



Project Number:	FP7-257123
Project Title:	CONVERGENCE
Deliverable Type:	Report
Dissemination level	Public
Deliverable Number:	D8.4
Contractual Date of Delivery to the CEC:	Month 33 (amended to Month 34) 31.03.13
Actual Date of Delivery to the CEC:	02.05.13
Title of Deliverable:	Report on trials and experimentations after cycle 3
Workpackage contributing to the Deliverable:	WP 8
Editor:	Richard Walker (XIW)
Author(s):	B. Benincasa, A. Pede, R. di Fuccio (XIW), A. Detti, M. Bonola, B. Ricci, A. Caponi, G. Morabito, D. Tassetto, G. Tropea, N. Blefari Melazzi (CNIT), A. –C. G. Anadiotis, C. Z. Patrikakis, I. S. Venieris (ICCS), Carsten Rust (MORPHO)
Abstract:	In this report we describe the results of the third round of the CONVERGENCE end-user trials and the second (final) round of network experiments. We conclude that with improvements in visual design and technical functionality the CONVERGENCE applications could provide a valid solution for user needs. The network experiments and simulations show how CONVERGENCE network (CoNet) supports massive content distribution while simultaneously limiting network overhead, maintaining scalability, and simplifying the effort to deploy data dissemination services.
Keyword List:	User, assessment, evaluation, usefulness, visual design, CONVERGENCE framework, performance, scalability, caching, routing, ICN

Executive Summary

Overview

In this report we summarize the results of the third round of CONVERGENCE user trials and present the results of a study of the performance, functionality and robustness of COMID (CONVERGENCE Middleware) and CONET (Information Centric Network).

The study is organized into two chapters dedicated respectively to the end-user trials (Phase 1-Track 3) and to network experiments and simulation (Phase 2 – Track 2).

End-user trials

In the third round of end-user trials, our goal was to measure user satisfaction in a real-world situation and to answer seven crucial questions:

1. How does the CONVERGENCE framework behave under real-life conditions?
2. If so, are the CONVERGENCE applications tested in the trials robust?
3. To what degree do the prototype applications meet the needs of target groups?
4. What are their most important and satisfying features?
5. Which features require improvement?
6. Could CONVERGENCE and its applications represent a valid alternative to existing systems?

To reach the widest potential audience, including potential participants who may initially be ill disposed toward the project, we decided to base our evaluation on a simple online questionnaire. Participants were given a list of tasks to perform with the application under test (see Table 1) and asked to complete the tasks before filling in the questionnaire. Unlike in the first two trials they did not receive any preliminary information about CONVERGENCE.

During the trials **563** users used the applications: **220** (39%) also compiled the survey (see Table 2). The majority were male (62.7%), in the age range 21-40 (70,4%), had a University education (93,0%) and saw themselves as high skilled uses of computer and the Internet (75,4%).

Our results show that although some aspects of the system need improvement, **CONVERGENCE's response to the needs expressed by trial participants is better than satisfactory.**

The prototypes satisfied participants' requirements: users assigned the applications a mean overall satisfaction score of more than 3.5. The applications received good scores for all items in the questionnaire. The items with the highest scores (Technical Reliability and Ease of Use) were also the aspects of the applications that users considered most important.

- 1. In terms of functionality and technical performance, participants in the third round of trials gave the applications a much higher score than they received in the second round.** The main focus of the study was on the applications' technical characteristics which the trial measured in terms of Usefulness, Performance and Technical Reliability. These were not only the characteristics which received the highest scores from participants (mean scores around 4) but also the characteristics to which users attached the greatest importance (mean scores around 4 again). This result is evidence for the solidity and "robustness" of the CONVERGENCE framework: a theme we will return to in the following chapter. Positive user reports are confirmed by the objective data on application performance: system logs show very few failures even when the applications were used continuously by a significant number of concurrent users. This is a major improvement with respect to the frequent crashes experienced during the second round of trials.
- 2. Most of the target groups found that the applications were easy to use.** They also considered this to be an extremely important characteristic of the application. This represents a major improvement with respect to previous trials.
- 3. The visual design and the help functions still need improvement.** These aspects of the applications received lower scores than the others and were considered by some users (e.g., those at LMU) as less than satisfactory. This result confirms user feedback from the previous trials. We note, however, that users did not consider these to be the most important aspects of the application.
- 4. Industrialized versions of the applications have the potential to compete with similar products from other producers.** The majority of participants believe that the CONVERGENCE applications can compete with similar products from other suppliers. In this respect they consider CONVERGENCE to be relatively satisfactory (score 3).

Considering these results together, we conclude that with improvements in visual design and graphics, the CONVERGENCE applications could represent a valid alternative to the tools the target populations uses currently. This conclusion matches findings from earlier trials showing that users had a strong interest in the services CONVERGENCE was offering and that with some changes to their design and technical characteristics the CONVERGENCE applications could offer a valid solution to users' requirements.

Network experiments and simulations

The second round of network experiments and simulations addresses the question: "is it possible to design a network architecture that supports massive content distribution while simultaneously limiting network overhead, maintaining scalability, and simplifying the effort to deploy data dissemination services?" In what follows, we present a practical use-case (a video-streaming scenario) showing that the CONVERGENCE network (CoNet) has the potential to fulfil this need.

The study used the infrastructure provided by PlanetLab Europe (<http://www.planet-lab.eu/>), a global facility for the deployment and test of experimental network services for the Future Internet. The study investigated three scenarios: a basic publish-subscribe scenario, a scenario demonstrating in-network caching and a scenario showing simple content replication. In all three cases, the trial demonstrated that CONVERGENCE could greatly facilitate network administration.

The study goes on to describe a P2P ICN application for live streaming of videos encoded at multiple bitrates (aka adaptive streaming) to mobile devices. The streaming application can be deployed either on top of CONET [7], or on top of a CCN network [6] implemented with the CCNx tool [8].

The study goes on to describe further tests and analyses on a broad range of issues including ICN video streaming for cellular environments; naming, content integrity and caching, alternative routing protocols (epidemic protocols), CONET support for peer to peer content sharing, and CONET support for adaptive video streaming.

INDEX

EXECUTIVE SUMMARY	2
OVERVIEW	2
END-USER TRIALS	2
NETWORK EXPERIMENTS AND SIMULATIONS	3
GLOSSARY	7
1 GOALS AND STRUCTURE OF THIS DOCUMENT	14
2 TRACK 1 PHASE 3 - USERS TRIAL	16
2.1 OVERVIEW	16
2.2 EVALUATION METHODOLOGY	17
2.2.1 Evaluation Tools	17
2.2.1.1 Online questionnaire	18
2.3 RECRUITMENT	19
2.3.1 Scenario Walkthroughs	21
2.3.2 Data collection and data analysis	22
2.4 DEMOGRAPHIC DISTRIBUTION	24
2.5 RESULTS OF ANALYSIS	25
2.5.1 Impact of population classes vs. features	25
2.5.2 User satisfaction	27
2.5.3 Application performance	35
2.5.4 Overall Satisfaction	35
2.5.5 General considerations	37
2.6 CONCLUSIONS	39
3 TRACK 2 PHASE 2 - NETWORK EXPERIMENTS AND SIMULATIONS	41
3.1 NETWORK FINAL TEST-BED	41
3.1.1 Test-bed description	41
3.1.2 Software setup	44
3.1.3 Workflows	44
3.1.3.1 Scenario one: basic publish-subscribe	44
3.1.3.2 Scenario two: in-network caching	46
3.1.3.3 Scenario three: simple content replication	46
3.2 PROGRESS ON SPECIFIC ISSUES	46
3.2.1 ICN video streaming for cellular environments	46
3.2.1.1 Scenario and assumptions	47
3.2.1.2 The streaming scheme	48

3.2.1.3	Video server and naming scheme	48
3.2.1.4	Video Peer	49
3.2.1.5	Test-bed results	53
3.2.1.6	Related Works and Conclusions	58
3.2.2	Naming, content integrity and caching	58
3.2.2.1	Naming and Content Integrity	59
3.2.2.1.1	Possible combinations of naming schemes and signature approaches	59
3.2.2.1.2	Overhead and verification time	61
3.2.2.2	Content integrity and caching performance	62
3.2.2.2.1	Single cache analysis	63
3.2.2.2.2	Cache network analysis	66
3.2.2.3	Conclusions	67
3.2.3	Alternative routing protocols	67
3.2.3.1	Motivation and background	68
3.2.3.2	Algorithm for degree-driven epidemic routing	69
3.2.3.3	Preliminary results	71
3.2.3.4	Conclusions	76
3.2.4	CONET Support for Peer-to-Peer Content Sharing in fixed networks	77
3.2.5	CONET Support for Adaptive Video Streaming in fixed networks	77
4	BIBLIOGRAPHY	78

Glossary

Term	Definition
Access Rights	Criteria defining who can access a VDI or its components under what conditions.
Advertise	Procedure used by a CoNet user to make a resource accessible to other CoNet users.
Application	Software, designed for a specific purpose that exploits the capabilities of the CONVERGENCE System.
Business Scenario	A scenario describing a way in which the CONVERGENCE System may be used by specific users in a specific context or, more narrowly, a scenario describing the products and services bought and sold, the actors concerned and, possibly, the associated flows of revenue in such a context.
CA	Central Authority
CCN	Content Centric Network
Cl_Auth_SC	Client Authentication with Smart Card (Challenge Response)
Cl_Auth_User_Pw	Client Authentication with Username and Password
Clean-slate architecture	The CONVERGENCE implementation of the Network Level, totally replacing existing IP functionality. See “Integration Architecture” and ”“Overlay Architecture” and “Parallel Architecture”.
CoApp	The CONVERGENCE Application Level.
CoApp Provider	A user providing Applications running on the CONVERGENCE Middleware Level (CoMid).
CoMid	The CONVERGENCE Middleware Level.
CoMid Provider	A user providing access to a single or an aggregation of CoMid services.
CoMid Resource	A virtual or physical object or service referenced by a VDI, e.g. media, Real World Objects, persons, internet services. It has the same meaning of “Resource” and it is used only to better specify the term “Resource” when there is a risk of a misunderstanding with the term “CoNet Resource”.
Community Dictionary	A CoMid Technology Engine that provides all the matching

Service (CDS)	concepts in a user's subscription, search request and publication.
CoNet Provider	A user providing access to CoNet services, i.e. the equivalent of an Internet Service Provider.
CoNet Resource	A resource of the CoNet that can be identified by means of a name; resources may be either Named-data or a Named service access point.
Content-based resource discovery	A user request for resources, either through a subscription or a search request to the CONVERGENCE system (from literature). See "subscription" and "search".
Content-based Subscription	A subscription based on a specification of user's preferences or interests, (rather than a specific event or topic). The subscription is based on the actual content, which is not classified according to some predefined external criterion (e.g., topic name), but according to the properties of the content itself. See "Subscription" and "Publish-subscribe model".
Content-centric	A network paradigm in which the network directly provides users with content, and is aware of the content it transports, (unlike networks that limit themselves to providing communication channels between hosts).
CONVERGENCE Applications level (CoApp)	The level of the CONVERGENCE architecture that establishes the interaction with CONVERGENCE users. The Applications Level interacts with the other CONVERGENCE levels on behalf of the user.
CONVERGENCE Computing Platform level (CoComp)	The Computing Platform level provides content-centric networking (CoNet), secure handling (CoSec) of resources within CONVERGENCE and computing resources of peers and nodes.
CONVERGENCE Core Ontology (CCO)	A semantic representation of the CoReST taxonomy. See "CONVERGENCE Resource Semantic Type (CoReST)"
CONVERGENCE Device	A combination of hardware and software or a software instance that allows a user to access Convergence functionalities
CONVERGENCE Engine	A collection of technologies assembled to deliver specific functionality and made available to Applications and to other Engines via an API
CONVERGENCE Middleware level (CoMid)	The level of the CONVERGENCE architecture that provides the means to handle VDIs and their components.
CONVERGENCE	The Content Centric component of the CONVERGENCE

Network (CoNet)	Computing Platform level. The CoNet provides access to named-resources on a public or private network infrastructure.
CONVERGENCE node	A CONVERGENCE device that implements CoNet functionality and/or CoSec functionality.
CONVERGENCE peer	A CONVERGENCE device that implements CoApp, CoMid, and CoComp (CoNet and CoSec) functionality.
CONVERGENCE Resource Semantic Type (CoReST)	A list of concepts or terms that makes it possible to categorize a resource, establishing a connection with the resource's semantic metadata.
CONVERGENCE Security element (CoSec)	A component of the CONVERGENCE Computing Platform level implementing basic security functionality such as storage of private keys, basic cryptography, etc.
CONVERGENCE System	A system consisting of a set of interconnected devices - peers and nodes - connected to each other built by using the technologies specified or adopted by the CONVERGENCE specification. See "Node" and "Peer".
Dec_Key_Unwrap	Key Unwrapping and Content Decryption
DIDL	Digital Item Description Language
Digital forgetting	A CONVERGENCE system functionality ensuring that VDIs do not remain accessible for indefinite periods of time, when this is not the intention of the user.
Digital Item (DI)	A structured digital object with a standard representation, identification and metadata. A DI consists of resource, resource and context related metadata, and structure. The structure is given by a Digital Item Declaration (DID) that links resource and metadata.
Domain ontology	An ontology, dedicated to a specific domain of knowledge or application, e.g. the W3C Time Ontology and the GeoNames ontology.
Elementary Service (ES)	The most basic service functionality offered by the CoMid.
Enc_Key_Wrap	Encryption and Key Wrapping
Entity	An object, e.g. VDIs, resources, devices, events, group, licenses/contracts, services and users, that an Elementary Service can act upon or with which it can interact.
Expiry date	The last date on which a VDI is accessible by a user of the CONVERGENCE System.

Fractal	A semantically defined virtual cluster of CONVERGENCE peers.
Group_Sig	Group Signature
ICN	Information Centric Network
Identifier	A unique signifier assigned to a VDI or components of a VDI.
Integration Architecture	An implementation of CoNet designed to integrate CoNet functionality in the IP protocol by means of a novel IPv4 option or by means of an IPv6 extension header, making IP content-aware. See “Clean-state Architecture”, “Overlay Architecture”, “Parallel Architecture”
IP	Identity Provider
License	A machine-readable expression of Operations that may be executed by a Principal.
Local named resource	A named-resource made available to CONVERGENCE users through a local device, permanently connected to the network. Users have two options to make named-resources available to other users: 1) store the resource in a device, with a permanent connection to the network; 2) use a hosting service. In the event she chooses the former option, the resource is referred to as a local named-resource.
Metadata	Data describing a resource, including but not limited to provenance, classification, expiry date etc.
MPEG eXtensible Middleware (MXM)	A standard Middleware specifying a set of Application Programming Interfaces (APIs) so that MXM Applications executing on an MXM Device can access the standard multimedia technologies contained in the Middleware as MXM Engines.
MPEG-M	An emerging ISO/IEC standard that includes the previous MXM standard.
Multi-homing	In the context of IP networks, the configuration of multiple network interfaces or IP addresses on a single computer.
Named resource	A CoNet resource that can be identified by means of a name. Named-resources may be either data (in the following referred to as “named-data”) or service-access-points (“named-service-access-points”).
Named service access point	A kind of named-resource, consisting of a service access point identified by a name. A named-service-access-point is a network endpoint identified by its name rather than by the Internet port numbering mechanism.

Named-data	A named-resource consisting of data.
Network Identifier (NID)	An identifier identifying a named resource in the CONVERGENCE Network. If the named resource is a VDI or an identified VDI component, its NID may be derived from the Identifier (see “Identifier”).
Overlay architecture	An implementation of CoNet as an overlay over IP. See “Clean-state Architecture” and “Integration Architecture” and “Parallel Architecture”
Parallel architecture	An implementation of CoNet as a new networking layer that can be used in parallel to IP. See “Clean-state Architecture” and “Integration Architecture” and “Overlay Architecture”
PKI	Public Key Infrastructure
Policy routing	In the context of IP networks, a collection of tools for forwarding and routing data packets based on policies defined by network administrators.
Principal (CoNet)	The user who is granted the right to use a <i>CoNet Principal Identifier</i> for naming its named resources. For example, the principal could be the provider of a service, the publisher or the author of a book, the controller of a traffic lights infrastructure, or, in general, the publisher of a VDI. A Principal may have several Principal Identifiers in the CoNet.
Principal (Rights Expression Language)	The User to whom Permissions are Granted in a License.
Principal Identifier (CoNet)	The Principal identifier is a string that is used in the Network Identifiers (NID) of a CoNet resource, when the NID has the form: NID = <namespace ID, hash (Principal Identifier), hash (Label)> In this approach, hash (Principal Identifier) must be unique in the namespace ID, and Label is a string chosen by the principal in such a way that hash(Label) is unique for in the context of the Principal Identifier.
Publish	The act of informing an identified subset of users of the CONVERGENCE System that a VDI is available.
Publisher	A user of CONVERGENCE who performs the act of publishing.
Publish-subscribe model	CONVERGENCE uses a content-based approach for the publish-subscribe model, in which notifications about VDIs are delivered to a subscriber only if the metadata / content of those VDIs match

	constraints defined by the subscriber in his Subscription VDI.
Real World Object	A physical object that may be referenced by a VDI.
REL	Rights Expression Language
Resource	A virtual or physical object or service referenced by a VDI, e.g. media, Real World Objects, persons, internet services.
Scope (in the context of routing)	In the context of advertising and routing, the geographical or administrative domain on which a network function operates (e.g. a well-defined section of the network - a campus, a shopping mall, an airport -, or to a subset of nodes that receives advertisements from a service provider).
Search	The act through which a user requests a list of VDIs meeting a set of search criteria (e.g. specific key value pairs in the metadata, key words, free text etc.).
Serv_Auth	Server Authentication without Smart Card
Service Level Agreement (SLA)	An agreement between a service provider and another user or another service provider of CONVERGENCE to provide the latter with a service whose quality matches parameters defined in the agreement.
Sig	Signature
Smart_Card Role_Auth_SC	Role Authentication towards Smart Card
SP	Service Provider
Subscribe	The act whereby a user requests notification every time another user publishes or updates a VDI that satisfies the subscription criteria defined by the former user (key value pairs in the metadata, free text, key words etc.).
Subscriber	A user of CONVERGENCE who performs the act of subscribing.
Timestamp	A machine-readable representation of a date and time.
Tool	Software providing a specific functionality that can be re-used in several applications.
Trials	Organized tests of the CONVERGENCE System in specific business scenarios.
Un-named-data	A data resource with no NID.
Us_Reg_IP	User Registration to Identity Provider
Us_Reg_SP	User Registration to Service Provider

User	Any person or legal entity in a Value-Chain connecting (and including) Creator and End-User possibly via other Users.
User (in OSI sense)	In a layered architecture, the term is used to identify an entity exploiting the service provided by a layer (e.g. CoNet user).
User ontology	An ontology created by CONVERGENCE users when publishing or subscribing to a VDI.
User Profile	A description of the attributes and credentials of a user of the CONVERGENCE System.
Versatile Digital Item (VDI)	A structured, hierarchically organized, digital object containing one or more resources and metadata, including a declaration of the parts that make up the VDI and the links between them.

1 Goals and structure of this document

In this report we summarize the results of the third round of CONVERGENCE user trials and present the results of a study of the performance, functionality and robustness of COMID (CONVERGENCE Middleware) and CONET (Information Centric Network).

The study is organized into two chapters dedicated respectively to the end-user trials (Phase 1-Track 3) and to network experiments and simulation (Phase 2 – Track 2).

Chapter 2 summarizes the end-user trials, as described in Figure 1 below. After describing the main questions the trial aims to answer, the first part of the chapter outlines our methodological framework and describes our data collection tools, recruitment strategies and the data analysis. The second part summarizes the results which we compare with the results obtained in previous trials. We go on to offer a tentative interpretation of some of the trends observed.

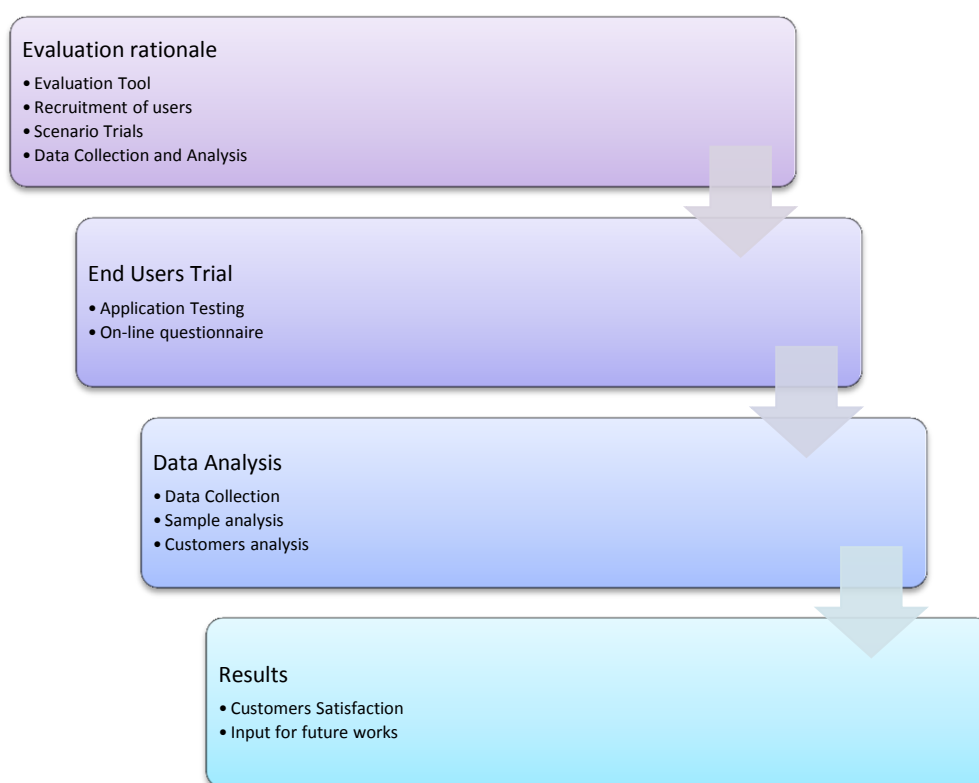


Figure 1- Evaluation Structure

Chapter 3 describes Track 2 Phase 2. The first part of the chapter presents the Track 2 demo, which we will show at the final review. The report goes on to describe a series of network experiments testing the effectiveness of the CONVERGENCE Middleware (i.e. COMID) and Information Centric Network (i.e. CONET) in supporting CONVERGENCE applications in a distributed network environment. The experiments tested both functionality and performance.

Most of the functions specified in the CoMid API (see D3.3, Section 7.1.6) have been successfully tested. Those that were not tested (e.g. `sendToLocation`) made no important contribution to the applications. We go on to describe a test of the effectiveness of the complete CoMid/CoNet stack using a demo application (a Video Distribution Service). The main focus is on the scalability of ICN networking and the CONVERGENCE publish-subscribe infrastructure, throughput during data transfer and the performance of specific ICN applications including video distribution and p2p file sharing.

The second part of the chapter describes a series of additional studies carried out during the last months of the project. These include an investigation of DASH video streaming on an ICN network; an analysis of the interplay between naming, content integrity and caching; and a comparative study of routing protocols for CONET.

2 TRACK 1 PHASE 3 - USERS TRIAL

2.1 Overview

Following criticisms from reviewers, the third round of CONVERGENCE introduced a number of novelties compared to earlier rounds. In particular:

1. The trials involved a higher number of participants, who participated via the web.
2. They did not include a focus group.
3. The questionnaires were simpler – improving the quality of the information collected from participants.
4. The standard trial report form was shorter.

Given that the main focus CONVERGENCE is on middleware and content centric networking, the primary goal of the trials was to test whether the middleware could support the development of applications capable of attracting potential users. Given this goal, the design of the user interfaces was deliberately rudimentary and did not have the usability we would expect in a commercial product. In these circumstances, we decided not to invest in systematic usability testing. Such testing, we believe would have given no information of any value to the partners, the commission or the community.

Normally, software is improved by a process of continual evaluation (Formative Evaluation) until the final product is tested by users through a process known as Summative Evaluation (see Figure 2). The goals, the methodologies and the tools used in the two evaluation processes are different.

