Sixth Framework Programme

Priority IST-2002-2.3.1.4

Mobile and Wireless Systems beyond 3G

Project 507134

WWI Ambient Networks

D 3.3 Connecting Ambient Networks – Final Architecture, Protocol Design and Evaluation

Due date of deliverable: 2005-12-31

Start date of project: 2004-01-01 Duration: 2005-12-31

Revision: 1.0

Document Properties:

Document Number:	IST–2002-507137-AN/WP3/D/3-3
Document Title:	Connecting Ambient Networks – Final Architecture, Protocol Design and Evaluation
Document responsible:	Inge Einar Svinnset and Brynjar Viken, Telenor
Author(s)/editor(s):	Ian Marsh (SICS), Alan Smith (BT), Jorge Andrés Colás (Telefónica I+D), Di Zhou (Siemens AG), Nadeem Akhtar (U Surrey), Rui Campos (INESC Porto), Paulo Mendes (DoCoMo), Carlos Pinho (INESC Porto), Janne Riihijärvi (RWTH Aachen), Louise Burness (BT), Cornelia Kappler (Siemens AG), María Ángeles Callejo Rodríguez (Telefónica I+D), Dan He (U Surrey), Yaning Wang (U Surrey), Philip Eardley (BT), Manuel Ricardo (INESC Porto), José Ruela (INESC Porto), Laura Feeney (SICS), Joachim Hillebrand (DoCoMo), Roger Kalden (Ericsson), Gösta Leijonhufvud (Ericsson), Pekka Koskela (VTT), Tobias Poschwatta (TU Berlin), Morten Schläger (TU Berlin), Petteri Pöyhönen (Nokia) and Ossi Raivio (RWTH Aachen).
Dissemination level[1]:	PU
Security Type:[2]	Public
Status of the Document:	Final
Version	1.0

[1] Dissemination level as defined in the EU Contract:
PU = Public
PP = Distribution limited to 6th FP participants
RE = Distribution to a group specified by the consortium
CO = Confidential, only allowed for members of the consortium

[2] Security Type:
Confidential Internal circulation within project (and Commission project Officer if requested)
Restricted Restricted circulation defined in the document or dissemination level PP
Internal With no confidential content but intended for internal use
Public *Public document*

Table of Contents

1 Executive summary

Ambient Networks is a large-scale collaborative project within the European Union 6th Framework Program, investigating future communications systems beyond today's fixed and 3rd generation mobile standards.

Composition of networks is a core concept of Ambient Networks. Composition is a uniform, dynamic form of network cooperation to achieve control plane interworking and control sharing between networks. Today this is not possible without extensive manual configuration. Furthermore, for each pair of network types, the process is different.

Objective:

This deliverable provides the reader with details of the composition process, the GANS protocol, on QoS in Ambient Networks and on migratability of Ambient Networks. Particularly, the focus of the deliverable is on evaluation of results. We discuss scalability, performance and prototypes of results described in earlier deliverables.

Results:

When Ambient Networks (AN) compose, their control planes, called ACS (Ambient Control Space), coordinate such that, to the outside world, the composed networks can appear as a single uniform Ambient Network. The interface that an Ambient Network presents to the outside is the ANI (Ambient Network Interface). The Ambient Networks communicate among themselves via the ANI using the Generic Ambient Network Signalling (GANS) protocol. An ACS is subdivided in Functional Entities (FEs) that encapsulate management and control functions in a network, e.g. QoS-FE and Congestion Control FE (CC-FE). The Composition FE (C-FE) is indispensable and orchestrates the Composition process.

Work Package 3 "Network Composition and Connectivity" addresses several issues vital to Composition, and hence at the heart of the Ambient Networks project. We described composition and the composition procedure in more detail, including a number of scenarios in which we investigate to what extent existing technology can cover the composition functionality, and where new technology, e.g. GANS, is needed. The GANS protocol is used for negotiating control functionalities, e.g. what address configuration procedure to use, or for the negotiation of SLAs. For such functionality usually no other suitable protocol is available. It is important to emphasize that we do not aim to replace existing protocols. The GANS protocol was also designed in this Work Package.

We show GANS is scalable by means of simulations. We also verified the GANS specification with the implementation of a prototype. Additional activities have included collaboration with other work packages to verify the composition procedure and a joint elaboration of use cases.

As one important part of composition various facets of QoS were investigated. We developed a framework that allows ANs to dynamically control Inter-Network QoS agreements as part of a composition, to achieve seamless QoS between ANs. This included the design of a QoS FE, a QoS application for the GANS protocol and a specification of the QoS-related part of the ANI. Furthermore we developed highly adaptive congestion control mechanisms based on network information sharing to ensure efficient and fair use of network resources in dynamically changing environments.

The QoS framework allows ANs to advertise and negotiate Service Level Specifications and then to monitor the service level. In a scalability analysis we show that the number of non-provider ANs has no impact on the overall network signaling overhead. This happens because for any new customer or customer-provider AN there is only a local exchange of messages with its adjacent neighbour. The message exchange aims to allow the new AN to collect information about the network services (which may have scopes beyond the neighbours networks) offered by its neighbours, and to negotiate a sub-set of required services. However,

the number of available network services may raise some scalability concerns, leading to the need to develop an aggregation mechanism in the second phase of the project. The raise of network services may happen due to policy changes in existing networks, or due to the arrival of new provider networks. Moreover, we have estimated, via analysis and simulations, that the response time to network failure is acceptable. In this context we also discussed several loop-avoidance approaches.

Four new congestion-control mechanisms complement the QoS framework. These mechanisms address different network conditions: Checkpoint nodes can be deployed at the border of a network in order to protect it from congestion. TCP-FEC and Utility-fair Congestion Control work end-to-end as TCP but employ novel methods to detect and handle congestion mechanisms. Finally, Contention Resolving Congestion Control is suited for wireless-multihop environment, where it balances overall network capacity utilization and energy savings due to mobile devices going into sleep-mode. Here we report the results of detailed numerical investigations on the effectiveness and overhead of the different mechanisms. We conclude that in future Ambient networking scenarios it is important to strengthen the role of the network in congestion control, e.g. by introducing Checkpoint nodes. The enhanced end-to-end congestion control schemes proved to be more efficient than normal TCP, particularly in situations of high network load.

As another alternative for streaming real-time applications which cannot work with TCP-based congestion control, we studied Datagram Congestion Control Protocol (DCCP) as a transport protocol. We evaluated how flows using TCP and DCCP can coexist and share the bandwidth. Furthermore we evaluated different schemes that have been proposed for differentiating between losses due to congestion and wireless losses.

Finally, we dealt with the integration of legacy networks as well as migrating today's networks towards Ambient Networks. We have derived a number of migration principles that should be adhered to in order to facilitate successful migration from existing networks to Ambient Networks. We also identify several migration phases. In this report we give a migration analysis of the results of several work packages. We conclude that the Ambient design, to the extent specified currently, mostly adheres to migration principles.

Status and next steps:

We have shown that the results delivered by WP3 on composition and QoS are scalable, performant, implementable and migratable. Work on composition, QoS and migration will be continued in WP-G of Ambient Networks Phase 2.

2 Terminology and Acronyms

2.1 Terminology

Ambient Service Interface (ASI): The ASI interfaces towards service infrastructures and allows applications and services to issue requests to the Ambient Control Space concerning the establishment, maintenance, and termination of end-to-end connection between functional instances connecting to the ASI.

Ambient Resource Interface (ARI): The ARI offers control mechanisms that the ACS can use to manage the resources residing in the Ambient Connectivity. These resources can be routers, switches, but also media transcoders, filters and proxies.

AQE - Ambient Network QoS Entity: Also known as INQA-FE.

Congestion Control: Congestion control is a mechanism that permits un-coordinated entities to inject traffic into a network, such that the network maintains reasonable levels of efficiency, stability and fairness for all. The characteristic that distinguishes congestion control from other network control mechanisms is that it is entirely decentralized and does not rely on explicit coordination.

Congestion Control FE Control Space (CC-FE-CS): The CC-FE-CS resides logically in the ACS and comprises all functions to interact with other Functional entitys and communicates via the ambient network interfaces.

Congestion Control FE Connectivity (CC-FE-CY): The CC-FE-CY resides logically in the ACY and comprises all deployed congestion control mechanisms including legacy protocols such as TCP.

Composition: Composition is a dynamically created cooperation of networks. It is based on a Composition Agreement negotiated between the composing networks. Composition particularly addresses cooperation on the control plane, e.g. regarding mobility or security. Composed networks may appear as a single Ambient Network to the outside, exhibiting a joint ANI.

Composition Procedure: The Composition Procedure specifies a well-defined set of basic steps ANs should follow in order to get composed. Five basic steps are defined: Media Sense, Discovery/Advertisement, Security and Internetworking Establishment, Composition Agreement Negotiation, and Composition Realization.

Composition Type: The Composition type defines how tight two Ambient Networks get composed, i.e., it defines the level (or depth) of cooperation between the composing ANs. The Composition type defines how resources shall be managed and controlled after the ANs get composed. The Composition type is not something ANs aim at before the Composition. Rather, it is something that can be identified after the Composition has been established, i.e., it is the outcome of the Composition.

Control Sharing: Control sharing is a composition type, in which individual ANs exercise joint control over shared resources according to the negotiated Composition Agreement. In this case, a new common/virtual ACS is always required.

Degree of Composition: This term has been replaced by “Type of Composition”.

Destination Endpoint Exploration Protocol (DEEP): The DEEP protocol together with name resolution service(s) is used to resolve an abstract name (e.g. globally unique service name or an E.164 address) into contact address information (e.g. an IP address) based on which a signaling message routing can be done towards the signaling destination.

End-point signalling notification (ECN)*: Signalling by the TCP-end points to convey congestion information. It does not require any support for ECN in the routers. The receiver can detect lost packets if the losses are correctable, and sets the ECN echo bit letting the sender take the appropriate action.

Extended GIMPS (EGIMPS): This term has been replaced by “Extended GIST (EGIST)”.

Extended GIST (EGIST): This term is used to mean the component of GANS Transport Layer Protocol (GTLP) designed based on General Internet Signaling Transport (GIST) developed by the NSIS working group of IETF. EGIST extends GIST to meeting the specific requirements of GANS and is backward compatible to GIST.

Fairness: Congestion control should ensure that all flows receive a fair bandwidth share. This does not imply equal bandwidth share; each flow shall receive a minimum bandwidth share appropriate for efficient operation. To decide on the bandwidth share context information may be considered.

Functional Entity: A Functional Entity (FEs) is a groups of functions. This replaces the former term Functional Area. The term is further explained and motivated in D1.5 [14].

GANS Transport Layer Protocol (GTLP): The name of the GANS transport layer providing generic message transport services for control signaling applications.

GANS Signaling Layer Protocol (GSLP): The name of the GANS signaling application layer.

INQA: Framework that allows ANs to dynamically control Inter-Network QoS Agreements, which are defined by SLSs, and can be monitored based on an agreed SLI. INQA encompassed four operational modules (INQA-A, INQA-N, INQA-M and INQA-AC) and one signaling protocol (INQA-GSLP).

INQA-GSLP: GANS Signaling Layer Protocol aiming to dynamically control bi-lateral Inter-Network QoS Agreements between adjacent ANs.

INQA-A: Module of INQA that implements an SLS Advertisement mechanism

INQA-N: Module of INQA that implements an SLS Negotiation mechanism

INQA-AC: Module of INQA that implements an SLS Admission Control mechanism

INQA-M: Module of INQA that implements an SLS Monitoring mechanism

INQA-FE: Within the Ambient Network Control Space (ACS) the functionality of INQA is implemented in the INQA Functional Entity (INQA-FE), also known as Ambient Network QoS Entity (AQE). The INQA-FE is a component of the QoS-FE, which may include other QoS related functionalities, such as Ambient Network Virtual Pipes for QoS.

Network Composition: Network Composition is a basic, scalable and dynamic form of network cooperation based on a Composition Agreement (CA) negotiated between the composing ANs.

Network Integration: The composition type, in which all involved resources and services are under the common control and therefore a composition results a new common/virtual ACS and constituent ACSs will be hidden from outside.

Network Interworking: The composition type without any common control resulting constituent ANs to preserve their control of own resources and services after a composition. This type of composition involves a new common/virtual ACS only if multi-lateral support is preferred.

NSIS Transport Layer Protocol (NTLP): RFC4080 - Next Steps in Signaling (NSIS): Framework. R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch. June 2005 (http://www.ietf.org/rfc/rfc4080.txt).

Next Steps in Signaling (NSIS): RFC4080 - Next Steps in Signaling (NSIS): Framework. R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch. June 2005 (http://www.ietf.org/rfc/rfc4080.txt).

NSIS Signaling Layer Protocol (NSLP): RFC4080 - Next Steps in Signaling (NSIS): Framework. R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch. June 2005 (http://www.ietf.org/rfc/rfc4080.txt).

Service Level Agreement (SLA): Bilateral contract between two (Ambient) networks, logically adjacent, in which one network is the service provider and the other is the customer. SLAs include certain non-technical guarantees and issues that do not bear directly on packet handling. An SLA is a partially technical document that is determined by network administrators, lawyers, and others, and is communicated via means ordinarily appropriate to that sort of agreement.

Service Level Specification Template (SLS-T): Information model that defines the technical part of an SLA. More specifically, the SLS-T defines the parameters required to allow networks to specify the levels of QoS that certain type of traffic will get. The defined template is independent from QoS provisioning techniques used within each ambient network, such as IntServ, DiffServ, etc.

Service Level Specification (SLS): Instantiation of the SLS template, which means that an SLS encompasses the specification of how certain traffic, meeting certain conditions and arriving from a peer network on a certain interface, will be treated. There is one SLS in each direction. If the destination of the traffic is not in the receiving network, then the traffic will be passed on to another network on the path toward the destination with which a similar SLS exists.

Service Level Indication (SLI): Is an instantiation of the SLS-T, allowing customer-ANs to negotiate the capability to monitor provider-ANs, in what concerns a set of previously negotiated SLSs.

TCP-FEC: Adding extra data segments based on the senders current window. Block coding is used to encode the redundant data from the available data. The block coding type is open for implementors to choose. Reed-Soloman coding, based on algebraic codes is one option.

Type of Composition: There are different levels of compositions; “Network Integration”, “Control Sharing” and “Network Interworking”. “Network Integration” represents the highest level of cooperation where all resources are controlled of a common (distributed) “Ambient Control Space, ACS”. Correspondingly, “Networking Interworking” represents the lowest level of cooperation where control of resources is intact after a composition.

QoS Functional entity (QoS-FE): Functional entity of an ACS that manage the QoS of an AN. The QoS-FE may have several components, one of which is INQA-FE for the advertisement, negotiation and monitoring of SLSs.

QoS Association: It defines the result of the negotiation of SLSs. To achieve a QoS association two main steps have to be performed: first, the negotiation of SLSs via the ANI and then, the automated configuration of QoS, based on the agreed SLSs, through the ARI.

2.2 Acronyms

Acronym	Meaning
AAA	Authentication, Authorization, and Accounting
ACS	Ambient Control Space
ACY	Ambient Connectivity
ADSL	Asymmetric Digital Subscriber Line
AN	Ambient Network
AN-CC	Ambient Network Congestion Control
ANI	Ambient Network Interface

API	Application Programming Interface
AQE	Ambient Network QoS Entity
ARI	Ambient Resource Interface
AS	Autonomous System
ATM	Asynchronous Transfer Mode
CA	Composition Agreement
C-FE	Composition Functional entity
C-GSLP	Composition GSLP
CC-FE	Congestion Control Functional entity
CC-FE-CS	CC-FE functions in ACS
CC-FE-CY	CC-FE functions in ACY
CSMA	Carrier Sense Multiple Access
CSMA/CA	CSMA with Collision Avoidance
DCCP	Datagram Congestion Control Protocol
DEEP	Destination Endpoint Exploration Protocol
DiffServ	Differentiated Services
DNS	Domain Name System
EGIMPS	Extended GIMPS
EGIST	Extended GIST
e2e	end-to-end
ECN	End-point signalling notification
FE	Functional entity
FEC	Forward Error Correction
GANS	Generic Ambient Network Signalling
GIMPS	General Internet Messaging Protocol for Signalling
GIST	General Internet Signaling Transport
GLL	Generic Link Layer
GSLP	GANS Signalling Layer Protocol
GTL	GANS Transport Layer
GTLP	GANS Transport Layer Protocol
HMM	Hidden Markov Model
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPPM	IP Performance Metrics
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
MAC	Media Access Control
MRM	Message Routing Method

MRI	Message Routing Information
NBP	Network Border Patrol
NSIS	Next Steps in Signalling
NTP	Network Time Protocol
OSPF	Open Shortest Path First
PAN	Personal Area Network
PNNI	Private Network-to-Network Interface
QoS	Quality of Service
QoS-FE	Quality of Service Functional entity
INQA-GSLP	GANS Service Layer Protocol for QoS
QoS-NSLP	QoS NSIS Signalling Layer Protocol
ROTT	Relative one way trip time
RTT	Round trip time
RTP	Real-Time Transport Protocol
SCTP	Stream Control Transport Protocol
SLA	Service Level Agreement
SLI	Service Level Indication
SLS	Service Level Specification
SLS-T	Service Level Specification Template
TCP	Transmission Control Protocol
TFRC	TCP Friendly Rate Control
TLS	Transport Layer Security
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VoIP	Voice over IP
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
3G	Third Generation (of mobile networks)
3GPP	Third Generation Partnership Project

3 Introduction

The Ambient Networks project has developed an innovative, industrially exploitable new network vision based on the dynamic composition of networks to avoid adding to the growing patchwork of extensions to existing architectures. This provides access to any network, including mobile personal networks, through instant establishment of inter-network agreements.

The project adopted the design paradigm of horizontally structured mobile systems that offer common control functions to a wide range of different applications and air interface technologies. Such a radical change requires the definition of new interfaces and a multitude of standards in key areas of future media- and context-aware, multi-domain mobile networks.

The strategic project goals valid over the full scope of all project phases are:

- to define an affordable and scalable mobile communication network, that opens up new ways to communicate, for billions of people and devices,
- to facilitate "ambient networking", allowing increased competition and cooperation, enabling efficient use of resources in an environment populated by a multitude of devices, technologies and business actors,
- to achieve a broad industry consensus on ambient networking, creating new business opportunities for mobile network service provisioning, and
- to develop a solution that provides easy to use, rich and trustworthy multimedia communication services for all.

The work is geared towards increasing competition and cooperation in an environment populated by a multitude of user devices, wireless technologies, network operators and business actors.

3.1 Scope of final Deliverables

We have organised the project in three phases that reflect changing priorities. Phase 1 is ending in 2005 and Phase 2 starts directly after in 2006. Following a natural pattern, we defined these phases as: the exploration of new concepts and approaches and establishment of their feasibility, the development of these concepts into the actual communication technologies, and the synthesis of entire systems out of the individual technologies.

The different types of work encompass many different aspects of technology. The set of final deliverables for Phase 1, produced at the end of 2005, form together the output of the first phase of the project:

D-1-5	AN Framework Architecture
D-1-6	AN Case Studies Report
D-2-4	Multi-radio Architecture - Final report
D-3-3	Connecting Ambient Networks – Final Architecture, Protocol Design and Evaluation
D-4-3	Ambient Network Mobility Framework & Mechanisms
D-5-3	SMART – Final Architectural Design
D-5-4	SMART – Final proof of concept and Demo
D-6-3	Second paper on context aware networks (results + further work)
D-7-2	Ambient Network Security Architecture
D-7-3	Security and Trust Cross-Issue
D-8-3	Ambient Network Management - Proof of concepts and evaluations

Table 1: Overview of final deliverables

3.2 Presentation of work package 3

Composition of networks is a core concept of Ambient Network. Composition is a basic, scalable and dynamic form of network cooperation to achieve control plane interworking and control sharing between networks. Today this is not possible without extensive manual configuration.

When Ambient Networks (AN) compose, their control planes, called ACS (Ambient Control Space), coordinate such that to the outside world the composed Ambient Network appears as a single uniform Ambient Network. The interface an Ambient Network presents to the outside is the ANI (Ambient Network Interface). The ACSes of Ambient Networks communicate among themselves via the ANI using the Generic Ambient Network Signalling (GANS) protocol. An ACS is subdivided in Functional Entitys (FEs) that encapsulate possible management and control functions in a network, e.g. QoS-FE and Congestion Control FE (CC-FE). The Composition FE (C-FE) is indispensable and orchestrates the Composition process.

Work Package (WP) 3 “Network Composition and Connectivity” addressed several issues vital to Composition, and hence at the heart of the Ambient Networks project. In task 1 (Composing Ambient Networks), the Ambient Network Interface (ANI) and the GANS protocol were designed. Task 2 (QoS in Ambient Network) aimed to provide dynamic Service Level Agreements (SLAs), to achieve seamless QoS between ANs. Task 3 (Network-assisted Congestion Control for Ambient Networks) developed highly adaptive congestion control mechanisms based on network information sharing to ensure efficient and fair use of network resources in dynamically changing environments. Finally, task 4 (Migration Strategies and Resource Provisioning in heterogeneous networks) dealt with the integration of legacy network as well as with migrating today’s networks towards Ambient Networks.

3.3 Motivation and scope of this document

This document on the one hand gives an overview of the work performed in WP3 during the first phase of the Ambient Networks project, particularly it gives an overview of the GANS protocol, QoS in Ambient Networks and migratability of Ambient Networks. Additional results of WP3 are detailed in D1.5. The focus of this deliverable is the evaluation of results obtained.

The GANS protocol is evaluated by means of a scalability analysis of GTLP (the lower GANS layer) and a prototype implementation. The QoS Framework (INQA) is evaluated regarding scalability and robustness. Also standardisation possibilities are discussed. Four complementing congestion control schemes were designed and evaluated for their use in an Ambient Network environement. We also evaluate Datagram Congestion Control Protocol (DCCP) and several schemes for differentiating loss regarding whether it is due to congestion or to other causes, e.g. wireless losses.

This document also gives an overview of design guidelines for migrating existing networks to Ambient Networks and evaluates to what extent Ambient design adheres to these principles.

3.4 Document Structure

Chapter 4 is a summary of work already reported in detail in other deliverables. It gives an introduction to the composition concept and proceeds with a summary of requirements on composition. The architectural highlights regarding composition, congestion control and QoS are reported, as well as the protocols developed in WP4, and how all components fit together. Finally a brief summary of the work on migration is given.

Chapter 5 describes the architecture components and protocols in more detail. In particular the GANS protocol suite, the INQA architecture for QoS, the work on QoS measurements and on QoS interworking with legacy networks are presented. Also the congestion control architecture is described.

Chapter 6 is the focus of this deliverable. It contains the evaluation results regarding the GANS protocol, INQA (the QoS Architecture), the congestion control algorithms, network measurements and migration.

Chapter 7 summarizes the main conclusions and give an outlook for phase 2 work.

The Technical Annex contains details on the following topics

- GANS Scalability Analysis
- Usage of INQA to support elastic and non-elastic traffic
- Dynamic Control of Inter-Network QoS Agreements (INQA) Analysis Scalability and Convergence
- INQA Loop Avoidance
- INQA Robustness Evaluation
- Analysis on INQA standardization
- Interworking Functionalities

4 Technical overview

For the convenience of the reader, this chapter gives an overview of results to be evaluated, and the next chapter gives enough background information for the document to be pretty much self-contained.

Regarding composition, we developed a number of composition scenarios, and derived requirements [9], which we summarize below. We define composition as a dynamically negotiated cooperation between networks. Network cooperation includes not only routing of user traffic, but, particularly, control-plane cooperation. Different types of composition are possible depending on how closely the networks cooperate: network interworking, control sharing and network integration [14], [25].

WP3 also elaborated how a typical composition process proceeds, and analysed a number of use cases (now in D1.5 [14]). We additionally illustrated to what extent composition can be realized with today's technologies, and where new methods and protocols must be added [25], [63].

When Ambient Networks negotiate a composition, they need a protocol to do so. For this purpose the GANS protocol was developed. GANS is a flexible, extensible protocol that FEs can use to communicate via the ANI. The GANS protocol is described in more detail in later sections.

Another focus of WP3 was a QoS architecture and a GANS-based protocol for negotiating, realizing and enforcing SLSs between Ambient Networks. The Internet control plane enables packet routing between networks, which makes it suitable to provide best-effort data transport between an increasing number of end-hosts. Regarding traffic with extra quality requirements, more advanced features are needed to control QoS between end-hosts. Currently there are a number of available solutions to perform the provisioning of IP networks, from static-provisioning to over-provisioning, passing by signalled-provisioning. Independently of the current mechanism to control QoS inside networks, a major limitation for the deployment of end-to-end QoS in heterogeneous environments is the lack of a dynamic approach to control QoS agreements between networks. This situation tends to get worse in a scenario with Ambient Networks, since such scenario encompasses more dynamic topologies with multi-homed and mobile hosts and networks. Ambient Networks must provide the capabilities to offer multiple QoS classes for end-to-end services, across different types of network technologies and different address domains.

There are some research efforts to develop mechanisms to manage and control internetwork QoS. One of them is the SIBBS protocol of the QBone group of Internet2 [http://qbone.internet2.edu/bb/]. This protocol follows a flow-based approach, in which QoS internetworking is configured based on flow demands, having no previous knowledge about the capabilities of networks. To add the pro-activeness, required by Ambient Networking, to QoS interworking, the Mescal project [http://www.mescal.org/] followed a tight approach to the connectivity internetworking seen in the Internet. This is, the Mescal project proposes QoS extensions to BGP [RFC1772]. This approach allows QoS internetworking to be configured with a previous knowledge about the capabilities of networks in what concerns the treatment of specific types of traffic. However, the Mescal approach lacks an integrated approach to advertise, negotiate and monitor QoS agreements between networks. Moreover, the tight coupling of QoS advertisement mechanisms with BGP, may not only absorb the scalability problems of BGP, but may also make an extension to network scenarios requiring other type of Internet routing impossible. Therefore, the AN QoS architecture is presented as a better solution, since it integrated mechanisms to advertise, negotiate and monitor QoS agreements in an approach dissociated from specific Internet routing protocols.

In case QoS cannot be signalled or for best-effort traffic, a variety of congestion control mechanisms, each specific to the particular network situation, can improve network throughput. As alternative for streaming real-time applications which cannot work with TCP-based congestion control, we studied DCCP as a transport protocol. Furthermore we studied schemes that have been proposed for differentiating between losses due to congestion and losses due to other problems (e.g. wireless link).

We have also looked at working with legacy networks. We have looked at measuring the performance and QoS offered by these networks. This means that even when an SLS cannot be agreed with all networks on a path we can be sure that the application requirements are met. Furthermore we must be able to add Ambient Networks functionality to existing networks to improve interworking. What is needed is a smooth migration path from legacy networks to Ambient Networks. To ensure designs are suited to migration we have developed a set of migration principles.

4.1 Requirements

The overall requirements and design principles for Ambient Networks are given in D1.5 [14] (sections 3.3 and 3.4). Here we are referring to those by highlighting how they are considered when the detailed requirements for WP3 were defined [9]. We also discuss how the requirements derived for WP3 in the beginning of the project are reflected in WP3 results.

The principles defined in D1.5 (section 3.4 on “Architectural Design Principles”) are below commented in relation to our work on “GANS protocol suite”, “INQA-FE” (previously called QoS-FE) and “Congestion Control” (CC-FE).

Principle #1: Ambient Networks build upon open connectivity and open networking functions

Principle #2: Ambient Networks are based on self-composition and self-management

Principle 1 and 2 relates to dynamic generation of Service Level Agreement (SLA) and QoS management. Observe that this aims at interaction between all combinations of network elements. Private hosts (terminals), Personal Area Networks, as well as all kinds of Private and Public Wide Area Networks.

Principle #3: Ambient Networks functions can be added to existing networks

This relates to migration from existing networks to future “Ambient Networks”. It also relates to bilateral composition agreement like SLA for QoS that shall be performed between ANs in a flexible manner.

Principle #4: Ambient Networks simplifies network control

Dynamic SLA is required in order to maintain network simplicity when QoS functionality will be required.

Principle #5: Ambient Networks scale between PANs and WANs using the same protocols

The "GANS protocol suite" is designed to meet this requirement. This is also thought for the service specific protocols e.g. related to QoS and Congestion Control.

D1.5 do also list a number of main requirements under section 3.3 "Requirements on Ambient Networks":

1. Heterogeneous Networks
2. Mobility
3. Composition
4. Security and Privacy
5. Backward Compatibility and Migration
6. Network Robustness and Fault Tolerance
7. Quality of Service
8. Multi-Domain Support
9. Accountability
10. Context Communications
11. Extensibility of the Network Services Provided
12. Application Innovation and Usability

The relation between these requirements and WP3 internal requirement from D3.1 can be described as follows. Observe that only a limited set of requirements is selected to highlight our concern described in this document.

R3.2 All designs must be scalable and modular.

This has been a general goal in our design (and the described issues relates mostly to D1.5 requirements 1, 3, 5 and 7 above). The scalability is however so far not verified / solved.

R3.3 Designs shall be as semantically independent of specific data transport technologies as possible.

This is particularly true for the GANS protocol suite but also to large degree to QoS and Congestion Control (1, 3, 5, 7 and 8). It is verified by the fact that the GANS protocols can be carried on any network including non-IP networks.

R3.10 ANs must be able to exchange Functional entity control information via ANI.

Generally true for the GANS protocol suite (3 and 10). The individual x-GSLP can be designed to meet all foreseen requirements of any FE.

R3.32 The QoS Functional entity (QoS-FE) shall be able to signal QoS independent of the degree of Composition.

Related to QoS and Congestion Control (3, 5 and 7). The existing test implementation supports this requirement but may still be incomplete seen from for a future perspective.

R3.34 CC-FE should ensure fairness and stability.

Focused on Congestion Control (7). Verified be simulations.

R3.37 GANS must provide the means for the QoS-FE to set, maintain, and release network resources.

True for the GANS protocol suite (3, 7 and 10). This is verified by the existing INCA-GSLP implementation

R3.38 GANS shall be able to operate between non-composed, partially composed, and fully composed ANs.

This is an important requirement from a migration and heterogeneity point of view (1, 3). The minimum requirements on x-GSLP signalling are still not fully defined. This may have to be defined for each FE.

R3.41 AN control signalling (both GANS and legacy protocols) should be transported with adequate QoS.

Relates to QoS and Congestion Control (7). The mechanism to define bearers for the signalling itself is so far not fully defined even though the capabilities are there.

R3.45 GANS must be able to transport any type of signalling information between FEs in the same AN and in different ANs.

Relates to the GANS protocol suite (3 and 10). This is possible since GANS support any x-GSLP implementation and also existing IETF defined protocols (NSLP)

R3.53 Migration should be incremental and of appropriate "granularity".

Relates to migration (5, 12). AN of different development states will be able to interact and can use existing non-AN network for transport.

4.2 Architectural highlights

4.2.1 GANS protocols suite

GANS (Generic Ambient Signalling Protocol) represents a protocol suite specially designed to support communication between Ambient Control Spaces. The GANS protocols meet the requirement for a future signalling protocol. In the figure below it supports the communication between two FE (Functional entity) which so far is designed within the AN project, the INQA-FE designed for QoS control and the CC-FE-CS for Congestion Control.

Observe that the CC-FE is split into a CS part for the Ambient Control Space (ACS) part and a CY part for the Ambient Connectivity Layer. The CC-FE-CY represents the Control Function at the transport layer and the different CC-FE-CY use inband signalling, carried in the user data, which depends on different implemented approaches.

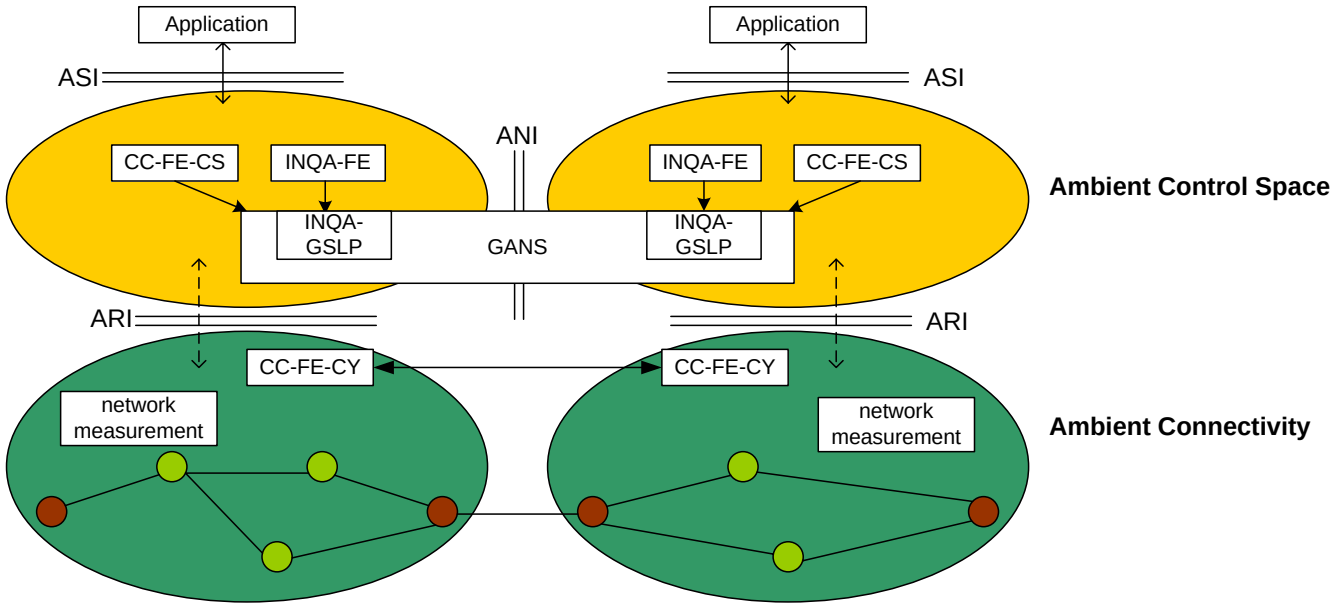


Figure 1: Communication architecture

The GANS protocol suite is depicted below and its main component is GTLP and GSLP. We have a generic part GTLP and a service specific part xGSLP, which also can be seen as part of the specific FE.

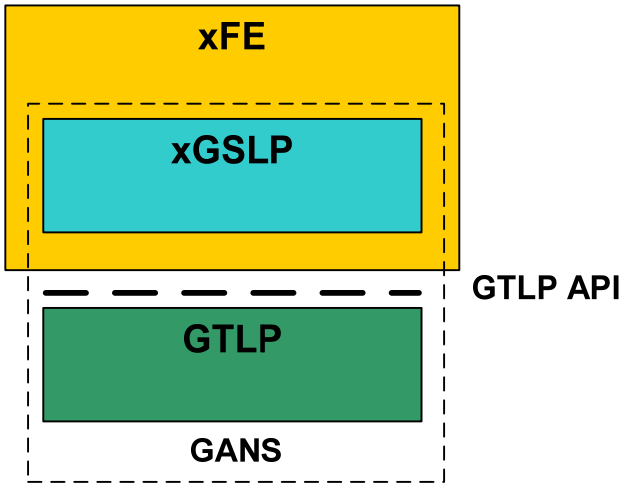


Figure 2: Protocol structure

4.2.2 Dynamic Control of Inter-Network QoS Agreements (INQA)

In a scenario with ANs, any mechanism aiming to control inter-network QoS must allow the automatic establishment and maintenance of QoS agreements between ANs that may be mobile. The concept of QoS agreement allows the separation of inter-network and intra-network QoS control, supporting heterogeneous intra-network QoS technologies. With this goal in mind, we developed a mechanism to dynamically control *Inter-Network QoS Agreements* (INQA).

The INQA proposal allows the establishment and maintenance of bi-lateral QoS agreements between ANs, based on a publish-subscriber mechanism that permits ANs to:

- Advertise to its neighbours a set of possible QoS agreements, via which the advertiser-AN promises a certain level of QoS to a certain amount of traffic. This means that advertised QoS agreements allow ANs to be aware of the QoS level of available communication services.

- Negotiate previously advertised QoS agreements, based on which the negotiator-AN is allowed to send a certain amount of traffic to a certain destination network.
- Monitor previously negotiated QoS agreements, permitting an advertiser-AN to update previously offered agreements and permitting a negotiator-AN to establish a QoS agreement with a different neighbour.

Within the INQA framework, each AN can have three roles: provider, customer, and customer-provider. In the provider role ANs advertise local network services, whereas in the customer role ANs negotiate network services that where offered in previous advertisements. Negotiated services may be used by local applications. Networks operate in the customer-provider role when they want to "resell" services advertised by their neighbours.

Within the Ambient Network Control Space (ACS) the functionality of INQA is implemented in the INQA Functional Entity (INQA-FE), also known as Ambient Network QoS Entity (AQE). The INQA-FE is a component of the QoS-FE, which may include other QoS related functionalities, such as Ambient Network Virtual Pipes for QoS.

Four modules and one signalling protocol are implemented by the INQA-FE. The modules implement the capability to advertise (INQA-A), negotiate (INQA-N), monitor (INQA-M), and admit (INQA-AC) QoS agreements. The INQA signalling protocol is implemented as a *GANS Signalling Layer Protocol* (GSLP), as so it is referred to as INQA-GSLP. As any other GSLP, INQA-GSLP uses the transport services provided by the *GANS Transport Layer Protoco*l (GTLP). It also uses the services of the *Destination Endpoint Exploration Protocol* (DEEP) allowing INQA to operate based on names and not IP addresses.

INQA has a strict relationship with modules to measure the network behaviour and with modules to control network congestion. The former provides INQA, via the Ambient Network Resource Interface (ARI), with information about the status of the network, such as available bandwidth and levels of delay and packet loss between pair of edge points. The latter allows a customer-AN to know if a certain provider is not respecting a negotiated QoS agreement. For this, the congestion control mechanism notifies INQA if congestion occurred in a path to a network that was suppose to be covered by a negotiated QoS agreement. INQA can also trigger the congestion control mechanism if some QoS agreement failed to be established. This will allow the control of congestion in a path without QoS assurances.

4.3 Common Use Cases and Composition Verification Scenarios

The composition verification scenario work that was carried out by WP3 is represented in [32] and it was WP3's internal effort to validate the composition process and related concepts. This work contains of series of verification scenarios and two of them have extended descriptions, one of which is an example of Network Integration while the other illustrates Control Sharing. It must be noted here that what is now called "Type of Composition" was formerly referred to as "Degree of Composition" with full and partial being the degrees used to characterize Composition [9]. The verification scenarios illustrate concrete examples of how composition can be implemented using existing technologies and what benefits it provides and what functionalities need to be added to support the composition in the presented environmet. In many scenarios, one of the main issues was the interfacing of existing technologies and how different technologies that were not originally designed to work together may co-operate. Another important issue was how the technology designed for static networking environment could be deployed in dynamic environment.

Common Use Cases work aims to provide more detailed information on how different control functionalities like FEs developed by different WPs are co-located and how they interact. The use case work is further described in [15].

There are 3 different common use cases:

1. Media distribution to a number of people with different accesses/terminals
2. Continuous connectivity

3. Network composition

Two first use cases do not concentrate on illustrating the composition related aspects, thus the first use case is highlighting the smart multimedia routing in media delivery and second one illustrates how continuous connectivity is maintained in dynamic networking environment from multi radio resource management and mobility perspectives. These two use cases base on the third use case and its composition related work and therefore they do not repeat the definition of the composition process.

Third use case concentrates on the composition process by describing control functions involved and how they do interact during the composition process. This interaction can be divided into two parts; inter ACS and intra ACS communication. The former contains co-operation of control functions carried out over the ANI and the latter contains control functions co-operation within the ACS, which does not use the ANI.

Network composition contains 3 different (sub) use cases; 1) Forming of a PAN, 2) Inter-operator handover and 3) Moving train and routing groups. These use cases validate the composition work and results, since so far there has not been formally the project wide work to study how various FEs are involved before, during and after the composition. One of these use cases also considers how multiple compositions are used in co-operative manner; i.e. how a composition requiring other composition(s) to be created has been triggered and established.

The forming of a PAN use case describes how a PAN is formed so that multiple devices form an AN. This has been done mainly from the composition and management point of views and therefore the use case has made some working assumptions on what (context) information is available and when. The Inter-operator handover use case concentrates to verify the composition procedures according to [49]. It covers control functions developed in different WPs like mobility and multi radio resource management, composition, connectivity, context management and management overlays. The use case provides a specific view on composition for all involved control function to illustrate how functionalities are aligned with the composition process and what occurs before and aftere the composition. Also, uses of multiple dynamically triggered compositions that have interdependencies are represented and some related issues are addressed. The moving train use case concentrates on routing groups, which is one central concept of mobility work and how these groups co-exist with different kind of hierarchical overlays.

The main results from the composition work perspective were clarified roles of Composition-FE, ConCord-FE and Management-FE during the composition process, dynamic composition triggering and different sequences of the composition procedures.

4.4 Migration

A key part of our work has been to look at migration. Many proposed communication systems have failed to be deployed, for example Internet QoS. We have asked the question: “Can anything that is done during the design phase of a new network have any impact on how easy it will be to migrate to the new solution?” And have looked at both successful and unsuccessful migrations. From this work we have devised a set of migration principles and also suggested migration phases.

A critical part of migration is interworking with legacy networks. We have looked at how end-systems can adapt to the offered service levels across many networks. As part of this work we have looked at loss-differentiation algorithms which are necessary in order to differentiate wireless losses from network congestion. Another important aspect is to discover what QoS is offered by legacy networks and to provide automatic interworking functions.

We have studied many past examples of migration – including the Internet approach to the shortage of global IPv4 addresses, wireless LAN hotspots, ADSL broadband networks and cellular network evolution. From these we have developed a number of general principles or

guidelines that give a good indication of how migration-friendly a solution will be. We believe that if these guidelines are followed during the design of systems beyond 3G then it will be easier to migrate to them. The three principles identified are:

Contain Coordination and Constrain Contamination

Solutions are easier to deploy if they affect

- minimal numbers of components
- minimal types of components
- minimal layers in the protocol stack
- minimal types of interested party

So, for example, it is important to identify suitable, independent functions. A good example is that of IP itself. Its design philosophy is to push functionality to the edge and keep the network as simple as possible. The 'scope' of changes required for a new service is therefore confined to the edge.

As a less successful example, one of the reasons that IP QoS has been hard to deploy is that it requires changes to all elements along the end-to-end packet path. This is perhaps also a difficulty with 3G deployment as it requires simultaneously new handsets, a new RAN and (in practice) new core network elements.

Simplicity, especially of interfaces

Complexity does not make for easy migration.

- Complex solutions take longer to develop than simple ones
- Complex solutions can be difficult to install, manage, debug and fix
- Each interested party requires a simple interface to the solution so that they can readily understand and use it
- Simple rather than grand plan solutions are usually easier to deploy incrementally

In general there is a trade-off between functionality and complexity. However, simple solutions have often been more successful than complex ones, even if the latter have multiple capabilities. An obvious example is the IP protocol: it does not provide a lot of functions by itself but it is widely accepted partly due to its simplicity. Moreover, this simplicity makes it easier for higher layers to understand, and so makes it easier to implement new capabilities.

The benefit of migration should be obvious to (at least) the party migrating

All parties should benefit, and, if any party can be seen not to benefit or even to lose out, they will not migrate, and may even try to impede the migration – perhaps through delaying standards and introduction of a competitive technology. This type of behaviour was felt by some to be evident in the demise of Hiperlan2 and the rise of IEEE802.11b.

5 Architecture components and protocols

5.1 GANS protocol suite

The GANS protocol suite consists of a two-layer model: GANS Transport Layer Protocol (GTLP) and GANS Signalling Layer Protocol (GSLP). GTLP defines a classical transport layer, and comprises new functionalities to be developed (e.g., support of symbolic names). The GSLP includes control signalling application protocols used, for instance, for AN Composition and SLS negotiation (see Section 5.2- INQA).

