Realtime mobile sharing of multimedia and context

Dennis Bijwaard

Realtime mobile sharing of multimedia and context

Dennis Johannes Adrianus Bijwaard

Composition of the Graduation Committee:

Prof. Dr. Ir.	T.	Mouthaan	(UT, EWI)
Prof. Dr.	P.J.M.	Havinga	(UT, PS)
Prof. Dr.	S.J.	Mullender	(UT, PS)
Prof. Dr.	L.J.M.	Nieuwenhuis	(UT, ICSM)
Prof. Dr.	S.M.	Heemstra de Groot	(TU/e, HNA)
Prof. Dr.	A.J.	Kassler	(KAU, CS)
Dr. Ir.	E.H.	Eertink	(Novay, INCA)
Dr. Ir.	G.	Karagiannis	(UT, DACS)

This work was conducted within the projects Freeband 4G+, Daidalos I & II (IST-2002-506997, IST-2006-026943), and SENSEI (IST-2007-215923).

Pervasive Systems Research Group
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente, The Netherlands.

UNIVERSITY OF TWENTE.

CTIT

CTIT PhD Thesis Series No. 12-238
ISSN 1381-3617
Center for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands

Keywords: Wireless Communication, WSN, Heterogenous networks, MultiMedia, Pervasive Systems, Service Platforms

This thesis was edited with gvim and Texmaker, typeset with LaTeX2e, and printed by Wöhrmann Print Service, Zuthpen, The Netherlands.

Cover design: Dennis & Sonny Bijwaard

ISBN 978-90-365-3498-7
http://dx.doi.org/10.3990/1.9789036534987

REALTIME MOBILE SHARING OF MULTIMEDIA AND CONTEXT

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 11 januari 2013 om 12.45 uur

door

Dennis Johannes Adrianus Bijwaard

geboren op 12 November 1969

te Breezand, gemeente Anna Paulowna.

Dit proefschrift is goedgekeurd door

Prof. Dr.	P.J.M	Havinga	(promotor)
Dr. Ir.	E.H.	Eertink	(assistent promotor)

Acknowledgements

I started rather late with my PhD utilizing what I have done and published in several research oriented companies in the past decade. Initially I thought my list of papers at that time was almost equivalent to a PhD. However, putting everything together as a coherent whole was more difficult and time consuming that I anticipated. And of course, more research was necessary to make it more actual and give it more body. I therefore thank my promotor Paul Havinga and assistant promotor Henk Eertink for guiding me during this effort, and giving valuable feedback on my research and writing.

All in all, it has been quite a challenge the past 2.5 years to get three more papers and a book chapter published, to work full-time at Inertia Technology and to give proper attention to my wife and 4 kids while writing the thesis. Certainly, some things had to suffer.

Therefore, I awe most thanks to my wife Sonny for being patient during this time, and taking much more than half of the responsibilities around the house and kids. Moreover, I thank her for all the evenings she worked so I could work on my PhD without distraction. I also thank my kids: Mandy, Wim, Tom and Daisy, for enduring the lack of attention, and for baring my temper after many late-nighters. Luckily, I could still bring them to school everyday, attend Judo and football matches, and see some plays they performed. Fortunately, they were mostly sleeping or reading in bed when I worked on my thesis.

I also want to thank my colleagues at Bell Labs for the 8 years that we worked on various aspects of telecom and multimedia systems. I also want to thank my colleagues at Ambient Systems for the 2 years that we worked to get the series 3000 wireless sensor networks working, tested and easily interfacing with third-party software systems. I also want to thank my colleagues at Inertia Technology for giving me some slack when I had a paper deadline. I also want to thank my fellow PhD students and other members of the Pervasive Systems group for not noticing my absence in the many PS-group meetings. I especially

would like to thank Nirvana Miratia, Berend Jan van der Zwaag, and Hylke van Dijk for publishing papers together.

I also want to thank the people in the Daidalos project who put up with me, first as architecture lead for the service provisioning workpackage, and later as workpackage leader. I thank Riccardo Pascotto, Rui Aguiar and Francisco Fontes for welcoming me in the technical management team and board of Daidalos. I thank the activity leaders Telma Motta, Susanna Sargento, Antonio Skarmeta and Andreas Kassler for tolerating my leadership style. It was a pity that I could not see the project to its end, since Alcatel-Lucent decided to remove Bell Labs from the Netherlands. I thank Andreas Kassler for continuing the workpackage lead in the last year.

I also want to thank my brother, sister, father and mother for understanding that I was socialising a bit less the last few years. I additionally had to decline numerous invites for playing Ruzzle and Wordfeut from my nieces and nephews.

Abstract

Today's inter-connected networks enable feature-rich applications that adapt according to situational and environmental changes, and the networks and objects that are discovered. These so-called pervasive applications can be composed from both locally and globally available multimedia resources such as audio and video, web services and context sources. Context sources in these pervasive applications can vary from your mobile phone's sensors to dedicated sensor networks deployed in buildings and vehicles, sensor nodes attached to beings and objects, and events generated from devices and applications.

The sharing of resources and the dynamic changes in these pervasive applications pose a number of challenges on the enabling infrastructure. This thesis focuses on methods for realtime sharing of wireless access networks, sensor information and multimedia among applications on mobile devices. We propose to use application-level techniques to support mobility and sharing and enable using lower-level techniques where available. Our contributions include methods and supporting communication architectures for efficient sharing of wireless access networks, multimedia and wireless sensor networks. Additionally, we contribute a reasoning framework to compare and combine communication architectures that support pervasive applications.

In short the main contributions of this thesis are the following:

1. **Sharing bandwidth in a wireless network**: We propose a bandwidth-distribution mechanism for broadband access technologies that uses realtime characteristics of the network medium and feed-forward control mechanisms.

2. **Mobility and sharing of wireless sensor networks**: We analyse the mobility and sharing of wireless sensor networks in logistic and person monitoring scenarios. We provide guidelines for dealing with mobile and overlapping wireless sensor networks and the most promising scheme for

sharing wireless sensor networks in applications. A middleware layer is designed and created to support real-time remote monitoring and maintenance of wireless sensor networks in logistic scenarios. Its middleware messaging efficiency is compared with web protocols and improvements are proposed.

3. **Mobility and sharing of multimedia**: We propose a seamless roaming experience in multimedia applications using SIP across heterogeneous networks. We use terminal intelligence to detect and select access networks from federated network operators. We compare MobileIP, SIP and their combination, and share the issues we encountered with our prototype. We propose a network initiated method to distribute a multimedia session over multiple devices in proximity to the user. Its applicability is verified with a prototype, and the combination with a terminal-initiated method is described. We propose to dynamically group multimedia streams from the same origin per network segment based on network characteristics and stream popularity, using relaying or multicast/broadcast when available.

4. **Pervasiveness in a competitive multi-operator environment**: We design, develop and validate a framework for next generation mobility-enabled networks offering seamless roaming with quality guarantees for multimedia sessions, broadcast integration, privacy and anonymity. We propose operator federations to enable personalized, context-aware, composite services to mobile users.

5. **Reuse of pervasive system architectures**: We propose a conceptual reasoning framework and use it to compare and integrate a number of pervasive system architectures. Additionally the required scalability, efficiency, pervasive and maintainability properties are compared and recommendations are given towards flexible pervasive system architectures.

Through these contributions, this thesis enables efficient realtime sharing of wireless access networks, mobile WSANs and multimedia streams. This work helps to pave the way towards pervasive applications that use dynamically changing networks, multimedia, web and context resources.

Samenvatting

De hedendaagse netwerken ondersteunen rijke applicaties die zich aanpassen aan situaties en veranderingen in de omgeving, en netwerken en objecten die ze tegenkomen. Deze zogeheten pervasive applicaties kunnen worden opgebouwd uit lokaal en globaal aanwezige multimedia services zoals audio en video, web services en bronnen met context informatie. Context bronnen in deze pervasive applicaties kunnen variëren van sensoren in de mobiele telefoon tot speciale sensor netwerken die zijn genstalleerd in gebouwen en voertuigen, sensoren aan objecten of personen, en events die zijn gegenereerd op apparaten en door applicaties. Het delen van bronnen en de dynamische veranderingen in deze pervasive applicaties vormen uitdagingen voor de ondersteunende infrastructuur. Dit proefschrift richt zich op methoden voor het realtime delen van draadloze netwerken, sensor informatie en multimedia tussen mobiele apparaten. We stellen technieken voor op de applicatie-laag voor mobiliteit en delen, die technieken van onderliggende lagen kunnen gebruiken indien aanwezig. Onze bijdrage hierbij zijn methoden en een ondersteunende communicatie architectuur voor het efficiënt delen van draadloze netwerken, multimedia stromen en draadloze sensor netwerken. Daarnaast stellen we een beredeneringsmodel voor om communicatie architecturen voor pervasive applicaties te vergelijken en te combineren.

In het kort zijn de belangrijkste bijdragen in dit proefschrift de volgende:

1. **Delen van bandbreedte in een draadloos netwerk**: We stellen een bandbreedte-verdelings-mechanisme voor in breedbandige toegangsnetwerken dat gebruik maakt van de realtime karakteristieken van het netwerk medium en controle mechanismen.

2. **Mobiliteit en delen van draadloze sensor netwerken**: We analyseren de mobiliteit en het delen van draadloze sensor netwerken in logistieke en persoons-monitoring scenario's. We geven richtlijnen voor het

omgaan met mobiele en overlappende draadloze sensor netwerken en de meest veelbelovende manier van delen van draadloze sensor netwerken in applicaties. Er is een middleware laag ontworpen en gemaakt voor realtime monitoring en onderhoud van draadloze sensor netwerken in logistieke scenario's. We vergelijken de efficiëntie van de berichtenuitwisseling in deze middleware met web protocollen en verbeteringen worden voorgesteld.

3. **Mobiliteit en delen van multimedia**: We stellen een naadloze roaming ervaring voor over heterogene netwerken met multimedia applicaties die SIP gebruiken. We gebruiken intelligentie in het apparaat om toegangsnetwerken van gefedereerde netwerkbeheerders te detecteren en te selecteren. We vergelijken MobileIP, SIP en hun combinatie, en delen de problemen die we ondervonden met ons prototype. We stellen een methode voor om vanuit het netwerk een multimedia sessie te distribueren naar apparaten in de omgeving van de gebruiker. Deze methode wordt geverifieerd middels een prototype en de combinatie met een gebruikers-genitieerde methode wordt beschreven. We stellen een methode voor om multimedia stromen van dezelfde origine dynamisch te groeperen per netwerk segment gebaseerd op netwerk karakteristieken en populariteit van de stromen.

4. **Pervasiveness met meerdere competitieve partijen**: We ontwerpen, ontwikkelen en valideren een infrastructuur voor de volgende generatie mobile netwerken die naadloze roaming ondersteund met kwaliteitsbehoud van multimedia sessies, integratie van broadcast, privacy en anonimiteit. We stellen federaties tussen netwerkbeheerders voor ter ondersteuning van gepersonaliseerde, contextgevoelige, samengestelde services voor mobiele gebruikers.

5. **Hergebruik van pervasive systemen**: We stellen een modelleringsvorm voor en gebruiken deze voor het vergelijken en integreren van een aantal pervasive systemen. Daarnaast vergelijken we de schaalbaarheid, efficiëntie, pervasiveness en onderhoudbaarheid en geven we aanbevelingen voor het maken van flexibele architecturen van pervasive systemen.

Middels deze bijdragen maakt dit proefschrift efficiënt realtime delen van draadloze netwerken, mobiele draadloze sensor netwerken en multimedia stromen mogelijk. Dit werk helpt de weg te banen naar pervasive applicaties die de dynamisch veranderende netwerken, multimedia, web en context bronnen gebruiken.

Contents

Chapter 1

Introduction

In the past decade we have seen a number of technology breakthroughs. Multimedia like audio and video moved from analogue to digital, enabling free audio/video calls over the Internet. Mobile devices got support for a number of network types and bandwidth is gradually increasing. Sensors became wireless and form wireless sensor networks for environmental monitoring, and multiple sensors are added to mobile devices, enabling various interaction modalities and situation awareness. The number of mobile devices and their capabilities are gradually increasing, and server clouds have been created that provide remote processing and storage.

As a result, today's inter-connected networks enable feature-rich applications that adapt according to situational and environmental changes, and the networks and objects that are discovered. Some currently available applications already adapt according to changes in the environment and the situation that the user is in. Examples are route-planners that adapt to traffic conditions along your route, games that react to movement of a mobile device, mobile devices that automatically connect to wireless networks that are in reach.

However, most of current adaptive applications are dedicated to a single task, and there is only limited sharing of information between applications of different vendors. Furthermore, the performance of applications is often determined by the server-side bandwidth and capacity, which is often consumed by distributing the same information to many mobile devices. At the same time, the mobile devices change network and can be temporarily without network. Efficient sharing of multimedia content is nowadays limited to that of dedicated content providers, and the content is not seamlessly continued when the network

is changed.

Future applications are envisioned to be adaptive to various changes in network, environment and situation. These so-called pervasive applications will be composed from both locally and globally available multimedia resources such as audio and video, web services and context sources. Context sources in these pervasive applications can vary from your mobile phone's sensors to dedicated sensor networks deployed in buildings and vehicles, sensor nodes attached to beings and objects, and events generated from devices and applications. Higher level context can be obtained by reasoning based on this sensor information and events. Example pervasive applications include:

- sharing your own realtime video or context information with your mobile friends seamlessly, also when they change to different networks;
- showing a realtime video to a dynamically changing group of users in similar environments or situations;
- automatically switching the realtime video from your mobile to a nearby wall-display and adjusting the surrounding lights based on the nature of that video or your preferences;
- automatically detecting low remaining shelf-life of stored or transported strawberries from environmental changes and transferring them to a nearby shop before they are spoiled.

In this thesis we will focus on methods and enabling communication architectures for realtime sharing of wireless access networks, sensor information and multimedia among applications running on mobile devices.

1.1 Problem statement

We want pervasive applications to perform efficiently using a scalable enabling infrastructure.

This is hampered by the rapidly increasing pervasiveness in today's networks, in terms of the number of mobile devices, the different networks they use during the day, and the amount of data they generate and share (near) realtime. This leads to numerous efficiency and scalability related challenges in a variety of application domains. We will analyse and propose solutions to a number of these challenges, and focus mainly on mobility and sharing of multimedia streams and Wireless Sensor and Actuator Networks (WSANs).

1.2 Challenges

This section describes research challenges for realtime mobile sharing of multimedia and context. We assume that a mobile device can have multiple network interfaces that can be be connected to different networks simultaneously. Normally applications just use the default network interface, but they can use specific network interface for each connection they make.

- **Mobility and sharing in heterogeneous networks**: For network mobility we distinguish changes in network attachment of devices (such as mobile phones) and mobile networks (such as a Wireless LAN (WLAN) in the train). A user would typically want to use the network or combination of networks that offers the best tradeoffs between cost, bandwidth, and latency properties. When network connectivity is lost, the user would like applications to start using another network without a hiccup. Moreover, when multiple users are using the same wireless network the user would not like his video stream to be interrupted by less time-critical traffic such as file downloads. The challenge is therefore to offer seamless mobility across heterogeneous networks and efficient sharing of wireless networks such as WLAN.

- **Mobility and sharing of WSANs**: When a WSAN in a truck or on a body is used by applications over the Internet, it can temporarily loose network connectivity and may have to change to other networks as it moves. These mobility changes have impact on the bandwidth and latency of the information coming from the WSAN, and on the reachability of the WSAN for remote configuration and actuation. Conflicts can arise when multiple applications try to send configuration and actuation commands to the WSAN. Another type of conflicts can arise when WSANs that use the same wireless resources move in each other's coverage area. The challenge is to support mobility and sharing of WSAN monitoring and control.

- **Mobility and sharing of multimedia sessions**: A multimedia session is usually a combination of session control and multimedia streams between endpoints. For multimedia session mobility we therefore distinguish between changes in network interface attachment of session control and multimedia stream endpoints. The latter enables splitting a multimedia session across multiple devices, e.g. move the video from your mobile to a nearby wall display and moving it back later. Multimedia streams can also be shared by multiple recipients when they are multicasted or

otherwise duplicated, the challenge is to do this efficiently with realtime data to a dynamically changing and mobile group of users.

- **Service platforms**: A service platform enables access to networks and services that are offered in heterogeneous networks. Federation between Service Platforms extends network and service usage to those of other service platforms. Challenges for service platforms are: (1) offering appropriate Quality of Service (QoS) and security while roaming, (2) sharing your identity across networks and applications and (3) enabling anonymous use of web and multimedia applications.

- **Pervasive service platforms**: Pervasive service platforms extend service platforms by supporting composition of tailored and context-aware services, multimedia streams and context into a pervasive application. The challenge for pervasive service platforms is to offer adaptability of the pervasive application to all sorts of changes such as environmental ones, the situation the user is in, available and attached networks, and available bandwidth in these networks.

In short, there are a number of challenges to enable seamless mobility and efficient sharing of mobile devices, wireless networks, WSANs, multimedia streams, context and services across heterogeneous networks.

1.3 Research question and hypothesis

In the light of the above challenges, this thesis focusses on enabling efficient sharing of networks, multimedia, and context across mobile endpoints. The main research question of this thesis is:

> *Analyse the tradeoffs in federated middleware for mobility and efficient sharing of networks, multimedia, and WSANs among applications on a multitude of mobile devices.*

1.3.1 Objectives and scope

Two main mobility types can be distinguished, namely (a) mobility across networks and (b) mobility across devices. The latter type of mobility is often referred to as *transfer*. Parts of multimedia sessions (i.e. session control and multimedia stream endpoints) can be switched to other networks available on

the same device (a) or transferred to other devices (b). Devices can change attachment to other networks (b). In the light of the identified challenges we focus on the transfer across devices of multimedia session parts (a), and on the mobility across networks of mobile devices and multimedia session parts (b). A number of different mobile devices can be distinguished, in the light of the identified challenges we focus on mobile devices that enable access to a WSAN and those that host applications using multimedia or WSANs.

For sharing of resources among applications we consider sharing of networks, multimedia session parts and WSANs. In the light of the identified challenges we focus on efficient sharing of: (a) a wireless network among applications on multiple devices, (b) multimedia stream endpoints among devices, and (c) a WSAN among applications over the Internet.

Regarding federated middleware we mainly focus on the scalability and efficiency of the interactions over TCP/IP.

1.3.2 Hypothesis

There are a number of trends for enabling networked services, namely: the Internet of Things, Web 2.0, peer-to-peer computing, and Cloud Computing. The Internet of Things suggests to put an IP stack in each device. Web 2.0 suggests to make all functionality available as a web service and to use application-level protocols for access and transport. Peer-to-peer computing suggest to distribute work between a number of nodes with similar capabilities. With cloud computing processing and storage can be accessed as a service, usually in the form of web services on virtualized servers. So, the trends seem to agree about using web services for everything. However, on the one hand peer-to-peer computing and the Internet of Things distribute functionality across devices, whereas cloud computing centralizes access to functionality offered by a group of virtualized servers.

Our hypothesis is that none of these technologies provide enough capabilities in their own right to solve our challenges, and we expect that acceptable performance for realtime mobile sharing can only be achieved by a combination of centralisation and decentralisation.

1.4 Contributions

The following paragraphs describe the contributions of this thesis in line with the challenges [35] described in Section 1.2.

1.4.1 Mobility and sharing in heterogeneous networks

A lot of progress has been made on seamless mobility across heterogeneous network in the IST-Daidalos project [17]. For sharing of wireless networks, we reported a traffic shaping solution for different QoS-classes and best-effort in [114], that shared knowledge on QoS queue-lengths. A more recent approach is that of IEEE 802.11e that uses a differentiated scheme with prioritized QoS classes including best-effort, video and audio. IEEE 802.11e also has non-mandatory extensions that can enforce the traffic constraints per terminal and QoS class. The contribution is summarized below.

- **QoS support in shared networks** [114], see Section 4.1: We propose a bandwidth-distribution mechanism for broadband access technologies that uses real-time characteristics of the network medium and feed-forward control mechanisms. A prototype has been developed on a wireless shared medium and employs legacy network elements that lack QoS capabilities.

Note that the bandwidth of WLAN networks gradually increased over the past years which diminished the need for QoS mechanisms. Besides, most users accept poor quality on free shared WLANs since alternatives like paid WLAN hotspots, fixed connections and telecom networks are available. However, people that have used WLAN in the train and other busy places for more than web browsing can probably agree that quality improvements are still welcome.