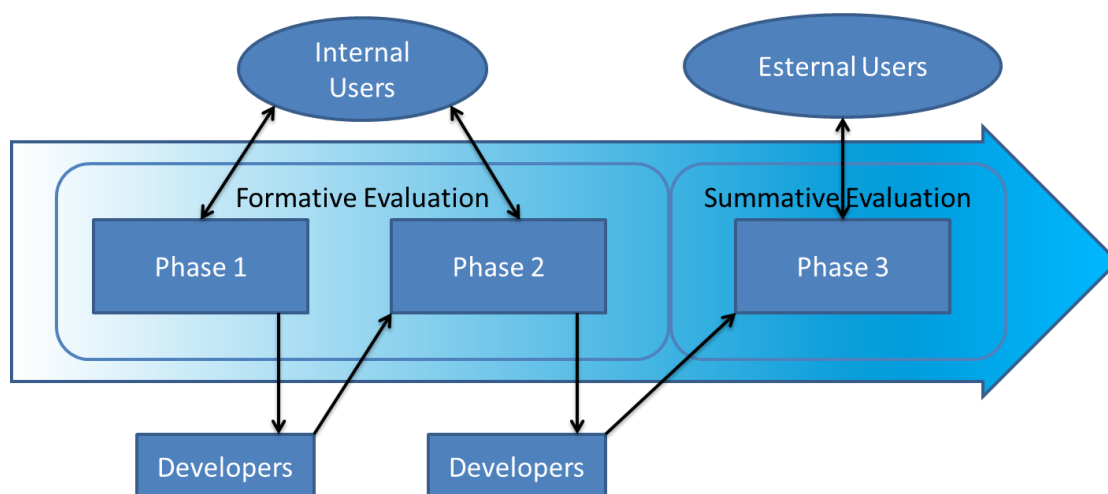


Figure 2 - Evaluation Process

In the CONVERGENCE project the formative evaluation (rounds 1 and 2 of the user trials) involved friendly users, who provided feedback for the improvement of the product (see D8.2 e D8.3). In the third round of trials, by contrast, our goal was to measure user satisfaction in a real-world situation and to answer seven crucial questions:

1. How does the CONVERGENCE framework behave under real-life conditions?
2. If so, are the CONVERGENCE applications tested in the trials robust?
3. To what degree do the prototype applications meet the needs of target groups?
4. What are their most important and satisfying features?
5. Which features require improvement?
6. Could CONVERGENCE and its applications represent a valid alternative to existing systems?

2.2 Evaluation Methodology

2.2.1 Evaluation Tools

To reach the widest potential audience, including potential participants who may initially be ill disposed toward the project, we decided to base our evaluation on a simple online questionnaire. The advantages are well attested in the literature. Online questionnaires:

- Are relatively easy to administer
- Can reach users in many different locations
- Can reach a varied target population
- Can efficiently collect information from large numbers of respondents
- Eliminate many sources of researcher biases.

The critical issue is response rate, which even in the best commercial survey is rarely higher than 2%. Suggestions for improving response rates include the following [1]:

- Brevity (keeping it down to a single page)
- Use of non-monetary incentives such as pens, notebooks; participation in a draw, contest or lottery; discount coupons; promise of contribution to agreed charity
- Preliminary notification
- Guarantees of anonymity.

The literature also suggests other factors that need to be taken into account [2][3]:

- *“Respondents’ motivation, honesty, memory and abilities: respondents may not be motivated to give accurate answers or may be motivated to give answers to present themselves in a favourable light or may not be fully aware of their reasons for any given answer”.*
- *“The Inevitable Self-Selection Bias: people who choose to respond to the survey may be different from those who do not respond, thus predetermining the outcome or giving a false reading / conclusion”.*
- *“Question design: many choices, such as “moderately agree” have different meanings to different people. The question can easily fail to draw out such differences”.*

2.2.1.1 Online questionnaire

On basis of the considerations above, we decided that the questionnaire should be short and anonymous and that each user should compile it just once. These represent a major change compared to our methodology in rounds 1 and 2 which used much longer questionnaires administered several times.

The questions covered features of key importance for any commercial software application (see section 2.1). The language was kept deliberately simple, in order to guarantee that users fully understood the questions.

The final questionnaire consisted of three short sections:

- a) Demographics: Four Multiple Choice questions on:
 - Gender
 - Age
 - Highest Educational Qualification
 - Computer / Internet Proficiency
- b) Assessment: Seven questions divided into three groups, with all responses on a five step Likert Scale:
 - Effectiveness (Interface and management)
 1. Visual Design
 2. Ease of use
 3. Help Functionality
 - Productivity (satisfaction of the needs of the target group)
 4. Usefulness
 5. Performance
 6. Technical Reliability
 - Competitors (Comparison with other similar products already on the market)
 7. Functionalities compared to other products

For each variable, users were asked to express a score for their satisfaction, and a second score expressing the importance they attached with this aspect of the application.

- c) Overall Satisfaction: A single question evaluated on a Likert 5-step scale.

For the implementation of online surveys we used the “LimeSurvey”¹, open source platform. The platform was already localized in several languages. Questionnaires were translated into the six languages used in the trials:

- *ALI – Italian*
- *LMU – English*
- *UTI – Romanian*

¹ www.limesurvey.org

- *WIPRO – Portuguese*
- *FMSH – French and Spanish²*

The LimeSurvey platform was installed on the XIW server. For each questionnaire XIW supplied a direct link to the localized form. Each form was presented automatically to the end user.

The survey platform recorded access data anonymously, and stored users' replies to the questionnaire. In cases where users exited before the end of survey, the system still recorded the users' log. In this way it was possible to determine and exclude false inputs due to automatic web crawlers like Google and Yahoo³. Figure 3 shows the final online questionnaire.

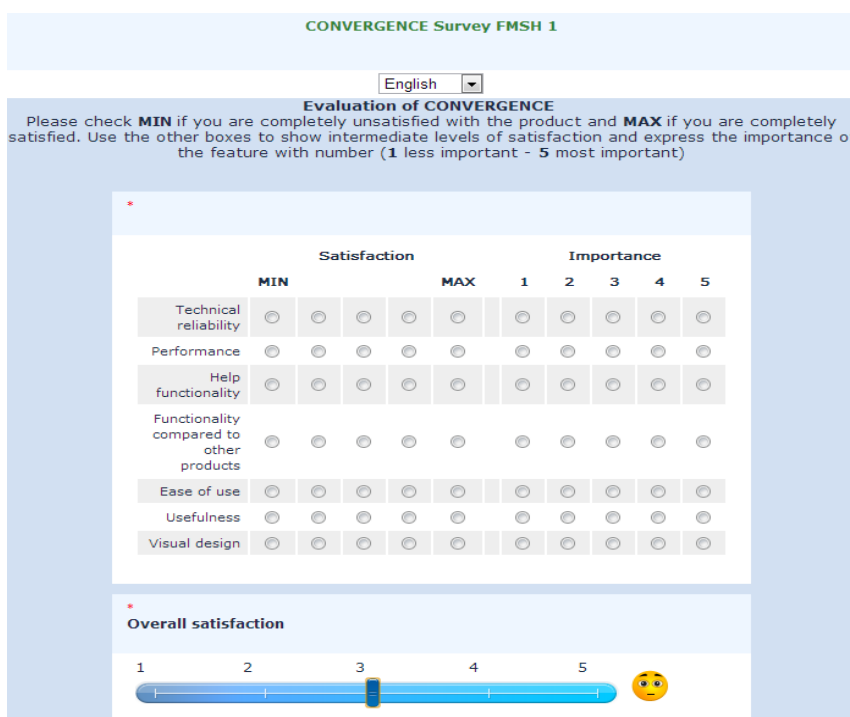


Figure 3 - Survey form

2.3 Recruitment

The partners used a range of tactics to reach potential trial participants. In two cases participants were offered discounts on a product (ALI) or university credits (LMU).

Photos in the Cloud and down to Earth (ALI): ALI involved the non-profit “Associazione Amici del Gioco del Ponte⁴” in its recruitment campaign. This choice made it possible to

² For the Latin-American target population

³ This problem can be avoided by using CAPTCHA verification. We decided not to do this to avoid annoying genuine participants

⁴ “The “Associazione Amici del Gioco del Ponte” was formed in 1970 in Pisa (Italy). It is the oldest civic association tied to the event, and remains active all year promoting the Battle on the Bridge among aficionados,

carry on the trials after ALI entered liquidation at the end of November, 2013⁵. Users were recruited via the ALI mailing list, the mailing list of the Amici del Gioco del Ponte, the association's website, and their Facebook fan page (which reaches about 13,000 people). The application was also promoted through a special eBook⁶ containing a selection of pictures of the Gioco. Participants in the trial who completed the survey, were offered a discount on the book.

Videos in the Cloud and Analysis on Earth (FMSH): FMSH used a range of recruitment methods to create two groups of users, external and internal to FMSH.

FMSH1. Subscribers to the ARA Newsletter⁷ (about 10,000) and the FMSH Newsletter (about 20,000); "friends" on thematic video channels (Archaeology, Languages and Cultures, Latin America, Andean Literature and Culture) and on Facebook (about 5000 names); individuals working in these areas and individuals with a personal interest in the relevant fields⁸.

FMSH2. FMSH staff, working in the field of video analysis & channel broadcasting; video producers; analysts, archivists; heritage content editors (institution/web site/channel owners or heads of communication); direct recruitment of FMSH students in Digital Heritage & Communication and participants in a workshop on "Augmented Audio-visual Archives".

None of the participants in the trial had anything to do with CONVERGENCE or the group running the project.

Podcast creation and Publication (LMU). LMU recruited end users directly, offering students course credits as an incentive. The recruitment campaign created two groups:

LMU1: Lecturers, mostly PhDs, working at the LMU or with similar background. The majority of users work in media informatics groups.

residents, and tourists alike. In the early 1980s the association played an important role in the restoring of the game and today remains actively involved in organizing the most important event of the year for Pisans. The Battle on the Bridge comes alive with two armies dressed in elaborate and shimmering sixteenth century Spanish costumes. It is divided into five phases: the march of the troops along the Lungarni until their arrival in their respective bases, the formal opening of the battle by the Anziano Rettore, the "call to arms" by the troops, the challenges made by ambassadors on horseback, and finally, the battle itself, as the various magistrates compete under the strategic command of their respective leaders" (source www.amicidelgiocodelponte.it).

⁵ See the amendment February 2nd, 2013

⁶ <http://it.blurb.com/b/4054072-il-gioco-del-ponte>

⁷ Audio-visual Research Archives Program (ARA) of ESCoM

⁸ The localization in Spanish is for this group of persons

LMU2: Students of the course “Human Computing Interaction 2” and students already engaged in previous trials. In this case, all the students that tested the application and compiled the survey had as an incentive an educational credit bonus.

In both cases, all users were external to the project.

Smart retailing (UTI): UTI recruited a group of users among employees of the UTI retail and IT sectors as well as normal users of retailing systems.

Smart retailing (WIPRO): WIPRO recruited professionals (Project Engineers, Senior and Junior Retail Consultants) involved in WIPRO activities towards the retail sector.

2.3.1 Scenario Walkthroughs

Participants were given a list of tasks to perform with the application under test (see Table 1) and asked to complete the tasks before filling in the questionnaire. Unlike in the first two trials they did not receive any preliminary information about CONVERGENCE.

Photos in the Cloud and down to Earth (ALI)	
<ol style="list-style-type: none"> 1. Access and register to the application 2. Upload a picture 3. Annotate a picture 4. Select the proper license 5. Search personal and third author's collections navigate 6. Logout 7. Fill the survey 	
Videos in the Cloud and Analysis on Earth (FMSH)	
FMSH1:	FMSH2:
<ol style="list-style-type: none"> 1. Register and login the application 2. Subscribe to channels and browse video channels 3. Browse subscriptions, notifications and selections 4. Try freely the other application features using the help of the online user manual 5. Logout 6. Fill the survey 	<ol style="list-style-type: none"> 1. Register and login the application 2. Upload a video 3. Subscribe to videos 4. Upload an analysis, and subscribe to analyses (optional) 5. Create a channel and post analyses on channels (optional) 6. Subscribe to channels and browse video channels 7. Browse subscriptions, notifications and selections 8. Logout 9. Fill the survey

Podcast creation and Publication (LMU)	
LMU1: <ol style="list-style-type: none"> 1. Register and login to the podcast creator application 2. Upload a video 3. Upload a slide 4. Upload the lection contents 5. Publish a VDI 6. Logout 7. Fill the survey 	LMU2: <ol style="list-style-type: none"> 1. Register and login to the podcast services 2. Perform a search for “MMI” in the Fractal HCI 3. Watch the podcast. 4. Subscribe to the podcast. 5. Logout 6. Fill the survey
Smart retailing (UTI)	Smart retailing (WIPRO) ⁹
<ol style="list-style-type: none"> 1. Login in the retailer section 2. Navigate through products 3. Add an offer 4. Visualize the offers, the matches and the subscriptions 5. Visualize statistics about what the clients subscribed for 6. Logout from the retailer section 7. Register to the application as client 8. Navigate through products 9. Add a subscription 10. Visualize the subscriptions, the matches and the offers 11. Logout from the client section 12. Fill the survey 	<ol style="list-style-type: none"> 1. Access, registration and login to the manufacturer section 2. Create product 3. Browse created products 4. Publish product 5. Logout 6. Access, registration and login to the retailer section 7. Create subscription 8. Browse subscriptions 9. Browse subscription matches 10. Logout 11. Access, registration and login to the consumer section 12. Create subscription 13. Browse subscriptions 14. Browse subscription matches 15. Fill the survey

Table 1 - CONVERGENCE scenario walkthroughs

2.3.2 Data collection and data analysis

At the end of trials, all the collected data from the LIME SURVEY was inspected and false inputs excluded. The cleaned data were exported to Excel and imported into Matlab for analysis. The data analysis consisted of five steps.

The first step consisted of sample evaluation: the characterization of the sample populations.

In the second step we examined if the demographic characteristics of the sample populations (sex, age, computer proficiency and education) influenced the evaluations. To achieve this, we compared the results from subsets of users with specific characteristics against results from the whole group and displayed the results in radar plots, (see section 2.5.1).

The third step in the analysis focused on users’ level of satisfaction with specific features of the application and the importance they attached to these features [2]. The results are shown in Satisfaction vs. Importance Plots (below) and Box Plots.

⁹ Users played the triple role of manufacturer, retailer and consumer.

In Satisfaction vs. Importance Plots plot the x axis, shows user satisfaction and the Y axis shows importance, as seen in Figure 4.

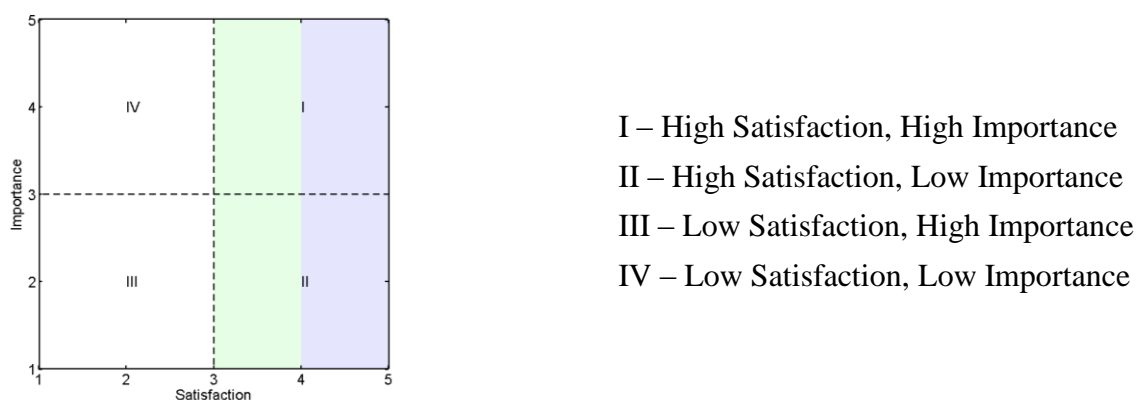


Figure 4– Example of Satisfaction vs. Importance plot

Box Plots¹⁰ (see Figure 5) shows the dispersion of scores around the median (the variability in individual responses). In all our results the median and the mean values for the distribution are very close (see section 2.5.2).

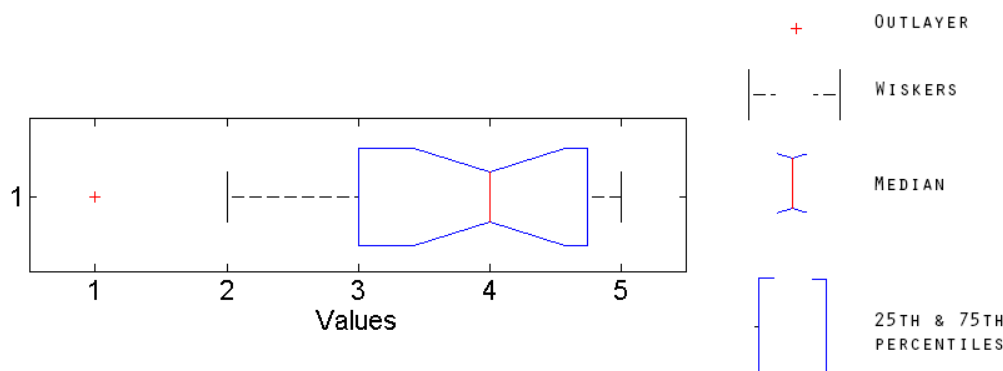


Figure 5– Example of Box Plot

The fourth step consisted of the analysis of logs generated by the applications. Each application server recorded a log containing records of individual user log-ins, use time and any systems errors that occurred. This data allowed us to create a table summarizing system uptime, the total usage and the number of log-ins for each application (see section 2.5.3).

¹⁰ On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. Points are drawn as outliers if they are larger than $q3 + w(q3 - q1)$ or smaller than $q1 - w(q3 - q1)$, where $q1$ and $q3$ are the 25th and 75th percentiles, respectively. The default of 1.5 corresponds to approximately $\pm 2.7\sigma$ and 99.3 coverage if the data are normally distributed. The plotted whisker extends to the adjacent value, which is the most extreme data value that is not an outlier (<http://www.mathworks.it/it/help/stats/boxplot.html>). Maximum whisker length is w . In our representation we set $w=1.5$.

The fifth and final step was to calculate mean user overall satisfaction with the CONVERGENCE applications, collectively and individually (see section 2.5.4).

We concluded our analysis by examining a number of conditions which may have influenced the results, positively and negatively (section 2.5.5).

2.4 Demographic distribution

During the trials **563** users used the applications: **220** (39%) also compiled the survey (see Table 2). The majority were male (62.7%), in the age range 21-40 (70.4%), had a University education (93.0%) and saw themselves as high skilled uses of computer and the Internet (75.4%).

Variable	Class	Users Groups									TOT
		ALI	FMSH1	FMSH2	FMSH tot	LMU1	LMU2	LMU tot	UTI	WIPRO	
Gender	Male	9	15	7	22	20	44	64	26	17	138
	Female	11	4	16	20	6	21	27	18	6	82
Age	< 21	2	1	0	1	0	0	0	1	0	4
	21 – 30	4	2	9	11	19	64	83	29	14	141
	31 -40	6	11	8	19	7	1	8	12	9	54
	41 -50	4	1	1	2	0	0	0	0	0	6
	> 50	4	4	5	9	0	0	0	2	0	15
Highest Educational Qualification	High School	8	0	0	0	0	7	7	0	0	15
	University (1 st degree)	4	8	2	10	6	53	59	16	12	101
	University (adv. degree)	8	11	21	32	20	5	25	28	11	104
Computer / Internet Proficiency	Beginner	1	0	0	0	0	0	0	2	0	3
	Intermediate	12	3	15	18	1	10	11	10	0	51
	Advanced	7	16	8	24	25	55	80	32	23	166
Surveys	Compiled surveys	20	19	23	42	26	65	91	44	23	220
	Number of access	106	226	36	262	39	87	126	46	69	563
	Ratio (%)	18	8	63	16	66	74	72	95	100 ¹¹	39

Table 2 – Demographic distribution

Response rates were best for users recruited directly rather than through mailing lists or Facebook groups. In this way, WIPRO, UTI, FMSH2 and LMU achieved response rates >63%. The LMU students (LMU2), who received course credits had a much higher response rate than the lecturers (LMU1), who received no credits.

¹¹ WIPRO users accessed three times for each role and compiled the survey only once.

Response rates for the other groups (recruited via Facebook or mailing list) were less than 0.1%. This is not uncommon for this kind of recruiting. In mass mailing, 2% is considered a good redemption rate. Figure 6 shows redemption rates for the ALI and FMSH1 recruitment campaigns (see figure below).

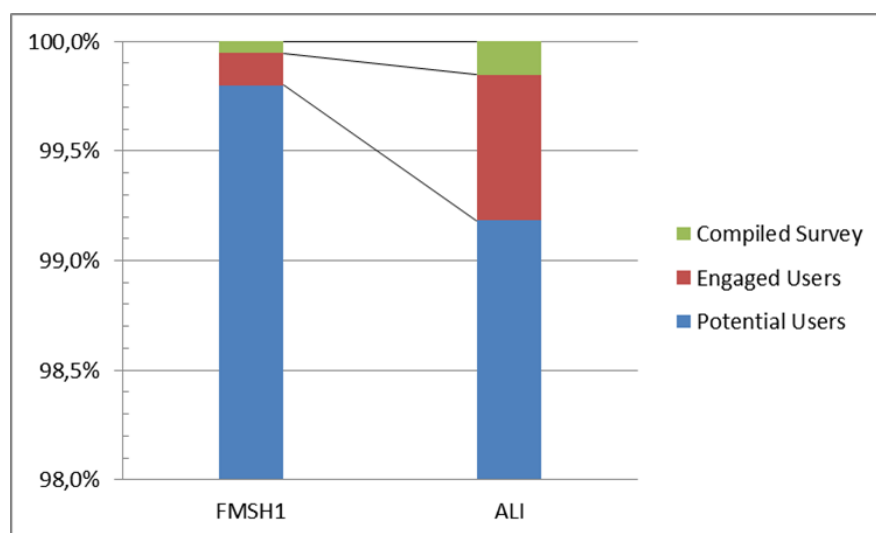


Figure 6– ALI and FMSH1 recruitment campaigns

It is significant that the ALI campaign (15000 potential users) that offered a reward (see section 2.3) obtained a better redemption rate than the FMSH1 campaign (35000).

We observe that the total number of responses for individual campaigns was in several cases low. This obviously limits the inferences we can draw from the results.

2.5 Results of Analysis

2.5.1 Impact of population classes vs. features

To test whether the specific features of the sample population affected results we applied the methodology described in section 2.3.2. For each demographic variable (age, gender etc.) for which we had data we divided the population into classes, plotted the mean result for each class (except empty classes) for each item on our questionnaire and compared the plot against the equivalent plot for the whole population. Figure 7 shows a sample result in which the variable considered was user computer skills. As is clear from inspection there is no significant difference between the results for users with different levels of skills. Our analysis showed no significant differences for any scenario or for any variable.