Besides transportation of signalling messages with required service and security quality, the task of the lower layer (GTL) is to locate signalling destination, i.e. address resolution (or delegating address resolution). The resolved routing address will be stored locally so that subsequent protocol messages could be forwarded directly without using the abstract address again. GTLP supports both multi-hop and single-hop signalling transport.

In figure below, the GANS protocol stack structure is illustrated and all new functionalities extending the NSIS protocol stack are coloured in green.

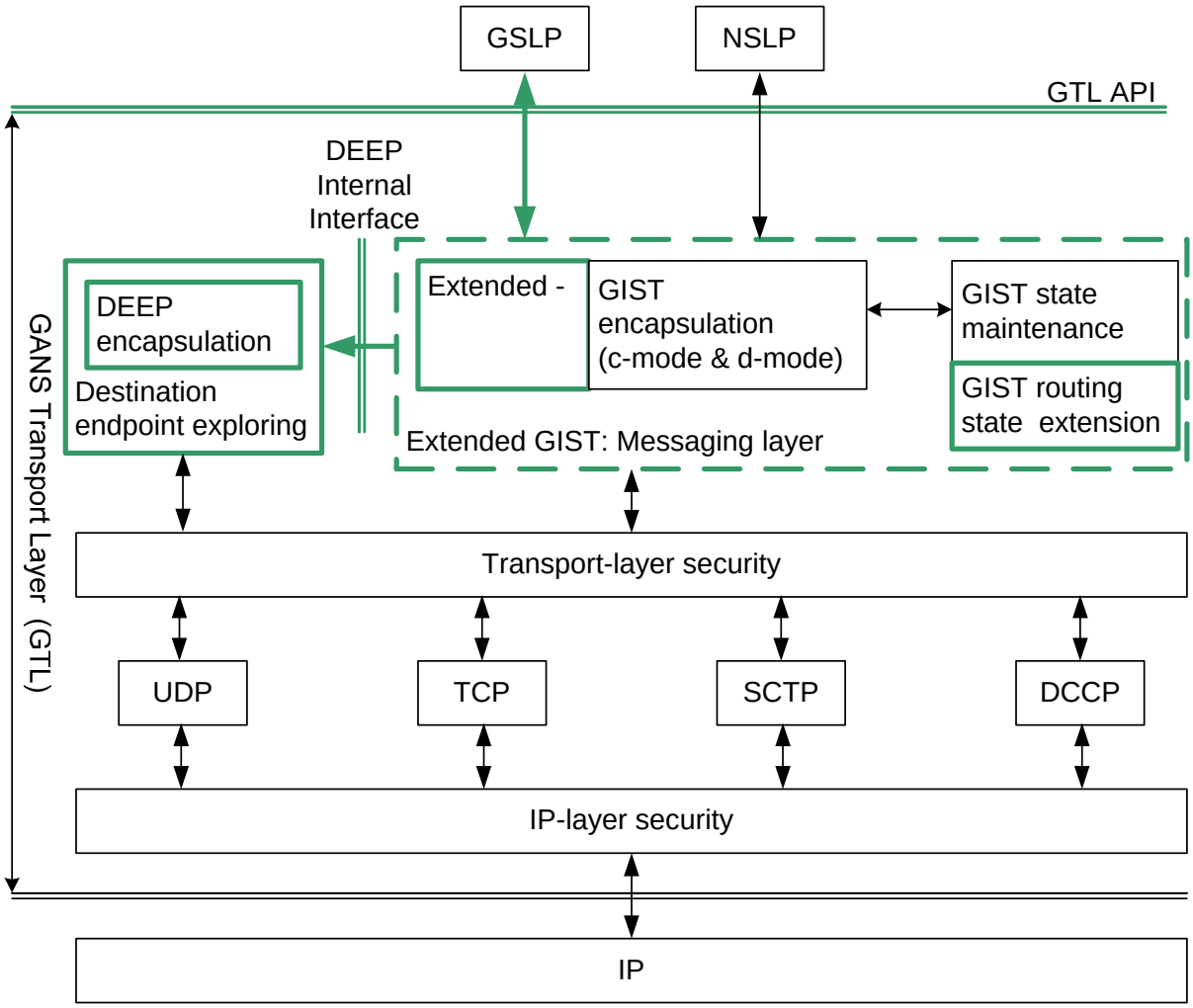


Figure 3: GANS protocol stack components.

GTL API is the interface provided by GTL to GSLPs. It is a simple extension of and backward compatible to GIST API. Comparing to GIST API, GTL API enables GSLPs using symbolic names instead of IP-addresses to address the destination endpoints of signalling messages and it also supports single-hop signalling applications.

EGIST is based on GIST and considers the needed add-ons with respect to GANS and ANs scope. It is responsible for the transportation of GSLPs signalling messages. Like GIST, EGIST uses services offered by the underlying layers, for example the transport services from TCP, UDP, SCTP, or DCCP, and relies on the routing functionality implemented by the underlying network layer.

DEEP is designed for exploring the signalling destinations specified by symbolic names and it implements the symbolic name support in GTLP. By allowing the use of symbolic names we are able to interconnect domains with incompatible address spaces and to traverse firewalls and NATs. The internal interface of DEEP hides local name resolution service details from the rest of GTLP components. Only the GSLP signalling messages addressing destination using symbolic names will involve the use of DEEP name resolution services. The signalling messages using directly IP-addresses are processed without the use of DEEP.

5.1.1 INQA-GSLP

As any other GSLP, INQA-GSLP uses the transport services provided by the *GANS Transport Layer Protocol* (GTLP). It also uses the services of the *Destination Endpoint Exploration Protocol* (DEEP) allowing INQA to operate based on names and not IP addresses.

We can say that INQA-GSLP is a path-decoupled protocol, since it is designed to signal bilateral operations between AQEs of adjacent ANs, without having any knowledge about the path to be followed by any flow. The protocol makes use of an XML-based description to exchange information about SLSs, SLIs and further auxiliary information.

Each INQA-GSLP XML message contains a <message> element, which forms the root element of each INQA-GSLP message. The <message> element contains a <body> element and an optional <header> element. Here is an example for a simplified message:

```
<?xml version="1.0" encoding="UTF-8"?>
<message xsi:noNamespaceSchemaLocation="inqa.xsd">
        <header>
                address information of involved ANs
        </header>
        <body>
                information describing SLSs/SLIs etc
        </body>
</message>
```

Figure 4 shows the tree XML elements for an INQA-GSLP message. The information present in the header may be omitted in normal protocol operation when provided to/from the underlying transport protocol.

As shown in Figure 4, the INQA-GSLP allows AQEs to combine SLSs to the transmitted, by creating an SLS-Combo. Each SLS-Combo has a common field, created with the parameters common to all SLSs, and by a sequence of SLS fields, each of which containing parameters specific to each SLS. Figure 4 also provides an insight about the type of messages of INQA-GSLP, such as ADV for SLS advertisements, NEG for SLS negotiation, and SLINEG for SLI negotiation.

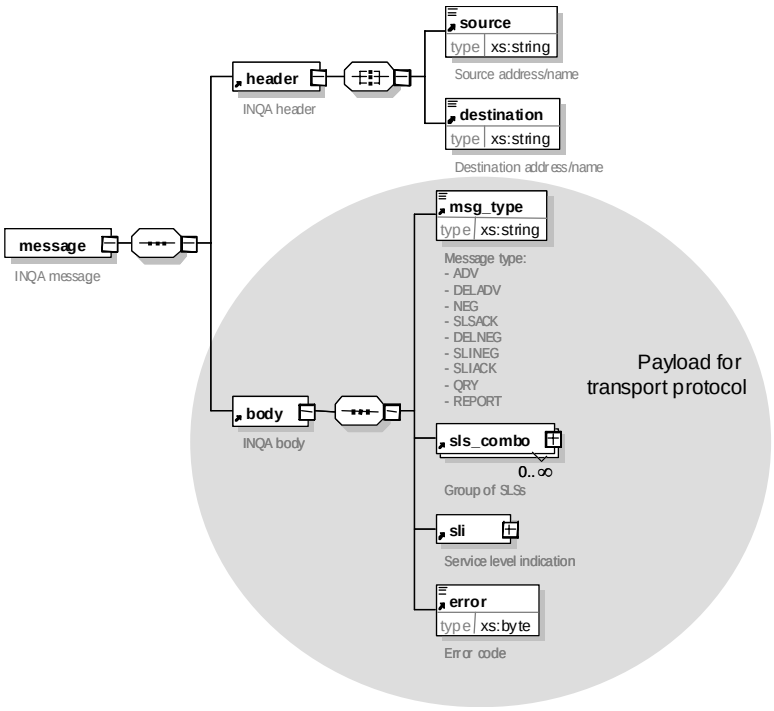


Figure 4: Basic INQA-GSLP structure

The deliverable R3.2 “Protocol Specification and Simulation” provides a detailed description of INQA-GSLP.

5.1.2 GTLP

The GTLP protocol is the lower layer of the GANS protocol. It comprises two main building blocks: the Destination Endpoint Exploration Protocol (DEEP) used for naming resolution purposes, and the Extended General Internet Signalling Transport (EGIST) protocol, which is based on the GIST protocol being specified by the IETF NSIS WG and provides the actual transport framework for the GSLP messages received through the GTLP API. The latter is the standard interface provided to the GSLPs, and is backwards compatible with the GIST API specified in [22], so that applications running on top of GIST can also run over GTLP without any modification. The parameters and primitives defined for the GIST API are maintained. However, a new Message Routing Information (MRI) object is specified in order to support single-hop oriented signalling applications, and the use of both IP addresses (IPv4, IPv6) and symbolic names to identify signalling endpoints. In this context, single-hop signalling applications mean those applications where only two nodes are involved in the signalling process at the application layer; multi-hop signalling applications refer to the scenario where multiple nodes supporting the same signalling application get involved.

When GTLP receives a message from a GSLP through the GTLP API on the same node, it extracts the abstract name used by the signalling application to identify the destination signalling endpoint. Then, the DEEP module is invoked in order to find out the IP address of the node on the target AN to which the message should be sent. Afterwards, the EGIST module encapsulates the GSLP message and sends it to the IP address returned by DEEP; depending both on the information passed from GSLP and on local policies, EGIST will use an unreliable (e.g., UDP) or reliable (e.g., TCP) underlying transport protocol to do so. Upon reception of a message, the GTLP may perform a handshake with the GTLP on the sending node enabling them to directly exchange messages in the future, as long as the corresponding state is refreshed timely. If desired, the two nodes also establish a security association. In the following subsections both the EGIST and DEEP protocols are described in more detail.

5.1.2.1 EGIST

The EGIST protocol comprises the actual transport framework provided by GTLP to the signalling applications running over the GTLP API. As mentioned before, EGIST is fully based on the GIST protocol. This protocol interacts with GSLPs, DEEP and the underlying layers, e.g., TCP, UDP, and Transport Layer Security (TLS). The latter ones provide the transport and security services that EGIST needs to convey the GSLPs' messages towards the right destination and fulfill the requirements settled by the signalling applications using the proper parameters at the GTLP API. EGIST comprises all functionality already deployed by GIST plus further functionality required for the GANS protocol suite. The interaction between EGIST and GSLPs occurs through the GTLP API, the interaction between EGIST and DEEP occurs via an internal interface, and the interaction between EGIST and the underlying layers happens through a well-defined interface (sockets API). Basically, EGIST can be seen as the same protocol as GIST plus some further functionality required for the GANS protocol suite. The new functionalities are:

- support of symbolic names to address signalling endpoints;
- interaction with a new protocol implementing a name resolution scheme;
- support of single-hop oriented signalling applications;
- name binding state storage and maintenance without modifying the routing state table specified by GIST;
- name binding state update.

Concerning the way signalling messages are routed towards the proper adjacent peer, GIST defines the concept of Message Routing Method (MRM). The MRM defines the algorithm for discovering the route that the signalling messages should follow between the source and destination signalling endpoint at the GIST layer; in order to forward the signalling messages between GIST nodes, i.e., nodes that support the GIST protocol, GIST relies on the legacy network layer routing. GIST design supports multiple MRMs, hiding in this way the routing-dependent details from the signalling applications. The default MRM is the "path-coupled" MRM, which defines the activation of the IP router alert option to deliver the signalling application messages to the correct adjacent peer. In conjunction with the MRM concept, GIST defines the MRI object, which allows signalling applications to specify the concrete MRM that GIST should use to forward their signalling messages towards the right destination. For instance, concerning the default MRM, where signalling information is supposed to travel across the data path, the MRI object includes the "MRM path-coupled" identifier and the flow for which signalling is being exchanged. Based on this MRI object, GIST will find out its adjacent GIST peer towards the destination of the flow identified in the MRI object and will send the signalling information there; the process is repeated until the destination of the flow, or the last GIST node in the data path, is reached. For more information on this process the reader is referred to [22].

In EGIST a new MRM is defined, considering the nature of the signalling applications comprised in the GANS protocol suite. This new MRM is called "path-decoupled/single-hop" MRM and requires modifications with respect to the internal processing of the messages received from the signalling applications, namely regarding the way messages are routed and the way the two operation modes defined by GIST are used. The new defined MRM is incorporated in the current GIST specification so that single-hop oriented signalling applications and symbolic names for identifying the signalling endpoints are supported. This integration is performed according to the guidelines provided in the GIST specification.

The support of symbolic names inside EGIST requires the storage of information about the mapping between symbolic names and the corresponding IP addresses. The DEEP protocol is assumed to be a stateless protocol, i.e., it does not maintain any state information internally. Therefore, EGIST is the entity responsible for maintaining such state information on behalf of

the overall GTLP protocol. For that purpose, EGIST uses the routing state table already specified by GIST [21], and encapsulates the name binding state information inside the new defined MRI object. This is possible due to the flexibility associated with the routing state table, which stores the MRI object as a whole instead of each parameter in separate columns, hiding the specific details associated to each MRM. In this sense, every new MRI object defined in the context of a new MRM can be stored within the same table.

A further extension to GIST is related to the update of a name binding state. When there is a relocation of signalling endpoint, which causes a change in the IP address and/or in the port being used by one of the signalling endpoints participating in a signalling relation, there is a need for updating the related name binding state. EGIST is responsible for such update, as it is the entity storing the name binding state. To that end, the GIST refresh procedure specified in [21] that is based on the exchange of two messages (GIST-Query/GIST-Response) is re-used.

In a nutshell, it can be said that the EGIST protocol is, as the name implies, an extension to the GIST protocol. It does not modify any of the GIST functionalities specified in [22]. Rather, EGIST extends the applicability domain of the GIST protocol by providing further features not included in the original protocol. In that sense, EGIST is backwards compatible with GIST, supporting both the signalling applications already defined in the NSIS scope, so-called NSIS Signalling Layer Protocols (NSLPs), and the GANS signalling applications (GSLPs).

5.1.2.2 DEEP

Destination Endpoint Exploring Protocol (DEEP) is designed for exploring the signalling destinations specified by symbolic names. The main purpose of the DEEP is to support a distributed resolution of an abstract name into an IP address. This is feature that enables to establish relations between domains using separated and incompatible address spaces. It also allows the capability to traverse Firewalls and NATs.

DEEP provides internal name resolution services for GTLP via a unified internal interface, which hides the implementation details of a local name resolution service. DEEP is not tied to any specific naming scheme, thus it was designed to support generic abstract names that can be based on both flat and hierarchical name spaces.

DEEP is a multi-hop protocol based on a query-reply messaging model, in which the query part is multi-hop oriented, i.e., involves typically multiple DEEP nodes, and the reply part is single-hop oriented, i.e., involves only two DEEP nodes; see example presented in the figure below. The protocol is stateless meaning that no additional protocol or name resolution related state information is stored in any DEEP node. It is transparent to DEEP how local name resolution services are storing name resolution information like caching it locally for further use. And even if DEEP does not provide any name resolution caching, the underlying name resolution services (e.g., DNS) may do so resulting in improved name resolution service response times and decreasing the overall DEEP message processing times in DEEP nodes.

DEEP name resolution is typically gradual process, which resolves an abstract name into a locator (e.g. IP address) in distributed manner involving multiple DEEP nodes and name resolution services that can be independent on each other. While a variety of mechanisms is conceivable, we settled for a sequential method in which the abstract name is resolved in several steps, by sending the same DEEP message from one name resolver to the next one until the IP address of the responsible node is discovered. In common case, DEEP name resolution involves the exchange of DEEP messages and this is done before control signalling application messages are exchanged.

Each DEEP node is configured to provide name resolution services to other DEEP nodes and externally DEEP nodes use mainly two interfaces; a local name resolution interface and a message transport service interface. Some DEEP nodes may be configured to act as a

"forwarder"; i.e. based on a local configuration, all received DEEP messages are always sent towards the same DEEP nodes.

For example, let's consider that we have two ANs; AN1 and AN2. A control signalling application in AN1 wants to contact its peer application in AN2 (e.g. FEy@AN2) and it does not know its peer application's contact address. Therefore, a DEEP message with the address FEy@AN2 is first sent to a name resolving node local to the originating AN1. This node may either know the IP address of the name resolver at AN2, or the IP address of an intermediary name resolver that is able to resolve a part of the current abstract name towards the complete name resolution by using the sequential procedure mentioned above. Upon knowing the IP address of the next resolver, the current DEEP node sends the "DEEP query" message there. This process is sequentially repeated until the name resolution process reaches the name resolver at AN2. The name resolver at AN2 knows the IP address of the contact point for FEy. At this point, the name is fully resolved and the "DEEP response" message is sent back to the originating node in AN1.

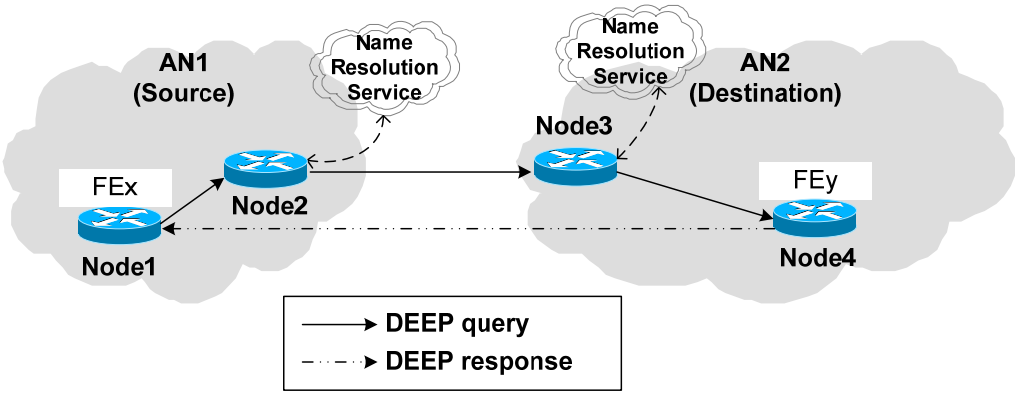


Figure 5: An example of DEEP name resolution.

With this method, it is possible for ANs to let a central "AN Gateway" handle all incoming control signalling messages e.g. in order to hide its internal infrastructure. This can be achieved by the local name resolver always returning the IP address of the AN Gateway. It is also possible however to return the IP address of more specific nodes, thereby distributing the load more evenly.

Additional set of DEEP features has been designed (also referred as Extended DEEP) and introduced in [47].

5.2 QoS Framework

As mentioned in section 5.1.1, our QoS framework, INQA, mitigates the current lack of control of inter-network QoS agreements in heterogeneous and dynamic environments. To achieve its goals, INQA has four major characteristics:

- Clear separation between inter-network and intra-network QoS signalling and control. Within a network, resource management is assumed to be done based on approaches such as DiffServ [51] and MPLS [46] or even over-provisioning.
- Handle Service Level Specification (SLS) [45] and not flows, in order to allow the system to scale and to allow operators to control the traffic entering and leaving their networks. The latter is not possible when using end-to-end flow-based signalling, which may only be prohibited to enter a network.
- Automatic adjustment of inter-network QoS agreements due to network movement or changes in the capability of each network.

- Support quality of service for elastic traffic (soft guarantees) and non-elastic traffic (hard guarantees), based on bilateral QoS agreements and admission control mechanisms, instead of per-flow end-to-end signalling.

Figure 6 illustrates the implementation of INQA in AQEs. The figure represents AQEs has network devices, which reflects the current prototype. However, INQA can also be implemented by a distributed system of AQEs. Three[3] of the INQA modules use INQA-GSLP to signal between adjacent neighbours via the *AN Network Interface* (ANI). While INQA-A uses the ANI to advertise SLSs, INQA-N uses it to negotiate previously offered SLSs, and INQA-M uses the ANI to negotiate *Service Level Indications* (SLI), that is, the capability to monitor previously negotiated SLSs.

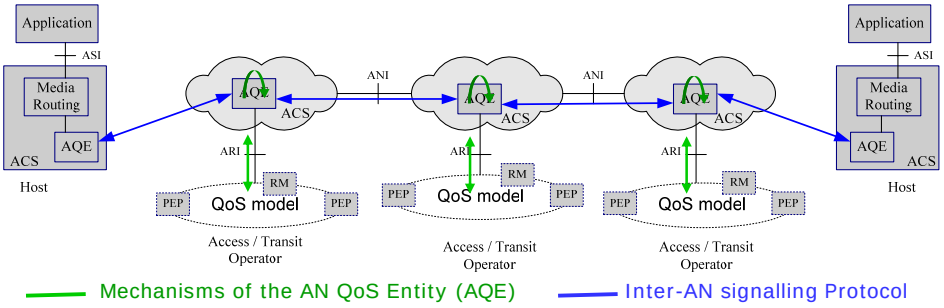


Figure 6: Illustration of INQA signalling

Figure 6 illustrate also the internal interaction between different INQA modules in the same AQE, as well as the interaction between the AQE and network dependent functions via the *AN Resource Interface* (ARI). While INQA-A does not use the ARI, INQA-N uses it to indicate to the local QoS control mechanism the need to realize a set of negotiated QoS agreements. On the other hand, the INQA-M module uses the ARI to collect, from local measurement mechanisms, information about the capability of each of the paths between a pair of edge devices.

The remaining of this section provides a brief description of each of the modules of INQA. The deliverable R3.2 “Protocol Specification and Simulation” provides a detailed description of the specification of each INQA module. The INQA-GSLP is further described in section 5.1.1.

INQA-A: Module for Advertisement of SLSs

The main task of the Advertisement Module of the AQE (INQA-A) is to receive advertisements, create and send advertisements, and manage the corresponding databases.

INQA-A can be divided into two functional halves. The passive half is responsible for receiving SLS Advertisement (Adv) and SLS Delete Advertisement (DelAdv) messages and for handling them. The passive part is operational in both Customer-ANs and Customer-Provider-ANs. Thus, it is a compulsory part of INQA-A. The active half is responsible for creating Advs and DelAdvs and for sending them to appropriate ANs. The active part is operational only on Provider-ANs and on Customer-Provider-ANs. It is an optional part of INQA-A.

INQA-N: Module for Negotiation of SLSs

The main task of the Negotiation Module of AQE (INQA-N) is to negotiate QoS agreements (SLSs) with neighbour ANs.

The negotiation module can be divided in three functional parts, according to the role played by the AN. When playing the Customer role, ANs do not offer network resources to other ANs and they negotiate SLSs for their own use (i.e., they buy resources). Provider-ANs offer

[3] The INQA-AC functionality is internal to AQE, having no interaction over any interface.

resources to other ANs. They participate in the negotiation of SLSs to be used by Customer-ANs, but they do not buy resources offered by other ANs. Customer-Provider-ANs can buy and sell resources. For instance, a transit network acts as a Customer-Provider-AN, since it buys resources from particular ANs (Provider-ANs) and sell those resources to other ANs (Customer-ANs or other Customer-Provider-ANs – e.g., an access AN). Depending on the role played by the AN, a set of functionalities that support its specific requirements is implemented.

INQA-M: Module for Monitoring of SLSs

The monitoring module of INQA (INQA-M) encompasses two major functionalities: intra-AN monitoring and inter-AN monitoring. Intra-AN monitoring provides up-to-date QoS state information within the AN to other modules of INQA and other functional entitys of the AN. Inter-AN monitoring checks the fulfilment of SLSs negotiated with other ANs.

INQA-AC: Module to Control the Admission of SLSs

The Admission Control mechanism (INQA-AC) only exists in Provider-ANs and Customer-Provider-ANs and is responsible for storing information about the AN capabilities as well as the amount of resources negotiated. The INQA-AC analyses the possibility of accepting (or not) new SLSs requested by Customer-ANs or Customer-Provider-ANs.

INQA-AC receives information from INQA-M and is consulted by INQA-A and INQA-N. INQA-M is responsible for monitoring the AN state and for gathering information about the resources available in the internal nodes of each ingress-egress path of the AN. INQA-N interacts with INQA-AC to verify whether a particular SLS request, sent by a Customer-AN, can be accepted. Finally, INQA-A interacts with INQA-AC to collect information about the available resources in order to create SLSs meant to be advertised to Customer-ANs in the vicinity.

5.3 Network Assisted Congestion Control

5.3.1 Overview and contribution

The congestion control design work carried out in Ambient Networks focused on congestion control techniques that address the foreseen challenges posed by the network environments of the future ANs. In [9] six technical scenarios were defined to illustrate the specific shortcomings of present-day congestion control schemes (Extremely short-lived Composition for burst-like transfer, Multihoming, Handover/handoff, Heterogeneous - partially AN-opaque transfer paths, Non-participating endpoints/non-conformant protocols, Mobile networks). These have been beard in mind when designing the new congestion control mechanisms.

Further a design objective for the whole congestion control approach was to make the adoption of the techniques developed as simple as possible. Enabling incremental or partial adoption of the AN-CC mechanisms was a major design objective. To achieve this goal, we have specified a modular architecture, and have designed four orthogonal techniques (see Figure 7), that can be deployed independently but also, be adopted in any combination without harmful interactions in most network types.

First, one mechanism (Checkpoint Nodes) can be deployed at the border of the network or at critical links in order to protect the surrounded network from congestion. Second, two mechanisms (TCP-FEC, Utility Fair CC) act end-to-end, similar to like TCP does, but they introduce new methods to detect and handle congestion situations. Finally, one mechanism (Contention Resolving CC) deals with congestion situations resulting from wireless multihop scenarios.

As described in the technical annex, the checkpoint nodes, TCP-FEC and Utility Fair CC can use feedback from the network to improve their performance. Measurements therefore can be seen as a complement to Network assisted congestion control.

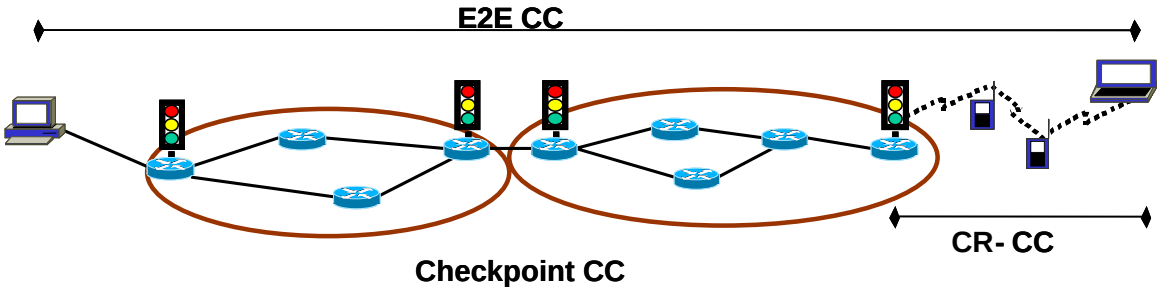


Figure 7: End-to-end view of the CC solution

Due to the nature of congestion control, the mechanisms are fairly independent of other ambient network elements. They can act in any heterogeneous scenario. Therefore, they are beneficial in most AN network scenarios. However, in some aspects (e.g. when it comes to network feedback), the mechanisms benefit from support from other AN-FE.

Although separate they actually address different problems within the congestion control framework. They are able to both act independently and together to address the problems of congestion control in different places and situations within an AN.

NACCAN Name	Problem addressed	AN Coverage	Advantage of	Interacts with
Checkpoint Nodes	Non-responsive flows Network feedback	Router to router	AN Network Control	Utility-Fair & TCP-FEC
Utility-Fair CC	Bandwidth allocation	End-to-end	Fairer sharing than TCP	Checkpoint nodes
TCP-FEC	Protecting admitted flows	End-to-end	Flow robustness to loss	Checkpoint nodes
Contention Resolving CC	Wireless contention	Hop to hop	Better energy/ link usage	-

Table 2: Problems addressed within AN Congestion Control

5.3.2 Overall Design

The congestion control architecture we have developed is naturally designed to follow the principles of the overall AN architecture. For efficiency and optimisation reasons the anchor point of the Congestion Control Functional entity (CC-FE) is the connectivity plane (ACY), though some functionality is, at least logically, placed in the ACS. These collections of functionalities are called the CC-FE-CY and CC-FE-CS, respectively. The CC-FE-CS is composed of all the functions to interact with other functional entitys (for example, to obtain information regarding bandwidth allocated to flows benefiting from QoS mechanisms) and to communicate via the Ambient Network Interfaces (ANI), while CC-FE-CY consists of the actual congestion control mechanisms, and techniques enabling the interaction with legacy solutions, such as TCP. The CC-FE-CS and the CC-FE-CY communicate via the ARI. This overall CC-FE architecture is illustrated in Figure 8.

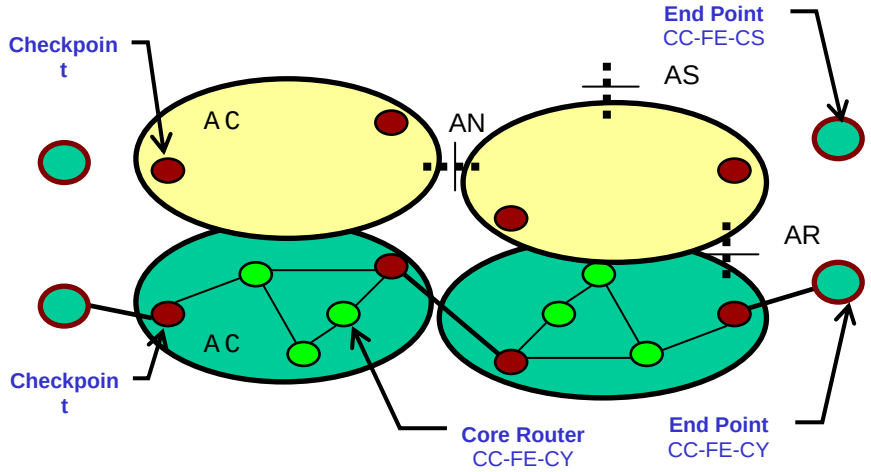


Figure 8: Congestion Control Architecture

The architecture differentiates between four classes of nodes, namely

- End Systems
- Checkpoint Nodes (e.g. edge routers)
- Core Routers
- Observers (e.g. Layer 2 bridges; not illustrated in the figure for clarity)

End systems denote the senders and receivers of ongoing end-to-end communication, while checkpoints and core routers are forwarding nodes on the path between the two communicating end systems. Of these, checkpoints act as intelligent edge routers, performing network protection functions, especially metering unresponsive traffic flows attempting to seize an unfair share of the network bandwidth. They can also assist end nodes in their congestion control activities by providing feedback regarding various network conditions. Core routers are envisaged as implementing only the simplest of CC-FE functions, and, most importantly, not keeping per-flow states in any manner. Finally, observers are nodes that are not necessarily active in the ACY, but instead only serve to gather information of, for example, link-layer conditions, that can assist other nodes in congestion control decisions. Thus, observers can be said to enable distributed cross-layer optimisation in the Ambient Networks CC-FE framework.

5.3.3 Realisation

5.3.3.1 Checkpoint Nodes

Checkpoints can be thought of as routers with extra, congestion control related functionality. This extra functionality can be roughly grouped into two categories:

- Active congestion control measures, using which checkpoints work together to prevent congestion collapse from unresponsive flows, and try to ensure that TCP-friendly flows get an approximately fair share of the available bandwidth.
- Providing more information to the end systems, allowing for more informed decisions related to congestion control response.

The basis of the checkpoint operation is the Network Border Patrol (NBP) proposal [9], which defines a mechanism edge routers can use to protect the network core from congestion collapse. Primary shortcoming of the NBP approach is the signalling overhead. Therefore, in the present checkpoint node design this effect is remedied by employing dynamic flow management. Furthermore, connection admission control to prevent new flows from entering the network core in case of imminent congestion collapse, or during the recovery phase

immediately following a collapse. Earlier studies indicate that these measures can speed up the network recovery, and thus increase the quality of end user experience.

The checkpoint approach is very migration-friendly. In principle edge routers and possibly selected core routers can be updated individually. For maximum benefit, all edge routers of a given network should be updated, although even partial deployment brings advantages compared to the present-day situation. For individual unresponsive flows the rate control mechanisms work as long as both the ingress and egress routers for that flow are updated. The state of other edge routers has no impact on the flows that do not pass through them.

5.3.3.2 Utility Fair Congestion Control

Utility fair congestion control is a novel approach for efficient bandwidth allocation in networks. In contrast to most traditional approaches that try to share bandwidth equally among flows, utility fair congestion control shares bandwidth based on application-specific quality metrics, represented by their utility functions.

Utility fair congestion control is a feedback based protocol. Routers continuously measure and compare the offered load (for a given output port) to the capacity and update the *link price*. If the load exceeds the capacity, the link price will increase. If the load is below the capacity, the link price will decrease until it becomes zero. Link prices are accumulated along a path and the resulting path price (*congestion measure*) gives the feedback for senders.

Again, utility fair congestion control is very migration friendly. We see several application scenarios that do not require full end-to-end support of routers. Ambient Networks can benefit from localized deployments in that utility fair congestion control can provide a flexible tool to control usage of critical resources. However, the full potential can only be exploited if applications are aware of the control framework. That is, utility functions need to be designed and tuned to the individual application's needs.

5.3.3.3 TCP-FEC

The basic idea is to add correcting codes to the stream of data carried by the TCP protocol as a block. By adding extra (redundant) data to the TCP stream, retransmissions to fill-in missing segments may be avoided. The self-correcting codes can be used against both congestion losses and noisy channel induced losses. In order to get the best performance out of TCP-FEC, it must be adaptive to current network conditions. The advantage of TCP-FEC is that it protects ongoing transmissions from packet loss. This way, ongoing TCP transmissions can be finished quicker and leave the system. It also avoids retransmissions, which might further block resources.

The cost of protecting the information is adding redundant data when implementing FEC, i.e. extra load on the network. However simulation results have shown, that TCP-FEC is fair with respect to bandwidth consumptions towards other TCP flows.

Migrating to TCP-FEC happens transparently to the network. No Network support is required. The changes needed are in the end-nodes, in the TCP protocol implementation. Once our proposed mechanism is included into the TCP part of an operating system, it can be used by all applications.

5.3.3.4 Contention Resolving Congestion Control

In a wireless multihop environment, node-disjoint traffic flows may contend for wireless resources, leading to a congestion situation that cannot be resolved by a common control element. Our proposed technique operates therefore distributed and decentralized among the involved nodes. The contention resolving congestion control is based on adapting the transmission schedules imposed by the energy management mechanism so as to minimize congestion resulting from contention due to transmissions associated with unrelated flows. The mechanism consists of two components: the energy management component and the adaptation component.

In order to minimize energy consumption, battery powered mobile devices must maximize the amount of time the network interface spends in a low power consumption sleep state. The scheduling of such sleep intervals is particularly challenging in the considered AN multihop environment. The protocol is therefore designed for fully asynchronous operation. Each node maintains an independent periodic sleep/wake pattern. Since the sleep/wake schedules of the nodes are determined independently, some intervals must be found in which the nodes can exchange traffic. A good distribution of sleep/wake schedules could provide some of the beneficial aspects of globally scheduled links. However, it is not possible to negotiate such a schedule in an asynchronous, decentralized, multi-hop environment. Such schedules must therefore be approximated adaptively, allowing nodes to alter their sleep/wake schedules to reduce contention. The latter is done by the adaptation component.

Migrating to Contention Resolving Congestion Control, is restricted to the nodes in a multihop environment. Our proposed technique operates solely within the local wireless environment and is intended to be transparent to higher layer CC mechanisms. In order to participate and benefit from the proposed mechanism, all involved nodes need to have this protocol deployed on link layer protocol level (layer 2).

5.4 Network Measurements

5.4.1 End-User QoS and Loss Differentiation in Wireless Networks

A key requirement for ease of migration is that independent functionalities are created. It has long been recognised however that IP QoS and mobility functionalities are not at all independent. Coupling between the functionalities occurs because QoS is tied to a specific route, and this route changes as a result of a mobility event. Without this coupling, when a mobility event occurs the Quality of Service state is no longer associated with the new route. Thus there is no QoS on the new route, and un-used QoS state on the old route (the later can affect how efficiently the network is used). Thus, QoS and Mobility Management need to be coupled to ensure that QoS is repaired after handover. There is also a problem whilst the handover is taking place.

This is essentially what TCP congestion control does today for data traffic – it achieves its required QoS (maximum bandwidth) through responding to the state of the network. However, different congestion control techniques will be needed for streaming and real-time (conversational) traffic types. For example, such traffic is more interested in throughput stability than in maximum throughput. Another difference is that the applications are typically not fully elastic, but instead have defined rates at which they like to operate.

Real-time interactive applications can typically switch between these pre-defined rates, for example by changing codec – the applications are adaptive – in response to the current state of the network. DCCP and its constituent algorithms (e.g. TCP Friendly Rate Control) offer a transport protocol that can be used by adaptive real-time applications whereby they can adjust their rate to the congestion experienced on the end-to-end route.

If we are to use DCCP and its constituent algorithms then we need to be able to identify which losses are due to congestion loss and which are due to wireless losses. Misclassification of losses as congestion losses triggers congestion control algorithms unnecessarily and results in loss of throughput and under utilization of resources. Several loss differentiation schemes for wireless networks have been identified from literatures. The modified *spike* scheme, based on relative one way trip time (ROTT) measurements at the receiver, has been studied over multi hop wireless networks with TCP and TFRC protocols using *ns2* simulations.

The simulation results show that the modified *spike* scheme improves the performance of TCP over multi hop wireless networks. It identifies the state of the network based on the ROTT measurements, invokes congestion control mostly during network congestion and avoids initiating slow start algorithm on every packet loss.

However the results obtained with TFRC protocol over multi hop wireless network are unsatisfactory. It has been observed that TFRC's behaviour is unstable over multi hop wireless networks and the ROTT measurements obtained at the TFRC receiver does not show any clear variation with network congestion. Further investigation is required on TFRC's behaviour in multi hop wireless networks and appropriate loss differentiation schemes for TFRC in such networks.

5.4.2 QoS Measurement Framework

Ambient Networks support both elastic and real-time applications. Many real-time applications require quality of service from the network. To support QoS-aware applications, one has to apply appropriate QoS measurement methods to find out the network QoS state. Existing approaches based on active and passive probing measurements have been investigated by IETF and ITU-T community. In active measurement, the measurement approach should not neglect extra probe traffic injected into the networks. Moreover, the QoS information should be available for individual applications.

The measured QoS parameters can be used as indicators for other QoS components such as admission control, congestion control, and SLS validation. This study has been taken into account of QoS measurement via a unified measurement framework. We will present our approaches for QoS parameters measurement.

Definitions of QoS parameters and measurements are given in many documents by ITU T1A1.3 working group and by IETF IPPM working group. IETF IP Performance Metrics (IPPM) working group is active in this area [17]. In [11], we concluded to consider IPPM for QoS measurement. There are a number of parameters that an application might have controlled for its traffic, such as packet loss, delay and jitter, and bandwidth. Many TCP-based applications often use packet loss to estimate the network state. It is end-user QoS issue. End-user QoS uses DCCP protocol to keep the network out of congestion.

Our approach is based on active probing packet to capture the network information via packet pair dispersions. We use statistical models to reason about QoS parameters in order to minimize injected probing packet costs.

5.4.3 QoS interworking with legacy networks

In the migration phase 1a, only layer 2 and layer 3 capacities are going to be used. Since no negotiation framework is defined for this phase, the gateway controls all the interworking functions, such as mapping different services, triggering authentication procedures… The main part of these processes will be configured manually: manual mapping, manual agreements between operators, manual configuration of these off-line agreements…

Obviously, in the framework described above, no signalling mechanism to negotiate a composition agreement is defined in order to provide QoS. Therefore, if a scenario where the transport plane of a legacy network should be used to habilitate the composition between two Ambient Networks, it would be desirable that these edge Ambient Networks should be able to infer the performance of the legacy network. In order to do that, a framework for QoS measurement would be very useful to facilitate the interworking between these networks and, therefore, to facilitate the migration of the current legacy networks, as it is explained in the following paragraphs:

If the legacy network is QoS-unaware, a good way is the use of a QoS measurement framework, able to evaluate some QoS parameters such as the loss ratio, the delay or the jitter.

On the other hand, if the network implements some QoS mechanism (such as DiffServ), the QoS framework could be used as well as to evaluate QoS parameters, to infer QoS mechanisms. In this approach, an Ambient Network could be able to model the performance of the legacy network by means of processing some measurements.

It should be noted, that the methodology to infer QoS capabilities in a legacy network should be built over the proposed framework to perform QoS measurements. In this way, the migration requirement related to the incremental introduction of new functionalities could be met.

6 Evaluation and verification

6.1 GTLP evaluation

Since the GANS protocol, in particular its transport layer part (GTLP), is meant to be used in a wide variety of scenarios ranging from simple interworking between devices to interworking between operator networks, the scalability of the protocol is of utmost importance. Thereby, in this section we present the evaluation of the GTLP protocol from scalability point of view. In section 6.1.1 the scalability of GANS is performed from the GTLP point of view. The scalability of the GTLP protocol is analyzed with respect to overhead, memory and processing requirements. This do also involve latency of the DEEP name resolution protocol (see 6.1.1); the DEEP protocol is evaluated separately in Section 6.1.2. The implementation of GTLP used for the scalability analysis is presented in Section 6.1.3; it represents a suitable platform for further evaluation of the GANS protocols in phase 2.

6.1.1 GTLP Scalability analysis

A scalability analysis was performed after the design and specification phases of the GANS protocol stack. The analysis is important since the GANS protocol is meant to be used in dynamic scenarios, executed in devices with different capabilities in heterogeneous networks (potentially from the simplest device to the most complex network), and has to support a different number of signaling applications (GSLPs). The purpose of the study was to evaluate the performance of the GANS protocol suite, and in particular the performance of GTLP from scalability point of view, which is the protocol supporting all signaling applications used between ANs, as mentioned before. In this context the study focused on the evaluation of processing and memory requirements at end-points, GTLP overhead and the scalability of DEEP name resolution mechanism.

In order to study the scalability aspects of GTLP, simulation models representing different AN scenarios were developed. GTLP was implemented into the NS-2 network simulator framework [42] for executing the simulation models. In the simulations different scalability related measurements were performed. The goal was to select input parameters which would influence the scalability of the protocol, and measure output parameters which could be used in the evaluation of the protocol. Based on the results obtained with measurements, an evaluation on the scalability aspects of GTLP was produced. Section 6.1.1.1 discusses properties of the simulation models, and Section 6.1.1.2 presents the executed simulation scenarios. Section 6.1.1.3 presents results of the evaluation work.

6.1.1.1 Simulation models

Regarding the evaluation of the GTLP, simulation models representing different scenarios in the AN environment were developed. In this context a simulation model is considered to comprise the network model of a specific scenario to be studied along with the scalability parameters to be varied and measured. In the following, different properties of the simulation models are considered. Simulation models for specific scenarios were developed based on these properties. Figure 9 describes two communicating ANs, Intermediate Nodes (IM1, IM2) and the GANS protocol stack. The simulation model has at least the following variable parameters, which are relevant for the scalability of GTLP:

Number of ANs
The number of ANs must be taken into account, because it affects how many GTLP messages will be sent between the communicating ANs, and in addition the state to be maintained inside of GTLP.

