long does it take to complete a task after (re)opening the laptop? Can a user quickly sit down and complete a previously interrupted task? May she even do so while standing?

The above results clearly show that this goal is met. The CHIANTI DP-Basic protocol design supports efficient recovery and is more sensitive to a communication channel becoming available again. While plain TCP misses the initial part of the communication window after being connected again, the CHIANTI system can exploit this capacity.

This is of key relevance to all CHIANTI scenarios, especially the mobile ones: short-time alternating on/off situations may frequently be encountered when travelling on a train or a car. In those cases, the fast recovery property of CHIANTI is essential to provide a seamless connectivity experience to the user even though the underlying connectivity is not.

A further observation can be made when looking at the (variation of the) slopes in the above figures. The CHIANTI curves are more “straight” hinting at less variability. This becomes apparent from Figure 25 where we plot the variation as error bars. As can clearly be seen, plain TCP exhibits much broader variation and thus less predictable performance for all the settings with on/off periods of 180s or less. Only for 300s and 600s, the “lost” initial connectivity window becomes negligible and plain TCP is able to recover.

 © The CHIANTI consortium – 2008-2009

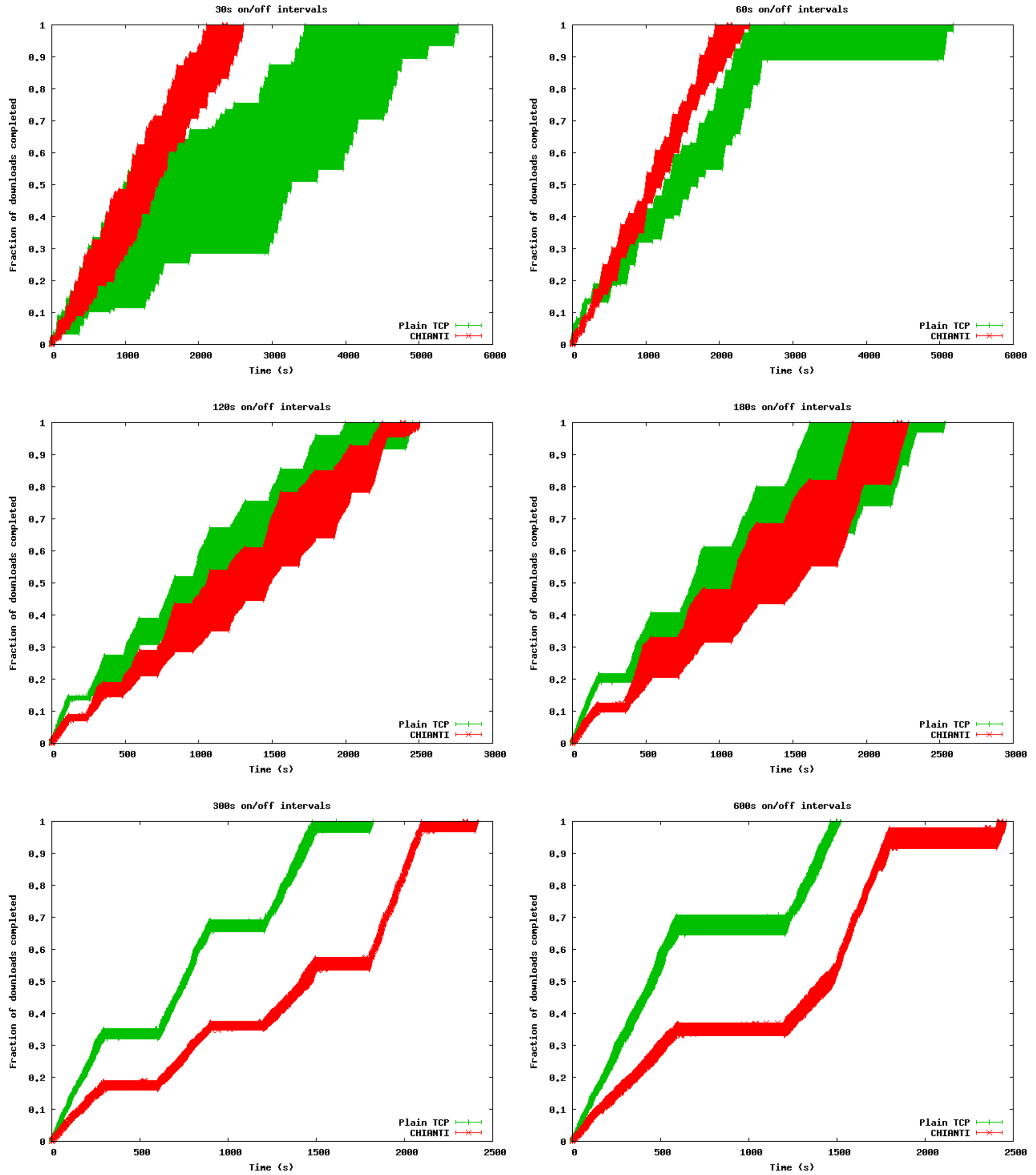

Figure 25: Parallel download, CHIANTI vs. Plain, showing variation as horizontal error bars

We investigate this further in Figure 26 where we plot the mean completion time (plus error bars) over the duration of the on/off periods, thus aggregating all these finding of the above graphs. This figure clearly shows that the CHIANTI performance is roughly constant irrespective of the disruption and connectivity intervals. And the very small variation also show that the CHIANTI system exhibits predictable performance: CHIANTI users will thus know what to expect and they will get it.

In contrast, Plain TCP is very sensitive to the delay. This is a function of TCP recovery (as we know), but also of application layer timeouts. The key point is that the user does not know about either and usually cannot configure or adapt any of them. This yields unpredictable and highly variable performance (as the long error bars clearly indicate).

This (and 2.) is especially important for nomadic scenarios, but also for truly mobile scenarios: users do not know what their connectivity periods will be and thus delivering constant performance (even if it is a bit slower in the end) is much more preferable than unpredictability.

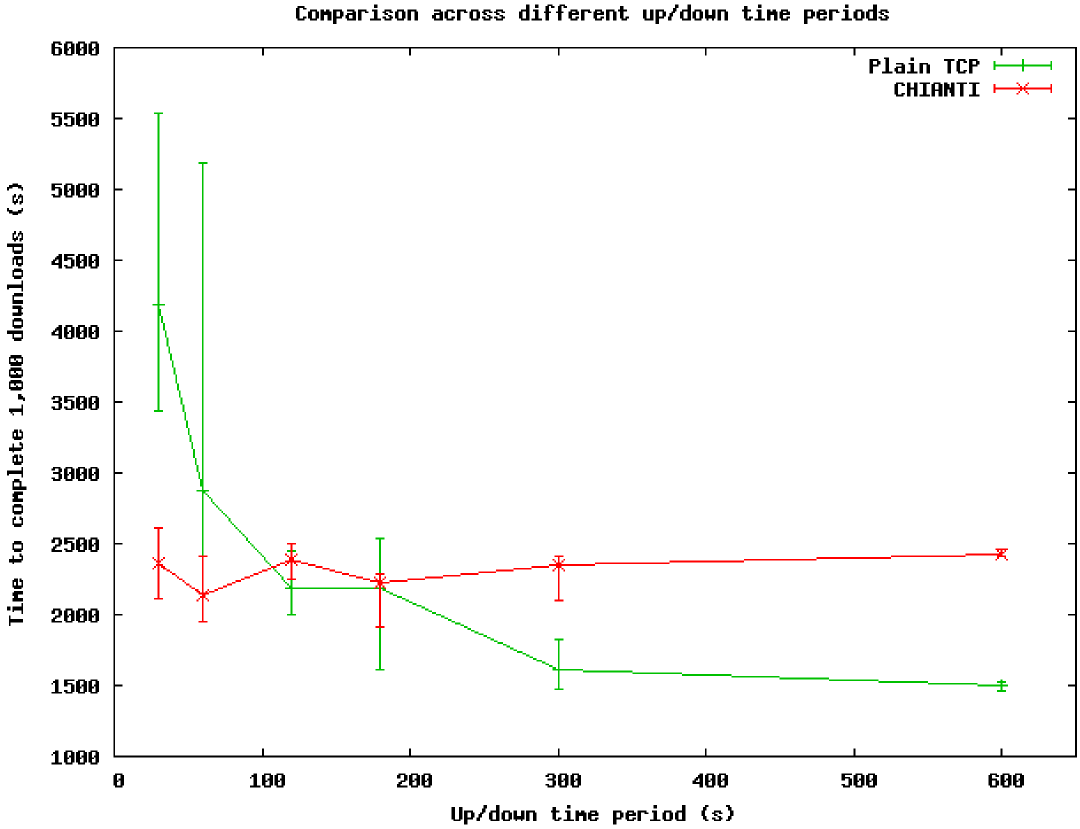

Figure 26: Comparison of CHIANTI vs. Plain TCP across the different on/off times

Figure 26 also confirms what we observed from the steepness of the slopes in the figure above: the performance of plain TCP is better than CHIANTI if there is ample time to communicate. These findings stem from parallel measurements of plain TCP and CHIANTI. Looking especially at Figure 23 and Figure 24, we observe that the slope of the CHIANTI curve becomes steeper once plain TCP finishes (around t=1500s). This indicates that the running plain TCP in parallel affects the CHIANTI performance. In contrast, we do not observe a noticeable impact of CHIANTI on plain TCP: referring to Figure 19 and Figure 20, the TCP performance does not pick up (i.e., the curve does not become steeper) once CHIANTI is finished.

This seems to be in line with our discussion above that CHIANTI uses additional CPU resources and, of course, both systems share the same bottleneck link (i.e., the DSL access). It appears that CHIANTI can make use of additional resources once they are freed, whereas TCP cannot do so effectively for short connectivity periods. We note, however, the effect cannot be explained by the link capacity alone: both DP-Basic in CHIANTI and Plain TCP use a TCP connection across the access link. Two TCP connections (using the same operating system implementation) should share the path capacity fairly – and both connections share most of the same path, except for the last two hops in the well-overprovisioned university network. Hence, one would expect both curves to have roughly the same slope – which is clearly not the case in Figure 19 and Figure 20; this is confirmed when looking at individual

 © The CHIANTI consortium – 2008-2009

runs. Hence, local processing and operating system overhead and more processing competing for local resources are expected to contribute here.

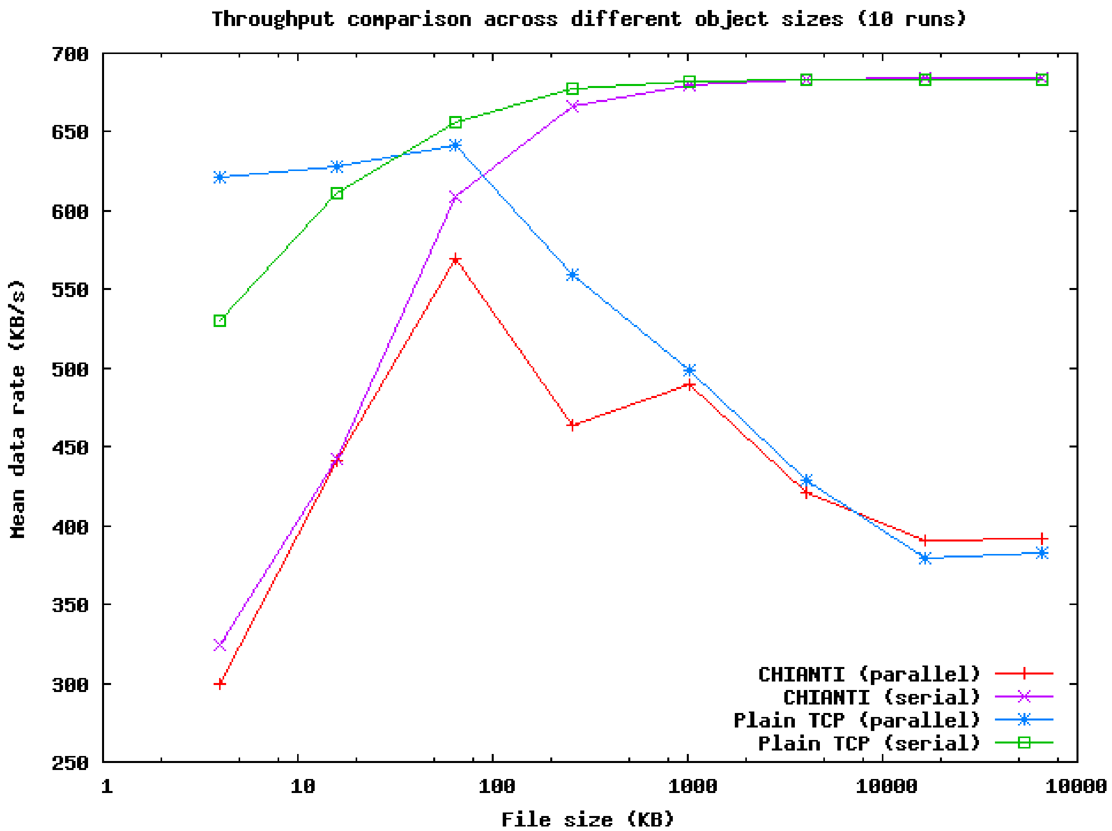

Figure 27: Comparison of CHIANTI vs. Plain TCP for different object sizes using a permanently connected DSL link

To assess the impact of parallel operation and the impact of different object sizes further, we run the measurements both in parallel and in sequence in a permanently connected setup. As the CHIANTI protocol incurs additional setup overhead and minor protocol overhead, we systematically vary the size of the objects downloaded. The results are plotted in Figure 27. The runs take place at night to minimise the impact of cross traffic. However, such cross traffic effects could not be completely eliminated and all our measurements had one or two of such outliers in different positions. In Figure 27, we see such an outlier for an object size of 256 KB in the parallel measurements where both the CHIANTI and the plain TCP values are lower than they could/should be.

We find that – expectedly – the CHIANTI overhead (an additional round-trip time during connection setup) is noticeable for small objects up to 16 KB, becomes less pronounced after 64KB, and disappears for 1 MB and above. This holds for both parallel and serial operation.

In the serial scenario, we observe a similar growth in performance for plain TCP as the download object size grows until it reaches 1 MB when it seems to reach the capacity achievable under the given bandwidth × delay product. The gross capacity of the DSL access link is 8 Mbit/s, the effective capacity has been measured to be mostly around 6 Mbit/s: the observed 680 KB/s are about 5.5 Mbit/s, so well in line and are achieved for both plain TCP and CHIANTI.