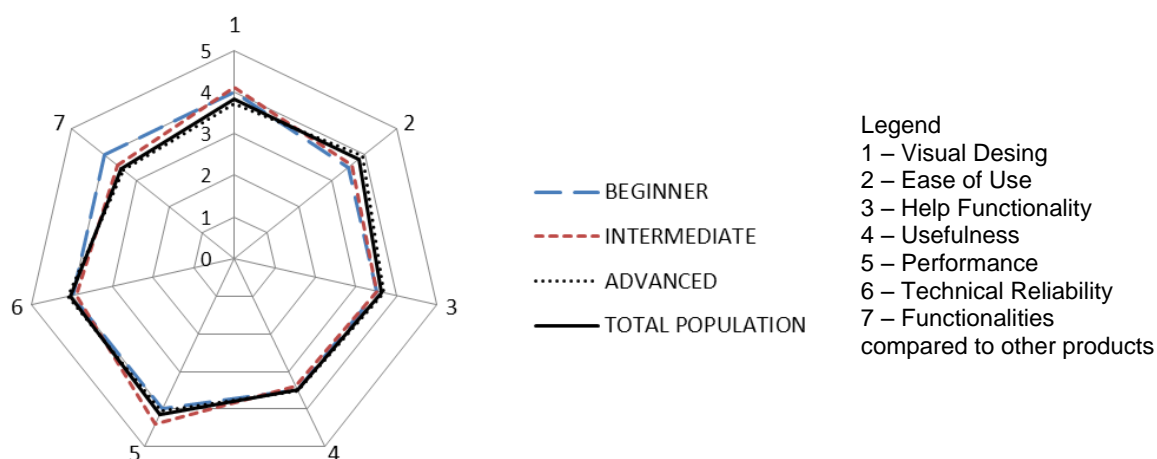


Figure 7 – Impact of population class (Computer / Internet Proficiency) vs. features for satisfaction – UTI scenario

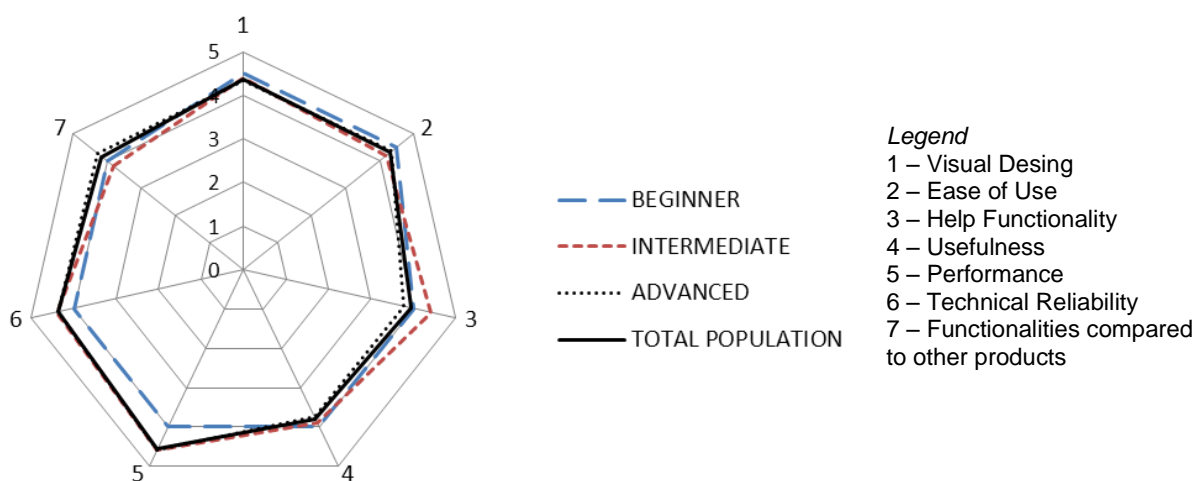


Figure 8 – Impact of population class (Computer / Internet Proficiency) vs. features for importance – UTI scenario

2.5.2 User satisfaction

1) Photos in the Cloud and down to Earth (ALI)

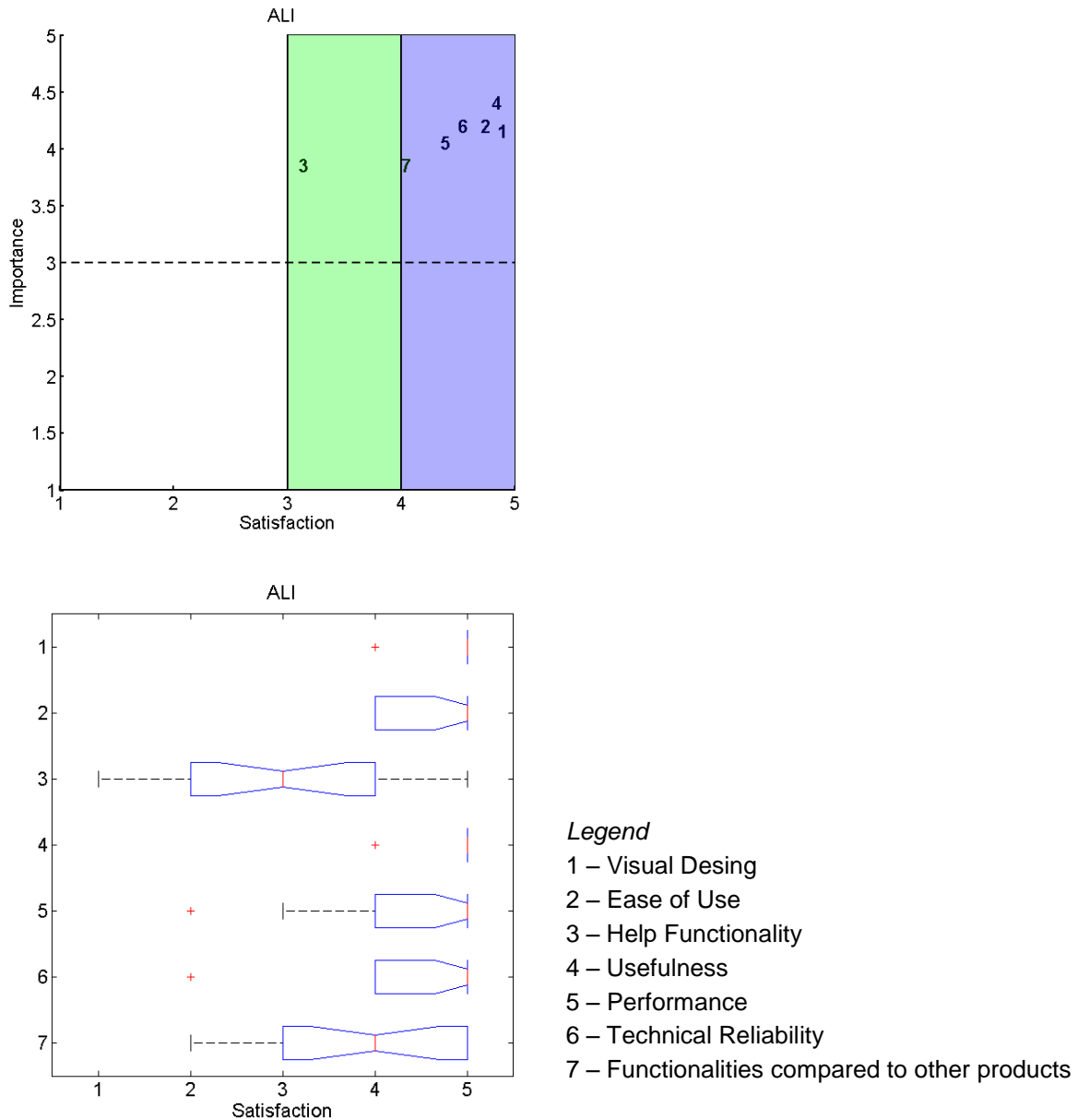


Figure 9 – Satisfaction and Importance perceived by users in the considered scenario

The analysis shows high customer satisfaction values (scores between 4 and 5) for most of the features, especially for technical aspects of the application. The only exception is the Help Function (score around 3). However, users do not attach a great deal of importance to this feature.

The second box plot, shown in Figure 9 shows that there is relatively little variation in user responses. This is especially true for Visual Design and Usefulness.

Compared to results from the previous trials, the user response was much more positive. All told, the application appears to work well from various points of view and is robust and functional. This result is confirmed by the overall satisfaction analysis (see paragraphs 2.5.4).

2) Videos in the Cloud and Analysis on Earth (FMSH)

As described in section 2.3, FMSH recruited two groups of users. We began our data analysis by analysing the two groups separately and comparing the results (see Figure 10), which proved to be very similar. We were thus able to merge the data for the two groups without losing important information.

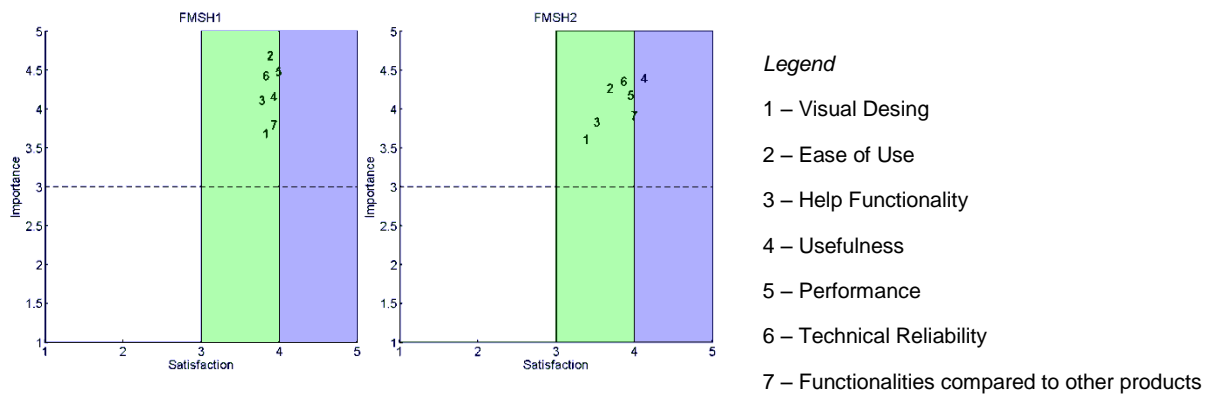


Figure 10 – Satisfaction and importance perceived by the two groups

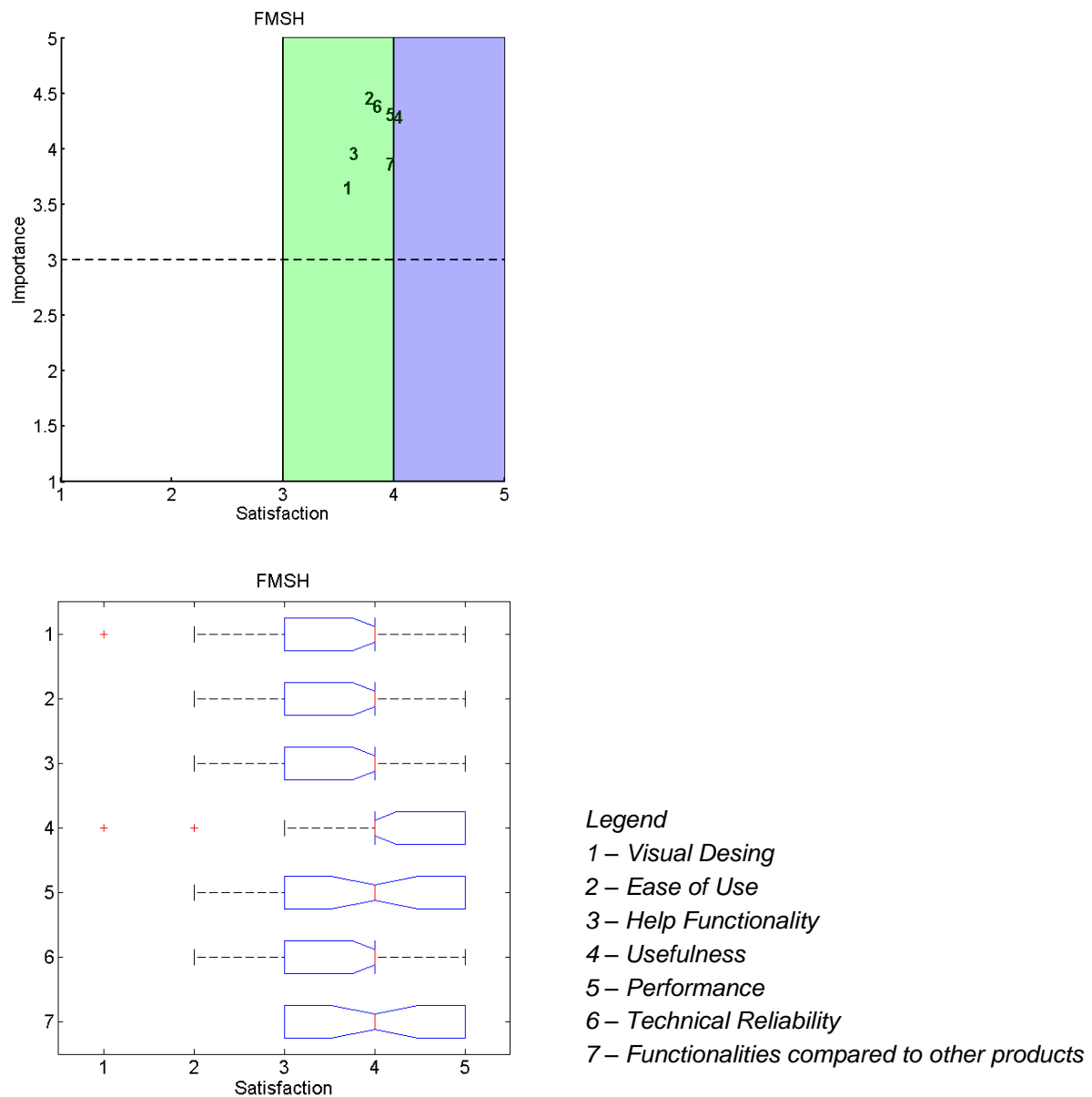


Figure 11 – Satisfaction and Importance perceived by users in scenario

The analysis shows user satisfaction scores (scores around 4). The lowest scores are for Visual Design and Help Functionality (score around 3.5). These features however are considered less important than the others.

Compared to the results of the previous trial, the third trial gave higher scores for Technical Reliability, Help and Ease of use.

The verdict on Usefulness was generally satisfactory (score: 4.0). Users considered this to be extremely important (score: 4.4).

Of special interest is the final verdict comparing other systems. The average vote is good (3.93), if not evenly distributed. This would indicate that users regard the tool as competitive in the market.

In summary, users consider the application to be usable, robust and competitive. The Visual Design and the Help Functionality should be considered as a priority for future improvement.

3) Augmented Lecture Podcast (LMU)

LMU recruited two groups of users (see section 2.3). As with FMSH, we began by analysing the two groups separately and comparing the results. Given the very similar results (see Figure 12) we were again able to merge the two datasets.

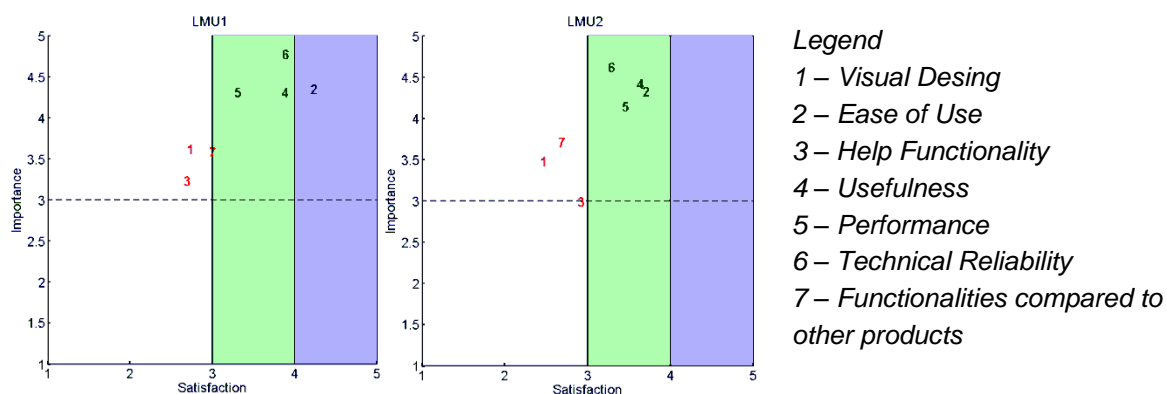


Figure 12 - Satisfaction and importance perceived by the two groups

This was the application that received the most criticism from participants. The users gave the applications scores between 3 and 4 for Usefulness, Performance and Technical Reliability. The other features were considered to be unsatisfactory. However, users also considered them to be relatively unimportant. These features will need to be improved for any future commercial applications.

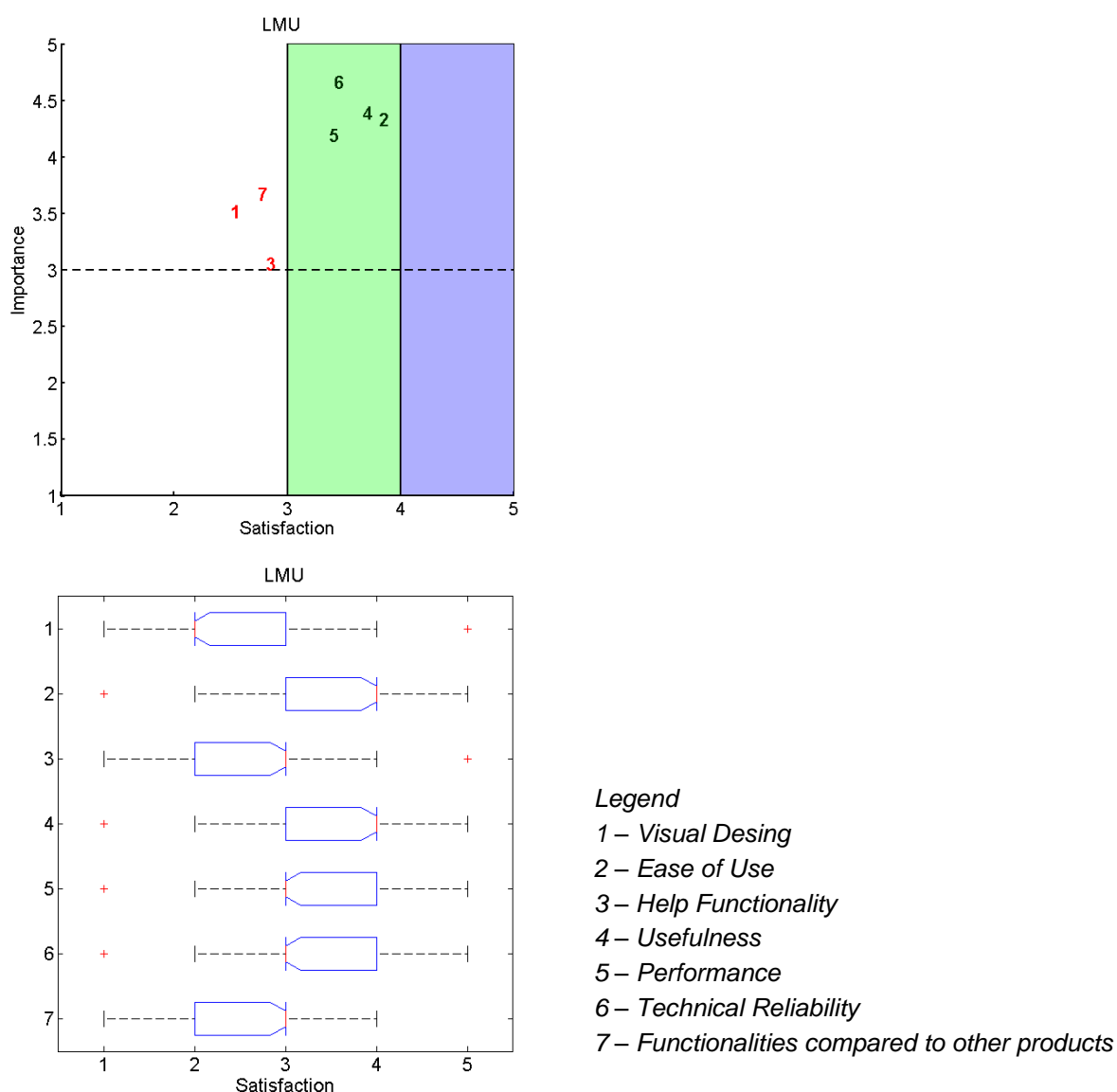


Figure 13 – Satisfaction and Importance perceived by users in the considered scenario

Comparing results from this and the previous round of trials, the application is still considered easy to use; it also receives a slightly higher score for its technical features. However, the user interface is still considered poor, and the score for the Help Function is no higher than previously. These features should have a higher priority in future development, with more attention to the kind of features provided by typical commercial products.

4) Smart Retailing (UTI)

The analysis shows good customer satisfaction (scores around 4) for most of the features.

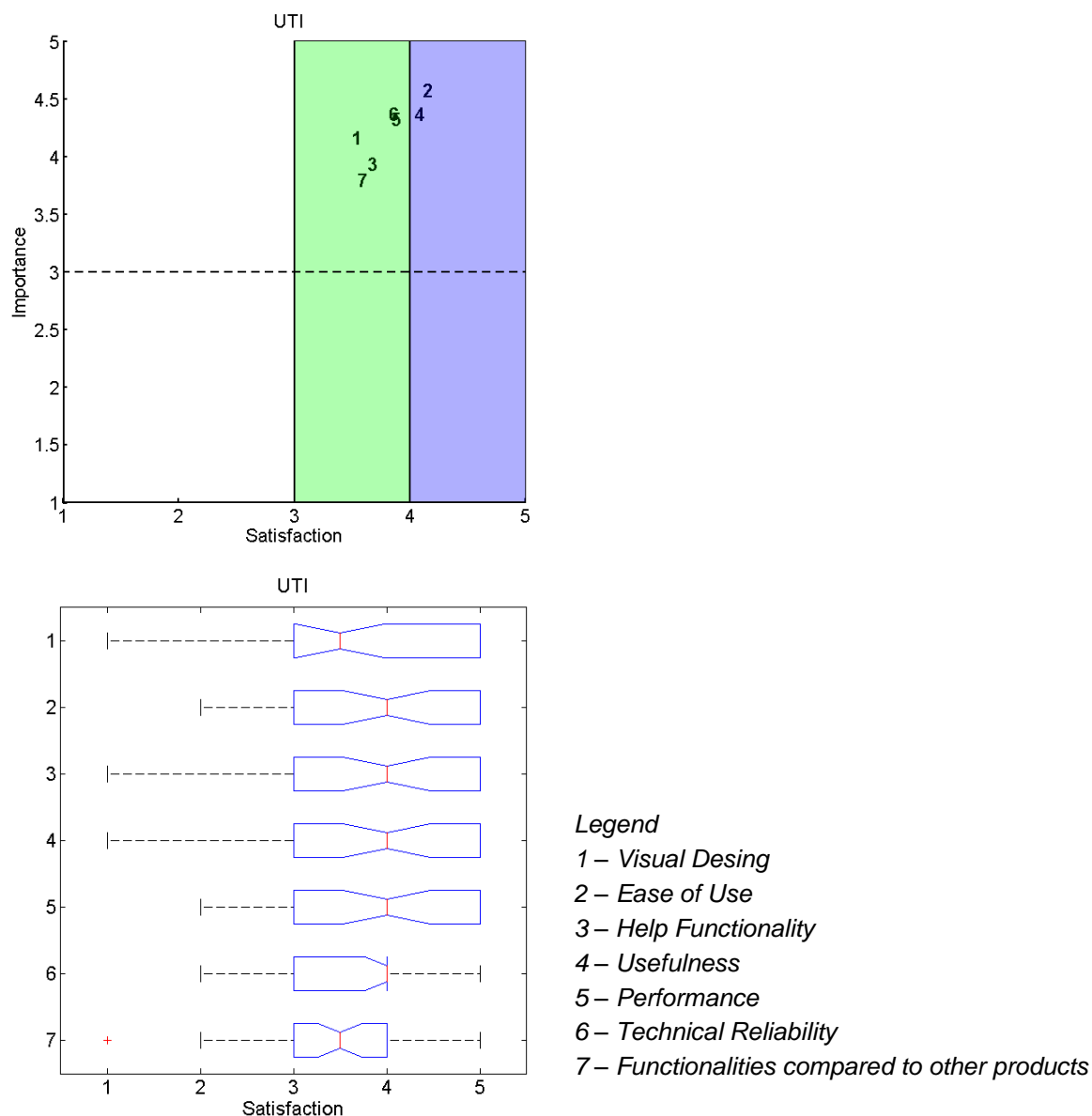


Figure 14 – Satisfaction and Importance perceived by users in the considered scenario

The highest scores are for Ease of Use and Usefulness (scores >4), and for Performance and Technical Reliability (scores around 4). The lowest scores are for Visual Design, Help Functionalities, and ability to compete (scores around 3.5). However, users do not consider these aspects of the application to be particularly important.

All the results are better than those obtained in the previous round of trials. The “friendly” users in round 2 considered the application too complex to use, due to poor design and a lack of Help Functionality. In this trial, users were more satisfied, considering the application to be usable and robust. Again there is a need for more attention to Visual Design and Help Functionality.

5) Smart Retailing (WIPRO)

The analysis shows encouraging customer satisfaction values (scores around 4) for most of the features. The lowest scores are for the Help Functions and ability to compete (scores around 3.5). As in the other scenarios, however, users did not consider these features to be particularly important. The results are similar to those obtained in the previous round. However, we note a small improvement in the scores for Visual Design, Performance and Help Functionality, and in overall user satisfaction (see section 2.5.4).

All told, the application works well from various points of view and is robust and functional.

The lowest score was for Help Functionalities. The feature that users considered as most important was Performance. These should be priorities for future development.