Number of GTLP sessions
It is important to consider how the number of GTLP sessions reflects the state to be maintained (memory) and processing to be required by GTLP.

GTLP characteristics

GTLP has internal characteristics which are important from the scalability aspects point of view. The frequency of GTLP messages, frequency of GTLP state refreshes, and transport mode selection (C-mode, D-mode) have an effect on the number of messages to be exchanged and involves processing at endpoints.

Topology

The topology between the communicating ANs will be taken into account, when studying the scalability aspects of DEEP, because it is important to consider how DEEP scales in terms of the number of intermediate DEEP-nodes.

By changing the above-mentioned parameters in a simulation model, information about the scalability of GTLP was obtained by measuring the following output parameters:

- Memory consumption of Message Routing State (MRS) table and Message Association (MA) table at end-points
- Processing of MRS and MA tables at end-points
- Scalability of DEEP name resolution mechanism
- GTLP overhead (the number and sizes of GTLP messages to be exchanged).

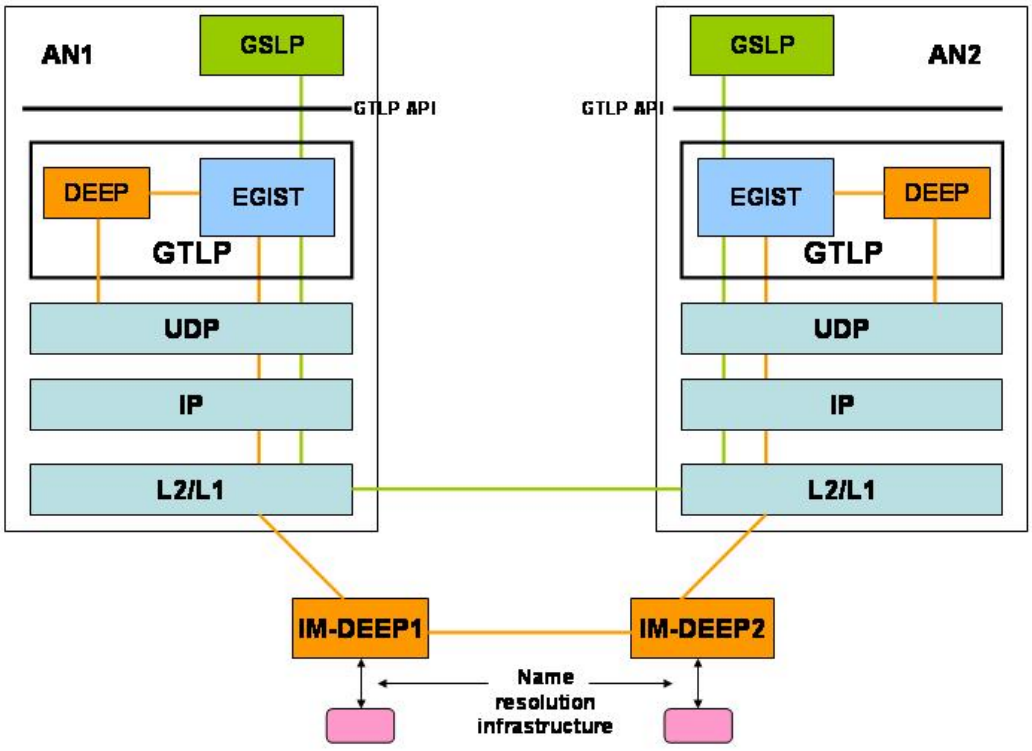


Figure 9: GANS protocol stack.

6.1.1.2 Simulation scenarios

6.1.1.2.1 Sending of GTLP messages

This scenario models communication between two ANs. At first a number of GTLP sessions were created between the ANs, and then GTLP messages were exchanged. The scalability parameters to be varied were the number of sessions to be created, the frequency at which the messages were sent and the GTLP transport mode. The parameters to be measured were Message Routing State (MRS) table and Message Association (MA) table memory

consumption, number of accesses to the state tables, and messages to be exchanged between the endpoints (GTLP overhead).

6.1.1.2.2 Refreshing of GTLP state

The scenario models a varying number of ANs refreshing state with a remote AN. Two procedures were evaluated: periodic refreshing and an update procedure. The scalability parameters to be varied were the frequency of refreshes, the number of ANs, the number of sessions and the probability of resorting to DEEP name resolution. The scalability parameters to be measured were GTLP overhead, the number of accesses to MRS table, and memory consumption of the MRS table.

6.1.1.2.3 DEEP name resolution

The purpose of this scenario was to study the scalability aspects of DEEP name resolution. In particular the latency of DEEP session setup was measured, when the number of intermediate DEEP-nodes, local name resolution delay, and Round Trip Time (RTT) were varied.

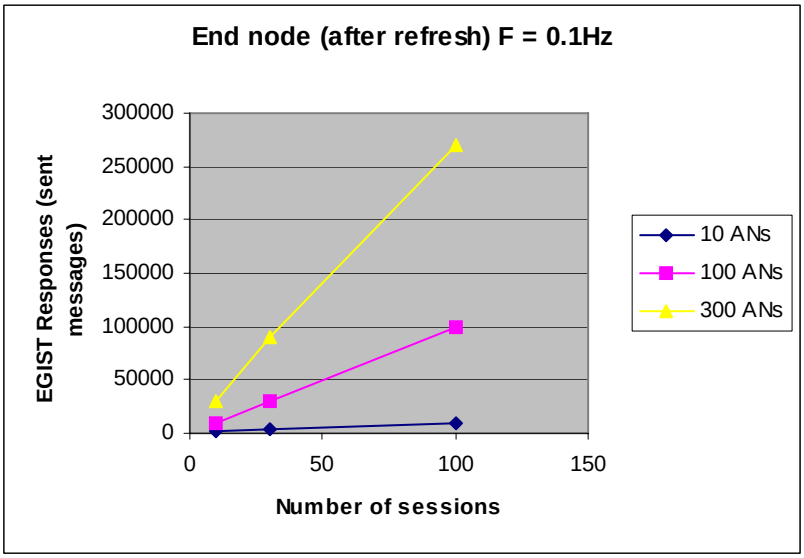
6.1.1.3 Evaluation of results

In this section the most important aspects of the GTLP scalability aspects have been described. EGIST and DEEP overhead, memory requirements, and processing requirements (the number of search operations) have been dealt with in their own subsections. For more detailed information of the results the reader is referred to the Technical Annex [63].

6.1.1.3.1 EGIST overhead evaluation

In [63] we have demonstrated that four procedures contribute to the overall EGIST overhead: the refresh procedure, the update procedure, the GSLP message sending procedure, and the message association establishment procedure. Each of these procedures has more or less influence in the overall overhead of the protocol and, therefore, has more or less impact in the scalability of the protocol. In the following we analyse the impact of each of these procedures in the overall overhead and conclude about the most critical procedure(s) from scalability point of view. In addition, we conclude about the scalability of the EGIST protocol. Herein, we present the analysis and evaluation of the EGIST overhead and provide some simulation and theoretical results. For further details about the results used as basis for the evaluation, the reader is referred to the technical annex of this deliverable [63].

In what concerns the refresh process the name binding states stored internally by the EGIST protocol, the ultimate conclusion is that the overhead introduced by the process is directly proportional to the frequency of refresh. From the simulation results, presented in [63] and in



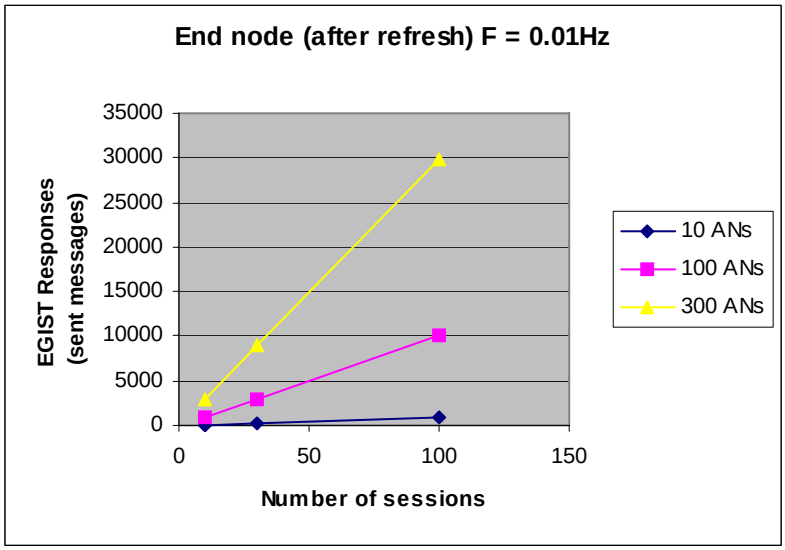
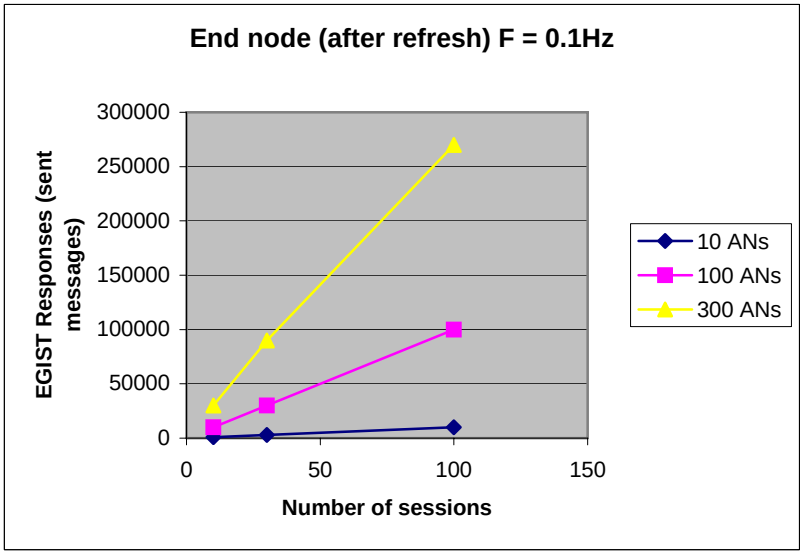


Figure 10 concerning the refreshing of GTLP state scenario, and as it was empirically expected, we conclude that a reduction of a power of 10 in the frequency of refresh represents a similar reduction in the overhead of the protocol. Then, the frequency of refresh needs to be carefully defined so that the protocol can be scalable. On the other hand, the overhead introduced by this process depends of course on the size of the signalling messages exchanged between the signalling endpoints. After the analysis made in [63], we have verified that the EGIST-Query and EGIST-Response messages considered in the simulations were too long, and therefore better results than the ones obtained in the simulations can be achieved, as shown in Figure 11. Furthermore, from the theoretical analysis performed in [63] we verify that the overhead introduced by these messages is mostly influenced by the symbolic names included in the MRI object. Therefore, we conclude that the GANS specification should be fixed in order to specify the symbolic names within the MRI object as TLV information elements, since this will contributed significantly to the scalability of the protocol, among other aspects.



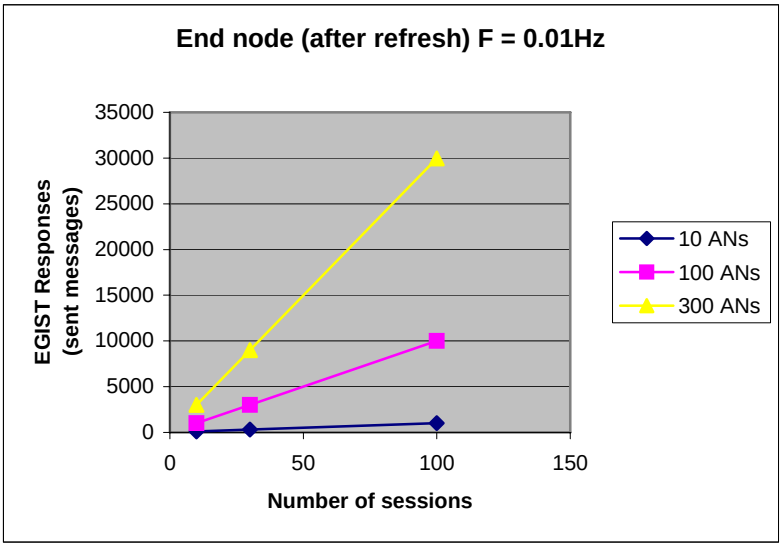


Figure 10: EGIST Responses messages sent by the end node in the refreshing scenario

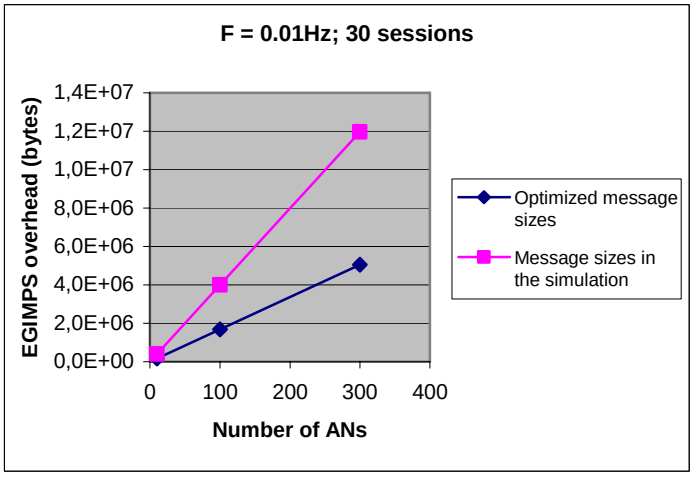


Figure 11: Comparison between theoretical calculations and simulation results concerning the EGIST overhead for the refresh procedure when the size of the signalling messages is optimized.

Concerning the update procedure, we can draw the conclusion that it is not expected to have significant impact in the overall EGIST overhead. This conclusion becomes clear concerning the simulations results presented in [63] for this procedure. From those results it can be seen that even if all ANs connected to the remote AN in the refreshing of GTLP state scenario perform an update procedure, which is the extreme scenario, the additional overhead introduced is not significant, when the frequency of refresh is 10 times the frequency of update (in fact, the frequency of update is envisioned to be 10 or even 100 times lower than the frequency of refresh). Thus, the update procedure is envisioned to introduce just residual overhead and, in that sense, is not expected to bring up scalability problems.

With respect to the message association establishment procedure, the overhead introduced is concentrated on the instant of creation of new message associations. Therefore, it is envisioned that the major scalability problems will come up when we concentrate on the interval of time where several signalling sessions can be set up between multiple ANs and a central AN, such as in the scenario described in refreshing of GTLP state scenario. In that situation, a huge amount of signalling traffic will be generated leading to significant overhead. Actually, the amount of traffic is in general directly proportional to the number of sessions created between pairs of ANs. Therefore, an increasing in the number of sessions will lead to

an increasing in the signalling overhead. However, in the GANS specification, the reuse of message associations already established between signalling endpoints is specified. In the simulations we have considered this property in the sending of GTLP messages scenario and proved that the message association reuse contributes to the scalability of the protocol, since a single message association can be used to support signalling information of several sessions established by the upper layer protocols. In spite of just considering signalling exchange between two ANs, this scenario allows us to extrapolate that in the general case, where more than two ANs are involved, the message association reuse contributes to a better scalability of the protocol, as it would be expected. Nevertheless, the percentage of message association reuse will not be 100% (i.e., a single message association supporting all signalling between ANs) in the general case, since that would require an AN to be supported by a single physical node that would also support all signalling endpoints inside an AN. Actually, within the Ambient Networks context, the signalling endpoints within an AN may be distributed across several physical nodes, which invalidates full message association reuse.

Regarding the message sending procedure, the frequency of GSLP messages sending is envisioned to be low in average. Typically, several GSLP messages will be exchanged at the beginning of a session, but afterwards zero or almost zero messages will be transferred between signalling endpoints. However, rather than the other procedures being analysed herein, the overhead introduced by this procedure is very depend on the actual signalling applications running over it. Therefore, we cannot draw any relevant conclusion here.

In conclusion, we may say that the relevant procedures from scalability point of view are the message association establishment and the refresh procedures. However, the potential scalability problems of EGIST protocol illustrated above are already solved by the GANS specification, as we have demonstrated. In fact, the message association reuse will contribute to the reduction of the overall overhead. On the other hand, concerning the refresh procedure, in spite of having assuming in the simulations that for each new GSLP session a new name binding state is created, this strong assumption may be relaxed if we consider the indirect communication scenario considered as an extension to the GANS base specification. In such scenario, a signalling gateway within each AN is defined, which is receiving the signalling from the multiple signalling endpoints inside the same AN, and is acting as the contact point from the outside point of view. In this scenario, it is actually possible to have a single name binding state installed in each signalling gateway, and therefore, a single periodical refresh must be performed between the ANs, instead of the multiple refreshes needed if direct communication between signalling endpoints was accomplished. The benefit of this approach from signalling overhead point of view is illustrated in Figure 12, which considers the EGIST overhead for the refreshing of GTLP state scenario. In fact, it can be confirmed that for the indirect communication scenario the signalling overhead in bytes is fully independent on the number of sessions established by each AN. Thus, the signalling overhead is significantly reduced in comparison with the scenario where the different endpoints inside each AN exchange signalling directly. Actually, this indirect communication approach will increase the message association reuse up to 100%, wherein a single gateway for each AN is defined.

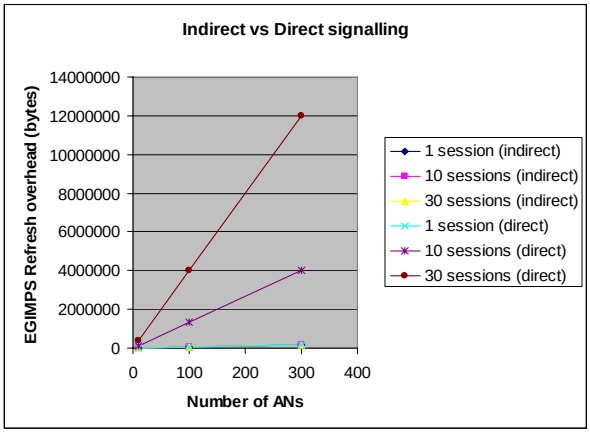


Figure 12: Indirect communication scenario vs direct communication scenario when message association reuse is applied.

In this context, there is actually a trade-off between signalling overhead and fault-tolerance, i.e., a trade-off between a decentralized and centralized approach. On the one hand, a decentralized approach will bring up the scalability problems aforementioned. On the other hand, the centralized approach has the disadvantage of a single point of failure. Then, the best approach seems to be the hybrid one, where multiple signalling gateways are available within each AN. That approach will overcome the single point of failure problem and will at the same time drastically reduced the signalling overhead in comparison with the decentralized approach. Therefore, the EGIST protocol will scale if we consider the indirect communication scenario, potentially with multiple signalling gateways, as a way to reduce the number of refreshes performed between ANs and increase message association reuse.

6.1.1.3.2 DEEP scalability

This section presents evaluation results of the scalability aspects of the DEEP protocol. The evaluation deals with DEEP overhead and scalability aspects of the name resolution mechanism.

Overhead in the Name Resolution procedure

The DEEP name resolution procedure is used when the GTLP source initiates communication with a communication endpoint (CEP) in the destination by using a symbolic name. In the procedure the symbolic name of the GTLP CEP is resolved to an IP address by exchanging DEEP EXPLORE and RESPONSE messages [25]. This causes overhead on the CEPs and the network.

When an AN is addressed with one symbolic name, DEEP has to be executed only once during the lifetime of the name binding, and is independent on the number of GTLP sessions. In that case the results show that the overhead increases in direct proportion to the number of ANs, and is not considered as significant. Further calculations show that the overhead can become significant, if the CEPs in the destination ANs are addressed with multiple symbolic names. In fact the overhead increases in direct proportion to the number of symbolic names multiplied by the number of ANs, as can be seen from equation 6.1.1, which gives the amount of overhead on a node (E=size of DEEP EXPLORE, R=size of DEEP RESPONSE, t=time, L=lifetime of the name binding state).

$$\text{DEEP\_overhead\_node\_bytes}(t)=\sum_{i=1}^{numberOfANs}\frac{numberOfCEPs_i * t * (E+R)}{L}, \qquad (6.1.1)$$

Overhead in the Update procedure

When the lifetime of a name binding state in GTLP must be changed, DEEP may be involved in the procedure [63]: GANS Specification]. Whether or not DEEP is used depends on local

configuration or policies [63]. If DEEP is involved in the Update procedure, DEEP messages have to be exchanged for each GTLP session to be updated. The amount of overhead is dependent on the number of sessions to be updated, number of ANs, and the frequency of updates as can be seen from equation 6.1.2, which describes the overhead on a node (P=probability of using DEEP in the Update-procedure).

$$DEEP\_overhead\_node\_bytes\_update(t)=\sum_{i=1}^{numberOfANs} numberOfCEPs_i * t * P * (E+R) * numberOfSessions * F$$

(6.1.2)

The results show that the use of DEEP as part of the update procedure can produce significant overhead on the network, when the number of ANs and sessions is increased.

Overhead overall

The overall DEEP overhead consists of the overhead from the Name Resolution and Update procedures given by equations 6.1.1 and 6.1.2. The results show that the overhead from the Update procedure is a dominating factor in the overall overhead. The total overhead increases significantly, when the number of CEPs is increased with the number of ANs and the number of GTLP sessions,

DEEP message size

The size of the DEEP EXPLORE and RESPONSE message depends on the size of the symbolic name and IP address encapsulated inside the messages. If the maximum size for the symbolic name is considered to be the size of a Fully Qualified Domain Name (FQDN), and the IP address the size of an IPv6 address, then the maximum size of both messages is under 300 bytes.

Scalability of DEEP name resolution mechanism

DEEP is a protocol, which is capable of integrating distributed name resolution mechanisms together [25]. DEEP has three important aspects which influence the scalability of the protocol, namely the delay of performing name resolution (NR) in intermediate DEEP nodes along the path towards the destination, the Round Trip Time (RTT) of DEEP messages, and the number of intermediate DEEP-nodes.

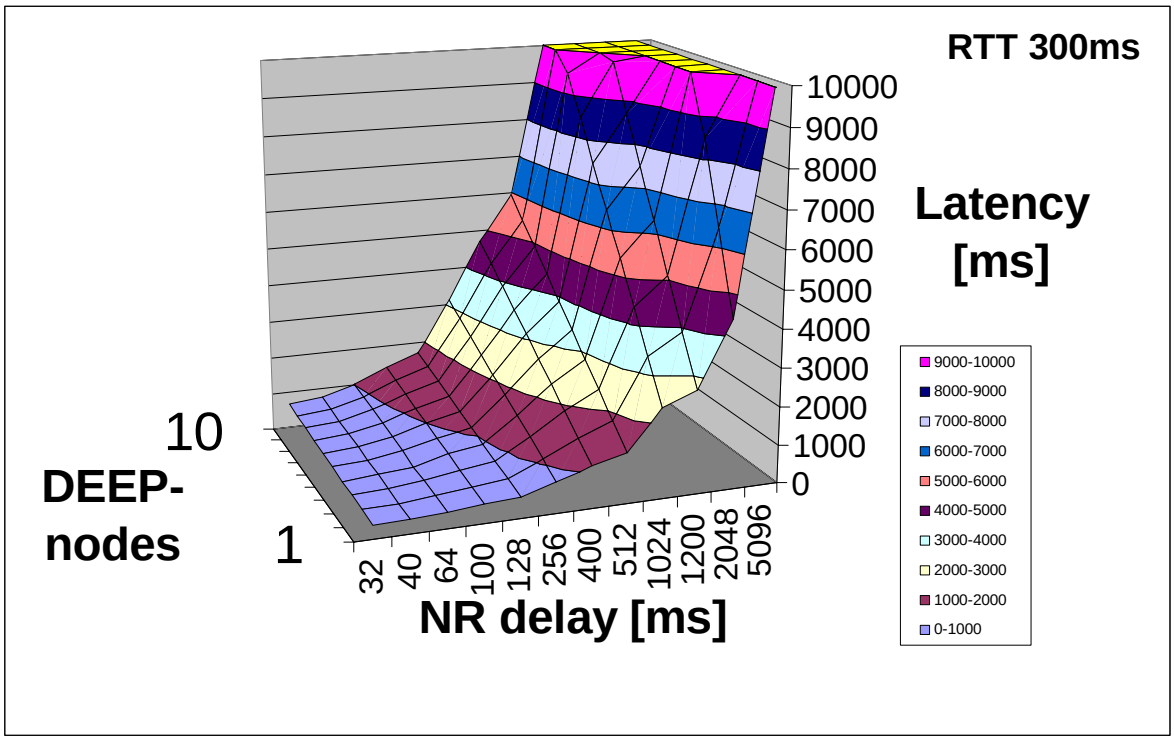


Figure 13: DEEP latency as a function of the number of DEEP-nodes and name resolution delay, when RTT is 300 ms.

In the study DNS was considered as one alternative for performing the name resolution in intermediate DEEP-nodes. Figure 13 shows as an example the scalability of DEEP latency with a RTT of 300ms, when the NR delay and the number of intermediate DEEP nodes is varied. As a summary of all the RTTs it can be stated, that if DNS is considered to be used with DEEP as a name resolution mechanism, and DEEP NR delay in intermediate nodes is assumed to be within the median DNS delays (40-100 ms), the DEEP latency is under 2.6 seconds for different RTTs and the number of intermediate DEEP-nodes up to 10 (see

Table 3). However if NR delays bigger than the median DNS delays are considered, the results show that DEEP latencies up to 10 intermediate DEEP-nodes are significantly bigger (maximum between 13.5-57.6 seconds).

In general criteria for DEEP NR scalability is difficult to be obtained, because the NR latency depends on the name resolution mechanism and topology of DEEP nodes in the Internet both of which are not known at the moment. This means that a generic scalability analysis for DEEP is difficult to be performed reliably based on the current specification of DEEP. Table 1 gathers the results of the DEEP name resolution latency calculations.

RTT	**300 ms**	**600 ms**	**1000 ms**	**1500 ms**
NR delay				
40-100 ms (DNS median)	**380-1400 ms (*100*DEEP-nodes+400*)**	**680-1700 ms (*100*DEEP-nodes+700*)**	**1080-2100 ms (*100*DEEP-nodes+1100*)**	**1580-2600 ms (*100*DEEP-nodes+1600*)**
100-1200 ms (DNS 90th percentile)	**500-13500 ms (*1200*DEEP-nodes+1500*)**	**800-13800 ms (*1200*DEEP-nodes+1800*)**	**1200-14200 ms (*1200*DEEP-nodes+2200*)**	**1700-14700 ms (*1200*DEEP-nodes+2700*)**
1200-5000 ms	**2700-56356 ms (*5000*DEEP-nodes+5300*)**	**3000-56656 ms (*5000*DEEP-nodes+5600*)**	**3400-57056 ms (*5000*DEEP-nodes+6000*)**	**3900-57556 ms (*5000*DEEP-nodes+6500*)**

Table 3: Summary of DEEP latencies for different RTTs and Name Resolution delays.

Memory Consumption

Memory is consumed to store the name binding state for signalling sessions and also for storing information relating to message associations (MAs) although the latter is applicable only for c-mode. A name binding is maintained per session in the binding state table (BST) at the end nodes. Assuming that each binding entry consumes B_1 units of memory, the amount of memory consumed at each end node is:

$$Total\_memory\_consumed = numberOfsessions * B_1 \qquad (6.1.3)$$

In the c-mode, a MA is setup at the start of the signalling session and an entry is created in the MA table at both ends for each session. Assuming that each entry is of the same size B_2, the total memory consumed by the MA Table is given by

$$Total\_memory\_consumed = numberOfsessions * B_2 \qquad (6.1.4)$$

The amount of memory consumed in storing an entry in either the binding table or the MA table depends on the implementation. The size of a name binding entry for our implementation is 538 bytes and the size of an MA entry for IPv4 is 53 bytes. We plot the theoretical and simulation results below.

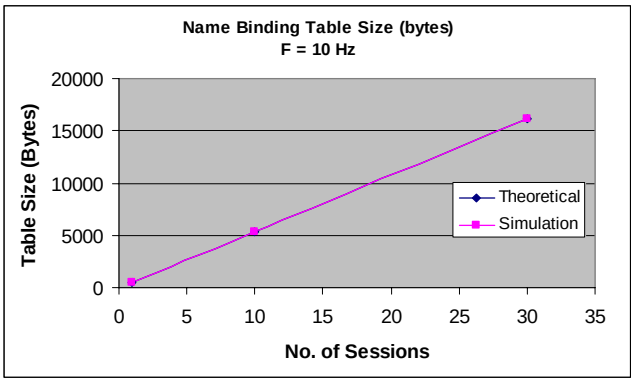


Figure 14: Binding table size comparison

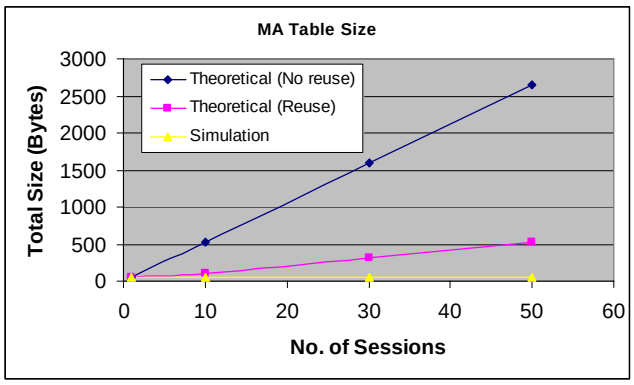


Figure 15: MA Table Size Comparison

The simulation and theoretical results for the binding table size are identical. Figure 15 shows the results for the MA Table. For the theoretical results, we consider two cases. In the first case, the sessions are established between a single source AN but different destination ANs. In the second case, when number of sessions is more than one, it is assumed that 5 sessions each are established between distinct source-destination pairs, which allows for reuse of MA. As we can see from the graph, the memory used in the simulation is constant despite an increase in number of sessions. This is because in the simulations all sessions were between the same source-destination pair. The results indicate that memory usage is dependent on the number of sessions and the size of symbolic names. Furthermore, reuse of message associations helps reduce memory consumption quite significantly.

Number of Search Operations

In c-mode as well as d-mode, the binding table is searched when a message is received from GSLP to check if a corresponding binding exists. Furthermore, in c-mode the MA table is also searched to check if an MA has already been established. In c-mode, an MA is required for session establishment. This requires a 3-way handshake between the two endpoints. The number binding table search operations is given by:

$$Num\_BST\_searches\_sender\_before = numOfSessions * 5 \quad (6.1.5)$$

$$Num\_BST\_searches\_receiver\_before = numOfSessions * 2 \quad (6.1.6)$$

The number of binding table search operations after session setup is:

$$Num\_MA\_searches\_after = \sum_{i=1}^{numOfSessions} F_{MS}(i) * T_s(i) \quad (6.1.7)$$

The sender and receiver both have to perform one MA table lookup during the session establishment phase. Therefore,

$$Num\_MA\_searches\_before = numOfSessions \quad (6.1.8)$$

After the session has been established, one search is performed per message at both endpoints. The graphs below show a comparison of selected theoretical and simulation results for the search operations in c-mode.

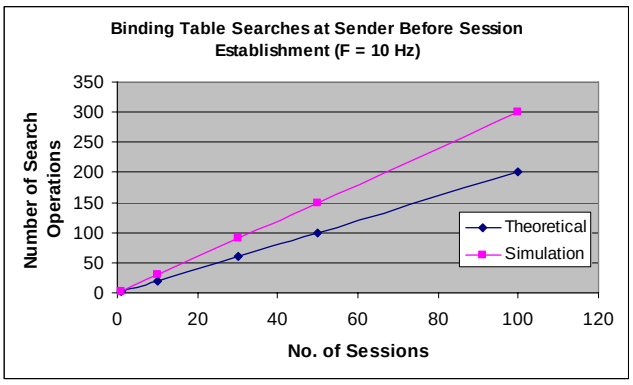


Figure 16: Comparison for search operations at sender before session setup (d-mode)

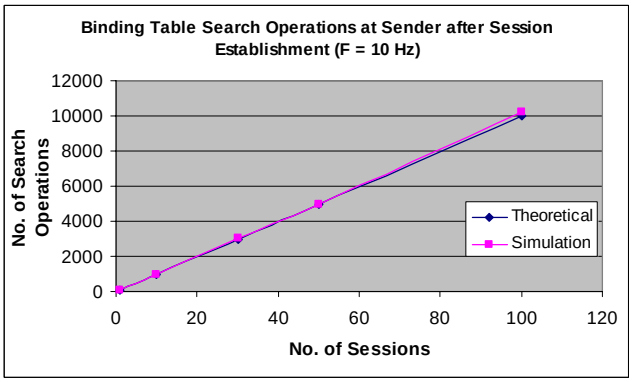


Figure 17: Comparison for search operations at sender after session setup (c-mode)

The results show a very close match between the theoretical and simulation results. However, the theoretical values for search operations on binding table before session setup are different from the simulation values as fewer search operations are required in the simulation because the MA is re-used.

The results show that the number of binding table searches is directly proportional to the number of sessions. However, the actual number of sessions depends on the specific scenario. However, because of re-use of existing MAs in some scenarios, the number of search operations is much less than the number predicted by mathematical model. Therefore, it has to be said that in situations where an AN has to establish a large number of sessions with other ANs simultaneously, it may put excessive load on the signalling endpoints.

In addition to the number of search operations, another important consideration is the search algorithm itself. The time taken to complete a search operation is obviously dependent on the number of entries in the table itself. However, even for the same number of entries, the time taken to search for an entry depends on the algorithm used. The absolute amount of time taken to search depends on a number of factors such as the processor speed, RAM size, operating system, quality of the implementation code etc. Therefore, absolute time is not a useful indicator of the algorithm speed. We can use the notion of algorithm complexity based on the 'big-oh' notation to compare different search algorithms.

In linear or sequential search algorithms, the list of items is traversed from start to finish for locating the relevant entry. In the worst-case $O(n)$ comparisons are required while $O(n/2)$ operations are needed on the average (here n is the number of entries in the list). Therefore,

the order of complexity for linear search algorithms is of the order of n. When a binary search algorithm is used, the order of complexity drops to *O*(log n). However, binary search can only be applied only on sorted lists. This puts extra complexity on list maintenance. In hash tables, average time to search for an element is *O*(1), while worst-case time is *O*(*n*). Thus, the number of search operations is meaningful only when seen in conjunction with the search algorithm because the latter is an equally important measure of the processing overhead.

6.1.2 Relation of DEEP and DNS

This section represents an example of how DEEP could be used with DNS to implement multi-step name resolution and to study which Resource Records are suitable for this and how they could be configured. Section 6.1.3.4.1 represents how these guidelines introduced herein are used in prototyping.

Considering what the signalling source is requesting, what information it has to offer for a name lookup and what are available DNS mechanisms to store information (Resource Records - RRs), there are multiple approaches how to use DNS. The source is asking a service in a specific domain by providing the domain (network) and service (FE) identities. Based on these two identities, “public” DNS should return the Fully Qualified Domain Name (FQDN) of a node in a specific domain. The signalling source should also be able to resolve this FQDN into an IP address or other locator according to the used addressing architecture to send a message. After the next node has received the message, it uses locally available DNS to resolve the communication endpoint contact address information based on delivered identities; domain and service. After this, the message is sent to the communication endpoint for further processing and no further name resolutions are involved.

Perhaps the most suitable type of RR is the SRV RR that is specified in [2] and it is meant to be used by hosts to request a specific service for a specific domain. Its RR type code is 33 and its format is as follow:

_Service._Proto.Name TTL Class SRV Priority Weight Port Target

_Service defines the service name according to [50] or alternatively it can be locally defined. _Proto defines preferred protocol and Name defines the target domain. TTL and Class parameters are used according to [39]. Priority is defined for each RR to be used by a client when multiple RRs are returned as response. Weight value is further used in selection process when multiple RRs have equal priority. Port number defines what port is bound to the service in the target host that is defined by Target parameter; target host’s FQDN.

How multiple SRV RRs for the same service in a specific domain could be used based on priorities and weights is omitted, and instead generic guidelines to use of SRV RR are shown to represent how SRV RR can be used to implement the name resolution involving multiple name resolution points.

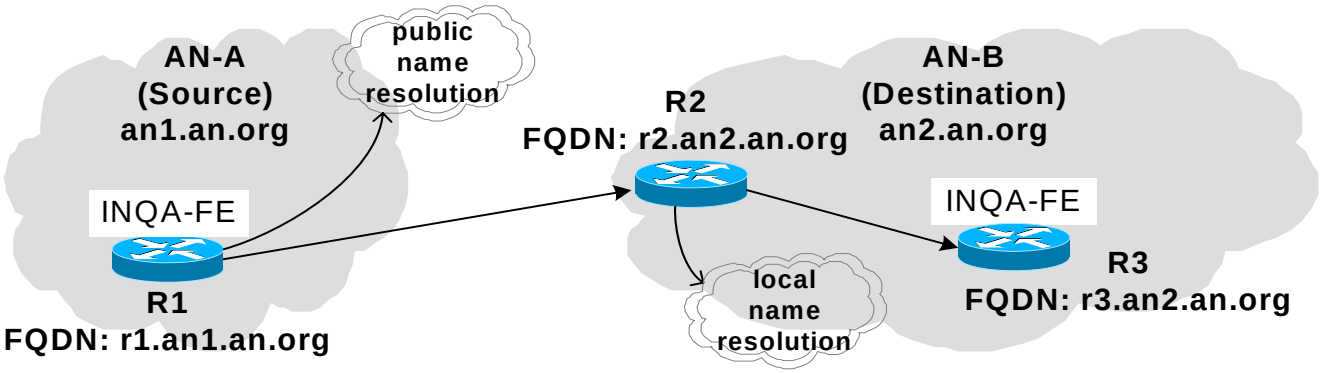


Figure 18: An example of the DNS use in case of distributed name resolution.

Figure 18 depicts an example how DNS can be used to support DEEP incremental name resolution. There are two ANs; source AN (AN-A) and destination AN (AN-B). The source has *an1.an.org* domain name and correspondingly the destination has *an2.an.org*. The signalling source node in AN-A is R1 and its FQDN is *r1.an1.an.org*. The signalling destination R3 is located in AN-B and its FQDN is *r3.an2.an.org*. The intermediary involved is R2 located at the edge of AN-B and its FQDN is *r2.an2.an.org*. R1 has access to public name resolution via its local name resolution services. To be more specific, this name resolution does not need to be literally public, but its scope should be wide enough to include AN-B's RRs. R2 uses its local name resolution and there is no need to provide access to this name resolution to the outside.

Because there are no AN specific naming conventions available, we decided to use a new type of URL – AN URL. It follows the format of SIP URL but its parts have AN specific meaning and the '@' delimiter character is replaced with '.';

AN URL: <Service_type>.<AN_domain>

Service_type could refer either to the type of FE or alternatively it could refer to a specific control function. The latter case supports the scenario, in which a specific type of FE implements multiple control functions. AN_domain specifies the destination AN, in which Service_type is valid. The scope of AN_domain specifies the whole communication session scope and Service_type needs to be valid only within a specific AN_domain. Identities scope's also represents how the name resolution should be done and where it can be done. In our example, INQA-FE implements QoS control signalling application protocol.

INQA-FE in R1 requests a message transfer service and it addresses the destination point as QoS.an2.an.org. The local name resolution service in R1 is implemented using the DNS and it performs a DNS name lookup using public name resolution as illustrated in Figure 18. This lookup has following format:

QNAME=_qos._gans.an2.an.org, QCLASS=IN, QTYPE=SRV

The result will be the IP address of R2, protocol is GANS and port number is 19001 according to the following SRV RR for *an2.an.org*:

```
$ORIGIN an2.an.com.
      @               SOA r2.an2.an.org. admin.an2.an.org. (
                          <these parameter values are omitted> )
                      NS r2.an2.an.com.
      ; All queries for _GANS protocol are resolved to the same node
      ; independently on a related service.
      *._gans         SRV 0 1 19001 r2.an2.an.org.
      an2             A <IP address is omitted>
      ; NO other protocols are supported
      *._tcp          SRV 0 0 0 .
      *._udp          SRV 0 0 0 .
```

Note that SOA (Start Of Authority) RR [39] could be configured differently depending on a specific AN configuration. Also, it was assumed that R2 provides name service to the outside as presented by NS (Name Service) RR [39].

This SRV does not reveal to the outside what are currently supported services in the AN domain – *an2.an.org*, instead it maps all name lookups for any service to a specific port number in *r2.an2.an.org*. After R1 has received the R2's FQDN as the name lookup result, it uses a DNS name lookup to resolve this name to IP address according to the standard DNS. Then it sends a DEEP message to R2 carrying the original AN URL (QoS.an2.an.org). Once R2 has received the message, it uses locally available name resolution service to perform the name resolution within the AN to find out how QoS service is currently configured. This name resolution is implemented using the DNS and *my_internal_dns.an2.an.org* node provides it. This DNS service is only for the internal use and all supported services in the AN are

configured using SRV RRs providing a mapping between a specific service type like QoS and IP address, protocol and port number. A SRV for QoS has following information:

```
$ORIGIN an2.an.com.
      @               SOA r2.an2.an.org. admin.an2.an.org. (
                          <these parameter values are omitted> )
                      NS my_internal_dns.an2.an.com.
      my_internal_dns A <IP address is omitted>
      ; All queries for QoS service type are resolved to the R3 node.
      _qos._gans      SRV 0 1 12345 r3.an2.an.org.
      an2             A <IP address is omitted>
      ; NO other GANS based services are supported
      *._gans         SRV 0 0 0 .
      ; NO other protocols are supported
      *._tcp          SRV 0 0 0 .
      *._udp          SRV 0 0 0 .
```

After R2 has performed the name resolution resulting *"r3.an2.an.org"* FQDN, it performs another DNS name lookup to resolve R3's IP address and then it replies the results back to R1. The result contains the IP address of R3, protocol is GANS and port number is 12345.

Alternative method would be to reveal a basic set of services that is always supported; for example if we consider that QoS and Composition are always supported and others are optional that may be supported. In the following SRV RR, R2 is taking care of both QoS and Composition service resolution, but also different nodes could be configured to do so. QoS service uses port number 19001; Composition uses port number 19002 and other services use port number 19003.

```
$ORIGIN an2.an.com.
      @               SOA r2.an2.an.org. admin.an2.an.org. (
                          <these parameter values are omitted> )
                      NS r2.an2.an.com.
      ; All queries for QoS and Composition service type are resolved to the
      ; R3 node.
      _qos._gans          SRV 0 1 19001 r2.an2.an.org.
      _composition._gans  SRV 0 1 19002 r2.an2.an.org.
      an2             A <IP address is omitted>
      ; All other GANS based services are resolved as follow
      *._gans         SRV 0 1 19003 r2.an2.an.org.
      ; NO other protocols are supported
      *._tcp          SRV 0 0 0 .
      *._udp          SRV 0 0 0 .
```

This example presented one possible way to use the DNS to support incremental name resolution with GANS. There are also other possibilities involving different RRs, for example NAPTR RRs [36] can be used to hold service specific contact addresses and URLs. When deciding what type of RR to use, it should be reminded that the result should include also other communication parameters besides IP address like port number.

In this example, it was only considered how to do a single resolution involving multiple name lookups. But it is equally important to consider how well the dynamic nature of ANs can be supported by name resolution mechanisms. In case of DNS, this means for example how caching works so that it contains only valid information at any time, how configuration changes are distributed with zones files, etc. Also when two ANs compose, how this new name is created and injected to the name resolution service and how long it takes before it is propagated through the name resolution infrastructure are also important things to be considered. We used only two levels of indirection that was provided by the recursive use of

SRV RRs. There are certain limitations when using only SRV RRs or NAPTR RRs and these are outlined in RFC 3958 [31].

In our example, only GANS was supposed to be used, but considering that legacy protocols could be used instead of GANS when applicable, we should also consider how we can support the use of multiple protocols (GANS and legacy protocols) in RRs and as presented, use of SRV RRs supports this.