The curve looks differently for parallel operation. While the relative performance of the two shows a similar behaviour, we notice that the throughput first grows and then shrinks again. A closer look reveals that the combined throughput of both parallel connections seems to ex-

 © The CHIANTI consortium – 2008-2009

ceed the available link capacity. However, we need to remember that one connection may complete prior to the other, and, since the time dimension is not captured in this plot, the two connections utilise the link not that much in parallel for small objects. Plain TCP may complete while CHIANTI still performs the initial handshake. Moreover, we need to consider the start-up behaviour of TCP. Both connections compete for the shared capacity and start out in TCP *slow start* where they double their transmission rate roughly every RTT until packet loss is observed; only after this, TCP enters *congestion avoidance* and oscillates around the link capacity. Slow start is a phase, where TCP picks up speed quickly, and the extra RTT for CHIANTI matters particularly much. For small objects, TCP never leaves slow start and the object transmission completes before the first loss is observed. For larger objects, this initial "advantage" of plain TCP becomes negligible as the two connections share the link most of the time.

Overall, our experimental assessment shows that the CHIANTI system exhibits the desired characteristics: its long-term performance comes close to TCP, but it is capable of exploiting short connectivity periods more effectively and more predictably and recovers faster from disconnections.

3.1.1.3.2 Exemplary Nomadic Access in Hot-Spots

In addition to the systematic evaluation of the transport and disruption tolerance performance of the CHIANTI system, we have investigated the performance when moving between WLAN hot-spots. We have carried out a series of 10 to 20 experiments with different motion patterns between the three or four of the hot-spots described above, running both plain TCP and CHIANTI in parallel to make the results comparable.

Since the conditions cannot precisely be reproduced, we restrict ourselves here to an exemplary evaluation. We have observed two representative results that are shown in Figure 28 and Figure 29**.**

Figure 28 depicts the case in which we set the `wget` timeout to *infinite*, i.e., `wget` attempts to complete an operation without aborting and retrying at the application layer. We see that the plain TCP simply halts as soon as we leave the first hot-spot. When disconnecting and reconnecting at a different location, the IP address changed, so that the existing TCP connection would not receive any further packets. Communication could resume if we returned to the same location (which we did not in this case). In contrast, the CHIANTI downloads progresses continuously at every hot-spots. Moving back and forth between different locations yields a total of six connectivity periods.

Figure 29 shows a case if we set the `wget` time to 300s, i.e., after not having received any data for this duration, `wget` will itself retry at the application layer (note that many application do not support such a feature but rather revert to manual retries as discussed earlier in this deliverable. The application layer support makes support makes `wget` with plain TCP recover. However, it misses one connectivity opportunity in the middle, whereas CHIANTI exploit every opportunity for communication. (In this case, plain TCP is catches up in the third connectivity period, as it can download faster as we discussed above.)

 © The CHIANTI consortium – 2008-2009

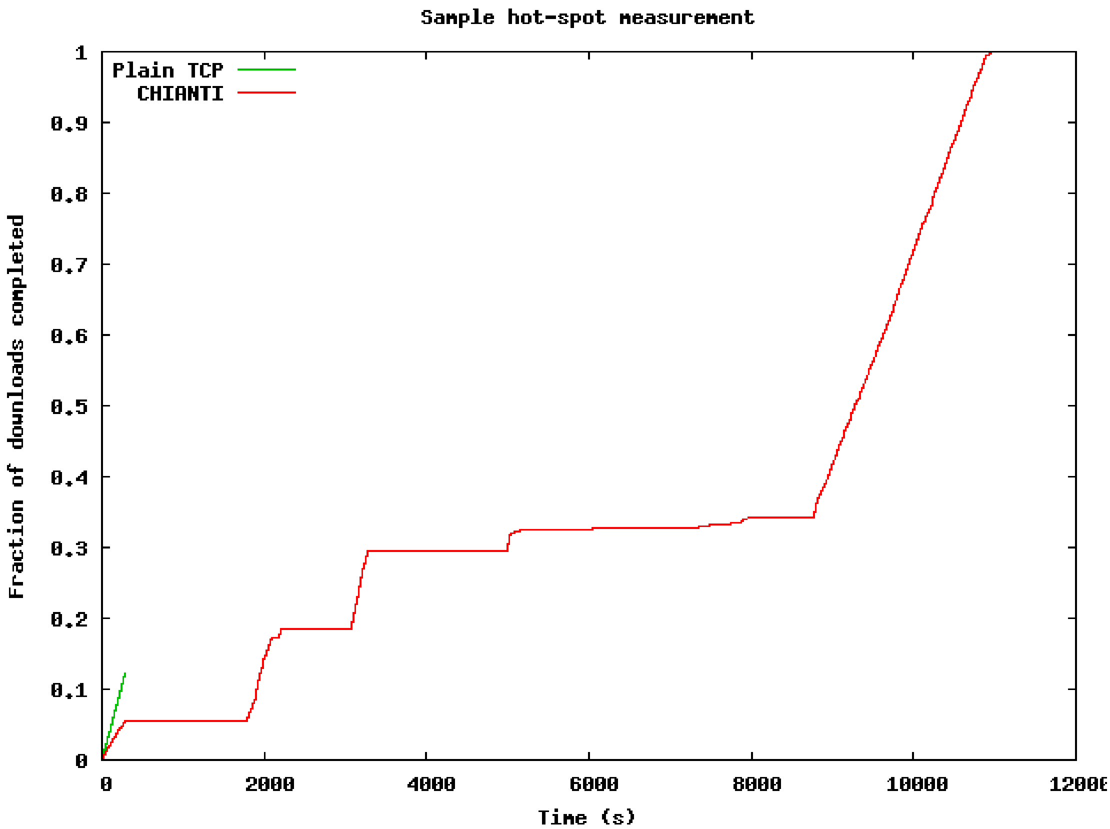

Figure 28: Nomadic trial setup

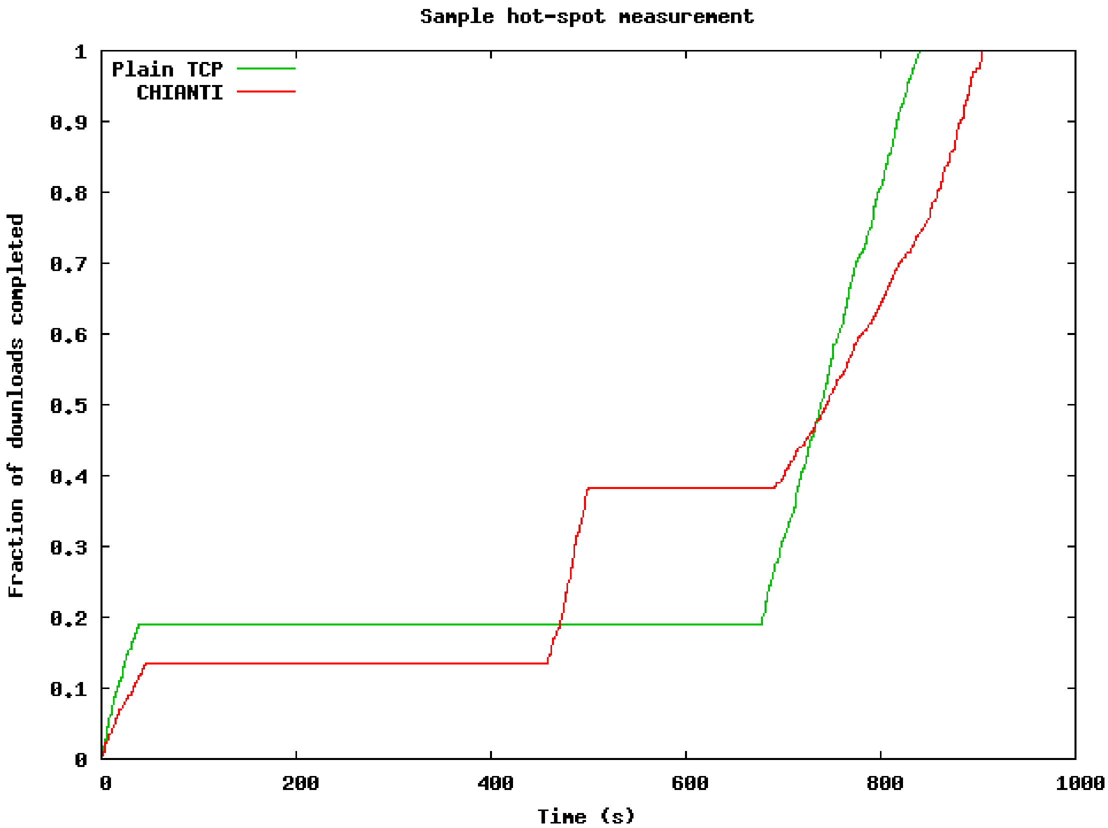

Figure 29: Nomadic trial setup

Overall, the CHIANTI system performs properly for moving between hot-spots in a nomadic fashion. We also conducted experiments in which we interspersed 3G connectivity for a short period of time – which was also picked up efficiently by the CHIANTI system.

These on/off scenarios for nomadic users essentially provide a suspend/resume feature for communications, allowing users to stay logged in, proceed seamlessly with incomplete downloads, etc. In general, however, nomadic users will often have sufficient time in each spot so that the efficient resumption is less of relevance compared to the illusion of uninterrupted – i.e., *seamless* – communication offered by CHIANTI.

If the up/down periods becomes shorter, the nomadic scenarios gradually turns into the mobile scenario – and the aforementioned efficient recovery gains relevance as we will show in the next subsections.

3.1.2 Using CHIANTI within a vehicle

The trials were accomplished during several rides in trains as well as in cars. Thus the main means of transport were covered. We assume that tests in busses and coaches are very similar to car rides and therefore do not have to be tested individually.

Vehicle tests were performed on different routes in Northern Germany. The in-car tests included motorways and country roads as well as urban environments. The trials on the trains covered German Metronom and Intercity routes.

3.1.2.1 Trial Setup

The trial environment for the nomadic vehicle tests is very similar to the pedestrian usage environment. It consists of three components: a mobile node, a stationary server where the CHIANTI proxy is installed and a web server. This configuration is depicted in Figure 30.

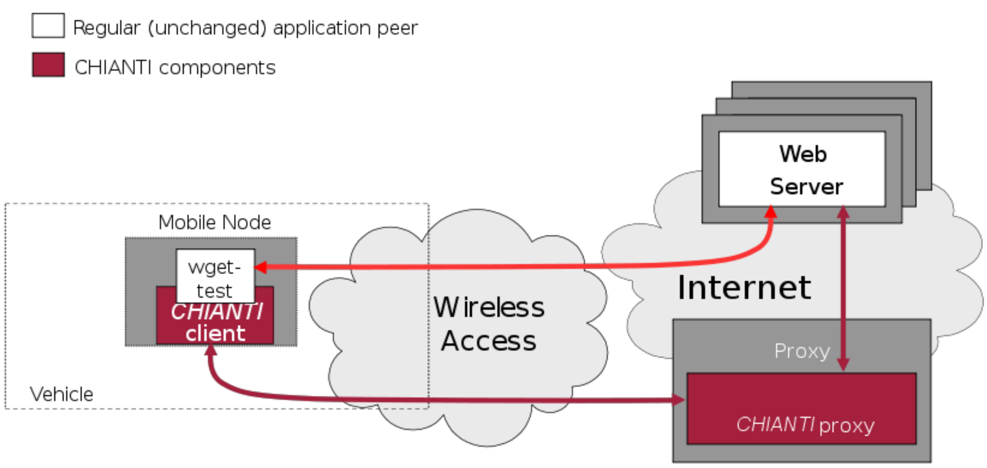

Figure 30: Nomadic Vehicle Trials

On the mobile node a CHIANTI client as well as a set of test scripts is running. The purpose of these scripts is to observe the download of files from representative web sites (see 3.1.2.2). The tool `wget` serves as a configurable web browser replacement.

In order to get comparable results, two tests are performed in parallel, one with and the other without CHIANTI support. Because of the TCP congestion avoidance algorithm both processes have a fair share of the available bandwidth. This is validated by the results of two

 © The CHIANTI consortium – 2008-2009

parallel tests which were both run without CHIANTI support. Figure 31 shows the number of Mbytes finished over time. As can be seen, the results are very similar for both processes. Although one process may receive a greater share of bandwidth for a short amount of time, this effect is soon equalised.

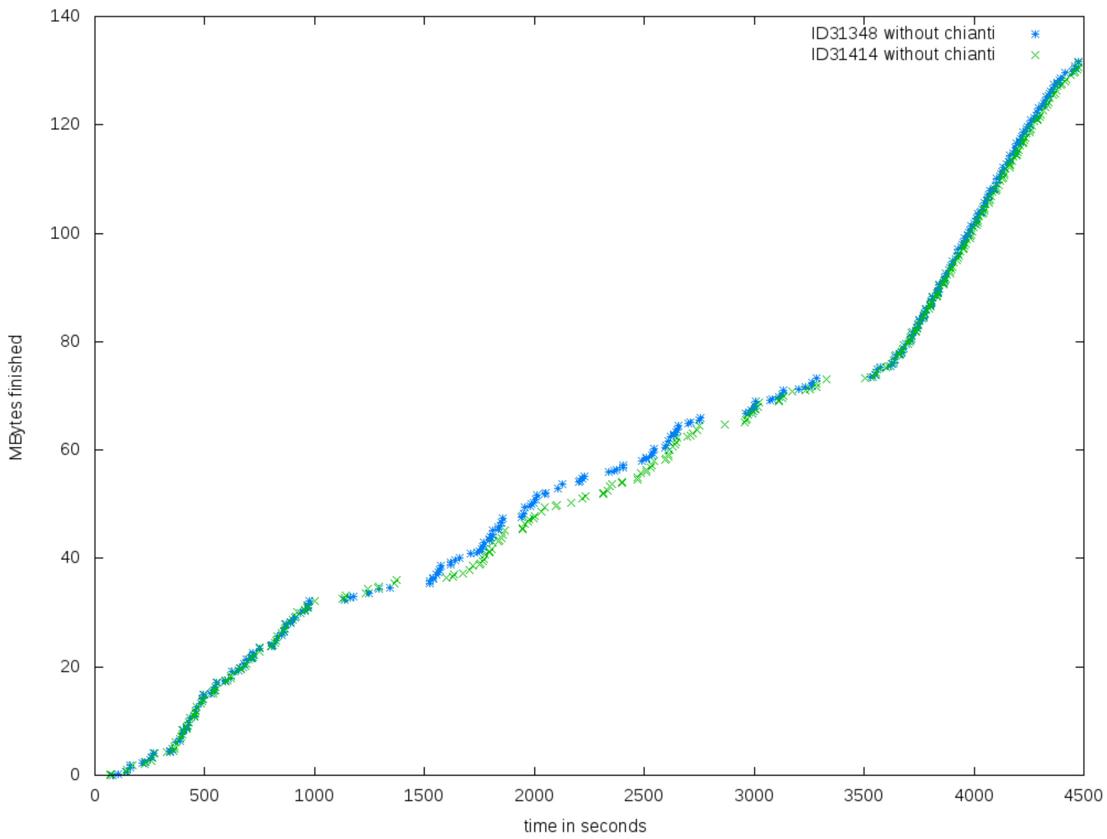

Figure 31: Two parallel wget tests without CHIANTI

The mobile node communicates via UMTS or GPRS. While without CHIANTI support the communication occurs directly between mobile node and web server, the CHIANTI traffic has to be routed through the CHIANTI proxy first.