1.4.2 Mobility and sharing of WSANs

We presented a middleware solution in [36] that supports mobility and sharing for logistic processes. In [34] we analyzed mobility and shared usage of WSANs for different scenarios. The contributions are summarized below.

- **Efficient middleware for shared use of mobile sensor networks** [36], see Section 4.3: A middleware layer is designed and created to support real-time monitoring and remote maintenance of WSNs in logistic scenarios across the Internet via wired and mobile wireless network access technologies. This middleware offers easy integration with third-party applications for remote WSN monitoring and configuration. The messaging efficiency of this middleware is compared with those of well-known web protocols and recommendations are given to further increase the messaging efficiency.
- **Analysis of Mobility and Sharing of WSANs used by Applications** [34], see Section 3.3 and 4.3: We analyse the mobility and sharing

of sensor networks in logistic and person monitoring scenarios. The types of mobility in both scenarios are analysed, and the required degree of mobility support in each scenario is identified. Additionally, different degrees of support for shared usage of mobile WSANs by multiple applications are analysed. We provide guidelines for dealing with mobile and overlapping WSANs and the most promising scheme for shared use of mobile WSANs by applications.

Mobility and sharing of WSANs is still a very active area of research. Examples of working systems range from remote monitoring of fresh produce, to remote health monitoring and environmental monitoring.

1.4.3 Mobility and sharing of multimedia sessions

We analysed different seamless roaming options in [29]. We compared user-initiated approaches for mobility of multimedia streams, and a network-initiated method was reported in [13]. We described an approach [153] for efficient sharing of multimedia content to a dynamic user group. The contributions are summarized below.

- **Mobility management for multimedia sessions** [29], see Section 3.1: We propose a seamless roaming experience in multimedia applications using SIP across heterogeneous networks. We compare the suitability of different mobility solutions for maintaining multimedia sessions while roaming across GPRS, UMTS and WLAN. The advantages and disadvantages of each mobility management solution are described, as well as encountered implementation issues in our prototype.

- **Partial session mobility across devices** [13], see Section 3.2: Existing methods for distributing a multimedia session across devices are analysed and compared. None of them allowed initiating the distribution from within the network. We propose a network initiated method in a multimedia signalling node to distribute a multimedia session over multiple devices. This method can for instance be triggered by context changes such as availability of more capable devices in proximity to the user. A prototype has been created that implements this method and its applicability is verified. Additionally we describe how our method can be combined with a terminal-initiated method.

- **Efficient personalized content distribution** [153], see Section 4.2: We propose an flexible approach to blend general and targeted live content

streams in multiple multimedia sessions while enabling efficient distribution of streams per network segment. We propose to dynamically group multimedia streams from the same origin per network segment towards the destinations based on network characteristics and popularity of the stream in those segments, using relaying or multicast/broadcast when available. This change could be made dynamically by observing the number of users that use the same stream in a network segment from the multimedia signalling and the network characteristics and capabilities.

Note that seamless multimedia handovers are still hardly used outside the telecom operator domain, where it is handled transparently to the end-users. Seamless handover of multimedia streams is not available in the default client software. WLAN hotspots on trains often use seamless handover on the mobile router. Unfortunately, video and audio downloads appear to be blocked on the hotspots of Dutch trains. Movement of multimedia session parts to other devices has not taken off yet, apart from lacking operator and client support most audio and video devices are not yet multimedia session aware. Efficient sharing of multimedia is still limited nowadays to multicast in broadband access networks of content from specific providers.

1.4.4 Federated Service Platforms

We analysed mobility schemes supporting seamless mobility in the Freeband 4Gplus project [29] and IST-Daidalos project [15, 17]. The IST-Daidalos project also maintains QoS and security associations while roaming, sharing identities for network and service access, and anonymous use of services. The contributions are summarized below.

- **Mobility management in heterogeneous networks** [29], see Section 2.3: We propose usage of service platforms to create a service control layer that hides the heterogeneity of networks from end-users, 3^{rd}-party service providers, and access network providers. The proposed mobility management treads location and handover management on a per service and per session basis, respectively. This enables an application-centric mobility management approach in which applications can independently deal with mobility issues according to end-user preferences. Our proposed service platform offers mobility support on the Internet layer using Mobile IP and on the session layer using SIP and allows complementary use of both approaches. Intelligence for mobility management in the terminal is

proposed to detect network interface changes and select access networks based on application needs and user preferences.

- **Federated service platforms in next heterogeneous networks** [15, 17], see Section 5.1: We design, develop and validate a framework for next generation mobility-enabled networks. Envisioned scenarios include heterogeneous access networks, while requiring ubiquitous, services of adequate quality, broadcast integration, as well as the ability to support privacy and anonymity while making life easier for the end-user. Our main focus is on the service provisioning architecture, online charging and multimedia session management.

Note that service platforms by telecom operators did not take off as predicted years ago. Mobility appeared to be less of a problem for regular user actions (e.g. web browsing, playing games, watching videos, messaging, and calling). Most actions do not need to be mobility aware since they use small transactions that can easily be redone, and a broken video stream can easily be resumed. When a phone call degrades or drops, most people just try to call again. In the mean time service platforms arose on the Internet that offer services ranging from instant messaging to audio/video conferencing and social media. Some of these service platforms even offer their services to foreign users.

1.4.5 Pervasive Service Platforms

We proposed a pervasive service platform in [17, 13] that offers context awareness, and terminal, network and service adaptivity to all sorts of changes. We created a reasoning framework in [37] to enable comparing and combining pervasive communication architectures. The contributions are summarized below.

- **Pervasiveness in a multi-operator environment** [17, 13], see Section 5.1: We propose a communication infrastructure for next-generation networks that enables personalized, context-aware, composite services to mobile users. The basis for this infrastructure is federation between operators who create a pervasive environment for service provisioning, integrated support for mobility and security, virtual identities for users, and resource management. Our main focus is on the service provisioning architecture, multimedia session management and broadcast integration.
- **Reuse of pervasive system architectures** [37], see Section 5.2: We propose a conceptual reasoning framework for comparing and integrating pervasive system architectures and pervasive service platforms. This

framework enables decomposition of an architecture in its building blocks and makes its interactions explicit. Additionally the required scalability, efficiency, pervasive and maintainability properties of a number of architectures are compared and shortcomings are identified. We give recommendations towards flexible pervasive system architectures.

Examples of currently available pervasive applications are smart phone apps that use services from one provider and targeted ads. Context usage is still limited to location delivered by the phone, search queries and social interactions.

1.5 Structure of the thesis

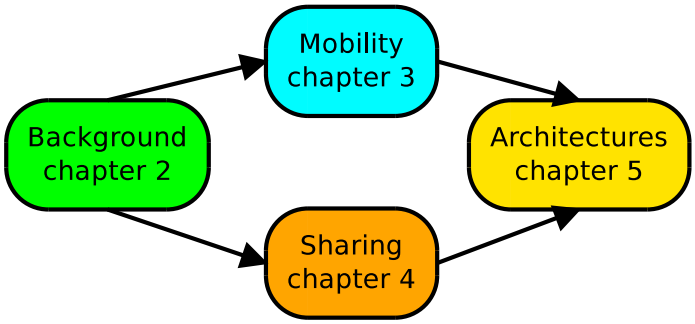


Figure 1.1: Thesis organisation

Figure 1.1 shows the organisation of the thesis. First we describe background and concepts related to mobility, sharing and service platforms. Chapter 2 describes methods for mobility and efficient sharing of wireless networks, multimedia and WSANs, it covers the contribution on mobility management in heterogeneous networks. Chapter 3 covers the following contributions: Mobility management for multimedia sessions, partial session mobility across devices, and the analysis of mobility of WSANs used by Applications. Sharing in Chapter 4 covers the following contributions: QoS support in shared networks, efficient personalized content distribution, and the analysis of shared use of WSANs by applications. Finally, in Chapter 5 supporting mobility and sharing are described, modelled, analysed, compared and conceptually integrated. Chapter 5 covers the following contributions: pervasiveness in a competitive multi-operator environment and the reuse of pervasive system architectures.

Chapter 2

Mobility, sharing and service platforms

This chapter details the background and concepts for supporting efficient realtime sharing and mobility of multimedia and context [35]. In Section 2.1, mobility and sharing in heterogeneous networks including sensor networks is described, and in Section 2.2 mobility and sharing of multimedia is described. Section 2.3 describes service platforms handling mobility and roaming, and Section 2.4 describes pervasive service platforms that enable composed services from web, context and multimedia services.

2.1 Heterogeneous networks

In this section we describe different networks that enable users access to the Internet (See Section 2.1.1), and Wireless Sensor and Actuator Networks (WSANs) (see Section 2.1.2) that gather information from the environment and allow actuation in that environment.

2.1.1 IP networks

In the heterogeneous networks of today, users have access to an increasing number of different access networks, both wireless and fixed. The combination of fixed and wireless networks enables end-users to be almost always on-line and connected to their preferred network(s).

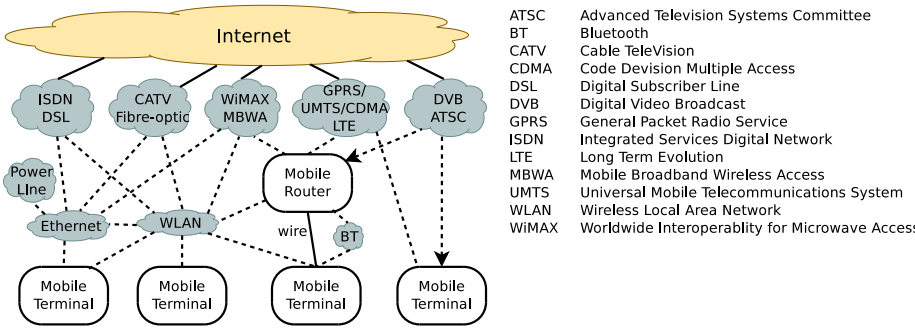


Figure 2.1: Different IP connectivity options

Heterogeneous networks consist of a variety of wireless and wired networks as core and access networks. End-user terminals and service providers are the end-points of these networks. Global IP-connectivity exists between all these networks and all end-to-end communication is IP- based. See Figure 2.1 for a number of IP connectivity options including unidirectional broadcast networks. Note that multiple access networks can be used simultaneously (multi-homing). With a mobile router, a mobile data connection can be shared with other devices over wireless technologies like Wireless LAN (WLAN) and Bluetooth or wired. When doing this with a mobile terminal this is often called tethering. This creates a potentially moving network, e.g. when used in a train.

2.1.1.1 Mobile Internet Protocol

Mobile IP [116, 77] (MIP) allows transparency of network changes and allows to maintain all TCP/IP connections while changing networks. MIP is mostly beneficial for connections with longer duration. A lot of tasks on mobile devices (such as web browsing and fetching/sending email) are not troubled so much by network changes since they are done rather quickly and can be easily be repeated when they happen to fail by a network change. Mainly longer sessions like Virtual Private Networks (VPNs), large up/downloads, and multimedia sessions need to be maintained while changing networks.

Mobile IPv4 [117] is the IETF standard for supporting mobility at the network layer in IPv4 networks. The terminal denoted as the Mobile Node (MN) gets a home IP-address assigned to be used for all communications. When the MN is not in its home domain, a so-called Home Agent (HA) forwards (tun-

nels) traffic to the MN's current location in a foreign network. In the foreign network, the MN obtains a Care-off-Address (CoA) from a Foreign Agent (FA) or a DHCP server, resulting in a FA-CoA (which is the address of the FA itself) or a co-located CoA, respectively. A co-located CoA has the advantage that an FA is not required in every visited network. Each time an MN changes its CoA it must re-register it with its HA in order to receive traffic directed to its home IP-address.

Mobile IPv6 [77][118] addresses a number of the Mobile IPv4 shortcomings such as the triangle routing problem. Route optimisation in Mobile IPv6 circumvents the triangle routing problem by sending binding updates, containing the current CoA of the MN, from the HA to all correspondent nodes.

Extensions have been proposed to Mobile IP to also handle moving networks with Network Mobility [48, 115] (NEMO). Examples of such moving networks are trains and planes that share their connection to a celular network like Universal Mobile Telecommunications System (UMTS) with the people they transport.

2.1.1.2 IP data flows

Communication in heterogeneous networks is a combination of data flows between applications. These data flows can be connection oriented with protocols like Transmission Control Protocol (TCP) and Stream Control Transmission Protocol [145] (SCTP) or connection-less with protocols like User Datagram Protocol (UDP). These flows can be protected from eavesdropping using security measures, and their quality can be maintained using Quality of Service (QoS) measures.

Security of IP data flows can be done at multiple layers of the TCP/IP model:

- at the network access layer by encrypting the packet payload
- at the Internet layer by using Internet Protocol Security [83] (IPsec)
- at the application layer by using protocols like Secure Socket Layer [63] (SSL) and Transport Layer Security [50] (TLS) for secured bidirectional connections, and Pretty Good Privacy [43] (PGP) for securing individual messages.

In order to provide QoS, packets of separate IP flows can be classified differently (e.g. as best-effort, audio and video), such that they can be treated

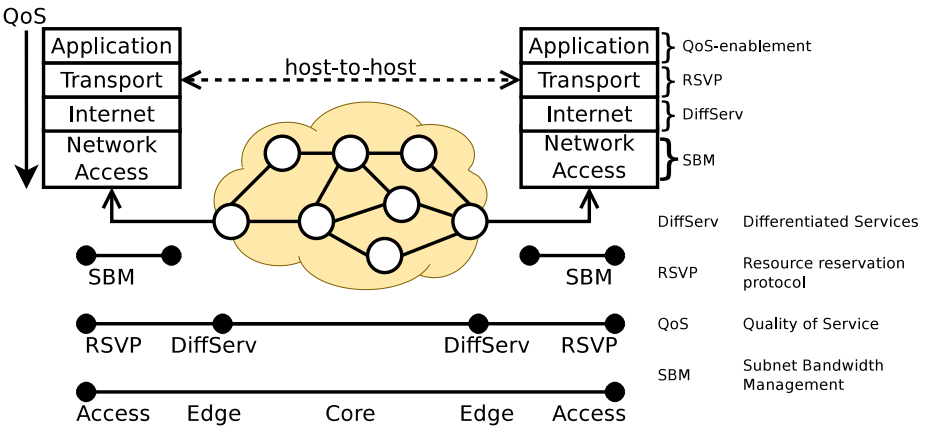


Figure 2.2: End-to-end QoS across access, edge and core domains

properly in the network. QoS treatment involves all network layers in every network element in the communication path, as illustrated in Figure 2.2. End-to-end QoS is determined by the lowest weakest link among all network elements between sender and receiver, and end-to-end QoS can be solved by dividing the problem along network domain boundaries [101], as illustrated in Figure 2.2.

2.1.1.3 Digital broadcast, multicast, unicast

The main difference between broadcast, multicast and unicast is that broadcast is destined to everyone that is able to listen. Multicast is for a selection of listeners, and unicast is directed to a specific listener. A distinction can be made between bidirectional broadcast in which the same (wireless) medium can be used to send something back and unidirectional broadcast that is only one way. Unidirectional broadcast can use a return channel on another access medium to send something back.

Traditional broadcast uses (radio) technologies to broadcast content to a large number of users, such as analog audio channels and Television (TV) via the air or via cable, and the last decade digital broadcast gradually became the new standard with mainly Digital Video Broadcast (DVB) and to a lesser extend Advanced Television Systems Committee (ATSC).

In fixed telephony, any of various Digital Subscriber Line technologies (xDSL) is used for multicasting television to the end-users and offering interactivity with

Internet Protocol television (IPTV). Broadcasting all channels is not really an option in xDSL since the last hop to the user is the dedicated twisted pair phone line which currently only supports high data rates over small distances.

In the mobile telephony standards UMTS and Code Division Multiple Access (CDMA), Multimedia Broadcast Multicast Service (MBMS) offers multicast and broadcast on handsets and via data cards (e.g. for laptop). MBMS is an enhancement feature of the UMTS architecture aiming at providing the capability for Broadcast and Multicast Services in the network (under Release 6). MBMS is divided into two parts: the User Services and the Bearer services. The User Services are provided by the Core Network to the mobile end user by means of the MBMS Bearer Services. The Bearer Services describe the operation of the radio link between the Radio Access Network (RAN) and the mobile terminal, and the capability to deliver IP multicast datagrams to multiple receivers using minimum network and radio resources. Since UMTS is not a broadcast-dedicated technology like DVB, the mobile terminal is supposed to be able to support simultaneous services (for example the user can originate or receive a telephone call) while receiving MBMS video content. This means that the mobile terminal resources must be shared with unicast traffic (same resources bandwidth in the cell), and so the MBMS radio bearer should be allocated only when needed. The reception at the mobile terminal must also start only when (and if) needed.

DVB is availabe in a number of types, including DVB-Satelite (DVB-S), DVB-Cable (DVB-C), DVB-Terrestrial (DVB-T), and Digital Video Broadcast - Handheld (DVB-H). All DVB data and digital data in ATSC is transmitted in Moving Picture Experts Group (MPEG) transport streams, which enables transmission and storage of audio, video, and data.

There are basically two types of multicast over IP, namely Source Specific Multicast [32] (SSM) and Any Source Multicast [46] (ASM). In ASM the user expresses its interest in a specific multicast group, in SSM, the user expresses interest in a combination of a specific source and multicast group. In both cases the routers between the source and destination need to make sure that the users that joined the multicast group get the associated IP streams efficiently (without unnecessary duplication).

2.1.2 Wireless Sensor and Actuator Networks

A WSAN typically consists of a large number of low-power sensor and actuator nodes. These nodes are equipped with a wireless transceiver, a small microcontroller, a power source and multi-type sensors such as temperature, humid-

ity, light, heat, pressure, sound, motion, etc. Additionally, the nodes can be equipped with actuators such as Light Emitting Diodes (LEDs), switches, and even motors. WSAN nodes are some of the smaller devices that collectively generate context information that can enhance pervasive applications. When these WSANs also have processing capabilities, they are also referred to as Pervasive Systems, i.e. systems containing a large number of collaborating tiny sensing, actuating, routing, and processing devices.

WSANs are commercially available in various forms, shapes, sizes, and functionality running various operating systems (e.g. TinyOS [42] or AmbientRT [68]). Interaction between sensor nodes and applications has not yet been standardized.

Applications involving WSANs are very diverse and involve one or a combination of various types of sensor networks. We can identify at least six types of wireless sensor networks, namely (based on [100]):

- **Environmental Sensor Network (ESN)**: These are the very first type of wireless sensor networks. Traditionally, ESNs were solely deployed for monitoring and data collection purposes. ESNs are often large scale, static, non-dense, and are deployed in harsh and unattended environments. Energy efficiency, long network life-time and security have always been the major concerns of ESNs.

- **Body Sensor Network (BSN)**: BSNs are sensor networks consisting of few wireless sensor nodes on or around a living being's body connected to a more powerful device such as a smart phone. Monitoring of vital signs, tracking, and data collection have been the main objectives of these sensor networks. Interaction with sensor-enabled objects [39], such as a dumbbell or ball, is an interesting upcoming usage area. BSNs are small scale, use different types of sensors and are usually limited to single-hop wireless communication. Since personal information can be collected by these networks, both security and privacy are major concerns.

- **Structure Sensor Network (SSN)**: SSNs consist of medium to large numbers of wireless nodes usually attached to or in buildings (e.g., office), structures (e.g., bridges), infrastructure (e.g., rails) or deployed in specific venues (industrial sites). Wireless nodes can also be attached to objects moving inside the structure and between structures. SSNs usually extend their wireless coverage with multiple hops of wireless communication and often use a variety of sensors.

- **Transport Sensor Network (TSN)**: Transportation means such as cars, trucks, and trains, have a number of sensors. Over the past few years, many efforts have been directed towards wireless communication and networking between transportation vehicles (e.g. vehicle to vehicle communication via IEEE 802.11p). Each individual vehicle can be seen as a sensor node, which locally observes its own state while it also monitors its surroundings.

- **Vehicle Sensor Network (VSN)**: The sensor data from within a moving vehicle (e.g. a car, boat, train, plane) can also be transferred wirelessly (e.g. via General Packet Radio Service (GPRS)) to a central server, and be monitored remotely and/or merged with data from other sensor networks. In warehouse logistics, VSNs are often used together with SSNs, e.g. when monitored goods are transported in a truck from one warehouse to the other.