The relation between DEEP and existing name resolution systems like DNS is such that DEEP relies on use of these name resolution systems, instead of replacing them. Moreover, DEEP name resolution does not restrict the use of any subsequent resolutions after the DEEP resolution is finished. For example a successful DEEP name resolution may result a host identifier, which is then resolved by using DNS name lookup services to find out an IP address of related rendezvous server and this resolution is transparent to the DEEP. This relation is depicted in figure below; DHT refers to Distributed Hash Table based name resolution mechanisms and LDAP refers to Lightweight Directory Access Protocol based directory mechanisms that can be used for name resolution purposes.

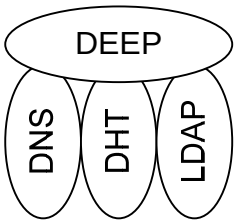


Figure 19: Relation between DEEP and existing name resolution systems

As conclusion, it can be said that the DNS is suitable name resolution mechanism to be used for multi-step name resolution assuming that proper types of RRs are used. Further references considering on how to use different types of RRs together to implement additional indirections were provided. There are no explicit rules defining how different name resolution domains should be used to implement symbolic names, because this depends on many things like what types of names are in question and what kind of information is revealed to the outside. Perhaps the most important thing is to understand how use of multiple resolution points in sequential manner affects on information visibility and scope. Additionally, we should not forget to consider on name resolution mechanisms characteristics while deciding on how to deploy them. The things like how fast a name resolution mechanism is able to update name-address mappings and how supported caching mechanism affects on dynamic updates are important things to consider among other things. In our examples, we considered only the situations, in which a symbolic name was resolved into an IP address.

6.1.3 Prototyping

6.1.3.1 Introduction

Development of the GTLP prototype is intended to be a *proof-of-concept* exercise. The prototype is used to validate and verify the design and specification of GANS in general and GTLP in particular. It could also possibly be used to refine the original GANS design if any problems or bottlenecks are identified during the development and testing of the prototype. As the GTLP is derived from the GIST protocol, the prototype design has been based on the GANS specification as well as the GIST specification.

The GTLP prototype software is based on the open-source implementation of the GIST protocol from the University of Gottingen. The GIST software has been modified to include the extensions to GIST proposed as part of the GANS design. In the following sections, we provide the functional specification of the GTLP prototype, followed by the software design specification.

6.1.3.2 Prototype Function Specification

6.1.3.2.1 Feature and Requirement Analysis

The primary objective of the GTLP prototype development is the validation and verification of GANS design. Therefore, it is important to identify the requirements for the prototype. As the GTLP implementation is based on the GIST prototype, we only consider the extensions required to be made to the GIST software. In the following, we list a set of requirements and features for the GTLP prototype.

6.1.3.2.1.1 Message Transport

In accordance with the GTLP design, the prototype provides message transport services to diverse signalling applications. As GTLP is backward compatible with the NSIS Transport Layer Protocol (NTLP), NSIS Signaling Layer Protocols (NSLPs) must also be supported in addition to GSLPs. Furthermore, both single-hop and multi-hop signalling applications are supported so that it can be used by single-hop oriented applications such as INQA-GSLP or multi-hop applications such as QoS-NSLP. In order to support 'path-decoupled/single-hop' applications like INQA-GSLP, the newly defined Message Routing Method will be implemented in the GTLP prototype. In cases where the new MRM is used, it must be ensured that the Router Alert Option (RAO) is not used. No changes are needed in the message routing state table as the MRM object is encapsulated inside the MRI object which is independent of the MRM type.

6.1.3.2.1.2 Addressing

GTLP supports the use of both IP addresses as well as symbolic names by signalling applications for addressing destination endpoints. This constitutes a significant difference between GANS and NSIS where only IP addresses can be used. For example the INQA-GSLP can address a signalling peer by its symbolic name. In contrast, the QoS-NTLP always addresses the destination by IP address. The GTLP module in the prototype must be able to accept and process messages received that address destinations with abstract symbolic names. As the GANS design does not place any restrictions on the type and format of the names, the prototype must be able to handle names of different formats and sizes.

6.1.3.2.1.3 Name Resolution

To support the use of symbolic names by the GSLPs, a name resolution service based on the Destination Endpoint Exploration Protocol must be implemented to map the symbolic names to the corresponding IP contact address. Once a name has been resolved, the corresponding name binding must be stored for future reference at least for the duration of the signaling session at the start of which the resolution was performed. Furthermore, name binding update procedure must be implemented to ensure that the binding stays valid even when the contact address changes occur during a session.

6.1.3.2.1.4 External Interfaces

The prototype will implement the GTLP API which serves as the service interface to signalling applications. This interface will be used to send and receive messages from the applications and also for exchanging control information such as error reports. This will be the only external interface. As there is very little difference between the GTLP and GIST APIs, the latter can be re-used as it.

6.1.3.2.1.5 Internal Interfaces

In addition to the GTLP interface, the prototype will implement a set of internal interfaces to enable interactions between different components of the GTLP. The DEEP interface will be used for message passing between DEEP and EGIST. The interface defined in the GANS specification should be implemented. In addition, an interface is also required between DEEP

and the local name resolution infrastructure so that symbolic names received in DEEP EXPLORE messages may be fully/partially resolved.

6.1.3.2.1.6 Backward Compatibility with NSIS

The GTLP prototype will be fully backward-compatible with NSIS. More specifically, the same service interface will be presented to all signalling applications, whether they are GSLPs or NSLPs. Furthermore, the message transport service should not make a distinction between GSLP and NTLP messages.

6.1.3.2.1.7 Integration with signaling applications

The prototype has to interwork with diverse signaling applications. Therefore, it must provide a mechanism to allow these applications to connect to the GTLP module. As the signaling applications are expected to be developed independently of the GTLP, it is essential that this mechanism must be generic. Furthermore, the GTLP module must be able to identify the different signaling applications that reside over the GTLP API.

6.1.3.2.2 Functional Structure

The functional specification of the GTLP prototype was described in the previous section. Based on the requirements and the GANS specification, the functional structure of the prototype was developed as illustrated in Figure 20.

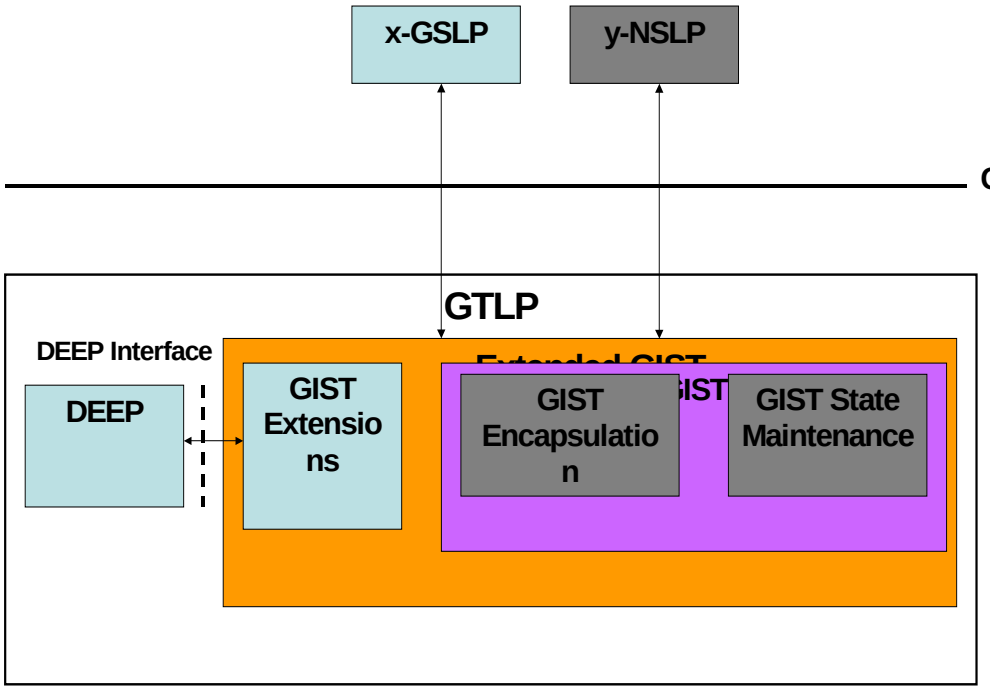


Figure 20: Functional structure

As can be seen from the figure, the prototype structure is based on the GIST implementation and it reuses all existing components of the latter. The two main functional blocks of are Extended GIST and DEEP. Furthermore, the GTLP API which is the interface through which GTLP offers its services to diverse signalling applications such as INQA-GSLP and QoS-NSLP etc.

6.1.3.2.2.1 Main Components

The primary function of EGIST is to provide session management and message transport services to different signaling applications. As shown in the figure, EGIST includes a set of extensions to the functionality provided by GIST [23]. Note that the figure is simply a logical

view and some extensions may actually be embedded in GIST itself. The extensions are required to support the additional requirements created by the signalling applications proposed for Ambient Networks, as described in Section 6.1.3.2.1. This includes the use of symbolic names to address destination endpoint and support for *path-decoupled single-hop* applications. The latter is supported by introducing the new MRM to GIST. In this case, the extension is going to be embedded into GIST state maintenance module rather than as a standalone part.

To support addressing using symbolic names, changes are required to the GIST state machine so that when a message with symbolic names is received, the name is resolved first before the message is processed any further. Name resolution is performed using DEEP. It is used to explore the destinations that are addressed by symbolic names. When a request is received from EGIST, the DEEP module first uses the local name resolution infrastructure to see if a binding is available. If no such binding exists, then DEEP EXPLORE-RESPONSE exchange is triggered. The result of the name exploration is then sent to EGIST which is then responsible for further actions.

6.1.3.2.2.2 Interfaces between main components

Based on the functional structure described above, two interfaces are implemented in the prototype. The first one is the GTLP API which is an external interface in the sense that it is visible to external entities such as GSLPs. The DEEP interface shown in Figure 20 is an internal interface and it is not accessible from outside.

6.1.3.3 Prototype Design Specification

6.1.3.3.1 Software Structure

6.1.3.3.1.1 Software architecture

The software architecture of the prototype is illustrated in Figure 21. It shows the different software components of the prototype and also the way they are connected to each other.

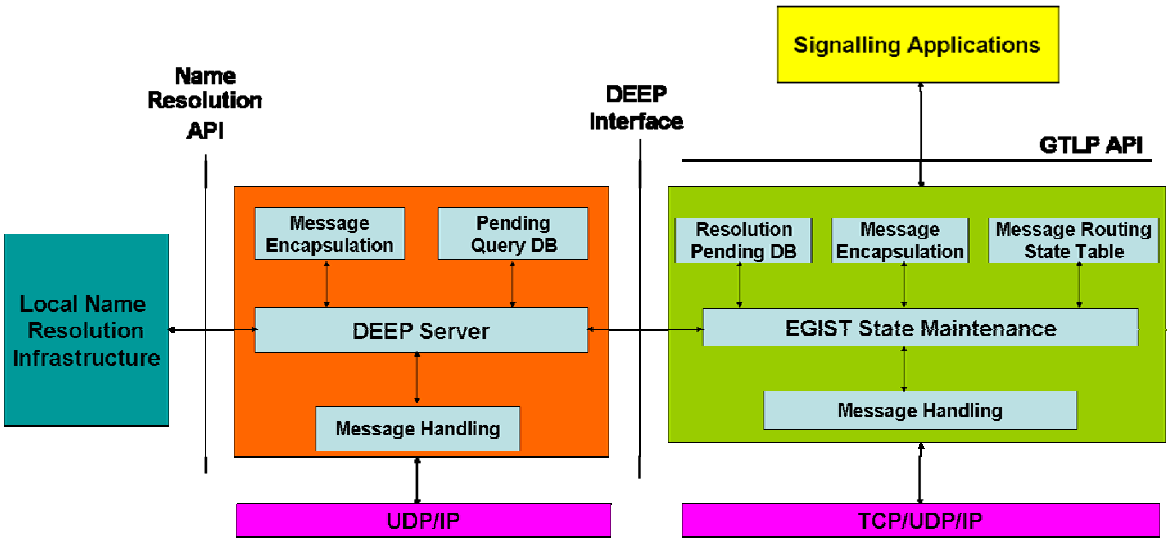


Figure 21: Software Architecture

There are three main modules in the prototype. First, there is an EGIST module which provides session management, message encapsulation and transport services to the applications that comprise the signalling application module. The DEEP module provides name resolution services to the EGIST module but it is not visible to the signaling applications. The DEEP module includes an API to query the local name resolution infrastructure when resolving abstract names. It must be noted that although the name resolution infrastructure by

itself is not part of the GTLP prototype. The signalling applications module consists of diverse signalling applications that use GTLP when setting up sessions with peers in other networks.

Each of these components is described in more detail below.

6.1.3.3.1.2 Software components

Extended GIST Module

The extended GIST module implements the transport layer functions described in the GANS Specification. As shown in Figure 21, this module consists of five smaller sub-modules. The core of EGIST module is the State Maintenance component which implements the EGIST state machine. The Message Handling part is responsible for processing incoming messages from remote endpoints before they are handed over to the state machine. The Message Encapsulation part is needed for creating different types of EGIST messages, such as Query, Response, and Data etc. The Message Routing State (MRS) Table is the database where routing information for each session is stored. An entry is created in this table at the start of a new session and it is periodically refreshed during the lifetime of this session by the EGIST State Maintenance component. Finally, the Resolution Pending database is used to keep track of pending name resolution requests made to DEEP. This database ensures that responses received from DEEP can be matched to pending requests. Furthermore, it helps avoid sending multiple requests to DEEP for resolving the same name.

DEEP Module

The DEEP module embodies all the functionality required to handle requests from the EGIST module for resolving abstract names. The engine of this program is the DEEP Server which is assisted by the other functional blocks to provide name resolution services to EGIST. It connects with EGIST using the DEEP interface. The interface is used by EGIST to send resolve queries to DEEP and it is also used by DEEP to send the results of the name resolution procedures, irrespective of whether the outcome was successful or not. Like the EGIST module, the DEEP module has Message Handling and Message Encapsulation components. It also maintains the so-called Pending Query database which contains information on names for which DEEP signaling is already underway. In addition, this module has an interface that is used to query the local name resolution infrastructure when resolve requests are received either from the EGIST module or from remote DEEP nodes (in the form or EXPLORE messages).

Signaling Applications Module

The signaling application module consists of a set of signaling applications. Since many different applications are being designed and specified, it is not possible to implement all of them in the prototype. Therefore, only a few of them are included in this module. In particular, a simple ping-based application and the more sophisticated INQA-GSLP module. The latter is being developed as a separate entity in Task3.2.

Class Diagram

The GTLP prototype has been developed using the C++ programming language. The class diagram of the prototype is shown in the figure below.

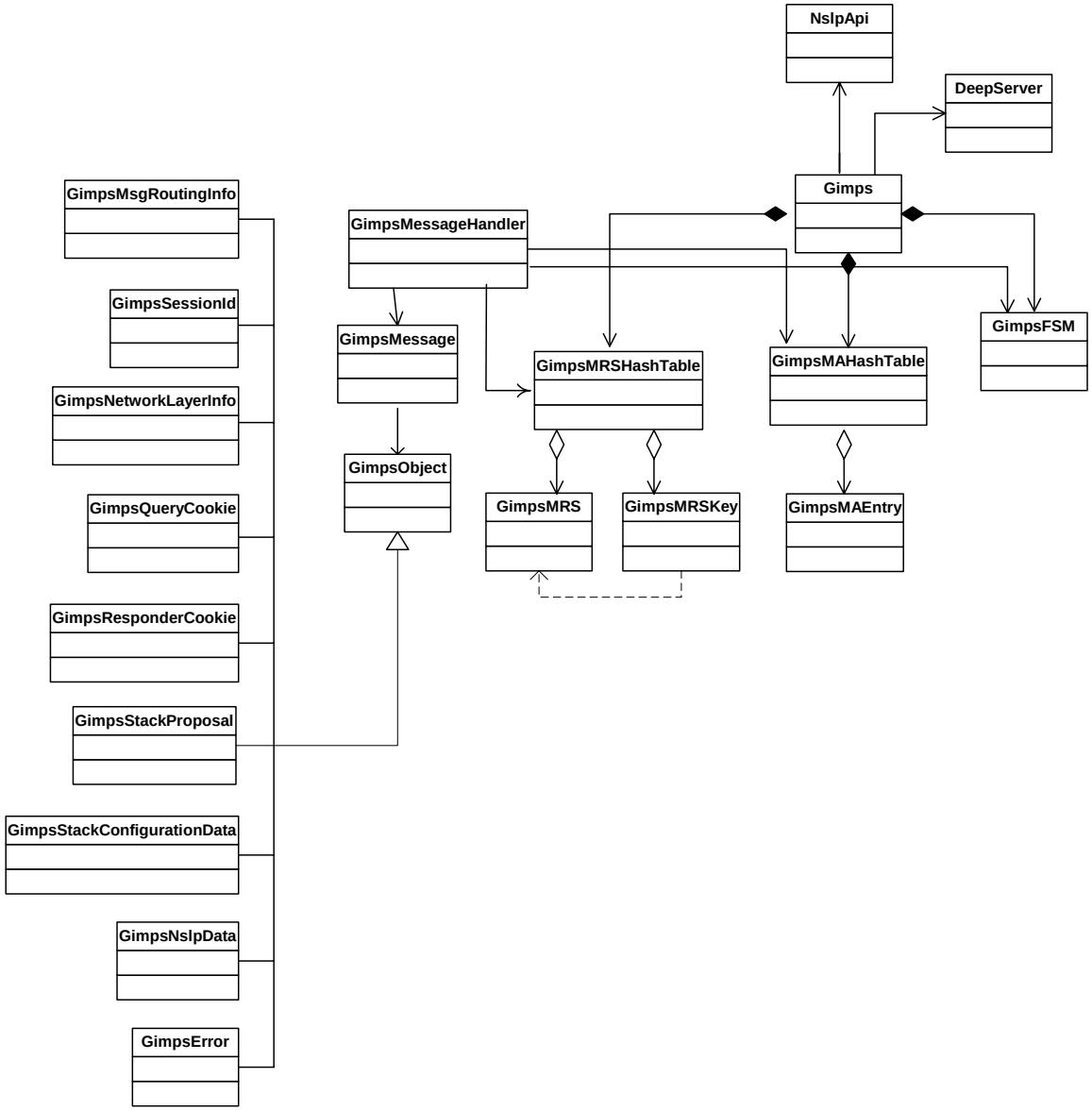


Figure 22: Class diagram of GTLP prototype

6.1.3.4 Testing and Validation

The prototype has been tested under a few test scenarios. Although the tests carried out are not exhaustive, they have been useful to check for compliance with the requirements described in Section 6.1.3.2.1. In this section, we briefly describe the test scenarios and provide signaling diagrams for each of them.

6.1.3.4.1 DEEP test scenario

The DEEP test scenario was designed to verify the DEEP module of the prototype. The scenario is shown in the figure below.

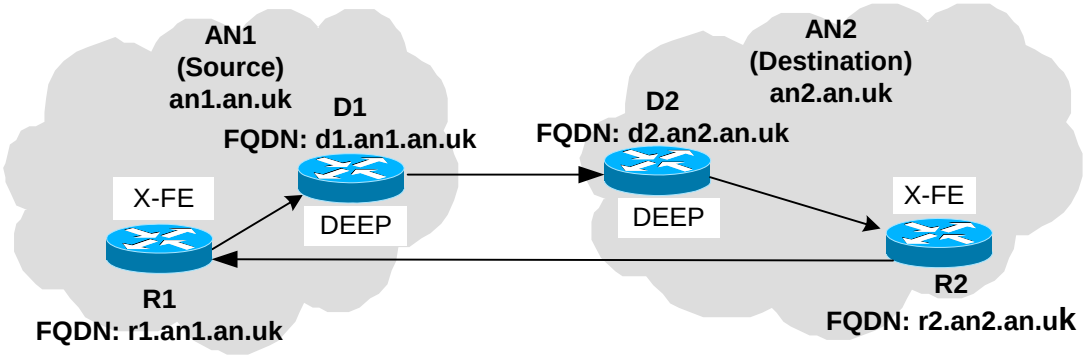


Figure 23: DEEP test scenario

The scenario above comprises two Ambient Networks, AN1 and AN2. Each of the AN has a dedicated DEEP node (labelled D1 and D2, respectively) which acts as forwarder of DEEP messages to external DEEP nodes. Furthermore, the figure shows that both the networks have a node each representing a functional entity, X-FE (labelled R1 and R2, respectively) that represents some abstract control function of the ACS. The networks may also have other FEs but they are shown in the figure for the sake of simplicity. The networks have their own DNS domain names and also maintain DNS servers that have been omitted from the picture. It is assumed that when the networks come into contact, they exchange information about the respective DEEP nodes (D1 and D2).

When the signalling application corresponding to X-FE in AN1 wants to set up a signalling relation with its peer in AN2, it creates a message which addresses the destination using the symbolic name *xfe.an2.an.uk*. When this message is received by the local EGIST module over the GTLP API, first the MRS table is searched to determine if there are any existing sessions with the destination. Since this is a new session, no entry is found. Then, the 'Resolution Pending' database is checked to see if the name is already being resolved by DEEP. Finally, the EGIST module uses the DEEP interface to send a name resolution request to the local DEEP module. DEEP also checks it own 'Pending Query' database to ensure that the name is not being resolved currently. Then the DEEP module creates an EXPLORE message with *xfe.an2.an.uk* in the 'Symbolic Name' field of the message and sends it to node D1. We assume here that all the other nodes in AN1 know the contact address of D1 and the same holds true for AN2 as well. When the message is received by D2, it uses queries the local DNS infrastructure for the symbolic name and this response is in the form of the address of D2. Therefore, the EXPLORE message is forwarded to D2. Upon receiving the message, D2 uses its local DNS server to find out that the symbolic name refers to node R2 in the network and forwards the message to it. Finally, R2 sends a RESPONSE message directly to node R1 providing its own contact address. When the message is received by the DEEP module in R1, the name binding is passed on to the EGIST module which can then start setting up the signalling session. The signaling flow diagram for the scenario is shown in the figure below.

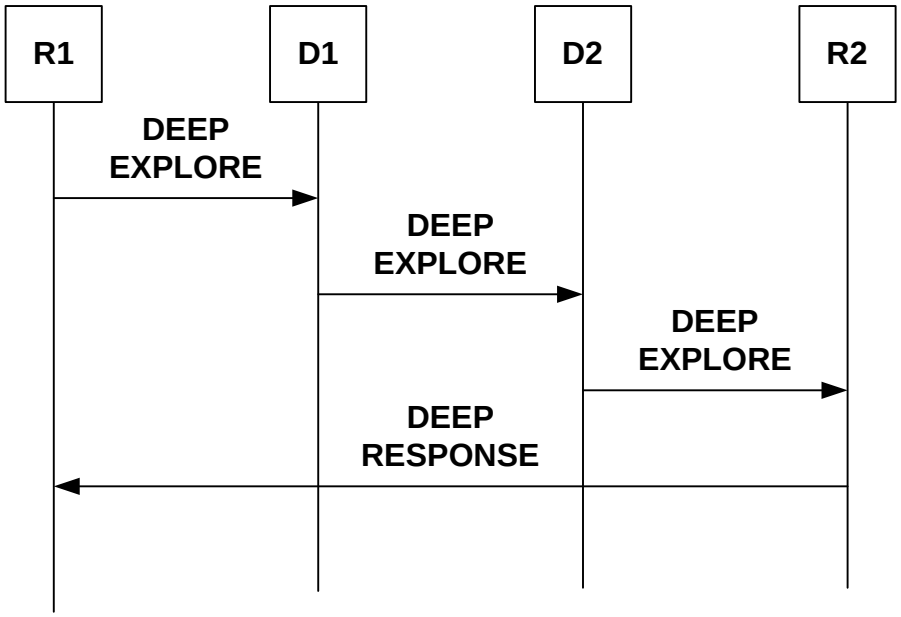


Figure 24: Signaling sequence for DEEP scenario

6.1.3.4.2 EGIST Test Scenario

The second test scenario has been designed to test the functionality of EGIST. The figure below illustrates the scenario.

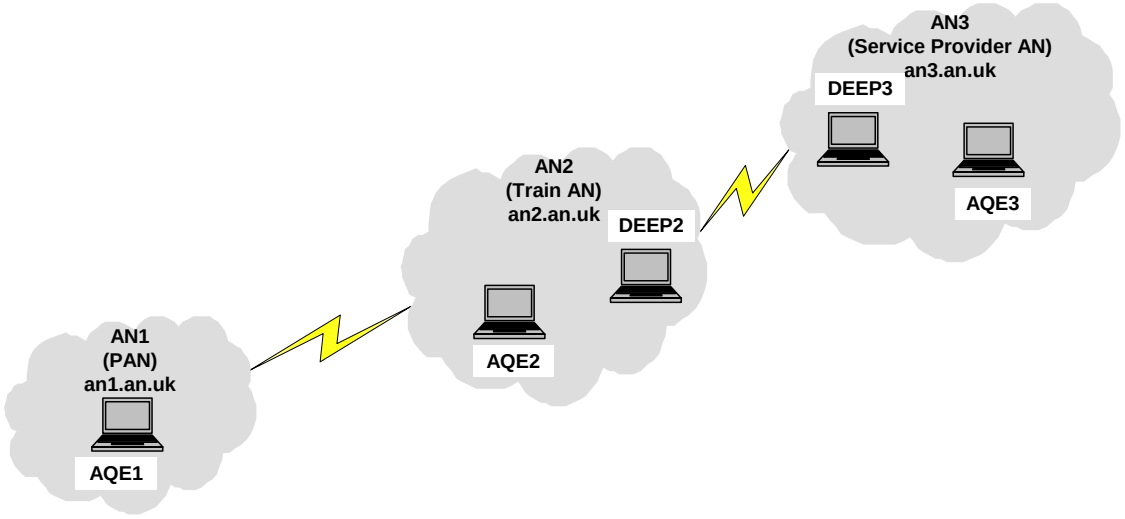


Figure 25: EGIST test scenario

The scenario has three Ambient Networks. AN1 represents a PAN, AN2 represents a train network while AN3 represents a service provider. Each of the networks has a node that represents the INQA-FE (labelled as AQE1, AQE2 and AQE3 respectively in the figure). Furthermore, AN2 and AN3 have a dedicated DEEP node (shown as DEEP2 and DEEP in the figure). The role of the DEEP nodes is same as their counterparts in Figure 24. The networks are shown as sub-domains of the *an.uk* domain. It is assumed that each network has its own local name resolution infrastructure in the form of DNS servers. The AQE nodes all use the INQA-Protocol to establish SLS agreements with each other. INQA-P is a signalling application that uses the services of GTLP. The scenario is used to test the session management and transport services provided by EGIST to INQA-P. In the scenario, the signaling between AQE2 addresses AQE3 using the symbolic name aqe.an3.an.uk whereas AQE1 uses the IP address of AQE2 for INQA-P signalling. This is intended to show that EGIST can work with both IP addresses and symbolic names. The figure below shows the

signaling sequence corresponding to a use case where bilateral SLSes are established first between AN3 and AN2 and then between AN1 and AN2.

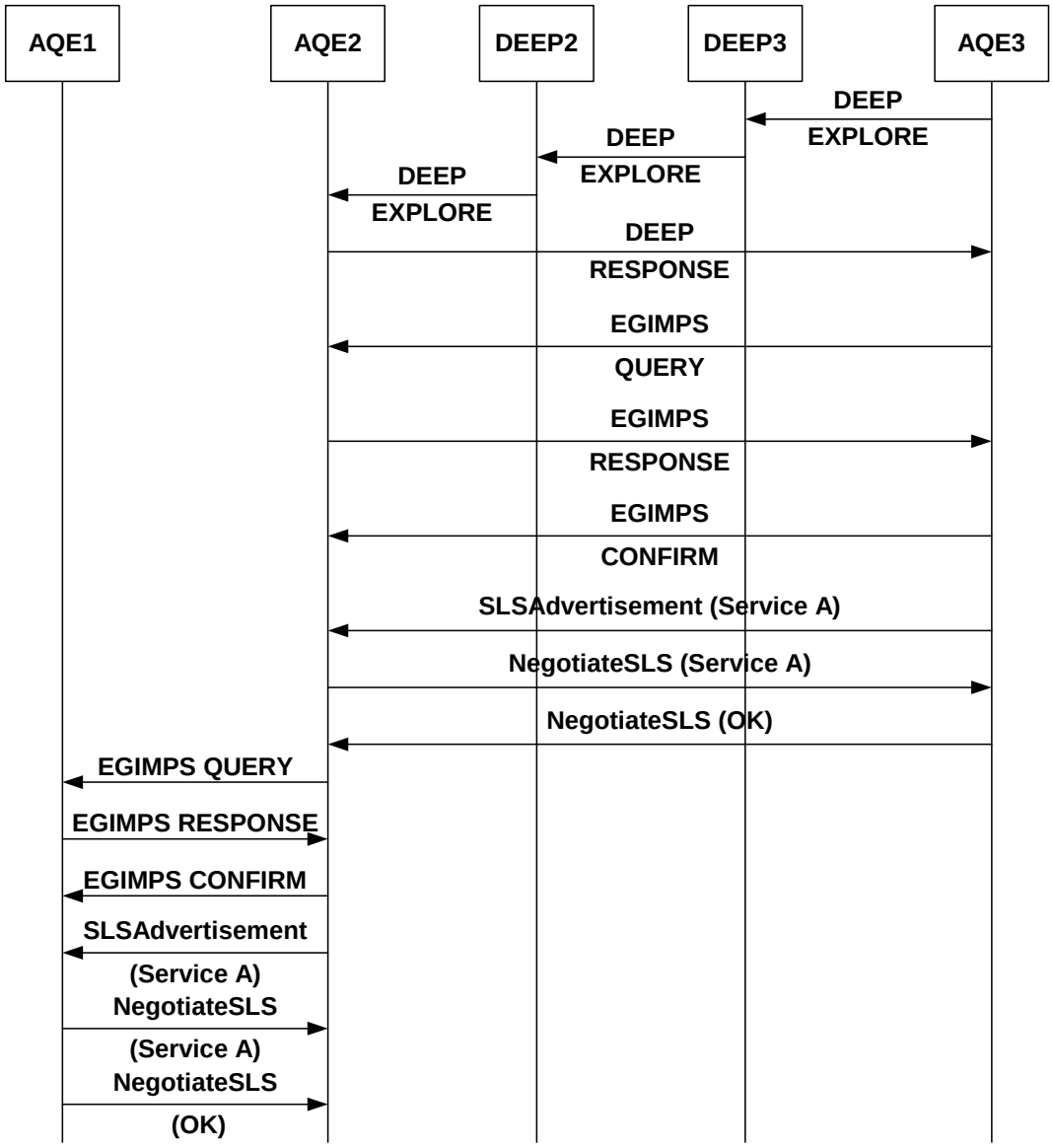


Figure 26: Use of EGIST for INQA-P signaling

The scenarios given here are simply examples of the tests carried out to verify EGIST and DEEP functionality. In reality, the prototype is being tested in a wider variety of scenarios.

6.1.3.5 Conclusions

The GTLP prototype is a work in progress. As described in the preceding sections, it is being continuously tested using the scenarios described above. The tests carried out so far have not revealed any bottlenecks or major issues as far as the protocol is concerned. However, more extensive set of tests is required, especially in more complex scenarios, when the prototype becomes stable. Furthermore, it is essential to study the performance of the prototype itself with respect to processing and memory.

6.2 INQA

6.2.1 Support for elastic and non-elastic traffic

6.2.1.1 Introduction

The widespread of the Internet coupled with the improvement on the Internet access rates, achieved with the appearing of ADSL, created new opportunities for the support of an increasing number of services. Some of these services are assumed to be provided to end users with guaranteed quality levels, such as VoIP for example. Thus, we assist today to an increasing demand for the support of traffic with QoS guarantees. However, in order to provide the needed guarantees to this more demanding traffic, network operators have to adopt mechanisms to provide those guarantees. Parallel to the issues rose above, is the increasing demand for solutions that are capable of transparently address the users mobility.

The Intra-Network QoS Agreements (INQA) model was developed in order to address the QoS demands of future communication scenarios, in accordance with the vision of this project. INQA enables the dynamic negotiation of QoS agreements; independently of the particular QoS model adopted by the involved ANs, and is capable of efficiently address the mobility of networks as well as the usage of QoS resources. Moreover, INQA avoids the usage of e2e QoS signalling, which is particularly important when crossing heterogeneous networks that do not understand the signalling information and will not provide the necessary QoS guarantees.

This section presents a conceptual analysis about the usage of INQA to support traffic with relative guarantees, called elastic traffic, and traffic with absolute guarantees, called non-elastic traffic. This analysis starts to describe some aspects of INQA as well as its capability to address the future requirements. Furthermore, some requirements to be able to provide the agreed QoS assurances at the specified SLSs active periods are provided.

6.2.1.2 INQA solution

The solution adopted in Ambient Networks (ANs) to support QoS guarantees is based on the dynamic advertisement, negotiation and establishment of bilateral QoS agreements, materialized in SLSs. Any AN, able to provide services, may be able to announce those services and negotiate the corresponding service requests. Thus, in order to sell its capability to support QoS guarantees an AN must implement a set of modules that we call INQA, not only to be able to advertise its QoS capabilities, but also to perform dynamic negotiations and admission of SLSs as well as to monitor its network state after the admission of SLSs. This solution does not replace current QoS models for QoS reservations but complements them by providing the network the capability to dynamically negotiate QoS agreements with other networks and trigger the QoS reservations within its network when necessary.

The fact that Provider ANs advertise (to other ANs) their QoS capabilities to reach particular ANs or sub networks and the fact that, after negotiating with Provider ANs, the Customer-Provider AN can also advertise the previously negotiated resources supported through its own network, enables the support of e2e guarantees. The result of this procedure would be compared to a chain of “QoS tunnels”. After the establishment of a QoS agreement, the involved ANs are responsible for its realization (i.e., the reservation of the negotiated resources within its network) when the SLSs enter in their active period. This SLS realization involves the allocation of the resources negotiated in a particular SLS and triggers, in the Provider AN, the policing of the traffic benefiting from the negotiated QoS guarantees.

We argue that the solution adopted in this project is able to satisfy the requirements previously presented, since it enables the dynamic and automatic negotiation of SLSs as well as their automatic establishment. These QoS negotiations are dynamic and automatic since INQA’s Negotiation module (INQA-N) can automatically negotiate QoS agreements when receiving a request. Moreover, the resources present in the SLS can be dynamically adjusted, through SLS renegotiations, and SLSs are automatically negotiated with other Provider ANs when the AN moves. ANs can also have pre-defined SLSs which are automatically negotiated when a

SLS advertised by a Provider AN suits the pre-defined SLS requirements. SLSs are technology independent and can be mapped in several different QoS models [45]. INQA automatically deletes the QoS agreements when they reach their expiration time.

The advertising solution adopted in this project facilitates the establishment of QoS agreements since Customer ANs have immediately available the information about QoS resources available towards a set of destinations ANs or sub-networks within ANs.

Due to the fact that QoS agreements are bilateral, the signalling information (i.e., SLS negotiation messages) is exchanged only between two networks. This characteristic also simplifies the Customers re-establishment of previous QoS agreements in case of mobility. Bilateral agreements can also avoid the need to use e2e signalling, which can be hardened by the distinct QoS models adopted by the networks in the e2e path. Although e2e signalling can be avoided, our solution can co-exist with it. For instance, QoS NSLP can be used. In this particular case an optional parameter could be introduced in QSPEC object to identify the SLS ID (this parameter must be actualized every time the NSIS message leaves one domain to enter another). Another issue that needs to be solved when using QoS NSLP e2e signalling in ANs is the control of the path followed by the QoS NSLP signalling messages. These messages should cross the corresponding set of ANs with SLS agreements established between them for a particular QoS as received by the Customer AN through the SLS advertisement, and independently if those ANs understand NSIS signalling or not.

The following table summarizes the requirements for future communication scenarios presented before and the solutions adopted by INQA to address them.

Requirements	INQA solutions
Dynamically and automatically negotiate QoS agreements - independent of specific QoS models or technology	- INQA-N, supported by the Policy information DB, mobility information and automatic detection of new Providers, is able to dynamically and automatically negotiate/establish SLSs
Efficiently address the mobility - fasten the e2e QoS establishment	- Advertisement of QoS resources materialized in SLS (independent of QoS models) provides immediately information about e2e QoS paths - Mobility of nodes/networks requires only bilateral negotiations
Efficient usage of resources	- SLSs are automatically deleted when expire - SLS's active periods enable a better resource utilization - Reserved resources are automatically released if connectivity is lost with the customer (due to mobility)
Avoid using e2e signalling – networks crossed in an e2e path can adopt different QoS models	- Negotiated SLSs are mapped into the corresponding AN's QoS model or technology at the specific SLS active period triggered by their own INQA-N module

Table 4: Requirements for future communication scenarios and INQA solutions

6.2.1.2.1 SLS active Periods

The adoption of active periods in SLSs enables the multiplexing of reservations over the same link at different time intervals (the existence of the active periods also enable the possibility of adopting different tariffs according to the time that resources are reserved – e.g., day or night).

One active period is characterized by a start time, an end time, periodicity and expiration time. The adoption of "active periods" in SLSs increases the complexity of the SLS Admission Control, that, additional to the link capacity and the amount of resources requested by a

particular SLS has to consider also the time intervals over which the resources are requested. However, this complexity can be largely compensated by the potential resource optimization that can be achieved.

6.2.1.3 Requirements to provide QoS assurances to elastic and non-elastic traffic

This section presents the main requirements for ANs to be able to transport elastic and non-elastic traffic, supported by the usage of the INQA to negotiate/establish QoS agreements.

6.2.1.3.1 The impact of the creation of SLSs to be advertised in an e2e scenario

One of the main problems that can arise in the support of e2e guarantees resides in the creation of QoS guarantees to be advertised to other ANs based on agreements established with other ANs. If, in an e2e scenario, an AN makes a mistake in evaluating or instantiating the QoS parameters of the SLS to be advertised, this error propagates in the ANs chain. It is then likely that the advertised QoS guarantees will not be achieved in an e2e scenario for non-elastic guarantees; if the QoS advertised under evaluates the real QoS capabilities (regarding elastic traffic, the possibility of guaranteeing the advertised QoS depends on the error value). It is important for the AN to have a "precise" knowledge about their network capabilities in order to avoid errors and be confident about the QoS parameters advertised.

In order to create an SLS for advertisement, based on an agreement previously established with another AN, the AN should consider the group of QoS capabilities (i.e., parameters and values) supported by itself ("G1") as well as the QoS capabilities present in the established SLS ("G2"). Thus, the QoS capabilities for the new SLS to be advertised are obtained as G=f (G1, G2), where "f" is a function used to obtain the QoS values for the based on the values of the negotiated SLS and the QoS supported in the network. The definition of "f" is out of the scope of this document. The definition of such a function could possibly be a local matter – that is, ANs may use different ways of defining such "f". Moreover, different functions may be used to compute each of the composed QoS parameters that are included in the SLS. Using this approach, the creation of SLSs to be advertised in an e2e scenario will be consistent and ANs can be confident about the QoS values present in the advertised SLSs they receive, independently of the number of ANs composing the e2e scenario. We argue that using this approach e2e guarantees can be supported. A different matter is whether the advertised SLS suits the QoS requirements of the receiving AN but this happens for any SLS. The "composed" or concatenated SLS will certainly degrade, for example the e2e delay which accumulates. The question is not about the final value announced (which certainly is higher that for each individual SLS / AN) but about the confidence/guarantee of the announced value.

6.2.1.3.2 End-to-end QoS information

INQA is Provider initiated, this is, SLSs are advertised and established from Providers towards Customers. This model can lead to a huge amount of SLSs advertisements between ANs, not only to the fact that each AN can advertise its own SLSs towards different destinations ANs but also due to the changing of QoS resources available, which requires new advertisements to replace the previous ones. This requires the adoption of a compression/aggregation mechanism in order to reduce either the number of SLSs being advertised or the information advertised.

Since INQA is based on bilateral agreements, it needs a mechanism to provide e2e information. This mechanism should be able to indicate to a particular Customer AN, the QoS that it can benefit towards another AN, based on the SLSs established between ANs composing the e2e path.

We believe that all of the issues raised above can be solved by a single mechanism. This mechanism should be able to:

- reduce the information in SLS advertisements to be transmitted in Wireless links without loosing information (or loosing the minimum information possible)

- provide “QoS routing”, i.e., informing about established SLSs and their available resources
- the e2e information should be provided in such a manner that any AN could, very quickly, evaluate the resources available between itself and another AN in the path

The development of such a mechanism is out of scope of this document and is postponed to phase 2.

6.2.1.3.3 Mechanisms that influence the support of e2e QoS

The model adopted in ANs for supporting e2e QoS (based on bilateral agreements) is more dynamic and flexible than traditional models but raises particular issues that have to be addressed in order to support the e2e QoS. One of those issues raised by this solution is related with the synchronization between the peer ANs. After the establishment of the QoS agreements, it is a provider’s responsibility to realize the bilateral SLSs in the corresponding active periods (defined by timestamps) so that the traffic sent by the customer benefits from the negotiated QoS guarantees. In a chain of ANs, all the involved ANs have to reserve the negotiated resources, in the corresponding active periods of the negotiated SLSs, so that the e2e guarantees are provided. However, if one of the ANs in the chain, for some reason, does not realize the SLS or delays itself in the SLS realization, the flow will not benefit from the QoS guarantees and the negotiated QoS will not be respected, either during a period of time or for the complete duration of the SLS active period.

The timestamps used in SLSs under negotiation should use absolute time references, so that each AN can translate that time reference to its current local time and avoid the problems inherent to different time zones. This is particularly important in moving networks, negotiation of roaming agreements, etc. Moreover, in order to transmit confidence between the negotiating ANs or to guarantee, at least, that both ANs have the capability to respect the agreement (i.e., the customer sends traffic at the negotiated active period and the provider will have the required resources reserved within those periods), is to be synchronized. The mechanism for ANs to synchronize is out of scope of this document. For instance, Network Time Protocol (NTP) [12] could be used for AN synchronization.

Another mechanism that can influence the support of e2e QoS is the Admission Control mechanism located at the network level. Since the negotiated SLSs can be used to support different flows: from different services and also from different sources towards different destinations (e.g., one SLS for a VPN with QoS guarantees between two offices), this mechanism has to be able to recognize any flows that must benefit from QoS guarantees. If this mechanism fails to recognize these flows, the support of e2e QoS will not be respected. This can be particularly problematic for non-elastic traffic, while for the elastic traffic it is possible, in some circumstances, to guarantee the e2e QoS.

This “per-flow” Admission Control mechanism located at the network level residing in edge routers, receives information from the SLS Admission Control mechanism located at the ACS of the AN. The part of the Admission Control mechanism located in the ACS is responsible for storing a precise picture of the network, namely information about paths between edge nodes of the AN, the QoS capabilities and the resources available. This information is updated based on information provided by INQA’s Monitoring module (INQA-M) that gathers this information from the network, and also based on the operations performed by INQA-N affecting the AN resources, such as SLS negotiation/deletion (for a more detailed understanding about these mechanism please consult [46]. INQA-N is also responsible for triggering the SLS realization when SLSs enter in their active periods. When receiving this trigger, the entity responsible for the SLS realization will reserve the required resources in the network nodes/ routers.

The edge routers are responsible for implementing a set of mechanisms such as the “per-flow” admission control mechanism, traffic policing, scheduling and, in some cases shaping and/or dropping. This admission control mechanism is responsible for the identification of the flows

that benefit from QoS guarantees (i.e., belong to SLSs established) to control the amount of QoS resources already being used and reject new flows when the negotiated resources are exceeded. The acceptance/rejection of traffic belonging to negotiated SLSs has also to consider the active periods of SLSs, since flows belonging to SLSs out of their active periods shall be rejected or be marked as best-effort traffic.