As mobile node serves a notebook with an Intel 1.6 GHz CPU and 2 GB RAM. The server with the CHIANTI proxy is equipped with an Intel 2.5 GHz CPU and 1 GB RAM. The web server is provided with an AMD Athlon 2800+ CPU and 2 GB RAM.

3.1.2.2 Test Methodology

The evaluation of the CHIANTI service during the user trials is based upon the findings of the protocol and application usage analysis in WP2 [4] and the initial investigation of prevalent use cases in WP1 [3]. According to these results, more than 80 % of packet flows between two communicating endpoints are HTTP connections and more than 75 % of the downloaded data volume is transferred via TCP. Applications built on top of TCP—and Web-based applications in particular—thus take a reasonable share of the overall Internet usage in the investigated environment. Given this, the performance of existing Web sites is a reasonable indicator for the perceived quality of the Internet access service.

In order to provide objective measurements for service quality, the CHIANTI software was evaluated against a set of reference Web sites that have already been used for investigation

 © The CHIANTI consortium – 2008-2009

of access link performance. To avoid unforeseen changes in this reference set and to prevent sending bogus requests to existing Web servers, parts of the reference sites have been mirrored on a server under the project's control, and the automated tests were designed to retrieve objects from this server only. Although side-effects caused by caches or enhancing proxies on the path between client and server may still occur as the setup shall resemble real-life Web traffic, interference between independent test-runs are reduced to a minimum by using independent copies of the mirrored pages.

To evaluate the impact of the CHIANTI service in the nomadic user scenario, a set of automated tests is executed, with the results being logged and processed afterwards. Every test run retrieves one of the mirrored Web pages together with its embedded objects concurrently on a plain TCP connection and via SOCKS using the CHIANTI service. For each completed object, the size and duration are stored in a log file. The graphs shown in the following sections are created from the collected data together with post-processed traces from the captured IP traffic between the test scripts and the web server.

The reference web sites that are used to generate HTTP traffic include a news channel, a net magazine, an online shop, and two encyclopaedias. Most of the sites' contents are static to simplify the counting of successful transactions. Note that interactive sites that make extensive use of AJAX technologies[9] will result in HTTP traffic that has a large number of very small objects being transferred. [10] shows that browsing web pages and file download still make nearly two third of the overall HTTP traffic. Although the observed use of web applications increased by more than factor 14, those applications still make up about 16 % of the overall HTTP traffic.[10] The major benefit of using Web 2.0 technologies also is attributed to interactive usage patterns and improvements to the user interface. Since these aspects of usability have not been in the focus of the CHIANTI project, the automated benchmark tests have been designed to use static web pages only.

The properties of the reference web sites reflect typical web pages of average complexity. Figure 32 shows the distribution of distinct object types according to their size. More than half of the data that is transferred by the test scripts is image data, less than 20 % of which is GIF, and more than 80 % JPEG. Portable Network Graphics (PNG) is hardly used at all. The next big share (20 %) is taken by HTML documents, followed by JavaScript (JS, 13 %) and Cascading Style Sheets (CSS, 11.4 %). Other object types can be neglected as all of those together make up less than 1 % of the data being transferred.

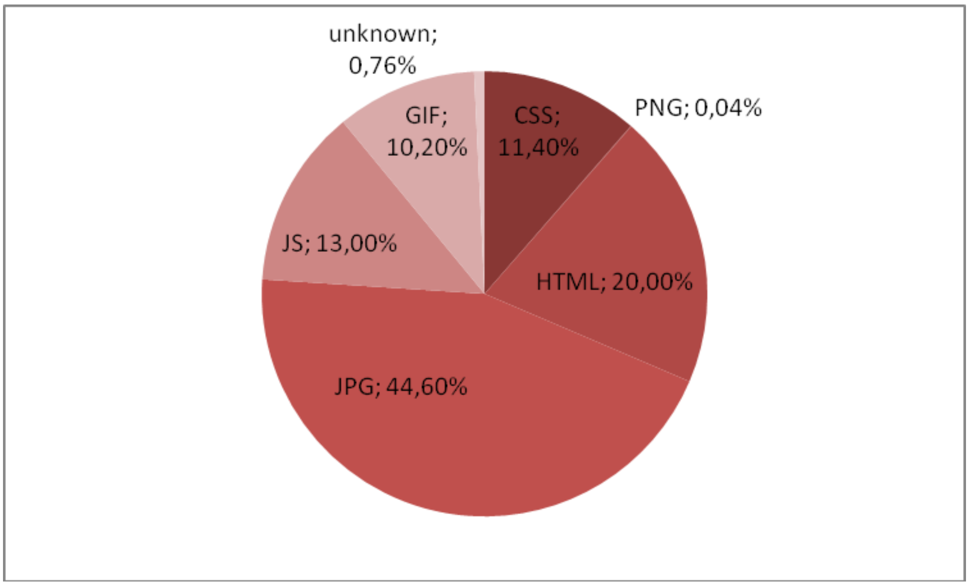

Figure 32: Object Size By Type

The chart in Figure 33 shows the number of objects of a specific type that have been transferred by the target web server. Again, nearly half of the overall amount is JPEG data (46 %), followed by GIF images (39 %). Thus, three out of four objects queried from the server contain image data. While the numbers for JPEG are similar for both, object count and size, the web pages contain a lot of very small GIF images.

The textual content (HTML), visual style (CSS), and application logic (JavaScript. JS) take a rather small fraction of the displayed chart as there are only few objects of these types.

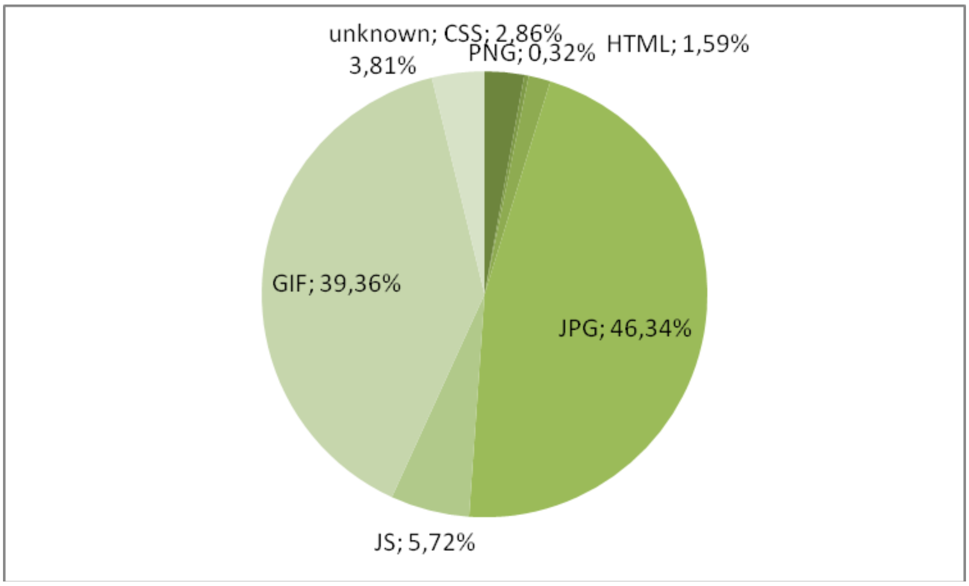

Figure 33: Objects By Number

A closer look at the data traffic caused by these objects reveals that the objects can be matched to four or five categories as depicted in Figure 34. Here, five categories have been used to arrange the web objects. The smallest objects of less than 200 Bytes comprise only a few small CSS documents and a significant number of GIF images. The next category of objects between 200 Bytes and 1 KiB largely consist of GIF and JPEG images, which is also true for the class of objects between 1 KiB and 4 KiB. Textual contents, style and application logic need more than 4 KiB, and a significant number of text documents is bigger than 100 KiB.

 © The CHIANTI consortium – 2008-2009

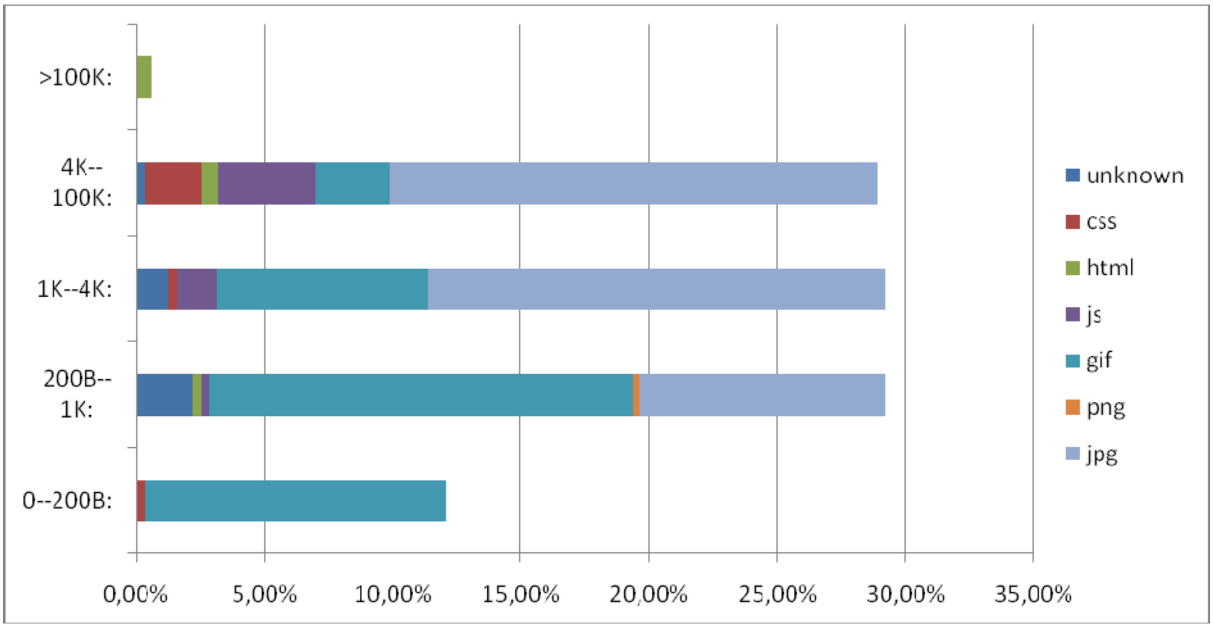

Figure 34: Distribution of Object Size

The web pages were retrieved by the test scripts using the tool `wget` with the network time-out parameter set to 900 seconds and no retries. On the server side, `lighttpd` has been used to serve the static pages. The only change to the server's default configuration was made to extend the maximum lifetime of a TCP connection to 900 seconds for most of the tests to be able to measure the impact of long connectivity gaps.

Test runs consist of the subsequent retrieval of an entire web page including its embedded objects. To mimic real users' behaviour, a random waiting time between 5 and 10 seconds is included before the next web page will be fetched. Several test scripts running in parallel simulate several users (or multiple tabs in a single user's web browser, respectively) concurrently surfing the web. To avoid overloading the service provider's infrastructure with stale TCP sessions that hang around for a long time when the script fails or when a disruption takes too long, tests are cancelled automatically after 900 seconds.

3.1.2.3 Results

This section contains representative samples for the vehicle trials. They were accomplished in Northern Germany in trains and cars using a notebook with a CHIANTI client installed (see section 3.1.2.1). The first test was measured on an Intercity train between Rostock and Bremen. The behaviour of wget with and without CHIANTI support is shown in Figure 35. It depicts the total amount of Mbytes successfully downloaded by wget. The test was started at 13:39:28 at Rostock and ran for more than an hour. Figure 36 contains the corresponding total bandwidth per second on the mobile device as well as the signal strength. It becomes apparent that CHIANTI performed better during small bandwidth periods than the direct download.