- **Participatory Sensor Network (PSN)**: Mobile phones are becoming more and more equipped with sensors (e.g., Global Positioning System (GPS), accelerometer, gyroscope, camera) and different types of connectivity mediums (Bluetooth, wifi, Global System for Mobile Communication (GSM), etc.). This combination makes the mobile phone and in fact people carrying them a valuable source of collecting and transmitting information. Information collected by people through their mobile phones can range from personal health conditions and their trajectory to environmental conditions and pictures of the area in which they move around.

Mobility is typically covered within the WSAN, i.e. nodes within the WSAN can move around and use alternative nodes to stay connected. Mobility of nodes across WSANs and mobility of Internet-connected WSANs that are potentially used by multiple applications are still research topics.

The following application areas are considered [100], where WSANs are mobile and are potentially used by multiple applications:

- **Cool chain logistics**: In the cool chain market, it is important to optimise the quality of perishable products by ensuring optimal storage and transport conditions. In addition, assets can be tracked when they enter or leave certain areas.

- **Environmental/habitat monitoring**: Monitoring is done in the environment or the habitat of living beings, usually for extended periods where

user-intervention is either expensive or disturbing. Data mules are sometimes used to collect the sensor information when no wireless coverage is available. In habitat monitoring also the animals themselves can wear a sensor node.

- **Surveillance**: Building, vehicle and infrastructure monitoring to detect forcefully opened or unlocked doors/windows, theft and damage.
- **Smart spaces**: Smart spaces adapt to the needs of the users that enter and leave. They typically contain sensors and actuators that can be monitored and controlled by applications running in the environment and on user devices.
- **Remote eHealth**: In remote eHealth, sensor networks consist of few wireless sensor nodes on or around a living being's body. Typically, these nodes are integrated with a smart phone or a stationary device at home. Monitoring vital signs, and tracking are the main objectives of these sensor networks. Analysis is often done offline but increasingly becomes realtime.

Table 2.1 lists which WSAN types are typically used in each application area, and what items are mobile.

2.2 Multimedia sessions

Multimedia sessions are sessions that contain one or more multimedia streams. In this thesis we mainly focus on audio and video streams. Examples of multimedia sessions are Voice over IP (VoIP), audio/video teleconferencing, Video on Demand (VOD) and IPTV.

This section first describes protocols for multimedia session control, then mobility for multimedia sessions and then compares multimedia session mobility with mobileIP.

2.2.1 Session Control

For controlling realtime multimedia sessions over the Internet between two or more parties, a number of standards are available, namely:

- The Real Time Streaming Protocol [136] (RTSP) supports video-like control over a multimedia session with a streaming server. It can for instance

Table 2.1: Typical associations in specific application areas

Area / association	Cool chain logistics	Environment monitoring	Surveillance	Smart spaces	Remote eHealth
Mobile entity	truck, node	data mule, node	vehicles	user-device, object	user-device
Domains	depot, warehouse	geographic area	building, infrastructure	place	clinic
WSANs	areas, trucks	sub-areas	vehicles, areas, different types	different types	patients
Nodes	roll container	animal, object	door, window	object	object, user device
WSAN types	SSN, VSN	ESN	SSN, VSN	BSN, SSN, PSN	BSN
Apps	views, triggers	views, triggers	views, triggers	experiences	views, feedback

be used to establish, play, fast-forward, pause and stop a multimedia session containing multiple media flows;

- Revision 5 of HTML (HTML5) which is still under development supports playing audio and video files and is expected to support realtime multimedia playing in a web browser using RTSP.
- the Jingle [62] protocol extension to Extensible Messaging and Presence Protocol [132] (XMPP) enables signalling via an XMPP server for multimedia session setup;
- H.323 that uses telephony-style signalling from the International Telecommunications Union Telecommunications Sector (ITU-T);
- Session Initiation Protocol [31, 123] (SIP) using HyperText Transport Protocol (HTTP)-style signalling from the Internet Engineering Task Force (IETF);

Apart from those, closed approaches are available such as Skype and the flash player.

RTSP and HTML5 are mainly used for controlling unidirectional multimedia either from or to a streaming server and are not further considered. Jingle, H.323 and SIP do support setting up a multimedia session with multiple multimedia streams in any direction. Jingle does not support session mobility yet. There is one extension for session transfer [160] that has the deferred state. Jingle is designed to interwork with SIP. Because of the current lack of session mobility, Jingle is not further considered.

H.323 [75] is a standard published by the ITU-T for audio, video and data communication across IP networks. The H.323 Recommendation can be applied to voice-only handsets and full multimedia video-conferencing endpoints, and others. H.323 is part of the H.32X series for enabling video-conferencing across a range of networks including Integrated Services Digital Network (ISDN), Public Switched Telephone Network (PSTN) and IP networks.

H.323 does only provide seamless mobility while roaming when the network point of attachment does not change during handover (see recommendation H.510 from the ITU-T, e.g. when a mobility mechanism like MIP is in effect, or when all communication is tunnelled to the home provider network). H.323 is not further considered in the remainder of this Chapter.

SIP, as described in [130] and [129], is a signalling protocol used for establishing, maintaining, and terminating multimedia sessions and providing presence information in an IP network. Traditionally, resource discovery in SIP is done in a centralized manner, i.e. each domain has a local resource directory where all identities and their preferences are stored. SIP is adopted by the IP Multimedia Subsystem (IMS) of the 3rd Generation Partnership Project [8] (3GPP) [9]. Peer to Peer (P2P) SIP, offers a distributed mechanism for resource discovery which can reduce (or even eliminate) the need for centralized servers. In the remainder of this chapter only traditional SIP is considered unless specifically stated otherwise.

SIP can, in addition, provide user mobility functionality because the identification of users with SIP is independent of underlying IP addresses. Wedlund and Schulzrinne in [158] proposed to use mobility support in SIP to support real-time communication. Most current SIP user agents on mobile terminal do not support these methods.

2.2.2 Multimedia Session Mobility

SIP has its own mechanisms for mobility management [158] for SIP-based applications as well as functionality for session adaptation.

Application layer mobility solutions, for example based on SIP, can either replace or complement network-layer mobility [137].

No single approach to IP mobility applies across heterogeneous applications in heterogeneous networks [104]. To meet the requirements of applications and deal with harsh networking environments multi-layered mobility management solutions and architecture are proposed, see for example [52] and [121].

2.2.3 Multimedia Session mobility versus mobileIP

There have been a number of studies comparing SIP-based and MIP-based mobility management. The comparisons of the performance of the two protocols in [158] and [29] demonstrate that, in general, application-layer mobility management protocols, such as SIP, perform worse than lower-layer protocols in terms of hand-off delay, signalling overhead, and transparency. However, when suitability for deployment in next-generation networks is considered, it appears that SIP is a better mobility management solution for multimedia sessions, because it obviates the need for protocol stack and infrastructure changes [29]. A number of studies indicate that the suitability of a mobility management solution depends primarily on the type of application for which it is being considered. For long-lived TCP connections (such as FTP) and most standard Internet applications (such as Web browsing and chat), MIP offers a generic solution for roaming that seems to work well. However, for real-time applications, SIP is recommended [154, 158], because real-time applications (e.g., multimedia applications) have strict timing requirements that are not taken into account by MIP because it is a network-layer protocol. To optimize roaming behaviour, applications should be able to influence or even control the mobility management process, as they can when SIP is used as the mobility management solution. An additional benefit of using SIP for application-layer mobility management is that it allows applications to adapt their service behavior, based on the mobility management strategy selected, to provide the best possible end user experience.

2.3 Federated Service Platforms

Service platforms (see Figure 2.3) enable access to service providers to devices that are connected via heterogeneous networks. Federation between service platforms allows guest use of services, and realises a service control layer for that purpose. This layer enables third-party service providers to offer their services to roaming end-users, while being shielded from network-specific details.

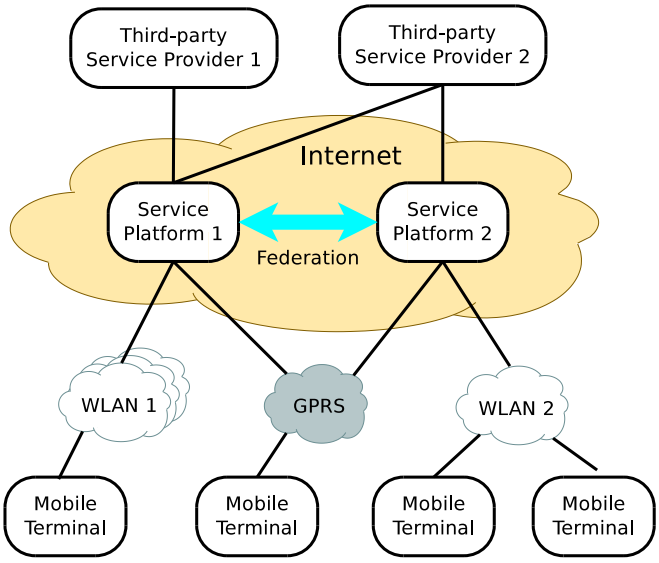


Figure 2.3: Federated service platforms enabling services to mobile terminals

In addition, end-users with a Service Platform subscription can use the services to which they are subscribed while switching access networks (including foreign ones). The Service Platform adds value for functionality such as mobility management, session control, authentication, user profiles, and user localization.

We first describe the functionalities of service platforms, then further detail mobility management.

2.3.1 Functionalities

There are a number of different functionalities that a Service Platform can offer, among others:

- Bridging legacy systems: E.g. a multimedia gateway can be used to bridge telephony between GSM/PSTN/ISDN networks and VoIP.

- Providing identity across federated service platforms: a user has a home service platform and can use its identity to use foreign access networks and services provided by other service platforms.

- Multimedia and messaging: E.g. SIP application server, to handle voice, video, and messaging applications.
- Charging: real-time and offline charging for services. Example charging types are time-based, volume-based, event-based.
- 3rd Party service interfaces: Example standards are the OSA/Parlay and their RESTful successors [21] for telecom-based service platforms.
- Seamless use of different Access Networks: i.e. providing mobility management, see Section 2.3.2.

2.3.2 Mobility management

To facilitate seamless continuation of services across access networks, users should be able to roam seamlessly from one access network to another and/or attach to multiple access networks simultaneously (multi-homing). Mobility management, which is the technical prerequisite for roaming behavior and service access, involves controlling the network(s) to which the user's terminal is connected and which service runs through which access network. I.e. mobility management discovers new access networks, and controls the handover between these networks [30]. It is also responsible for roaming services (e.g. continuous access to SMS and IP services). The services that can be supported on or across access networks depend on the characteristics (e.g., the bandwidth restrictions) of these networks; certain services may not be supported on certain networks. Therefore, it may be necessary to adapt ongoing service sessions to changes in the network environment. A typical example of such an adaptation is dropping video from an audio-video session for a low-bandwidth access network.

Mobility management plays a key role in dealing with user and terminal mobility in heterogeneous networks. Following [18] and [19], mobility management can be defined as a functional component that firstly keeps track of the IP-addresses of mobile end-users, and secondly modifies the IP routes of the ongoing sessions of mobile end-users[1]. The mobile end-users' IP-addresses can be tracked per session. This Mobility management function enables other end-users to initiate new sessions towards the mobile end-user. Similarly, modification of the IP routes of ongoing sessions can be done collectively (for all sessions) or individually (per each session). Modification of active sessions is subject to

[1] a session can be an instantiation of a service that is established between two or more end-points (i.e., users and/or machines). A more elaborate service concept is described in 2.4.1 and session concept is described in 2.4.2

the requirements of the sessions involved; examples of such requirements are minimal bandwidth and cost.

A mobility management system in heterogeneous networks as described above has the following characteristics:

- Mobility management concerns[2] both user mobility and terminal mobility aspects. Therefore, the end-user (and not just her/his terminal) is the one whose mobility is tracked and handled by the mobility management.

- An end-user is likely to be associated with multiple IP-addresses corresponding to the active network interfaces of her/his terminal(s).

- Due to diverse requirements of applications in heterogeneous networks, no single approach to IP mobility applies across these applications [104]. Therefore, the mobility management should provide multiple IP mobility solutions at different layers of the OSI model to handle mobility issues for services, individually or collectively.

- This asks for a multi-layered mobility management approach (i.e. mobility at different layers of OSI model) where the scope of the mobility management spreads from each individual service (and its sessions) to an aggregation of all services (and their sessions), associated with an end user. In other words:

  - The IP address to be tracked by the mobility management is the routable IP-address of the terminal interface, to which the end-user is attached for initiating a session of a particular service or any subset of services (this subset can include all her/his services).

  - The IP route to be modified by the mobility management corresponds to the terminal interface, via which the end-user is involved in an ongoing session or in any subset of ongoing sessions (this subset can include all her/his sessions).

- At any given time, the IP-address of an ongoing session of a service can differ from that for initiating a new session of the same service.

[2] A full solution involves the cooperation with other system functions like AAA, personalization, session control, etc..

2.4 Pervasive Service Platforms

A pervasive service platform (see Daidalos [17]) offers pervasiveness in addition to the Service Platform. Pervasive applications are applications that can be composed from existing services and be personalized and situation aware, by utilizing sensor information, context, profiles, and history of the user and the environment. Even when the user is not connected itself, the pervasive service platform and services can act on behalf of the user.

Pervasive Service Platforms are a distributed form of pervasive systems that provide a home base for the users, and give them a digital identity at that base. Federation of platforms allows the user to communicate with users at other bases and use services provided at other bases.

Other forms of pervasive systems can be organized as peering components (see e.g. Hydra [54]) that can discovered and hooked-up dynamically, for instance when they get close to one another.

In this section we describe the concepts for services, sessions, mobility and sharing in pervasive service platforms.

2.4.1 Service Concept

A *service* is defined as a (potentially distributed) software application that provides certain functionality accessible via well-defined communication protocols. The type of service is defined by the offered functionality and the supported access protocols. Service sessions are running implementations of the service's functionality and protocols. According to this definition the type of a service is defined by the communication protocols it supports and the functionalities it uses and offers. This definition allows us to include a wide field of services including data services (e.g. a currency translator or email) and usage/configuration of hardware devices (e.g. a display or a printer). A service session is a concrete implementation of a service type that is actually running. Services often follow a traditional publish/discover/subscribe paradigm, meaning they are registered on a server, can be discovered by querying this server, and once discovered a service can be accessed directly.

A *pervasive service* is a service that exposes its functionality and attributes in a standardized way, and is made available via specified service discovery protocols. The service can be integrated into a composite service. It may be security and privacy aware, context aware and allow for personalisation. Table 2.2 summarises the six requirements for a pervasive service.

The whole concept of the pervasive services is their adequacy as building

Characteristic	Requirement	Short description
Discoverable	Required	A pervasive service has to expose its functionality, the supported protocols and its attributes in a standardised way, independent from the particular service discovery protocol. Nevertheless, it has to support at least one (e.g. SLP).
Composable	Required	A pervasive service needs to be able to cooperate with other services.
Context-aware	Optional	A service may be context aware, i.e. adapt according to for instance situational, network or environmental changes.
Personalisable	Optional	A pervasive service may be aware of the user's personal preferences, i.e. it may have parameters that can be personalised.
Private and Secure	Optional	A pervasive service may specify privacy and security requirements when accessing sensitive user-related data.

Table 2.2: Pervasive Service requirements

blocks for more complex services, denoted as composite service: A set of cooperating pervasive services. A composite service may also be a pervasive service (recursive definition). A running composite service session is called a (composite service) session. Since service sessions that are being part of a composite service session, it may come and go frequently and necessary context information may change often, and therefore re-composition may need to be carried out regularly. During re-composition, service sessions might be added, reconfigured, removed or replaced by alternative ones. At a certain point in time the composite service session will be terminated, i.e. the composite service is stopped. During this process the individual services are disconnected and released. A composite service may also be called an application since the composite service provides the functionality to the end user. In order to create a composite service, knowledge is needed about how a composite service shall be created. This can be done both by the network provider, or by third parties.

2.4.2 Session Concept

When discussing data connections and streams it is necessary to describe the relation between the data packets, terminals, network nodes and services. This relation is usually known as a session, and may not always be easily identified (e.g. the set of packets under scope of a SIP application may not be identifiable by the traditional double set of IP addresses and ports). Another characteristic of a session is that it defines the relationship between a set of network nodes. For instance, a SIP session will involve a number of network nodes (SIP clients, SIP proxy) that are involved in the exchange of packets. The scope of what a session is varies with the aspect we are tackling. The following types of sessions are identified:

- **Network Access Session**: Session between a device (mobile terminal) and a wired or wireless network. In WLAN this usually involves knowing the encription key and sometimes authentication via IEEE 802.1X or a web page. In telecom networks this usually reguires a SIM card for authentication to the network.

- **Network Identity Session**: Includes all Network Access Sessions that use the same identity (or credentials) for authentication. This means e.g. having multiple virtual or physical interfaces on a mobile terminal authenticated using the same credentials.

- **Transport Session**: A transport session connects and/or exchanges data to and/or from a node in the network. Multiple transport sessions can exist within the same network access session, typical examples of transport sessions are TCP connections and UDP streams.

- **Application Session**: Contains (zero or more) transport sessions. Can exchange application-specific packets among distributed application parts. An important subtype of the application session is the multimedia session.

- **Pervasive Session**: A session that is directly mapped onto user goals and intentions. Can be context-aware and personalizable. Will control overall coordination of multiple application sessions that might interact with each other based on user context.

The following paragraphs further detail these session types.

Network Access Session The network access session starts with authentication of the user identity and checking authorisation for its access from a given MT on a given network interface to a specific access network. When authentication is not necessary, the network access session starts by connecting to the network. The network access session ends when the network access is terminated on that network interface, which could be the case when a re-authentication is necessary in another network (e.g. for inter-domain mobility) or when the user logs off the mobile terminal etc.. All other sessions have to be supported on top of these network access session(s).

In relation to the access technology, a network access session is always bidirectional, even when bidirectional technologies are being emulated using Unidirectional Link Routing (UDLR). Nevertheless, we can be have mostly unidirectional (e.g. authenticated + authorised broadcast access) or mostly bidirectional (e.g. wifi, UMTS, broadcast access + return channel, etc.) access. This does not preclude for a user with a DVB-enabled MT to access free information on the DVB broadcast, or, after registration, to keep on receiving protected content for a given time without any uplink (until registration times-out).

A network access session can have QoS guarantees as a whole, or for its contained transport sessions. Also both the whole session and the sub-sessions can have associated costs. The QoS guarantees and costs can differ when the network access session is handed over to another network, or to another MT. During a network access session, the Home Address (retrieved based on selected identity) of the interface will stay the same, and only the CoA will change after a handover.

Transport Session Within a network access session, several transport sessions can exist, the most important ones are TCP, UDP and SCTP. TCP is connection oriented, is bidirectional and provides ordered reliable transport. TCP sessions have a clear start and end. UDP provides messaging over IP, and these messages can be combined into streams between the same sender and receiver(s). The end of a UDP session is much harder to determine and an UDP session is usually unidirectional (or multi-directional in case of multicast traffic). SCTP is message-oriented and can also ensure ordered and reliable transport like TCP, on top of that it supports multiple data streams in parallel within the same SCTP session and can support transparent failover in multi-homing scenarios. SCTP sessions have a clear start and end like TCP.

While authentication and authorisation are never done on the granularity of a transport session, QoS control and mobility management are typically referring

to transport sessions. In case of a handover between different domains that requires re-authentication, a transport session will typically continue, while the related network access session changes.

Network Identity Session A network identity session contains all concurrent (i.e. overlapping in time) Network Access Sessions that use the same identity for authenticating to the network(s). The same identity can be used for connecting to different physical interfaces using different technologies, or to different logical interfaces on the same physical interface. As long as the Network Access Sessions use the same identity they are considered to be part of the same network identity Session.

Application Session Over these access sessions, applications are running their own sessions. One example of an application session is a multimedia session; other examples are broadcast services, context services, web services, and applications like Secure Shell [162] (SSH) and File Transfer Protocol [122] (FTP). SSH, FTP and web services are more traditional sessions, and easier to identify. The complexity of application sessions is reflected in complex services, even if not multimedia. Other non-multimedia sessions may cover services such as lookups for a restaurant, or cinema, access to traffic (jam) data, or download of audio/video content for offline consumption (e.g. buying songs on iTunes). Some of these services may actually relate multiple applications, and involve changes of many connections and/or connection end-points. For those application sets an application session is comprised of all data transactions which take place until a service transaction (e.g. buy and download an mp3) is finished.