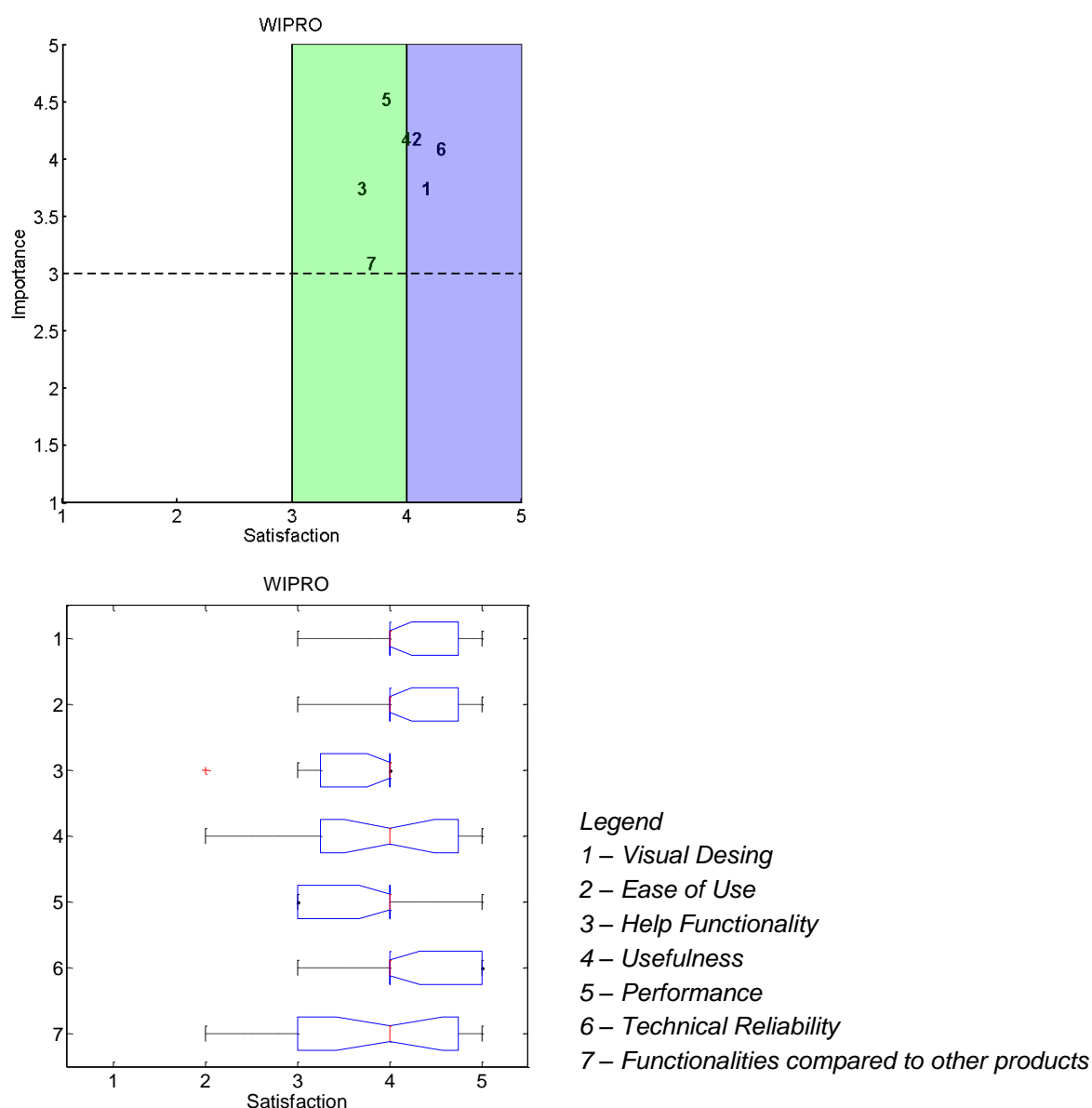


Figure 15 – Satisfaction and Importance perceived by users in scenario

2.5.3 Application performance

The table below shows, for each application: i) uptime, ii) the total number of log-ins, iii) total usage (in minutes), and iv) the average time taken by users to complete their tasks.

The partners reported no system failures. All the application servers ran continuously, with the exception of the LMU application, which encountered a regularly occurring blocking issue twice a week. To avoid this issue, LMU decided to schedule a server restart every hour. This produced two minutes of downtime every hour. The result was an uptime of 96.7%.

<i>Application</i>	<i>Application Online/Offline (hours)</i>	<i>Application Uptime (%)</i>	<i>Date of trials (start-end)</i>	<i>Total number of access</i>	<i>Total usage time (minutes)</i>	<i>Average time for completing tasks (minute)</i>
Photos in the Cloud and down to Earth (ALI) ¹²	1512 / 0	100	17 Jan 2013 19 Mar 2013	106	584	5,5
Videos in the Cloud and Analysis on Earth (FMSH)	1200 / 0	100	13 Dec 2012 1 Feb 2013	262	2494	9,5
Augmented Lecture Podcast (LMU) ¹³	360 / 12	96,7	16 Jan 2013 31 Jan 2013	126	489	3,9
Smart Retailing (UTI)	360 / 0	100	15 Jan 2013 29 Jan 2013	46	338	7,3
Smart Retailing (WIPRO)	48 / 0	100	9 Jan 2013 11 Jan 2013	69	903	13,1

Table 3 - Applications performance data

In summary, the applications proved themselves to be robust and reliable.

2.5.4 Overall Satisfaction

In this last analysis, we analysed users' overall satisfaction and compared it to their satisfaction in the previous trial (Track 1 Phase 2).

Figure 15 shows the scores for each scenario: the blue columns represent the results of previous trial; the red columns represent the scores obtained in the current trial.

¹² Even if the data collection is frozen to March 19th, the application is still running and will work for the rest of 2013.

¹³ The application server was restarted every hour.

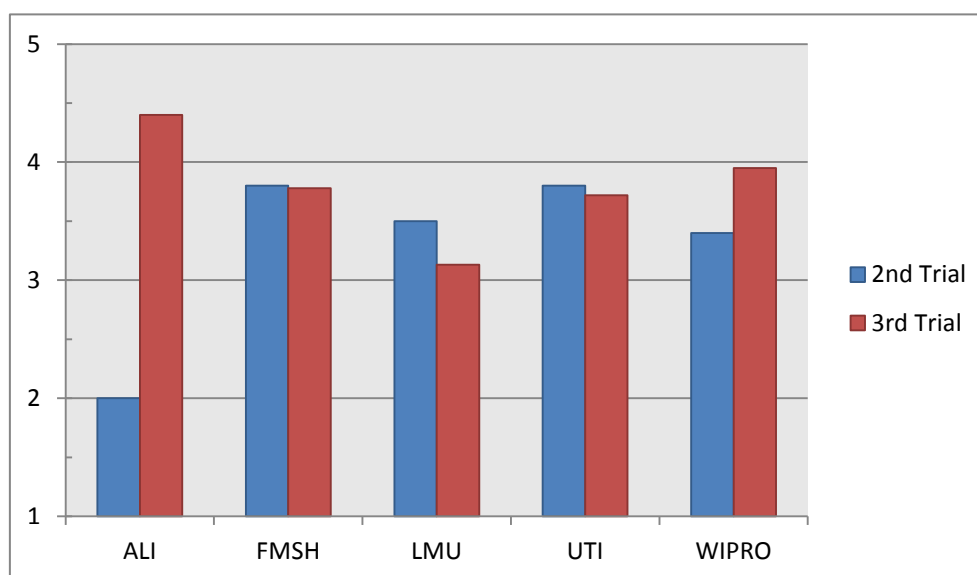


Figure 16 – Overall Satisfaction for each scenario

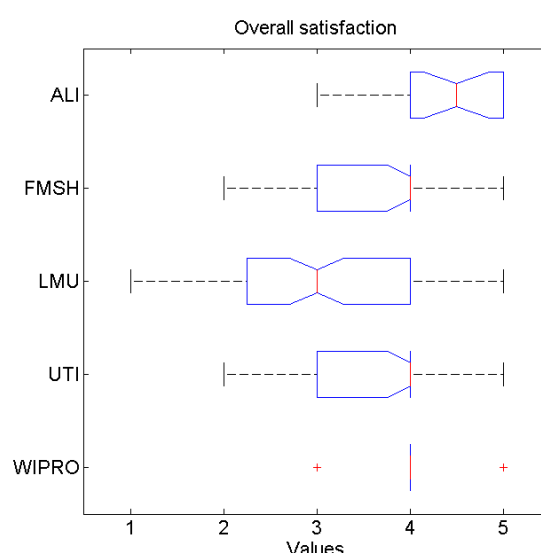


Figure 17 –Distribution of overall satisfaction

Given the characteristics of the trial (external users, indirect recruitment, see 2.3) we expected overall satisfaction to be lower than in the previous trial. In reality the scores were relatively similar, with two important exceptions. The users of the ALI expressed more than twice the level of satisfaction they reported in the previous trial, with relatively little inter-individual variation. The LMU group, on the other hand, shows a small *decrease* in satisfaction, with large variations between respondents. This result is important for our general analysis of the trial results (see section below).

2.5.5 General considerations

In interpreting the data from our study, it is necessary to take account of the design of the trials and the conditions in which they took place:

1. The applications tested in the trials had very different designs; trial participants used them for very different purposes (see D2.2).
2. In the third round of the trials, unlike the previous rounds, participants were not given any specific information about CONVERGENCE or the aims of the project.
3. Recruitment strategies deliberately targeted the kind of users who would be likely to use the applications “in real life”.
4. Some sub-groups of users were too small to draw reliable conclusions about the behaviour of specific target populations.

On the basis of these considerations, we offer a preliminary interpretation of our results – which although not demonstrable in rigorous statistical terms, can be a source of useful ideas for exploitation planning and for future studies.

1. Target populations who wanted to use the applications for work were much more demanding and critical than those who used it in their leisure.

All participants in the trials were positive about the applications' technical characteristics. However, their evaluations of the applications' visual design, ease of use and help functions were variable. Although participants gave the applications better scores for these features than in previous trials, overall scores were still less than satisfactory. The two most interesting results were those at the two extremes.

On the positive side, users in the ALI trial gave the ALI application an extremely high score both for visual design and for ease of use – a major improvement with respect to the scores it received in the previous round of trials.

On the negative side, the visual design and ease of use scores for the lecture podcast application (LMU) were no better than and sometimes worse than the scores it received in round two. There are two possible explanations:

- a) ALI had much improved its application while LMU had not
- b) The scores given to the applications were strongly influenced by the characteristics of the user population. The participants in the ALI trial spanned a broad range of different age groups and professions: their only point in common was their common participation in the “Gioco del Ponte”. They used the application above all for their own enjoyment. This could explain why the group was less critical towards the user interface and the visual layout than the LMU group. In this trial all the users came from roughly from the same well-defined age groups and backgrounds. Their reasons for using the application were also very well defined. Lecturers wanted to use it to transmit knowledge to students; students wanted to use it to learn. This made them very sensitive to any aspect of the product, which made it easier or harder for them to reach their objectives.

2. The better the application matched well-defined user needs, the less it suffered from comparisons with competing products.

When trial participants compared the CONVERGENCE applications with competing products, they were reasonably satisfied. However, they also said that they did not consider this kind of comparison to be particularly important (see section 2.5.2). Cross-tabulating users' general level of satisfaction against the importance they attached to the comparison, we see that the more satisfied they were, the less concerned they were by the comparison. Consider, for example the two groups with the highest and the lowest levels of “general satisfaction”. The ALI group, which was very satisfied (score 4.40) gave very little importance to comparisons with competing products. By contrast the least satisfied users – those from LMU (score: 3.13), gave much more importance to this factor. Users from UTI and WIPRO show a similar trend – though the effect is not so marked.

3. User evaluations are strongly affected by expectations

Users in the last round of trials were generally more positive and less critical than those in the previous round. This may be due to the way the trial was prepared. Although the third round of trials used different recruitment strategies and different questionnaires than the second, the target populations and the variables studied were basically the same. The only substantial difference was in the way CONVERGENCE was presented to participants. In the second round, participants received exhaustive information about the CONVERGENCE project; in the third they were told nothing at all. The results of the focus groups in the second round suggest that the information given them created expectations that the applications were unable to satisfy. This could have had a negative effect on user perceptions.

2.6 Conclusions

The main goal of this last phase of the trials was to test the functionality and reliability of CONVERGENCE and to see whether real-life users would perceive these characteristics when working with CONVERGENCE applications. A secondary goal was to test the applications themselves, and to identify features that would need to be improved in a future commercial product.

Our results show that although some aspects of the system need improvement, **CONVERGENCE's response to the needs expressed by trial participants is better than satisfactory.**

1. **The prototypes satisfied participants' requirements:** users assigned the applications a mean overall satisfaction score of more than 3.5. The applications received good scores for all items in the questionnaire. The items with the highest scores (Technical Reliability and Ease of Use) were also the aspects of the applications that users considered most important.
2. **In terms of functionality and technical performance, participants in the third round of trials gave the applications a much higher score than they received in the second round.** The main focus of the study was on the applications' technical characteristics which the trial measured in terms of Usefulness, Performance and Technical Reliability. These were not only the characteristics which received the highest scores from participants (mean scores around 4) but also the characteristics to which users attached the greatest importance (mean scores around 4 again). This result is evidence for the solidity and "robustness" of the CONVERGENCE framework: a theme we will return to in the following chapter. Positive user reports are confirmed by the objective data on application performance (section 2.5.3): system logs show very few failures, even when the applications were used continuously by a significant number of concurrent users. This is a

major improvement with respect to the frequent crashes experienced during the second round of trials.

3. **Most of the target groups found that the applications were easy to use.** They also considered this to be an extremely important characteristic of the application. This represents a major improvement with respect to previous trials.
4. **The visual design and the help functions still need improvement.** These aspects of the applications received lower scores than the others and were considered by some users (e.g., those at LMU) as less than satisfactory. This result confirms user feedback from the previous trials. We note, however, that users did not consider these to be the most important aspects of the application.
5. **Industrialized versions of the applications have the potential to compete with similar products from other producers.** The majority of participants believe that the CONVERGENCE applications can compete with similar products from other suppliers. In this respect they consider CONVERGENCE to be relatively satisfactory (score 3).

Considering these results together, we conclude that, with improvements in visual design and graphics, the CONVERGENCE applications could represent a valid alternative to the tools the target populations uses currently. This conclusion matches findings from earlier trials showing that users had a strong interest in the services CONVERGENCE was offering and that with some changes to their design and technical characteristics the CONVERGENCE applications could offer a valid solution to users' requirements.

3 Track 2 Phase 2 - Network experiments and simulations

3.1 Network final test-bed

In line with the plan defined in D8.1, track 2 of the CONVERGENCE trials involved two consecutive phases of testing. D8.3 described phase 1. In this report, we describe phase 2.

In both phases, the goal was to demonstrate the feasibility of running ICN over current network technology and to show the advantages of ICN with respect to the current Internet architecture. Today, the web is mainly used for distributing and consuming content. There is thus a strong need for network architectures offering better handling of massive content distribution. Obvious solutions include Content Distribution Networks (CDNs) and Peer-To-Peer (P2P) networks. However, both have their own scalability and cost-efficiency issues.

So the question is: “is it possible to design a network architecture that supports massive content distribution while simultaneously limiting network overhead, maintaining scalability, and simplifying the effort to deploy data dissemination services?” In what follows, we will show in a practical use-case that the CONVERGENCE network (CoNet) has the potential to fulfil this need.

Service providers predict that global Internet traffic in coming years will be driven by video streaming. We therefore decided to test CoNet with a video-streaming scenario. In our scenario, CoNet caches video content in the network and easily support permanent content replication, while ensuring that users always fetch content from the closest network point where the content is available. Consequently, users perceive low latency while effective load balancing benefits the whole network.

The trial taught us that one of CoNet’s main strengths is its ability to simplify very complex networking tasks that are undeniably necessary to obtain an effective data dissemination service. Compared to current IP-based solutions, CoNet does not need a lot of configuration to be deployed in a safe and stable way; building and maintaining the routing-plane is relatively straight forward; there is no need to create a dedicated infrastructure. Most of CoNet’s most powerful functionality is integrated natively at the network layer where applications can access it via a simple unified API. Caching, replication, redirection, routing- and fetching- by-name are all provided “out-of-the-box” by the network. In this way, CoNet avoids the need to deploy and orchestrate a *patchwork* of different technologies: content-based routing protocols, overlays infrastructure, security frameworks, etc.

3.1.1 Test-bed description

The test used the infrastructure provided by PlanetLab Europe (<http://www.planet-lab.eu/>), a global facility for the deployment and test of experimental network services for the Future Internet. The test used the network shown in the Figure 18.

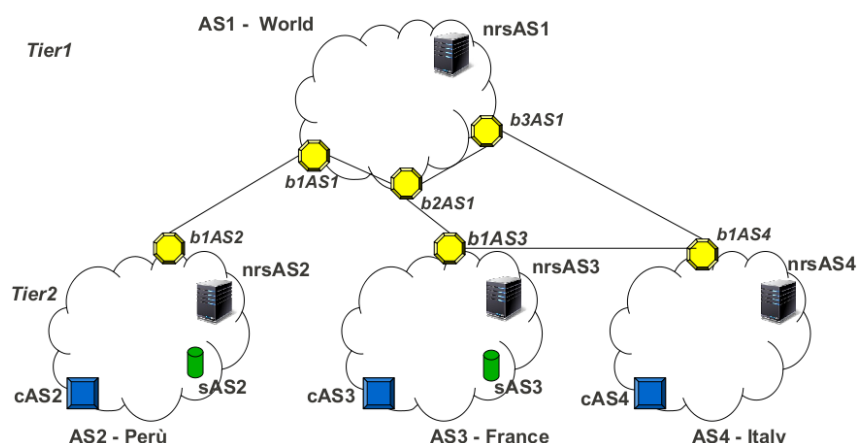


Figure 18- CoNet scenario of the test-bed

As stated in D.5.3, CoNet is an (inter-)network layer that provides users with network access to remote named-resources. CoNet interconnects CoNet SubSystems (CSSs). A CSS could be a layer-2 network, i.e. Ethernet, or a layer-3 network, IPv4/IPv6 network, or a whole IP Autonomous System. CSSs can be defined rather freely; in our scenario CSSs coincide with Autonomous Systems. CSS contain CoNet nodes that can be logically classified as end-nodes (ENs), serving-nodes (SNs), border-nodes (BNs), internal-nodes or name-routing-system nodes (NRSs). End-nodes are content consumers that request named-data by mean of Interest messages. Serving-nodes store and provide named-data, and advertise prefix-names to form the routing-plane. Border-nodes, located at the border between CSSs, forward packets using the CoNet routing protocol and may in some cases cache them. CSSs use NRS nodes to support name resolution for the CoNet routing-by-name process within a CSS, i.e. when a CoNet node does not have a routing entry to forward an Interest message, the node lookup the entry in the NRS, which hold a centralized Routing Information Base (RIB). To support inter-domain routing, NRSs of different ASs advertises to other NRS the name-prefixes of the content stored within.

Our test-bed scenario (Figure 18) is an *evolutionary* deployment of CoNet. The network consists of four Autonomous Systems, one Tier1 AS, labelled AS1 World, and three Tier2 ASes: AS2 Peru, AS3 France and AS4 Italy. Each Autonomous System contained one Point of Presence (POP), none or one serving-nodes, one or more Border Gateway nodes, and one NRS node.

Customers (not shown in the figure) access to CoNet via the Point Of Presence (cAS_x) using a traditional HTTP Web Access. Indeed, a CoNet POP is CoNet end-node and Web server, thus it acts as a proxy between HTTP and CoNet protocols. For instance, an HTTP GET operation will be relayed as a CoNet GET and the received content will be sent back through the HTTP socket. To assess the effectiveness of CoNet in supporting the CONVERGENCE application framework (i.e. middleware + application), we deploy in the test-bed scenario the FMSH application, as it specifically deals with video streaming. Accordingly, each cAS_x includes a

full CONVERGENCE stack, FMSH application + CoMid + CoNet, and provides web access for external users. Users interacted with cASx through normal web browsers. In this way the presence of CoNet ICN was completely transparent to users, in the spirit of an evolutionary approach.

Serving-nodes (sASx) are content repositories for content distribution providers and also offer synchronization mechanisms to replicate content to remote mirror repositories. The trial shows that it is easy for content distribution providers to manage the Serving-nodes they use to distribute content. Accordingly, we created a simple administration panel allowing providers to monitor the bandwidth consumption on their nodes and to perform content replication.

Border Gateway nodes (bnASx) perform inter-routing between different autonomous systems.

Point-of-Presence, Border Gateway and Serving node provides in-network caching functionality with a FIFO replacement policy. Therefore they cache any forwarded named-data.

As described in D.5.3 (section 4.2.4.2), NRS nodes manage the “routing-plane” for the Lookup-and-Cache architecture [23] in a centralized way, similar to what occurs in a Software Defined Network [24][25]. For what concerns the inter-domain routing, NRS nodes exchange among each other name-prefixes of the named-data (i.e. content) hosted in serving-nodes of their Autonomous System, and this exchange enable to setup their Routing Information Bases (RIBs). For what concerns the intra-domain routing, the RIB is used as centralized name-based routing table computed by the NRS for the Autonomous System. It is formed by so called *ICN routes*, and it is looked up by internal nodes to temporary install the ICN routes in their Forwarding Information Base.

An ICN route has the format $\langle \text{prefix-name}, \text{next_hop}(i) \rangle$, where prefix-name is the first part of the content-name (e.g. cnn.com) and next_hop(i) is the IP address:port identifying the next CoNet entry of the path with respect to the i -th node of the AS. Therefore, in the test-bed CoNet is actually deployed as an IP overlay network connecting CoNet nodes.

NRS nodes have inter-domain relationships that influence the way the routing-plane is constructed and maintained. Following the actual BGP strategy, we envisage two different kinds of relationships between NRS nodes: “customer-provider” relationships and “peer-peer” relationships. In a “customer-provider” relationship the NRS customer advertises the prefix-names it can serve to the NRS provider, that is, the NRS customer shares its RIB to the NRS provider. So, each time a Serving-node advertises a new prefix-name to the NRS of its Autonomous System, the NRS propagates the ICN route to its provider NRS. However, the NRS provider does not share routing information with its customers. Instead two NRS peers can create a “peer-peer” relationship in which they share all the ICN routes they have in their RIBs. Every time an NRS node receives an ICN route, it stores the record in its RIB and propagates the information to all its NRS peers. The figure below shows the NRS inter-domain relationships within our test-bed.

Every NRS node for a Tier2 AS has a “customer-provider” relationship with the NRS node for a Tier1 AS. nrsAS3 and nrsAS4 also have a “peer-peer” relationship.

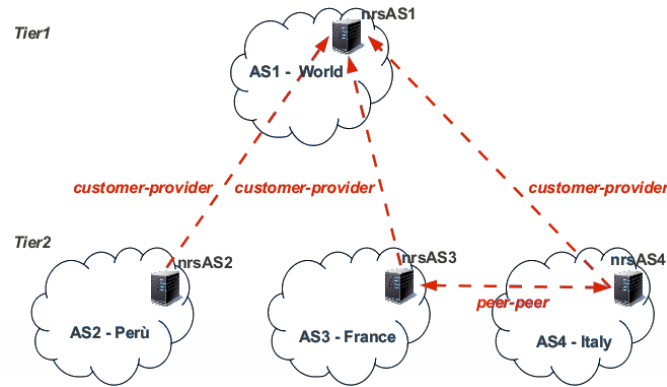


Figure 19 – Inter-domain routing plane relationships of the test-bed

3.1.2 Software setup

All test-bed CoNet devices were PlanetLab nodes running GNU/Linux OS (Fedora 12 release). Apart from NRS, all other nodes ran a CONET implementation based on a modified version of the CCNx protocol [www.ccnx.org] 0.5.0 implementing our own Lookup-and-Cache routing approach. NRS nodes ran a java implementation of our routing scheme.

cASx nodes (PoP) are the only CoNet nodes running the complete CONVERGENCE stack (application + CoMid + CoNet). These nodes offer to CONVERGENCE customers an attractive web application they could use to publish and subscribe to video content. The nodes ran a slightly modified version of the FMSH application used in phase three of the track one trials.