The traffic policing mechanism is responsible for ensuring that the traffic benefiting from QoS guarantees is conforming to the SLS established. Many mechanisms can be used to control and ensure this conformance, e.g., Tocken Bucket for peak rate traffic and Leaky Bucket for throughput. On the other hand, the Traffic shaping/dropping mechanism enforces the conformance of the traffic to the negotiated SLS. The adoption of each one of these policies: shaping or dropping, is a provider decision.

6.2.1.3.3.1 Traffic Identification

SLSs of Customer ANs established with Providers or Customer-Provider ANs are identified by these latter based on the SLS ID and AN ID. This is because different Customer ANs can generate and negotiate equal SLS IDs with Provider or Customer-Provider ANs.

Provider and Customer-Provider ANs have to identify the traffic sent by Customer ANs that should benefit from QoS guarantees. This can be achieved based on the SLS ID present in the negotiated SLS (the Source ID / Destination ID can also be used to guarantee that the SLS ID is not being improperly used by another node/network). These parameters have to be provided to the edge routers of the AN that are responsible for identifying and providing the corresponding QoS guarantees for the traffic entering the AN. Additional parameters can be used for a more tight control over the SLS traffic and statistics purposes (Source ID, Destination ID, Ingress ID, Egress ID, Service ID).

For simplicity reasons, the fields used to transport the SLS ID should be easy to access (e.g., avoid the extension headers of IPv6) and should also have equal number of bits in both versions of the IP protocol. One possible solution is to use the “Type of Service (8 bits)” field in IPv4 packets and the “Traffic Class (8 bits)” field in IPv6 version. However, this solution also poses some constraints in the SLS IDs. For instance, only 255 (excluding the zero) independent identifiers can be used simultaneously by an AN to identify SLSs. At the Provider or Customer-Provider ANs there is no problem, since the SLSs are uniquely identified based on the Customer AN ID and the SLS ID, enabling the Provider AN to support up to 255 independent SLSs per Customer AN.

At the Customer side, two situations can occur. The number of individual SLSs that an AN can have simultaneously established can be limited to 255 (excluding the zero), if using the 8 bits as a limit to the number of SLSs that can be uniquely identified. Alternatively, using the Provider/Customer-Provider AN ID coupled with the SLS ID to uniquely identify the SLSs, then each Customer or Customer-Provider AN can establish a total of 255 independent agreements for each Provider/Customer-Provider AN. Both solutions can be used and it is possible that both of them have their application field, for instance, the first solution can be suitable for small/restricted devices (e.g., sensors, etc.) while the second can be adopted by big networks (e.g., operator networks) or periodically moving ANs (e.g., train ANs that establish periodic SLSs with every station AN in the path).

6.2.1.4 Conclusion

This section has presented a conceptual analysis regarding the INQA capacity to support elastic and non elastic traffic. The INQA model adopts a different perspective regarding the support of e2e QoS of those currently existing. Nevertheless, INQA is able to provide e2e QoS guarantees for elastic and non-elastic traffic, according to a chain of negotiated bilateral SLS. In order to be able to provide the negotiated guarantees, INQA must consider a set of recommendations and/or mechanisms.

INQA is currently missing a concatenation/aggregation mechanism, able to provide e2e QoS information and reduce the information exchanged. The development of this mechanism is postponed to phase two.

Also, the use of an e2e probing mechanism to verify and even activate particular SLSs shall be considered in the future. Using NTLP for e2e signalling or other mechanism is for further analysis.

6.2.2 Scalability and Convergence analysis

6.2.2.1 Introduction

Currently there are a number of available solutions to perform the provisioning of IP networks, from static provisioning to over-provisioning, passing by signalled-provisioning. Independently of the mechanism used to control QoS inside each network, the major limitation for the deployment of end-to-end QoS in heterogeneous environments, as happened with Ambient Networks, is the lack of a dynamic approach to control QoS agreements between networks. To mitigate this limitation, the Ambient Network project proposes a solution, called INQA, which allows the dynamic control of Inter-Network QoS agreements. This solution is described section 2.

In this section, we provide a first insight about the scalability of INQA. The scalability analysis is based on the INQA prototype developed based on a topology with four Ambient Networks. This small topology, together with the small number of available services and possible requests, limit the range of our analysis. Nevertheless, being based on a prototype, this analysis provides us results that characterize with good accuracy the behaviour of INQA.

The current experimental set-up does not allow us to have a clear understanding of the scalability degree of INQA. For that, we plan to thoroughly analyse INQA scalability based on a simulation model that can more clearly reflect the current and possible future dimension of the inter-working between Ambient Networks.

Nevertheless, besides allowing us to test the state machine of INQA (which can provide us good indications to create a realistic simulation model), the current analysis allows us to understand the scalability differences between two different paradigms: per-flow (either a unique flow or a flow aggregate) and per-SLS signalling. The former, implemented by protocols such as RSVP and QoS-NSLP, allows an initiator end-host to signal a set of network elements in the path to a destination end-host, in order to set network resources for a flow of a group of flows. The latter, implemented by protocols such as INQA-GSLP (a component of INQA as mentioned in section 2), allows two adjacent networks to establish and maintain a bi-lateral agreement to differentiate traffic and assure QoS based on different SLSs. In the case of INQA, the end-to-end assurance of QoS is provided by a chain of QoS agreements, which are established and maintained based on INQA capability to advertised SLSs (not only local ones, but also SLSs offered by neighbour networks), to negotiate SLSs, automatically and based on the demand of local applications, and the capability to monitor and adjust established SLSs.

The paradigms mentioned in the previous paragraph have different functional models, which scale differently in terms of the signalling overhead, and amount of state to be stored. Moreover, in a mobile scenario, these two approaches have different convergence times and different packet losses due to the handover of Ambient Networks. Therefore, in this work we evaluated the scalability and convergence of INQA and QoS-NSLP-alike[4] protocols in a scenario with static networks and in a scenario with moving networks. In both scenarios we use a terminal Ambient Network (AN1), which is placed inside a train network (AN2), which is

[4] We say QoS-NSLP-alike protocols, since in our test-bed we did not implemented the complete specification of QoS-NSLP, but only the state machine needed to set, refresh and tear-down per-flow reservations.

moving (or not) between two access Ambient Networks (AN3 and AN4), as illustrated by Figure 27.

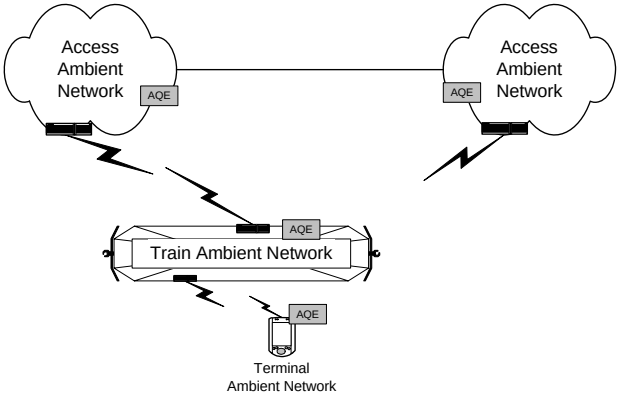


Figure 27: Prototype used in the Scalability Analysis

In a scenario in which all networks are stationary we compare the behaviour of INQA and QoS-NSLP-alike protocols in what concerns the state that they need to keep in network elements, and the overhead generated by the signalling needed to establish, maintain and remove such state. This comparison is made while increasing the number of connected Ambient Networks, the number of available services and the number of requested flows. To do this, the prototype shown in Figure 27 was configured as follows:

- Configuration A:
  - Fix number of available services: one in AN4
  - Increasing number of networks. During the experiments AN3 first attaches to AN4, then AN2 to AN3 and finally AN1 to AN2.
- Configuration B:
  - Fix number of available services: one in AN4
  - Fix number of networks, connected as shown in Figure 27.
  - Increasing number of flows, all being originated by AN1 and having as destination a subnet of AN4. These flows have all a specific duration, which in a real situation depends on the application. During the duration of the application, the state being used by the flow traffic has to be kept alive.
- Configuration C:
  - Fix number of networks, connected as shown in Figure 27.
  - Increasing number of available services. In this case each network advertised a service, providing access with QoS assurances to a specific local sub-network, or to a sub-network in another networks.
- Configuration D:
  - Stable scenario with unchanged number of networks, available services and requested flows.

In the analysis of INQA scalability in the static scenario, besides the measurements made in the prototype, we also did a brief analytical analysis based on different topologies. The goal of this analysis was to get more information about the degree of INQA scalability. The analytical analysis is described in the technical annex.

In a scenario in which AN2 is mobile (and consequently AN1), we compare the behaviour of INQA and QoS-NSLP-alike protocols[5] in what concerns the state that they need to keep in network elements, and the overhead generated by the signalling needed to establish, maintain and remove such state. Moreover, we also analyse what is the impact that INQA and QoS-NSLP-alike protocols have in the interruption of communications due to the handover of Ambient Networks, and on the amount of lost packets. To do this, the prototype shown in Figure 27 was configured as follows:

- Fix number of networks (four as shown in Figure 27)
- Fix number of services (one in AN3)
- Fix number of requested flows (one from AN1 to AN3)

The technical annex provides a thorough analysis of the comparison of the scalability and convergence times of INQA and QoS-NSLP-alike protocols. In the remainder of this section we provide a glimpse of our findings.

6.2.2.2 Scalability and Convergence Times

In what concerns a scenario with an increasing number of networks, as described in configuration A, INQA introduces 25% less signalling overhead than QoS-NSLP-alike protocols, as illustrated in Figure 28. This happens because, for every new Ambient Network, INQA needs three messages to set (advertise, negotiate and acknowledge) a QoS agreement that may be used by several flows. On the other hand, QoS-NSLP-alike protocols need a number of messages equivalent to [(Number of ANs – 1)*2] to set resources for each flow originated by any new Ambient Network. When there are only two networks, INQA needs one more message than QoS-NSLP-alike protocols to set its state. However, while in INQA the signalled state is referred to a SLS, which can be used by different flows, with QoS-NSLP-alike protocols, the state is only used by the traffic related to one flow. So, considering the trade-off between signalling overhead and data, INQA has a worse behaviour than QoS-NSLP-alike protocols if the set SLS is to be used only by one flow, which may only happen between a sender and an access network.

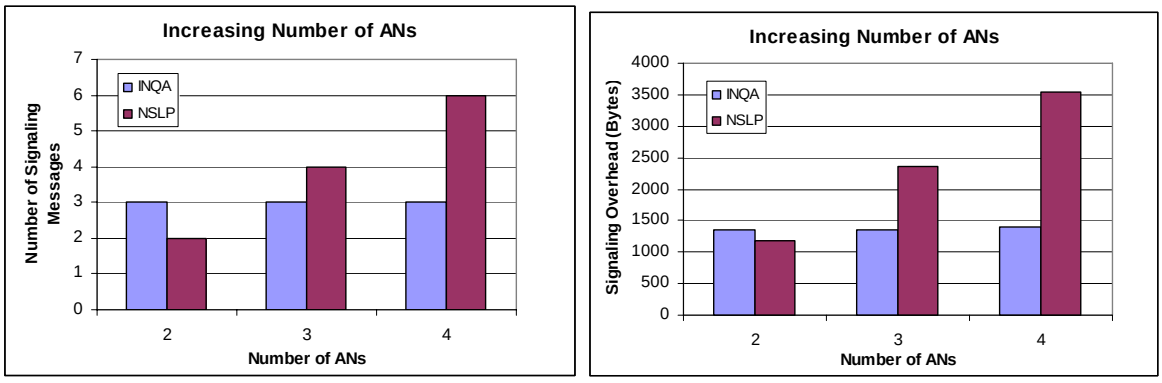


Figure 28: Signalling overhead with increasing number of Ambient Networks

Results also show that the difference between INQA and QoS-NSLP-alike protocols increases with the number of flows. This occurs because with QoS-NSLP-alike protocol, every new flow will need end-to-end signalling to set the required state, if available in the network. On the other hand, with INQA, in the best case no signalling is generated since the end-host has information enough to admit the new flows, which will then use the network resources of the SLS to which they are mapped. On the worse case, two messages will be generated between

[5] QoS-NSLP does not work with network mobility, since it is an end-to-end protocol. However, there is already an Internet-draft describing how QoS-NSLP can react to the mobility of hosts. However, there is no explanation in the Internet-draft of how can QoS-NSLP react to the movement of networks. Hence, in this work, we adjusted the behaviour of QoS-NSLP in order to react to information sent by access networks about the movement of the latter.

the end-host and its access-network in order to negotiate a per-flow SLS based on a previous SLS offer done by the access-network.

Moreover, an important difference between INQA and QoS-NSLP-alike protocols is that, once the SLS is negotiated and agreed between two or more ANs, INQA does not need to refresh the state in any data path. This because, operating between adjacent networks, INQA can react to changes such as mobility of networks and hosts, broken links, and not fulfilled SLSs. In contradiction to this, QoS-NSLP-alike protocols periodically refresh the state of each flow in the data path. Obviously, this process adds extra signalling overhead, which depends on the duration of each application. For a scenario in which QoS-NSLP-alike protocol refresh the state of flows every 30 s and applications last for 90 seconds, our measures show that INQA clearly outperforms QoS-NSLP-alike protocols in terms of signalling overhead, since it transmits 40% less signalling messages. In a real situation, this difference will be even higher since it is expected that multimedia applications will have an average duration much higher than 90 s.

All these results show that INQA has no major scalability problem with the increase in the number of networks and flows. However, the same cannot be said for an increasing number of available services. It is clear that if more than one AN offer services at the same time, or at different times, the SLS advertisements performed by INQA may result in a significant number of signalling messages. This scenario clearly indicates some scalability problems for INQA, but the extend of such problems require further investigation. An aggregation mechanism, implemented into the INQA protocol may improve its performance in case of increasing number of services.

In what concerns mobility, the hand-over time, this is the time needed for a moving-network to disconnect from the old access-network and connect to the new access-network, is always the same either for INQA or for QoS-NSLP-alike protocols. However, the convergence time, this is the time that a protocol needs to restore the network state, differs, since INQA only needs to update SLSs close to the moving network, while QoS-NSLP-alike protocols needs to update the state of the flow in the complete data path. In the prototype used in our experiments, INQA is used to update SLSs between three ANs (the moving network AN2, the new access-network AN4, a the user network AN1). This corresponds to the worse case, since this means that the SLS advertised by AN4 to AN2 is offering a QoS level lower to the SLS established between AN2 and AN1. If the QoS level offered by AN4 would be equal or higher to the SLS established between AN2 and AN1, AN2 could opt not to advertise the new SLS to AN1, reducing the convergence time, not to mention the signalling overhead. On the other hand, QoS-NSLP-alike protocols need to exchange messages between all four ANs. This is illustrated in Figure 29:

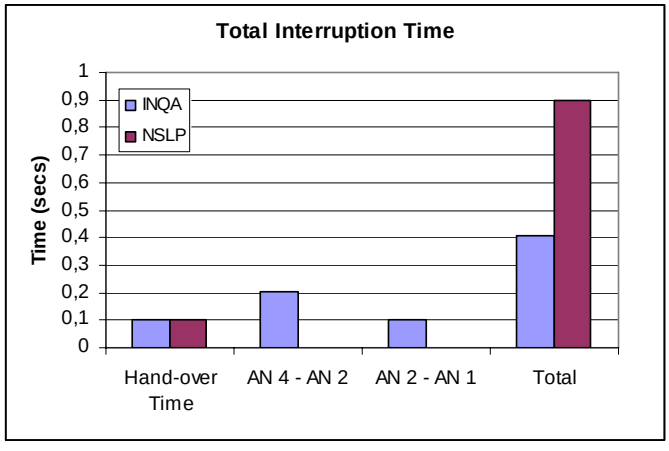


Figure 29: Interruption time induced by INQA and QoS-NSLP-alike protocols

As already mentioned, the handover time caused by the physical medium is the same for both protocols. After the connection of AN2 to AN4, AN4 advertises a new SLS to AN2, which automatically negotiates it. In this scenario AN2 re-advertises the available SLS to AN1, which immediately negotiates it, in order to make it available to the application traffic. Hence, in the case of INQA the overall interruption time, including the handover time, is of 40 ms. In case of QoS-NSLP-alike protocols, AN1 needs to reserve resources for the flow used by the application in the complete path to AN3.

6.2.2.3 Conclusion

This work provides an analysis of the scalability and convergence of the INQA prototype, while being compared to the performance of an implementation of the QoS-NSLP (only the mechanisms to establishment, refresh and teardown flow reservations were implemented).

Our goals are two-fold: First, we aim to analyse the signalling overhead, and the state required by both protocols when facing an increasing number of networks, flows and available services, in a scenario only with static networks. Second, we aim to analyse not only the above-mentioned criteria, but also the convergence time of both protocols in a scenario with moving networks.

Results show that INQA scales better than QoS-NSLP-alike protocols in both scenarios. However, it is clear that INQA may have scalability problems with an increasing number of available services, which may require the development of an aggregation mechanism. In the mobile scenario, results show that the INQA implementation converges faster than the QoS-NSLP-alike protocol after handover.

6.2.3 INQA Loop avoidance

The objective of this section is to clarify the need for loop detection and prevention in INQA, and present possible techniques and discuss their suitability for the task in this particular context. The discussion is backed up by references to similar approaches and simulation results. More extensive references, the description of the simulator and more detailed results form the simulation are to be found in the technical annex.

Even though the SLS negotiations take place between two ambient networks, the end result is end-to-end quality of service guarantee between the service provider and the service user. Thus, there exists a path between the service and the user. This path can form loops in which case the data passes through one and the same AN more than one time. Clearly this behaviour leads to unnecessary utilization of resources when the traffic is looped in the network. For customer this means e.g. longer delays, smaller bandwidth, and worse QoS in general. For the transit networks along the data path loops mean wasted resources and thus less served customers than possible.

6.2.3.1 Possible solutions

Loop detection, avoidance and prediction are used in several different applications, e.g. in routing protocols or mail delivery systems. Different systems pose diverse requirements for the loop avoidance techniques. This section presents some example solutions and discusses their feasibility in INQA.

We will use the Figure 30 as the reference to describe the current situation and the possible solutions.

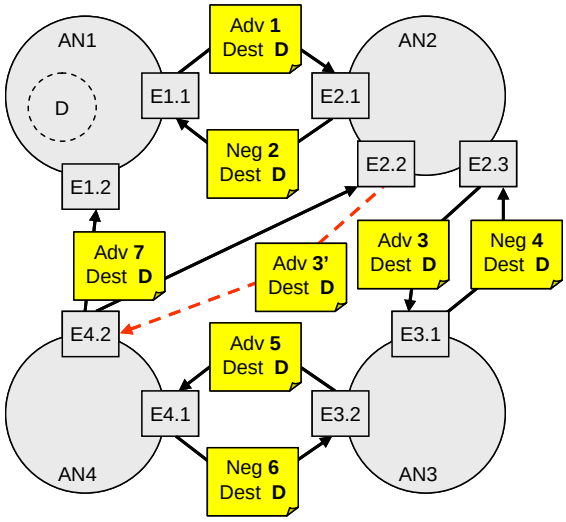


Figure 30: Example of four Ambient Newtorks where a subsequent advertisements and negotiations are forming a loop. ANs are numbered from 1 to 4. The edge points are marked as E*x.y* where *x* is the AN ID and *y* is AN-unique identifier. The advertisements (Adv) and negotiations (Neg) are numbered in temporal order. The arrows show the direction of data flow. D marks the destination (Dest). The path of Adv 3' is dashed to denote that AN4 does not receive this advertisement. (It should be noted that even though the advertisements and negotiations are drawn in the figure to go from one edge point to another, in reality it is the AQEs of several ANs which are communicating with each other.)

6.2.3.1.1 Current situation

In its current state, the INQA specification offers very little from the point of view of loop detection and avoidance. First, the specification prohibits *backward advertisement*. In Figure 30 this would mean that after AN2 has negotiated with AN1 and decided to advertise this further, it would not send the advertisement through edge point E2.1 even if there were other ANs connected via it other than AN1. This assures that the advertisement is not send back to the AN with whom a negotiation was just performed and whose resources are being advertised. Second, when AN receives an advertisement, it will *check the destination* information. If the destination lies within the AN itself, the advertisement is discarded. In Figure 30 AN1 would discard the Adv 7 because this advertisement has traffic destination D which belongs to AN1.

Simulation in a discrete event simulator shows that by using the current INQA specification, assuming a random network, and a worst-case scenario where advertisement is always followed by a negotiation followed by an advertisement based on this negotiation, the network will never stabilize and the advertisements–negotiation chains contain a large amount of loops (the length of the loop obeys in random graphs roughly an exponential distribution: short loops are very common, long ones rare).

6.2.3.1.2 Hop count

It is clear from the beginning that simple limitation on hop count (or time to live) does not prevent loops but only restricts the maximum length of a loop. E.g. when the first advertiser sets hop count to ten, there can be loops which are shorter than this. Naturally short loops are more probable to form, so the hop count should be kept in minimum. On the other hand, hop count should be large enough so that all the potential customers receive the advertisement. Finding a proper consensus value between these two extremities can be an impossible task. Even if it could be done, it would not prevent loops shorter than the loop count. In the simulation, the distribution of loop lengths is the same as in previous case except that probability for loop longer than hop count is zero.

6.2.3.1.3 Hash-based loop detection

Hash-based loop detection means a method where a hash function is applied to the whole message or part thereof, and the result is cached. Plain cache-based loop detection can be considered as a special case of hash-based solution, where identity function is used as the hash function. This approach is used e.g. in some e-mail delivery systems where it is especially suitable because e-mail loops are usually caused by automated replies and thus the message body remains very much the same. Hashing a message and comparing the result to cache is enough to catch most of the loops.

In INQA, the situation is very different. The advertisements are not forwarded per se but an advertisement is followed by a negotiation and the result of the negotiation is used as a basis for new advertisement. The contents of the advertisement can change a lot when it passes a single advertisement–negotiation cycle. Therefore applying hash function to INQA advertisements and caching the results does not provide any protection against loops.

The only piece of information which remains at large parts the same is the destination. However, AN should not discard an advertisement which has the same destination as one of the already existing negotiations because the new advertisement can have very different QoS characteristics and costs.

6.2.3.1.4 Inheritance-based loop detection

Inheritance-based loop detection means a scheme where a message always contains information on which other messages it is based or which entities have participated in forwarding or manipulating the message. E.g. Border Gateway Protocol (BGP) stipulates that whenever a route information is passed from one Autonomous System to another, the sending AS must add its ID to a list which forms part of the route information.

Adopting this method in INQA means that the advertisement must have the list of all the ANs through which it has traversed. In Figure 30 this would mean that Adv 1 has AN ID list {AN1}, Adv 3 and 3' have {AN1, AN2}, Adv 5 has {AN1, AN2, AN3}, and Adv 7 {AN1, AN2, AN3, AN4}. When AN2 receives Adv 7, it would notice its AN ID in the list and discard the advertisement.

In static situation, this method fully prevents loops from forming. Dynamic situations (network change and composition) are discussed in 6.2.3.3.

6.2.3.2 Dangers of loop prediction

Loop prediction means methods where network entities use some heuristics to determine if loop is possibly forming or not. Care should be taken that if loop prediction techniques are used, they do not lead into unwarranted denying of service for some ANs.

6.2.3.3 Possible pit falls of inheritance-based loop detection

This section discusses three major problem areas related to the inheritance-based loop detection, namely change in the network graph, splitting the network graph into two or more unconnected parts, and AN composition.

6.2.3.3.1 Change in the network graph

According to both measurements and simulations, BGP can have transient loops which are caused by change in network topology and delay in propagating the new routing information to all concerned parties. The main difference between BGP and INQA is that INQA does not refresh any information and everything in INQA is bilateral, between two ANs.

If now connection breaks between AN1 and AN2, and AN2 successfully negotiates with AN3, AN2 does not inform anyone about this change. Now a loop can arise if the advertisement–negotiation chain propagates from AN2 via other ANs to AN3. AN3 does not find its ID in the ID list, so it is incapable of detecting the loop.

There are two solutions to this problem. The simple solution is that we allow this single loop to form because further loops are all the ANs in the ID list. This possibly leads to poorer performance of AN3, but this remains an open issue to be studied.

Second solution is to tell AN3 in the negotiation phase that the negotiation is actually a renegotiation which is supposed to replace a broken agreement. This information would include all the path information which is known at the time. After this, AN3 has to check two things when it receives an advertisement. Like normally, it checks whether it finds its own ID in the list or not, and acts accordingly. But it has to check if destination is found in the renegotiation cache and that the path prefix matches the information given in renegotiation phase. This means much more processing for each advertisement received and, if aggregation is used, recognizing the destination might be difficult. Thus, there is both a risk of not noticing a loop and a risk of false positive loop detection.

Which one of these two solutions to use, requires more detailed simulation of the INQA, and remains currently an open issue.

6.2.3.3.2 Disconnected network graph

Disconnection is a special case of change in network topology. It means a situation when a single link is the only one connecting two regions of the network and it fails causing the two regions to become unconnected with each other. For the proposed loop detection, disconnection of the network into two or more parts does not pose a problem because each AN is responsible for detecting loops it is causing itself.

6.2.3.3.3 Composition

The proposed scheme relies on permanence of Ambient Network identifiers: AQE must be able to recognize its AN ID from the list in each received advertisement. Composition can cause the situation that when an advertisement returns to an AQE through which the advertisement has already traversed, the AN ID of that AQE has changed.

The most profound mode of composition is absorption where two or more ANs compose and lose their original identities and a new management domain with a new AN id is created. Closely related is gatewaying where the ANs retain their identities but a new overlay with new AN ID is created. From the point of view of the loop detection, the “new” AQE must know the identities of the predecessor ANs. In decomposing, the knowledge on the ID of the decomposed AN must be passes to successor ANs. Here an open question is, how long the new ANs should keep the ID of the old AN in cache.

In interworking composition, two or more ANs can select one FE to take care of a particular functionality on behalf of all the other ANs. All the ANs retain their identities, and no new AN ID is created. Here the AQE acting on behalf of other AQEs in required to mark the AN IDs to advertisement in a truthful manner, i.e. even if it is AN1’s AQE acting on behalf of AN2, if the advertisement contains resources from AN2, the ID of AN2 must be placed into the list. In decomposition, there is no need to cache any information (for loop detection) because the advertisements which were sent out in composed state faithfully reflect the true network structure.

6.2.3.4 Conclusions

The previous analysis (which is presented in more details and with more examples in the technical annex) justifies that a loop detection and avoiding mechanism should be added to INQA. The proposed scheme is as follows:

1. The originator AQE of an advertisement marks its AN ID to the advertisement.
2. Every time an advertisement is based on a negotiation, AQE must append its AN’s ID to the list of AN IDs which was present in the advertisement which lead to the negotiation.

3. When AQE receives an advertisement, it must check the AN ID list. If it finds its own AN's ID in the list, it must reject the advertisement.

Results show that the INQA inheritance-based proposed solution is able to detect all possible loops with low bandwidth overhead and low computational overhead. Nevertheless, further evaluation work needs to the done to understand the impact of the length of AN IDs in INQA messages, together with the open issues described in the next section.

6.2.3.5 Open issues

6.2.3.5.1 Privacy

Containing the full path from the service provider to the customer reveals a significant amount of information on how ANs are connected to each other. In current Internet, the BGP does the same, but this information remains within the network operators and is not relayed to end users.

Stripping the path information from the advertisement before sending it to the end user is one way to limit the information to those who need it and not revealing it to everyone. However, this clearly breaks the whole loop detection scheme. As a result, the service providers could accept advertisements from their end user customers only when the advertisement contains resources from the end user's AN (i.e. the end user is the destination of flow indicated in that advertisement). The feasibility of this approach remains an open question.

6.2.3.5.2 Renegotiation

Change in the path due to link failure and renegotiation was discussed in 6.2.3.3.1. The selection between the two solutions proposes, remains an issue of further research.

6.2.3.5.3 Composition

The requirements posed for the composition process are identified in 6.2.3.3.3. These should be taken into consideration. Further study is needed to determine the proper time to keep the ID of decomposed AN in the cache of the successor ANs.

6.2.4 Robustness

6.2.4.1 Introduction

The INQA robustness evaluation aims to analyse the performance of INQA when the ANs are facing network changes. The network changes include mainly three aspects, e.g. routing change, fading channel change and AN provisioning change.

All the three kinds of changes may cause failure of connection or falling of available bandwidth, in which case the INQA protocol can completely fail and has to restart. The goal of this document is to evaluate the robustness of the INQA protocol in the two situations, e.g. failure and decline, respectively.

The evaluation is done with a self-designed C simulator. It simulates several ANs connecting with each other based on predefined topologies.

6.2.4.2 Scenarios of Network change

There are three network change cases: routing change, fading channel change and AN provisioning change. To evaluate the robustness of INQA, the three kinds of changes should be simulated in the simulator so that the performance of INQA can be evaluated. Routing change is caused by the handover of AN. From the routing change, the connection situation among different ANs should be considered. Therefore, *the routing change can be abstracted to the connection status change*. Fading channel change is caused by the varying environment of the moving AN. It will change the bandwidth of the wireless connection and further influence the parameters defined in the SLSs. So *the fading channel change can be abstracted to the change of bandwidth*. AN provisioning change is caused by the policy of

provider AN. The direct result is that the available resources offered to the customer AN may be decreased. Therefore, *AN provisioning change can also be abstracted to bandwidth change*.

6.2.4.3 Parameters Definition

To analyse robustness of INQA quantitatively, some parameters have to be defined and measured during the evaluation. A **reacting time** can indicate how long INQA can remain working since the change until the failure; while a **recovery time** tells us how long it takes for INQA to work in a stable state again. **Failure probability** indicates the probability that INQA fails to returns the stable state in the change of network circumstance.

Reacting time --- It is the time interval from the instant of network change to the instant that INQA reacts the change.

Recovery time --- It is the time interval from the instant INQA is restarted to the instant INQA works in a stable state again.

Failure probability --- It is the probability that INQA fails when the network is changing continuously.

$$P_f = \frac{1}{N_c}$$

In the equation, P_f is the failure probability, and N_c is the number of network changes before a failure occurs.

6.2.4.4 Simulation Scenarios

The simulation is done on a C simulator, which is organized as a network matrix. The matrix element number stands for ANs at the ends of the connection. The simulator is originally used for advertisement loop avoidance. Simulation for robustness evaluation starts from the time step that the system enters a stable state.

6.2.4.4.1 Connection failure

To evaluate the robustness of INQA to the connection failure, a network with *N* ANs is set up. Among the *N* ANs, one AN acts as a provider and advertises a SLS. Its neighbour ANs negotiate SLSs with the provider AN and resell the advertisements to their customers. When the process ends and the system become stable, one link between two ANs will be down. According to INQA mechanism, the AN is assumed to detect the failure of the link and the SLS on the failed link should be deleted. If neighbour ANs are reselling the failed SLS, many SLSs can be affected by the link failure and the SLSs should be renegotiated. In this scenario, the simulation will check if the system can be reach a stable state again and the number of messages exchanged among the ANs after the connection failure is detected.

6.2.4.5 Conclusion

From the simulation, INQA can detect connection failure immediately, and return in stable state in short time. In as large as 100-AN domain, no system collapse happens. Through the simulation results, we can conclude that INQA is robust facing the connection failure.

6.2.4.6 Open Issues

6.2.4.6.1 Bandwidth falling

A scenario should be implemented to simulate the bandwidth falling in the network and observe the reaction of INQA. In the network, an original bandwidth of 1 is initiated to each link between two ANs. SLSs are negotiated based on this scenario. When all SLSs are successfully negotiated and stable, the bandwidth of one of links falls to 0.1. The reaction of the ANs will be observed; metrics e.g. reacting time and recovery time will be measured.

6.2.4.6.2 Reacting time

Some mechanism should be designed to simulate the reacting time of INQA to network changes. This needs an implementation of INQA protocol and complete monitoring mechanism.

6.2.5 Analysis on INQA standardisation

6.2.5.1 Introduction

This section analyses the most suitable ways to promote INQA as a universal solution for dynamic control of inter-network QoS agreements. For this reason, a conceptual analysis of the main architectures being currently defined to offer end-to-end services with QoS guarantees has been made. This includes the IP Multimedia Subsystem (IMS) architecture being standardised by the 3rd Generation Partnership Project (3GPP), the Next Generation Networks (NGN) architecture being defined by the Telecoms and Internet converged Services and Protocols for Advanced Networks (TISPAN) group of the European Telecommunications Standards Institute (ETSI) and the proposals from the Focus Group in Next Generation Networks (FGNGN) group of the Telecommunication area of the International Telecommunication Union (ITU-T). The analysis compares the referred architectures with INQA in order to conclude whether functionality similar to INQA is still needed by these architectures or not and take the most appropriate actions.

Anyway, it has to be noticed that all these standardisation bodies use protocols already published as standards by the IETF and specify their usage in more detail taking into account their global architectures. Therefore, an analysis of how INQA could start being standardised within IETF and how to schedule such a standardisation effort has also been made.

6.2.5.2 NGN reference architectures

In the past few years, it is well recognized that there has been a shift from the voice-centric to data-centric applications, which is inevitably driving the telecommunication players to the adoption of packet based networks, by replacing the old circuit switches. This leads to a scenario based on a unified IP-based transport plane, the All-IP scenario, which is becoming the support for the Next Generation Networks (NGN). In this context, the service layer will be perfectly independent of the underlying transport technologies, which are all envisaged to be interconnected by means of the IP layer overlay.

In order to achieve this global scenario, 3GPP has developed the IMS solution for mobile networks, which allows the achievement of several key benefits, mainly the sharing of a common infrastructure for different services which will allow the nomadism of users among different access network technologies.

However, as mentioned, IMS has been designed for mobile networks, and there was the need to perform a parallel and, sometimes, complementary work in what the fixed network is concerned. This is the reason why ETSI, and also ITU-T, have started later on new similar projects. TISPAN, in the case of ETSI, and FGNGN, in the case of ITU-T, target the elaboration of standards that either partially or entirely endorse those that have been produced by 3GPP. In fact, there are plans from all these standardisation groups to align their architectures in further releases in order to have a unified reference architecture to cover the Fixed-Mobile Convergence (FMC) scenario. Therefore, we also refer to them in this document in a generic way as the 3GPP/TISPAN architecture.

For further information about these architectures, in the report “Analysis on INQA standardisation” included in the D3.3 Technical Annex, a description of the 3GPP IMS architecture as well as of the ETSI TISPAN and the ITU-T FGNGN architectures is given based on the existing available documentation.

6.2.5.3 Comparison of INQA with other inter-network QoS signalling proposals

Looking at the proposals in 3GPP IMS, ETSI TISPAN and ITU-T FGNGN, the 3GPP TR23.802 is the most formal document produced by these organisms that deals with inter-

network QoS signalling. In the case of ETSI TISPAN, it is working in an intra-network solution for signalling between the different QoS signalling nodes in a network (currently being called RCIP, Resource Connection Initiation Protocol), but not in an inter-network QoS signalling solution to deal with inter-network traffic bearers, as INQA is.

Regarding 3GPP TR23.802, the presented approaches do not explain how the signalling is really performed, so no protocol can be identified as suitable for that inter-network interface yet. Anyway, it is clear that this signalling must be based on flows for the three-presented approaches. This means that, based on these approaches, networks allocate resources reacting to the QoS requirements of a specific flow. This has scalability problems and requires the use of an aggregation mechanism associated to the signalling protocol. We argue that a first step to improve the scalability is to use traffic-based signalling instead of flow-based, as INQA does. Trying to follow a similar terminology than the analysed architectures, we could also call it bearer-based signalling, where bearer has a different meaning depending on the considered network (mobile access, fixed access, core…).

In addition, the integration of advertisement, negotiation and monitoring mechanisms allows INQA to support local reparation of unsuitable QoS agreements. For instance, when an SLS cannot be fulfilled anymore in the network of a provider, INQA-M notifies the customers and INQA-A advertises a new SLS based on the current network capabilities. After receiving the notification, the INQA-N module of the customer tries to negotiate the notified SLS with another provider. In case of success, a new SLS mapping is requested via the ARI allowing flows, which are identified by their SLS, to be automatically redirected.

In a mobile scenario, traffic-based signalling, as that supported by INQA, may more easily lead to small communication interruption times, is due to its bilateral nature in opposition to the end-to-end nature of flow-based signalling. This because traffic-based signalling reacts to the movement of networks, updating only the necessary QoS agreements near the moving network, and not the allocation of resources in all intermediary networks in the old and new data path.

Therefore, INQA proposal fulfils all the requirements for the dynamic control of QoS agreements, and presents conceptual advantages in relation to the other analysed proposals. Therefore, we could suggest the adoption of INQA as a proposal for the AIPN scenario that 3GPP, ETSI and ITU-T have in mind. In order to make this integration as smooth as possible, we suggest doing it by evolving the Path-decoupled BCF QoS signalling approach already accepted by the 3GPP. This suggestion is justified by the fact that the BCF approach presents network elements (PDF/BCF) and interfaces (Gq, Gu and Go) which can be evolutioned to implement the INQA mechanisms and protocol. In this sense:

- The PDF/BCF elements would implement the INQA functionality, being equivalent to the AN QoS Entities (AQEs).
- The Gq interface would be equivalent to the interface provided by INQA to upper layers, used by the Media Routing functions (WP5) or directly by applications (ASI) in the AN architecture.
- The Go interface would be equivalent to the interface used by INQA to communicate with lower layers, that is, the ARI of the AN architecture to communicate with legacy networks.
- The Gu interface would be equivalent to the ANI interface in the AN architecture, used to communicate AQEs of different ANs, that is, the inter-network interface

Finally, it has to be noted that in the Ambient Networks architecture being defined in the project, INQA uses EGIST as the signalling transport layer of GANS in order to use the ANI, but this is not a mandatory requirement in the case of INQA. In fact, INQA supports any transport protocol that can carry INQA messages among INQA entities, that is, AQEs. This

means that alternative transport protocols, such as, SOAP (following Web services paradigm) or simply TCP could be used as well, at least considering only INQA requirements.

6.2.5.4 Alternatives for INQA standardisation

6.2.5.4.1 Introduction

Regarding the analysis of alternatives for the standardisation of INQA, it must be noticed that the three commented groups, 3GPP, ETSI and ITU-T make use of protocols previously standardised within the IETF, which are integrated in their architectures by defining a specific usage of them. Therefore, the way to include INQA proposals in the current NGN architectures should be by means of its standardisation in the IETF.

After an analysis of the current work being done at the IETF and the known interests from different companies willing to contribute, there are two main options in the short- and mid-term:

- To contribute to the existing IETF Working Group (WG) where INQA best fits, the Next Steps In Signalling (NSIS) WG.
- To promote a new WG covering end-to-end QoS architecture issues, where INQA could fit.

6.2.5.4.2 Contribution to the IETF NSIS WG

As commented before, INQA may be implemented based on a protocol as simple as TCP, since in INQA only the initiator and destination of INQA messages have to be signalling-aware nodes, and there is no need to signal other entities in the path among them. But since within the IETF framework there is the NSIS WG, which is responsible for standardizing an IP signalling protocol with QoS signalling as the first use case, INQA may be implemented as part of the NSIS protocol suite. The NSIS WG concentrates on a two-layer signalling paradigm (transport sub-layer and service sub-layer). In this context, INQA may be implemented as a new protocol for the service sub-layer in the case that the protocol for the transport sub-layer can provide path-decoupled signalling.

Despite the common interests of INQA and NSIS, it must be noticed that nowadays NSIS is not focused on SLA-based QoS signalling. In addition, all the current NSIS specifications are path-coupled oriented and previous history, such as a PAth Decoupled Signalling (PADS) BoF, which took place in March 03, shows that path-decoupled solutions, as INQA is, have failed in being accepted within this WG.

The main weakness exposed to path-decoupled signalling has been the need of an external mechanism for discovering the next signalling node, that is, some topology awareness mechanism. Also problems with packet forwarding in case of load balancing and policy routing mechanisms were raised.

However, the 3GPP IMS and TISPAN NGN architectures have solutions for the topology awareness by other functional blocks instead of the QoS signalling. Therefore, the path-decoupled signalling makes sense for these architectures and there is an important requirement of this kind of signalling from this standardisation bodies to the IETF. In this way, a new ‘problem statement’ draft has been recently released (first version in May 05, second in July 05, third in October 05), analysing the modifications required in current NSIS framework to support path-decoupled solutions.

We can conclude that, despite being something really missing in the current Internet architecture, previous history shows that it is hard to succeed in the IETF in this area. Therefore, it would be needed the co-operation with several vendors jointly with a more thoroughly knowledge of the views of the people currently involved in the NSIS WG. In any case, additional companies, especially vendors, should support a potential standardisation work about INQA to have a good probability of success.

6.2.5.4.3 New IETF WG on end-to-end QoS architecture

Alternatively to the contribution to the NSIS WG, INQA could promote a BoF in a future IETF meeting in order to create a new WG focused on end-to-end QoS architecture issues, where INQA would fit.

It is well known that many people is working in different forums and projects on end-to-end QoS, since network operators need it in order to offer end-to-end services with QoS guarantees, but it is an issue not completely addressed in standardization, currently focused mainly on intra-domain (access and core) QoS signalling:

- ETSI TISPAN studies only access QoS for Release 1.
- ITU-T FGNGN has just started to study inter-domain QoS.
- IETF focuses only on a path-coupled solution within NSIS, while RACS/RACF/PDF have not their corresponding WG at the IETF.

In this context, the main worldwide network operators, jointly with their main equipment providers, are exchanging points of view in different forums and projects in order to achieve a standardised end-to-end QoS solution. The possible ways could be:

- The arrangement of BoF meetings at the IETF in order to establish a new WG which would provide inter-network protocols that could be used by the 3GPP, ETSI TISPAN and ITU-T FGNGN groups for their architectures.
- The creation of standardisation groups alternative to the IETF, which would specify inter-network standards to provide end-to-end QoS. Examples of these forums are Fireworks (nowadays inactive) and IPSphere Forum (previously called Infranet). Later, specifications from these groups could be moved to the IETF with a big support from operators and vendors.

The drawback for INQA in this case is that the standardisation process could be too long. On the other hand, INQA could fit quite well in the whole architecture that these fora have in mind.

Within this alternative, a BoF called DCPEL (DiffServ Control Plane Elements) took place in the 64th IETF meeting at Vancouver (November 2005), with similar interest that INQA but only focused in an intra-domain scope. It is thought that after agreements on intra-domain solutions for QoS signalling at the IETF, inter-domain solutions will be addressed.

6.2.5.4.4 Additional considerations about moving INQA to standardisation bodies

Regarding INQA standardisation, it has to be noticed that the mid-term aim for 3GPP, ETSI and ITU-T is to extend their specifications to provide nomadic services to the end-users, that is, fixed-mobile convergence solutions supported by an unified architecture independently of the access network technology. This can be achieved by means of a unified control plane (as Ambient Networks proposes) based on an All-IP transport plane. However, Ambient Networks goes a step further. Regarding INQA the following features are added:

- Flexibility in the services offered by networks with the usage of the SLS information model. Services can be negotiated based on policies predefined by network administrators and no standardisation on services is required (e.g. QoS classes for 3GPP). In addition, the use of active periods in the SLSs allows a more efficient network usage.
- Ability to negotiate on-line SLSs without previous (off-line) contracts with the negotiation mechanism and monitor them with the monitoring module.
- Support of scenarios with moving networks and short-lived SLSs due to the SLS advertisement and negotiation mechanisms.

In this context, it is considered that a phased approach is the best migration path to include INQA solutions in these architectures, progressively providing the functionality required by the

reference architectures from FMC scenarios to AN scenarios while maintaining the lowest degree of complexity required for the QoS signalling. In addition, dynamic agreements among networks of different owners also impose important changes in the regulatory framework which must be solved jointly to the technological changes that AN project proposes. In this way, we will ease the introduction of Ambient Networks concepts in further standardisation steps, when more dynamic scenarios are considered in these bodies.