Any kind of control traffic (like multimedia session setup, RTCP control messages, TCP acknowledgements) related to an application is part of the application session (typically consisting of several transport sessions), and that applications sessions may as a whole be subject to mobility and QoS management. The level at which this can be globally handled depends on the specific session characteristics.

Because of its relevance in this thesis, the multimedia session is further detailed below.

Multimedia Session A multimedia session (see also Section 2.2) can be established between two or more endpoints (users or service). The session usually starts after an invitation of a participant is accepted by another participant.

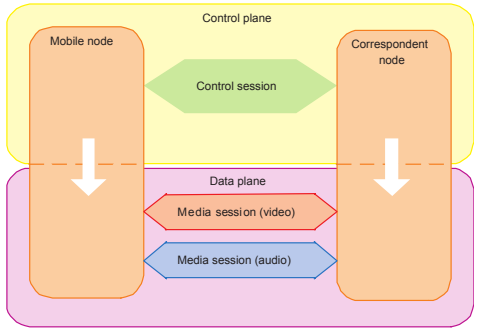


Figure 2.4: Multimedia session with control, audio and video sessions

A multimedia session can contain a number of multimedia streams and connections between the endpoints (usually audio, video and/or instant messaging), and session-specific messages (signalling) can be exchanged between the session participants. During the session a number of things can change:

- Multimedia streams/connections can be dropped, added or moved to other endpoints (partial session mobility)
- Quality of multimedia streams/connections can be changed
- Endpoints of the session signalling can change network location
- Endpoints of the multimedia streams/connections can change network location (e.g. handover or load-balancing when multi-homed)
- Endpoints may join or leave the session
- Session may be transferred to another endpoint.

A multimedia session usually contains a control session on the control plane that defines and controls media session endpoints on the data plane. Figure 2.4 shows how the control session on the control plane and the media session endpoints at the data plane are related to each other for a typical audio/video session between a mobile node and a correspondent node.

Pervasive Session A pervasive session, in its simplest form, can be considered a collection of application sessions, both multimedia and web service (or other)

sessions. A pervasive session will run a pervasive service, where the different parts of the service can be engaged in different application sessions (e.g. in a multimedia session and/or FTP session). Note that a Pervasive Session is contained inside an identity session. A pervasive session converges application sessions for a specific user experience, where the convergence is guided by the application logic needed to satisfy the user needs. So, while most connections and streams in a pervasive session could be setup by application sessions like the multimedia session, the pervasive session also contains the logic for context-aware starting and stopping of these sessions, adaptation of itself and its parts.

As an example consider a unified conferencing (UC) pervasive service (running in a network node) that is in charge of supporting virtual meetings between three persons in any possible way (depending on the communication technology that is available at any given context for the two people). If initially person A wants to use UC to meet person B there are different scenarios that can be supported by UC:

- Both A and B have mobile phones available to them and can be engaged in a phone conversation. In this case UC could set up a SIP session between A and B.

- While engaged in the SIP-based voice session, A and B might want to share a document. UC might initiate an FTP session or issue a SIP instant message to allow A send a file to B, and later to set up a data conferencing session where A and B can collaboratively browse through the document.

- If A or B's context changes in such a way that one of them gets access to a video camera (e.g. after having moved to another room while in the conference) UC might decide to add a video session to allow A or B see the other party.

- A third person C comes into the conference. C has access to only a chat program. UC might choose to add a speech-to-text conversion session so that C can participate in the conference without disturbing the conversation flow.

- UC might detect that C cannot receive documents on FTP and does not have any document sharing tool available. UC might decide in this case to print the document under discussion to the printer that is located close to C.

The above example shows many characteristics of a pervasive session. UC comprises a pervasive session that implements a multiparty conference among three people. This conference session can be regarded as an overlay application session that is continuously initiating other application sessions depending on the needs of the conference. For all these sub-sessions, the pervasive conference session keeps track of states of the different nodes, is informed about new nodes (e.g. a printer) becoming available, etc. An extended definition of a pervasive session might include application-initiated management of network access and transport sessions. It is easy to see the usefulness of such a concept. For instance, UC above could be extended to make active use of interface selection in order to set up and tear down network access sessions. As long as none of the participants in a conference have access to multimedia tools, UC might choose to use a low-bandwidth low-QoS network connection. Once users get access to multimedia, UC might choose to initiate network access and transport sessions to use better quality network available.

To summarize, a pervasive session has the following properties:

- It is a session that is often long-lived. I.e. it might live in the background and respond to stimuli from its surroundings (e.g. start an application session when there is a context change).
- It is heterogeneous, and will contain different types of sub-sessions. In its simplest form, these sub-sessions will be all application sessions. In its extended form, sub-sessions might also include network access and transport sessions. The pervasive session is the overlay session that manages the sub-sessions.
- It might include resources from different administrative domains. This means that setting up and tearing down sub-sessions might involve federation, authentication, etc.
- It will have several states. It might be running (actively using resources), suspended (not reacting to any stimuli), waiting (in the background, reacting to stimuli) etc.

2.4.2.1 Session relationship example

Figure 2.5 gives an example of intertwined session relationships. It shows that a *network access* session can exist for different access technologies and that all *network access* sessions for one identity are within the same *network identity*

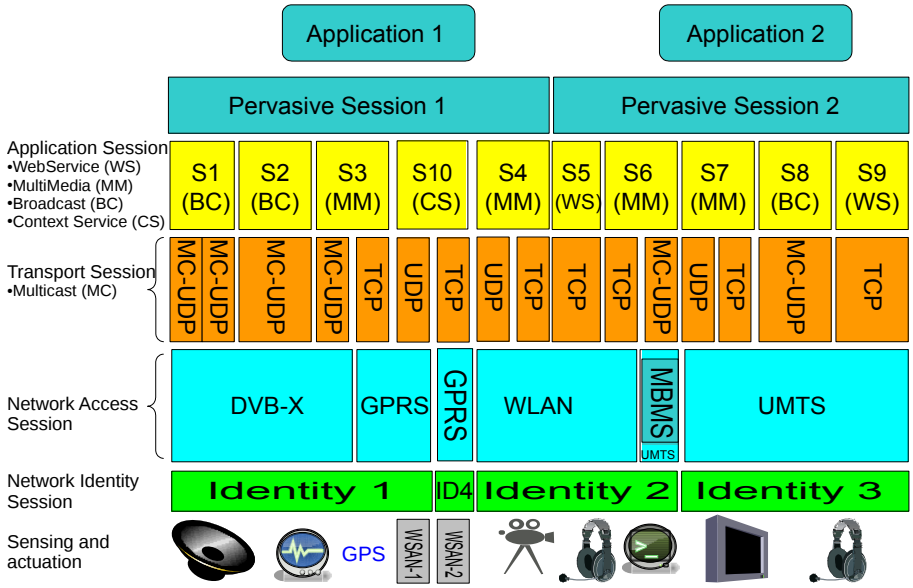


Figure 2.5: Example Session relationships

session. It also shows that an *application* session can potentially contain multiple *transport* sessions, e.g. a multimedia session (e.g. S3) can contain both a multicasted UDP stream and a TCP connection for signalling via different access technologies (DVB-X + GPRS), a broadcast session (e.g. S1) can contain multiple multicasted UDP streams, and a WebService session (e.g. S5) can contain one or more TCP connections. A context service (e.g. S10) can get context from *application* sessions, sensors on a user device and sensor networks connected via different identities (e.g. identity 1 and 2 over GPRS) and offer this context information or an inferenced version thereof for usage in all constituents of a pervasive session.

At a higher level, Application 1, could start Pervasive Session 1 which contains multiple Application sessions, namely S1, S2, S3, S4 and S10. Application 2 could start Pervasive Session 2, containing Application sessions S5, S6, S7, S8

and S9.

Regarding the relationships above, a set of information is needed in order to keep the overall complex view consistent at runtime. Some of the needed shared information is listed here:

- Identity-related information: The identity used to form a *network access* session will be used for associated *transport* sessions. Utilizing the service platform as an identity provider, this (network) identity could also be re-used by the *applications* and *pervasive* sessions within the *network access* session. However, the identity for those *application* sessions does not need to be shared.

- Preference outcomes: Preferences for using a Network Access Session might depend on the Application Session running on top of it. This information needs to be communicated.

- Context information: Context information might affect how lower-level sessions are configured. Moreover, access to different *network access* sessions might guarantee or deny access to different sets of context sources.

2.4.3 Mobility

A distinction is made between the following types of mobility:

- **user mobility**, a user can access the network from multiple devices, i.e. the user actually is able to connect and act in a seamless way from all mobile terminals.

- **device mobility**, a device can change its attachment point to the network, i.e. it handles mobility of network access sessions between different access networks. When re-authentication is necessary, the change from one network access session to another would also be considered terminal mobility (but may not be considered seamless and may break running sessions).

- **interface mobility**, a session can be moved from one interface to another in the same device.

- **service mobility**, the provider of a service can be moved during the provisioning of that service

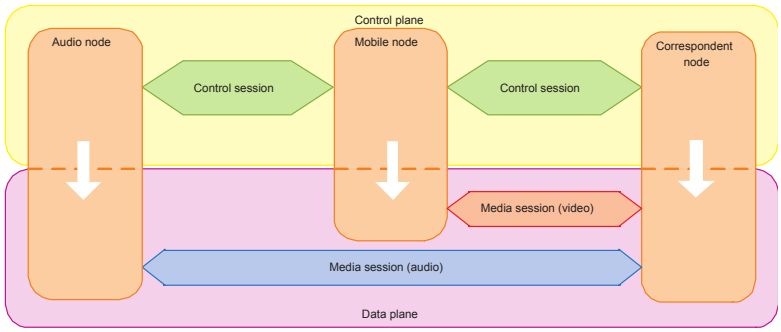


Figure 2.6: Audio stream endpoint moved from Mobile Node to Audio Node

- **session mobility**, a session or its parts can be transferred between devices.
- **WSAN node mobility**, a node moves within the WSAN or between WSANs.
- **WSAN mobility**, the WSAN may move and therefore change its network point of attachment (device mobility) or change to another network interface (interface mobility).

Different abstraction layers can be considered, both on the network side and the terminal side to abstract technology specific issues, enabling both local and remote communication for enhanced handover procedures. Handover can be initiated either by the mobile node (mobile initiated handover) or by the network (network initiated handover). More advanced concepts such as network aided mobile initiated handover can also be considered.

Session mobility can be related to mobility of: network access sessions, transport sessions, application sessions and pervasive sessions with associated transport sessions (streams and network connections). Since application and pervasive sessions can be composed of sub-sessions and services that use multiple network access sessions, session mobility can have many aspects, and may cover several of the above mentioned mobility types:

- **Network access session mobility**: the network access session is changed from one network interface to another, or is moved to another device. This would require support for multi-homing from the network access provider.

- **Transport session mobility**: a transport session is moved from one interface to the other or between devices.
- **Partial session mobility**: either signalling/control or contained connections/streams move
  - Multi-homing: part of a session moves from one to another network interface
  - Multi-device: part of a session moves from one device to another device (see Figure 2.6)
- **Full session mobility**: the whole session moves
  - Multi-homing: The whole session, including signalling, is moved from one interface to another.
  - Multi-device: The whole session, including signalling, moves to another device.
- **Service session mobility**
  - Multi-homed service session: Part of a composed service session moves from one to another network interface of a device (could be service session on 3rd-party server or on terminal).
  - Part of a composed service moves from one device to another device (could be 3rd-party service or service on terminal)
  - When moving to another domain a candidate service can be instantiated in that domain when the user, personalisation, or context indicates a that service to be similar enough. Such a replacement of service instantiations is also called re-composition.

2.4.4 Sharing of content and context

Content and context (including sensed information) can be shared from (mobile) sources to multiple (mobile) destinations. Mobility here means that sources, destinations and intermediate nodes can move and be temporarily unavailable. Movements of source and destination may also happen simultaneously.

Realtime content and context have a notion of freshness and priority. In a lot of situations, older data is no longer relevant after a temporal outage or limited available bandwidth, and can be discarded, such as with video broadcast. In

other cases, such as cool-chain logistics, the history of context data needs to be recorded but can arrive later.

When the number of destinations increases, unicasting to all destinations will consume more bandwidth and processing power. To overcome this bottleneck, one data stream can be sent towards a group of destinations and be divided there (e.g. multicast). Also for checking who is allowed to get the data, the source may not be able to handle all requests when the number of destinations increases.

The sharing can also be influenced by the destinations, not all destination may require the same rate or selection of information. Therefore, remote configuration is required, in a controlled manner. The configuration of one destination, should not affect the experience of another destination. For instance when a destination requires a context update every 5 minutes, the others can still get it at the default 15 minutes and the source could sent it more frequently towards the first destination. Something similar holds for actuation of the source, destinations can potentially sent conflicting actuation commands, so control is required to determine who has authority and priority to make these changes.

2.5 Conclusion

In this chapter we have described the main concepts involved in realtime mobile sharing, namely networks, sessions, services, mobility, federated service platforms, sharing and pervasiveness. We have also observed the dynamics of network attachment, multimedia sessions and context and how they can trigger and enable adaptations of pervasive applications. The bottom line is that sharing and mobility are intertwined, and the performance of realtime sharing and mobility handling impacts the efficiency and scalability of pervasive applications. Therefore the next step in line with our research question is to look into more detail at mobility, sharing and supporting infrastructures. In Chapter 3 we analyse how mobility can best be supported for devices, multimedia streams and WSANs. In Chapter 4 we analyse how wireless networks, multimedia and WSAN context can efficiently be shared among applications running on multiple devices.

Chapter 3

Mobility

This chapter describes how mobility can be supported for multimedia sessions and sensor networks across heterogeneous networks.

In 2004 [29], see Section 3.1, we proposed MobileIP for long-lived legacy services, and compared the use of MobileIP and SIP for multimedia services, according to a developed prototype. Furthermore, advantages and disadvantages of both mobility approaches are discussed.

In 2006 [13], see Section 3.2, we proposed a network-initiated method to switch the endpoint of a stream in a multimedia session dynamically to a different device, and compared this method with existing user-initiated approaches.

In 2011 [34], see Section 3.3, we analysed the mobility of Internet-enabled Wireless Sensor and Actuator Networks (WSANs) used by applications, and the mobility of WSAN nodes across WSANs.

3.1 Session Mobility

For session mobility (see 2.2.2) a distinction is made between generic Mobility Management (MM) for legacy services (like web browsing, Secure Shell [162] (SSH), etc.) using MobileIP and Mobility Management for multimedia services (like VoIP or streaming video) using SIP. End-users are assumed to use terminals with functionality for selecting different types of access networks. For multimedia sessions, a distinction is made between changes in network access while in a session or not in a session. If the end-user is not in a session, he or she needs to be able to start new sessions and receive invitations for new sessions. When a

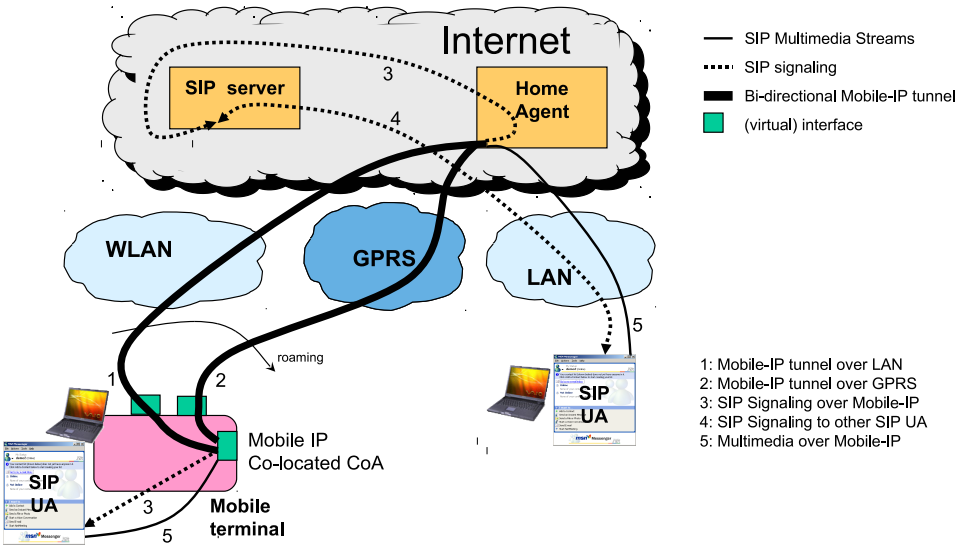


Figure 3.1: Roaming with MobileIP

better access network becomes available while the end-user is in a session, the session needs to be handed over to the new access network as seamless as possible from the perspective of the end-user. An integrated but flexible solution is proposed that facilitates MM with a customizable transparency to applications and end-users. The actual experienced "seamlessness" highly depends on the bandwidth of the networks, the performance of the implementation, and, last but not least, the application error and delay handling capabilities.

3.1.1 Mobility management with MobileIP

Figure 3.1 shows how the user of the terminal on the left, with network interfaces to LAN, Wireless LAN and GPRS, can setup a multimedia session with the other terminal. Figure 3.1 shows two alternative bi-directional MobileIP tunnels:

1. The bi-directional MobileIP tunnel over Wireless LAN between terminal and the Home agent

2. The bi-directional MobileIP tunnel over GPRS between the terminal and the Home Agent

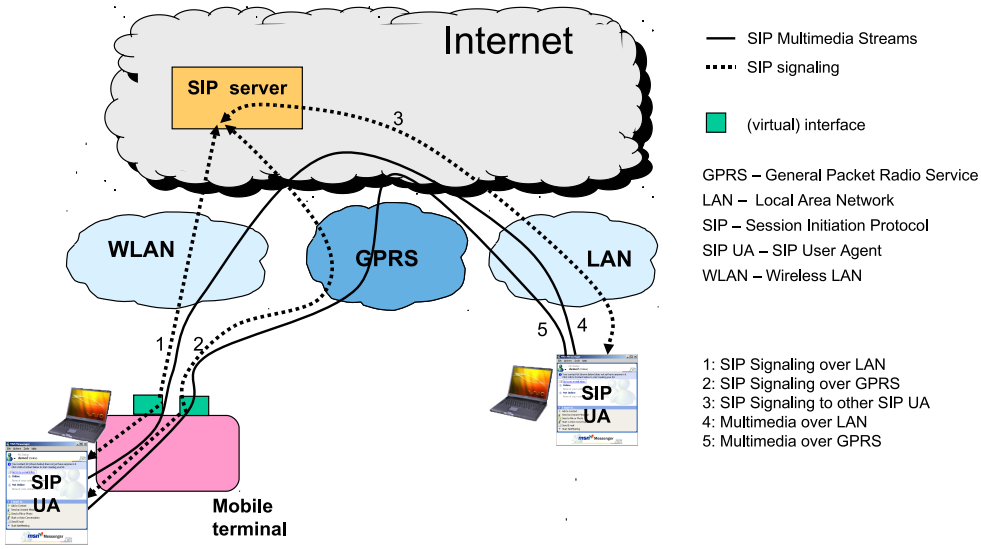


Figure 3.2: Roaming with SIP

When only one of the access networks is available or one is explicitly chosen for Mobile-IP, the respective tunnel through that access network is used for the MobileIP traffic.

Since MobileIP operates in the network layer of the TCP/IP protocol stack and handles mobility of IP addresses for multiple access networks transparently for applications. Mobile IP cannot hide the network characteristics, which means that the behavior of the applications can be unpredictable when the required network characteristics (such as amount of bandwidth) cannot be met in a network that is switched to. Since a co-located Care-off-Address (CoA) is used in the terminal, no foreign agents are required in the different networks.

3.1.2 Mobility management with SIP

Figure 3.2 shows how the user of the terminal can setup a multimedia session with the other terminal in the situation where Mobile IP is not used. SIP offers applications the possibility to autonomously trigger the handover, and, in addition, allows applications to change the session characteristics (e.g. codec change or remove media).