3.1.3 Workflows

The test used three distinct workflows. To show three of different key strengths of CONVERGENCE stack, namely: publish-subscribe, in-network caching and easily handling of content replication.

3.1.3.1 Scenario one: basic publish-subscribe

In this scenario an AS France user publishes a video of playing cats, recorded with her smartphone and coded in the Apple Live Streaming format. Later, an AS Peru user subscribes to videos about cats, and receives a match for the France user publication. The Peru user fetches and plays the video.

In what follows, we describe the workflow step-by-step, showing how CoNet supports the flow of Interest messages and named-data items from publisher to subscriber.

1. France user connects to France point-of-presence (cAS3) through a web browser.
2. France user uploads her video (composed of a m3u8 file and video segments) to cAS3. In turn, this implies a set of CoNet and CoMid operations described in what follows:
 - a. cAS3 stores the m3u8 and video segments on sAS3. The video segments and the m3u8 files are named with the common prefix “youtubeStorage/”. Specifically, the name of the m3u8 file is “youtubeStorage/video_id.m3u8” and the name of the xth segment of the video is “youtubeStorage/video_id_x.ts”.
 - b. sAS3 distribute the prefix “/youtubeStorage” to the routing plane, sAS3 advertises the route to nrsAS3 and nrsAS3 propagates the new route to nrsAS1 and nrsAS4.
 - c. To support middleware (CoMid) functionality, in addition to the video content cAS3 creates and stores a R-VDI containing a reference to the video source m3u8 file with the following name (aka Network Identifier – NID) “/youtubeStorage/rvdi_id”. The RVDI is stored on sAS3.
3. To make aware customers about the presence of the video on CoNet, the France User resorts to the content-based publish subscribe feature offered by CoMid. Accordingly, the France user publishes the video meta information (namely a P-VDI) within the CoMid overlay by using the CoMid functionality of cAS3.
4. At a later stage, a Peru user connects to Peru Point-of-Presence (cAS2) through a web browser.
5. Peru user subscribes to videos matching specific search criteria (e.g. videos of playing cats). This operation is supported by the CoMid functionality on cAS2, which disseminates the subscription (namely a S-VDI) in the CoMid Overlay.
6. CoMid functionality detects a match between the France user publication (P-VDI) and the Peru user subscription (S-VDI), generates a match and notifies Peru user via cAS2.
7. The notification is an Event Report message containing the NID of the R-VDI of the video published by the French user.
8. The Peru user chooses to play the notified video content. Consequently, its Point of Presence (cAS2) downloads the R-VDI (“/youtubeStorage/rvdi_id”). This action involves the following set of routing operations:
 - a. cAS2 express an Interest to fetch “/youtubeStorage/rvdi_id”
 - b. cAS2 does not know the route to forward related Interest messages, so it contacts nrsAS2
 - c. also nrsAS2 does not know the route either, but it has a default route towards its AS provider (AS1), so it replies to the request coming from cAS2 by indicating as next hop for the prefix “/youtubeStorage” the node b1AS2
 - d. cAS2 stores this answer in its FIB and forwards the Interest to b1AS2, which will do the same interaction with the nrsAS2 and forward the Interest to b1AS1.
 - e. given that nrsAS1 and nrsAS3 both know the right ICN route for “/youtubeStorage”, all CoNet nodes now know how to route Interest messages for named-data with the “/youtubeStorage” name-prefix

9. After the download of the R-VDI, cAS2 extracts the NID of the m3u8 file, i.e. “/youtubeStorage/video_id.m3u8” and sends it to the Peru user. Then Peru user plays the video by fetching via cAS2 the m3u8 file and then the video segments one-by-one. Within the CoNet, the Interest messages needed to fetch the video traverse the path from cAS2 to sAS3 (cAS2-b1AS2-b1AS1-b2AS1-b1AS3->sAS3).

3.1.3.2 Scenario two: in-network caching

An Italy user makes a subscription and fetches video segments. In this case the video segments are fetched from the cache of France’s Border Gateway node b1AS3 – since this node has been already traversed by the video segments in the first scenario. Therefore, the content distribution provider sees no additional traffic on France’s Serving Node (sAS3) and Italy user benefits from lower latency (the copy is closer to the user than the original version).

3.1.3.3 Scenario three: simple content replication

Content distribution provider decides to replicates “youTube” contents to Peru Serving Node (sAS2) to better support a growing demand coming from this region. This enables the provider to offer a better service to Peru users and to reduce load on the France Serving Node. The content distribution provider performs this operation explicitly through a web based administration console. The two repositories begin synchronization. When they have finished, the replica Peru Serving Node issues a routing-plane update message, advertising that AS2 can now serve resources with the ““/youtubeStorage” name-prefix.

Once content replication and the updating of the routing plane are complete, any Peru user, who requests a video, sees her request automatically routed to the nearer serving node sAS2.

3.2 Progress on specific issues

3.2.1 ICN video streaming for cellular environments

In this section, we study a P2P ICN application for live streaming of videos encoded at multiple bitrates (aka adaptive streaming). The streaming application can be deployed either on top of CONET [7], or on top of a CCN network [6] implemented with the CCNx tool [8]. In what follows, therefore, we use the terms CONET, CCN, and CCNx interchangeably.

The streaming format is MPEG DASH [14]. Peers are a small set of neighbouring mobile cellular devices, such as smart-phones or tablets cooperatively using use their cellular connections to improve playback quality, with respect to the quality they could achieve by downloading the stream independently.

The application logic resembles the distributed BitTorrent approach sketched in Figure 20. Peers fetch different segments through the cellular interface, and exchange them with other neighbouring peers via a proximity (one-hop) connection. Although BitTorrent-like

approaches have been widely investigated in the literature, the combination of methods we used ensured that our application was completely novel.

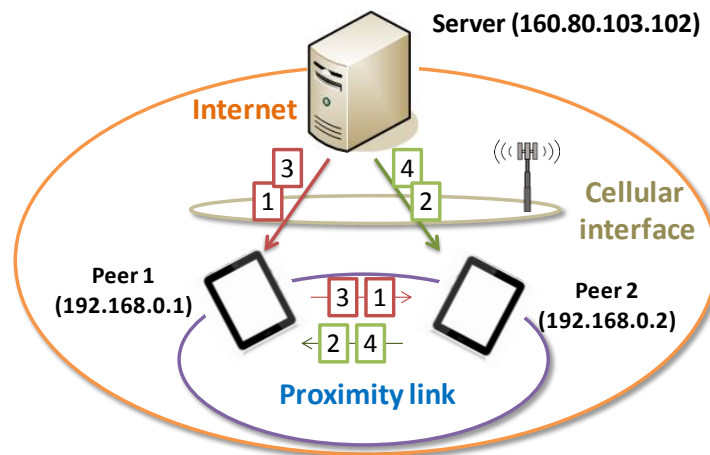


Figure 20 – Application scenario

The specific characteristics of our solution can be summarized as follows.

- *We developed the application on top of an ICN solution (i.e. CCN or CONET).* We could have used the raw TCP/IP API, but this would have been a complex solution. By contrast CCN natively provides several of the essential functions needed by the application: in particular name-based routing, caching and multicasting. We exploit these functionalities and propose solutions to orchestrate their interplay;
- *The video is distributed as a live stream.* Hence, at a given time t it is possible to fetch only segments that have been published by the server up to time t . This requires an extension of the plain BitTorrent approach, where files are fully available from the beginning;
- *The video is available at multiple bit-rates.* Hence, we designed a distributed rate selection algorithm. This leads all peers to choose the same coding rate, always choosing the highest rate possible.
- *There is no central coordination entity*, like a Torrent tracker, that assists peers in discovering which data items other peers may offer. Consequently, we devised a CCN-based distributed content discovery function.

To assess the performance of the application, we used an experimental test-bed based on Linux devices and real 3G connections. To allow replication of our results, we released the code of our application as open-source [19].

3.2.1.1 Scenario and assumptions

As shown in Figure 20, we consider a group of neighbouring users using their mobile devices to watch the same live video stream, offered by a server in the fixed Internet. The group size is quite small (e.g. five peers). Thus scalability of the application with respect to the number

of peers is not a central design issue. Mobile devices are connected to the Internet through a cellular interface. Additionally, the devices use a proximity wireless technology, e.g. Wi-Fi Direct [15] to form a full mesh (i.e. one-hop) network with each other. The transfer capacity of the proximity mesh is much greater than that of the cellular network. Thus the cellular interface is the main bottleneck on the network. The mobile devices and the server all implement CCN functionality. Mobile devices run the P2P streaming application. The server is merely a CCN Repository, storing the video data.

3.2.1.2 The streaming scheme

CCN is meant for *pull*-based services, where clients fetch contents from the network, without caring about where the data comes from. Therefore, CCN deployments use streaming schemes in which video parts are pulled from clients, rather than being pushed by servers.

Nowadays, there are HTTP-based streaming schemes using a pull service model. The advantage of these schemes is that they can leverage HTTP caching to improve the scalability of video distribution. The video server is an HTTP server that publishes *video segments* (e.g. 2 sec of playback) addressed by URLs; the video client is an application that downloads the segments through HTTP GET operations. Examples of this kind of streaming scheme include Apple Live Streaming, Microsoft Smooth Streaming and MPEG Dynamic Adaptive Streaming over HTTP (DASH) [14].

In principle, any pull-based streaming scheme could be deployed on a CCN (or CONET), simply by replacing the HTTP video server with a CCNx Repository and the HTTP GET operations of the client with CCNx GET operations (e.g. `ccngetfile` of CCNx). For our P2P application we choose MPEG DASH: a standardized solution that supports rate adaptation.

MPEG DASH video segments are usually M4S files, available at different bit-rates. Receiver-driven rate adaptation can be easily implemented by dynamically changing the coding rate of the downloaded segments. An XML Media Presentation Descriptor file (MPD) provides meta-information about the video segments: URLs, coding, playback timing, etc.

To play the video, a DASH client fetches the MPD and then starts to pull and play M4S video segments from the network. A local control algorithm exploits the MPD timing information to select the bit rate.

3.2.1.3 Video server and naming scheme

The video server is merely a CCNx Repository that publishes MPD and M4S files. The MPD file is available from the start of the video distribution; M4S video segments are published during the live streaming.

To synchronize peers and video source, the server publishes Video Timing Information (VTI) for each new segment. This contains the segment number of the last published video segment and the server clock.

MPD, VTI and M4S data items use the following naming schemes:

MPD ccnx:/server-prefix/filename.mpd

VTI ccnx:/server-prefix/filename.vti

M4S ccnx:/server-prefix/filename/SN=x/BW=y.m4s

where x is the video segment number and y is a coding rate. Thus, 'ccnx:/foo.eu/video1/SN=10/BW=100.m4s' identifies an M4S segment where 'foo.eu' is a prefix identifying the video server, 'video1' is the file name, the segment number is 10 and the coding rate is 100 bps.

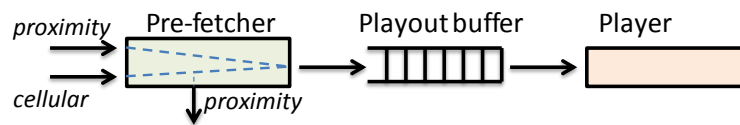


Figure 21 – Pre-fetcher, playout buffer and player

3.2.1.4 Video Peer

Mobile devices (aka peers) share their cellular access to cooperatively download the video stream at a coding rate higher than any one of them could have achieved by downloading the stream independently. In this section we describe the main operations carried out by a video peer to achieve this goal and discuss how these operations are implemented by CCN.

Peer joining

To join the video stream, the peer downloads the VTI file and synchronizes with the video source, i.e. it becomes aware of the latest segment number published by the source and of the source clock. This synchronization is not very precise. However, it is sufficient. After synchronization, the peer fetches the MPD file and begins to cooperate with other peers to pre-fetch and play segments.

Collaborative pre-fetching

Figure 21 shows the main components of a video peer. A *pre-fetcher* concurrently downloads video segments from the proximity and cellular interfaces, and upload segments received on the cellular interface to the proximity interface. Downloaded segments fill a playout buffer, drained by a DASH video player that starts the playback when the buffer contains $2P$ segments. Thus, the playout delay is $2 P T_s$ where T_s is the playback duration of a video segment (e.g. 2 sec).

The pre-fetcher considers the stream as formed by a sequence of *windows* each containing P segments (a similar concept is used in [20]). When a new window of segments is published by the source, a new *pre-fetch round* starts. At the beginning of a pre-fetch round, a peer randomly shuffles the sequence of segments to download, thereby avoiding *duplicated cellular fetches*, when two or more peers download the same video segment over the cellular interface. Once the sequence of segments to download is settled, the peer starts to pull one

missing segment at a time using the cellular interface; at the same time, the peer tries to download the missing segments from the proximity interface. Since all segments are available at the source at the start of the pre-fetch round, all peers use both the cellular and proximity interfaces, maximizing the use of radio resources and the achievable coding rate.

Figure 22 shows the sequence of events involving the video source, the pre-fetchers ($P=5$) and the players of two peers while the source is publishing segments 15÷19. During this period, the video players play segments 5÷9 and the pre-fetchers collaboratively download segments 10÷14 from the cellular interface, sharing these segments through the proximity interface. The random sequence chosen by peer 1 is {12 10 14 13 11}. Peer 2 chooses {13 14 11 10 12}.

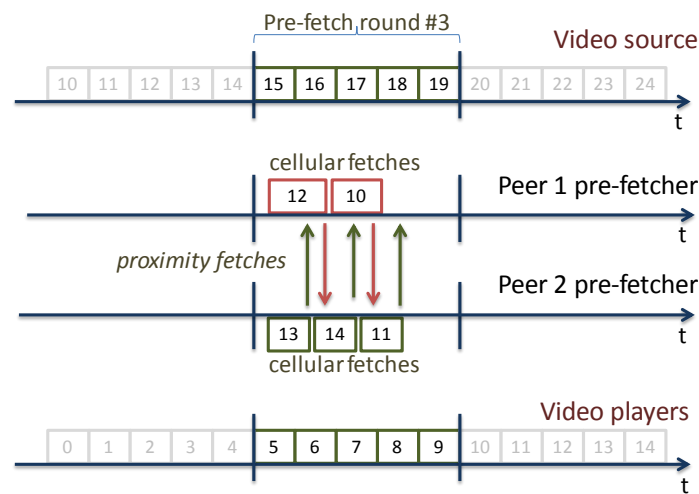


Figure 22 – Sequence of events for the video source, the pre-fetchers ($P=5$) and the players on the two peers

Cellular fetches

To support cellular fetches, the CCN FIB of each peer is preloaded with a *remote-route* that addresses the server-prefix and points to the remote video server. For example, in Figure 23, the FIBs of peers 1 and 2 contain the remote-route ‘ccnx:/foo.eu’, directed to the video server 160.80.103.102:9595, via the cellular interface (rmnet0).

In Figure 22, when peer 1 has to fetch segment #10, it requests the segment by name from the CCN API. Using the remote-route, the CCN functionality forwards the request and sends back the data through the cellular link.

Proximity fetches

To support proximity fetches, a peer temporarily inserts a *proximity route* into its CCN FIB. The route points to the neighbouring peer via the proximity interface. Then, the peer requests the segment by name to the CCN API. The CNN functionality uses the proximity route to forward the request and sends back the data through the proximity link.

Figure 23 describes the FIB of peer 1 and 2 during the pre-fetch round depicted in Figure 22. Peer 1 downloads segments 10 and 12 ('ccnx:/foo.eu/video1/SN=10/BW=100.m4s' and 'ccnx:/foo.eu/video1/SN=12/BW=100.m4s') from the cellular interface.

To fetch these segments from the proximity interface, peer 2 inserts the two related proximity-routes into its FIB, which now points to peer 1 (192.168.0.1:9695) through the proximity interface (wlan0).

FIB of peer1	
Name prefix	output-face
ccnx:/prd	224.0.0.1:9695 (wlan0)
ccnx:/foo.eu	160.80.103.102:9695 (rmnet0)
ccnx:/foo.eu/video1/SN=11/BW=100.m4s	192.168.0.2:9695 (wlan0)
ccnx:/foo.eu/video1/SN=13/BW=100.m4s	192.168.0.2:9695 (wlan0)
ccnx:/foo.eu/video1/SN=14/BW=100.m4s	192.168.0.2:9695 (wlan0)
...	

FIB of peer 2	
Name prefix	output-face
ccnx:/prd	224.0.0.1:9605 (wlan0)
ccnx:/foo.eu	160.80.103.102:9695 (rmnet0)
ccnx:/foo.eu/video1/SN=10/BW=100.m4s	192.168.0.1:9695 (wlan0)
ccnx:/foo.eu/video1/SN=12/BW=100.m4s	192.168.0.1:9695 (wlan0)
...	

Figure 23 – CCN FIBs of peers

Every segment request matches to a remote-route. Where a proximity-route is available this is preferred, since it offers a longer prefix match.

Proximity route discovery

Using the proximity route discovery functionality, a peer promptly finds out the availability of proximity routes, i.e. of video segments in the neighbourhood. The discovery protocol is the following.

When a peer starts downloading segment 'ccnx:/server-prefix/filename/SN=x/BW=y.m4s' from the cellular interface, it publishes a *Proximity-Route-Info* (PRI) signalling message (from a CCN point of view, this is similar to any other content). This message contains the control information needed to setup the related proximity route on other peers.

Specifically, a PRI contains: i) the IP address and CCNx port of the peer, ii) the coding rate y and iii) the estimated net cellular rate of the peer (discussed later). The PRI name includes the name of the related video segment, without the coding rate component BW=y, and with the *control prefix* 'prd' (proximity-route-discovery). Specifically, the naming scheme of the PRI is:

ccnx:/prd/server-prefix/filename/SN=x

During a pre-fetch round, peers periodically attempt to discover the availability of missing segments on the proximity interface by retrieving the related PRIs. Requests for PRIs are routed-by-name to a preconfigured multicast address on the proximity interface. To achieve this, the FIB of each peer is preloaded with the entry ‘ccnx:/prd’, pointing to such multicast address (see Figure 23). When a PRI is retrieved, a peer inserts the related proximity route in the FIB and then requests the video segment to the CCN API, so realizing a proximity fetch.

Figure 24 reports an example of a discovery procedure, followed by a proximity fetch. Peer 1 downloads segment ‘ccnx:/foo.eu/video1/SN=10/BW=100.m4s’ which it is not available in neighbouring devices through the cellular interface. It then publishes a PRI, with identifier ‘ccnx:/prd/foo.eu/video1/SN=10’. The PRI includes the peer IP address:port (192.168.0.1:9695) and the coding rate of the segment (BW=100). Peer 2 misses segment #10, so it periodically attempts to pull its PRI; once it has been published, the PRI is promptly discovered by peer 2, which inserts the associated proximity-route into the FIB and starts downloading segment #10, by sending related Interest messages over the proximity link.

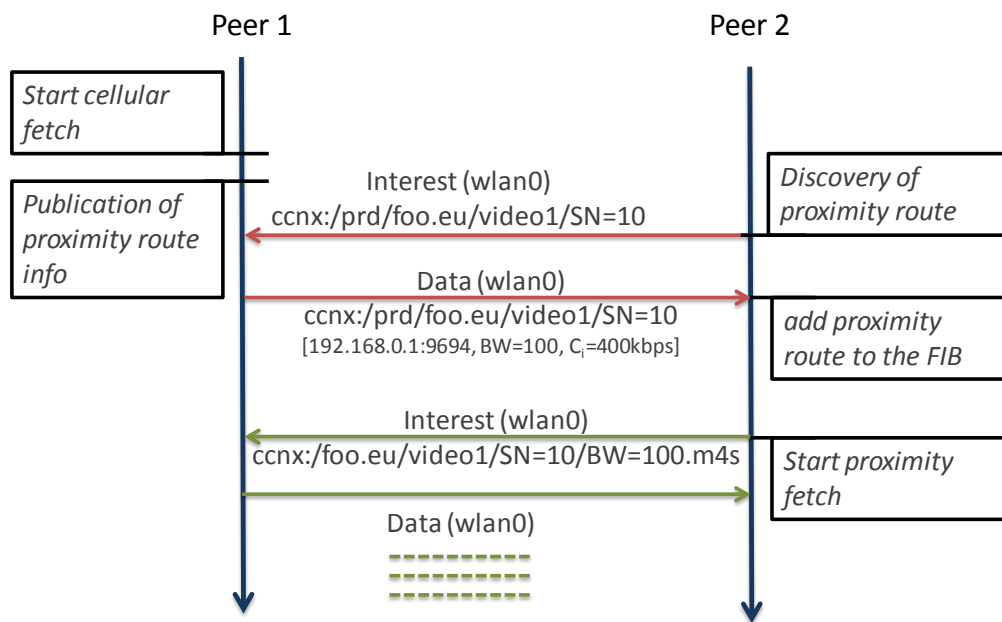


Figure 24 – Discovery and Proximity fetch

When Interest messages from peer 2 reach peer 1, the pre-fetcher on peer 1 is already downloading the segment from the cellular interface. In this condition, the PIT mechanisms of the CCN form a temporary multicast delivery tree, so that the chunks of the video segment received from the cellular interface of peer 1 are delivered both to the pre-fetcher of peer 1 and to the peers requesting them through the proximity interface, e.g. peer 2. If discovery is late, the same result can be obtained by using peer 1’s content cache.

Selection of the coding rate

At the end of a pre-fetch round, each peer computes the coding rate of the video segments that will be downloaded in the next pre-fetch round. Peers use a heuristic rate selection algorithm that operates on the base of: i) the available video coding rates BW_h ; and ii) the *net* rate C_i that each peer can obtain through the cellular interface, i.e. the maximum download rate seen at the application layer.

Assuming that the video has L possible coding rates, a peer discovers these rates from the MPD file fetched during the join operation. Values for C_i are periodically distributed to all peers allowing them to dynamically adapt coding rates to changing conditions on the cellular links. To achieve this, peers communicate their estimated average cellular rate $C_i(k)$ during the previous round $k-1$ (see Figure 24) by piggybacking the information in the PRI, during pre-fetch round k . It calculates the rate either by monitoring the time needed to download video segments from the cellular interface or by downloading dummy files suitably arranged beforehand.

Then, each peer sorts the M peers by decreasing order of $C_i(k)$. To compute the coding rate BW_h to be used in round k , the peer solves a constrained maximization problem:

$$J_{i,h} = \text{floor} \left[\frac{P C_i(k)}{BW_h} \right] \quad (1)$$

$$\max_h \left\{ s. t. \sum_{i=1}^{\min(P,M)} J_{i,h} \geq P \right\} \quad (2)$$

This is necessary because of a *quantization-constraint*: a peer cannot choose to download part of a video segment, but only a whole segment. In this setting, Eq. (1) represents the number of video segments that a peer can download completely over the cellular interface during a pre-fetch round, assuming that segments are coded with a constant bit rate BW_h . Maximizing (2) yields the highest possible rate h such that the peer can download all the segments before the end of the round.

The sum of (2) is limited to $\min(P,M)$ since at most P peers are required to fetch P segments from the cellular interface. Thus, at most P peers can actually carry out cellular fetches. Furthermore, even when $P \geq M$, the solution of (2) may be such that some peers cannot download segments from their cellular interface. Only peers with $J_{i,h} > 0$ carry out cellular downloads. Peers with $J_{i,h} = 0$ are unable to download even a single segment before the end of the round and do not therefore carry out cellular downloads.

3.2.1.5 Test-bed results

A prototype of the streaming application was implemented using Java and the CCNx tool (currently the forwarding engine of CONET). The prototype runs on laptops with the Linux, Kubuntu 12.04 distribution.

The video player - VLC 2.1.0 - interacts with the streaming application through a local HTTP connection. Therefore, from the VLC point of view, the streaming application is an HTTP proxy.

We verified the effectiveness of the application on a test-bed formed by a video server and five peers. The video server is on the public Internet. Peers are connected to each other by a Wi-Fi ad-hoc link and connected to the video server either through a real HSDPA cellular connection or through an emulated connection. In the former case, each laptop is tethered via USB with a mobile phone, connected to the HSDPA service provided by the same mobile operator. In the latter emulated case, each laptop is directly connected to the server through a dedicated Ethernet link, whose rate is controlled by the Linux Traffic Control tool. We streamed the MPEG DASH video “Big Buck Bunny” [21], whose resolution is 480p; there are fourteen available video qualities, with bit-rates ranging from 100 kbps to 4.5 Mbps; the video is made up of about 270 segments. The segment duration T_s is 2 sec.