 © The CHIANTI consortium – 2008-2009

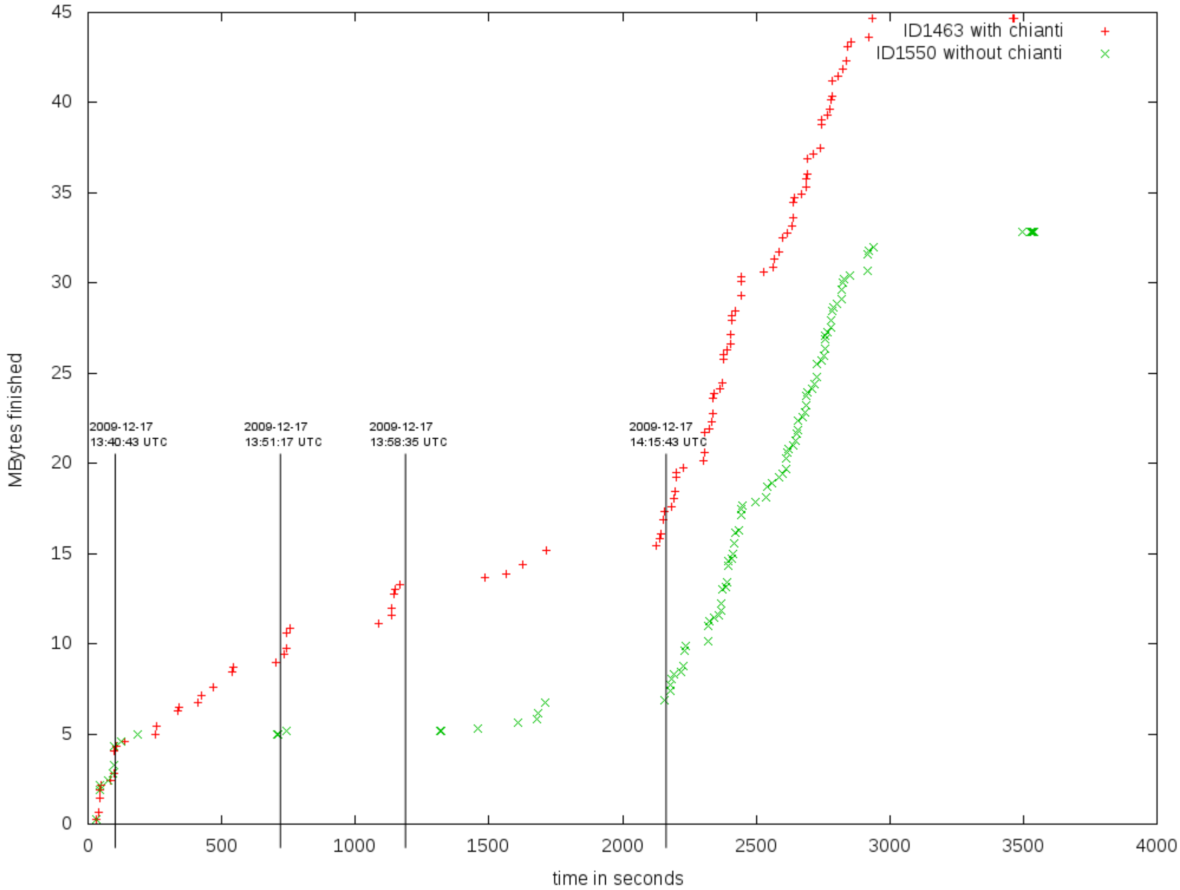

Figure 35: Connectivity Around Rostock

Between 13:42 and 13:51 no data was transmitted without CHIANTI while about 5 Mbytes were downloaded with CHIANTI. The green crosses at 13:51:16 indicate aborted downloads due to the wget timeout. At 13:51:52 the direct download received only 2 Kbytes. Another set of direct downloads started at about 13:51:18 was aborted by wget at 14:01:25. Until 14:15:26 more than 17 MBytes were transmitted using CHIANTI while the direct wget was only able to finish the download of 6 Mbytes.

Figure 37 and Figure 38 indicate similar results for a car-ride on a German country road (see Figure 39). Here, the throughput increases significantly at approx. 1400 seconds after beginning of the test due to the fast link detection of the CHIANTI service. The plain TCP connection finished downloading of very few objects during the period between 1000 seconds and 2000 seconds. After that, the throughput graphs go nearly parallel, i.e. they show nearly the same behaviour.

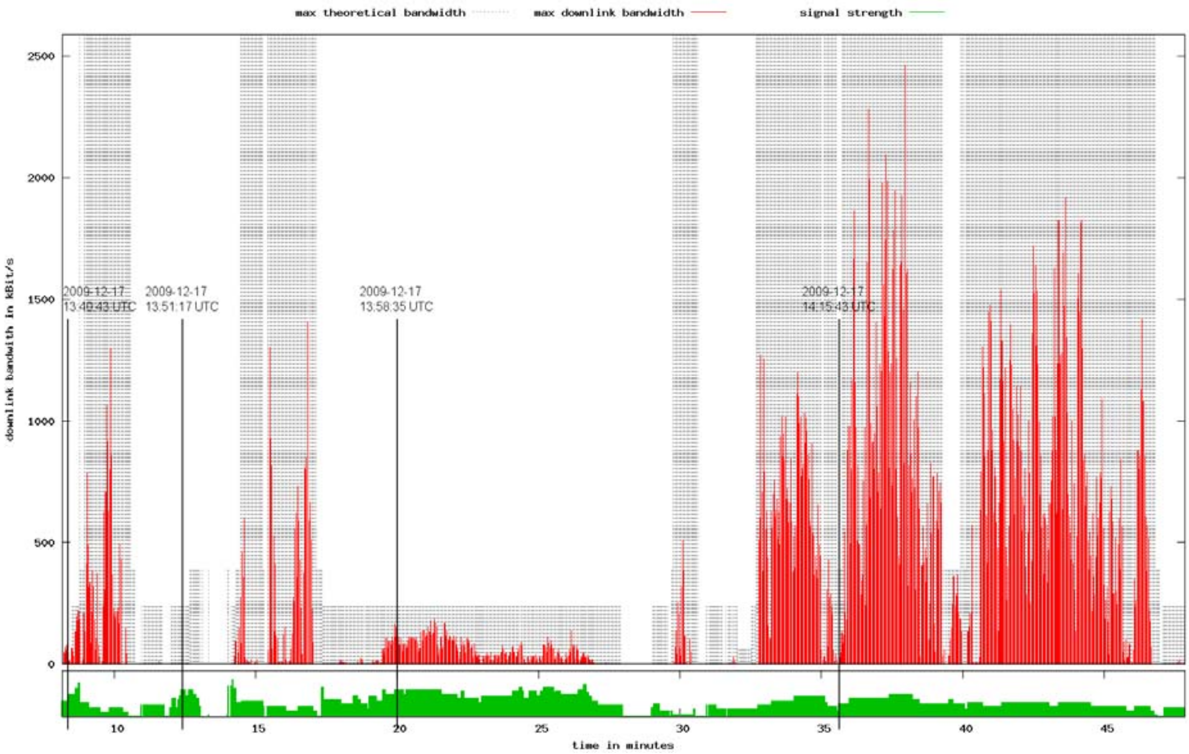

Figure 36: Bandwidth and Signal Strength Around Rostock

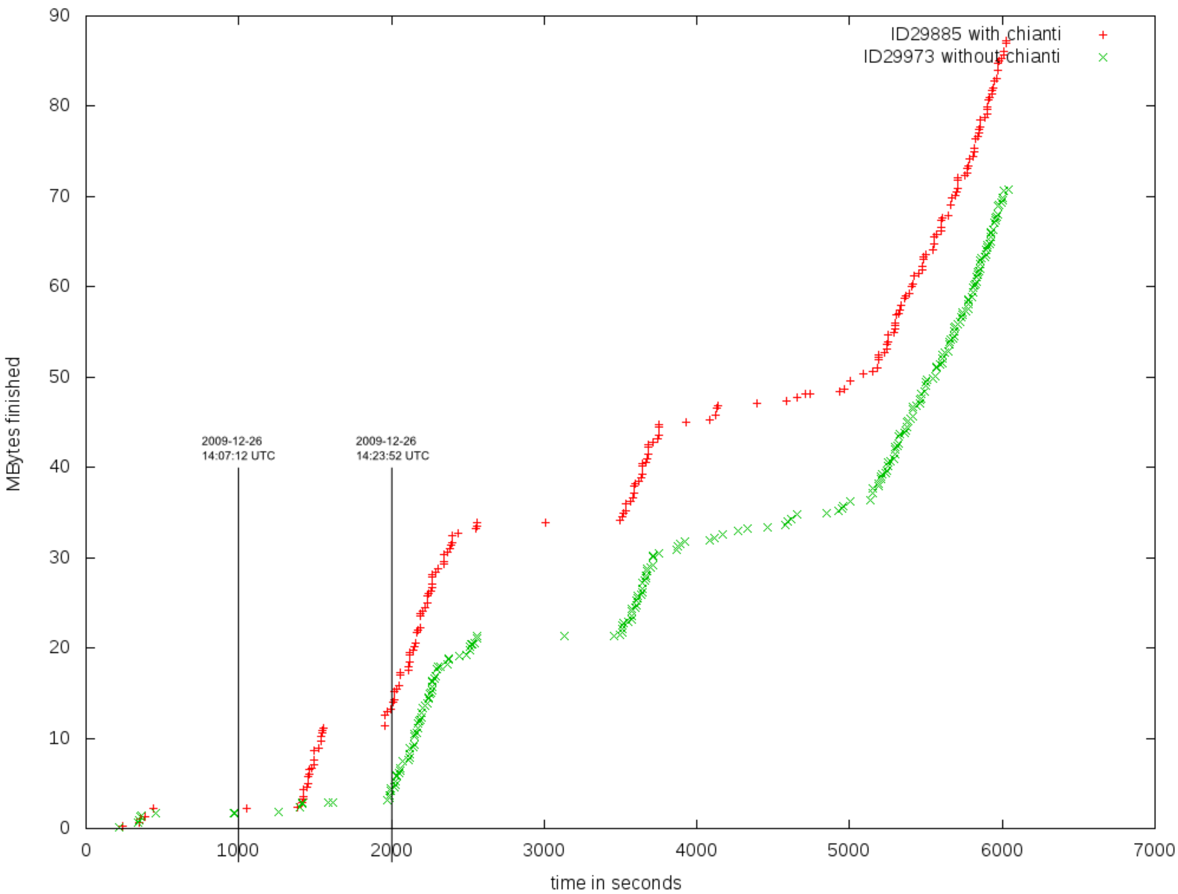

Figure 37: Intermittent Connectivity Around Winsen

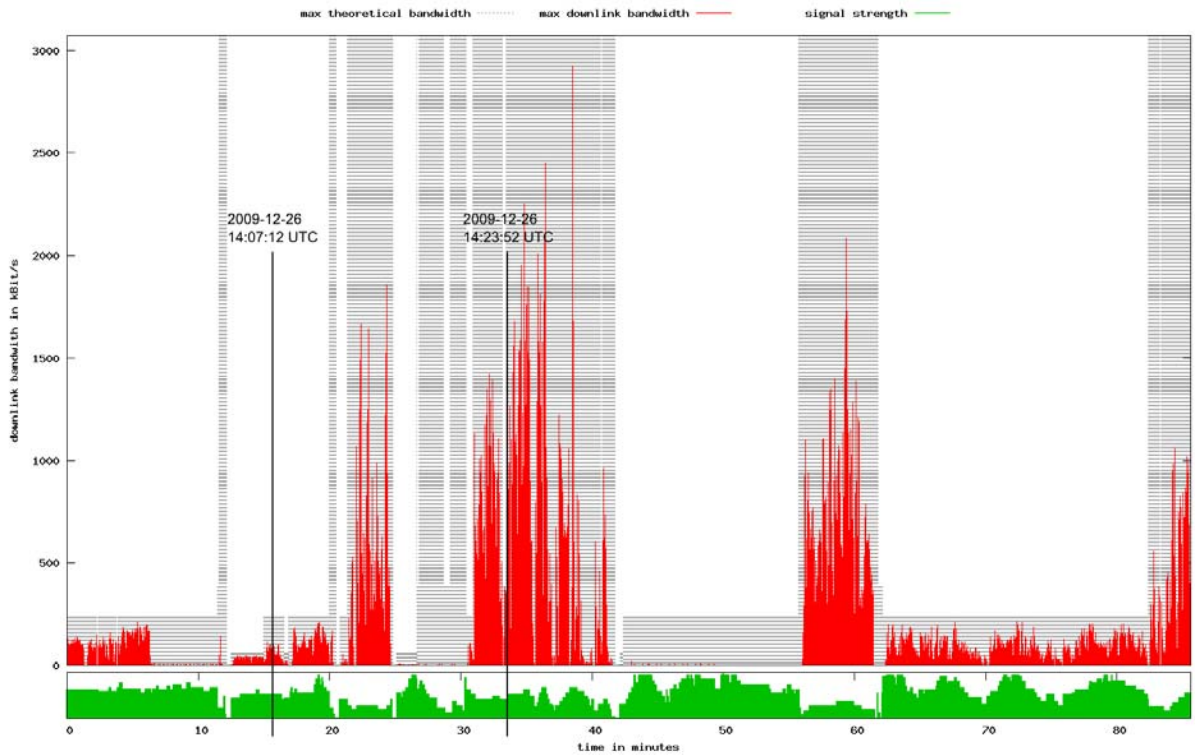

Figure 38: Bandwidth and Signal Strength Around Winsen

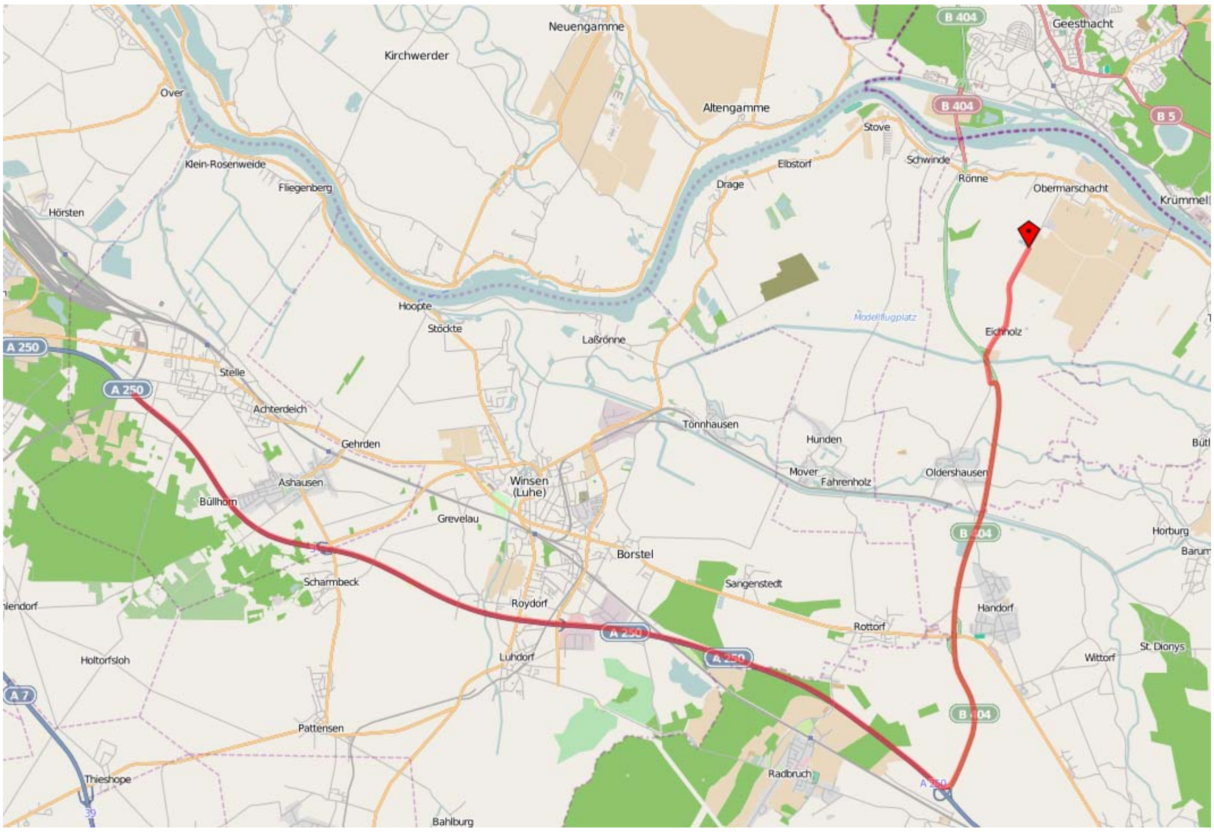

Figure 39: Car-ride on a German country road (2009-12-26 14:07-14:23)

3.2 Mobile Use

To evaluate the CHIANTI system under real-world conditions with a large number of users, the software has been deployed on several high-speed trains that serve various routes in North England, specifically covering rural areas with connectivity gaps of sometimes more than 30 seconds. As the entire TCP traffic of a certain tariff class has been routed through the CHIANTI platform, the consortium had to make sure that no outages induced by software failure would occur. Moreover, the routing architecture for the wireless Internet access service on the trains had to be adjusted to ensure proper operation of the existing authorisation and accounting mechanism.