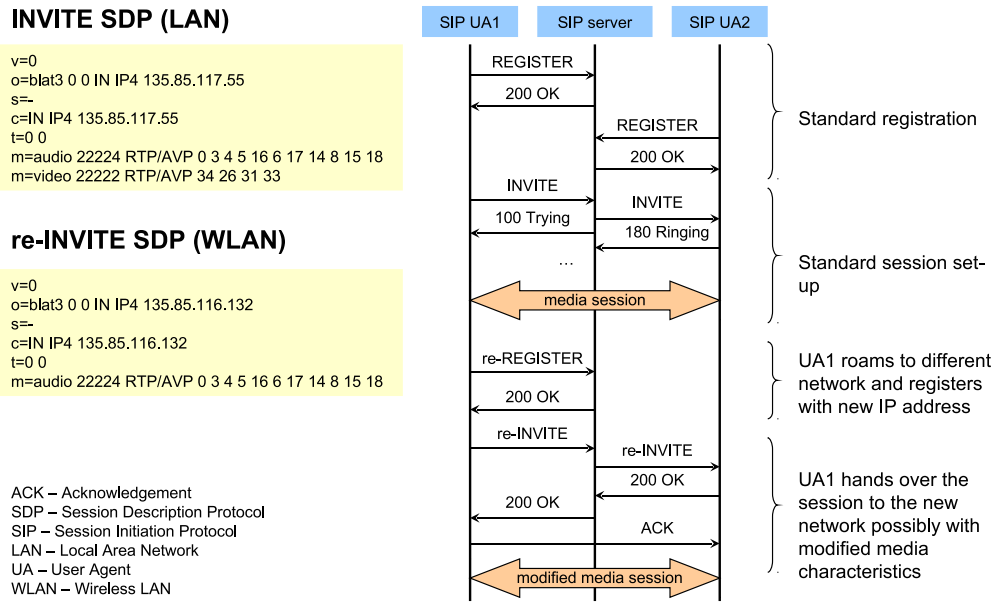


Figure 3.3: Roaming with SIP

SIP clients can use the so-called re-REGISTER and re-INVITE messages for mobility management. In fact, the re-REGISTER message can contain a new IP address at which a SIP client is reachable after roaming to a different network. The re-INVITE message can also be used to adapt an existing SIP session, e.g. add video to or remove video from an existing session, for instance after roaming to a different access network. This is illustrated in Figure 3.3.

First, a media session is established between UA1 and UA2 in the standard way. Then, the terminal of UA1 roams to a different access network from which it obtains a new IP address. As a result, UA1 registers itself again with the new IP address by sending a re-REGISTER message. Next, UA1 modifies the existing session by sending a re-INVITE message. As an example, this allows a session with both audio and video on a WLAN network to be changed to a session with audio only over a GPRS network. Note that on the final ACK of the re-INVITE, the original media session is adapted according to the new agreed session description parameters in accordance with RFC3261 [130]. As a result, for a short period, the non-roaming terminal may transmit data to a non-reachable destination.

A more serious problem is that, when the roaming terminal moves to a network with much less available bandwidth, the original media streams can result in flooding on the new network connection, preventing SIP signalling messages to reach the SIP server. The recommended solution to this problem is to reduce the bandwidth with a reINVITE before the switch to the new network, combined with Quality of Service (QoS) measures to give signalling traffic preference over media traffic (e.g. with traffic shaping and/or better QoS class for signalling traffic). It may be clear from the above that an intelligent entity is needed at the terminal of the roaming user that is informed of any changes in the network environment of the terminal and that can decide whether or not to trigger the re-REGISTER and/or re-INVITE messages. Section 3.1.3 describes a complete SIP client architecture that contains such an entity.

3.1.3 SIP client with Mobility Management

Our Session Initiation Protocol [31, 123] (SIP) client architecture [124] to support mobility in a heterogeneous network environment is shown in Figure 3.4. The core of the SIP client is a standard non-mobility aware SIP client [109] using a standard SIP stack. To support mobility, we have complemented the client with functionality to receive information about network changes and the capability to create Session Description Protocol [127] (SDP) values that match the network characteristics. We have also added the re-invite and re-register

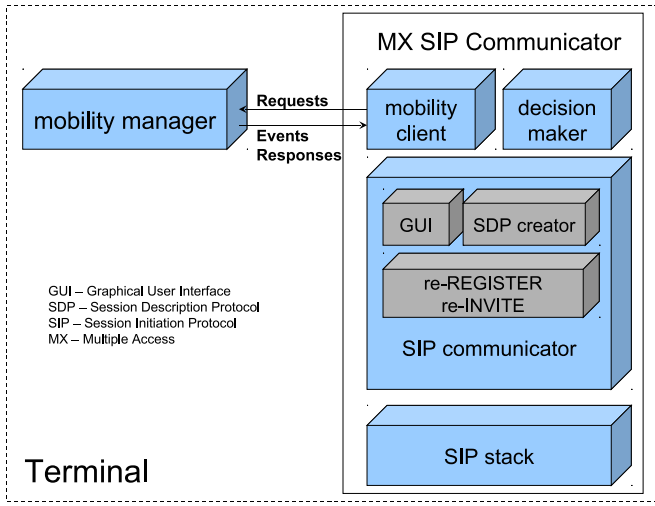


Figure 3.4: SIP client architecture

functionality to be able to adapt the existing sessions to a new network. A so-called decision maker is in control of the entire mobility management within the SIP client.

3.1.3.1 Decision maker

The responsibility of the decision maker is to make mobility management decisions within the application while taking into account the information available on the active networks and their characteristics, such as bandwidth and cost of usage. After the external mobility manager initiates a handover to a new access network or detects changes in the network parameters, the decision maker will trigger the SIP application to update the SIP sessions parameters according to the new network properties. The decision maker obtains its information from the mobility client component. The quality of the information provided by the mobility manager and the policy functions inside the decision maker determines the quality of the decisions that can be made. Therefore we have defined a number of network attributes to enhance the roaming behavior. Currently, two distinct cases of mobility are identified, one where Mobile IP is active and one where it is not. When Mobile IP is active it will take care of the terminal mobility so that network changes do not affect the IP address of the application.

However, it could affect the network parameters, such as the type of network used by Mobile IP. Since SIP is needed to provide the session control in order to adapt the session to new network parameter, the decision maker has to issue re-INVITE messages for all active sessions in the application that are affected by the changed network parameters.

When Mobile IP is not active, SIP is used to perform both the terminal mobility and the session control functions, and the decision maker must take into account that after a network change also the IP address should be changed. In this case the decision maker has to re-REGISTER the application at the SIP server with the new IP address. Then, re-INVITEs have to be sent for all active sessions. Figure 3.5 gives a simplified state diagram of the decision maker in the SIP application after a network change event has been received. The decision maker takes all decisions in the diagram, the input signals are the triggers from the external mobility manager, and the tasks are operations in the SIP client. It is also possible to have the decision maker separate the data and control paths if there are multiple active network interfaces available. For example, control data could be sent over a General Packet Radio Service (GPRS) network and the multimedia data over Wireless LAN (WLAN).

3.1.3.2 SDP creator

When a non-mobile terminal is connected to one network, the SDP can stay the same over time (determined by the applications requirements, terminal capabilities, network capabilities and user preference). However, when a terminal roams between different networks, the SDP has to be adapted according to for instance change of bandwidth and delay. The SDP creator is able to create an SDP profile based on user preferences, terminal characteristics and the network type it is connected to. With this SDP scheme, the application can adjust the amount of sent and received data, for instance dropping the video part on a low bandwidth network. Examples of SDPs are given in Figure 3.3.

3.1.3.3 re-INVITE / re-REGISTER

Most applications only implement the registration and session initiation part of SIP and do not support mobility. The re-INVITE message has the same format as the standard INVITE requests but can have different parameters and is used during a session. Also the response to a re-INVITE is different than for an INVITE since there is already a session and the purpose of the re-INVITE is to adapt the characteristics of that session. The re-REGISTER message is

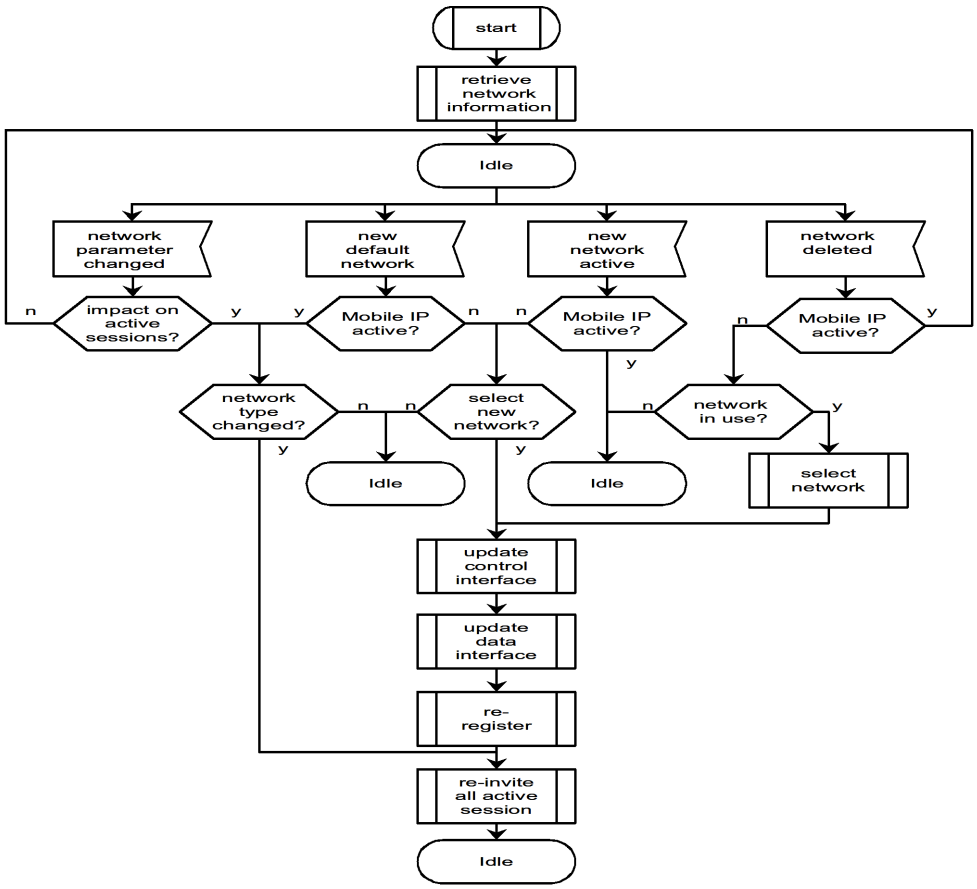


Figure 3.5: State diagram for roaming decisions

the same as the standard REGISTER. This REGISTER supports already the retransmission mechanism as part of the registration expiration process. For the re-REGISTER we do not wait until the timer expires but as soon as the mobility manager switches to a new network a re-REGISTER is sent containing the new network address.

3.1.3.4 Mobility manager

The mobility manager is external to the SIP application but running on the same terminal. It manages all network changes, such as network detection (new networks available, networks disappearing), automatic network authentication and maintains the routing table. The mobility manager assures that users are always best connected according to their preferences. The mobility manager keeps up-to-date with all available networks and their characteristics and can switch between them, for instance when the terminal goes out of reach of one network and comes within reach of another. Although the mobility manager is capable of maintaining network connectivity it has no application level knowledge and therefore is not aware of the impact on existing sessions with a network change. This is the domain of SIP and the decision maker inside the SIP application.

3.1.3.5 Mobility client

The mobility client communicates with the mobility manager to exchange network status information. This information is formatted in XML. There are two types of messages: event type messages that are sent towards the SIP client and request or command type messages originating from the client. To support mobility in an application and to keep an application up-to-date on the status of the available networks the following messages and events are identified (see Table 3.1).

Figure 3.6 illustrates the flow of messages that occurs when a new network is detected. This causes the decision maker to decide to switch over to that network. This is the most complex case where both the IP address of the data channel and the control channel have to change and subsequent the multimedia streams have to be rerouted to the new interface.

Name	Type	Purpose
Register for event [1]	Request	Express interest for a certain type of event from the mobility manager
Set network default	Request	Initiate selection of another network as the default network
Get info [1]	Request	Retrieve detailed network information from the mobility manager
Network available	Event	Indication that a new network has been found. The network may be inactive and not connected yet
Network active [1]	Event	Indication that a network has been connected (LAN, dialup) or active (WLAN)
Network unavailable [1]	Event	Indication that a network has been deleted
Network parameter changed [1]	Event	Indication that a network parameter has changed, i.e., status, bandwidth (WLAN), cost of usage
New default network selected [1]	Event	Indication that a new default network has been selected
Network connectivity lost	Event	Indication that no network is available

Table 3.1: Messages between mobility manager and application

[1] As this overview presents a generic interface for all kinds of applications, not all requests and events are used by all applications. For example, the described SIP client prototype only needs the requests and events marked with this footnote.

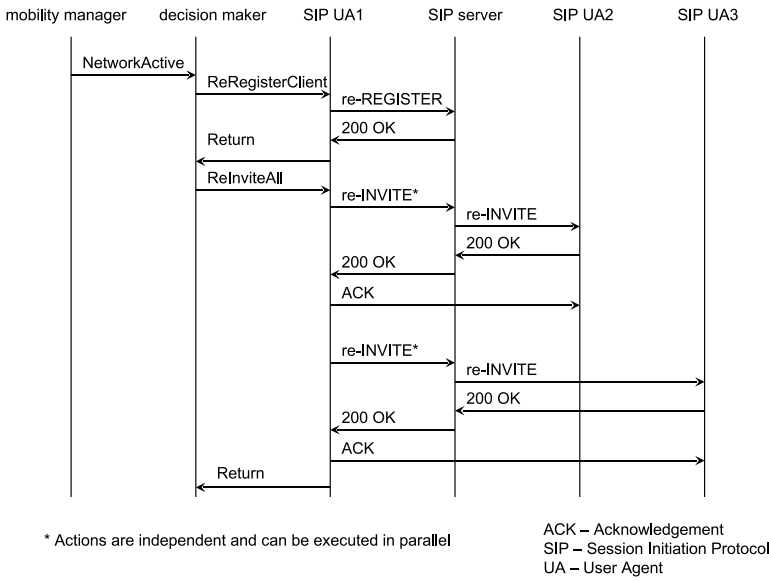


Figure 3.6: Message sequence diagram for SIP roaming

3.1.4 Prototype implementation

The architecture as described before has been implemented in a prototype. This prototype can demonstrate a seamless roaming solution across heterogeneous wireless and mobile network technologies and adapts sessions based on changing network characteristics. The prototype implementation extends the Java®-based SIP Communicator, a public domain SIP client application based on a public domain JAIN®SIP stack from NIST [108]. It runs on, among others, Microsoft Windows®and Linux®and uses the Java Media Framework [146] (JMF) for the media streams. The JAIN APIs bring service portability, convergence, and secure network access to telephony and data networks and allows for rapid prototyping. We have extended the application with re-INVITE and re-REGISTER functionality and combined it with a mobility manager. The result is called the MX SIP Communicator.

The original SIP communicator already had a mechanism to perform the register and re-registration after the expiration of a register. This code was re-used to implement the re-REGISTER. Re-registration is performed to communicate address changes to the SIP server. After a change in the network address we initializes the SIP stack with the new properties. For the implementation of the re-INVITE we used the standard INITIVE functionality and added support to use the information of the existing call and to provide storage for the data that needs to be changed during a re-INVITE. The decision maker was written from scratch and implements the state diagram from Figure 3.5. The decision maker caches the network information it retrieves from the mobility manager. It uses a simple policy to decide when a network change will result in a re-REGISTER or re-INVITE. The decision maker is implemented in a modular way so that it can be enhanced with new functionality when needed. The SDP creator constructs SDPs depending on values in a configuration file and the network type that is received from the decision maker. Depending on the value of the SDP the MX SIP Communicator can add audio and or video to a session. The mobility manager prototype supports the described XML interface so that it can provide the information needed to implement the prototype.

3.1.5 Roaming with MIP and SIP

Figure 3.7 shows the experimental set-up in which the functionality of the SIP client is tested and the performance is measured. This section describes several roaming scenarios for SIP sessions with or without MobileIP.

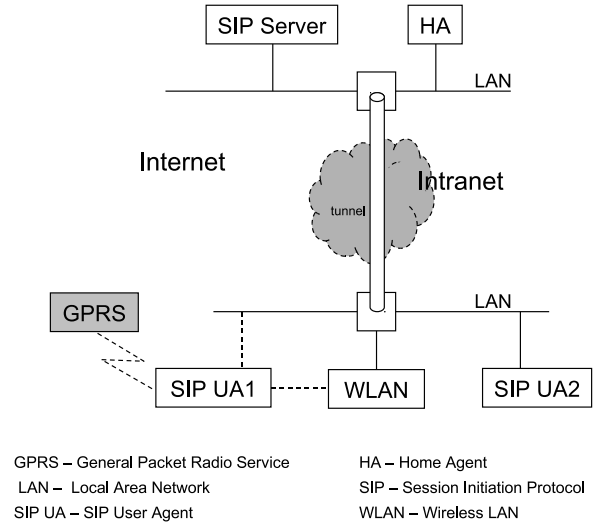


Figure 3.7: Experimental test set-up for SIP roaming

3.1.5.1 MobileIP for Interface- and SIP for Session-Changes

In this scenario UA1 (on a WLAN) and UA2 (on a Local Area Network (LAN)) start a media session; then UA1 moves from a WSAN to a GPRS. The session is handed over and the characteristics of the media stream are changed. In this scenario, Mobile IP [116, 77] (MIP) handles terminal mobility and SIP handles session control. MIP is active on the terminal of UA1 and it has obtained an IP address from the Home Agent (HA). Both WLAN and GPRS networks are available, and WLAN is selected. The scenario starts after both UAs are registered on the SIP server and an active audio session with an expensive (i.e., high-bandwidth and low-latency) codec has been established. The first step in the scenario is the detection of a degraded WLAN signal by the mobility manager. The mobility manager selects GPRS as the new default network and sends a Network parameter changed event to the decision maker. The decision maker knows that a re-REGISTER message is not needed because MIP is active, so it triggers only a re-INVITE message for all active sessions. As a result, the session is handed over with an interruption, not because of an IP stack change, but because different codecs are used and with different media characteristics (i.e., audio only).

3.1.5.2 SIP for Interface and Session Changes

The main difference between this scenario and the previous one is that, in this scenario, MIP is not active on the terminal of UA1. In this scenario, two variations are possible.

1. UA1 (on a WLAN) and UA2 (on a LAN) start a media session; then UA1 moves from a WLAN to GPRS. WLAN and GPRS networks are both available, WLAN is selected, and the terminal of UA1 has two active network interfaces with different IP addresses. The scenario starts after UA1 and UA2 are both registered on the SIP server and an active multimedia session with both audio and video has been established. The trigger for a network handover could be the detection of a WLAN signal loss. The mobility manager reacts by selecting GPRS as the new default network and sending the New default network event to the decision maker. Then, the application sends re-REGISTER and re-INVITE messages

2. UA1 (on a WLAN) and UA2 (on a LAN) start a media session; then UA1 moves from WLAN to LAN. MIP is not active on the terminal of UA1. This scenario can only be executed if the terminal of UA1 has both a WLAN interface and a LAN interface and the LAN interface is not active. The scenario starts in the same way as the previous one, i.e., after UA1 and UA2 are both registered on the SIP server and an active multimedia session with both audio and video has been established. The detection of the availability of the LAN interface on the terminal of UA1 is the trigger that starts the network handover. The mobility manager now has a choice of two networks; it selects the best possible network for this scenario (i.e., a LAN) and sends the New default network event to the decision maker. The decision maker finishes the scenario equally to the the previous variation, except that the signalling now uses the LAN connection rather than the GPRS connection used in the previous variation. As a result, the session is handed over with an interruption because of the change to the IP stack and because other codecs are used and possibly with new media characteristics, using SIP messages only.

3.1.6 Experimental results

The roaming scenarios as described above have been used as the base for the performance measurements of the prototype implementation of the MX SIP Communicator in combination with a SIP server. From the roaming scenarios a

	LAN (sec)		WLAN (sec)		GPRS (sec)	
	reREG	reINV	reREG	reINV	reREG	reINV
Mobile IP/SIP	n/a	2.5...5.6	n/a	2.5...12.3	n/a	2.7...13.6
SIP only	0.3	0.9	0.3	1.0	1.9	2.2

Table 3.2: Measured duration of the re-REGISTER (reREG) and re-INVITE (reINV) messages

combination of network changes was made that could be repeated in a consistent way, both with and without Mobile IP. This test scenario where one mobile client roams from LAN→LAN→GPRS→LAN and the other user agent does not move. The clients were configured such that only audio was available on WLAN and GPRS; both audio and video were available on LAN. The test scenario was executed twice; once with Mobile IP and once without Mobile IP. The SIP client used the UDP protocol both for data and signalling. During the execution of the scenario the behaviour of the client was observed. After the scenarios had been executed the performance was calculated.