Tests with HSDPA connections

Figure 25 reports the coding rate for video streaming (BW_h) when three collaborating peers are connected to the server over HSDPA links. The pre-fetch window size P is equal to 10. The ticks of the y-axis of the plot indicate the first eleven available coding rates. The figure also shows the cumulative net bandwidth $C_{tot} = \sum_i C_i$ available on the cellular (HSDPA) interface. Obviously this places an upper bound on the coding rate. The bit rate is measured on peer 1 (HTC Desire HD), which is present from the beginning of the test; peer 2 (Samsung Galaxy SII) and peer 3 (Samsung Galaxy Nexus) join the group at 140 sec and 220 sec, respectively. We observe that by increasing the number of peers, the cumulative net cellular bandwidth and the coding rate of the streaming follow a similar behaviour and increase too. At the end of the test, we observe increased cellular congestion (for external reasons). In these conditions the algorithm reduces the coding rate – the appropriate response.

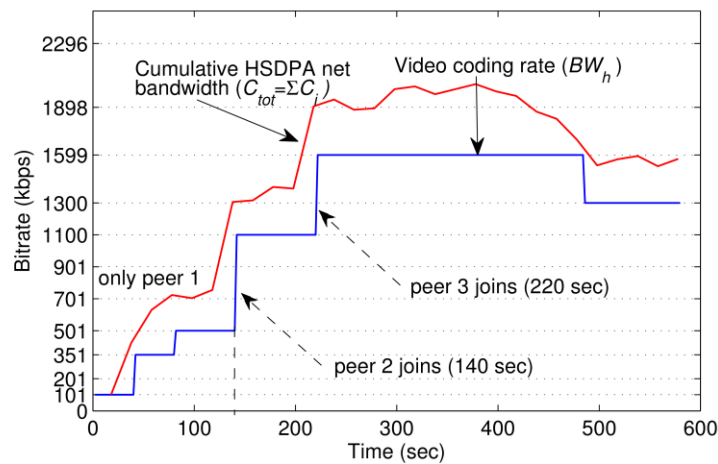


Figure 25 – Video coding rate and cumulative net cellular (HSDPA) rate seen by peer 1 in case of three peers and $P=10$

The selected video rate BW_h is sometimes much lower than the cumulative net bandwidth C_{tot} available on the cellular links. For instance at sec 350 C_{tot} is about 1950 kbps and BW_h is 1599 kbps rather than the maximum possible rate of 1898 kbps. These figures suggest that the rate selection algorithm may be too conservative. However, this is due to the quantization-constraint. In other words the conservative behaviour is due to the floor operator in (1). In preliminary tests, we removed the operator, obtaining coding rates close to the cumulative net cellular bandwidth. However this led to freezes during playback. When we used Eq.(1) to select the bit rate there were no freezes.

Figure 26 shows *raw* traffic (including protocol overhead) received on the HSDPA and Wi-Fi interfaces by peers 1 and 2, during the test shown in Figure 25. (Values for peer 3 were similar). During the period shown, all peers were present and the stream was coded at 1599 kbps. Peers used the cellular interface continuously and equally, generating raw HSDPA traffic at an average rate of 600÷650 kbps. The reason for these rather low rates was that we carried out the measurements during the lunch break, when most of the students on campus use their mobile phones - a worst-case scenario for our application. The Android Speedtest.net application confirmed this interpretation of our data.

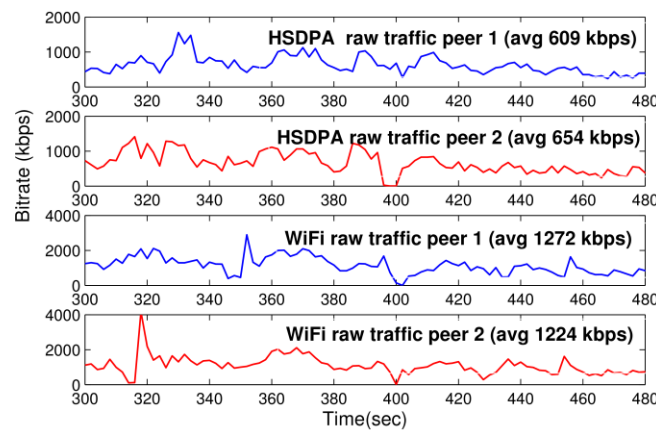


Figure 26 – Received raw traffic for three peers with $P=10$

The sum of the raw bit rates downloaded from HSDPA was roughly 1880 kbps. Given that the data downloaded from HSDPA by neighbouring peers was shared over Wi-Fi, this value is equal to the sum of the HSDPA and Wi-Fi rates obtained by a single peer. The average video coding rate measured in the specific period of the plot was about 1520 kbps (nominal:1599 kbps). This means that the protocol overhead on the HSDPA interface due to CCN, and lower layers, was approximately 19%.

Tests with emulated connections

In the scheme we are proposing, one of the most fundamental parameters is P : the length of the pre-fetch window. Given that the playout delay is equal to $2 P T_s$ it would be preferable to have a small pre-fetch window. However, windows that are excessively small can be a source of inefficiency. We thus have to find a reasonable trade-off.

To analyse the impact of P on performance we used a controlled test-bed, in which we *roughly* emulate the cellular interface with a dedicated Ethernet link, whose rate is controlled by the Linux Traffic Control tool. This approach ensures that the only parameter that varies during the test is P .

We considered a scenario with three peers ($M=3$) with raw cellular rates of 1000, 1000, 400 kbps respectively; and another scenario with five peers ($M=5$), with raw cellular rates of 1000, 1000, 400, 400, 400 kbps. The cumulative net cellular rate C_{tot} was about 1930 kbps with three peers, and 2590 kbps with five peers. These values are consistent with 19% protocol overhead estimated previously.

Given these values, we expected to obtain video coding rates of 1599 and 2296 kbps, the highest possible rates lower than C_{tot} with three and five peers, respectively. However, Figure 27 shows that it was only possible to reach such values when P was higher than a minimum threshold value: $P=6$ for three peers and $P=15$ for five peers.

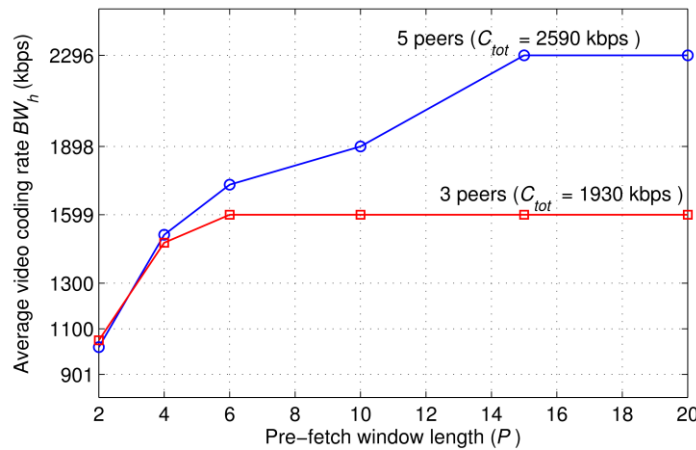


Figure 27 – Video coding rate versus pre-fetch window length (P) with emulated connections

This behaviour is due to the interplay of three factors (discussed hereafter), whose effects tend to vanish as P increases. The relationship between P and M , can be described as follows:

- (for $P < M$): the quantization-constraint implies that no more than P peers can perform cellular fetches. Thus, for $P < M$ there are always $M - P$ cellular links that are cannot be exploited;
- (for $P < 2M$): A small pre-fetch window (e.g. $P < 2M$) can lead to duplicated cellular fetches, a waste of cumulative net cellular bandwidth and temporary reductions in the coding rate. When this happens the coding rate temporarily switches between two neighbouring levels. Consequently, the *average* coding rate is different from any of the coding rates available to the selection algorithm (y-axis ticks);
- (for $P < 3M$): the conservative effect of the floor operator in Eq.(1) is more severe for smaller pre-fetch windows. With five peers the effect vanished for $P \geq 3M$; with three peers the limit was $P \geq 2M$.

Understanding these *specific* results makes it possible to derive a *general* approach for choosing P . As P increases, the effect of the floor operator in Eq. (1) tends to disappear. Therefore, in what follows we only consider the effects of Eq. (1).

We set an *exploitation target*, expressed in terms of a percentage T of the cumulative net cellular bandwidth C_{tot} . Then we look for the minimum value of P compatible with a stream coded at $T \cdot C_{tot}$. To this end, we assume an ideally coded video that can be delivered at all possible rates and which solves the following constrained minimization problem:

$$\min_P \left\{ s. t. \sum_{i=1}^P J_{i,h} \geq P \right\} \quad (3)$$

where $J_{i,h}$ is evaluated using Eq. (1), assuming $BW_h = T \sum_i C_i$.

We solve the minimization problem using a MATLAB simulator, in which we assume $M=\{3,5\}$ peers, each with a cellular rate C_i defined by a uniformly distributed random variable in the range $R \pm \Delta \cdot R$. In this way the random variable $C_i / (T \sum_i C_i)$ in Eq.(1) is independent from the mean rate R . Thus the result of the minimization (3) is also independent from R .

Figure 28 plots the values of P , normalized to the number of peers M , against the exploitation target (T). For a given value of T , we observe that P/M increases gradually with increases in: i) the variability of cellular rates among peers, e.g. from $\Delta=20\%$ to $\Delta=80\%$, or ii) the number of peers, e.g. from 3 to 5. However, for any given T , the increases are quite small. Conversely, changes in T lead to much larger differences. In summary, the more we want to exploit cellular resources, the larger should be the pre-fetch window P (and the playout delay).

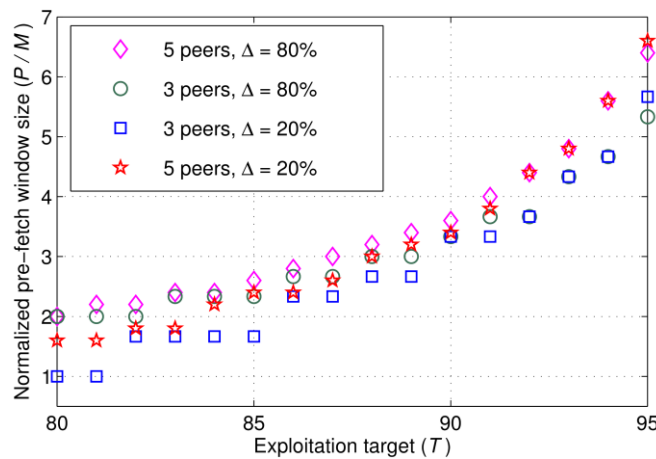


Figure 28 – Simulated minimum pre-fetch window length (P) versus exploitation target (T)

These results can be used to choose an optimal size for P . For instance, Figure 28 shows a case in which we can exploit 85% of the cumulative cellular rate, by choosing a pre-fetch window size between 2 and 3 times the expected maximum number of peers. If the

corresponding playout delay is not acceptable, we should reduce the length of the video segments.

3.2.1.6 Related Works and Conclusions

Studies rather similar to this paper can be found in [16][17][18][20] and [28]. Paper [16] is our direct precursor.

In [17] the authors propose AMVS-NDN, a CCN adaptive video streaming application, which enables a mobile device either to use its own 3G/4G connection or to connect via WiFi to another device, which in turn is connected to the video server through its 3G/4G interface. In other words, AMVS-NDN does not aggregate the capacity offered by different cellular links but simply uses the best link. In this way, the resulting video quality is limited by the bit-rate available through this *single* link.

In [28] the authors set up a test-bed for video streaming over a CCN, which they call NDNVideo. In their scenario, they consider a *fixed network* and a *traditional client-server interaction model*, i.e. without P2P cooperation. This is a rather different approach to ours. However the naming scheme is similar to our scheme.

In [18] the authors propose a similar application *based on TCP/IP*, which deals with *on-demand single-rate* video streaming. In [20] the authors propose a BitTorrent approach for live streaming in a *fixed network*. The video is *single-rate* and the goal is to *offload the server*.

Given that we use multiple (cellular) links to fetch data, our application has some resemblance to “multi-homed” video streaming [27], However, in a multi-homed scenario, the same device uses different links and *distributed cooperation with other devices is not required*.

Clearly there are many papers on P2P video streaming (see e.g. [26] and its references). However, we believe that our own work is the first to propose a P2P ICN application for multi-rate (DASH) live video streaming over cellular devices.

3.2.2 Naming, content integrity and caching

In this section we discuss the interplay between naming, content integrity, and caching, focusing on issues that are common to different ICN proposals [1][5][6][7][8][9][10][11][12][13], regardless of their specificities. For the purpose of concreteness, we base our quantitative investigation on an ICN model similar to our CONET [7] and CCN [6] where: i) content is identified through names (strings); ii) content is segmented into chunks; iii) each chunk has a unique network identifier, which includes the content-name as a prefix; iv) to fetch a content, a client sequentially downloads all the component chunks [22]; v) network nodes *route-by-name* requests for chunks on the shortest-path and by using a name-based routing table [23] [30]; vi) network nodes provide an *en-route caching* service [31]; vii) the latter two processes take place *at line rate* [43].

Our first goal is to extend previous analyses of the impact of different content naming schemes on the security properties of the information-centric network [32][33], and to discuss the pros and cons of different combinations of human-readable and self-certifying names with traditional or Identity-Base signature schemes. Second, and perhaps more significantly, we evaluate the impact of signature processing on ICN caching performance. Our major finding, which at a first glance may appear counter-intuitive, is that *the limited speed of practical signature verification algorithms is not a critical problem for an ICN* even if it does not support content caching at line rate (a substantial fraction of content cannot be verified in time for caching). We show on the contrary, that caching performance, in terms of cache hit probability, may even *increase* when the network uses an LRU caching policy.

3.2.2.1 Naming and Content Integrity

3.2.2.1.1 Possible combinations of naming schemes and signature approaches

It is well known [32][33] that the choice of the naming scheme in an ICN has implications on security. Table 4 classifies naming schemes along two main independent axes: impact on routing plane and impact on security properties. Along the first dimension we have names that can be hierarchical or flat; along the second dimension we have names that can be self-certifying or human-readable.

<i>Security</i> \ <i>Routing</i>	Flat	Hierarchical
Human-readable	Foo.com.video1.mp3	Foo.com/video1.mp3
Self-certifying	0x3fb889fffa	0x65de3/video1.mp3

Table 4 – Naming schemes

The main impact of the choice between flat or hierarchical names is on the ICN routing plane. For an in depth discussion, readers are referred to deliverable D4.3, where we analyse the possibility of using the Principal/Label (P/L) [5] hierarchical naming scheme for the deployment of a worldwide ICN. This is the case we will consider in the remainder of this section.

A P/L hierarchical name is formed by a sequence of component strings, separated by a reserved character, e.g. “/”. The first component *P* is an identifier of the *principal* of the resource (e.g. Foo.com). Contents published by the same principal have the common prefix *P* and are differentiated by the following sequence of components, which form the so-called Label (*L*). At the chunk level, *L* also includes components used to identify specific chunks (e.g. Foo.com/video1.mp3/chunk1).

Since content is delivered to users not only from trusted origin servers but also from distributed caches, users can exploit “security” information included in the data item [6] to

verify the validity of the data they receive. As pointed out in [29] and [46], this possibility is also open to caching nodes. This prevents poisoning of caches by fake content. Content is considered valid if it meets the following criteria:

- **Integrity:** received content has not been modified, i.e. it is the originally published content
- **Provenance:** source of the content is authentic, i.e. the data is provided by the principal P
- **Relevance:** received content is really the content requested by the user

Assuming that the data publisher of data has a real-world identity and that she signs the content she publishes with her public/private key pair, these requirements can be satisfied using digital signatures¹⁴. There are many different digital signature schemes. The viability of a particular scheme for a specific case depends both on the properties of the name structure (e.g., human readability) and on performance issues (e.g. speed of verification, bandwidth overhead, etc.).

Table 5 sketches the format of a data unit incorporating signature-related information. Every chunk of data D is packaged with a header field N , which is the chunk name in the P/L form. The principal uses her private key to digitally sign the resulting package $C=\{N, D\}$. The signature S , is included at the tail, in a verification block V , along with any other information (*INFO*) that network nodes need to perform *on-line* verification of C against S . Each network data unit U is then $\{C, V\}$, where $C=\{N, D\}$ and $V=\{S, INFO\}$.

Since the name N is part of the digitally signed material C , and it is not possible to change the name of the data item without failing the digital verification, the *relevance* of the data unit is automatically guaranteed.

Data Unit	Content	Name = P/L
		Data
	Verification block	Signature
		Additional <i>INFO</i>

Table 5 – Data packaging model

We now explore trade-offs for three different combinations of name structure and signature algorithm.

1) Human-readable names & traditional signature - the principal identifier P is a human-readable string, e.g. “Foo.com”, and the signature is a traditional one, such as the popular RSA [37] or ECDSA [36] algorithms. The additional *INFO* field (see Table 5) contains the Digital Certificate of the Principal. This enables nodes to verify chunks *on-the-fly*, i.e. without having to download the certificate from a remote storing node. The digital certificate includes

¹⁴ In this paper we don’t deal with Confidentiality (content encryption) that remains an end-to-end service between publisher and end-user.

the public key of the principal, properly signed by a Certification Authority (CA). Verification of the signature S and of the certificate ensures integrity. To verify provenance, we want to be sure that the principal identified by P in the name N , really is the owner of the content. This is done i) by reading through the certificate, which connects P with its public key ii) and by verifying the signature. In this case, secure in-network caching requires the presence of a CA infrastructure, whose public keys are preloaded on network nodes.

2) Human-readable names & ID-based signature - IDentity-based signatures also use Public/Private key pairs. However, in this case, the Public key can be any string, e.g. the principal identifier P itself [35]. The user's Private key is generated by a Private Key Generator (PKG) infrastructure. The signature algorithm is usually based on the discrete logarithm problem and elliptic curves [34]. The Additional *INFO* field (see Table 5) includes an identifier for the PKG. The verifier uses the identifier to look up the configuration parameters for the chosen PKG. The parameters are preloaded on network nodes. Verification of the signature S using these parameters ensures integrity. Given that the public key is the principal identifier P , we do not need a certificate connecting P with its public key, to verify provenance. Secure in-network caching requires the presence of a PKG infrastructure. Distributed infrastructures are possible [44].

3) Self-certifying names & traditional signature – a self-generated Public key is used as the principal identifier P . Signature algorithms may be RSA or ECDSA. The Additional *INFO* field (see Table 5) is void. Integrity is assured by verifying the signature S . Like with identity-based signatures, provenance is verified by checking that the public key used for the signature is the principal identifier P . There is no need for a CA or PKG infrastructure. The drawback is that the principal identifier P is not human-readable. To go from keywords/descriptors to a principal identifier P , users will have to use trusted translation services (DNS, Google-like services, personal address books, etc.).

3.2.2.1.2 Overhead and verification time

Table 6 reports the sizes of the verification block V (see Table 5) for our different cases. Configuration parameters are chosen to provide the same level of security in all cases. Specifically, for RSA signatures we use keys of 1024 bits; for ECDSA signatures, we use an elliptic curve over a 160 bit prime field (NIST secp160k1 [39]); for ID-based signatures, we use the same elliptic curve combined with the approach proposed in [34], which does not require a (slow) pairing computation. Verification speed was evaluated using the OpenSSL API and an Intel I7 processor running at 1.8Ghz. In combinations 1a and 1b, the system would normally have to verify the digital certificate as well as the signature. We assume that the verification of the certificate takes the same time as the verification of the signature. For example, with combination 1a, it takes 0.06 ms to verify the signature S . We therefore assume it takes 0.06 ms to verify the digital certificate.

Techniques based on elliptic curves (ECDSA and ID-based) provide lower bit overhead with respect to RSA but have significantly slower verification performance. Considering that ICN chunks are relatively small (e.g. 4kB), the overhead introduced by combinations 1a, 1b and 3a is rather high. This large overhead could absorb all the well-known traffic-reducing effects of ICN. It seems, therefore, that the best choice is to use human-readable names with identity-based signatures (2) or self-certifying names with ECDSA (3b). However, in the Identity-Based scheme the network requires an external, human-managed PKG infrastructure. This may complicate network deployment.

Combination	Add. INFO	Sign. (bits)	Add. INFO (bits)	Verification time (ms)
1a- H.R. with RSA	Certificate	1024	2048	0.12
1b - H.R. with ECDSA	Certificate	320	408	0.58
2 - H.R. with I.B.	PKG id	506	2	0.33
3a - S.C. with RSA	None	1024	0	0.06
3b - S.C. with ECDSA	None	320	0	0.29

Table 6 – Sizes of the verification block fields and verification time in the case of Human Readable (H.R.) and Self-Certifying (S.C.) names, and RSA, ECDSA and Identity-Based (I.B.) signature schemes

3.2.2.2 Content integrity and caching performance

To avoid denial of service due to cache poisoning [46] (i.e. injection of fake content), ICN nodes need to check the validity of each cached chunk [29]. *Asymmetric* key algorithms (e.g. RSA, ECDSA, ID-based) are usually believed to be unsuitable for line rate operations on backbone nodes, which run at speeds in the order of tens of Gbit/s. The results shown in Table 6 suggest that, with chunks of 4kB, options (2) and (3b) can support signature checking at a maximum rate of 90 Mbps and 120 Mbps, respectively. As a consequence, verification on backbone ICN nodes may have to skip many incoming chunks. In this case the stream of data items entering the cache is a random sampled version of the whole ingress stream. This is what we call *lossy caching*.

In what follows, we use the functional model depicted in Figure 29. When a request r_i for the i th chunk reaches a node, the node looks it up in the cache. In the event of a hit in the cache, chunk c_i is retrieved from the cache and sent back to the user. In the event of a miss, the node forwards the request r_i upstream. When chunk c_i returns from an upstream serving device (the original server, an intermediate cache, etc.), the chunk is immediately forwarded to the user (not shown in figure). Meanwhile a copy of the chunk is inserted in the FIFO buffer of the signature verification process. If verification succeeds, c_i is inserted in the cache. Since the verify buffer is limited and the arrival rate could be greater than the verification rate, the buffer performs as a lossy queue. Thus, a request that suffers for a cache miss may, or may not, result in a later insertion of the corresponding chunk in cache.

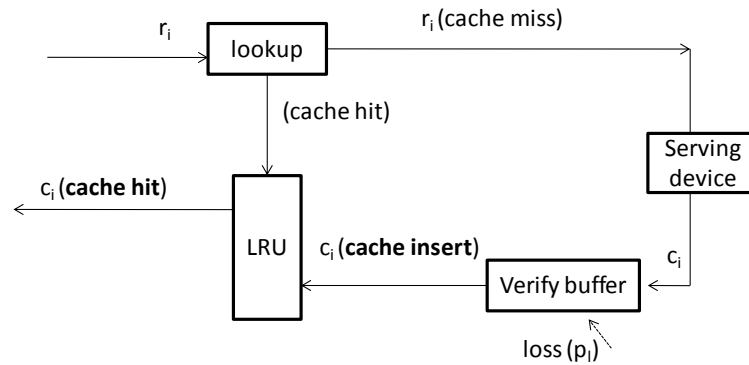


Figure 29 – Caching: a functional model

We consider a Least Recently Used (LRU) replacement strategy, e.g. implemented with a traditional *stack of references*. The stack contains the references of the data items actually stored in the cache memory. Each time an item is retrieved from or inserted into the cache, its reference is moved to the top of the LRU stack. In the case of insertion events (and cache full), the insertion of a new item implies the removal of the last item referenced by the stack, that is, exactly, the Least Recently Used item.

Storage capacity C is expressed in terms of the number of data items that can be stored in the cache. This is a reasonable assumption for in-network caching, where cache capacity may be bounded by the amount of fast memory (e.g. SRAM) available to store the references to actual content stored in a slow memory.

In what follows we begin by considering a single cache and then extend the analysis to a network of caches.

3.2.2.2.1 Single cache analysis

To evaluate the effect of the loss on a single cache, we used a simple MATLAB simulator to implement the model shown in Figure 29, replacing the verify buffer with a random (Bernoulli) dropper. In our analysis we generated requests using an independent reference model (i.r.m.) [40][41]. The sequence of lost items was generated using probabilistic sampling. In these conditions, the presence of a lossy verify buffer does not decrease the average cache hit probability. On the contrary, losses may sometimes lead to an *increase* in the average cache hit probability ¹⁵.