Concretely, this means that the advertisement functionality should be added in a second phase in standards. Even in this situation, INQA must be considered as a better solution for these reference architectures than the path-coupled signalling proposed by the IETF NSIS WG, due to its traffic-based approach, local repair capabilities and support for mobile networks, based on bi-lateral agreements, following the business relationships taking place in parallel among network owners.

6.2.5.5 Conclusions

After the analysis of the main architectures being defined to offer end-to-end services with QoS guarantees (i.e. 3GPP IMS, ETSI TISPAN and ITU-T FGNGN), it has been concluded that INQA fulfils all the requirements from these architectures, and presents conceptual advantages in relation to the other equivalent proposals, mainly due to its traffic-based approach, the local repair capabilities and the support for mobile networks, all this based on bi-lateral agreements.

Anyway, all these standardisation bodies use protocols already published as standards by the IETF. Therefore, an analysis of how INQA may be standardised within IETF and how to schedule such a standardisation effort has also been made. The conclusion is that there are two main options in the short- and mid- term: to contribute to the existing NSIS WG or to promote a new WG covering end-to-end QoS architecture issues, where INQA could fit as well.

In addition, it has to be noticed that the mid-term aim for these architectures is to extend their specifications to provide nomadic services to the end-users (fixed-mobile convergence), while Ambient Networks goes a step further with more challenging scenarios. Regarding INQA, it adds the flexibility in the services offered by networks with the usage of the SLS information model, the ability to negotiate on-line SLSs without previous (off-line) contracts, and the support of moving networks and short-lived SLSs with the SLS advertisement and negotiation mechanisms. In this context, it is considered that a phased approach is the best migration path to include INQA solutions in these architectures. In this way, we will ease the introduction of Ambient Networks concepts in further standardisation steps, when more dynamic scenarios are considered in these bodies.

6.3 Network Assisted Congestion Control

6.3.1 Checkpoint Nodes

In this section we shall go into more details on the checkpoint node design and evaluation. We shall especially be focussing on the active congestion control measures the checkpoints can undertake in order to protect the network they enclose from congestion collapse from unresponsive traffic. More details are available in earlier deliverables, and in the paper [27], on which this section is in part based on.

6.3.1.1 Architecture and Design

The network architecture of the basic checkpoint deployment is edge router centric, and can be seen in Figure 7. The egress and ingress checkpoints of a flow exchange bandwidth information on the rate flow is entering and leaving the network, and the difference of these rates is used as an indication of congestion in the network core. The rate difference is also used to carry out additional congestion control measures, such as admission control for the flows.

The node architecture for checkpoint nodes was presented in [25], and is only briefly recalled here. The architecture consists of mainly hardware-targeted functional blocks, divided between the ACS and the ACY according to Figure 31. The per-packet congestion control decisions are carried out entirely in the CC-FE-CY, while the functional blocks residing in the ACS are only used to set and report parameters and variables of longer time of validity. For example, per-flow and queuing policies are understood as storing and setting model-specific constants that are rarely changed.

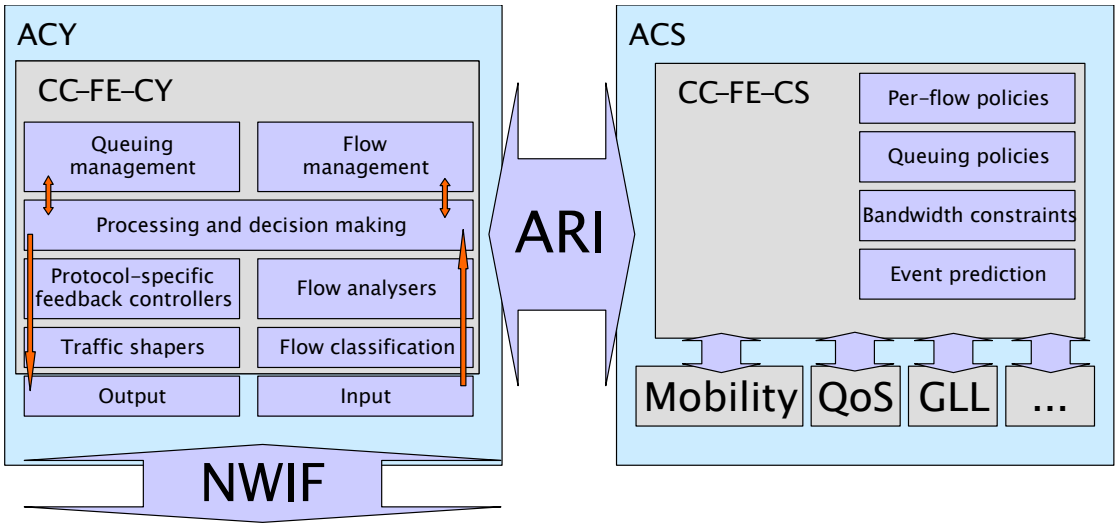


Figure 31: Checkpoint Architecture

The implementation approach to most of these functional blocks using state-of-the-art reconfigurable hardware is relatively straightforward. Main limitations come from the complexity of the processing block algorithms, and flow classification. Fortunately suitable classification techniques have been developed for firewall and QoS-enforcement purposes, and solutions up to line speeds in the core networks are available. For a recent contribution in this field, see [18] and references therein.

The original NBP framework maps to this general architecture in natural manner. Flow classification simply recognises the IP-address/port -pair, while flow analysers compute the ingress and egress rates. Processing block then derives the necessary feedback data, which is sent by the feedback controller block. Finally, traffic shapers are used as in the original NBP proposal. The flexibility of the architecture can, however, be also used to further improve the NBP proposal, as we shall see in the following.

As an initial approach the Network Border Patrol (NBP for short) is an excellent starting point, but for practical deployment it suffers from numerous shortcomings. Overcoming these has been a major work item in relation to the checkpoint approach. Flow aggregation mechanisms have been developed to reduce the signalling overhead under normal operating conditions. In the original NBP approach the flow information exchanged between the edge routers can consume significant proportion of the available bandwidth. In the present checkpoint node design this effect is remedied by employing dynamic flow management, the coarseness of aggregation depending on the number of flows and the available bandwidth present in the network.

In addition to flow management techniques, additional network protection mechanisms have been incorporated to the checkpoint node framework. Most important of these is the inclusion of connection admission control to prevent new flows from entering the network core in case of imminent congestion collapse, or during the recovery phase immediately following a collapse. Earlier studies pioneered by Roberts and Massoulié [34] indicate that these

measures can speed up the network recovery, and thus increase the quality of end user experience. To implement this kind of CAC procedure, we need estimate of the bandwidth flow would obtain if admitted to the network. Normally this kind of information is hard to obtain, but in our case it can be directly estimated from the backward feedback packets sent by the would-be egress router of the flow. Thus it suffices to set a threshold which will be compared to this estimate upon arrival of a new flow. As our simulations show, this simple approach works well in the scenarios considered.

Feedback towards end systems has been tackled on architectural level. Namely, provisions have been made to ensure that for existing transport protocols (TCP/ECN, XCP, and so on) the corresponding router feedback mechanisms can be dynamically incorporated into the checkpoint nodes.

Checkpoint nodes can also be used to support and be integrated with other functional entities developed within work package 3. As they provide functionality for flow-related measurements, they can also assist in monitoring functions related to (signalled) QoS support, and benefit from notifications from the QoS functional entity. Checkpoints can also utilize application preference information, if available, to enhance the CAC procedures. The admission decisions can, for example, be made based on application utilities as discussed in section 6.3.2.

6.3.1.2 Evaluation strategy

The basic evaluation methodology we have chosen in the standard simulation-based one. The checkpoint functionalities to be evaluated have been implemented in to the ns-2 network simulator, which is the standard academic simulation tool in the networking research field. The original basis of the implementation is the NBP package developed by UCI, which we have subsequently heavily modified. The original NBP implementation was developed for thoroughly outdated ns-2 version (2.1b4), as were the components it depended on. All of these were updated for ns-2 v2.27, and new features described above and in D3.2 were implemented into the new version.

In the evaluation process the operation of the checkpoint algorithms have been studied in several network topologies, and in widely varying traffic configurations. Examples of these are standard test topologies, such as the extended dumbbell topology and GFC-2 topology shown in Figure 32 together with example traffic configurations. Larger synthetic topologies are also being used to validate the approach in more complex environments. Several queuing disciplines were experimented with for the core routers to with or without the checkpoint functionality present, to provide a baseline for comparisons.

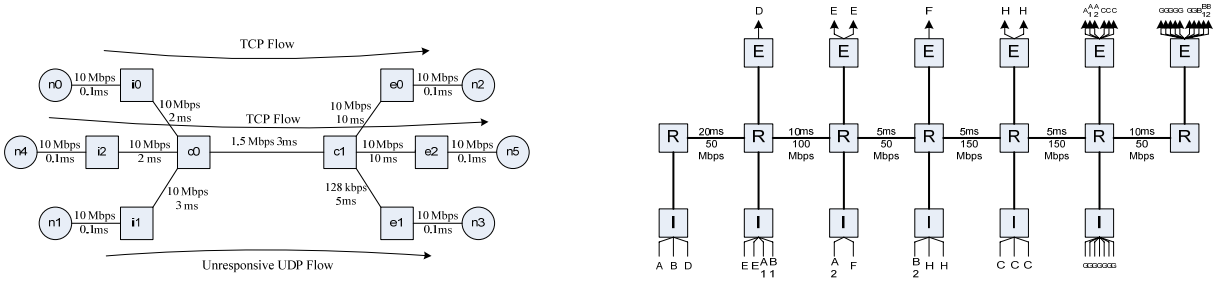


Figure 32: Two example topologies together with traffic configurations used in checkpoint evaluation process.

Large amount of simulation data has been gathered on the operation of flow aggregation levels, scheduler modifications and on connection admission control procedures. Standard simulation practices were followed throughout, that is, simulations were repeated a number of times for consistency, and standard deviations of results were observed to ensure reliability.

The proposed enhancements were evaluated in a number of different simulation topologies and traffic configurations.

6.3.1.3 Results from evaluation

Overall results from the evaluation and validation campaign have been positive. The enhancement of the NBP scheduling mechanism with flow aggregation, faster response, and admission control procedures leads to a practically realizable and deployable congestion control scheme which is very effective against unresponsive or misbehaving flows, without overloading the network with signalling information. As an example of the simulation results, Figure 33 shows the bandwidth obtained by two selected TCP flows in the extended dumbbell configuration as the function of the offered UDP load using various queuing disciplines.

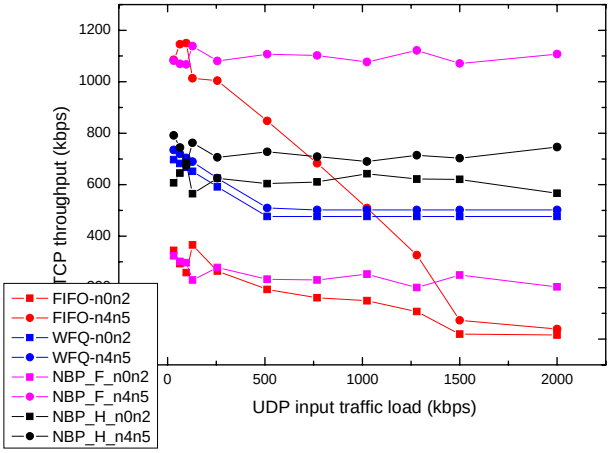


Figure 33: TCP throughput of n0-n2 and n4-n5, when there are 5 TCP flows between n0 and n2, and 20 TCP flows between n4 and n5. FIFO, WFQ (host ID), NBP (flow ID) and NBP (host ID) are compared.

As a second example, we demonstrate the effects of CAC procedure in the extended dumbbell scenario. To keep the number of flows manageable in the simulation, relatively high expected bandwidth threshold of 100 kbps was used for admission. TCP flows were timed to enter the network every 0.4 seconds, with 40 flows between nodes/networks n0 and n2, and 10 flows between n4 and n5. Constant UDP traffic with offered bandwidth of 2 Mbps was used between n1 and n3. Without CAC being used the throughput obtained by the flows between n4 and n5 degrades in expected manner (Figure 34, left panel), as larger number of new flows enter the core. When CAC is being used (Figure 34, right panel) some of the flows for n0-n2 connection are no longer allowed to enter the network as the overall available bandwidth is reduced, as can be seen from the even bandwidth obtained by the flows. In the end, nine flows between n0 and n2 were admitted together with eight flows between n4 and n5.

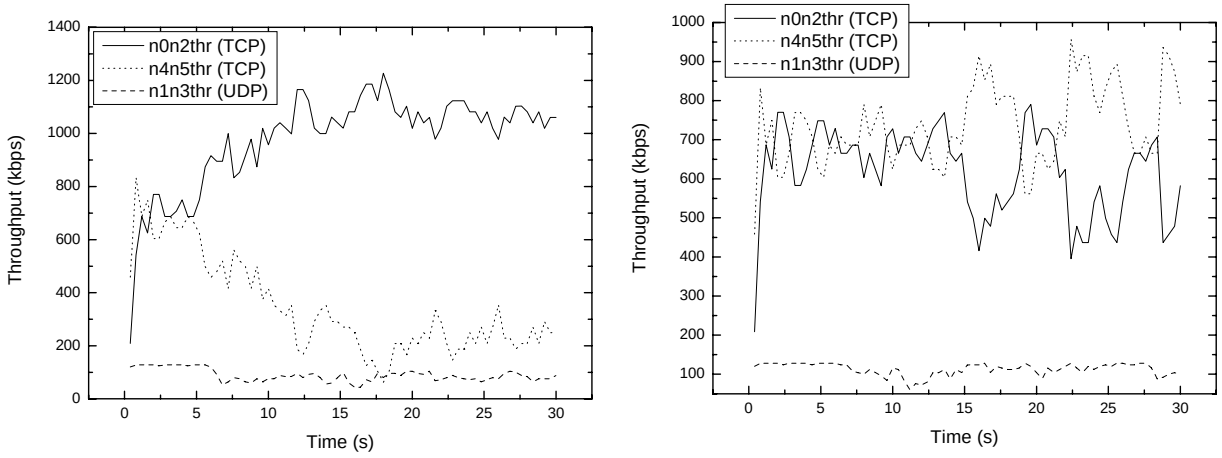


Figure 34: Throughput of different branches in extended dumbbell scenario with following configurations. Left figure: n0-n2 (40 flows) no CAC, n4-n5 (10 flows) no CAC, n1-n3 UDP (offered BW 2000kbps). Right figure: n0-n2 (40 flows) with CAC, n4-n5 (10 flows) with CAC, n1-n3 UDP (offered BW 2000kbps).

Due to the flexibility of the checkpoint architecture, several approaches warranting further study have also been identified. Some of these might be pursued in future phases of the project, as the remaining time in the present phase is devoted into performance evaluation and validation issues. We shall briefly discuss two topics that seem most interesting at the present time.

Misbehaviour detection. Checkpoints would be a very natural place for functionality to recognize and act upon flow misbehaviour. The classical examples include lack of congestion response, or, more seriously, sending of falsified congestion indication messages to induce unnecessary congestion responses in other nodes. Impact of the lack of congestion response in some nodes is, of course, mitigated by the standard checkpoint mechanisms already developed.

Checkpoints in the core. In the original NBP design checkpoints were located only on the network edges. In the context of connection admission control and network feedback placement of limited checkpoint functionality also in the network core should prove to be beneficial. This approach also raises further research issues, such as optimal checkpoint placement in the core.

6.3.1.4 Recommendations and migration issues

Despite all the advances in end-to-end congestion control methods, making the role of the network more prominent in congestion control activities appears to be unavoidable. There are two reasons for this. First, there is already a substantial amount of traffic in the Internet that is not subject to end-to-end congestion control. Second reason is the potential emergence of *purposefully misbehaving* traffic flows. The architecture and design presented can be used to guard the network core against these types of flows, and to guarantee responsive flows a fair share of the bandwidth.

The design is also very friendly to migrate to. Changes in the configuration of the network are as minimal as can be expected. Naturally, operators would need to update management software to configure the various parameters through which the checkpoint operation can be tuned, and to control the reprogrammable feedback and queuing blocks. In most cases this should be a minor effort, and with reasonable default configurations checkpoints should be deployable in almost “plug-and-play” manner.

It should be emphasised that the mechanisms providing added protection against congestion collapse are completely transparent to the end systems. Thus, no modification of the existing applications is necessary. Even to benefit from extra available feedback, changes are only necessary to the operating system networking protocol stack. Semantics towards the application do not change.

Finally, the modular design of the checkpoint architecture can be used to facilitate migration to other transport protocols form TCP and UDP that have prevailed in today's Internet. This would be especially useful for XCP-type of protocols that have proven to give suboptimal performance in the case of no available feedback from the network. Naturally, for optimal performance core routers should be updated as well, but in many cases this is highly impractical.

6.3.2 Utility-fair congestion control

Utility fair congestion control is a novel approach for efficient bandwidth allocation in networks. In contrast to most traditional approaches that try to share bandwidth equally among flows (e.g. TCP), utility fair congestion control shares bandwidth based on application-specific quality metrics, represented by their utility functions.

Utility fair congestion control is a feedback based protocol originally presented by T. Harks in [62] and further developed in [59] and [61]. It is based on optimization theory and an extension of work done mainly by Kelly and Low in [20], [58], [19].

6.3.2.1 Overview

The goal of utility fair congestion control is to provide a unified congestion control framework for both elastic and inelastic traffic. While the basic framework by Kelly and Low [20], [58] focuses on concave utility functions for elastic traffic only, Harks extended the approach to also support non-concave utility functions for layered real-time traffic [62].

It is a feedback based protocol employing all senders, routers and receivers. Routers continuously measure and compare the offered load with the capacity and update their *link prices.* If the load exceeds the capacity, the link price increases. If the load is below the capacity, the link price decreases until it becomes zero. Link prices are accumulated along a path and the resulting path price (*congestion measure*) gives the feedback for senders (see Figure 35).

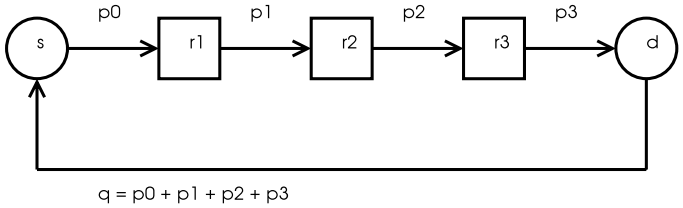


Figure 35: Explicit feedback loop. Link prices pi are summed along the s-d path. The total path price q is then returned to sender s.

Each sender is associated with a non-decreasing utility function that reflects its bandwidth requirements and adaptation behaviour. Figure 36 (left) shows three example utility functions, two concave functions U1 and U2 for elastic traffic and one multiple-step function U0 for layered real-time flows.

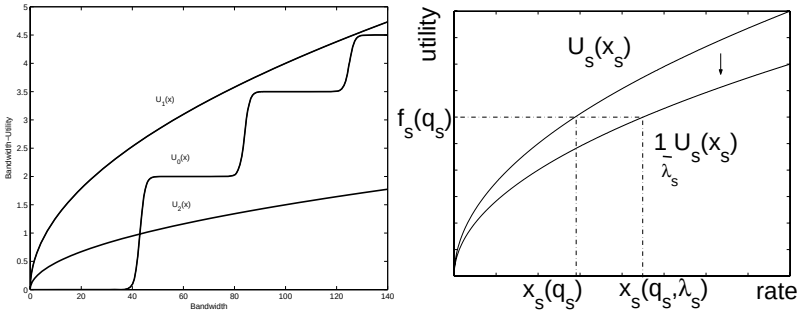


Figure 36: Example set of three utility functions (left). Mapping of available utility to sending rate and effect of scaled utility functions (right).

Using the path price, senders update their rates by first transforming the congestion measure q into an available utility f(q) ([62]), which is then mapped to a sending rate x(q) according to a utility function U(x) (Figure 36 (right)): $x(q) = U^{-1}(f(q))$. Both shape and scaling of the utility function has significant impact of the resulting bandwidth allocation. Figure 36 (right) gives an example of two utility functions with different scaling coefficients. As can be seen, the downscaled function achieves higher sending rates for equal utility values (and is in effect prioritized). For our simulations we use the strictly decreasing transformation function $f(q) = 1/q^{\kappa}$ which transforms a congestion measure q into an available utility. With $\kappa = 1$, rate allocation will be utility proportional fair. In [9], we demonstrate that $\kappa \to \infty$ approximates utility max-min fairness.

Complete mathematical formulations of the sender and link algorithms as well as references to the theoretical background can be found in [62], [9], and [61]. In [61], we also present a normalization approach that allows designing utility functions such that they behave comparably aggressive in the (possibly very) long term.

6.3.2.2 Evaluation

In [9], we presented the first set of simulation results. We have constructed utility functions for multi-layer real-time applications and demonstrated utility proportional and max-min fair bandwidth allocation for elastic and real-time flows. In [61], we introduced a method to dynamically prioritize individual flows through scaling of utility functions. In a pricing scenario where a user pays depending on the congestion and the chosen prioritization, we demonstrated advanced usage of the new possibilities. Using a time-constrained download as example, we showed how user agents can exploit dynamic prioritization to cost efficiently download files within a given time limit. Finally, in [60] we applied some of our findings to the framework of [19] and showed that automated dynamic prioritization of flows is applicable in other frameworks as well.

In the following, we present new simulation results of the utility fair congestion control approach in a dynamic environment where several flows share a link of varying capacity.

We consider elastic and real-time traffic. Elastic traffic is modelled by concave utility functions where the utility steadily increases with the available data rate. Real-time traffic on the other hand focuses on a finite set of data rates (layers) where the user perceived quality increases by leaps and bounds, while the gain in utility is small or negligible between two such steps ("sigmoidal utility functions").

Further, we distinguish between two types of real-time application behaviour. The first type comprises totally *inelastic* applications that can send only at one of a set of fixed data rates (layers). The smallest adaptation step that this type of application can perform is to switch from one layer to the next one, it cannot send at data rates in between. At the other extreme,

we see *smooth* real-time applications that are able to adjust their data rates as smooth as elastic applications, i.e. depending on the available utility, smooth real-time senders can send at rates between two layers (e.g. by discarding not-so-important packets already before transmission). Figure 36 contains example utility functions of elastic and smooth real-time applications.

We simulate and compare both types, start with smooth real-time senders and then compare the rate allocation with inelastic real-time senders. (For practical application examples, see [65] on MPEG-4 video streaming)

Scenario

Our topology models a path of four links with cross-traffic in both forward (R1-R2-R3-R4) and backward (R4-R3-R2-R1) directions. By changing the capacity of the R1-R2 link, we model a vertical handover situation as described in the moving networks scenario. On each capacity change, the link loses all currently queued packets.

We increase the number of flows over time as shown in Figure 38, and change the access link (R1-R2) bandwidth several times between 20Mbit/s and 10Mbit/s (Figure 39). The link R2-R3 has a fixed capacity of 15Mbit/s, link R3-R4 of 20Mbit/s. Access links of senders Sij and receivers Tij have a capacity of 100Mbit/s.

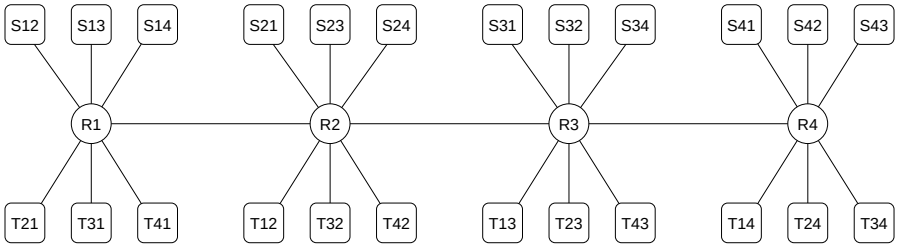


Figure 37: Network Topology

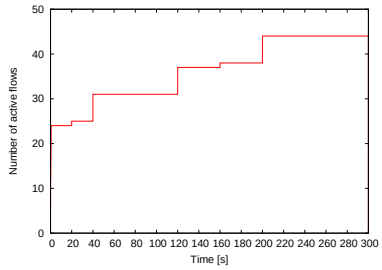


Figure 38: Number of active flows

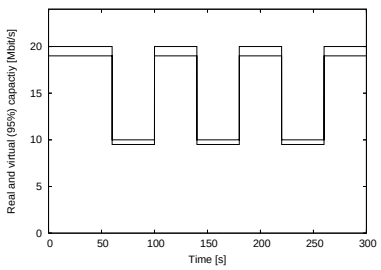


Figure 39: Capacity of dynamic R1-R2 link

Capacity of the R1-R2 link is reduced from 20Mbit/s to 10Mbit/s three times at t=60, 140 and 220 for 40 seconds. Note that a change of the R1-R2 capacity also implies a change of the network's bottleneck link (from R2-R3 to R1-R2 and back). At the beginning of each simulation, 24 low-priority elastic background flows are started for the full duration, 12 in each direction and two for every sub-path (R1-R2, R1-R3, R1-R4, R2-R3, R2-R4 and R3-R4), so that all links are fully loaded. Another 18 elastic flows (standard priority, full R1-R4 path) are started in three steps at t=40, 120 and 200. Finally, the simulation contains two real-time flows, starting at t=20 (R1-R2-R3) and t=160 (R1-R2-R3-R4) which will be discussed below.

Real-time traffic

In our earlier work [9], we simulated smooth real-time flows only. Because of the excessive layer-switching this can produce, we modified real-time behaviour to make them more robust

against transient fluctuations of the congestion measure. Real-time flows then switched layers only when the current layer was active for a predefined minimum amount of time. Still, a major problem was that unnecessary layer switches (especially down-switches) occurred whenever the network load was changing (e.g. new flows entering). Although the variations in the congestion measure were transient, a down-switch was performed and the lower layer was held for the minimum time between switches.

In this set of simulations, we modified our approach to reduce undesired layer switching in dynamic environments. The new approach tries to incorporate two observations. First, the congestion measure can oscillate when adapting to changed load or capacity. Second, a real-time flow should adapt in the smallest possible steps, i.e. it should switch layers one by one and not skip layers.

Our first step in implementing this is the introduction of a 'switch energy' variable and a set of thresholds. On receipt of acknowledgements, the layer corresponding to the new congestion measure is computed and compared to the current layer. If the new layer is above the current layer, switch energy increases, and decreases when the new layer is below. Once the switch energy exceeds a positive or negative threshold, an up- or down-switch to the next layer is performed and the switch energy is reset to zero. This approach has proven very effective in every situation where load or capacity changes, and prevented unnecessary layer switches.

Results

We now present results of two simulations of real-time flows, shown in Figure 40, Figure 41, Figure 42, and Figure 43.

Two perfectly adaptive real-time flows that can adjust to any data rate are shown in Figure 40 and Figure 41. The lower parts of the figures show the transformed network feedback (i.e. the available utility), the upper parts show the data rates (layers). The flow in Figure 40 starts after 20 seconds, while the second flow in Figure 41 starts later (after 160 seconds, when the background load is higher). The first flow traverses only the first two links of the topology, while the second flow uses all three links.

As can be seen, the first flow starts at the highest encoding layer and switches to the second layer due to a capacity reduction and increased background load (the available utility decreases below the layer border). Earlier changes of the link capacity (at 60 and 100 seconds) caused only short disruptions of the available utility, but did not cause a layer switch. The second flow (Figure 41) starts at the second layer and never sends at the highest rate. This is a result of proportional utility fairness. Since there is cross-traffic at every link, the second flow sees see same congestion measure as the first plus the non-zero link price of the R3-R4 link.

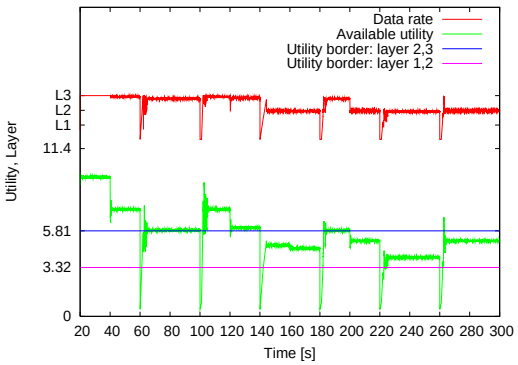


Figure 40: Adaptation of smooth real-time flow (R1-R2-R3)

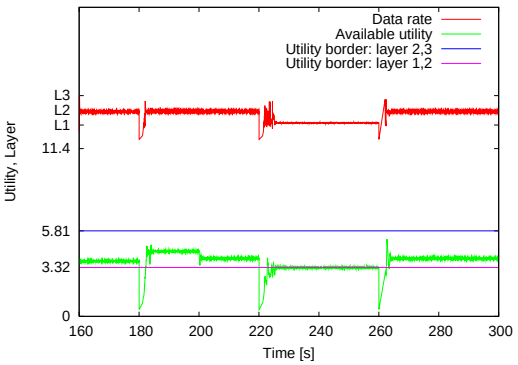


Figure 41: Adaptation of smooth real-time flow (R1-R2-R3-R4)

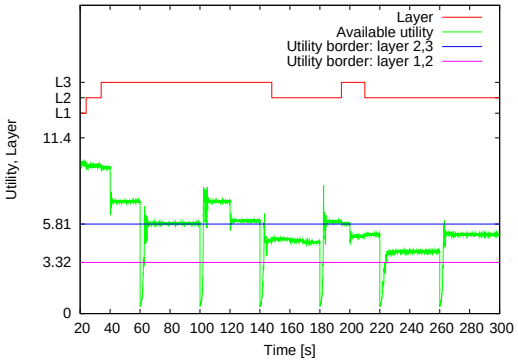


Figure 42: Adaptation of inelastic real-time flow (R1-R2-R3)

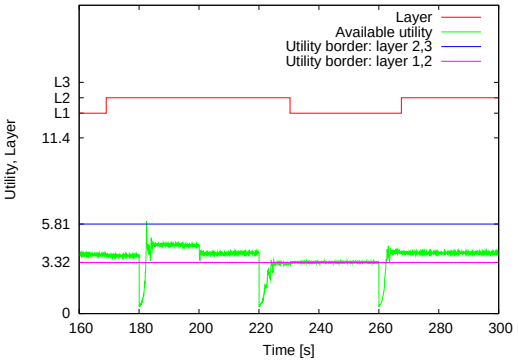


Figure 43: Adaptation of inelastic real-time flow (R1-R2-R3-R4)

In Figure 42, and Figure 43, we performed the same simulation as above, but with inelastic real-time flows that can only send at three discrete data rates (43, 83 and 125 packets per second). Here, the first real-time flow starts at the lowest encoding layer. The observed congestion measure, however, corresponds to layer three, which results in a switch-energy accumulation of two per second (two layers difference). After around five seconds, the switch threshold of ten is reached and the real-time flow switches to layer two. Still, the congestion measure corresponds to layer three, resulting in another switch-energy accumulation phase, this time at speed one per second (one layer difference). Thus, after another ten seconds, the real-time flow switches to layer three where it remains until the congestion forces it to switch down again after the capacity reduction at t=140.

The comparison of Figure 40 and Figure 41 with Figure 42 and Figure 43 shows almost identical layer-switching behaviour of smooth and inelastic real-time flows (except for a few seconds delay of the inelastic flows). Due to the introduction of the ‘switch-energy’, the inelastic flows in Figure 42 and Figure 43, however, remain stable when load changes occur (capacity changes, new background flows). This is an encouraging result. Although the theory behind utility fair congestion control requires flows to adapt smoothly, inelastic flows can be integrated such that their behaviour closely approximates that of ideal real-time applications.

6.3.2.3 Application in Ambient Networks

With utility fair congestion control deployed in Ambient Networks, we expect benefits for both users and network operators.

Users will appreciate better application performance and stability. We expect that performance will be better because bandwidth allocation will take application needs into account. Also, use of explicit feedback should result in better adaptation to changing paths or changing path

characteristics, and smaller router queues will result in shorter end-to-end delays. We expect better perceived stability of applications, first, because of the individual and more robust bandwidth allocation, and, second, because applications can use the congestion state to better adapt the data rate of application-level data (e.g. video and audio codices).

While users will benefit from end-to-end deployment, we also see possible benefits for network operators from local control loops. We expect that network operators can use utility fair congestion control as a tool to prioritize specific applications (like network management) or control bandwidth usage of traffic aggregates inside their administrative domain or Ambient Network (e.g. give low priority to transit traffic not originating from or destined for the network).Another crucial aspect in the deployment of utility fair congestion control is application support. Utility functions need to be designed and tuned to the individual application's needs. The full potential can only be exploited if applications are aware of the control framework.

6.3.2.4 Conclusion and Open issues

The work on utility-fair congestion control focuses on creating a unified congestion control framework for elastic and inelastic traffic. Therefore, we have created different types of utility functions for real-time and elastic traffic and performed simulations with competing elastic and real-time flows [59]. Further, we investigated advanced and strategic usage of our framework. Using a time-constrained download as example, we demonstrated how user agents can exploit the flexibility of the Utility Fair Congestion Control framework to achieve individual service that is beyond best-effort. In this context, we also presented a first idea how such usage can be charged [60],[61].

The NS-2 based simulation framework for Utility Fair Congestion Control currently includes various utility functions, prioritization through scaling, and example user agents for the download problem. However, it is still limited to an ideal environment in which all flows cooperate, i.e. all participants use "fair" utility functions and respond to congestion as described.

We are aware of a few other issues that, depending on the deployment scenario, require further attention:

- Routers need to compare load and capacity. In some environments, e.g. some types of wireless networks, estimation of the channel capacity is difficult.
- There are constraints on the maximum slope of utility functions. Some applications might not be adaptive enough to respect these constraints. In extreme cases, such flows might have a negative impact on system stability. In the evaluation above, we showed that inelastic real-time flows can be integrated such that they behave very similar to smooth real-time flows. However, here we assumed that there are a number of elastic flows capable to absorb load variations. It is still unclear, whether stability can be maintained in an environment with significantly more real-time than elastic flows.
- Finally, the interaction with traffic that does not follow the described framework can be an issue. Since flows that respect the network feedback back of when congestion increases, misbehaving flows might be able to degrade service of well-behaved flows. Also, integration of Utility Fair Congestion Control into existing protocols requires further research.

6.3.3 Evaluation of TCP-FEC

The first part of the evaluation shows that sending redundant data with the original data is providing better throughput for TCP applications. The methods for adding extra data to loss sensitive data for protection have been well established in areas such as satellite communication, networked link-layers and CD-ROM data. The question is how well does it work in communication networks that add protection at the transport or Transmission Control Protocol (TCP) layer.

TCP as we know it can be thought as backward error correction where errors are corrected after the original data, by retransmissions. Forward error correction, on the other hand, tries to correct errors before they occur by sending more data to allow the receiver to reconstruct the original data.

So the first question is how much more effective is TCP-FEC than TCP for a single flow for different loss probabilities. With uniform losses the values from a model developed by Padye is shown in the table below.

Loss (%)	Throughput of TCP without FEC	Throughput TCP with FEC	% Gain
0.1	1	0.9	-10.00%
0.2	0.72	0.9	25.00%
0.5	0.45	0.9	100.00%
1	0.3	0.9	300.00%
2	0.2	0.5	250.00%
3	0.15	0.32	213.00%
7	0.05	0.13	260.00%
10	0.04	0.06	150.00%
50	0	0	0.00%

Table 5: Throughput of TCP with and without FEC

It can be seen that for random losses the gain of using FEC can be well worth introducing it. Also for low or no loss situations there is a cost of sending extra packets. As can be see, 10% redundancy was added. This is of course can be dynamically controlled. If the receiver sees no duplicate acknowledgments then the FEC can be turned off at the sender. Note the effect of adding the redundancy is taken into consideration when calculating the losses. The load on the network is a factor, and in cases where it is not the case (e.g. wireless) then the gain can be up to three times that of normal TCP. If we now look at different TCP throughput in a higher capacity environment, 10 Mbits/sec and 100 ms delay with 10 flows using ns-2 we see the following results. Note the simulations were not done to compare with the above, rather than to be more realistic.

Loss (%)	TCP without FEC	TCP with FEC	% Gain
0	0.82	0.88	7.00%
0.01	0.55	0.81	47.00%
0.01	0.38	0.78	108.00%
0.02	0.32	0.69	115.00%
0.02	0.25	0.6	140.00%
0.03	0.22	0.45	104.00%
0.03	0.18	0.35	94.00%
0.04	0.16	0.31	93.00%
0.04	0.15	0.28	86.00%

Loss (%)	TCP without FEC	TCP with FEC	% Gain
0.05	0.14	0.22	57.00%
0.05	0.13	0.2	53.00%

Table 6: Simulations of throughput of TCP with and without FEC

As one can see the gains are more moderate but nevertheless well worth implementing in an Ambient network. Different topologies were tried, but with only one link have lossy characteristics. Since the errors are dealt with end-to-end it is not important if one or more links losses or drops packets. Other simulation results have shown that FEC gives throughput improvements between 20 ms and 100ms, where the throughput at 100ms is about +13%. Below 20ms it is not worth using TCP-FEC as the throughput is about 5% less, over 100 ms it is roughly the same as TCP-SACK.

6.3.3.1 Delay

Since one of the points of using TCP-FEC is to allow, serve and remove connections as quickly as possible, the delay incurred by TCP-FEC is an important factor.

Delay	Reno	TCP-FEC
20 ms	0.85	0.85
30 ms	0.83	0.85
40 ms	0.81	0.85
50 ms	0.8	0.85
100ms	0.77	0.85
200ms	0.72	0.8

Table 7: Delay of TCP with and without FEC

We have explored different designs for congestion control in Ambient networks. One of these is using TCP and adding forward error correction (FEC) to it. This will facilitate the flows entering, being serviced and leaving the network in a more controlled manner. The transfers and their rates can be better controlled, are more stable, and cope with the nitrogenous nature of an Ambient network than the TCP variations used in the Internet today.

6.3.3.2 Topologies and dynamically changing bandwidths

We have tried different topologies without any major differences. In the case where FEC is applied at the link layer then it is, of course that this protection is applied on links with losses. The problem with this approach is that losses further down the path need to retransmit again so the local link repairs can be unnecessary. Dynamically changing bandwidths could well be a reality in an Ambient network. As networks compose how well can TCP-FEC cope with paths that increase or decrease their capacities? We have looked at this in a separate study and found that in most cases TCP adapts well to the available capacity

6.3.3.3 Fairness of TCP-FEC with other TCP flows

This section deals with mixing TCP and TCP-FEC sessions. We have experimented with differing ratios of TCP flows to TCP-FEC. The minimum of one category was 1% and the maximum was 50%. Using Jain's fairness index we compared the throughput of 100 sessions using a 100 Mbits/sec link and a 10 Mbits/sec link. In all of the 100 Mbits cases the fairness index revealed an equal share was obtained for the flows with TCP-FEC, however it is lower

than those with normal TCP. As an example we looked at the TCP-FEC with TCP on lossless links and lossy links over two capacities.

Type	100 Mbits/s	Fairness Index	10 Mbits/s	Fairness Index
Max	100000	-	10000	-
TCP Reno	880654	0.99	86065	0.99
TCP Reno FEC	850654	0.97	83065	0.97
TCP with loss	42076	0.99	40543	0.98

Table 8: Jain's fairness index of TCP and TCP-FEC

The conclusion is that the fairness is a little lower than plain TCP, but not significantly. Also the fairness is somewhat lower on low capacity links, again not dramatically. If one would like TCP-FEC to be as fair, in terms of Jain's fairness index one can use ECN, which is discussed next. Also the topic of fairness has also been looked at in other aspects of NACCAN's work see section 3.3.2. We have looked at other fairness induces and found also good correspondence. Since throughput, or goodput is the main consideration then this is not that surprising.

6.3.3.4 Adding Explicit Notification (ECN) to TCP-FEC

Has the advantages of that when bit errors occur mean it is not necessary to decrease the window size. It is possible to differentiate bit errors and congestion errors, if an echo bit is received by the sender congestion-based loss (or there is an error in packet header). If no echo bit is received by the sender bit error in payload, then there is no need to send back echo, FEC corrects and does nothing. ECN is not needed in routers, only in the end hosts, so called endpoint signalling. ECN can be used to make TCP-FEC behave like TCP, ECN makes TCP-FEC more fair as the receiver sets the bit to indicate errors were detected, even if they were corrected.

Improvement of TCP-FEC with UDP as competing traffic in shown in the table below.

UDP load	Loss rate % with FEC	Loss rate % without FEC
25.00%	0.5	1.3
50.00%	0.7	2
100.00%	0.8	3
150.00%	1	3.5
200.00%	1.2	4

Table 9: TCP and TCP-FEC loss rates against UDP traffic

6.3.3.5 Effect of user terminal speed

To include the notion of speed and movement it is appropriate to consider how correlated the losses seen by the receiver are. The sender or receiver could be moving, but only the receiver can detect them in today's Internet infrastructure. Essentially the faster the moving terminals or even network, the lower the correlation of the observed losses. To in order to find a good

block length to use for the correcting codes, one should ascertain the effect of the speed on the error rate. This is shown below for two different speeds. In this case the speed is walking or running but this can be extended to any speed.

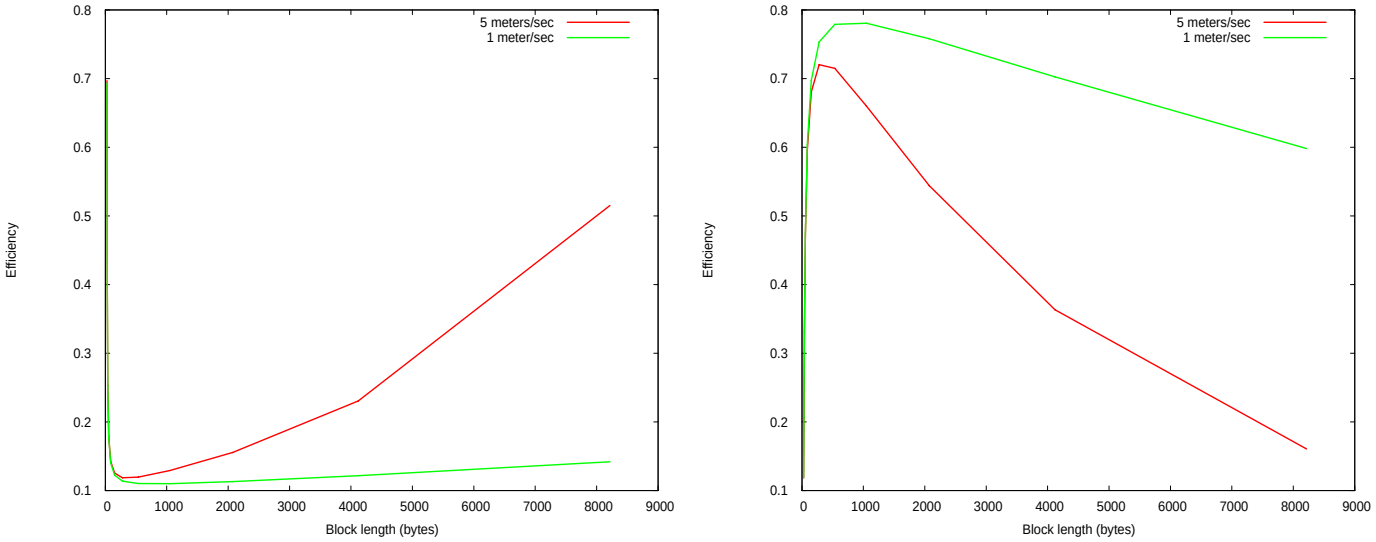


Figure 44: Block sizes needed for different speeds

6.3.3.6 Limitations of TCP-FEC

In this sub-section we discuss where TCP-FEC is either not favourable to deploy, or when its default behaviour should be as TCP. For networks with 0% packet loss it is better to use vanilla TCP or turn off the redundancy in TCP-FEC. For delays less than 20ms it is better to use normal TCP. For short transfers it is not necessary to protect these flows

6.3.3.7 TCP variants for Ambient networks

Since there are already a number of TCP variants to choose from, TCP-Tahoe, TCP-Reno, TCP-NewReno, TCP-SACK which one should Ambient use? The answer depends on the situation of course, protecting flows from numerous retransmissions is well provided for by TCP-FEC. This is especially the case for networks with high bandwidth-delay products where the capacity of the network is effectively wasted by retransmitting the lost segments.