The following subsections describe the preconditions and actual trial setup (section 3.2.1), deviations of the test methodology for the mobile usage scenario (section 3.2.2), and finally comment on the results of the user trial (section 3.2.3).

3.2.1 Trial Setup

The CHIANTI service was provided on various passenger trains serving the route London – Manchester during December 1st to December 23rd, 2009. As part of this trial, outbound user traffic was directed to the SOCKS interface of the CHIANTI client software by a combination of `transocks` and `iptables`, hence users were able to participate in this trial without making any changes to their devices' configuration.

The automatic test scripts were executed on the train's CCU to enable direct Internet access without being redirected through the CHIANTI system. Parallel test runs that used the CHIANTI service thus were explicitly configured to use the SOCKS interface of the CHIANTI client software. To do so, the tool `tsocks` has been used to call `wget` from a test script.

The trials have started with version 0.3.01 installed on the trains' CCUs and the corresponding home agent in the data centre. After investigation of early measurements, some performance optimisations on prototype III have yielded version 0.3.07 which was deployed incrementally while the trials were running. The resulting data thus not only show the difference between the CHIANTI service and plain TCP connections but also demonstrate the impact of these optimisations.

Since the trials were run in an operational system with a productive Internet access service being affected, a stable and production-ready version of the CHIANTI software had to be used. As a consequence, the trial results only cover DP-Basic as the most mature CHIANTI protocol version that has been implemented during the project's lifetime. To find a good ratio between robustness and system performance, the default configuration of prototype III has been changed as follows:

- The option `cleanup_stale_timeout` was reduced to 600 seconds for faster cleanup of internal state for disrupted connections. While this limits the maximum duration of outages that CHIANTI can bridge to 10 minutes, the proxy can handle more parallel connections.
- The `reconnect_interval` was set to 250 ms to speed up re-establishment of sessions after a network outage.
- By setting the option `expect_ack_timeout` to 10 seconds, connections can be re-established faster in case of short disruptions.

3.2.2 Test Methodology

For the evaluation of the final CHIANTI prototype on the Virgin train fleet, a set of test scripts has been developed to compare the CHIANTI service with plain TCP data transfer. These scripts were already used for evaluation of the software in the vehicular scenario of the nomadic use case described in section 3.1.2 and therefore the same test methodology is applied (cf. section 3.1.2.2).

With the test scripts installed on the train, sufficient HTTP traffic is generated to gather representative data to analyse the quality of the CHIANTI service. As described earlier, the CHIANTI service is designed specifically to deliver best results when network conditions are bad, i.e. in the presence of high variations of round-trip times and high packet loss rates. The benefit of using CHIANTI hence becomes visible specifically during connectivity gaps. The first step to evaluate the service for the vehicular use case therefore implies a thorough gap analysis along the train routes that were used. After a set of suitable train routes heading North from London was identified and the CHIANTI software was deployed on trains that serve these routes, the automatic tests have been run over several days to collect a significant amount of measurement data.

As the narrow focus of the CHIANTI project and its short lifetime did not allow for an extensive statistical evaluation of these tests, a manual analysis of the measurement data was performed after the data collection phase. The results of this analysis as shown in section 3.2.3 indicate a reasonable improvement that the CHIANTI service provides when compared to plain TCP connections.

3.2.3 Results

This section documents the results of the final trial phase that has been performed on various Pendolino trains running through North England. As described previously, these trains are equipped with wireless Internet access using various 3G networks for better coverage when riding overland, as well as WiFi or WiMax in metropolitan areas. Mobile IP ensures seamless handoff between distinct network providers and different access technologies.

During the trials, the connectivity of the CHIANTI-enabled trains has been monitored constantly by sending an ICMP echo packet every second to each of these trains. As the echo requests that were sent from the mirror server as well as the corresponding responses were recorded at the sending side, the request/response pairs can be used to calculate the round-trip time along the path between the train and the test server. Table 1 gives an overview of the automated test runs that were performed during the trials, listing the test name and date, the (pseudonymised) train unit and the overall duration of the test run.

The round-trip times calculated from more than one million recorded ICMP messages at the test server have been aggregated and are shown in the subsequent columns. The first column shows the median (med) of the round-trip times in seconds. Most of the time, this value is near 400 ms as expected under good conditions for 3G networks when the over-the-air interface is not overloaded. The minimum values are less than 100 ms most likely recorded when the train was in a station or has passed a WiFi hotspot. Outages, on the other hand, may cause delays of more than 120 seconds as was already seen for congested 3G networks in deliverable 1.2 [3]. Note that round-trip time calculation requires both, the echo request as well as echo response to be transferred successfully. Higher delays thus may occur in the network but were not included in the round-trip time calculation as the request or response packet might have been lost on the end-to-end path.

For completeness, the arithmetic average round-trip time is shown in column 7, with the standard deviation of the actual values being listed in column 8.

 © The CHIANTI consortium – 2008-2009

Table 1: Round-trip times for ICMP echo packets from TZI test server

date/test	train	duration	med [s]	min [s]	max [s]	avg [s]	std dev
2009-12-09 multi-test	C	03:21:43	0,291	0,083	79,800	0,860	2,990
	D	03:26:10	0,287	0,083	89,700	1,220	4,580
2009-12-10 512k-test	C	05:00:43	0,238	0,071	116,000	1,040	3,820
	D	05:02:09	0,318	0,075	54,600	1,110	3,380
2009-12-10 single-test	A	04:13:33	0,242	0,071	100,000	1,280	4,230
2009-12-11 multi-test	A	03:13:56	0,375	0,062	60,300	1,300	3,670
	B	02:48:38	0,572	0,070	103,000	1,790	4,000
	C	03:17:28	0,440	0,062	63,000	1,590	4,440
	D	03:13:34	0,285	0,069	44,800	1,150	2,650
2009-12-14 multi-test	A	09:44:33	0,540	0,069	114,000	1,760	4,480
	B	12:19:45	0,381	0,075	99,600	1,280	3,160
	C	12:31:58	0,342	0,061	77,200	1,190	3,090
	D	10:21:17	0,397	0,061	117,000	1,450	3,760
2009-12-15 multi-test	A	04:36:25	0,288	0,076	74,900	0,893	2,650
	B	02:52:44	0,279	0,061	77,300	1,730	5,510
	C	03:43:57	0,666	0,075	56,500	3,180	7,030
2009-12-16 multi-test	A	10:03:58	0,817	0,074	122,000	2,070	4,440
	B	10:37:57	0,451	0,069	75,600	1,490	3,510
	C	08:25:27	0,420	0,062	119,000	1,480	4,360
	D	03:18:01	0,340	0,069	80,100	1,030	3,850
2009-12-18 multi-short	A	02:24:23	0,442	0,065	78,000	1,200	2,630
	A	04:02:47	0,318	0,057	65,800	0,882	2,160
	B	02:24:18	0,306	0,058	59,400	1,020	2,490
	B	04:44:34	0,234	0,057	31,900	0,737	1,940
	C	02:24:22	0,214	0,078	25,800	0,399	1,100
	C	01:28:31	0,202	0,121	1,860	0,240	0,125
	D	02:24:32	0,416	0,057	111,000	1,340	3,490
	D	04:51:25	0,315	0,061	62,500	1,230	3,570

The results documented here are synthesised from the data collected during the test runs on 2009-12-11 and 2009-12-14, covering various interoperable combinations of the final CHIANTI prototype. Figure 40 shows a comparison between simultaneous runs of multiple wget-test processes on train C, with both, CHIANTI Client and CHIANTI Proxy running software version 0.3.07, i.e. the final setup.

The green points indicate the accumulated size of those web objects that have been completed at that time using multiple direct TCP connections to the target Web server while the red points show the CHIANTI-enabled data transfer. The graph has been mapped to the actual route of the train using the GPS information retrieved from the train's telemetry data to determine the train's location during the measurements.

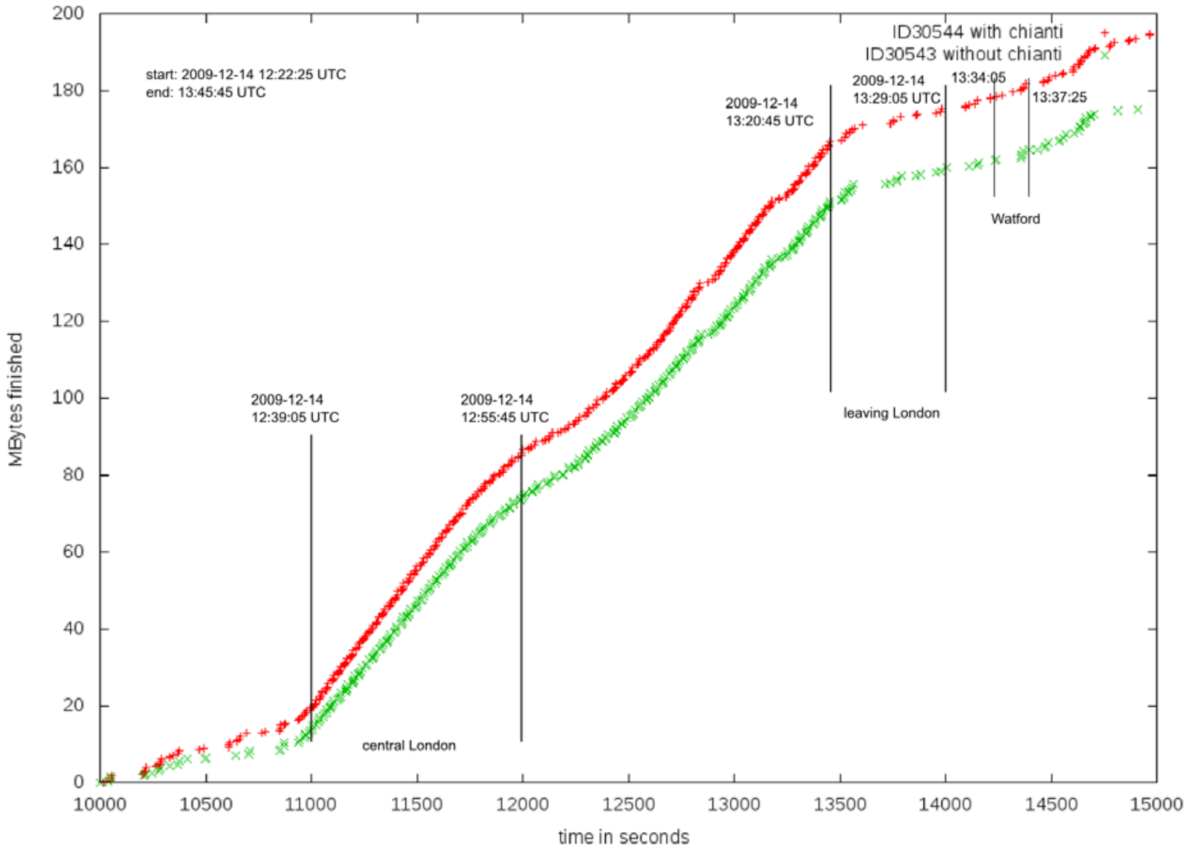

Figure 40: Completed HTTP Transfers

The graph shows a number of advantages that CHIANTI brings with the final setup deduced from previous tests (see below). First, the Nagle Algorithm [11] has been disabled to allow for small data packets being transferred immediately (cf. RFCs 896, 1122). The effect of this is a throughput optimisation for web traffic not only during periods of intermittent connectivity but also when entering WiMax coverage as seen in the central London sector of the graph in Figure 40.

More important than disabling the Nagle Algorithm to send out small packets immediately are bridging of connectivity gaps and rapid re-establishment of disrupted sessions. The effect of this functionality is visible for the Watford area where more HTTP GET requests complete through the CHIANTI service than over the plain TCP connection. A detailed view is provided in Figure 41. Within the period of 200 seconds shown here, the CHIANTI service has acknowledged 515 KB TCP payload data in 5085 IP packets while only 306 KB were acknowledged on the direct TCP connection in 2567 IP packets. The minimal packet size was 68 bytes, and most of the transferred TCP segments were of this size.

The size of individual TCP segments being acknowledged during the Watford gap is denoted by the graph in Figure 42. Table 2 gives a brief overview of the TCP statistics that have been collected from the 200 second traffic dump at the Watford gap.