In the independent reference model, requests for data items occur in an infinite sequence, in which the probability that the requested item has index i is regulated by the well-known Zipf law. Obviously, this simple model cannot account for so called *temporal locality*, i.e. the fact that requests for a specific item may be grouped in time. However, in a hierarchy of caches,

¹⁵ We note that other literature works (e.g. [31]) use controlled probabilistic operations to improve caching performance. But in our case these operations are non controllable and forced by exogenous aspects, i.e. the overflow of the verification process.

much of this temporal locality is absorbed by the leaf caches [45]. As a result, the temporal locality experienced by the core nodes of an ICN is rather limited.

As far as concerns our decision to use probabilistic sampling, it is well known from traditional (e.g., NetFlow) trace sampling, that under reasonable assumptions (large volumes of traffic, appropriate interweaving of packets, low sampling rates) deterministic sampling of data items is practically equivalent to probabilistic sampling [47]. Thus, from a performance point of view, a system in which the verification process deterministically accepts one item every T milliseconds is very similar to a system where the accepted items are randomly chosen, with the same acceptance probability.

Figure 30 reports the average cache hit probability h_t as a function of loss (p_l), assuming a cache with a capacity equal to 1% of the size of the data set. We consider a data set formed of 10^5 items, whose popularity follows a Zipf distribution with slope $\alpha=0.75$. Items are ranked according to their popularity: the lower the rank, the higher the popularity. We observe that with increasing loss probability, cache hit probability increases.

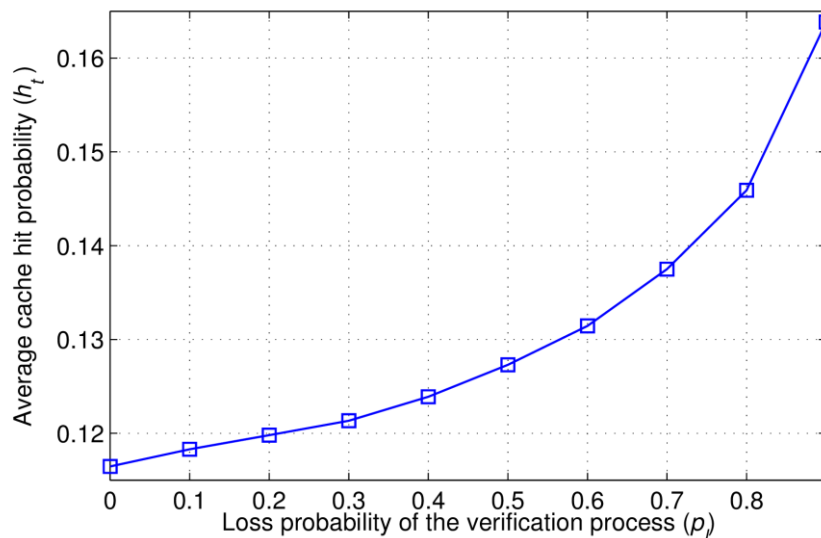


Figure 30 – Average cache hit probability versus loss

The reason behind this behaviour is that the loss yields a reduction of the number of requests that produce updates of cache status, since some of them are lost by the verify buffer. However, *the reduction is unequal among data items*. Un-popular items have a low cache hit probability, so it *often* happens that a request for an unpopular item has to gain admission in the lossy verify buffer. Conversely, popular items have a high cache hit probability, so a request of a popular item *rarely* has to gain admission in the lossy verify buffer. As a consequence, although all items experience reduction of their possibility of updating the cache, this reduction is most severe for un-popular items. With lossy verification, popular items remain in the cache for a longer time and obtain a greater cache hit probability, leading to a *global* increase of the cache-hit probability.

To support this conclusion we observe that the events actually updating the status of the LRU cache are: *cache hit* and *cache insert* events (Figure 29). In the presence of loss, update events are triggered only by a *subset* of requests. Conversely, without loss, every request leads to a cache update. Therefore the loss leads to a *reduction* of the cache update rate

Accordingly, let us define:

- $\lambda(i)$ as the *request rate* of the i th item (chunk)
- $\lambda_u(i)$ as the *caching update events rate* for the i th item, i.e. the aggregate rate of the events of cache hit and cache insert for the i th item
- $q(i)$ as the *popularity* of the i th item, i.e. $q(i) = \lambda(i) / \sum_j \lambda(j)$
- $h(i)$ the cache hit probability for the i th item
- h_t the average cache hit probability, i.e. $h_t = \sum_i q(i)h(i)$

Using the model shown in Figure 29 the relationship between the update rate $\lambda_u(i)$ and the request rate $\lambda(i)$ can be expressed as:

$$\begin{aligned}\lambda_u(i) &= \lambda(i)h(i) + \lambda(i)(1 - h(i))(1 - p_l) \\ &= \lambda(i)(h(i) + (1 - h(i))(1 - p_l))\end{aligned}$$

In the presence of a loss probability p_l , the update rate $\lambda_u(i)$ is equal to the request rate $\lambda(i)$, multiplied by the *reducing* factor $(1 - p_l)(1 - h(i))$. As a consequence, any increase in p_l leads to a reduction $\lambda_u(i)$ that increases with the index i of the item.

With two items i and j , and $i > j$, the more popular item j has a cache hit probability $h(j)$ greater than $h(i)$. Thus, the multiplier factor $(1 - p_l)(1 - h(i))$ is smaller (i.e. worse) for i than for j . Figure 31 shows $\lambda_u(i) : \lambda(i)$ for the different items, for $p_l = 0.8$. We see that $\lambda_u(i) : \lambda(i)$ decreases with popularity (i.e. higher rank).

Figure 32 shows cache hit probability as a function of item rank with a loss of 0.8 and with no loss. We observe that the presence of loss increases the cache hit probability for popular (low ranking) items. This is a consequence of the unequal reduction in update rates shown in Figure 31. Recall that, under the i.r.m. assumption, the sum of the cache hit probabilities $h(i)$ is equal to the cache capacity C [40][41]. In other words, loss shifts the “energy” C of distribution $h(i)$ towards lower indexes, where values of $q(i)$ are higher. This leads to an increase in the average cache hit probability $h_t = \sum_i q(i)h(i)$.

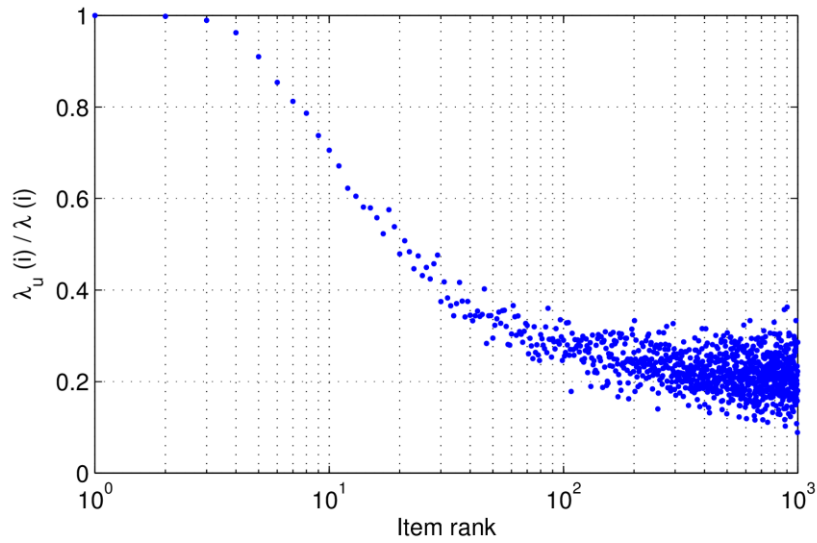


Figure 31 – Ratio of cache update rate λ_u to request rate versus item rank, assuming a loss probability of 0.8

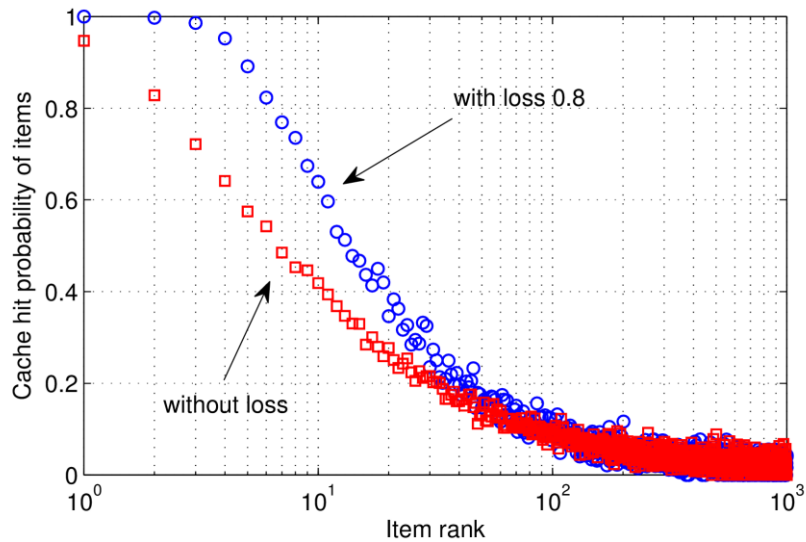


Figure 32 – Cache hit probability versus item rank

3.2.2.2.2 Cache network analysis

In this section, we use the ccnSim simulator [38] to evaluate the impact of lossy-caching in a network (e.g. the CONET). As in our previous analysis we assume that item popularity follows a Zipf distribution with slope $\alpha=0.75$. In this case, however, we simulate an actual signature verify buffer of length 100 using a signature verification time of 0.29 ms, i.e. the value for the “self-certifying with ECDSA combination” (3b) shown in Table 6.

The analysis is based on the GEANT network shown in Figure 33. The network consists of 22 nodes (see [42] for more detail). Node 0 is the repository for all content. On average, content is segmented into 3 chunks (4kB per chunk). Each node has a LRU cache with a capacity

equivalent to 1% of the size of the whole set of available chunks. Content requests are generated using an exponential negative distribution, and uniformly distributed among nodes.

Figure 33 shows the aggregate rate of chunks exchanged on all links of the network normalized by content request frequency, with and without signature checks. With signature checking, the increase in the request rate leads to a small reduction of traffic, and increases the loss on the verify buffer, making caching more effective.

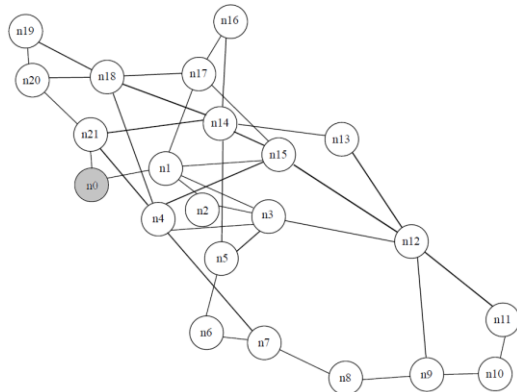


Figure 33 – GEANT topology

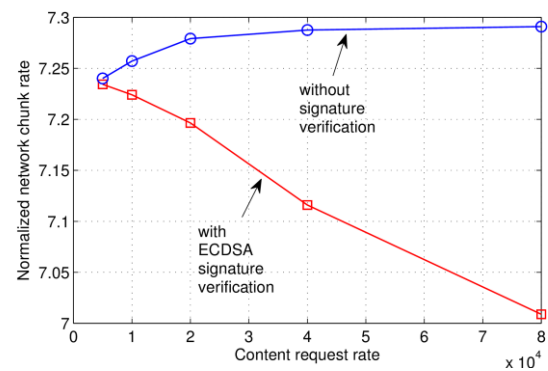


Figure 34 – Normalized network chunk rate versus content request rate

3.2.2.3 Conclusions

In this section, we have discussed different combinations of naming and signature schemes, and analysed the impact of signature checking on cache hit probability. Our most notable finding relates to the impossibility of caching all possible content at line rate when digital signatures are in use. We find that cache losses have no negative impact on the performance of a LRU cache and that they can sometimes lead to increased performance. This demonstrates that speed of signature verification is not a significant criticality for an ICN.

This result holds for request streams with no temporal locality. In theory, strong temporal locality may reduce cache hit probabilities leading to slower cache updates and slower adaptation to temporary changes in popularity. However, in practical terms, the most significant cache losses are in the network core, where temporal locality is strongly reduced [45].

3.2.3 Alternative routing protocols

During the last phases of the project, we used theoretical and simulation approaches to study alternatives to the routing protocols proposed in D5.3, paragraph 4.2.4.2.

We began by reviewing the literature on compact-routing algorithms [52][51], as well as studies of other distributed models developed by researchers in bio-inspired software engineering [48] and complex systems [50].

Inspired by these results, we designed a routing algorithm based on the idea that each node holds a partial view of the network, and that when it has to deliver a packet to an unknown destination, it forwards it to the highest degree node it knows, i.e. the node most likely to know a short path to the destination. In this way, nodes are not forced to retain a full view of the network in the FIBs, which can thus be relatively small. Each node's partial view of the network, expressed in its FIB, is represented by a collection of the shortest paths to each known node. Each node grows its FIB incrementally, by periodically exchanging information with its neighbours, and constantly computing shortest paths to every new node it sees, building on the information it has received.

The key point is that nodes do not exchange complete, well-formed FIB entries, but rather a random selection of "genomes" each containing the information necessary to compute a new shortest path. *Genomes* are held in a table separate from the FIB. They are just segments of network. Each consists of a valid list of connected hops that can be travelled in sequence inside the network. Genomes are uncorrelated, and unbounded in length. They represent a somewhat redundant "sea of information". This is where Dijkstra algorithm running locally fish out shortest paths.

Whenever a node gives a genome to a neighbour, the receiving node adds itself at one end (or possibly both ends) of the segments it has received, provided it is the next-hop for terminus. Thus every time nodes exchange segments, they grow in size by one hop. It is this exchange that adds new information to the segments. In this way, the exchange of information among nodes is the only source of information about network topology, inside the system.

Nodes keep all segments they receive in their genome table. This allows them to sample regions of the network that may be very far away. Until they have done so, these regions will not appear in the node's FIB: the Dijkstra algorithm cannot take account of unconnected/isolated islands. As soon as a node obtains information linking a previously isolated region to known, connected entries, the nodes for the path appear in the FIB.

3.2.3.1 Motivation and background

We developed a custom simulation environment to investigate how routing protocols other than BGP, or BGP-like schemes affect the size of FIBs in the nodes of networks with large-scale Internet-like topologies.

It is well known [52] that, on networks with Internet-like topologies:

- Routing tables for topology unaware flat identifiers cannot scale better than polynomially, with a lower bound of $\Omega(\sqrt{n^{1/k}})$ for stretch $2k-1$ ($k=1,2,\dots$);
- The number of routing signalling messages per topology change cannot grow slower than linearly.

If we want protocols that are scalable terms in terms of routing table size and communications overhead (i.e. protocols that scale logarithmically), we need new ideas.

It is well-known that humans can efficiently route messages through social networks without having a full view of the network topology [53]. This is a good starting point for new proposals.

Epidemic (or gossip) algorithms are a class of algorithms that share a number of features in common even when they are used for very different purposes. Their distinctive features [49] are that they rely on local information, work through “rounds” of information exchange, and have a bounded ability to transmit information. They are also very simple. The authors of [49] propose to apply these ideas to new "gossip-friendly" application areas, including network packet routing.

Our own implementation tries to adhere as far as possible to these basic ideas. Its main characteristics can be defined as follows.

1. Peers are selected randomly
2. Peer only have access to local information
3. Transmission of information is round-based (periodic)
4. The quantity of information transmitted and processed in any one round is limited
5. All peers run the same algorithm

Traditional implementations of distributed routing algorithms begin with desirable properties and try to engineer a design that displays these properties. By contrast, gossip protocols are inherently probabilistic and rely solely on local knowledge, often displaying unexpected emergent global properties that were not considered in the design phase. In our case, we implemented a network with nodes that forwarded packets with an unknown destination to the most densely connected node of which they were aware. What we wanted to know was whether it was possible to design a routing algorithm with flat identifiers with the emergent global-scale property of forwarding packets along the shortest path to destination.

Our goal was thus to discover (in order of increasing difficulty):

- Whether a gossiping approach can work
- Whether it can perform with near-1 stretch and avoid loops
- Whether it is possible to use routing tables that do not contain the whole topology
- Whether such an approach is as scalable as the best alternatives.

3.2.3.2 Algorithm for degree-driven epidemic routing

SYSTEM MODEL

Each node has a complete view of its local set of physical neighbours and knows its own degree.

Each node has a partial local view GEN (for "genome") of network connectivity, in terms of segments: each element of the GEN table is composed of exactly one path segment. A

segment is a list of connected nodes and their accompanying degrees and can be as small as a single node.

Each node periodically exchanges a random selection of its GEN entries with randomly chosen neighbouring nodes. The number of entries embedded within each signalling message is set to a maximum constant M characteristic of the protocol.

Each node also hosts a local partial view of network connectivity in the form of an organized FIB (Forward Information Base). This is computed by refining and aggregating information out of GEN. FIB entries are {destination_node, shortest_path_to_it} couples. There is one entry for each known remote node reachable from the local node. FIB entries are **ranked according to the destination_nodes degree**. Each node runs a standard Dijkstra algorithm to compute shortest paths.

Forwarding is based on source routing: if the node knows the destination of the packet, the entire path to destination is inserted in the packet. If it does not know the destination, it inserts the path to the gateway node with the highest rank in the FIB (i.e. the node with the highest degree).

Each packet carries state about the path it has travelled so far. The forwarding algorithm uses this information to avoid loops, forcing the packet to explore new portions of the network during its journey to destination.

This mechanism, combined with preference for well-connected intermediate nodes, ensures low-stretch, loop-free performance. Admittedly, this approach departs from the ideal of using only local information. However, we are not trying to use an epidemic protocol for forwarding. The role of gossiping is to slowly build a view of the network. Forwarding *uses* this view.

Certainly, this is not the only argument against piggybacking routing information inside data packets. For instance it is not clear that router hardware will be powerful enough to parse packet headers at line speed. This issue lies outside the scope of this initial study. But as has been shown many times in the past, refusing to explore new mechanisms because of hardware limitations, is an error to be avoided.

DESCRIPTION OF STATES

As in other state-of-the-art gossip protocols, the router's topology construction mechanism has an **active** and a **passive** state. The active state is triggered periodically. The passive state is triggered whenever a neighbour receives a signalling message. An additional **forward** state, is triggered when the node has to forward a packet to a destination. FORWARD has to run at line speed. ACTIVE and PASSIVE operations can be carried out in the background with lower priority.

The node starts with a genome (GEN) that contains only itself. At each (**ACTIVE**) periodic round, the local node LN:

1. Randomly selects a neighbour node NN to communicate with.
2. Randomly selects M entries from its GEN and pushes them to NN. M is a fixed constant of the protocol.

When local node LN receives (**PASSIVE**) a remote genome sample RGEN through neighbour NN:

1. It randomly selects M entries from its GEN and pushes them to NN.
2. It then enriches its "genetic material". For each segment in RGEN it behaves as follows:
 - 2.1) if one of the segment's extremity nodes (first or last) is NN, the node connects to the segment by adding its ID at the tail or head, creating a longer path on each round.
 - 2.2) It merges the received RGEN with its own GEN
3. The FIB is updated:
 - 3.1) Shortest paths are computed from LN to each reachable node present in GEN
 - 3.2) Shortest paths are inserted in FIB
 - 3.3) The FIB is sorted in descending order of degree

When a packet has to be routed (**FORWARD**) to node X from local node LN:

1. If destination X is in the FIB of LN, and SR_to_X is shorter than the route already present, SR_to_X is assigned to the packet.
2. If the packet already has a source route to X, it is routed to its next hop
3. If the packet does not have a source route to X, the system chooses the first destination (=the highest degree destination) in the sorted FIB; if SR_to_GW does not loop with PATH (the path the packet has travelled so far), the system assigns SR_to_GW to the packet
4. If all possible routes to other nodes loop with PATH, the system chooses the route that maximises network exploration, i.e. the path where there is the greatest number of hops to the loop.

3.2.3.3 Preliminary results

Simulations were performed on Internet-like topologies generated by the SIMROT-top tool (<http://simula.no/departement/netsys/software>). The simulations used topologies with sizes ranging from 200 to 9000 nodes. Considering that day-current BGP routing algorithms run on top of networks with 45000 AS, this represents a realistic (though scaled down) scenario.

CONVERGENCE

We measure convergence performance in terms of stretch, defined as the ratio between the number of hops travelled and the corresponding optimal shortest path calculated by a Dijkstra algorithm that knows the whole topology.

As more and more information is exchanged the performance of the algorithm improves. After a very brief transient, average stretch decreases **monotonically** and **linearly**, showing no slowing down as time passes (see next figure).

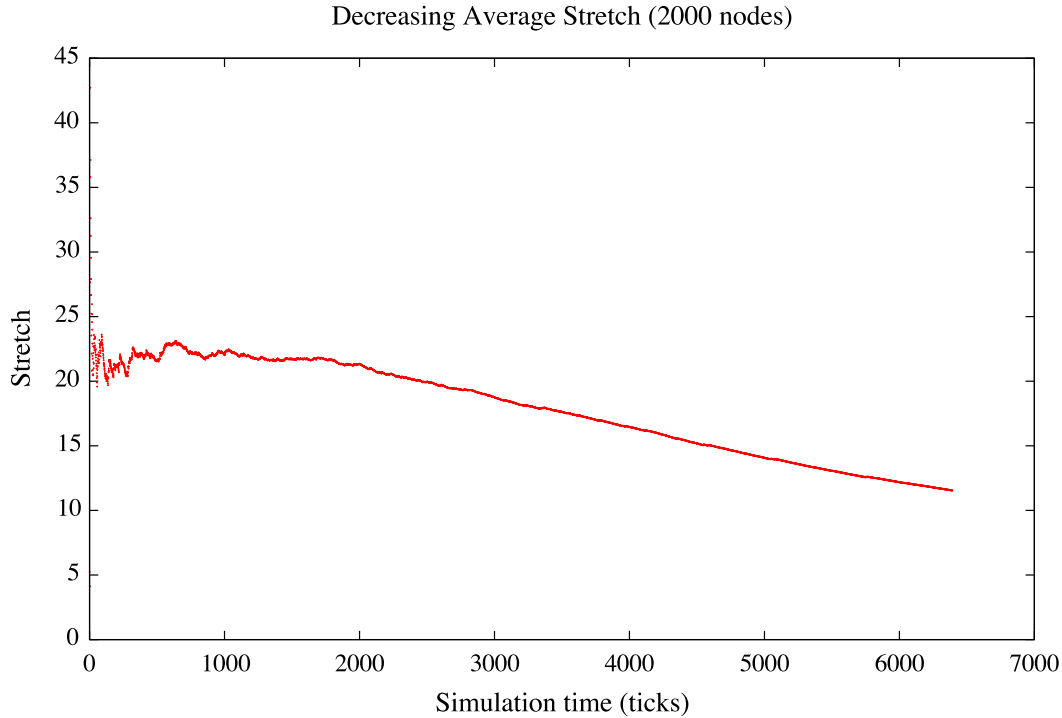


Figure 35 – Behaviour of average stretch in a network with 2000 nodes

At the beginning of the simulation nodes have absolutely no information about the network topology. This explains why the initial stretch is so high.

Loop-free

The following Table shows the number of *undelivered packets* for simulations using topologies of different sizes. A packet is considered to be undelivered if it travels a path longer than N hops, where N is the total number of nodes in the network. The table also shows the maximum stretch value observed during the simulation and the diameter of the networks. This shows that in the worst case the unlucky packet has travelled a number of hops equal to $worst = max_stretch * diameter$.

N	diameter	max_stretch	undelivered packets
200	5	173.6	0
2000	5	327.5	0
3000	5	729.5	0
4000	5	515.3	0
5000	5	1009.7	0

Table 7 - Number of undelivered packets for simulations using topologies of different sizes

The graph below shows the percentage of forwarded packets that travelled a number of hops larger than the network diameter (the length of the longest of the optimal paths between any possible source and destination). The graph is offset to show the behaviour of the network once it has converged. The moment of convergence is defined by the time when average stretch falls below a pre-determined threshold, in this case 1.05. In the graph this occurs at time 4921. From this time on the algorithm is dedicated exclusively to forwarding. The packet counter of packets starts at the moment of convergence.