To calculate the delay we used the time-stamps that the MX SIP Communicator adds to the messages that are logged in a file. Table 3.2 shows the duration of re-REGISTER and re-INVITE on SIP-message level. The re-REGISTER time is measured from sending the first REGISTER until the reception of the OK message. The re-INVITE time is measured from sending the first INVITE till sending the first ACK message. Please note that the handover time needed by the mobility manager is not included in these measurements and is completely independent of the SIP session control. During this handover time the following steps are taken: network detected, associated, authenticated, IP address available. Also the time needed for adapting the RTP streams is not included.

The measurements are repeated a dozen times. In case of a Mobile IP client with a co-located Care-of-Address and without a Foreign Agent, roaming with Mobile IP is slower because of the additional layer (SIP and MIP). The measured values for SIP-only are well reproducible, but measurement results for SIP over MIP have such a large spread that the minimum and maximum values are given. One reason for the large delays can be explained by the fact that the SIP control over UDP uses a retransmission scheme that doubles its back-off time after each retry.

3.1.6.1 SIP interoperability issues

In the SIP specification a number of options are mentioned. This leads almost by default to the conclusion that communication is not necessarily possible between two SIP entities despite the fact that both parties comply with RFC3261. A practical example we encountered was the fact that a signalling port number is optional in a SIP request. In that case the default port 5060 is assumed. Suppose that a user agent adds the default port number and the SIP server does not forward the default port number (because it is optional) to the destined user agent both SIP server and user agent comply with the standard. However, a problem arises during the processing of the ACK, because the ACK can be sent directly from one user agent to the other and now the port number is present but the destination does not recognize the message because it expects an ACK without the port number. In this case the SIP server was at fault, since RFC3261 states that specifying the default port results in another address than not specifying the default port.

3.1.6.2 SIP and Mobile IP

During the experiments with roaming and session hand-over the collaboration of Mobile IP and SIP could be studied in detail. Without the presence of SIP sessions Mobile IP provides a good mechanism to maintain connectivity between a mobile terminal and the connected network. But when a SIP application starts interacting as well with the network using the Mobile IP connection communication starts to get complicated. On one hand Mobile IP tries to hide network changes for the sessions. On the other hand SIP tries to optimize the sessions by re-inviting the other parties with session parameters that are adapted to the new network characteristics. In case the terminal moves to a lower bandwidth network there is no guarantee that the SIP control messages will reach their destination because the already established data flow can consume all available bandwidth leaving no room for SIP control messages. In this case the SIP re-INVITE will fail or will suffer a large delay. A practical example is the GPRS network where the assigned bandwidth varies and can be very low compared to LAN or WLAN. In case of in-band signalling, like the SIP messages, switching from WLAN that did allow high bandwidth to GPRS resulted more than once in message time-outs and terminated sessions.

3.1.7 Recommendations

After evaluating the results from the experiments we propose a number of recommendations to improve the mobility of SIP based applications. Enhancements can be made in the interface between the mobility manager and the SIP client. Also inside the SIP client enhancements are possible.

3.1.7.1 Notify network changes in advance

As a solution to circumvent network overload problems when moving to a network with lower bandwidth, a new event is introduced: *About-to-switch*. This event is to be provided by the mobility manager and will be sent a short time before switching to a new network, so that multimedia bandwidth can be adjusted just before switching to the new network. Then the following scenario can be implemented:

1. The mobility manager sends an event *About-to-switch* to the decision maker
2. If the current bandwidth demand of the session exceeds the bandwidth available on the new network the decision maker performs a re-INVITE on the old medium
3. The mobility manager is notified that the application is ready to switch to new medium
4. The mobility manager performs the switch
5. The decision maker performs a re-REGISTER on new medium and a re-INVITE on new medium for updating the contact address

3.1.7.2 Separation of data and control

Closely related to the previous recommendation is the recommendation to separate the data flow from the control flow in those cases where more than one network is available for the terminal, for example GPRS and WLAN. Control messages could be sent over the GPRS network and the data over WLAN. This will prevent that SIP control messages get lost. Separation of data and control can also be realized by used networks that support quality of service and can provide guaranteed bandwidth. To give priority to SIP control messages and Mobile IP binding updates, these messages can be given preference in the traffic shaper that exists in modern operating systems (like Linux, FreeBSD

and Microsoft Windows 2000/XP). It depends on the Mobile IP implementation whether this is possible or not because not all implementations allow a secondary interface to be active.

3.2 Partial Session Mobility

As mentioned in the multimedia session paragraph of Section 2.4.2, partial session mobility can be defined as session support where multimedia streams/connections can be dropped, added or moved to other endpoints. A typical use case for partial session mobility can be envisioned when people use video conferencing on their mobile devices to communicate with others. While having a video conference on a mobile device, entering a conference room could trigger a proposal for transferring the video to a big screen located in that room. This proposal could typically be shown on the mobile device of the user, and after the user agrees, the video stream could be automatically transferred to the big screen without any noticeable interruption. The mobile device could also offer the possibility to retrieve the video stream, for example, when leaving the conference room.

Developments in the SIP standard on session mobility [138] aim to use auxiliary devices discovered in the vicinity of the user to enhance ongoing multimedia sessions. These developments focus specifically on terminal-initiated transfer of multimedia stream endpoints to other devices. Typical devices to be discovered are those with a speaker, microphone, display and/or camera. The main goal of the work described in this section is to initiate such a transfer from a dedicated node in the network, in order to be able to support these transfers as a network service without necessarily having to upgrade all client devices. Network-initiation can also be preferable because terminal-initiated discovery of candidate devices and transfers may prove quite complex in terms of processing and (mobile) network usage. Additionally, guest use of devices and associated privacy and charging could be easier for an operator since it already has privacy and charging in place for other purposes.

Network-initiated partial session mobility would typically be triggered by external events like network-side discovery of nearby multimedia devices, such as a big screen. To offer network-initiated transfers of session parts as a service, the dedicated network node would typically be an Application Server (AS) when applied in the IP Multimedia Subsystem (IMS), standardized by the 3rd Generation Partnership Project [8] (3GPP). Since IMS is aware of all available multimedia devices and could be or become aware of their capabilities and loca-

tion as explained later in this section, an application server is a good candidate to initiate partial session mobility from the network. Device discovery from the mobile device is expected to consume more bandwidth, time, processing and associated costs and battery power consumption.

3.2.1 Objectives

Since users may not always want to enhance their multimedia sessions with additional devices, for example because they will not stay around the devices for a long period of time, the user needs to be involved in the decision to change session parameters. This involvement is guaranteed by first proposing the session change to the mobile device. The following information is required in such a session change proposal:

- The media stream endpoint to be transferred,
- The device to transfer the stream endpoint to, and
- The media parameters to be used for the media stream.

After a media stream endpoint has been transferred to another device, it is useful to enable its transfer back to the originating mobile terminal, e.g., when the user starts moving. This transfer may be initiated by either the dedicated network node or by the user. Since the user is more likely to know when to stop using auxiliary devices in a session, it should also be possible to initiate these transfers from the mobile device. Alternatively, when a device that has been added disconnects, the associated media stream endpoints are expected to be restored to the mobile device.

To ease deployment of the functionality described above, basic functionality should be offered for legacy SIP clients. The objectives described above can be summarized as follows:

- Transfer of individual media stream endpoints to other devices.
  - Support for both network-initiated and terminal-initiated transfer.
  - User involvement in a network-initiated transfer decision.
  - Transferred media streams can be transferred back to the mobile device.
  - The initiator of the transfer should be able to influence the media parameters of the transferred media stream.

- Compatibility with current SIP-User Agents (UAs).
- Robustness when involved nodes suddenly disconnect.

3.2.2 Existing methods

As described by Shacham and his co-authors [138], different techniques exist to enable session mobility in SIP. The mobile node control mode uses third party call control [128] to let the Mobile Node (MN) coordinate the transfer of media streams to another device, called Local Node (LN). With this technique, the MN stays in the signalling path, while the media streams flow directly between the new device and the Corresponding Node (CN). Figure 2.6 on page 35 shows control sessions mediated by the MN that specify a video streams between the MN and CN, and audio streams between the audio node and CN. The session handoff mode uses SIP REFER [143] to transfer the complete session to the other device. With this technique, the MN does not stay in the signalling path of the session, and therefore cannot retrieve the session directly using re-INVITEs.

Shacham [138] also describes how both the mobile node control mode and the session handoff mode can be used to transfer part of the media streams to another device. The mobile node control mode uses SDP capabilities to define different endpoints for each media stream. The session handoff mode uses a multi-device system (MDS), where the Multi-Device System Manager (MDSM) acts as a gateway for both SIP signaling and media, distributing the media streams to different devices.

Mani and Crespi [96] introduce a network-side mobility server that acts as the MDSM in the session handoff mode. They assert that the mobile node control mode is not acceptable in IMS because a transferred session will be controlled by a terminal which is no longer involved in the session. This might be true for full session mobility, however, the MN could still be involved in the session, (e.g., as mediator in the session to change or aggregate the session parts later, and as one of the possible media endpoints), and it could be charged for the whole or part of the session.

Within the Daidalos project [14][17] another method has been proposed to support the transfer of certain media stream(s) to another device. The Daidalos method also uses the REFER method [143], extended with multiple refer [44]. With this method, the MN sends a REFER message to the device of the other user in the session, the CN. This message contains references to the different devices to be invited for a specific part of the original session. This technique does not yet specify how the MN can indicate which part of the original session

must be transferred to what device.

A mobility SIP header is proposed by Peng in [102]. This header contains the call-ID of the session between the MN and CN, and information about the specific part of the session that must be transferred. This method is also based on REFER messages, but in contrast to the methods previously described, it uses a session-structure to correlate the different sessions that the CN sets up with the additional devices to the session between the CN and MN.

On the network side, other mechanisms exist for session transfer between a terminal's IP and circuit voice interface and are specified in the 3GPP voice call continuity (VCC) standards [7]. These mechanisms currently do not target usage of multiple devices at each side of the session.

3.2.3 Evaluating existing methods

As described in section 3.2.2, multiple methods exist to initiate the transfer of session endpoints from a mobile device to another device. This section evaluates these methods to find the most suitable method to support both network-initiated and user-terminal initiated partial session transfers. The methods are evaluated against the objectives stated in section 3.2.1. Additionally, separation of concerns identifies a general guideline to keep solutions simple by making sure certain logic is located on the most appropriate place in the architecture.

3.2.3.1 Mobile Node Control Mode

Using this method, the MN stays in the signalling path after the media stream(s) have been transferred to another device. This has a number of consequences: The MN can easily retrieve a media stream by sending a re-INVITE message to the CN. The MN has direct influence on the body of the media parameters of the transferred media stream(s) because the MN actually sends the INVITE message(s) to the LN(s). If a LN used in the session suddenly disconnects, the MN can change the session accordingly by issuing a re-INVITE message to the CN. If the MN suddenly disconnects, the media streams between CN and LNs remain in place while the signalling path is broken. This means these sessions cannot be closed properly. Because this method uses third party call control [128], it is also possible to let the sub-session controller manage the sessions and sub-sessions of the MN. In this case, the subsession controller [13] (SSC) must be located in the signalling path of these sessions. Also, because the mobile node control mode is based on existing functionality of SIP, no other extensions are necessary.

3.2.3.2 Session Handoff Mode

Using this method, the MN does not stay in the signalling path, but the whole session is transferred to the multi-device system manager. This has the following consequences: The MN cannot simply send a re-INVITE message to the CN to retrieve the session; it can only ask the MDSM, using a nested REFER message [144], if it wants to transfer the streams back. The MN does not control the media parameters of the media stream that is transferred to the LN because it cannot indicate this in the REFER message that is being sent to the MDSM. In case the MN suddenly disconnects after session transfer to the MDSM, the media streams that are not located on the MN will continue to exist; however the user cannot stop the session. If the MDSM suddenly disconnects, the user is also not able to retrieve the streams, because a nested REFER message has to be sent to the MDSM to do so. When the MN suddenly disconnects after a media stream has been transferred, the signaling path would not be broken, meaning the CN or MDSM could close the session properly. On the other hand, if the MDSM suddenly disconnects, even if the MN is still connected, the signaling path would be broken. A dedicated network node could send a REFER message to the MSDM, to let it replace the session between the CN and MN. With this mechanism, the dedicated node that initiates the transfer should be authorized to let the MSDM replace the session.

3.2.3.3 Multiple Refer

In the multiple refer method, the CN is the node combining related sessions instead of the MDSM. The MN uses a REFER message to let the CN transfer a media stream to another node. Using REFER messages, the MN cannot prescribe the content of the INVITE message, meaning the MN has no control over the media parameters. Because the MN is not in the signaling path of the created sub-sessions, it does not have all the information necessary to decide when it must transfer which stream to what device. However, it could initiate a partial session transfer to retrieve a media stream back to the MN, for which it would use a REFER message with a replace header. This way the MN is also able to retrieve a media stream. As with the session handoff mode, another node besides the MN could send the REFER message to the CN, however this node would need the call-ID of the session between the CN and MN. If this node does have access to this information, network-initiated partial session transfer is possible. With this method, the CN contains the intelligence to relate a session with the sub-sessions needed for a service provided by the media node. However,

ideally, the CN should not be controlling a partial session transfer requested by the MN, due to separation of concerns. Because the CN sets up the sub-session with the LNs, it would be hard to relate both volume- and signaling-based [120] sub-session charging to the MN. In addition, when the MN suddenly disconnects after a media stream has been transferred, the part that has been transferred can be closed properly because the MN is not located in the signaling path of the session between the CN and LNs.

3.2.3.4 Mobility Header

As with the multiple refer method, in the mobility header method, the CN is the node that combines the different sessions together. However, with this method the MN sends a REFER message to the LN if it wants the LN to take over a media stream. Peng [102] does not explicitly define how retrieval works, however, a media stream that has already been transferred can be retrieved because it has an ongoing session with the CN. This session can be re-activated with a re-INVITE message. Because all involved nodes should support the mobility header, support for partial session mobility for the MN depends on the capabilities of the CN and involved LNs. Just as with the multiple refer method, the MN does not have control of the actual media parameters of the transferred media stream. With respect to separation of concerns, for this method, as well as the multiple refer method, ideally the CN should not be saddled with controlling the transfer of a media stream to another device. The REFER message sent to the LN by the MN to transfer a media stream could also be sent by a SSC. When it is necessary to close the session between the CN and MN, the SSC should be able to act as a Back-to-back User Agent (B2BUA) in the session. Because the CN is the central node between the sessions, it is the weakest link with respect to robustness. At the moment the CN suddenly disconnects, the MN has no direct influence on the session between the CN and the local devices, although the MN might have to pay for those sessions. As with the multiple refer method, it also would be hard to relate charging of the sub-sessions to the MN.

3.2.3.5 Comparison of existing methods

The existing methods are compared along the objectives (see Section 3.2.1), i.e. the following evaluation criteria are used:

1. Potential to support network-initiated partial session transfers,

2. Transfer of a media stream back to the user device,
3. Control of the media parameters by the initiator,
4. Robustness,
5. Compatibility with SIP-UAs that do not explicitly support the method, and
6. Separation of concerns.

Table 3.3 compares the different methods with respect to these criteria based on the described advantages and disadvantages of the existing methods in section 3.2.3. When all evaluation criteria would have equal priority, the mobile node control mode is the most promising method to use. The session handoff mode, multiple refer method, and mobility header method all use REFER messages to execute a partial session transfer, while the mobile node control mode uses re-INVITE messages. The REFER based messages have the disadvantage of not having the ability for the initiator of the partial session transfer to prescribe the exact SDP body that should be offered to the LNs. Another problem that occurs in almost all methods that use REFER messages is that a partial session transfer cannot be undone after a node transfers a media stream, because the MN is no longer directly involved in the session that contains the transferred media stream. However, with the mobility header method, the original media stream can be set up again, though full retrieval of the stream is not yet supported. Both the multiple refer and mobility header methods involve the CN in a partial session transfer that is being executed for the benefit of the MN. With respect to separation of concerns, this is far from ideal. In the mobile node control mode, the CN is not involved in the partial session transfer on the control layer. Besides this, the mobile node control mode and the session handoff mode both do not need an extension of SIP.

3.2.4 Proposed method

Besides mediation from the media endpoints (as in the described exiting methods above), it is also possible to start and manage the control sessions of a multimedia session via a dedicated network node that operates on the control plane (i.e. from within the signalling path). The most promising method, i.e. the Mobile Node Control Mode, is taken as basis for a new method in a network node that initiates and manages partial session mobility. The proposed

method/criteria	1	2	3	4	5	6
Mobile node control	√	√	√	−	√	√
Session handoff	±	±	−	±	√	√
Multiple refer	±	√	−	±	±	−
Mobility header	√	√	−	±	−	−

Table 3.3: Matching different partial session mobility methods with evaluation criteria: fully fulfilled (√), partially fulfilled/fulfilled with issues (±), not fulfilled (−)

network node is called the SSC and must be able to start sessions on behalf of nodes and interpret sessions between nodes. Therefore, the SSC uses B2BUA functionality. A B2BUA can participate in multiple sessions and connect them by acting on behalf of the end nodes.

Figure 3.8 shows how the sub-session controller would be positioned within the control sessions after the audio session has been moved from the MN to the audio node. Before setting up the sub-session between MN and audio node, the sub-session controller would first need to propose the transfer of the audio session endpoint from the MN to the audio node. This proposal would typically be triggered by inferring the context of the user and its neighbourhood (e.g. when a big display is discovered in the neighbourhood during a audio/video session). How this context inferencing works is beyond the scope of this thesis. Details about a proposed method and validation with a prototype of the SSC implementing this method can be found in the next sections.

3.2.5 Method Overview

Proposing and initiating a partial session transfer involves a number of steps. Figure 3.9 shows the message sequence diagram for this method. The method assumes an ongoing session between the CN and MN (message 1). In message 2 through message 4, the SSC proposes a partial session transfer to the MN, which is accepted by the MN. In message 5 through message 8, the SSC uses third party call control to invite the LN to send audio to the CN. In message 9 through message 12, the MN is invited to stop sending audio to the CN. In message 13 through message 17, the SSC invites the CN to start sending audio to the LN instead of to the MN, without changing the video stream. The following sections provide further details.

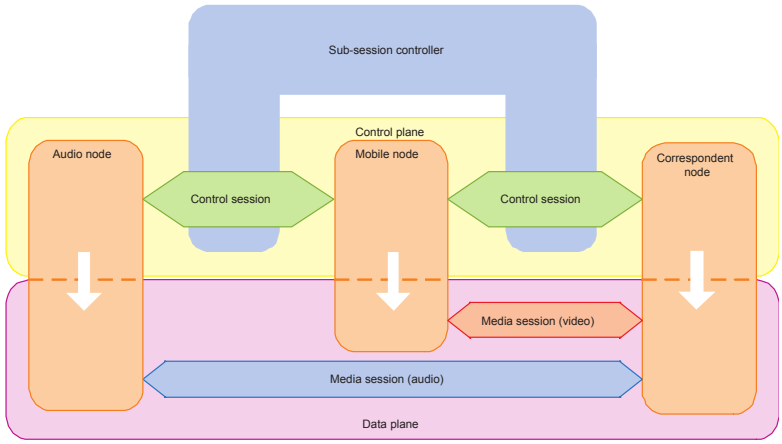


Figure 3.8: Sub-session controller after moving audio stream

3.2.5.1 Proposing a partial session transfer

As described previously, the SSC should propose a partial session transfer to the MN. The solution illustrated in Figure 3.9 uses an INVITE message (message 2) sent by the SSC to the MN on behalf of the CN. This INVITE message contains the method pst-control in the required header. This required header implies that the receiver of the message is only allowed to process the message if it supports that extension; if not, it should respond with a 420 (bad extension) response. When the SSC receives a 420 response, it knows the MN does not support the extension, and the SSC could apply user preferences in order to decide what to do next. Figure 3.10 illustrates the sequence of events when user preferences prescribe to continue with a partial session transfer after a 420 response, as indicated in message 3. As can be seen, the SSC continues the partial session transfer (message 5 through message 17) as in Figure 3.9 even after the MN replies with a 420 response.