 © The CHIANTI consortium – 2008-2009

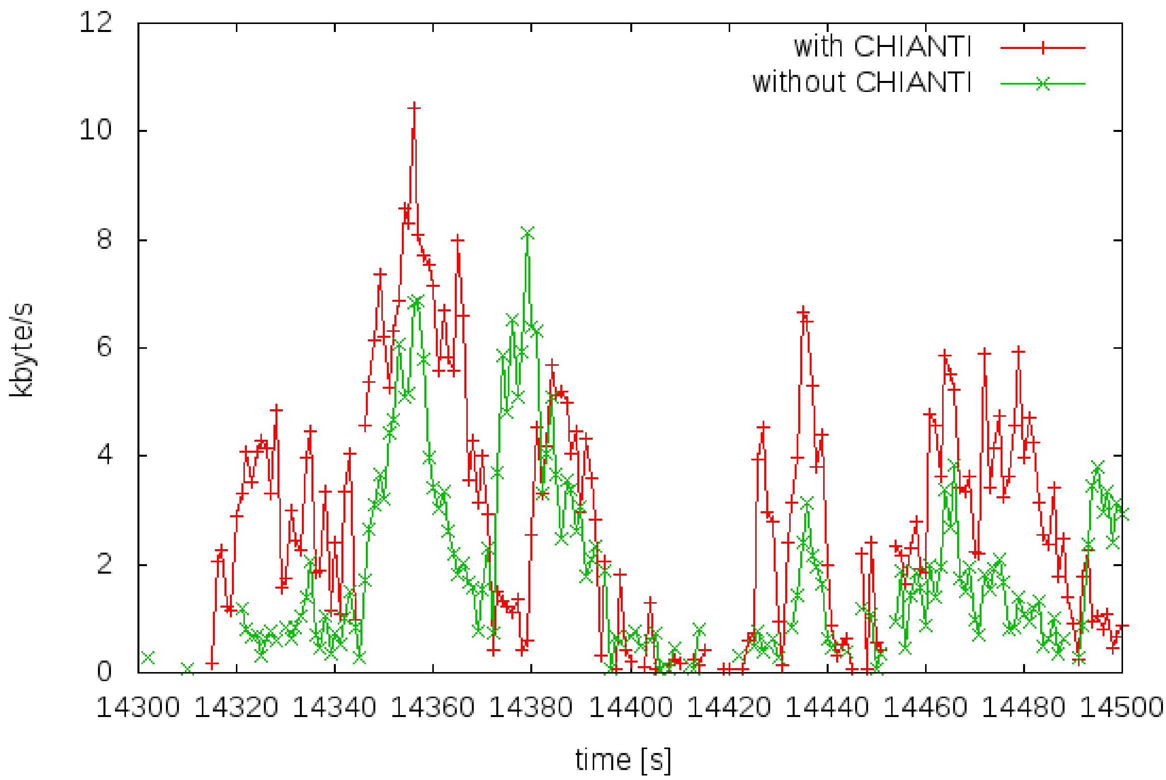

Figure 41: Received data during Watford gap (train C)

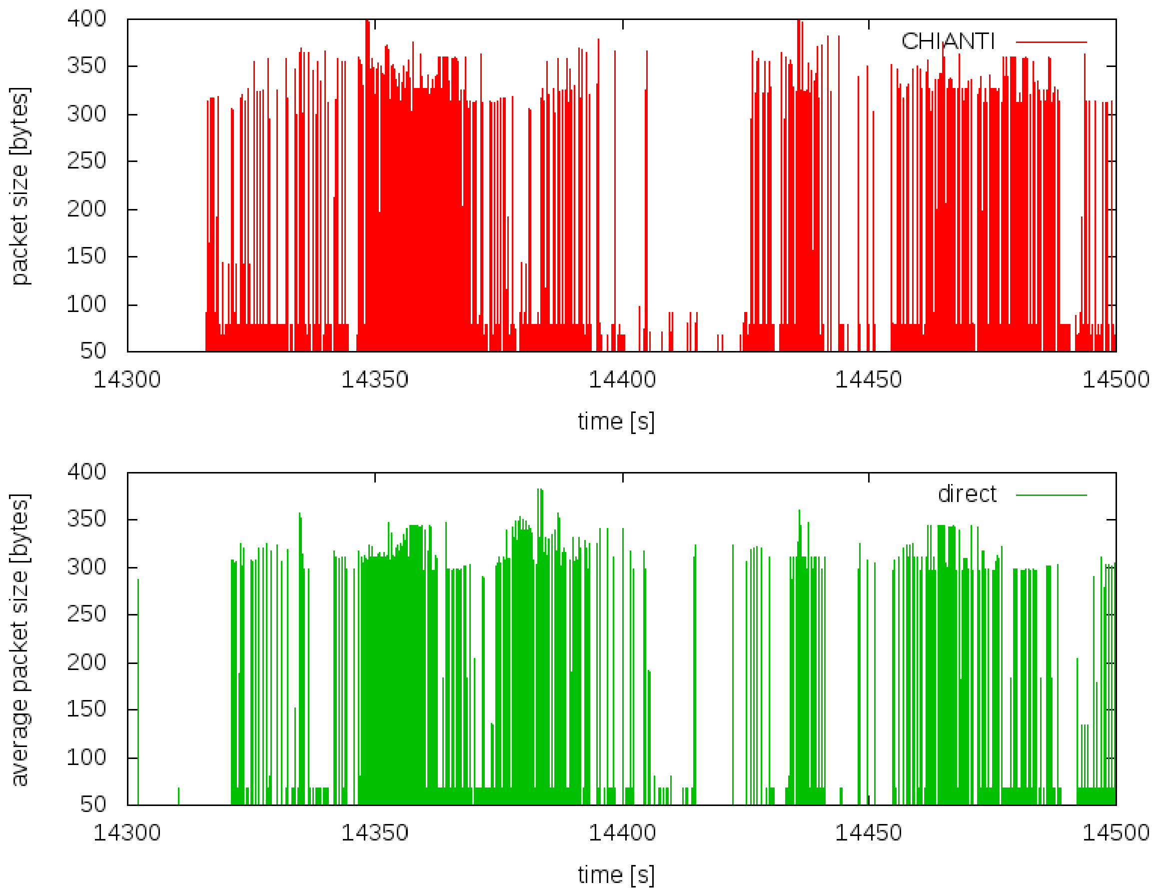

Figure 42: Size of TCP segments being acknowledged

Table 2: TCP statistics at Watford gap

	acknowl-edged [KB]	packets sent	min. seg. size [B]	max. seg. size [B]	median [B]	avg. size [B]	std dev
CHIANTI	515	5085	68	399	68	104	83
plain TCP	306	2567	68	383	68	102	97.9

These graphs show that the overhead added by the DP-Basic protocol is compensated by the additional throughput yielded by the CHIANTI service during intermittent connectivity. For example, an assumed overhead of 20 % for each packet still would result in 425 KB user data being transferred in this phase. Compared to the direct TCP connection, this was an improvement of more than 40 %.

The CHIANTI service not only helps to enhance throughput for intermittent connectivity but also improves the re-establishment of bridged connections at the end of a network outage. The graph in Figure 43 shows the behaviour of the wget-tests at the coverage gap that is illustrated in Figure 44. The coloured trail indicates different 3G networks being used during the train ride. The small image in the upper right corner in addition indicates the entire route passed by the train during the marked section in Figure 43.

In this graph, the wget-test processes yield the same throughput for the CHIANTI service and the plain TCP connection while the wireless network is present. During the outage, no data is transmitted at all. As soon as network connectivity comes back, the CHIANTI service detects the link and starts resuming its managed sessions. As the `reconnect_interval` was set to 250 ms, the TCP flows are re-established within less than 3 seconds, the minimum TCP retransmission interval (cf. RFC 1122).

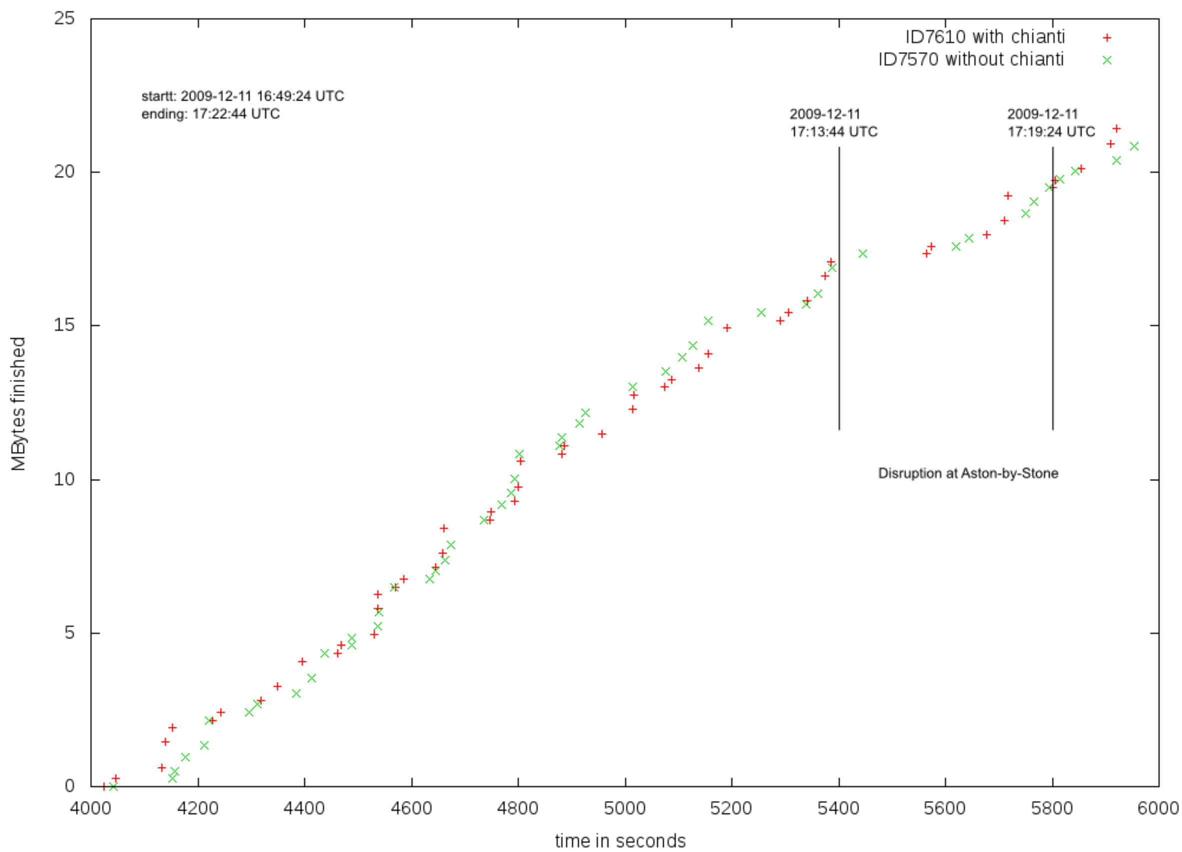

Figure 43: Disruption at Aston-By-Stone

 © The CHIANTI consortium – 2008-2009

The CHIANTI service then continues downloading of objects that have been requested by the wget-test process. As indicated in this graph, the objects are finished some seconds earlier by the CHIANTI-enabled test compared to the test using HTTP over a plain TCP connection.

Figure 44: Train Route at Aston Gap

A similar effect has been observed when train C passed Rugby on 2009-12-14. Within a period of 200 seconds, no objects are completed without CHIANTI, see Figure 45. Figure 46 shows the combined throughput of the wget-test processes in this phase. Apparently, the network conditions are too bad for a reasonable TCP performance. While the CHIANTI service has an acceptable throughput of 439 KB, only 22 KB were transferred successfully over the plain TCP connection. The aggregated TCP statistics are given in Table 3.

Table 3: TCP statistics for Rugby gap

	acknowledged [KB]	packets sent	min. seg. size [B]	max. seg. size [B]	median [B]	avg. size [B]	std dev
CHIANTI	439	4394	68	628	68	102	80.2
plain TCP	22	127	68	318	182	179	112

A closer look at the actual train route (Figure 47) reveals the existence of a network outage that lasted about 30 seconds. Several minor outages and frequent roaming between distinct 3G network providers might have interfered with the TCP RTT estimation, resulting in this inferior performance.

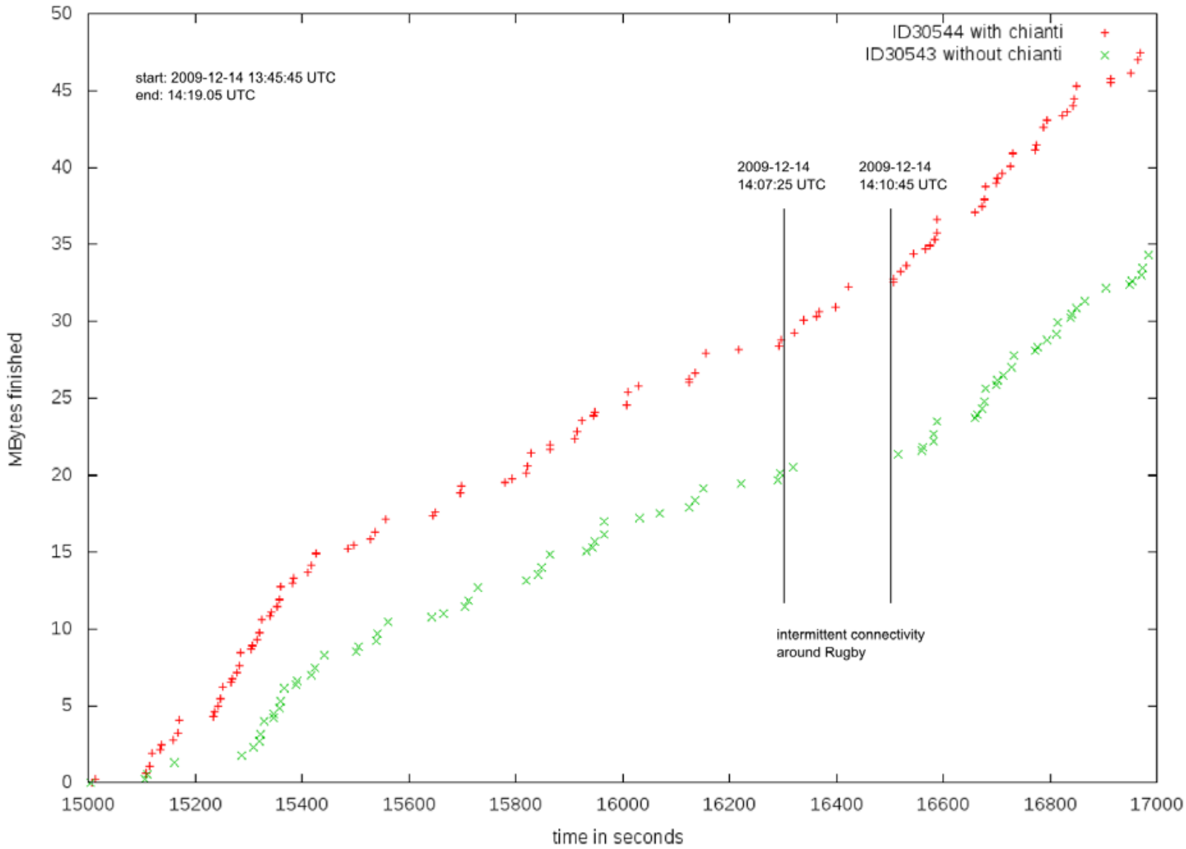

Figure 45: Intermittent Connectivity Around Rugby

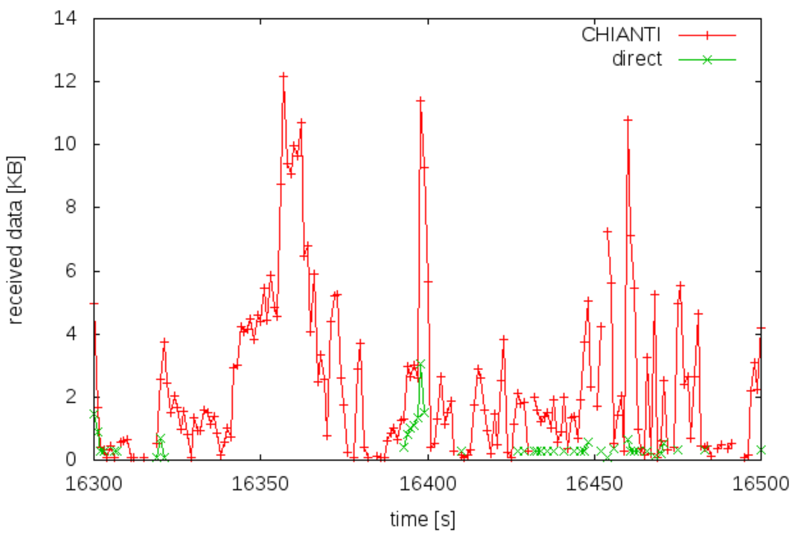

Figure 46: Throughput comparison

Figure 47: Train Route Through Rugby

These effects have been observed on various train routes during the trial phase. Figure 48 shows the results of measurements that have been done with version 0.3.01 of the CHIANTI software. Here, the CHIANTI service still performs better than plain TCP during the 400 seconds period while the train was leaving Manchester (see Figure 49). Distinct access technologies have been used along the way, starting with WiFi (green) and WiMax (red), finally switching to UMTS (yellow and blue). The throughput gain of the CHIANTI service is attributed to the switching of access technologies where CHIANTI benefits from fast link detection.