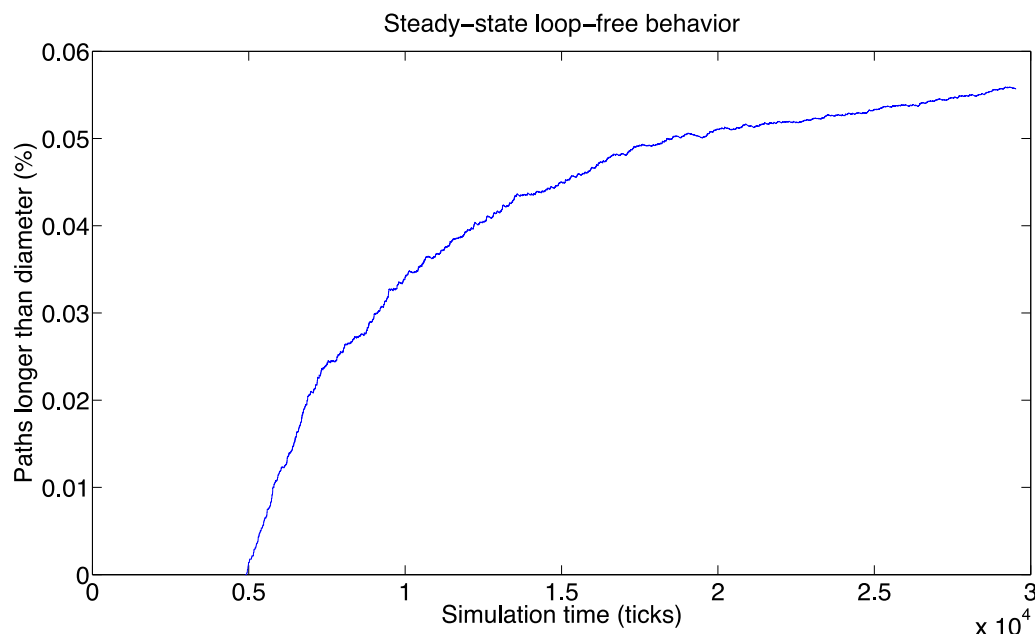


Figure 36 – Paths longer than diameter at steady state in a network with 2000 nodes

The percentage of packets that travel paths longer than the network diameter is approximately 5%. This demonstrates that the algorithm is able to perform without introducing loops and significant deviations in path lengths.

Stretch

The graph below, for a network with 8000 nodes, shows the stretch of each delivered packet until the moment of convergence.

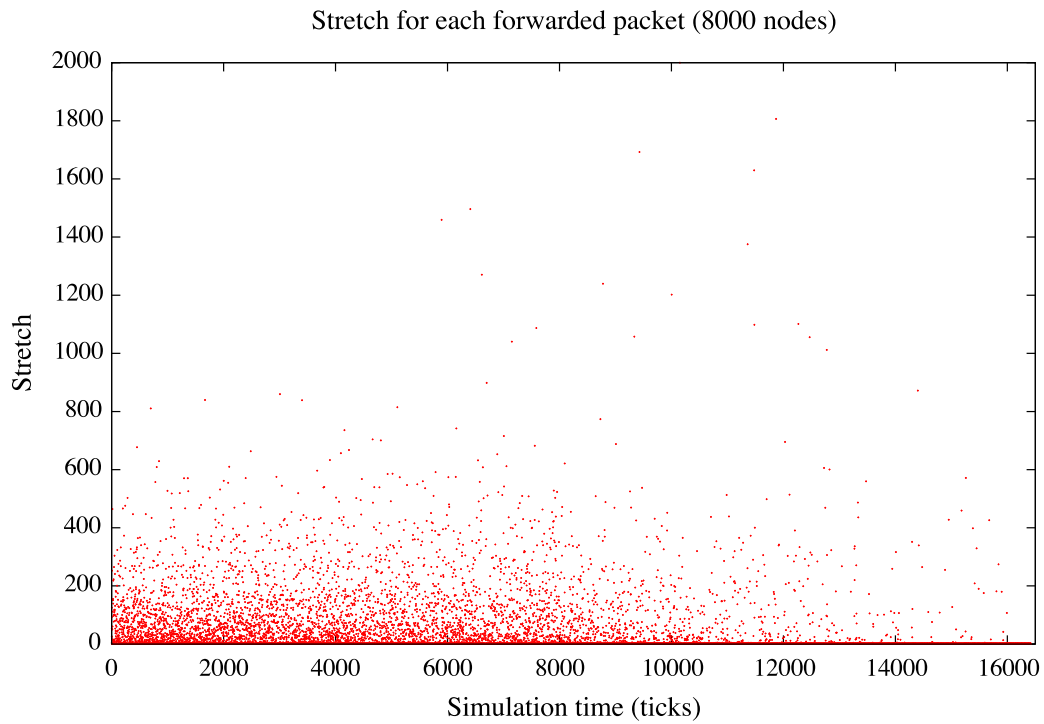


Figure 37 – Stretch for each packet, until convergence, in a network with 8000 nodes

As the simulation proceeds the number of packets with high stretch falls continuously.

The graph below zooms in on the region bounded by times 14000 and 16500 on the x-axis , stretch values between 1 to 3 on the y-axis.

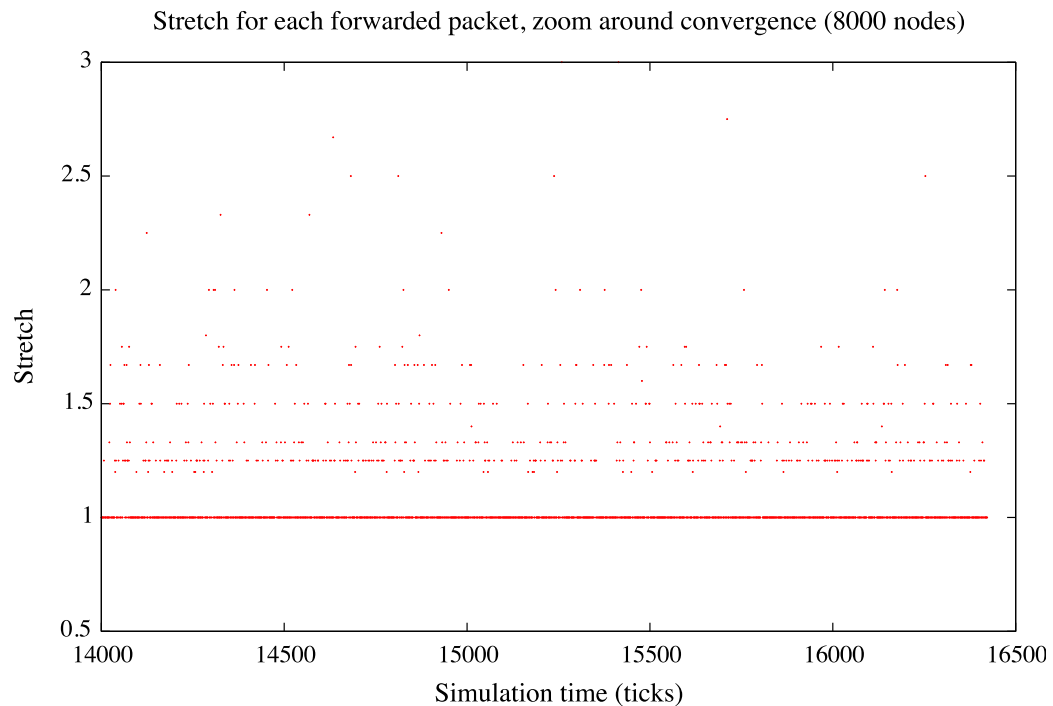


Figure 38 – Stretch for individual packets, around the moment of convergence, in a network with 8000 nodes

The majority of paths have a stretch of 1. Given that path lengths are discrete we observe some clustering. The following graph represents **average stretch**, computed over a moving window of 100 samples.

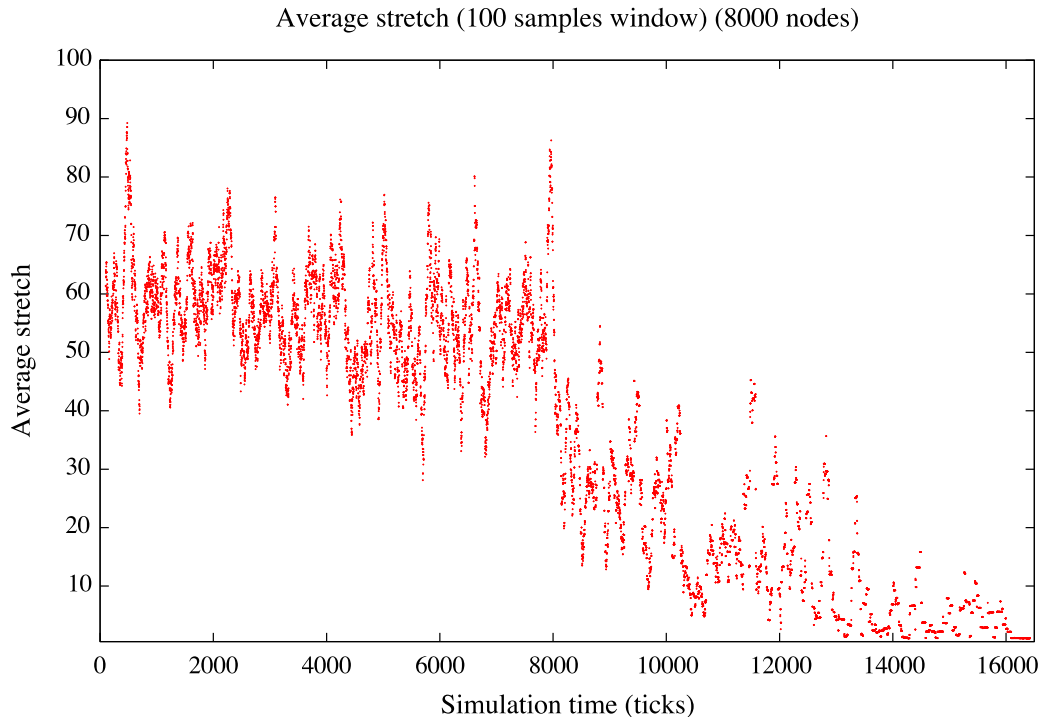


Figure 39 – Average stretch, sliding window of 100 samples, in a network with 8000 nodes

It is clear from the figure that average stretch decreases quickly. Once convergence has been achieved it maintains a value of approximately 1.

FIB Size

The following graph shows the ratio between the number of entries in the FIB (average for all FIBS in the network) and the total number of nodes. The ratio is averaged over five runs for each topology. If the ratio is less than 1, nodes do not need to have a full view of the network, i.e. to hold an entry for each possible destination.

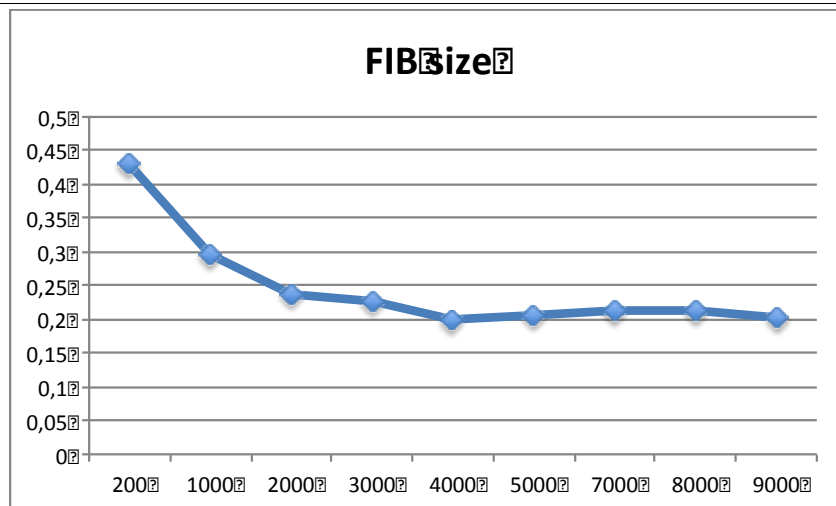


Figure 40 – Average FIB size for various networks

The figure shows that after convergence, FIBs only need to hold 20% of the whole network prefixes.

With large networks, the ratio of FIB to network size is constant. This implies linear scalability: the best-case in theoretical studies of compact-routing. With smaller networks performance is not quite so good.

3.2.3.4 Conclusions

The results of the network tests indicate that it is possible to construct a distributed, epidemic routing protocol, which:

- Only knows randomly selected, self-discovered sub-portions of the overall network
- Exchanges a limited amount of information with its physical neighbours on each round
- Favours well-connected, high-degree nodes when forwarding information
- Considers the history of the packet, when making forwarding decisions
- Uses a source-routing mechanisms.

The protocol is able to:

- Significantly reduce the size of FIBs, compared to current algorithms
- Achieve stretch-1 performance
- Achieve loop-free operation

With sufficiently powerful hardware it may be possible to achieve a trade-off between computing power on nodes and the size of the FIB. The scalability of our approach matches the theoretical lower bound in compact-routing theories. In future work, we will investigate scalability in terms of signalling messages/overhead.

3.2.4 CONET Support for Peer-to-Peer Content Sharing in fixed networks

According to recent reports [54] peer-to-peer (P2P) and in particular, BitTorrent traffic account for 31.8% upstream traffic and 12.1% of downstream traffic over Europe's fixed access network (14.9% of total traffic). Given the huge share of network capacity dedicated to supporting P2P applications, it is curious that research in information-centric networks (ICN) focuses mainly on client-server communications.

An ICN such as CONET is in some ways similar to a P2P system, in the sense that it provides its own caching functionality, allowing network routers to serve requests from their local caches, without having to forward the requests to the server.

Details on the way P2P content sharing is supported by CONET are provided in deliverable D5.3 [55]. Here, we focus on the performance of the proposed protocol. Our evaluation leads to some interesting conclusions concerning the sustainability of P2P applications running over information-centric networks.

Our analytical framework and performance evaluation can be found in [56].

3.2.5 CONET Support for Adaptive Video Streaming in fixed networks

Routing in information-centric networks, and in CONET, is performed on a per-name basis. HTTP-based video streaming (e.g. MPEG-DASH, HTTP Live Streaming, Microsoft Smooth Streaming, etc) involves splitting the video into segments of a known duration, each one of which is requested by the client (video player). As explained in Section 3.2.1.2, each video segment is accessible through the CONET repository by its name, using the CCNx GET operation.

In an adaptive streaming scenario, we have a single video encoded in different qualities, so that the player can choose the best available quality given local conditions (e.g. available bandwidth). The CONET repository contains versions of the video in all possible qualities each with a different name. As a result, network provider caches may contain video segments of the same part of the video, each with its own specific quality and name. Thus, a client requesting segment X with quality A may have to wait for the request to reach the video server, even if a router has already cached the same segment with quality B. This will satisfy neither the client, who wants to achieve the fastest possible round trip times, nor the network provider, who wants to maximize traffic over its own transport network and minimize its use of the public Internet. In other words, there is a risk that adaptive streaming over information-centric network, would cancel out its advantages over non-adaptive solutions.

4 Bibliography

- [1] Bob E. Hayes, *Measuring Customer Satisfaction: Survey Design, Use, and Statistical Analysis Methods*, 2nd Edition, ASQ Quality Press, Milwaukee, Wisconsin, 1998.
- [2] Terry G. Vavra, *Improving Your Measurement of Customer Satisfaction: A Guide to Creating, Conducting, Analyzing, and Reporting Customer Satisfaction Measurement*, ASQ Quality Press, Milwaukee, Wisconsin, 1997.
- [3] Stephen Kan, *metrics and Models in Software Quality Engineering*, 2nd Ed., Addison-Wesley Professional, 2003.
- [4] D. Cheriton, M. Gritter, "TRIAD: a scalable deployable NAT-based internet architecture", Technical Report (2000)
- [5] T. Koponen, M. Chawla, B.G. Chun, et al. "A data-oriented (and beyond) network architecture", ACM SIGCOMM 2007
- [6] V. Jacobson, D. K. Smetters, J. D. Thornton et al., "Networking named content", ACM CoNEXT 2009
- [7] A. Detti, N. Blefari-Melazzi, S. Salsano, M. Pomposini, "CONET: A Content Centric Inter-Networking Architecture", ACM SIGCOMM ICN Workshop 2011
- [8] CCNx project web site: www.ccnx.org
- [9] CONNECT project website: www.anr-connect.org
- [10] CONVERGENCE website: www.ict-convergence.eu
- [11] Named Data Networking project website: www.named-data.net/
- [12] PURSUIT project website: www.fp7-pursuit.eu
- [13] SAIL project website: www.sail-project.eu
- [14] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles", ACM MMSys 2011
- [15] WiFi Direct, <http://www.wi-fi.org/discover-and-learn/wi-fi-direct>
- [16] A.Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, A. Bragagnini, "Offloading cellular networks with Information-Centric Networking: the case of video streaming", IEEE WoWMoM 2012
- [17] B. Han, N. Choi, T. Kwon, Y. Choi, "AMVS-NDN: Adaptive Mobile Video Streaming and Sharing in Wireless Named Data Networking" to appear in IEEE NOMEN 2013, available at http://mmlab.snu.ac.kr/~dobby/assets/2013_NOMEN.pdf
- [18] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, A. Markopoulou, "Microcast: Cooperative Video Streaming on Smartphones", ACM MobiSys 2012
- [19] http://netgroup.uniroma2.it/Andrea_Detti/ICNvideo-live-DASH
- [20] J. J. D. Mol, A. Bakker, J. A. Pouwelse, D. H. J. Epema, H. J. Sips, "The Design and Deployment of a BitTorrent Live Video Streaming Solution," IEEE International Symposium on Multimedia, 2009
- [21] ITEC – Dynamic Adaptive Streaming over HTTP, <http://www-itec.aau.at/dash/>
- [22] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, N. Blefari-Melazzi, "Transport-Layer Issues in Information Centric Networks", ACM SIGCOMM ICN Workshop 2012
- [23] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, "Supporting the Web with an Information Centric Network that Routes by Name", Elsevier Computer Networks, vol. 56, Issue 17, p. 3705–3722

- [24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, Volume 38, No. 2, pp.:69–74. April 2008
- [25] N. Blefari-Melazzi, A. Detti, G. Morabito, S. Salsano, L. Veltri: "Supporting Information-Centric Functionality in Software Defined Networks", IEEE ICC 2012, Software Defined Networks Workshop, June 10-15, 2012 Ottawa, Canada
- [26] Bo Li and Hao Yin, "Peer-to-peer live video streaming on the internet: issues, existing approaches, and challenges", IEEE Communication Magazine, June 2007
- [27] Xiaoqing Zhu, Agrawal, P. ; Singh, J.P. ; Alpcan, T. ; Girod, B. "Distributed Rate Allocation Policies for Multihomed Video Streaming Over Heterogeneous Access Networks" IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 11, NO. 4, JUNE 2009
- [28] Derek Kulinski and Jeff Burke, "NDN Video: Live and Prerecorded Streaming over NDN", NDN Technical Report NDN-0007, September, 2012. (<http://www.named-data.net/techreport/TR007-streaming.pdf>)
- [29] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, J. Wilcox, "Information-Centric Networking: Seeing the Forest for the Trees", 10th ACM Workshop on Hot Topics in Networks, HotNets 2011.
- [30] N. Blefari Melazzi, A. Detti, M. Pomposini: "Scalability Measurements in an Information-Centric Network", in "Measurement-based experimental research: methodology, experiments and tools", Springer Lecture Notes in Computer Science (LNCS), vol. 7586, 2012.
- [31] I. Psaras, W. K. Chai, G. Pavlou, "Probabilistic in-network caching for information-centric networks", ACM SIGCOMM Workshop on Information-Centric Networking, ICN 2012
- [32] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, " Naming in content-oriented architectures", ACM SIGCOMM workshop on Information-centric networking, ICN 2011
- [33] D. K. Smetters, V. Jacobson, "Securing network content", PARC TR-2009-1; 2009 October
- [34] D. Galindo, F. D. Garcia, "A Schnorr-Like Lightweight Identity-Based Signature Scheme", in International Conference on Cryptology in Africa: Progress in Cryptology (AFRICACRYPT '09), Bart Preneel (Ed.). Springer-Verlag, Berlin, Heidelberg, 135-148.
- [35] Adi Shamir, "Identity-Based Cryptosystems and Signature Schemes", Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 7:47--53, 1984
- [36] D. Johnson and A. Menezes, "The elliptic curve digital signature algorithm (ECDSA)", Technical Report CORR 99-34, University of Waterloo, Canada, February 24 2000
- [37] Rivest R., A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the ACM 21 (2): 120–126, 1978
- [38] ccnSim: scalable chunk-level simulator of Content Centric Networks, <http://perso.telecom-paristech.fr/~drossi/index.php?n=Software.ccnSim>
- [39] Standards for efficient cryptography, "SEC 2: Recommended Elliptic Curve Domain Parameters", Certicom Research available at www.secg.org/collateral/sec2_final.pdf

- [40] C. Fricker, P. Robert, J. Roberts, “A Versatile and Accurate Approximation for LRU Cache Performance”, International Teletraffic Congress (ITC 24), September 4-7, 2012, Krakow, Poland
- [41] H. Che, Y. Tung, and Z. Wang, “Hierarchical web caching systems: modeling, design and experimental results”, IEEE JSAC, 20(7), 2002.
- [42] D. Rossi, G. Rossini, “Caching performance of content centric networks under multi-path routing (and more)”, Tech. Report, Telecom ParisTech 2011, www.enst.fr/~drossi/paper/rossi11ccn-techrep1.pdf
- [43] S. Arianfar, P. Nikander, and J. Ott. “On content-centric router design and implications”, in ReArch Workshop, volume 9, page 5. ACM, 2010.
- [44] A. Kate and I. Goldberg, “Distributed private-key generators for identity-based cryptography”, International conference on Security and Cryptography for Networks (SCN'10), Springer-Verlag, Berlin, Heidelberg, 2010.
- [45] R. Fonseca, V. Almeida, M. Crovella, and B. Abrahao, “On the intrinsic locality properties of web reference streams,” in IEEE INFOCOM, 2003
- [46] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, “DoS and DDoS in Named-Data Networking,” ArXiv, abs/1208.0952, 2012.
- [47] Y. Chabchoub, C. Fricker, F. Guillemin, and P. Robert, “Deterministic versus probabilistic packet sampling in the internet,” in 20th international teletraffic conference, ITC20, 2007.
- [48] Mark Jelasity, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. 2004. The peer sampling service: experimental evaluation of unstructured gossip-based implementations. In Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware (Middleware '04). Springer-Verlag New York, Inc., New York, NY, USA, 79-98.
- [49] Paolo Costa, Vincent Gramoli, Mark Jelasity, Gian Paolo Jesi, Erwan Le Merrer, Alberto Montresor, and Leonardo Querzoni. 2007. Exploring the interdisciplinary connections of gossip-based systems. SIGOPS Oper. Syst. Rev. 41, 5 (October 2007), 51-60.
- [50] Melanie Mitchell. 2006. Field review: Complex systems: Network thinking. Artif. Intell. 170, 18 (December 2006), 1194-1212
- [51] How Small Are Building Blocks of Complex Networks. Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Marian Boguna, and Dmitri Krioukov. http://www.caida.org/publications/papers/2009/small_blocks_complex_nets/small_blocks_complex_nets.pdf
- [52] Dmitri Krioukov, Kevin Fall, and Arthur Brady. 2007. On compact routing for the internet. SIGCOMM Comput. Commun. Rev. 37, 3 (July 2007), 41-52.
- [53] S. Milgram. The small world problem. Psychology Today, 1:61–67, 1967.
- [54] Sandvine – Intelligent Broadband Networks. Global Internet Phenomena Report – 2H 2012. Technical Report.
- [55] A. –C. Anadiotis et al. D5.3. Final protocol architecture. CONVERGENCE Project Deliverable. Technical Report. Oct. 2012.
- [56] A. –C. G. Anadiotis, L. Galluccio, G. Morabito, C. Z. Patrikakis, I. S. Venieris. P2P over information-centric networks: analysis of in-network caching performance. IEEE Communications Letters (Submitted for publication – Under review) Electronic version available at: http://research.icbnet.ntua.gr/paper_repo/P2PoverICN.pdf

-
- [57] A. –C. G. Anadiotis, I. S. Venieris, C. Z. Patrikakis. On Enhancing Adaptive Video Streaming Performance over Information Centric Networks. Computer Networks (Submitted for Publication – Under review) Electronic version available at: http://research.icbnet.ntua.gr/paper_repo/COMNET-S-13-00016.pdf