As illustrated in Figure 3.9, at message 3, the MN responds with a 200 (OK) response to indicate accepted partial session transfer by the MN. When the MN does not accept the transfer, it delivers a 603 (decline) response, as illustrated in Figure 3.11 .

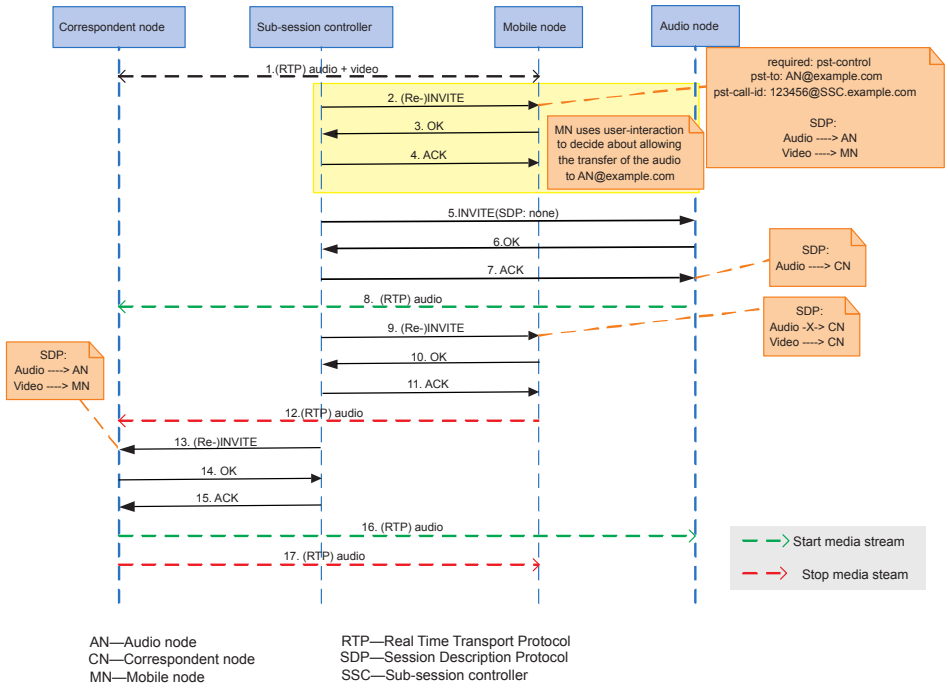


Figure 3.9: Method overview

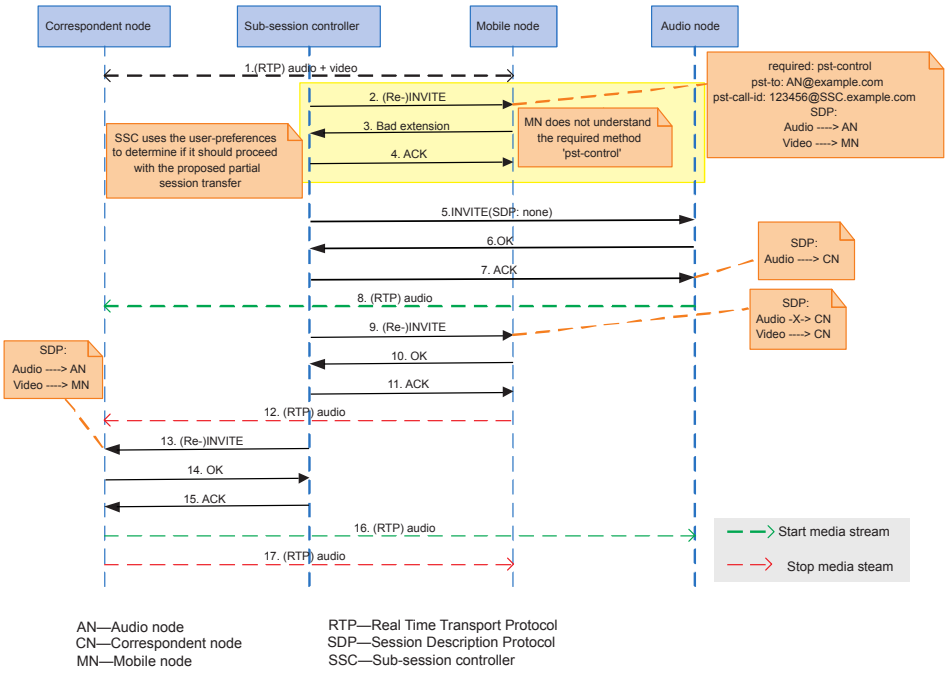


Figure 3.10: MN does not support extention

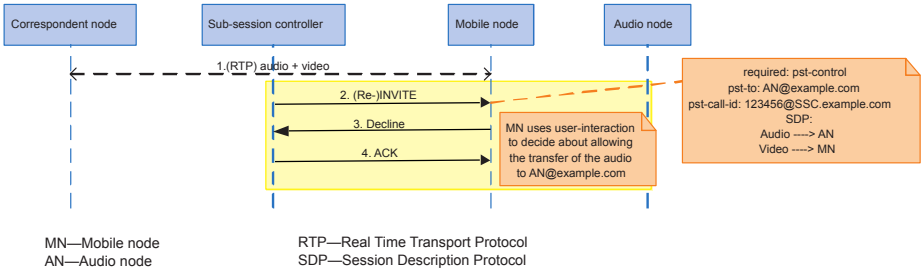


Figure 3.11: User declines the request

3.2.5.2 Partial session transfer description

The proposed transfer issued in INVITE (message 2) contains information in an SDP body, to support the user decision whether or not to proceed with the proposed transfer (as described in Section 3.2.1). This SDP body, however, should not be interpreted as usual in a normal negotiation [127]; it has another meaning as will be described below.

In SIP, a media stream is defined by the combination of the SDPs of each endpoint of the session, and each endpoint has its own view of the media session defined in the corresponding SDP. The SDP body in INVITE (message 2) contains the proposed session description to be sent to the CN. The MN can compare each media description in this SDP body with the corresponding media description in the SDP body that was last sent to the CN for setting up or changing the session. Since the IP-address and/or port number changes when doing partial session transfer, the media description in the proposal changes accordingly. This enables the MN to recognize the stream to be transferred, and possibly the changed media parameters, based on the changed media description. Because the SDP does not include a SIP uniform resource identifier (URI) to identify the device the stream is being transferred to, the new header pst-to is added to the proposing INVITE message (message 2) which contains it. Figure 3.9 also shows this header in the comment at message 2.

3.2.5.3 Retrieve or move transferred media stream

After the SSC or MN has executed a partial session transfer, it must be possible to transfer it again, back to the MN or to another LN. A network-initiated partial session transfer is only possible after a terminal-initiated transfer when the SSC correctly interpreted the signalling between the MN and other nodes. A terminal-initiated partial session transfer after a network-initiated partial session transfer is more complicated, because the MN does not yet have an explicit session with the specified LN. During the network-initiated partial session transfer, the SSC did set up a session with the LN on behalf of the MN, but the MN was not involved in the session setup and does therefore not know the call-ID of this session.

Because of this, the new pst-call-ID header is contained in the proposing INVITE message, with the call-ID. This new header is also illustrated in Figure 3.9 in the comment at message 2. With this call-ID, the MN can close or change the session with the LN when it wants to. The SSC contains the logic to make sure SIP messages sent with this call-ID are delivered at the LN. Figure 3.12

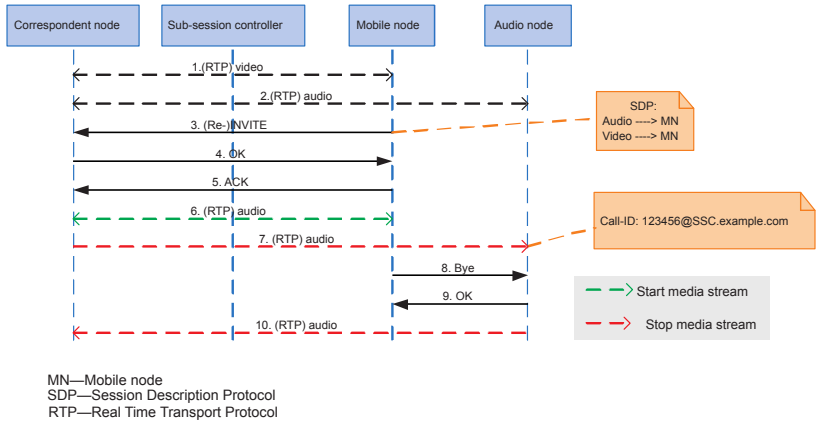


Figure 3.12: Transfer back a media stream

illustrates the situation where the MN retrieves the media stream (message 3 through message 7) and closes the sub-session with the LN (message 8 through message 10).

3.2.5.4 Managing Sub-Sessions

In both network-initiated and terminal-initiated partial session mobility, sub-sessions are set up between the mobile node and the local nodes. This section describes which component is responsible for dealing with these sub-sessions and in what situation. The basic principle for managing sub-sessions is that the node (MN or SSC) that last changed a sub-session is responsible for the sub-session. For example, when the MN sends a re-INVITE to the CN after a network-initiated partial session transfer, the SSC would be responsible for making corresponding changes to the sub-sessions with the LNs when the MN did not change the sub-sessions. Since the MN can only change sub-sessions when it initiates them or when it supports the pst-control extension, the SSC can easily deduce when it is responsible for sub-sessions with the following rules:

- When MN supports pst-control
  - The node that made the last change to an individual sub-session is responsible for that sub-session. So, after a network-initiated partial session transfer, the SSC is responsible for the involved sub-session

until the MN changes that sub-session.

  - This also means that if the MN is responsible for an individual sub-session, and if it does change the session with the CN, it should also make sure the sub-session with the LN is still consistent with the session between the MN and CN.
  - When the MN closes or changes a sub-session, it is responsible for changing the session with the CN accordingly.

- When the MN does not support pst-control:
  - When an MN supports terminal-initiated partial session mobility, the node that last changed a sub-session is responsible for that sub-session. This means that the MN is no longer responsible for an initiated sub-session, once the SSC modifies it.
  - When an MN does not support terminal-initiated partial session mobility, the SSC needs to make sure that the sub-session(s) remain consistent when the MN changes something in the session with the CN.

Please note that the combination of partial session mobility with full session mobility is not yet considered. Thus, one known issue with the rules above is a session transfer from an MN that does support the pst-extension to one that does not. As described earlier, the mobile node control mode does have a problem related to robustness at the moment the MN suddenly disconnects while one of the media streams has already been transferred to another device. In this situation, the media stream involved continues to exist, and the user is not able to use his mobile terminal to stop the media stream. In this case, the user should use the LN interface to stop the stream. With the solution described here, the SSC is always involved in all sub-sessions related to the MN. Therefore, the SSC could easily stop all related sub-session activity when it is notified of a disconnection of the MN.

3.2.6 Validation

The sub-session controller was prototyped in order to validate the following aspects of network-initiated partial session mobility:

- Proposing a partial session transfer to the user terminal.
- Executing the network initiated partial session transfer.

- Re-transfer of an already transferred media stream (also back to the mobile node).
- Controlling the media parameters of the transferred media stream.
- Compatibility with current SIP-UAs that do not support the extensions introduced in this thesis.
- Minimizing the disruption of the media stream (theoretical approach).

After partial session transfers were proposed and accepted by the mobile node, the prototyped sub-session controller could successfully transfer media stream endpoints multiple times between a number of local nodes including the mobile device. Measurements and analysis indicated that a full-fledged SSC would not influence the continuity of the multimedia streaming because it would apply make before break. This streaming continuity would be mainly affected by: a) the time it takes a particular SIP-UA to process SIP messages and initiate streaming; and b) network-level delays of both signalling and streams.

3.2.7 Conclusions and future work

We described a network node that can initiate partial session mobility in order to span a multimedia session across multiple devices. Major challenges involved finding a method that keeps the user informed and in control, that works together with terminal-initiated partial session mobility, and that has (limited) support for legacy SIP user agents. To overcome these challenges, the most appropriate terminal-initiated method was modified to be applied from within the signalling session between session peers while still allowing terminal initiation, and extended to allow user awareness and control. SIP-UAs that do not support the extension could use a user profile setting to indicate when to pursue a transfer.

The validation of the prototyped sub-session controller shows that the described network-initiated partial session mobility method works in practice and indicated continuity of multimedia streaming in real deployment of the method.

For future research, validation of the combination of terminal- and network-initiated partial session mobility is needed, for which a SIP-UA needs to be extended with terminal-initiated partial session mobility and the proposed SIP extension. Other interesting research topics include scalability, combination with user mobility, merging voice call continuity with the proposed method,

adding broadcast and multicast streams to an existing SIP session, and supporting group sessions for multicast group setup and switching between unicast, multicast and/or a conference server.

3.3 Mobility of sensor networks

In this section we analyse the mobility of Internet-enabled WSANs by applications. Example applications are remote monitoring of goods that are transported between warehouses, monitoring of persons with health-related problems, and remotely controlling lights or motors (actuation). We focus on the Body Sensor Network (BSN), Vehicle Sensor Network (VSN) and Structure Sensor Network (SSN) as defined in Section 2.1.2 where mobility can be a concern.

We analyse the different movements that can take place in and across WSANs. Furthermore, we analyse the movement of Internet-connected WSANs and applications that use them. These IP applications can use sensor information from the WSAN as well as configure and actuate the elements of individual nodes. The purpose of our analysis is to gain insight in the different types of mobility and to determine how they can best be supported in different usage scenarios. A lot of research has been done on mobility within WSANs (e.g. in [20, 119, 163]. However, in this section we focus on mobility issues of: a) nodes that move between WSANs, b) WSANs that move in each other's range, and c) IP applications that use the sensor information.

This section is organized as follows. In Section 3.3.1 these WSAN types are used in mobility scenarios where IP application(s) use the WSANs. In Section 3.3.2 the types of mobility related to WSANs and IP applications are further detailed and the consequences of these mobility types are analysed. Section 3.3.3 further analyses how to support these mobility types in the scenarios.

3.3.1 Mobility scenarios

WSANs can bring clear benefits to logistic processes with measurements of goods during storage and transport [36]. Healthcare, wellbeing and sport-related person monitoring with WSANs is another area that gains research attention [26]. We have defined four scenarios where different types of mobility take place when nodes, complete WSANs or IP applications using the sensor data are moving. Two scenarios are described where a truck with monitored goods moves between distribution centres and two where a monitored person moves around. For both the trucks and the monitored persons, an IP application can run remotely or

be directly attached to the WSAN while using information from other remotely running IP applications. Both a smartphone and router can be the IP gateway (IPG) for WSANs and IP applications. Applications can typically run on the smartphone, computers attached to the router, in the enterprise or on cloud servers.

3.3.1.1 Moving vehicle sensor network

In this scenario, goods are tagged [58] with a sensor node. This sensor node travels with it when it moves with a truck between distribution centres. The trucks have a VSN deployed and the distribution centres have an SSN deployed, see Figure 3.13. All sensor data, including Global Positioning System (GPS) location, are provided to the monitoring application. The VSN in Truck 1 may loose its connection to the monitoring application when travelling through low-coverage areas (for instance tunnels) and the IPG will roam to other GPRS network providers when going abroad. The monitoring application would typically offer realtime insight in the conditions of the goods, both when in storage and during transit. Based on condition deterioration, the truck could be re-routed to a closer-by destination.

3.3.1.2 Moving vehicle application

In this scenario, truck 2 in Figure 3.13 will have a GPRS connection to the Internet, and the vehicle application may loose its connection to the monitoring application when travelling through low-coverage areas. In addition, the IPG will roam to other network providers when going abroad. An example vehicle application could be monitoring the condition of goods in the truck, and comparing the measurements with the inventory list to see if nothing is lost, misplaced or spoiled. Via the monitoring application, the vehicle application could check historic conditions of the goods, and location of missing goods or replacements.

3.3.1.3 Moving body sensor network

In this scenario, a man with BSN 2 and a smartphone moves between two houses with WLAN coverage and a deployed SSN. The man uses objects that have sensor nodes attached that are compatible with the BSN. The BSN is used by a group application running remotely on the Internet (for example monitoring health status, location, and/or other monitoring applications), see

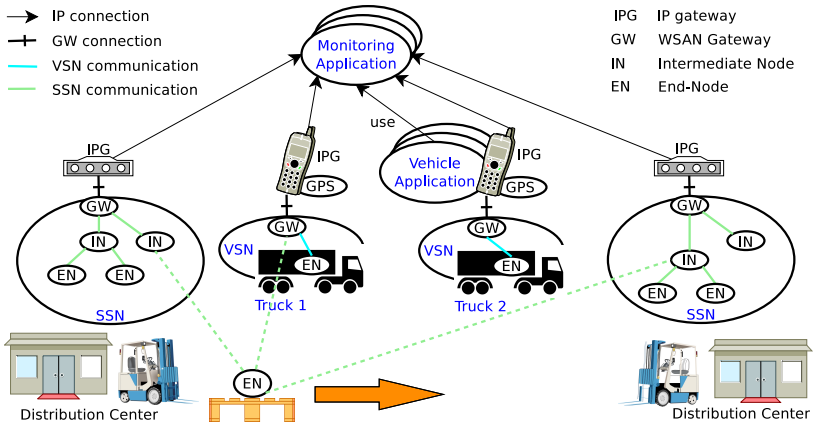


Figure 3.13: Monitoring moving goods in logistics

Figure 3.14. The smartphone will use the cheapest available Internet connection for communication to the Internet, such as WLAN.

3.3.1.4 Moving personal application

In this scenario a woman with BSN 1 and smartphone moves between two houses with WLAN coverage and deployed SSN and uses sensor information from these SSN nodes. The BSN is used by a personal application running on the smartphone that she carries, see Figure 3.14. The smartphone will use the cheapest available Internet connection for obtaining measurements from a monitoring application. This monitoring application provides real-time sensor information from buildings based on GPS location.

3.3.2 Analysis of mobility types

Since WSAN nodes and its gateway can be attached to multiple moving objects, multiple types of mobility can occur within and across WSANs. Additionally, a device that hosts an IP application using the sensor data can move as well. A wireless node can be an end-node equipped with sensors and/or actuators, or an intermediate node that extends the coverage area of the WSAN.

This section makes a distinction between the following WSAN nodes: the **gateway** that makes the WSAN available to applications, **intermediate nodes**

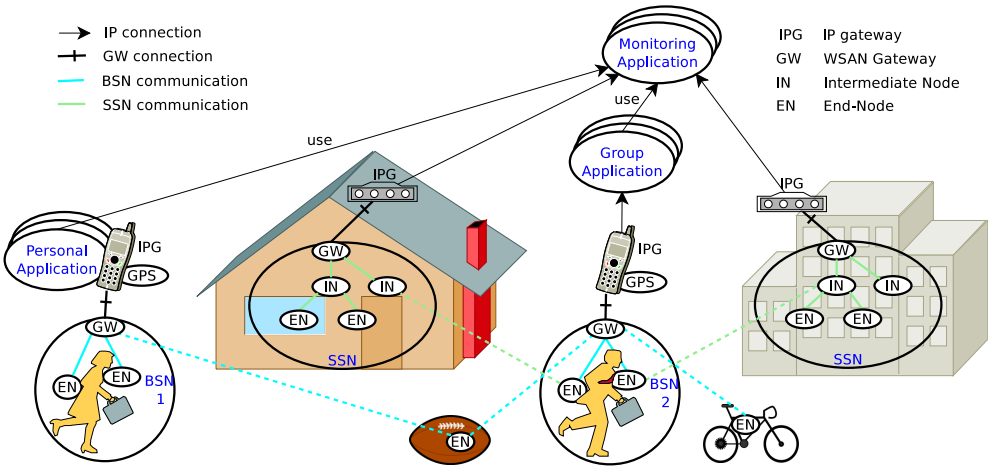


Figure 3.14: Moving BSN and personal applications

that extend the coverage of the WSAN gateway, and **end-nodes** that can connect to the intermediate nodes or gateway. Although we assume that the end-nodes do not change into intermediate nodes (like in the Ambient WSAN [36]), most of the mobility types apply when they do (such as with the Collection Tree Protocol [65] (CTP)). In the CTP, an end-node can join the WSAN via another end-node, turning the latter into an intermediate node.