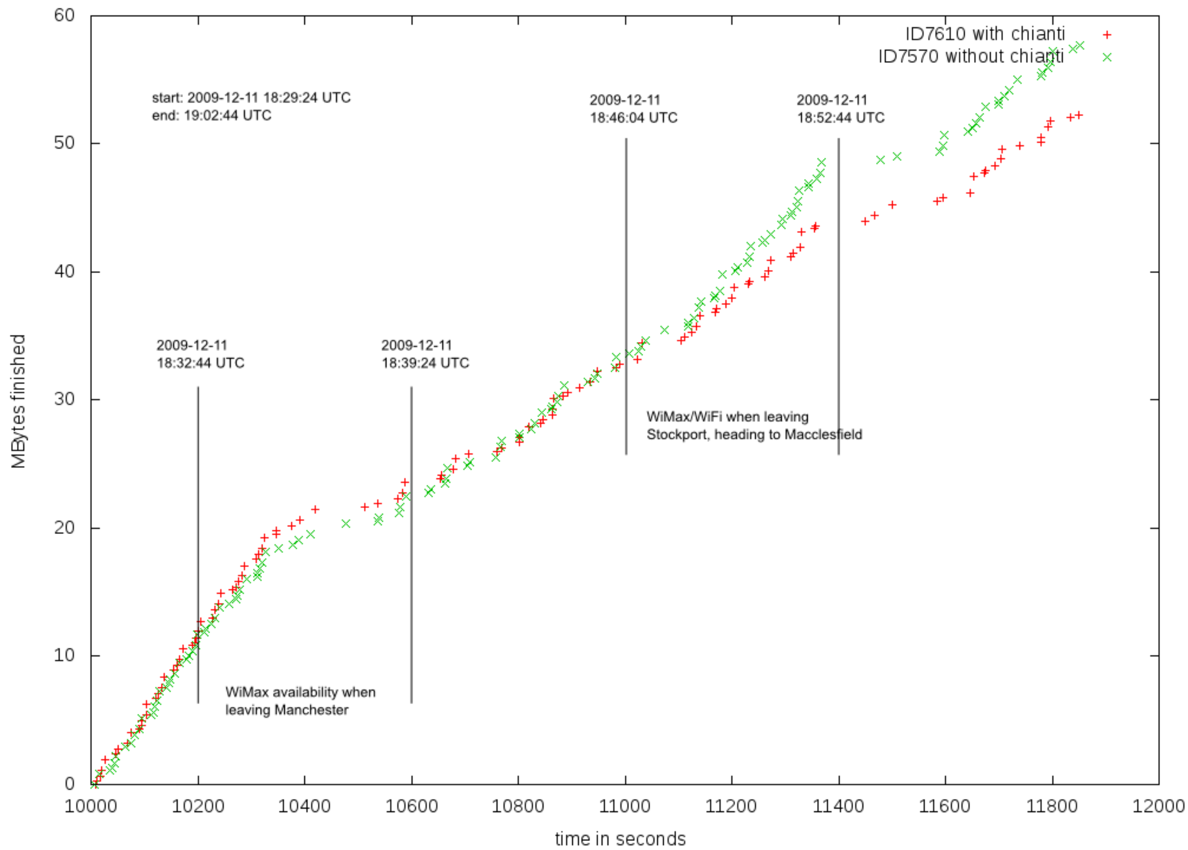

Figure 48: WiMax/WiFi Areas At Manchester Route

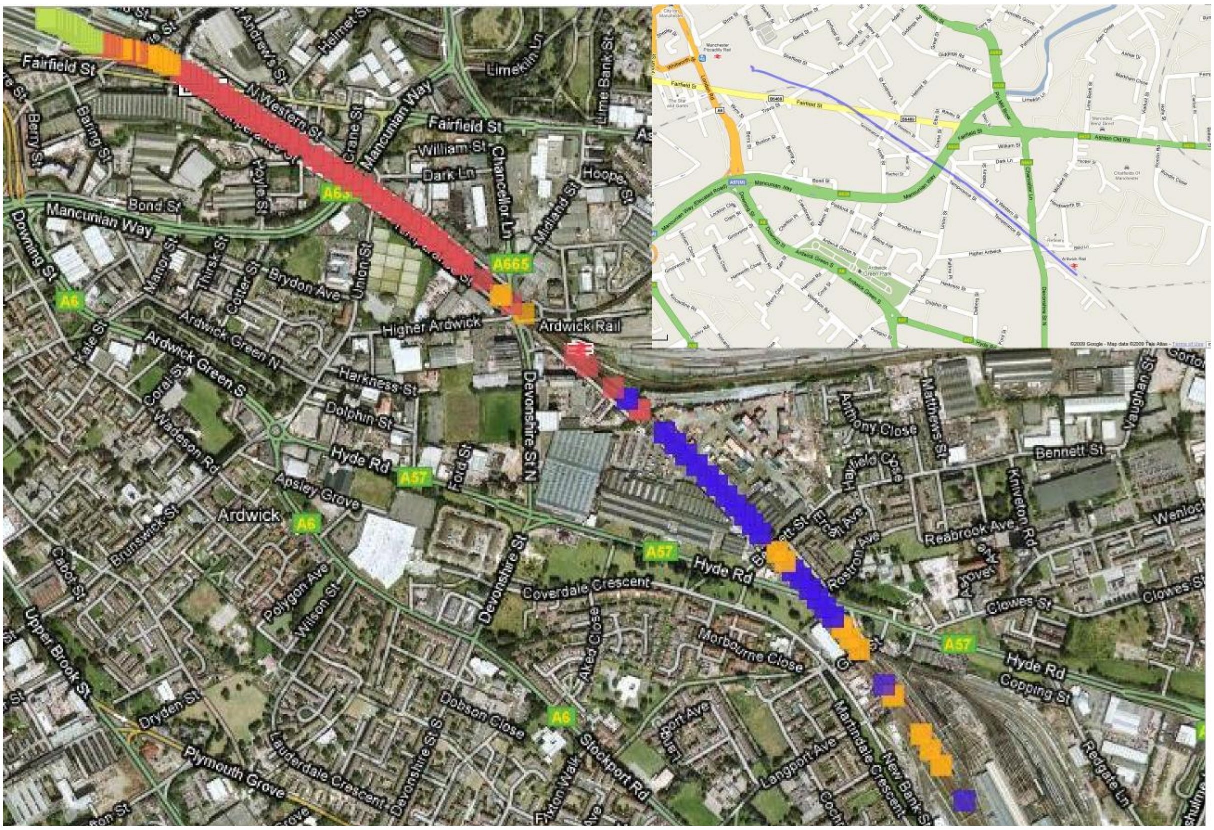

Figure 49: Train Route Through Manchester

The graph in Figure 48 also reveals that this software version was not able to outperform plain TCP all the time, unlike version 0.3.05 that has been used since December 14th, 2009 (see the observations that have been described earlier).

 © The CHIANTI consortium – 2008-2009

4 Standardisation and Dissemination

The activities 5.1 and 5.2 are continuous tasks that are carried out to increase public awareness of the project's goals and its achieved results. The project partners therefore have established links between relevant research activities inside and outside the EU and have contributed to workshops on the usability of networked applications in challenged environments.

These efforts revealed that standardisation of technologies for robust Internet access as targeted by the CHIANTI project is still in its early stages. While numerous standardisation bodies deal with node and network mobility, activities aiming at disconnection tolerance are rather rare as most of the focus is on providing ubiquitous connectivity. Therefore, the CHIANTI consortium puts its emphasis on contributions to pre-standardisation activities in the IRTF for the time being, at the same time following new developments in the IETF (e.g., on energy-efficient protocol design in the context of the RECIPE and Multipath-TCP communities).

CHIANTI partners have made contributions to currently two operationally crucial documents for the Delay-tolerant Networking Research Group and are actively following the discussions of other potentially relevant research groups.

Furthermore, the explorative research and engineering approach followed in CHIANTI helps the partners understand the upcoming issues with present applications and protocols and helps advance development of mitigation technologies (e.g., variants of performance enhancing proxies) which will feed into future standardisation activities once these become immanent.

Being part of the research programme on Information and Communication Technologies (ICT) in the EU's Seventh Framework Programme, the CHIANTI project has made various contributions to the EU's Future Internet (FI) activities. While the discussion on the Future Internet currently targets at fundamental changes to the basic architecture of the public Internet, the CHIANTI project has its focus on the near-term exploration of disruption-tolerant networking and to leverage user mobility in environments with partial network coverage.

The CHIANTI approach hence can be seen as a first step towards a more comprehensive overlay architecture to support mobility and disconnection tolerance in upcoming Future Internet activities. In this context, the MANA project already adopted the early results from CHIANTI as input for their discussion on novel overlay architectures for the Future Internet. An incremental shift towards a disconnection-tolerant architecture of a future Internet that has intrinsic mobility support has also been exemplified by the WPMC 2008 paper, using the well-recognised notion of overlays as testbed-enablers. As the CHIANTI technologies are provider-agnostic, the dissemination activities not only focus on mass transit companies but also on service providers and network operators.

The remainder of this section lists the various activities in Work Package 5. Because of the large economic relevance of short-term solutions to provide robust communication in challenged network environments, the CHIANTI project partners have been actively involved in industry fora and design workshops for the Future Internet. The background of the project as well as the insights of the project's first reporting period also have been brought into the relevant standardisation bodies, most notably the Internet Research Task Force (IRTF).

4.1 List of Standardisation Activities

The consortium made the following contributions to national and international standardisation organisations:

- Continued informal cooperation with the IRTF DTNRG, specifically participation in an interim meeting in San Francisco (trip not funded by the CHIANTI project);

© The CHIANTI consortium – 2008-2009

- K. Fall; S. Burleigh, Ed.; A. Doria; J. Ott; D. Young: The DTN URI Scheme, draft-irtf-dtnrg-dtn-uri-scheme-00, March 28, 2009;
- Mike Demmer and Jörg Ott: Delay Tolerant Networking TCP Convergence Layer Protocol, draft-irtf-dtnrg-tcp-clayer-02, November 2008.
- Participation at the 72nd IETF meeting in Stockholm, Sweden, July 2009.
- Participation in the 76th IETF in Hiroshima 9 – 13 November 2009.

4.2 List of Publications

The following papers relating to the project's foreground have been published by the consortium:

- Jörg Ott: Delay Tolerance and the Future Internet. (Invited paper,8p) 11th International Symposium on Wireless Personal Multimedia Communications, Lapland, Finland, September 2008.
- Jörg Ott: Towards More Adaptive End-to-End Applications. Keynote talk at the 3rd GI/ITG KuVS Workshop on the Future Internet. Munich, Germany, May 2009.
- Jörg Ott, Nils Seifert, Caleb Carroll, Nigel Wallbridge, Dirk Kutscher, Olaf Bergmann: *The CHIANTI Architecture for Robust Mobile Internet Access*. Proceedings of the10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Industry Track, Kos, Greece, June 2009.
- Jörg Ott: *Towards More Adaptive End-to-End Applications*. ICT Mobile Summit Workshop on Future Internet Architectures: New Trends in Service Architectures. Santander, Spain, June 2009.
- Olaf Bergmann, Stefanie Gerdes, Jörg Ott, Petri Ylikoski, Nils Seifert, Caleb Carroll, Nigel Wallbridge: *Dealing With Disruptions: Providing Internet Access to Mobile Users on a Train*, Extended Abstract, ICT that makes the difference: The future of Ambient Intelligence and ICT for Security, Brussels, Belgium, November 2009.
- Stefanie Gerdes, Olaf Bergmann: *Building a Test Environment for Emulating Link Characteristics of Disruptive Networks*, paper accepted for the IEEE Wireless Communications & Networking Conference 2010.
- Submitted journal paper "Towards More Adaptive Voice Applications" as a follow-up from the 2nd EuroNF workshop at the ICT Mobile Summit on *Future Internet Architectures: New Trends in Service Architectures*. The submission received quite positive reviews and the revised version is in preparation.