6.3.3.8 Recommendations

We believe that TCP-FEC is a good option for transport layer communication with congestion control in Ambient networks. In most cases it serves the TCP flows well and reduces spurious retransmissions. It can also protect against more heavy losses, both in bursty and uniform losses. It can provide the default TCP behaviour without adding any redundancy. It can help protect flows with packet or frame losses and can be used to protect on-going flows, this is an appropriate for Ambient in order or have flows, serviced and out of the system. Further error models should be tested to find situations where TCP-FEC performs considerably worse, however in this case it is always possible to switch to the default TCP by not sending any redundancy packets. In the simulations, a naive coding scheme was used. With more advanced coding schemes better performance could be improved.

6.3.4 Contention resolving congestion control

6.3.4.1 Motivation

The multi-hop wireless connectivity environment is an integral part of several AN user scenarios. In such a decentralized environment, independent flows traversing a common wireless domain compete for wireless resources, even though they share no common control elements. The result is a congestion situation that cannot be analyzed or resolved by the

common control elements defined in other components of the AN-CC functional entity. Contention resolving congestion control is intended to address this situation.

6.3.4.2 Overview

The main benefits of this AN-CC approach are its effective, integrated treatment of energy management and congestion control and its suitability for the multi-hop wireless environment. In particular, the protocol is decentralized, extremely lightweight, and fully asynchronous.

The CC mechanism is based on cross-layer management of packet transmission and energy management schedules and allows node to locally adapt their energy schedules so as to minimize congestion. The techniques described in this section operate solely within the control space of the local wireless environment and are intended to be transparent to higher layer CC mechanisms. In addition, few assumptions are made about the operation of the lower layer MAC protocol, which is assumed to provide some form of CSMA collision avoidance and broadcast capability.

6.3.4.3 Protocol description

The contention resolving congestion control is based on adapting the transmission schedules imposed by the energy management mechanism so as to minimize congestion resulting from contention caused by transmissions associated with unrelated flows, as well as sequential transmissions along a flow. There are two main protocol components: the energy management component and the adaptation component.

Energy Management: In order to minimize energy consumption, battery powered mobile devices must maximize the amount of time that the network interface spends in a low power consumption sleep state. The scheduling of such sleep intervals is particularly challenging in our target environment, as there are no energy-rich control elements available to support synchronized buffering operations.

The protocol is therefore designed for fully asynchronous operation. Each node independently schedules its sleep/wake cycle, using common network parameters. A simple asynchronous quorum mechanism ensures that each appropriately timed sequence of transmissions will be received by all neighbors regardless of the phase difference between their sleep/wake patterns.

The neighbor phase discovery (NDP) exchange uses this quorum mechanism to allow a sender to determine the phase difference between itself and each receiver. Because the NDP will only be needed in the case of a new connection, an explicit phase change (below), or cumulative clock drift, the protocol is very lightweight. Moreover, there is no traffic in a quiescent network.

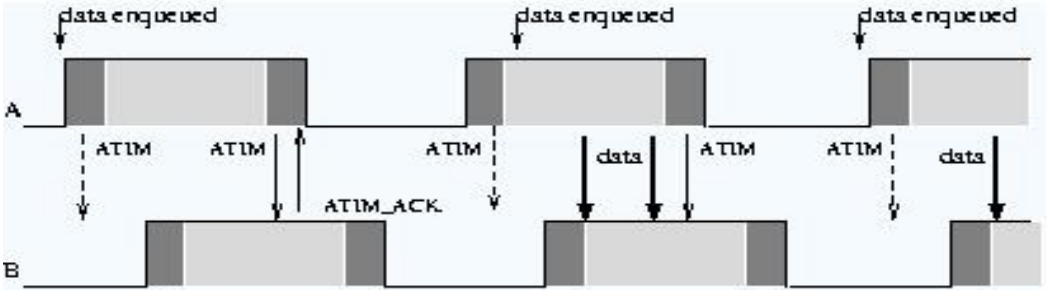


Figure 45: Operation of power save protocol and neighbor phase dicsovery.

A more detailed protocol description is available at [33].

Adaptation: The NPD exchange allows neighbors to determine the phase difference between their sleep/wake schedules and thus determine the transfer intervals (in addition to those ensured by the quorum) during which they can exchange traffic. It is clear that some distributions of these transfer intervals are “friendlier” than others. For example, congestion

due to contention between flows traversing links A-B and C-D, is most naturally reduced when the transfer intervals on these links are non-overlapping.

In effect, a good distribution of sleep/wake schedules can provide some of the beneficial aspects of globally scheduled links. However, it is not possible to negotiate such a schedule in an asynchronous, decentralized, multi-hop environment. Such schedules must therefore be approximated adaptively, allowing nodes to alter their sleep/wake schedules to reduce contention. Such adaptation, using randomized phase changes to avoid feedback loops, need not ensure contention-free links, as the underlying CSMA is ultimately responsible for MAC functionality

6.3.4.4 Interaction with other AN elements

Because decentralized multi-hop wireless communication is a relatively young area, there are no well-defined standards, even with respect to MAC protocols. Most existing work, including this work, is based on variants of IEEE 802.11 protocols.

Despite the low layers in which this congestion control mechanism operates, it has relatively few dependencies on lower layers. In particular, the proposed congestion control mechanisms adopt the transmission rules and behaviors of the underlying MAC layer. It is difficult to predict precisely how the mechanism would interact with some kinds of complex QoS functionality implemented directly in the MAC layer, however.

The operation of the contention resolving CC is transparent to higher layer CC schemes and other functional entities. It is intended only to provide greater efficiency and stability of transit times for flows traversing this challenging wireless environment.

6.3.4.5 Evaluation

Evaluation of the effectiveness and efficiency of these mechanisms currently focus on the impact that these proposed mechanisms have on the capacity of the network and the (work-in-progress) effectiveness of the adaptation mechanism.

A Matlab-based simulation has been implemented to perform static analysis of network capacity. The results confirm the well-known observation that the capacity of a multihop wireless network is very limited, but also show that the impact of energy management is also fairly limited. We conclude that significant saving of crucial energy resources (20-40%) justifies a relatively small (5-10%) loss in capacity utilization, even in quite dense networks.

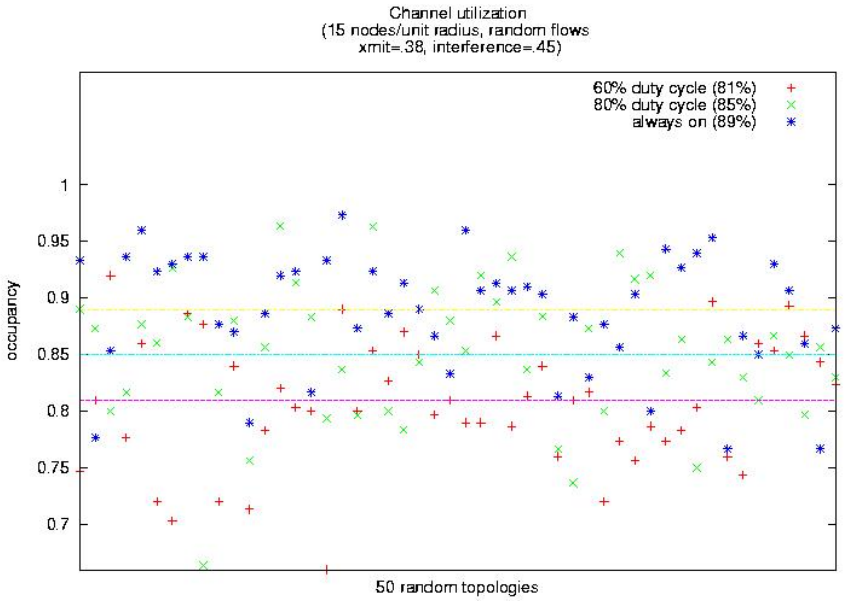


Figure 46: Channel utilization for various levels of power saving. A significant reduction in energy consumption results in a small reduction in capacity.

The importance of effective adapation is demonstrated by experiments showing that the steady-state capacity substantially depends on phase. This can be observed from the high variance in the number of flows supported, as the relative phases are adapted within the network.

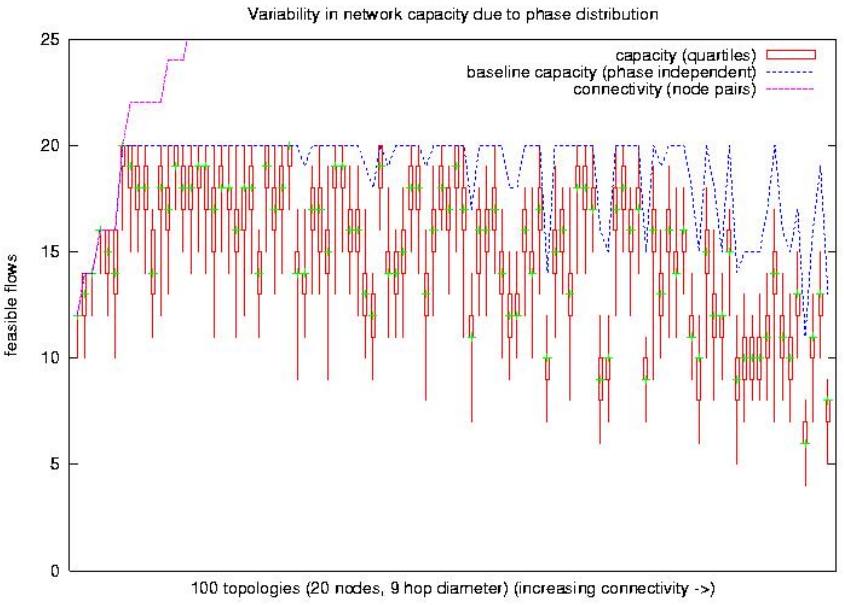


Figure 47: Variability in flow capacity due to phase distribution. The difference between the top and bottom quartiles is substantial.

6.3.4.6 Future and open issues

A key open issue is the analysis of the stability of the adaptation mechanism when applying randomized adjustment. Although simulation results may suggest that behavior is stable under a variety of conditions, this does not guarantee that unstable behaviors do not appear in real world operation.

6.3.4.7 Conclusion

The AN CC functional entity includes support for contention resolving congestion control for the multi-hop wireless environment, a key AN scenarios that requires localized control space mechanisms not provided by other elements of the CC functional entity

The contention resolving CC mechanism operates in conjunction with energy management schemes, adapting nodes' transmission schedules in order to avoid congestion due to inter and intra-flow contention. It is shown that energy management provides significant reduction in the node duty cycle, with minor impact of network capacity. The importance of adapting transmission schedules in response to congestion is demonstrated by the strong dependency between flow capacity and nodes transmission schedules.

6.4 Network Measurements

6.4.1 Evaluation of End-User QoS Techniques

In this work we have looked at two aspects of end-user QoS: transport protocols and algorithms for end-user controlled QoS and loss differentiation - differentiating between wireless losses and congestion losses.

6.4.1.1 Introduction to end-user QoS

In D3-2 [25] we described a proposal for end-managed QoS. This was proposed as an alternative to having QoS state within the network. Simply put, applications adapt their throughput to the congestion experienced in the end-to-end path. This is currently how data applications operate using TCP. If an end-to-end path can be provided with QoS guarantees under Ambient then there is no need for end-user QoS Techniques. This approach is for the case where the end-to-end path cannot be guaranteed e.g. where a non-Ambient network with no QoS support must be traversed. However, different congestion control techniques from TCP will be needed for streaming and real-time (conversational) traffic types. For such traffic it is more important to provide stability rather than maximum throughput. Another difference is that the applications are typically not fully elastic, but instead have defined rates at which they like to operate.

When people use video-conferencing systems, they typically prefer a stable codec and transmission rate than a higher average quality achieved through frequent codec changes. Frequent codec switching is not wanted. Therefore, the application should ensure before the start of the session that it has chosen a codec whose transmission rate is compatible with the long-term average available rate for the route. This can be achieved through probing the network prior to starting the data session. It then leaves the congestion control algorithm to respond to the short-term bandwidth fluctuations, and smoothes the resultant variation in delay with suitably sized (adaptive) jitter buffers. Should the probe return a negative result, the application should not start – this is a network busy signal.

6.4.1.2 Suitability of TFRC for end-user QoS

We have investigated the use of the TCP Friendly Rate Control algorithm [53] as a transport protocol for adaptive real-time applications. We first looked at the problem of non-rate controlled applications would have on the traditional best-effort Internet if the proportion of UDP grows significantly. There are two possible negative consequences.

- Congestion collapse is where the useful work achieved by the network decreases as the offered load goes up. One cause is where congested links send packets that would

only be dropped later in the network – i.e. if the flow crosses more than one congestion point.

- Reduction in fairness occurs because TCP traffic will back off in the presence of congestion, making room for the un-responsive UDP traffic, eventually allowing the UDP traffic to squeeze out TCP traffic.

We have performed a set of simulations to look at the effect of UDP on TCP under different queuing systems. Simulations were carried out using the ns network simulator [43].

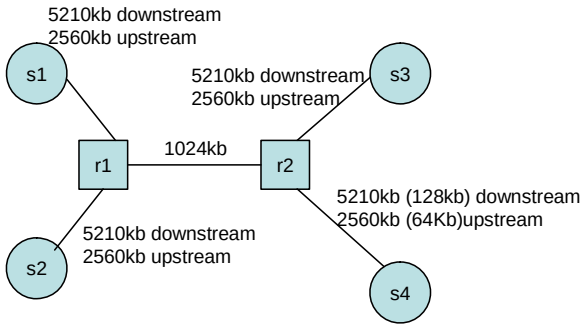


Figure 48: Simple simulated network

The network (Figure 48) has four end-system nodes (S1 to S4) and two routers (r1, r2). s1 and s2 are primarily sources and s3 and s4 are primarily sinks. In the simulations all TCP flows go from S1 to S3 and all UDP/TFRC flows go from S2 to S4. We show some typical simulations for illustration.

This first simulation (Figure 49) demonstrates the problem of unfairness. There are six UDP flows, 6 TCP flows, and the routers operate on a FIFO basis. The first TCP flow starts at 2 seconds and each subsequent flow starts at 0.2 second intervals. The first UDP flow starts at 10 seconds and each subsequent flow starts at 0.2 second intervals. At t=20, the UDP flows then begin to shut down. The link capacities are the larger values shown in the figure above. The graph shows bits received at the destinations. Here the UDP is seen to drive out the TCP before it reduces. If we had continued pushing UDP into the network, eventually the red TCP throughput goes to near zero. This graph also shows that the network can recover easily from this type of unfairness – TCP throughput rising as space is freed up by the UDP flows. This unfairness can be avoided by using Weighted Round Robin queuing mechanisms. In Figure 49 we show the benefit of this with respect to fairness – conditions as above. With a fair queuing system in place at the bottleneck router, the bandwidth is allocated more fairly – the green UDP line never exceeds the red TCP line.

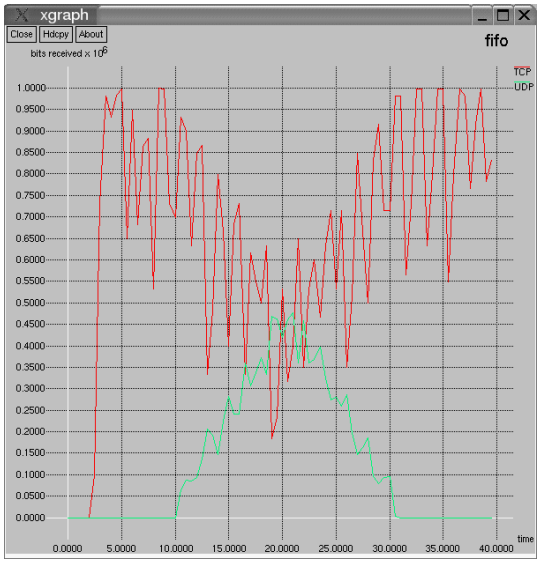


Figure 49: UDP drives out TCP

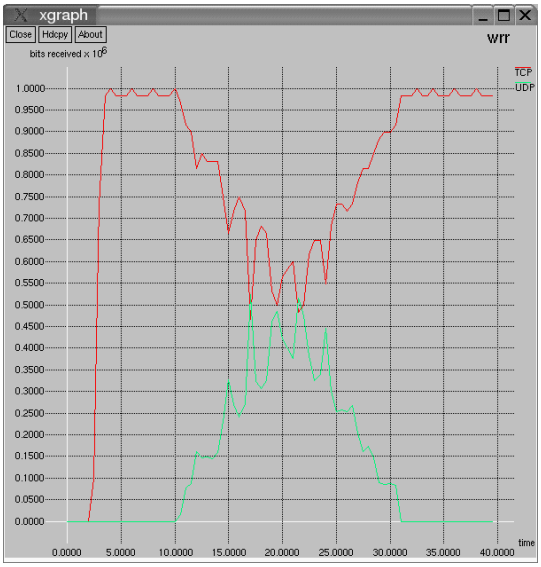


Figure 50: Unfairness to TCP can be controlled through use of WRR queuing discipline.

Note however that we believe that the majority of routers within the global Internet are FIFO not WRR.

Now, we look at a network where UDP is being sent via a restricted access link to its destination (the lower values for the R2-S4 link): Firstly (Figure 51), we have FIFO routers. Here we see the same drop in throughput of the TCP flows, but the throughput of the UDP flows is lower than in the previous simulation. This is because UDP drives out the TCP on the R1-R2 link but the packets are subsequently dropped on the R2-S4 link. This is an example of congestion collapse.

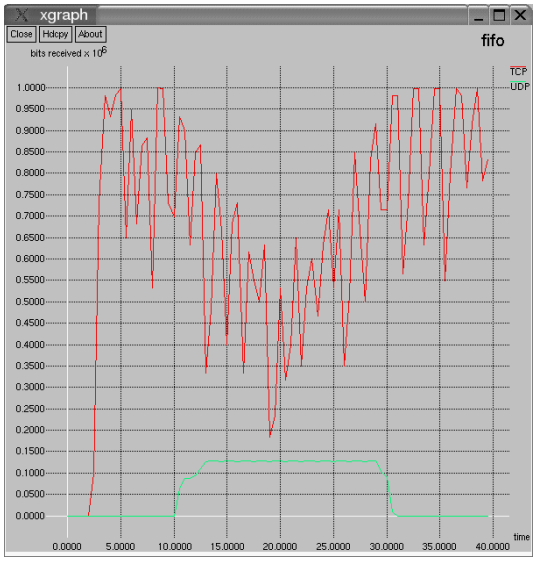


Figure 51: Congestion Collapse with multiple congested links

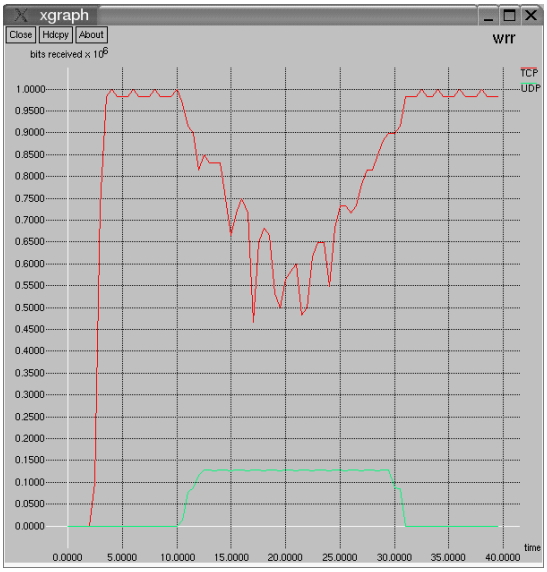


Figure 52: Congestion collapse not prevented by WRR

With WRR (Figure 52), the TCP flows get a fairer share but total throughput of the network is still reduced because of the UDP packets getting a fair share of R1-R2 only to be dropped downstream. Thus, the use of WRR queuing discipline at the bottleneck routers can control unfairness, but not congestion collapse.

Congestion collapse is perhaps viewed more easily in Figure 53. Here, in a scenario similar to above where there are two congestion points, using WRR, we increase the amount of UDP traffic being added to the network. Once the UDP is about 15% of the offered traffic (today it is 3%) we start to see a noticeable decline in the goodput of our network – its effective bandwidth is reduced – as well as a corresponding decrease in TCP throughput.

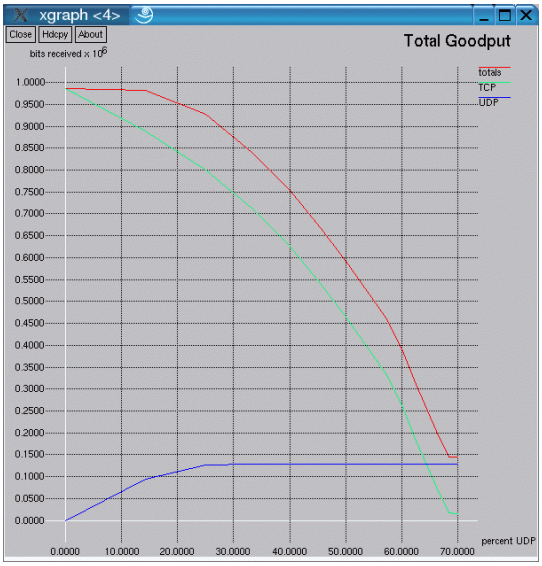


Figure 53: Congestion collapse is where the good-put of the network reduces overall.

6.4.1.3 DCCP AS A SOLUTION

Congestion control is required to be added to UDP streaming applications if this type of problem is to be prevented. There are three main media streaming products on the market: Real Player by RealNetworks, Windows Streaming Media by Microsoft, and QuickTime by Apple. The first two of these perform their own application level congestion adaptation. Essentially, the drive to develop TCP-Friendly rate control (TFRC) within the IETF was in part to provide such applications with a good congestion response that could be used instead of a proprietary solution.

The key requirements of the congestion control mechanism are to be able to provide unreliable data transport for real-time and streaming applications, whilst protecting the networks against congestion collapse and still being “fair” to TCP traffic. This can be used directly in the application, as the congestion control response required by the RTP standards. Alternatively, a UDP replacement protocol, DCCP has been developed, which can use TFRC. This could be used to hide the complexity of congestion management from the application altogether.

We have done some simulations using TFRC (a transport protocol within DCCP) rather than UDP. In this instance, we have assumed that a voice application is being carried by the UDP traffic.

Using the FIFO queues, we see (Figure 54) that, rather than the TFRC driving out TCP, TFRC flows can in fact be driven out by TCP flows! This is because the basic TFRC rate equation means that the TFRC flow would only gain its fair share of bandwidth if it was using 1500byte packets – an unreasonable assumption for real-time applications which may use packets as small as 40 bytes.

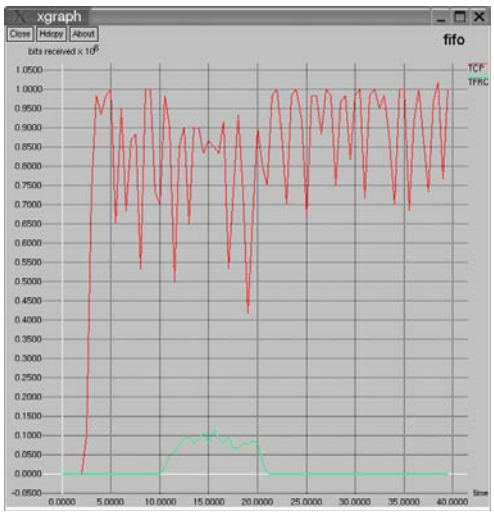


Figure 54: TFRC treated unfairly compared to TCP

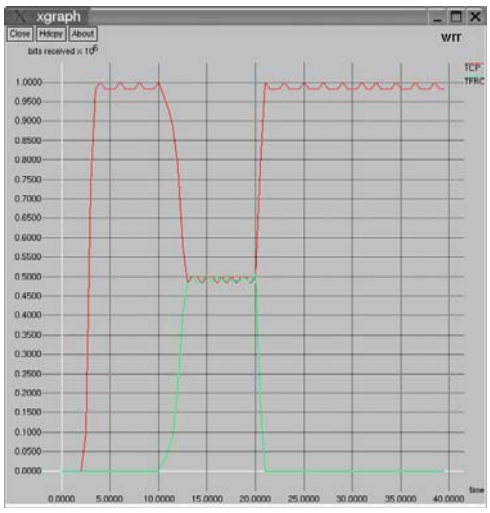


Figure 55: Benefit of TFRC on fairness using WRR

Switching to Weighted Round Robin (Figure 55), we see that the flows get a fair share of the bandwidth. We can also verify that congestion collapse (Figure 56) can be avoided.

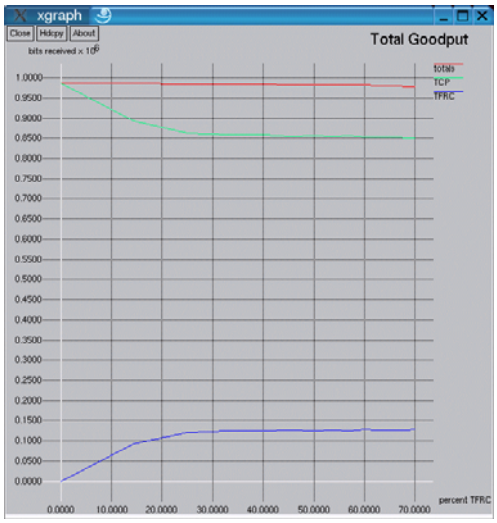


Figure 56: Congestion Collapse avoided with TFRC

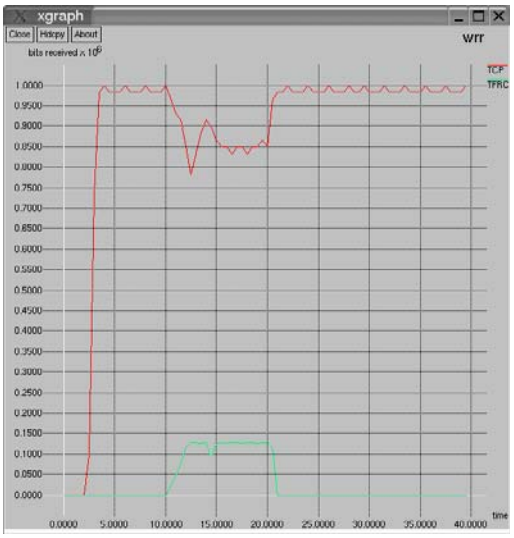


Figure 57: Although congestion collapse is avoided, the TFRC flows are not receiving enough bandwidth to be useful

Figure 57 shows the alternative view of congestion collapse avoidance. This particular illustration raises another problem - none of the TFRC flows is getting enough throughput to supply any application with sufficient bandwidth. In cases such as this, the application would have to give up trying to stream. If not, the user is likely to give up anyway (though probably after a number of re-tries).

6.4.1.4 Using DCCP in Ambient

From the simulations and it appears that TFRC will provide an effective means to congestion control real-time applications. DCCP (containing TFRC or another algorithm) could be a partial solution for end-systems to manage their own Quality of Service. It needs to be complemented with some additional technology so that end-systems perform their own admission control before starting flows. The use of probes seems the most likely technique. This was discussed in [25].

6.4.1.5 Introduction to Loss-Differentiation

If we are to use a congestion avoidance algorithm such as TFRC then it becomes important to differentiate between congestion and non-congestion losses. This is critical because wireless links can be lossy and hence can contribute significantly to the overall loss experienced at the receiver. Besides the data loss itself, an additional problem is that such losses maybe misclassified as congestion losses and hence trigger congestion control methods, resulting in a loss of throughput and under-utilization. There are a number of loss differentiation aschemes. They are introduced briefly here. More information can be found in the references.

Inter-Arrival Time Scheme: This scheme proposed in [55] discriminates wireless losses and congestion losses using inter-arrival time between two consecutive packets under the following conditions: (i) only the last link of the path is wireless, (ii) the wireless link is the bottleneck link of the connection, and (iii) the sender performs bulk data transfer. The scheme works best in isolation on wireless last hop topology and achieves excellent throughput while reacting to congestion well. However, with competing traffic in wired shared links, it misclassifies a significant number of congestion losses as wireless losses due to a very high upper limit and preventing the sending rate of a flow from being reduced which leads to severe congestion in the network.

Spike Scheme (based on Relative One-way Trip Time): The spike scheme proposed in [69] differentiates among degrees of congestion but does not explicitly differentiate wireless loss from congestion loss. It uses the Relative One-way Trip Time (ROTT) as congestion signal in the network. Since the plots of ROTT vs. time tend to show spikes during congestion, it is possible to identify network overloading by continuously measuring ROT T values. If the connection is in the spike state, losses are assumed to be due to congestion; otherwise, losses are assumed to be wireless.

Round Trip Time Scheme: The scheme, proposed in [41], estimates the cause of packet losses by detecting the knee-point in the load-throughput curve using RTT measurements and bandwidth estimation at the TCP sender. This scheme predicts congestion losses more accurately than the other schemes in low transmission errors. Moreover, the accuracy of packet loss classification for the RTT scheme decreases for packet error rate > 1%. The main reason for this behaviour is the inability of the bandwidth estimator to measure the bottleneck capacity when either the occurrence of wireless losses is high or the network topology becomes complex [57].

Rate Estimation Scheme: This scheme has been basically proposed for TCP Vegas and estimates the cause of packet losses based on rate estimates using Vegas predictor, $diff_V$. Given the parameters α and β (segments), when $diff_V \geq \beta$, TCP Vegas assumes that network is congested; when $diff_V \leq \alpha$, possible losses are assumed to be wireless losses and when $\alpha < diff_V < \beta$, the predictor assumes that the network state is same as previous estimation.

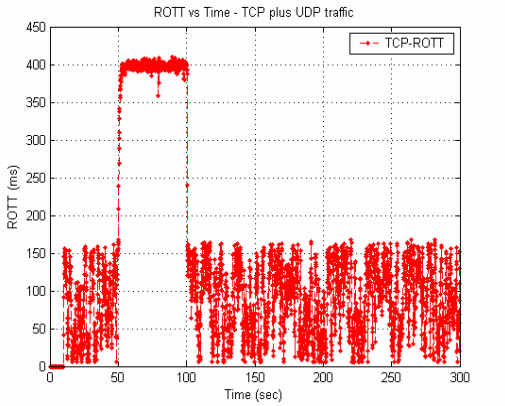
ECN Scheme: The schemes so far discussed address implicit mechanisms for loss differentiation. The losses can also be discriminated from the explicit signals received from the network. For example, using ECN (Explicit Congestion Notification) signal received from the network layer in an ECN capable network. ECN based loss discrimination identifies network congestion using ECN mark in the packets and invokes congestion control if the packets received are ECN marked, otherwise it assumes packet losses as due to transmission error. This scheme provides a more direct monitoring for network congestion. However it requires full support of all intermediate routers to mark packets during network congestion which may not be feasible in all networks.

6.4.1.6 Loss Differentiation in Multihop Wireless Networks

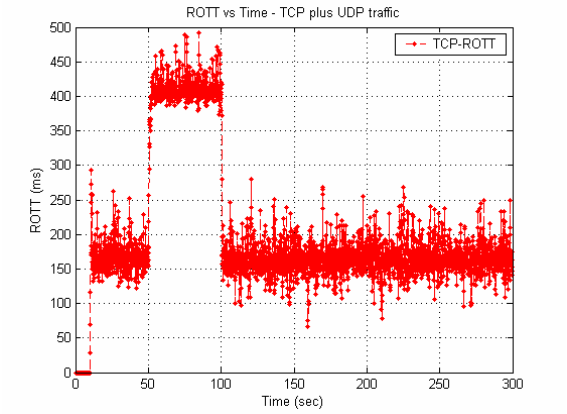
The modified spike scheme, based on relative one way trip time (ROTT) measurements at the receiver, has been studied over multi hop fixed wireless networks with TCP and TFRC protocols using *ns2* simulations.

First, the spike scheme has been implemented and studied on TCP. Figure 58 shows measured ROTT with time at the TCP receiver on different numbers of wireless hop connections when background UDP traffic is present between 50 s and 100 s. A clear variation can be observed on ROTT measurement during network congestion.

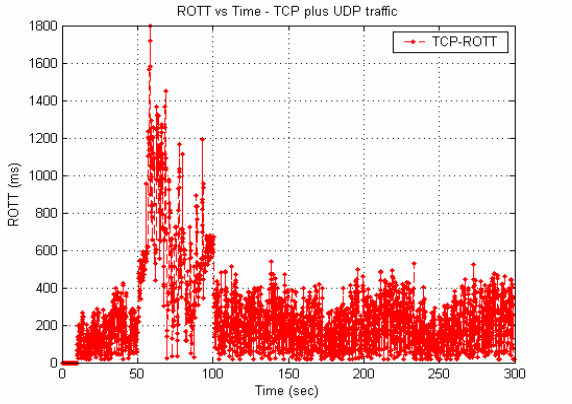
Congestion window variation has been observed at the TCP sender with and without the spike scheme over a 3 hop fixed wireless network when a background UDP traffic is present between 50 s and 100 s and the results are presented in Figure 59. It can be observed from Figure 59 that the TCP with spike scheme identifies the state of the network and invokes congestion control mostly during network congestion which improves the performance of TCP over multi hop wireless networks.



(a) Single hop network

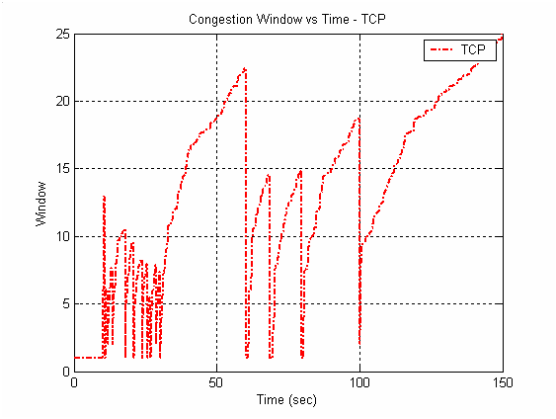


(b) Two hop network

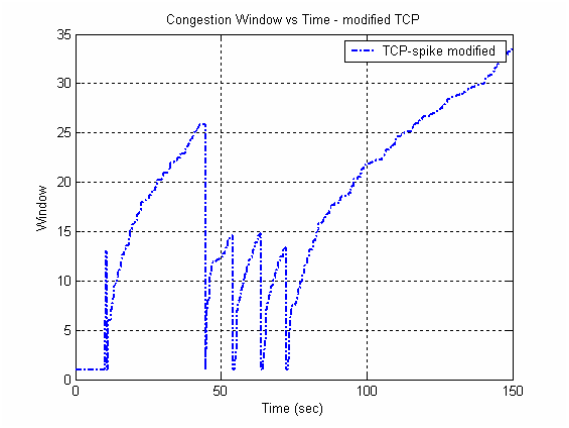


(c) Three hop network

Figure 58: TCP-ROTT measurement over multi hop fixed wireless network



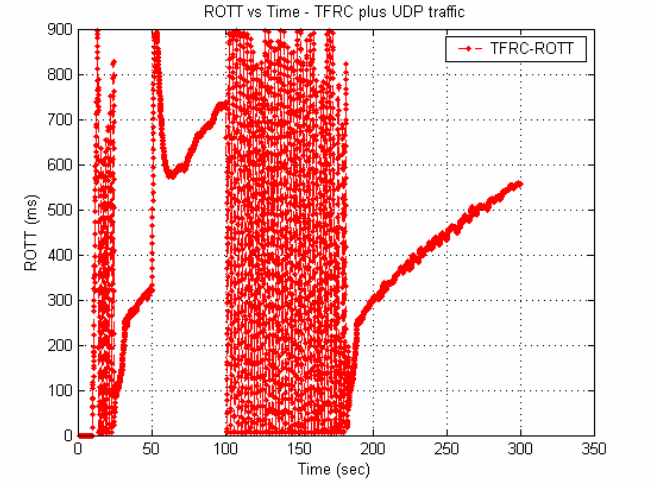
(a) TCP without Spike scheme



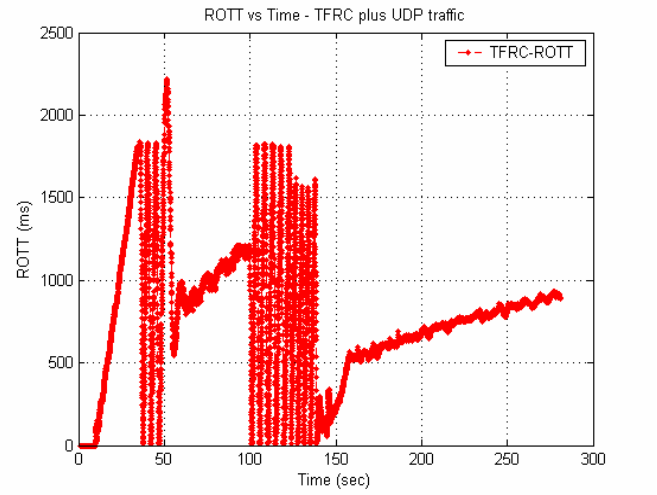
(b) TCP with Spike scheme

Figure 59: Congestion window variation at the TCP sender

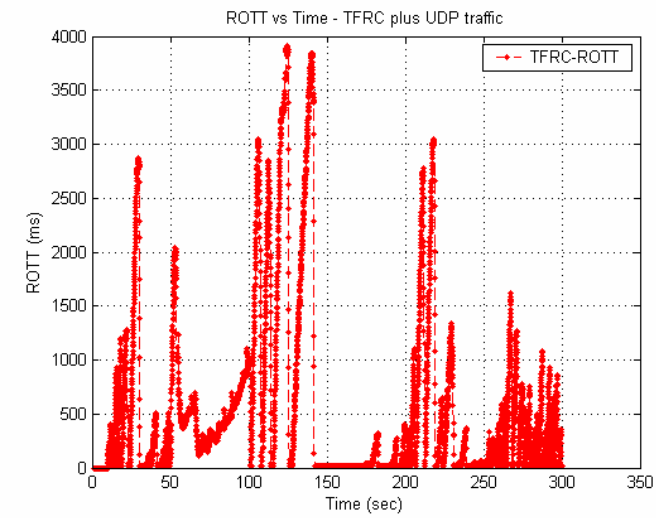
Also, the scheme has been implemented on TFRC and the ROTT measurements are obtained at the TFRC receiver with similar scenarios and the results are presented in Figure 60. The results obtained with TFRC protocol over multi hop wireless network are unsatisfactory. It has been observed that TFRC's behaviour is unstable over multi hop wireless networks and the ROTT measurements obtained at the TFRC receiver does not show any clear variation with network congestion. Further investigation is required on TFRC's behaviour in multi hop wireless networks and appropriate loss differentiation schemes for TFRC in such networks.



(a) Single hop network



(b) Two hop network



(c) Three hop network

Figure 60: TFRC-ROTT measurement over multi hop fixed wireless network

6.4.1.7 End-User QoS Summary

Our results give arguments both in favour and against the proposal to let end-systems manage QoS. From the simulations it appears that TFRC will provide an effective means to congestion control real-time applications. DCCP (containing TFRC or another algorithm) would provide a solution when complemented with some additional technology, such as probes, so that end-systems perform their own admission control before starting flows. However current implementations do not work well over wireless links. The current loss-differentiation algorithms perform badly at identifying between congestion losses and wireless losses. New algorithms or techniques would be needed.

6.4.2 QoS Measurements

Ambient Networks support both elastic and real-time applications. Many real-time applications require quality of service from the network. To support QoS-aware applications, one has to apply appropriate QoS measurement methods to find out the network QoS state. Existing approaches based on active and passive probing measurements have been investigated by IETF and ITU-T community. In active measurement, the measurement approach should not neglect extra probe traffic injected into the networks. Moreover, the QoS information should be available for individual applications.

The measured QoS parameters can be used as indicators for other QoS components such as admission control, congestion control, and SLS validation. This study has been taken into account of QoS measurement via a unified measurement framework. We will present our approaches for QoS parameters measurement.

6.4.2.1 A measurement framework

Definitions of QoS parameters and measurements are given in many documents by ITU T1A1.3 working group and by IETF IPPM working group. IETF IP Performance Metrics (IPPM) working group is active in this area [17]. According to our survey report [11] we conclude to take IPPM metrics into QoS measurement. There are a number of parameters that an application might have controlled for its traffic, such as packet loss, delay and jitter, and bandwidth. Many TCP-based applications often use packet loss to estimate the network state. It is end-user QoS issue. End-user QoS uses DCCP protocol to keep the network out of congestion. Our approach is active probing packet approach to capture the network information via packet pair dispersions. We use statistic models to reason about QoS parameters in order to minimize injected probing packet costs.

6.4.2.2 Packet loss measure

Existing approaches propose inference techniques for end-to-end multicast/unicast probes to estimate the loss rates on individual link. It is interested in inferring end-to-end loss performance on a given path because this performance ensures the relevance and perspective of applications. A HMM (Hidden Markov model) was proposed to capture the main characteristics of the loss process. The difficulty of packet loss, from perspective of applications is that probing algorithm is unable to send probes in exactly the same way as the target application does. HMM is used to construct the state of a path based on sampling pattern and the evolution of the path state. However, the HMM has two drawbacks: the convergence process is slow and depends on the number of states and the initial parameters; the forward-backward re-estimation algorithm suffers local optimization issue. This is the room for research improvement. We proposed a Maximum Entropy Markov Model (MEMM) [11] to target packet loss measurement in dynamic wireless network area. MEMM uses an online stripe packet-pair probing approach to estimate the packet loss rate that applications may suffer. The approach uses 2-order Markov chain to model the loss rate and loss burstiness. To reduce the computational complexity, we employ maximum entropy to estimate the parameters of the model. We use real Internet loss traces to validate the model. The results show the proposed approach has good accuracy.

6.4.2.3 Markov Model for Path Loss

Packet loss exhibits burrstones. This behavior has been described by a 2-state discrete-time Markov chain model. For stripe packet pair probing, intuitively if the first packet is lost, the second packet would be lost too. So using 2-state Markov chain model, the state X=$\{x_i\}_{i=1}^{N}$ is a stochastic process where x_i takes the value 0 if the i^{th} probing packet arrives successfully and the value 1 if it was lost. The probing sample interval (τ) is defined as some degree depending on the link capacity. The observed symbol is in the set $O=\{0,1\}$ per state. It is possible that during a good run, the receiver gets consecutive 0s or during bad runs consecutive 1s.

In 2-order Markov Model, the current state x_i depends only on the previous state x_{i-1} and x_{i-2}. 2-order Markov model is well to capture consecutive loss [38]. The state transition probability distribution defined by $A=\{a_{ij}\}$ where

$$a_{ij} = p(x_{ij} \mid x_{i-1,j-1}, x_{i-2,j-2}, O)$$

We will train 2-order Markov Model with probing packets to obtain a steady state probability $\{\pi_i\}$ and a_{ij}. The estimation packet loss is computed from:

$$\sum_{i=1}^{N} \pi_i a_{ij} \qquad (1)$$

6.4.2.4 Loss Burst Length

According to the previous study [38], the loss burst length distribution is IID random variables with the given distribution function. In MEMM, the burst length distribution of losses can be calculated from the steady state probability distribution (π), the loss probability [b_{jk}] and the state transition probability matrix [a_{ij}].

$$P(x_k = n) = (A * diag(B))^{(n-1)} * C^T, k = \{0,1\}, n \geq 1, C = (1-B) \quad (5)$$

It represents the burst of length n.