4.3 List of Dissemination Events

The consortium has participated in the following dissemination events:

- a joint meeting with members of the PSIRP project in Helsinki;
- participation in a Dagstuhl seminar on end-to-end protocols for Future Internet (organized by the TRILOGY project);
- participation in the Future Internet meeting in Bled, with contributions to the panel on content delivery; and
- contributions to the Content Delivery/Networking cluster as a follow-up activity from the Bled meeting.
- contributions to the EuroViews-2008 workshop in Würzburg;

- talk and paper at WPMC, September 2008; and
- presentation at the 2nd Concertation Meeting of ICT-FP7-FP6 , The Network of the Future, September/October 2008.
- Invited participation in an Eiffel Future Internet ThinkTank meeting in Frankfurt on 30 September and 1 October 2008.
- Joint workshop with the EU FP7 PSIRP project on 6 and 7 November 2008.
- Presentation at a meeting on "Wireless Mesh and Relay Networks" of the ITG special interest group 5.2.4 on mobile communications (2008-10-27)
- Participation in the Future Internet Assembly meeting in Madrid (2008-12-09/10)
- Lead organization and partial sponsoring of the social event (to foster informal discussion) of the Dagstuhl Seminar on Disruption- and Delay-tolerant Networking II (8 – 11 February 2009).
- Invited participation in an EIFFEL ThinkTank meeting in London on 17 and 18 February 2009.
- Continued informal cooperation with the IRTF DTNRG, specifically participation in an interim meeting in San Francisco (trip not funded by the CHIANTI project).
- Invited talk at the IET Seminar on Broadband on Trains, London, April 2009;
- keynote talk at the 3rd GI/ITG KuVS Workshop on the Future Internet. May 2009.
- participation in the Future Internet Assembly meeting in Prague, May 2009;
- Industry Track paper and talk on the10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Kos, Greece, June 2009.
- paper/talk and panel participation at the ICT Mobile Summit Workshop on Future Internet Architectures: New Trends in Service Architectures. Santander, Spain, June 2009.
- Successful participation in the DTNRG interop event for the RFC5050-compliant delay-tolerant communication protocols with the implementation developed for Symbian S60-based mobile phones to large parts developed in CHIANTI.
- Invited presentation “Towards a Delay-tolerant Future: Reconsidering some Assumptions in Networking” at TU Braunschweig, celebrating the anniversary of tubs.CITY. 2 July 2009.
- Presentation of a course about Delay Tolerant Networking and invited participation at the discussion panel of the Future Internet Summer School in Bremen, July 2009.
- Talk at the conference “ICT that makes the difference” in Brussels, November 2009.
- Invited presentation at a DTN workshop of Keio University on 14 November 2009.
- Participation in an EIFFEL ThinkTank meeting in Athens on 6 and 7 October 2009.

5 Conclusions

The CHIANTI project has targeted at demonstrating the use of existing Internet-technologies to improve the perceived quality of service for selected applications in a networked environment where continuous Internet access and seamless connectivity are not available. The project's rationale is to leverage a common notion of network disruptions as a normal trait of the mobile Internet, meaning a shift from the "always connected" paradigm to a paradigm where disruptions or degradation of network access and connectivity are accepted as routine.

The final evaluation stage has shown that the CHIANTI service can enhance utilisation of network connectivity to increase the actual throughput of user data when network coverage is available and an end-to-end connection can be established. Moreover, disrupted sessions on the transport layer will not cause application failures as long as their design allows for high round-trip times while network outages are bridged by the CHIANTI service.

The evaluation of the CHIANTI system was performed in multiple steps: First, the developed software has been tested in the lab and then validated against the user requirements specified in deliverable 1.2 [3]. With respect to the final user trial where the CHIANTI software was part of a revenue-generating Internet access service, the focus has been laid on the stability and robustness of the DP-Basic implementation. DP-Enhanced and DP-DTN therefore have been validated cursory to ensure delay-tolerance and robustness against disruptions, while feature-completeness has been sacrificed for the benefit of the final user trial in a production environment.

The second step was to evaluate the final prototype in real-world scenarios, including a user trial on various train routes in North England. As the project schedule did not allow for collection of statistical significant data from the customers of the Internet access service on those trains, and the changing network environment prohibited comparison of user-generated data traffic over time, the project consortium decided on a test methodology that has facilitated direct comparison of CHIANTI-enabled data communication and plain TCP.

To do so, the initial traffic analysis in deliverable 2.1 [4] was used as the basis for generating reference measurements that reflect the behaviour of real users of the mobile Internet. To ensure that the measurements are not falsified by implementation-specific behaviour of remote servers on the application-layer, a controlled server-environment has been setup as remote peer to test with.

In summary, the final user trials have revealed that the CHIANTI service comes with a small overhead compared to a plain TCP session. For the transfer of small objects up to 16 KB, this overhead is noticeable, but becomes less pronounced after 64KB, and disappears for 1 MB and above. For small objects, TCP slow start allows for fast transmission of user data while CHIANTI requires an additional round-trip for initial setup of its data session. In this case, the transfer of objects via plain TCP is complete before the TCP implementation even leaves the slow start phase and before the first loss is observed. For larger objects, this initial "advantage" of plain TCP becomes negligible as the two connections share the link most of the time.

Overall, our experimental assessment shows that the CHIANTI system exhibits the desired characteristics: its long-term performance comes close to TCP, but it is capable of exploiting short connectivity periods more effectively and more predictably and recovers faster from disconnections. During the final user trials, no service breakdowns have occurred due to errors in the software, system architecture or protocol design. The project thus not only met the functional requirements but also demonstrated the effective use under realistic conditions in a production environment.

 © The CHIANTI consortium – 2008-2009

Appendix A Core Requirements vs. CHIANTI Prototype III

Table 4 lists the core CHIANTI requirements as formerly defined by CHIANTI requirement definition and listed in Appendix B of CHIANTI D1.2 – Operational and User Requirements [3].

Resulting from the lab and live tests (including in-train tests), Table 4 maps the CHIANTI core requirements to the CHIANTI prototype III implementation. It thereby indicates which of these requirements are fulfilled by CHIANTI prototype III.

Table 4 is intended to be used for additional cross-checking only. Details about the measurements performed are included in the previous sections of this document.

Table 4: CHIANTI Requirements

Requirement	Description	Applies to: [2]			Reference
		WP2 Protocols	WP3 Architecture	WP4 Prototype Implementation	
R-1 CHIANTI prototype III **complies**	**Make use of the existing Internet.** Instead of waiting for the entire network to change itself. CHIANTI shall augment the network by adding protocol instances in the (virtual) path, using only a small number of components, and aiming to be of use for multiple different deployment scenarios. Thus making immediate use of the CHIANTI technology possible.#	X	X	X	[2] D1.1 sec 4, sec 5, sec 6
R-2 CHIANTI prototype III **complies**	**Make use of the existing end user applications.** Instead of forcing the end users to switch to new e-mail programs or web browsers. Thus allowing today's users and businesses to immediately introduce CHIANTI technology into their overall IT and application infrastructure.	X	X	X	[2] D1.1 sec 4, sec 5, sec 6

[2] X does apply
(X) does partly apply / compliance only partly required
– does not apply / compliance not required.

© The CHIANTI consortium – 2008-2009

Table 4: CHIANTI Requirements (continued)

Requirement	Description	Applies to: [5]			Reference
		WP2 Proto-cols	**WP3** Architec-ture	**WP4** Prototype Implemen-tation	
R-3 CHIANTI prototype III complies	**Allow CHIANTI support for CHIANTI unaware end user devices.** Within the CHIANTI Vehicle Support Scenario the designed CHIANTI architecture and protocol optimizations shall allow to provide CHIANTI optimizations even for CHIANTI unaware end user devices. Thereby commercial implementations based on the project outcomes are enabled to provide a virtually fully transparent CHIANTI service. However for the project internal prototype implementations e.g. special routing or proxy settings might be used and might cause e.g. the original client IP addresses to be hidden from e.g. Internet content servers.	X	X	(X)	[2] D1.1 sec 5, sec 6
R-4 CHIANTI prototype III complies	**Uninterrupted service operation across disconnections of more than 5 min.** Perceived uninterrupted (but delayed) service operation across disconnections of more than 5 minutes (due to loss of connectivity, switching between access technologies, changing IP addresses). Evaluation shall be done with e-transactions (sending via SMTP, receiving via POP3). However, depending on the email software used (e.g. Windows Mail), the end user will have to acknowledge an application (GUI) notification, that the transfer should be continued after outages exceeding e.g. 1 minute.	X	X	X	[3] D1.2 sec 2.4, sec 4

© The CHIANTI consortium – 2008-2009

Table 4: CHIANTI Requirements (continued)

Requirement	Description	Applies to: [5]			Reference
		WP2 Proto-cols	WP3 Architec-ture	WP4 Prototype Implemen-tation	
R-5 CHIANTI prototype complies	**Throughput optimization for dis-rupted connectivity.** Significantly increase user data throughput during disruptive network connectivity periods. Expected in-crease of user data throughput when using CHIANTI optimizations: above 30%. (To be measured with POP3 e-mail reception over 5.5 min. Interval with each 30 sec. of network availability followed by 30 sec. of network distrup-tion.)	X	X	X	[3] D1.2 sec 2.4, sec 4
R-6 CHIANTI prototype complies	**Maintaining loss less video stream-ing despite network disruptions.** Tolerate up to 5 minutes of disconnec-tion time for video streaming. (E.g. for CCTV video streaming for passenger and staff safety with central storage outside of trains.)	X	X	X	[3] D1.2 sec 2.5.3, sec 4 [2] D1.1 sec 5
R-7 CHIANTI prototype complies	**Provide Disruption tolerance op-timizations on the TCP layer.** Aiming at providing many of the TCP based application protocols with gains caused by TCP disruption tolerance optimizations.	X	X	X	[3] D1.2 sec 2.1, sec 2.4, sec 2.5.1

© The CHIANTI consortium – 2008-2009

Table 4: CHIANTI Requirements (continued)

Requirement	Description	Applies to: [5] WP2 Protocols	 WP3 Architecture	 WP4 Prototype Implementation	Reference
R-8 **not applying** to implementation	**Allow intermediate Mobile IP.** CHIANTI optimization software shall allow to be used within scenarios that do already implement mobile IP. (Mobile IP between CHIANTI Client and CHIANTI Proxy. Support will not be provided for all CHIANTI usage scenarios.)	(X)	X	–	[3] D1.2 sec 3.1, sec 2.5.2
R-9 CHIANTI prototype **complies**	**Do not depend on constant IP.** CHIANTI optimizations should be applicable for network scenarios that do not provide constant IP addresses after network outages/after switching between networks. (The CHIANTI service shall recover from changing IP addresses of the CHIANTI client.)	X	X	X	[3] D1.2 sec 3.1, sec 2.5.2
R-10 CHIANTI prototype **complies**	**Do not prevent intermediate encryption.** Do not prevent usage of intermediate typical VPN/encryption functionality when encryption is applied between the CHIANTI client and CHIANTI proxy. Allow the CHIANTI-P protocol data to be IPsec or IP-VPN encrypted by basing the CHIANTI-P data exchange on standard protocols.	X	X	X	[3] D1.2 sec 2.3, sec 2.4 [2] D1.1 sec 5.1.1, sec 5.1.2

Table 4: CHIANTI Requirements (continued)

Requirement	Description	Applies to: [5] WP2 Protocols	WP3 Architecture	WP4 Prototype Implementation	Reference
R-11 **not applying** to implementation	**Keep CHIANTI architecture open to include authentication portals.** The CHIANTI Architecture should allow the implementation of Authentication portals (allowing end users to authenticate themselves to a CHIANTI vehicle support infrastructure).	(X)	X	–	[3] D1.2 sec 3.4
R-12 CHIANTI prototype **complies**	**NAT towards end user application.** Keep the architecture open to allow NAT between mobile end user device / application and CHIANTI client. (However like for NAT usage in general the applications / application protocols need to allow NAT usage independent of CHIANTI. The CHIANTI prototype implementation shall not prevent NAT usage in general, but does not need to explicitly support e.g. certain NAT-traversal methods.)	X	X	(X)	[3] D1.2 sec 3.2
R-13 CHIANTI prototype **complies**	**Allow NAT between CHIANTI Client and CHIANTI proxy.** The CHIANTI protocol design for CHIANTI-P shall consider NAT between the CHIANTI Client and CHIANTI proxy.	X	X	(X)	[3] D1.2 sec 3.2

 © The CHIANTI consortium – 2008-2009

Table 4: CHIANTI Requirements (continued)

Requirement	Description	Applies to: [5] WP2 Proto-cols	 WP3 Architec-ture	 WP4 Prototype Implemen-tation	Reference
R-14 CHIANTI prototype complies	**Allow for basic firewall traversal between CHIANTI Client and CHI-ANTI proxy.** The CHIANTI protocol design for CHI-ANTI-P shall not prevent all basic fire-wall setups between the CHIANTI Cli-ent and CHIANTI proxy.	X	X	(X)	[3] D1.2 sec 3.3
R-15 not applying to implementation	**Keep CHIANTI architecture open to include content filtering.** CHIANTI service providers might want or have to implement content filtering functionality (e.g. Websense based filtering of adult-content). The CHIANTI architecture shall docu-ment possibilities to add content filter-ing mechanisms. Thereby commercial implementations based on the project outcomes are enabled to provide a virtually fully transparent CHIANTI ser-vices including content filtering func-tionality. (See remarks of requirement R-3 as well.)	(X)	X	–	[3] D1.2 sec 3.5

© The CHIANTI consortium – 2008-2009

Table 4: CHIANTI Requirements (continued)

Requirement	Description	Applies to: [5] WP2 Proto-cols	WP3 Architec-ture	WP4 Prototype Implemen-tation	Reference
R-16 **not applying** to implementation	**Keep CHIANTI architecture open to include legal interception.** CHIANTI service providers might want or have to implement legal interception functionality. The CHIANTI architecture shall document possibilities to add legal interception mechanisms. Thereby commercial implementations based on the project outcomes are enabled to provide a virtually fully transparent CHIANTI services including legal interception and end user identification. However the project internal prototype implementations will not yet implement legal interception tolls and e.g. special routing or proxy settings require by the prototype implementation might cause e.g. the original client IP addresses to be hidden from Internet content servers.	(X)	X	–	[3] D1.2 sec 3.6
R-17 CHIANTI prototype III **complies**	**Consider high RTTs for CHIANTI protocol design.** Today's mobile networks might provide delay intensive connectivity only. According to the general CHIANTI deployability approach, CHIANTI does not aim to exclude these networks from CHIANTI support or to wait for these network characteristics to change. For the CHIANTI optimization protocol design (CHIANTI-P) high RTTs shall be considered. However certain high RTTs might be treated like network outages. Independent of this requirement for the CHIANTI protocol design, the application service quality will suffer with increasing network delays.	X	X	X	[3] D1.2 sec 5

© The CHIANTI consortium – 2008-2009