6.4.2.5 Average Loss Rate Evaluation

Equation 1 is for average loss rate. To verify the accuracy of the model, we use two loss traces as the input to the 2-order MEMM. The loss traces have been obtained from Yajnik [38]. Trace 21dec97 has a sampling interval of 20ms over 2.5 hours, and records details on a path from Seattle to UMass. Trace 26jun98 is a 6 hour trace from Atlanta to UMass with a sampling interval of 40ms. We segment the measuring window T=10000 sample packets. In this measuring window, we use maximum likelihood estimation to generate an initial transition probability to train 2-order MEMM. Then we apply the trained model to estimate the loss rate. We compared the actual and estimated loss rates in Figure 61 and Figure 62. We observe that the results are fitting quite well.

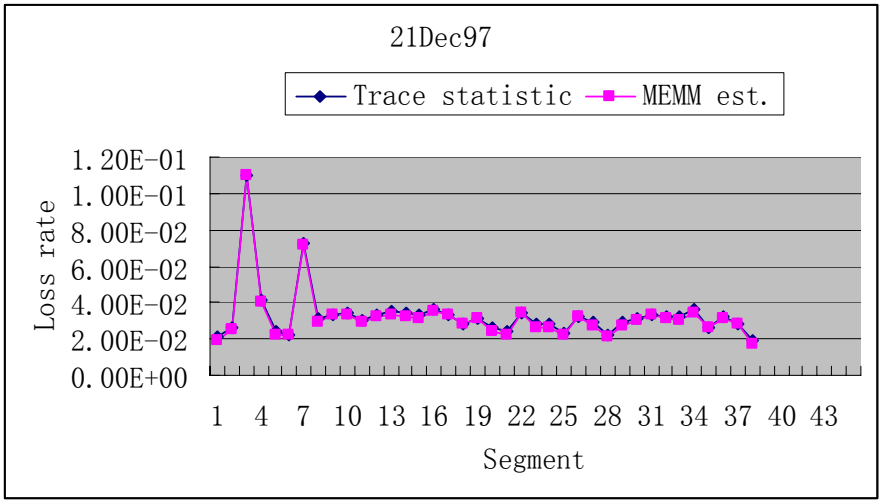


Figure 61: Loss rate

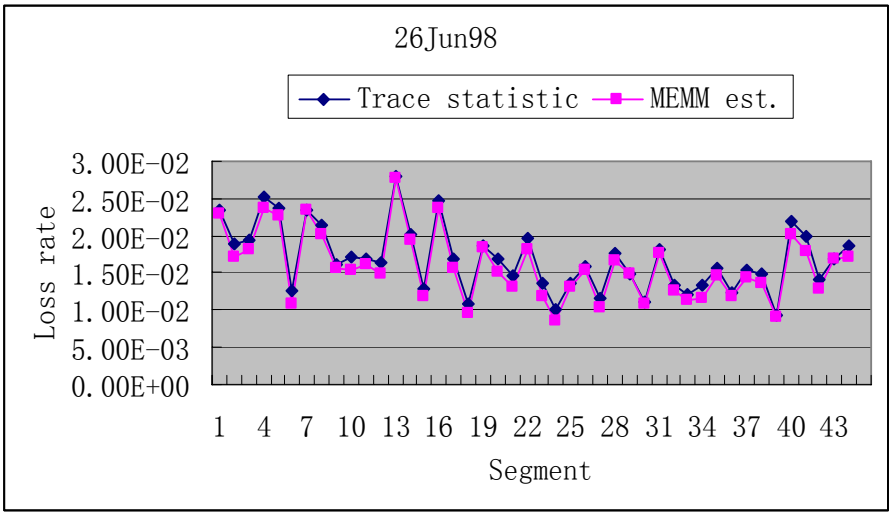


Figure 62: Loss rate

6.4.2.6 Loss Burstiness Evaluation

Loss burstiness length is another important performance indicator. The test data is the same trace as used in loss rate estimation. The results show that the model can provide reasonable estimation to the original data traces, especially in the short loss burst area in Figure 63 and Figure 64 for traces 21Dec97 and 26Jun98 respectively.

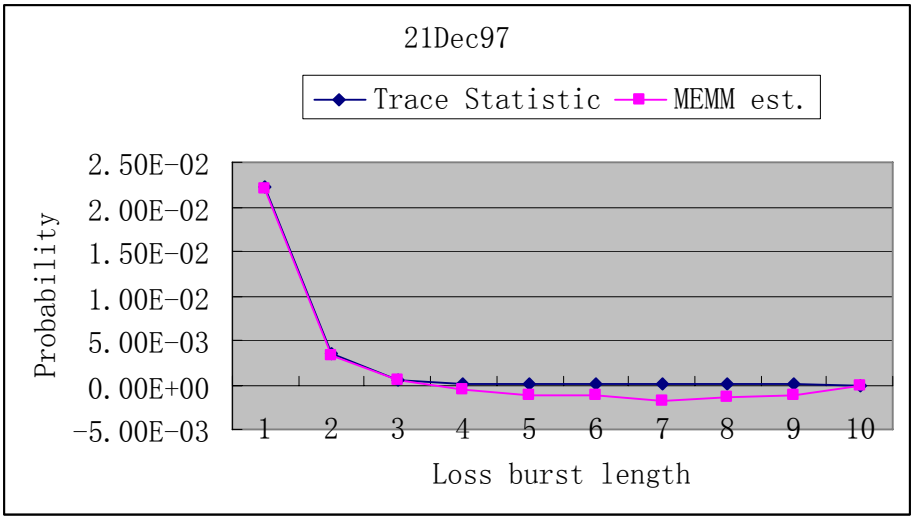


Figure 63: Probability of Loss Length

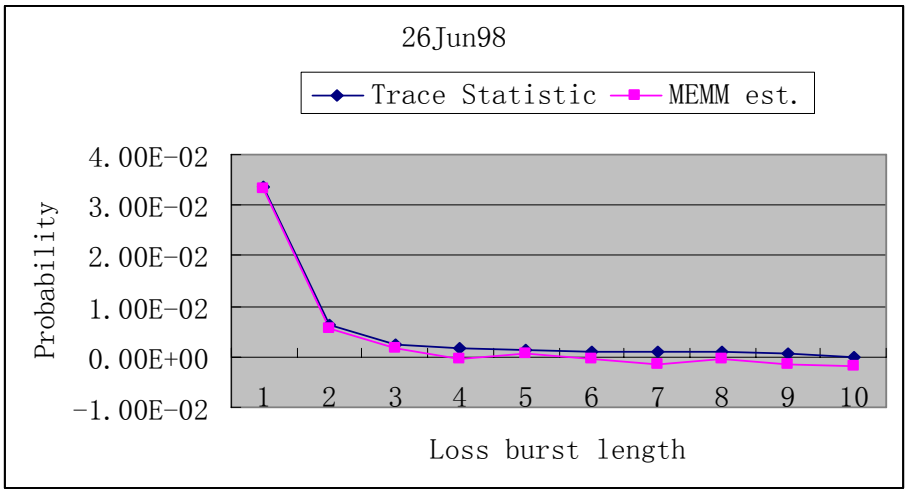


Figure 64: Probability of loss rate

6.4.2.7 Delay and Jitter measure

Compared to other parameters, delay and jitter estimation is relative simple. The key point is to collect packet pair delay and jitter. However, due to machine time skew, accurate one-way delay and jitter are still a research issue. It requires a perfect synchronization among the clock at the source and the destination of the probing packets. Using GPS-based synchronization and NTP (network time protocol) are considered too expensive to be used in wireless networks.

Our approaches are using statistic model to reasoning about the offset estimation. The one-way delay consists of propagation delay, queuing delay and errors (clock offset and clock skew) [40]. In asymmetric wireless environment, there are two forward and backward direction delays. The approach takes asymmetry into account. And collected probing data is analyzed by an estimation algorithm to remove clock offset.

Now the algorithm is doing the procedure. Let K={1,2,…,K} $\{S_k \mid k \in K$ be a set of packet size}. L_i be the size of i-th packet. Let b_j be the bandwidth of the j-th link and $\lambda = \sum_j 1/b_j$. Let μ be forwarding propagation delay and offset, μ' be the backward propagation delay and offset. Now the below formulations show queuing delays both in forward and backward direction.

$q_i = t_i^r - t_i^s - \lambda L_i - \mu$ (Forward queuing delay)

$q_i^{'} = t_i^s - t_i^r - \lambda^{'} L_i^{'} - \mu^{'}$ (Backward queuing delay)

Where t_i^r, t_i^s are the timestamp respectively. If we can estimate (λ, μ) and $(\lambda^{'}, \mu^{'})$ from probing packet measurement $\{(t_i^r - t_i^s, L_i) \mid 1 < i < N\}\}$ and $\{(t_i^s - t_i^r, L_i^{'}) \mid 1 < i < N^{'}\}\}$, we can estimate the offset o as follows:

$$\hat{o} = \frac{\hat{\mu}^{'} - \hat{\mu}}{2}$$

Unknown parameters λ and μ is estimated from the measured data. The bandwidth b_j can be measured by the below approach.

6.4.2.8 Capacity and Available Bandwidth Measure

Available bandwidth measurement [66] has been studied for a long time because of its importance to many applications such as adaptive streaming, congestion control and network diagnosis. However, available bandwidth is generally considered difficult to measure due to its dynamics. Most of efforts went into the measurement of the bottleneck capacity. They are based on packet-pair or packet train probing. The available bandwidth is defined as

$$A_i = C(1 - \frac{g_o - g_i}{g_i}).$$

Where C is capacity, g_o, g_i is output intra-packet train gap and input intra-packet train gap respectively.

To estimate asymmetric link capacity C by varying packet size, we adopt the CapProbe approach [35] via varying the packet size ratio between the probe (P1) and the ACK packets (P2), and observing the forward link capacity ($C^{'}_{1,}$ i.e. the probe packet size P1 divided by the dispersion time T'). The same metric of backward link capacity estimation is $C^{'}_{2}$. According to over-estimation or under-estimation issue of the measurement, the measure procedure is to increase P1 gradually to $\frac{p_2 x C_1}{C_2}$, $C^{'}_{2}$ converges to C_2 and $C^{'}_{1}$ converges to $C_{1.}$

6.4.2.9 Conclusion

We have compared and analyzed the advantages and drawbacks for existing technologies for QoS measurement. We also propose a general measurement framework for measuring QoS parameters in Ambient Network. To estimate packet loss, a Maximum Entropy Markov Model was proposed which shows good accuracy and has faster convergence. Measurement of delay and jitter requires clock synchronisation. A statistical model can be used for remove clock offset and skew which then makes it possible to estimate the delay and jitter accurately in the absence of synchronisation methods like GPS or Network Time Protocol. To find out available bandwidth, packet probing can be used although the estimates are slightly influenced by the probe packets themselves.

6.4.3 Evaluation of Interworking Functionality

As [68] states, Interworking functionalities (IWF) may extend legacy networks for a more compatible usage of their infrastructure and make them more and more compatible with the AN world.

For initial phases of development of Ambient Networks, maybe the easiest way to migrate legacy systems to Ambient Networks is the introduction of a set of Interworking Functionalities in order to simulate the behaviour of an Ambient Network in a legacy network. This section will provide a general view of those functionalities that should be included in legacy networks in

order to allow them to interact with Ambient Networks, since, depending on the specific underlying network, the set of functionalities can vary. For each network, we will have to identify which functionalities are required. E.g., for 3GPP mobile systems, most of the functionalities of a generic ACS (connectivity, context provisioning, security …) are present. Therefore, in 3G systems, it could be possible to implement an ACS with a reduced set of functionalities. The latter could control the functions provided in the legacy by means of the necessary translation functionality which maps the requests from the ACS into the specific interface/protocol/mechanism exposed from the legacy control plane. This translation mechanism could be implemented by an Inter-Working Function (IWF) or by means of an ARI Abstraction Layer. Pursuing the step-by-step migration approach, the phases would add functionality not present today to the legacy 3G system.

6.4.3.1 Revision of the Interworking functionalities

As stated in [68], one of the migration phases is the installation of IWFs to emulate the AN behaviour in the legacy network in order to appear as an AN from the perspective of other ANs as a legacy network form the existing environment.

Taking into account the current status of the Ambient Networks architecture and the composition framework, we are going to try to evaluate which functionalities should be included in the entities in charge of implementing the interworking functions by means of identifying the set of interworking functionalities required in order to develop a QoS aware interworking framework.

6.4.3.1.1 General Interworking Functionalities

Once the IP level connectivity is assured between two networks, in order to design a general QoS interworking framework, at least the next functionalities must be addressed:

- Signalling: both networks must be able to exchange signalling information in order to allow dynamic agreements.
- Admission control: if QoS is going to be assured, some admission mechanisms must be included and some mechanisms for avoiding congestion, too.
- Marking packets, if the legacy network has a DiffServ schema.
- Service mapping, how services are provided in the other networks.
- Routing resource reservation
- Monitoring: performance of measurements in order to validate the QoS
- AAA
- Charging

6.4.3.1.2 Interworking between legacy and Ambient networks

The last functionalities are general; that means, all these issues should be taken into account when an interworking solution between different networks is going to be developed. Next we are going to try to identify the specific functionalities to be included allowing for the specific design of Ambient Networks. Specifically, we will focus on both connecting Ambient Networks framework [1] and the management of composition for Ambient Networks [25].

6.4.3.1.3 Requirements for legacy and ambient networks

Taking into account the heterogeneity of interworking scenarios, the interworking framework should meet these requirements:

- Map technology specific locators to globally unique hierarchical ACS abstract locators in each AN.
- For ACS abstract locators, use e.g. IPv6 global routing prefix, with locally assigned subnet ID and interface ID.

- Each AN has a globally unique name, e.g. an IPv6 global routing prefix.
- Open to different types of name resolution mechanisms, e.g. traditional DNS, or distributed hash tables.
- Open to different types of intra AN routing mechanisms, e.g. mobile ad-hoc network routing, PNNI style, OSPF style,, etc., depending on the type of network.
- Open to different types of interdomain routing mechanisms, e.g. BGP style.
- Open for arbitrary network topologies.
- Host ID and locator split.

The first approach to facilitate the interworking between legacy and ambient networks, only takes into account the utilisation of the L2 and L3 capabilities of the legacy network. In this case, the legacy network has its own framework for providing QoS. The connectivity between different networks is assured by gateways.

Since no negotiation framework is defined for phase 1a, the gateway controls all the interworking functions. Specifically, these entities will be in charge of performing the functionalities required for any general interworking framework. That means, these entities will be in charge of mapping different services (e.g. selection of the PHB), triggering authentication procedures. Obviously, all these processes will be configured manually: manual mapping, manual agreements between operators.

Another important issue is how to provide QoS if no signalling mechanism is defined in order to negotiate a composition agreement. Therefore, if for example, network A wants to establish an agreement with network C and, in order to do that, must use the transport bearer of legacy network B, how can network A evaluate the performance of the legacy network B?

- If the legacy network is QoS-unaware, a good way is the use of a QoS measurement framework, able to evaluate the value of some QoS parameters such as delay or jitter.
- If the network implements some QoS mechanism (such as DiffServ) a good way to facilitate offline agreements could be the introduction of Inferring QoS mechanisms, that means, network A is able to model the performance of the legacy network (in the example, network B) by means of processing some measurements obtained with the developed QoS measurement framework (6.4.2). By this way, network A can decide if the use of the legacy network B is a good way to send traffic to network C.

Another option to allow the migration from legacy networks to ambient networks could be the introduction of interworking functionalities, in such a way that the legacy network appears as an AN from the perspective of other ANs and as a legacy network from the existing environment.

The main attribute of this framework is that a signalling schema is defined: legacy network provides an ANI interface, that means a set of protocols able to exchange messages related to different functional entitys.

Moreover, the legacy network B must introduce new functionalities (or enhance current mechanisms) in its framework in order to deal with the composition process. In fact these functionalities are:

- Connectivity functions.
- Security functionalities
- Composition functionalities
- Functions related to QoS

6.4.3.2 Inferring QoS capabilities Functionality

6.4.3.2.1 Motivation

"Inferring QoS mechanisms" framework is a new functionality that may be introduced in IWU (Interworking Unit) in order to reduce manual configuration in gateways. Moreover, this functionality could be helpful in order to define a decentralised admission control. The main goal of the IWUs is to introduce a set of functionalities in the edges of the networks in order to simulate the behaviour of an Ambient Network.

Nowadays, interworking networks implementing QoS mechanisms (such as DiffServ, IntServ/RSVP...) requires an offline agreement composed of both business and technical issues. The technical part of the contract should include descriptions related to interworking operations such as mapping of traffic priorities between different domains (DiffServ networks), admission control (IntServ/RSVP), network resilience and protection, exchange of accounting records, etc. On the other hand, ANs will work in highly dynamic environments where peering agreements will be needed more frequently than now. The main goal of the "inference of QoS capabilities" task is to reduce the technical issues in offline agreements by means of proposing a framework for an automatic (and then self-configurable) mapping between networks with different QoS technologies when they have to interwork.

We propose to introduce a new functionality in Interworking Units that could improve their performance (e.g. less human interaction when new configuration is detected), make easier the out of box agreement (that is, reduction of the issues to be agreed) and also reduce times to develop a specific interworking framework and/or to reconfigure the rules of the interworking if a change in a QoS domain is detected.

6.4.3.2.2 Techniques for inferring QoS capabilities

Two main techniques have been analysed: Measurement-Based Characterization and classification of QoS-Enhanced Systems and Inference of networks characteristics via moment-based estimators. In the following paragraphs only a general view of the mechanisms is presented, more information is available in the Technical Annex.

Measurement-Based Characterization and classification of QoS-Enhanced Systems: where a framework for monitoring, measurement and inference of multiclass services for the online case is developed. The main goal is to use passive monitoring of system arrivals and departures to detect if a class has a minimum guaranteed rate and/or a rate limiter. Moreover, if such elements exist, a method to infer the maximum likelihood parameters is shown. Beyond single class, an algorithm for interclass relationships assessment is provided. For example, this research devises tests, which infer not only whether a service discipline is work-conserving or non-work conserving, but also the relationships among classes, such as weighted fair or strict priority.

The main novelty in this proposal is that it proposes an online mechanism based on passive measurements to be used in operational networks. The algorithm proposed considers the system model shown in Figure 65, where a two-stage server with both non-work conserving elements and a multiclass-class scheduler is drawn.

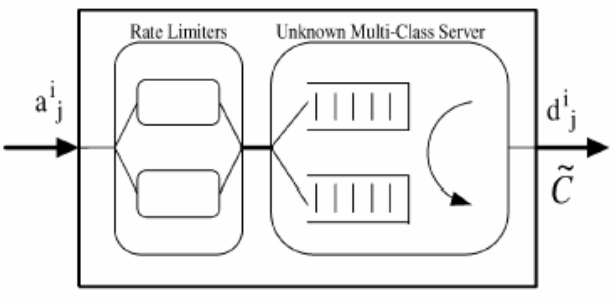


Figure 65: System model

In the inferring network characteristics via moment-based estimators technique a methodology that will allow applications to know network state is presented. In this way, applications can use this information to adapt their transmission rates to the inferred network capacities. A simple queuing model based on the single bottleneck link assumption is developed in order to estimate available bandwidth, buffer capacity and ground traffic intensity. Two inference models have been studied according to the queue models M/M/1/K and M/D/1/K.

6.4.3.3 Evaluation by simulation

The main goal of the simulations developed in the analysis of the proposals is to check the performance of the algorithms proposed in different scenarios. In fact, we are mainly focused on the evaluation of the methods to estimate the available bandwidth in a bottleneck, evaluation of the algorithms to infer the queue characteristics in a single queue and the estimation of the background traffic. It should be noticed that if these mechanisms work properly in different scenarios and with different types of background traffic, these frameworks could be used to infer the characteristics. Our work is mainly focused on the evaluation of the algorithm to infer the characteristics of a single queue In order to do that, the Network Simulator 2 has been used.

The scenario used to evaluate the evaluation of the single queue mechanisms is shown in Figure 66.

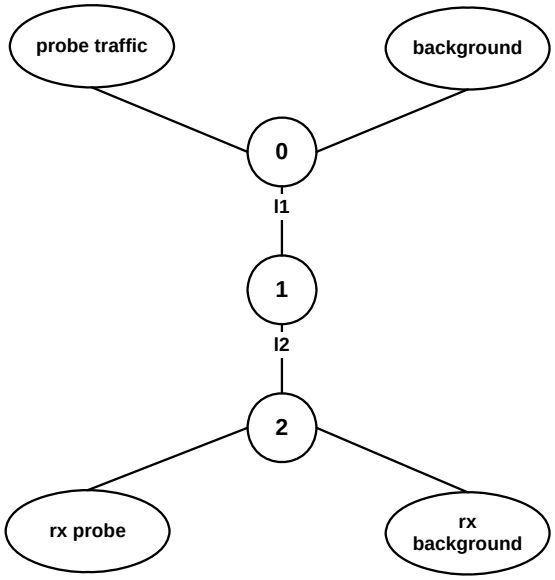


Figure 66: Scenario for the evaluation of the queue models

As it can be seen in Figure 66, this scenario is composed of 3 nodes and 2 links. We are going to infer the characteristics of the link between node 0 and node 1. Packets will be queued since the capacity in the second link is less than in the first one. The length of the queue can vary in different simulations.

Two types of traffic are considered: background traffic and probe traffic. In order to model the background traffic different approximations have been followed:

- 500 FTP over TCP flows
- Distribution of Poisson packets
- Others (aggregation of TCP and UDP flows)

For the probe traffic, we have considered only the a poisson flow of packets have been used.

We obtained the next parameters from the traces of the ns-2 for the probe traffic: arrival and departure times, losses and percentage of the occupancy of the server.

The goal with the described scenario is to emulate a single queue as it is represented in Figure 67. In order to obtain the characteristics of this system, we are going to model this queue as a M+M/M/1/K server.

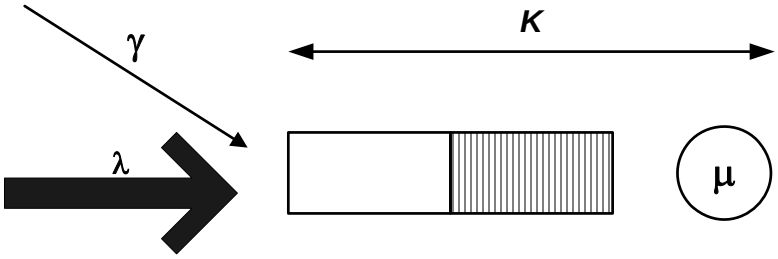


Figure 67: Queue model

For a length of the queue of 10 packets, we obtain the next results:

- Probability of the utilisation of the server (*U*): 0.99984
- Loss probability (P_L): 0.589807
- Average Response Time (*R*): 0.0166431

In this case we can say that:

$$U = \rho(1 - P_L) \rightarrow \rho = \frac{U}{1 - P_L} = \frac{0.99984}{1 - 0.589807} = 2.43748674$$

$$K = \frac{1}{\log \rho} \log\left(\frac{P_L}{1 - U}\right) = 9.217369839$$

So this could be a good approach to emulate the behaviour of the scenario since we have to take into account that the background traffic is not modelled as a Poisson packet flow.

The evaluation with other queue models and background traffic can be seen in the Technical Annex [63].

6.4.3.4 Conclusions on Interworking Functionalities work

According to the migration roadmap, two main interworking scenarios between legacy and Ambient Networks have been identified:

- In the first one, corresponding to the migration phase 1a, only layer 2 and 3 capabilities are going to be used. In this scenario no signalling mechanism between the legacy and the ambient network is considered. For this reason, the only way to characterise the performance of the legacy network is by means of the measurement of the QoS parameters (such as the delay). If the legacy network implemented some QoS mechanism (such as DiffServ), it could be interesting to discover it to be able to infer the performance of the network. For this reason, an overview and the evaluation of the mechanisms to infer these characteristics have been done.
- For the migration phase 2a, the legacy network includes some Interworking Functionalities (that could be present in a distributed way) that allow the legacy network to provide an ANI interface to other Ambient Networks. In order to face the composition process, at least the next functionalities must be included: connectivity, security, composition functionalities and other related to the provision of QoS. Since some legacy networks implement some of this basic functionalities (such as security, basic connectivity functionalities), it could be possible to define a simplified ACS in charge of managing the rest of the functionalities available in the legacy network and able to translate the requests from the ACS to the specific interface/protocol/mechanism provided by the legacy network. So, in this case, one important interworking function to be implemented would be the mapping between the ACS requests and the functions available in the legacy network.

6.5 Migration

6.5.1 Evaluation of Ambient Designs in Relation to the Migration Principles

We have evaluated [5] the designs of (selected) work packages to assess how well they had adhered to the migration principles. We found that the Ambient project has taken seriously the need for its designs to be migration friendly with activities to specifically look at this. The project has established a number of migration principles and migration phases. The principles should aid the creation of migration friendly designs and the phases [13] show a deployment path for working with legacy networks.

The work packages chosen for this evaluation were:

- 1: Architecture
- 3: Network composition and connectivity
- 4: Mobility and moving networks
- 8: Network management

Constrain contamination and contain co-ordination: The Ambient architecture is modular adhering to the contain co-ordination and constrain contamination principle. In this first phase of the Ambient project the designs are not sufficiently mature to conduct a detailed analysis. Most of the individual designs have taken this into account although not always explicitly stating how they have done so.

Some of the designs have explicitly addressed the issue of interworking with legacy networks. Specifically Work Package 4 has done the most work on specifying how the designs can use the migration phases for deployment. However, only UMTS networks have been taken into consideration. It would be good to see how other legacy networks could be incorporated, e.g. WLAN. In Work Package 3 the congestion control techniques have specifically addressed working with legacy networks and there is work on network measurements to cope with the case that there is a non-Ambient network in the end-to-end path.

Parties benefiting: Each of the designs discussed above have the potential to benefit most of the parties, if not all. The question is whether the benefits outweigh the costs of the status quo. A thorough analysis of business migration must address this issue and this is one of the tasks within WP-A of Ambient, Phase 2.

Simplicity: The use of the 3 high level interfaces (ANI, ASI, ARI) helps keep things simple. Furthermore these are not monolithic protocols but a set of protocols which can be deployed independently of any other protocol. However they are not clearly defined as yet. So a similar comment can be made as for the Constrain contamination and contain co-ordination principle.

6.5.2 Conclusion

A set of migration principles has been produced which will aid future designs of Ambient Networks. They become more important as designs are produced with more detail. Curently due cognisance has been taken of the Migration Principles but it is not always explicit in documentation. Detailed designs in phase 2 must address the principles. Each of the work packages in phase 2 should use the Migration phases to provide clear migration strategies.

7 Summary and Outlook

In this section we summarise the main results from phase 1 of Work Package 3 “Network Composition and Connectivity”. This includes general mechanisms for dynamically generating composition agreements between Ambient Networks. The concrete example of composition agreement that we have elaborated on is related to QoS and in particular the dynamic generation of Service Level Specifications. Also mechanisms for congestion control have been described and solutions for working with legacy networks and principles for migration.

The work on composition focuses mainly on research topics around the GANS protocol suite. It includes GANS design and specification of the GANS protocols, scalability study of GANS architecture and protocol suite by simulation and validation by building a prototype.

GANS architecture consists of two layers, i.e. application specific message signalling protocols (GSLPs) and a common message transport protocol (GTLP) taking care of the message delivery issues common to various GSLPs. These two layers provide an open framework and enable the separation of application specific signalling protocols from a common message transport.

The GANS transport layer GTLP consists of mainly two components, i.e. Extended-GIST and DEEP. The main task of Extended-GIST is to ensure end-to-end data transportation between different destinations end points identified with IP addresses. It also supports delivery of signalling messages with different properties and requirements. Extended-GIST is designed based on GIST developed by the NSIS group of IETF and is backward compatible to GIST.

DEEP is an independent component of GTLP to support symbolic names and handling of name resolution. It offers the capability of reaching over incompatible (IP) address spaces (NATs and firewalls).

The scalability of GTLP (part of GANS) was studied with simulations from overhead, processing and memory requirements, and name resolution latency point of view. The results show that signaling overhead in DEEP name resolution scales linearly with the number of ANs. The updating of GTLP-sessions can produce significant overhead if DEEP is used with the procedure. The latency of DEEP shows quite good scalability with low delays (under 100 ms) in intermediate DEEP-nodes, but with bigger delays the increase in latency is more considerable due to the number of intermediate nodes.

The most important procedures from EGIST overhead scalability point of view are the message association establishment and the refresh procedures. The apparent scalability problems illustrated by results (presented in Section 6.1.1) can be overcome by performing message association reuse, which should significantly reduce the signalling overhead of the protocol and, on the other hand, by actually using the indirect communication scenario considered as an extension to the GANS base specification, where there is one or more signalling gateway within each AN acting as the contact point from the outside point of view; this should reduce significantly the signalling overhead as well.

The memory consumption and number of search operations in endpoints indicate a linear dependency to the number of sessions. However, because of implementation choices, in some cases, it is possible to optimise the performance of the system with respect to these measures.

In what concerns the dynamic control of QoS Agreements we aimed to analyse INQA in terms of its scalability and convergence times, as well as its loop detection and robustness characteristics. Moreover, we analysed how much can INQA alone provide QoS assurances to non-elastic traffic and how well does INQA fit the current standardization efforts in 3GPP, ETSI and IETF.

Scalability and convergence times were analysed based on the INQA prototype, while comparing INQA performance with the implementation of the QoS-NSLP. First of all, we

analysed the signalling overhead, and the state required by both protocols when facing an increasing number of networks, flows and available services, in a static network scenario. Second, we analysed not only the above-mentioned criteria, but also the convergence time of both protocols in a scenario with moving networks. Results show that INQA behaves better than QoS-NSLP-alike protocols in both scenarios. However, it is clear that INQA may have scalability problems with an increasing number of available services, which may require the development of an aggregation mechanism (which will be done in the second phase of the project).

INQA's efficiency to detect loops and react to connectivity problems was analysed in a simulation environment. Results show that the INQA solution is able to detect all the possible loops with low bandwidth overhead and low computational overhead. Results also showed that INQA can detect connection failure immediately, and return to stable state in a short time. For instance, in a topology with 100 Ambient Networks, no system collapse occurs.

The study of non-elastic traffic provided evidence that a publish-subscriber approach, as used by INQA, may be used on its own to provide end-to-end QoS assurances to non-elastic traffic. This should be done in strict cooperation with a flow admission control mechanism and with additonal SLS advertisement rules. Moreover, it was also concluded that a solution that does not impose so many restrictions on INQA operations. This can be achieved by including an end-to-end mechanism capable of probing or activating SLSs. In order to have a better understanding of the optimal solution, more work is going to be done in phase 2, together with the development of an SLS aggregation mechanism.

To finalize the analysis of INQA, we studied its relationship with current standardization work. The results of our investigation showed that the INQA architecture is synchronized with 3GPP and ETSI directions, and that INQA signalling protocol may be standardized in IETF. In phase 2 of the project, we will try to make INQA-GSLP available to 3GPP and ETSI architectures as an IETF standard.

The network heterogeneity of Ambient requires a variety of congestion control mechanisms (CC) that can interoperate. Our evaluations of our selected schemes are: the checkpoint node mechanism has been shown to be effective against unresponsive or misbehaving flows, it is practically realizable and is deployable. The mechanism is transparent to the end systems which makes its migration simple.

The evaluation of utility-fair congestion control shows that users will benefit from better performance and stability, due to the fact that resource allocation takes application performance metrics into account. The results suggest good adaptation behaviour in a dynamic network environment with abruptly changing link bandwidths. In phase 2 the open implementation issues of utility-fair CC will be investigated.

TCP-FEC has shown to be better than plain TCP in nearly all normal operating circumstances. It protects the flows once in the system and hence can be removed from system not overloading it with unfinished flows. Again combined with the checkpoint node architecture, it can be deployed and beneficial in an Ambient network setting.

Finally the multi-hop wireless environment requires mechanisms not provided by the other CC elements, because congestion can be caused by nodes that do not share a common control element. The contention resolving CC operates in conjunction with the energy management, adapting nodes' transmission schedules to avoid congestion due to inter and intra-flow contention. The importance of this operation is shown by the strong dependency between the capacity and variation in nodes' transmission schedules.

A number of migration principles have been proposed. The three principles identified are:

1. Contain Coordination and Constrain Contamination
2. Simplicity, especially of interfaces

3. The benefit of migration should be obvious to (at least) the party migrating

Allied to this, work on network measurements has addressed a range of techniques for inter-working with legacy networks. The techniques include the following: discovering the available QoS on legacy networks, interworking with legacy networks and application adaptation.

Migration will be covered from a business perspective in Phase 2 of the Ambient project. Dynamic internetworking, including working with legacy networks will be undertaken. Our work has included looking at new transport protocols that have tried to keep within the Internet tradition of having control performed at the edge. The end-to-end the nature of existing transport protocols makes it difficult for the network to easily customise the routing, addressing, processing or transport of service communications. Phase 2 will be looking at these issues and producing Service-aware Adaptive Transport Overlays (SATOs).

One of the migration principles is "The benefit of migration should be obvious to (at least) the party migrating." One of the main objectives of phase 2 is demonstrating that Ambient Networks provide functionality that is beneficial to the end-user. This is perhaps one of the most crucial aspects to be addressed if consumers are to migrate to Ambient Networks.

8 References

[1] A. Bader, L. Westberg, G. Karagiannis, C. Kappler, T. Phelan, RMD-QoSM- The Resource Management in DiffServ QoS Model, NSIS Working Group, November 2005

[2] A. Gulbrandsen, P. Vixie, L. Esibov, *A DNS RR for specifying the location of services (DNS SRV)*, RFC 2782, February 2000.

[3] A. Kuzmanovic, E. W. Knightly, "Measurement Service in Multi-Class Networks"

[4] A. Kuzmanovic, E. W. Knightly, "Measurement-Based Characterization and Classification of QoS-Enhanced Systems", IEEE Transactions on Parallel and Distributed Systems, July 2003.

[5] A. Smith, Ed., "R3.3: Network Migration Approaches", IST–2002-507137-AN/WP3/R3-3, September 2005

[6] B Teitelbaum, "Why Premium IP service Has not Deployed (and probably never will)", available from the http://qbone.internet2.edu/, 2002

[7] C. Albuquerque, B.J. Vickers, and T. Suda, "Network Border Patrol", INFOCOM 2000.

[8] C. Jin, D. Wei and S. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance", INFOCOM 2004, March 2004.

[9] C. Kappler, Ed., "Scenarios, Requirements and Concepts", IST–2002-507134-AN/ WP3/D/3-1, Ambient Network WP3 deliverable 3-1, July 2004.

[10] D He, Nadeem Akhtar, Rahim Tafazolli1, Anne-Louise Burness, Philip Eardley, "On-line Packet Loss Rate Estimation", in: proceeding of the 16th annual IEEE international symposium on personal indoor and mobile radio communications, sep 11-14, 2005, ICC, Berlin

[11] D. He, N. Akhtar, K. Moessner, R. Tafazolli, "Measurements of QoS parameters", IST–2002-507134-AN/ WP3, Internal report. 2005-10-26

[12] D. L. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis", March 1992

[13] D. Moro, Ed., " D-1-3 Migration Strategies to Ambient Networks", IST–2002-507134-AN/WP1/D13, January 2005

[14] B. Ohlman, Ed., "AN Framework Architecture", IS T–2002-507134-AN/WP1-D05, Ambient Network WP1 deliverable D1-5, December 2005.

[15] B. Ohlman, Ed., "AN Framework Architecture, Technical Annex", IS T–2002-507134-AN/WP1-D05, Ambient Network WP1 deliverable D1-5 Technical Annex, December 2005.

[16] M. Brunner, Ed., " Ambient Network Management – Solution Design and Functions", IST–2002-507134-AN/D8-2, Ambient Network WP 8 deliverable D8-2, July 2005.

[17] E. Stephan, J. Jewitt, "IP Performance Metrics (IPPM) reporting Information Base", draft-ietf-ippm-reporting-mib-06.txt, July 2004

[18] F. Baboescu and G. Varghese, "Scalable packet classification," IEEE/ACM Trans. Netw., vol. 13, no. 1, pp. 2–14, 2005.

[19] F. Paganini, Z. Wang, J.C. Doyle and S.H. Low, "Congestion control for high performance, stability and fairness in general networks", IEEE/ACM Trans. on Networking, 13(1), pp. 43-56, February 2005.

[20] F.P. Kelly, A.K. Maulloo and D.K.H. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness, and Stability", Journal of the Operational Research Society, 49, pp. 237-252, 1998.

[21] GIST (see [23])

[22] GISTspec (see [23])

[23] H. Schulzrinne, R. Hancock, "*GIMPS: General Internet Messaging Protocol for Signaling*", Internet Draft draft-ietf-nsis-ntlp-05, Feb. 2005.

[24] IST-2002-507134-AN/WP8/D81, Ambient Network Management - Technologies & Strategies, December 2004

[25] J. A. Colás, Ed., "Connecting Ambient Networks – Architecture and Protocol Design", IST–2002-507137-AN/WP3/D/3-2, Ambient Network WP3 deliverable 3-2, March 2005.

[26] J. Qiu, E. W. Knightly, "Inter-class resource sharing using statistical service envelopes", in Proceedings of IEEE INFOCOM '99, New York, Mar. 99

[27] J. Riihijärvi, L. Wu and P. Mähönen, "Towards Practical Network-to-Network Congestion Control", accepted to ICN'06.

[28] J-C. Bolot, "End-to-End Packet Delay and Loss Behaviour in the Internet", Sep. 1993

[29] K. K. Ramakrishnan and S. Floyd, "The Addition of Explicit Congestion Notification (ECN) to IP", IETF RFC 3168, September 2001.

[30] K. Nichols, "Definition of the Differentiated Services field (DS Field) in the Ipv4 and Ipv6 headers", RFC 2474, December 1998

[31] L. Daigle, A. Newton, *Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)*, RFC 3958, January 2005.

[32] L. Feeney, M. Schläger and B.-A. Viken. Ed. "Scenarios, Requirements and Concepts, Technical Annex", IST–2002-507134-AN/WP3/D/3.1Annex, Ambient Network WP3 deliverable 3-1, Technical Annex, June 2004.

[33] L. M. Feeney. "Energy saving and traffic scheduling in multihop wireless networks". SICS Technical Report T2005:15. December 2005.

[34] L. Massouli and J. Roberts, "Arguments in favour of admission control for TCP flows," in Proceedings of ITC 16, June 1999.

[35] L-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, "End-to-end Asymmetric Link Capacity Estimation." IFIP Networking 2005, Waterloo, Ontario, Canada, 2005

[36] M. Mealling and R. Daniel, *The Naming Authority Pointer (NAPTR) DNS Resource Record*, RFC 2915, September 2000.

[37] M. Roughan, S. Sen, O. Spatscheck, N. Duffield "Class-of-Service Mapping for QoS: a statistical signature-based approach to IP traffic classification", ACM SIGCOMM Internet Measurement Workshop, 2004.

[38] M. Yajnik, S. Moon, J. Kurose and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss", in proc. IEEE INFOCOM, New York, March 1999

[39] Mockapetris, P. *Domain names - Implementation and Specification*, STD 13, RFC 1035, November 1987.

[40] Moon, S. B.; Skelly, P.; Towsley, D., "Estimation and removal of clock skew from network delay measurements" INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 1, 21-25 March 1999 Page(s):227 - 234 vol.1

[41] N.K.G. Samaraweera, "Non-congestion packet loss detection for TCP error recovery using wireless links," in *IEE Proceedings Communications,* vol. 146, no. 4, Aug. 1999, pp. 222-230.

[42] Network Simulator version 2, URL: http://www.isi.edu/nsnam/ns/tutorial/index.html

[43] ns, network simulator, http://www.isi.edu/nsnam/ns/

[44] P. Nikander, J. Laganier, *Host Identity Protocol (HIP) Domain Name System (DNS) Extensions*, Internet-Draft, Expires: August 2005, http://www.ietf.org/internet-drafts/draft-ietf-hip-dns-01.txt.

[45] Paulo Mendes, Jorge Andres-Colas, Carlos Pinho, "Information Model for the Specification of QoS Agreements among Ambient Networks", IEEE International Symposium on Personal Indoor and Mobile Radio Communications, Berlin, Germany, September 2005

[46] R. Kalden, Ed., "Connecting Ambient Networks – Protocol Specification and Simulation", IST–2002-507137-AN/WP3/R/3-2, Ambient Network WP3 Technical Report 3-2, September 2005

[47] R. Kalden, Ed., "Connecting Ambient Networks – Protocol Specification – Technical Annex", IST–2002-507137-AN/WP3/R/3-2/TA, Ambient Network WP3 Technical Report 3-2, Technical Annex, September 2005

[48] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, *Host Identity Protocol*, Internet-Draft, Expires: August 2005, http://www.ietf.org/internet-drafts/draft-ietf-hip-base-02.txt.

[49] R1.7

[50] Reynolds, J. Postel, *Assigned Numbers*, STD 2, RFC 1700, October 1994.

[51] RFC 2475: "An Architecture for Differentiated Services"

[52] RFC 3031: "Multiprotocol Label Switching Architecture"

[53] RFC3448, TCP Friendly Rate Control, http://www.ietf.org/rfc/rfc3448.txt?number=3448

[54] S. Alouf, P. Nain, D. Towsley, "Inferring Network Characteristics via Moment-Base Estimators"

[55] S. Biaz and N. H. Vaidya, "Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver," in *Proc. IEEE Symp. Application-Specific Systems and Software Engineering and Technology,* Richardson, TX, Mar. 1999, pp. 10-17.

[56] S. Biaz and N. H. Vaidya, "Distinguishing Congestion Losses from Wireless Transmission Losses: a Negative Result," in *Proc. of IEEE 7th Int. Conf. on Computer Communications and Networks*, New Orleans, LA, Oct. 1998.

[57] S. Bregni, D. Caratti and F. Martignon, "Enhanced Loss Differentiation Algorithms for Use in TCP Sources over Heterogeneous Wireless Networks," in *Proc. IEEE / GLOBECOM*, San Francisco, CA, Dec. 2003, pp. 666-670.

[58] S.H. Low and D.E. Lapsley, "Optimization Flow Control I", IEEE/ACM Trans. on Networking, 7(6), pp. 861-874, 1999.

[59] T. Harks and T. Poschwatta, "Utility Fair Congestion Control For Real-Time Traffic", In Proc. of 8th IEEE Global Internet Symposium, pp. 85-90, Miami, FL, USA, March 2005.

[60] T. Harks and T. Poschwatta, "On User Strategies in a Network Implementing Congestion Pricing", In Proc. of CCNC 2006, Las Vegas, USA, January 2006. To appear.

[61] T. Harks and T. Poschwatta, "Priority Pricing in Utility Fair Networks", In Proc. of 13th IEEE International Conference on Network Protocols, Boston, MA, USA, November 2005. To appear.

[62] T. Harks, "Utility Proportional Fair Bandwidth Allocation: An Optimization Oriented Approach", In Proc. of 3rd International Workshop on QoS in Multiservice IP Networks, Springer LNCS 3375, pp. 61-74, Catania, Italy, 2005.

[63] Technical Annex to D3.3

[64] UIT.T Y.1251. "Modelo arquitectural general para el interfuncionamiento "

[65] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Trans. on Circuits and Systems for Video Technology, 10(5), pp. 725-743, August 2001.

[66] X. Liu, K. Ravindran, B. Liu and M. Loguinov, "Single-Hop Probing Asymptotics in available Bandwidth estimation Sample-path analysis", IMC'04, October, 2004. Taormina, Sicily, Italy

[67] Y. Bernet, "A Framework for Integrated Services Operation over DiffServ Networks", IETF RFC 2998, November 2000.

[68] Y. Bernet, "format of the RSVP DCLASS object", IETF RFC 2996, November 2000.

[69] Y. Tobe, Y. Tamura, A. Molana, S. Ghosh and H. Tokuda, "Achieving Moderate Fairness for UDP Flows by Path-Status Classification," in *Proc. 25th Annu. IEEE Conf. Computer Networks (LCN 2000)*, Tampa, FL, Nov. 2000, pp. 252-261.

[70] Y. Xia, L. Subramanian, Ion Stoica and S. Kalyanaraman, "One More Bit Is Enough", ACM SIGCOMM, 2005.