The **wireless resources** used by a WSAN are characterised by one or more radio channels and the type of radio transmission. Example radio transmission types are: probabilistic such as in Carrier Sense Multiple Access (CSMA), using timeslots such as in Time Division Multiple Access (TDMA), and frequency hopping such as used in Bluetooth. Different WSANs are defined to be **compatible** when nodes of the WSANs can communicate with both gateways, i.e.: they use the same WSAN protocol, use the same wireless resources and share the same encryption key.

We distinguish the following types of mobility related to WSANs:

- A moving IPG. Network Mobility takes place when the IPG starts using another wireless or wired network technology or starts using a different network provider on the same network technology. The implication of this change is that the Internet Protocol (IP) address of the IPG changes which will break connections when there is no transparent mobility sup-

port (like MIP) in place. For short-lived connections like via HTTP, this connection break will result in a time-out. Movement can also make the IPG unreachable when there is no network coverage, or when it moves into a private or protected network. The moving IPG affects:

  - an attached WSAN. The IPG provides the WSAN with Internet connectivity for applications that want to use info from, configure or actuate nodes in the WSAN. Examples are moving BSNs and VSNs. The implication of movement can be (un)reachability and (dis)connection of IP applications.
  - an attached IP application. An IP application can use sensor data from nearby or remote WSANs via TCP/IP. The IPG movement can break existing connections from the IP application to the WSAN and make others possible.

- A moving WSAN, for example a truck with a WSAN gateway and nodes. When the WSAN moves in range of another WSAN, matching wireless resources may require changing these resources in one of the WSANs to avoid bandwidth degradation and possible collisions. When compatible WSANs move in range, the nodes may associate to both of them. When a compatible WSAN moves in range of an intermediate or end-node, that node may join the WSAN. When the WSAN moves out of range of an associated intermediate or end-node, the association will be lost.

- A moving intermediate node (with or without connected nodes)

  - within a WSAN, for instance a forklift with an intermediate node attached can extend the radio coverage of the WSAN in the direction it moves and allow end-nodes to communicate. When this intermediate node moves in range of a compatible WSAN gateway or other intermediate node, it has the option to join that WSAN, when it moves out of range it will loose the connection when it was associated. When the intermediate node moves in range of a compatible end-node, that end-node may join the WSAN. When the intermediate node moves out of range of an associated end-node, the end-node will loose its association.
  - across WSANs, for instance an intermediate node attached to a forklift moving between the coverage areas of compatible WSANs and picks up goods with attached end-node(s). The intermediate node can join the other WSAN when it is out of range of the other one, and

can choose the WSAN when it is in range of both. When it comes in range of another node, that node can choose to join it, when it goes out of range of a node that node will loose its association unless there is an alternative intermediate node or gateway in range.

- A moving end-node
  - within a WSAN, the node may have to communicate via multiple intermediate nodes depending on their radio coverage. When an end-node moves in range of a compatible WSAN or connected intermediate node it can join it. When it moves out of range of such a WSAN, it will be disassociated. When it moves out of range of a compatible intermediate node, it will be disassociated unless there is an alternative intermediate node or gateway in range.
  - across WSANs, for instance an end-node that is moved together with goods from a WSAN-enabled truck into a WSAN enabled storage area (see Section 3.3.1). When an end-node moves in range of a compatible WSAN or intermediate node, it can join it. When it moves out of range of such a WSAN it will be disassociated. When it moves out of range of an intermediate node, it will be disassociated when there is no alternative in range.

Table 3.4 summarizes the mobility consequences on the data link layer when a WSAN, intermediate- or end-node moves in or out of range of another compatible WSAN, intermediate- or end-node. Before this movement, the moving entity can be associated or not, for WSANs this does not apply. When WSANs come in each other's range, they may need to re-allocate their wireless resources when they use the same ones. When an intermediate or end-node comes in range of another WSAN, it has the option to associate with that WSAN (denoted as "option") or choose it as an alternative link. When an intermediate or end-node moves out of range of an intermediate node, it may still have an alternatives to use, else the association with the WSAN is lost.

Table 3.5 summarizes what happens when an IPG with attached WSAN or IP application moves in or out of range of another IPG with attached IP application or WSAN respectively. It assumes a bi-directional connection between the WSAN and the IP application. Moving in range here means that an IP connection becomes possible, moving out of range here means that the IP connection breaks (for example when no mobility protocol like MIP is in place and the IPG changes IP address). The moving IPG can have an attached IP application or WSAN that is connected or disconnected. When a connection

Table 3.4: Mobility consequences for compatible WSANs (data link layer), abbreviations: alloc. = allocate, alt. = alternative, assoc. = associated.

			Static					
			WSAN		interm. node		end-node	
		assoc.	in	out	in	out	in	out
Moving	WSAN	-	re-alloc.	ok	option	lost	option	lost
	interm. node	yes	alt.	lost	alt.	lost/alt.	option	lost/alt.
		no	option	lost	option			
	end-node	yes	alt.	lost	alt.	lost/alt.		
		no	option		option			

Table 3.5: Mobility of WSANs used by IP applications (network layer), abbreviations: alt. = alternative

				Static IPG with			
				WSAN		application	
			connected	in	out	in	out
Moving	IPG with	WSAN	yes			alt.	lost/re-route
			no			option	
		application	yes	alt.	lost/re-route		
			no	option			

breaks after moving, it may be re-established by setting up an alternative route or it may be lost when this is not possible. When a connection becomes possible after moving, this is denoted as "option".

Remarks on WSAN mobility types

- Clearly there are a number of options for connected nodes when another compatible WSAN comes in reach, how these nodes deal with this can vary per WSAN type. In Section 3.3.3 we analyse this further for the given scenarios.
- The above tables merely describe the case where *compatible* WSANs and its nodes are considered. When incompatible WSAN protocols, wireless resources or encryption are used, the nodes can not exchange information The gateway may still need to re-allocate resources when the other WSAN operates on the same channel. Section 3.3.3 analyses different solutions to support overlapping WSANs.
- Without mobility support, complete WSANs and IP applications will disconnect when the IPG changes IP address. For seamless mobility, a number of mobility schemes can be used (described in Section 4.3.1).
- WSAN nodes can potentially listen to messages in each of the WSAN they become part of, so they can also transfer information from one WSAN to another. Section 3.3.3 describes how data protection can be provided.

3.3.3 Analysing the mobility scenarios

In this section the mobility scenarios from Section 3.3.1 are analysed in the light of the different mobility types described in Section 3.3.2 and the level of mobility support that can be offered.

Important properties for mobility support in the scenarios are:

- **Security/privacy**: Security of WSANs is a complex issue. Cryptographic credentials can be used to authenticate a node in a network and to encrypt the traffic, examples of such credentials are keys and passwords. Keys can be symmetric, where one key is used for both encryption and decryption, or asymmetric, where a pair of keys is used for encryption and decryption. [91] provides a set of guidelines to handle security in WSANs. However, contrary to these guidelines, asymmetric encryption becomes possible in WSANs [93].

- **Interference**: networks that use the same wireless resources can potentially interfere with each other. This interference can take different forms. When the WSAN protocols use timeslots, misalignment may cause collisions in two slots for every message, while timeslot alignment limits this to maximally one collision per message. When the WSANs use probabilistic Media Access Control (MAC) protocols, the chance for collisions will increase since there are more nodes. When a combination of timeslots and probabilistic MAC protocols are used, all timeslots are likely to suffer packet loss. Adaptive MAC protocols (like [159, 66, 94]) could be used to reduce TDMA interference.

- **Overlap awareness**: when a WSAN is aware of the presence of another WSAN it can adapt itself accordingly. The first step to become aware is detecting an increase of interference. Next, a scan can be done to detect periodic traffic and silence on the radio channel. The detected periodic patterns can be used to adapt the WSAN traffic to reduce interference. Scanning can also be used to detect familiar WSAN types. When received messages can be decoded and are of non-registered intermediate nodes, there is a good chance that a compatible WSAN is nearby.

- **Wireless resource adaptation**: When a WSAN is aware of an overlapping WSAN, it can adapt its wireless resources to reduce interference. Examples of WSAN adaptation are: channel change, synchronisation and distribution of timeslots between WSANs, turning off the gateway, changing mode of operation (for instance change from gateway to intermediate node).

- **WSAN mobility**: what do nodes need to do to switch to another WSAN? Clearly this depends greatly on the WSAN type, for instance:

  - In the Ambient WSAN [36], all nodes have a unique 6 byte MAC address. The end-nodes (called SmartPoints) can sent messages (using CSMA) when they have the matching (symmetric) network key. The intermediate nodes (called MicroRouters) need the network key to announce them-selves to the gateway and to get (TDMA) timeslots assigned.
  - In an IPv6 over Low power Wireless Personal Access Networks [106] (6LoWPAN) network, the MAC address (2 upto 8 bytes) is used for node identification. Communication can be beacon-less (pure CSMA) or beacon-enabled (a hybrid of CSMA and TDMA). Nodes need to

register themselves using the 6LoWPAN customized neighbour discovery protocol, which makes a unique node address available in the WSAN and makes the WSAN network prefix available to the node. MIP can be used to make a node uniquely addressable when it moves between different WSANs. 6LoWPAN networks can utilize the symmetric keys of the IEEE 802.15.4 MAC.

- In the Inertia WSAN [72], the end-nodes have a 2 byte address assigned. This address is used in the registration message to the gateway to obtain a TDMA timeslot. There are no intermediary nodes, since this network is primarily targeted at small body-area networks. Objects with a node attached can be used by multiple WSANs in sequence. The messages are not (yet) encrypted.
- BSNs can also be constructed using Bluetooth which uses frequency hopping for radio transmission. Bluetooth is single-hop (research is done on multihop scatternets) and usually uses a powerful device like a smartphone or PC as master. For switching to another network, the master of the other BSN needs to pair with the device and connect to one of its services. When pairing is done beforehand, the master could be programmed to auto-connect to a specific service, which would enable mobility of devices between masters.

- **IP mobility**: Mobility often means that the IP address will change. Usage of transparent mobility schemes like MIP will hide the IP connection changes of the IPG and application. Connection outages would be experienced by the application as congestion, unless they are longer than the TCP/IP timeout. Because of this mobility transparency, it is more interesting to compare the properties of MIP for sharing, see Section 4.3.1.
- **Costs**: The wireless communication technologies have different associated costs. Using WLAN is generally cheaper or even free, while mobile data roaming via GPRS can vary from a relatively cheap data bundle to very costly when exceeding the bundle and when crossing nation borders. Internal WSANs communication is considered free of charge in this section.
- **Protocol robustness**: Protocols that are not robust against foreign messaging, will suffer most from interference. Methods to detect broken packages vary from a Cyclic Redundancy Check (CRC) check, to encryption (where decryption is likely to fail for broken packets). Techniques like forward error correction can be used to add redundancy to the messages to be able to reconstruct some of the broken messages when there is interference.

- WSAN compatibility, as defined in Section 3.3.2.

In the following subsections, we use these mobility properties to analyse the mobility scenarios from Section 3.3.1).

3.3.3.1 Moving vehicle sensor network (VSN)

In order to get a complete measurement trace (such as temperature and humidity variations) from the moment the sensor node comes out of storage in the first distribution centre until it arrives with the truck in the other distribution centre, measurements need to be merged at IP level in the monitoring application. In order to correctly correlate the measurements, an indication is required that the VSN gateway is in range of the SSN gateway. One indication is the fact that sensor nodes that were first reporting via the VSN start reporting via the SSN. Another indication is correlation of the GPS coordinates of the truck and the distribution centres. A third indication could be the detection of the SSN by the VSN gateway.

The most prominent changes that can occur when a VSN moves are:

- The VSN moves in range of the SSN or other VSNs (i.e. other trucks). When the WSANs use the same radio channel there can be interference. When the WSANs are compatible, nodes may report their measurements via the other WSAN.
- The VSN moves out of range of the SSN and potentially other VSNs. In this case the nodes that remain in coverage of the VSN need to associate with the VSN in order to transmit.
- The VSN moves in range of intermediate- and/or end-nodes. When these nodes are compatible with the VSN, they may associate with it.
- The VSN moves out of range of associated intermediate- and end-nodes. These nodes will no longer be able to transmit via the VSN, so they need to associate with the SSN or another VSN.
- The IPG in the truck may change attachment to different IP networks, e.g. when it moves from one country to the other. Additionally, Internet connectivity can be temporarily unavailable.
- The GPS coordinates of the truck and a distribution centre will differ when the truck is on the road, and be similar when the truck is close by. The proximity can be detected before the WSANs start to overlap.

As can be seen above, when compatible WSANs come within each other's range, their nodes can report to both WSANs. In this case the SSN should be capable to handle a few more nodes from the truck (since the nodes may go to storage anyway). The VSN however, has a limited Internet connection and could have a harder time with additional nodes. Additionally, the monitoring application would have a harder time distinguishing the additional nodes reporting to the VSN from the ones that are really inside the truck. Therefore, the following solutions are proposed to restrict the freedom of the nodes to attach to other WSANs:

- When a compatible WSAN is detected, the VSN gateway could be switched off. However, this could give problems when multiple VSNs are close together, since they may all decide to switch off. Furthermore, the nodes in the truck may not be able to reach the SSN from within the truck.

- When a compatible WSAN is detected, the VSN gateway could be switched to intermediate-node mode, so that it extends the coverage of that WSAN. However, this will put more load on the SSN and there may be a limit to the number of supported intermediate nodes (e.g. 64 in the Ambient network).

- Without detection, the WSANs can be separated by using different network keys, and only the nodes that need to be mobile between the WSANs can have multiple keys (i.e. the nodes that go from storage to transport to another storage). The nodes can decide themselves when they start using the other network key for transmission, e.g. switch to the SSN when the VSN link degrades.

Of course, also interference will be a concern for WSANs that share wireless resources. When using timeslots, this can partly be resolved by synchronizing and/or distributing timeslots. Alternatively, the VSN could be changed to use non-interfering wireless resources or different network key(s) before it reaches the distribution centre, for instance by detecting similarity in GPS coordinates of the truck and distribution centre and consulting via the monitoring application what resources are used by the SSN.

Additionally, since the IPG can change its IP address, it will need a mechanism to still report the measurements to the monitoring application. Obviously, the IPG could buffer measurements and send them after reconnecting to the monitoring application.

3.3.3.2 Moving vehicle application

The most prominent changes that can occur when a truck with vehicle application moves are:

- The IPG may connect to different GPRS or Universal Mobile Telecommunications System (UMTS) networks and optionally other wireless networks like WLAN.
- IP connectivity of the IPG can be temporary unavailable when there is bad or no wireless network coverage.

The implication of network attachment changes is often that the IP address of the IPG changes or becomes unavailable, which will break existing connections from the vehicle application or VSN to other IP applications on the Internet. When there is no connection, it will be impossible to connect to the monitoring application to fetch SSN measurements, in other cases the connection needs to be re-established.

Moreover, IP applications on the Internet that are using data from the vehicle application may be confronted with a changed IP address or unreachable IPG and associated connection breaks. The IP address of the IPG can be unreachable when not connected, when in a private area network, and when a restrictive firewall blocks the Internet traffic.

3.3.3.3 Moving body sensor network (BSN)

The most prominent changes that can occur when a BSN attached to a smartphone moves are:

- The BSN moves in range of the SSN and potentially other BSNs (i.e. other persons). When the WSANs use the same radio channel there can be interference. When the WSANs are compatible, nodes may report via the other WSAN.
- The BSN moves out of range of the SSN and potentially other BSNs. In this case the nodes that remain in coverage of the BSN need to associate with it in order to transmit.
- The BSN moves in range of objects with end-nodes. When these nodes are compatible with the BSN, they may associate with it and transmit their measurements.

- The BSN moves out of range of objects with associated end-nodes. These nodes will no longer be able to transmit via the BSN.

- The smartphone may connect to various wireless networks and Internet connectivity can be temporarily unavailable.

- The GPS coordinates of the smartphone and a SSN will differ when the person is out of range, and be similar when he/she is close by.

The following mobility support options can be considered in this scenario [1]:

1. **WLAN usage**: based on costs, the smartphone will have preference for WLAN instead of the more costly GPRS to send BSN messages to the group application (see Figure 3.14 on page 74). Of course a new connection needs to be established to the group application. When multi-homing is supported, the GPRS connection could be kept open while using WLAN. When moving out of WLAN range, GPRS will be used again and the WLAN connection to the application will break.

2. **Secured object use**: since objects can potentially listen, store and forward information, communication of more sensitive BSN sensor data should be encrypted.

3. **Separate uplink**: since the BSN and SSN need to connect to different applications, they use a separate IP connection. The BSN should use encryption for privacy-sensitive messages and its uplink should use encryption towards the application. Inter-BSN traffic is impractical for normal usage, so BSNs should use different encryption keys to protect privacy.

4. **BSN messages via compatible SSN**: when BSN and SSN are compatible, BSN end-nodes may use any intermediate SSN node or gateway to send their information upstream. The information could be encrypted such that only a specific application can decrypt it, for instance by using the public key of the application for encrypting the message payload. The connection details for the destined application should be conveyed to the IPG of the SSN gateway. This makes this a more customized and therefore less attractive option.

[1] Note that data protection is an important privacy aspect in BSNs

5. **Dual-stack BSN end-nodes**: end-nodes that can communicate both with the SSN and incompatible BSN. This can also be used to sent messages with encrypted payload upstream. Here, the BSN message destination also needs to be conveyed to the SSN gateway.

WLAN usage and encryption are a must for lowering communication costs and enhancing privacy. A separate IP uplink for the BSN and SSN messages is considered more practical than sending BSN messages via a compatible SSN.

3.3.3.4 Moving personal application

The most prominent changes that can occur when a person with a personal application on a smartphone with an attached BSN moves are:

- The smartphone may connect to different wireless networks and Internet connectivity can be temporarily unavailable. In case of WLAN, local access to the IPG of the SSN may become possible.
- The BSN can come in range of a SSN.
- The BSN can get out of range of the SSN.

The following options can be considered for a moving application (on a smartphone) that uses its attached BSN and nearby SSN data:

1. **Intranet access to SSN data**: Local access to SSN data may be possible in the associated Intranet when the smartphone is allowed to use this network. The SSN needs to advertise itself in some manner to enable discovery by the smartphone application.
2. **Public SSN server**: the SSN sends its sensor data to a publicly reachable server on the Internet from which applications can fetch it when they have the proper credentials. Retrieval could for example be based on the current GPS coordinates of the smartphone.
3. **Direct access to SSN nodes**: Intercepting sensor information from the SSN in an BSN end-node is not really feasible, since SSN nodes direct their readings only towards the gateway and sleep most of the time to save energy and bandwidth (so requests could take very long). It would also require a compatible WSANs.

The first two options are both viable. Direct access to SSN nodes is not really an option.

3.3.3.5 Conclusions for WSAN mobility scenarios

The following conclusions can be drawn for the WSAN mobility scenarios:

- Support for moving end-nodes between compatible VSNs and SSNs is feasible when all WSANs are controlled by one party (e.g. using [89, 36]). When multiple parties are involved, these WSANs are likely to use different encryption keys or protocols. For more flexibility, the end-nodes could be equipped with multiple keys so that they can operate in all WSANs that they have keys for. The downside of this is that the network keys could potentially be obtained from each end-node, so therefore the encryption should preferably work such that the encryption key only makes it possible to send something towards the gateway, not to decrypt all WSAN traffic. This can be accomplished by encrypting with the public key of the receiving gateway or the application. When using multiple applications, a group key could be used for the applications or the WSAN gateway (or its IPG) could do the encryption. In the latter case, traffic from the gateway to applications can then be encrypted separately.

- In order to reduce interference from overlapping WSANs, the moving one could adapt its wireless resources before the overlap, e.g. when similar GPS coordinates are detected.

- In order to reduce interference from overlapping compatible WSANs, the moving one could turn off its gateway [36] or change to intermediate node mode.

- In order to avoid end-nodes of compatible WSANs to move between one another, they can use different network encryption keys so that only nodes that have both keys can move to the other WSAN, and choose when changing WSAN is most appropriate.

- As discussed, merging SSN and BSN directly proves troublesome, especially for obtaining SSN measurements from nodes that often sleep. It is therefore more practical to merge BSN and SSN data at the application layer.

- Encryption needs to be in place when BSN nodes send privacy-related information, else foreign objects can store and forward it.

- WSAN protocols should be robust against foreign protocols, in order coexist with other WSANs that use the same wireless resources.

3.4 Conclusion

This chapter described the mobility of multimedia sessions and their parts and the mobility of and across sensor networks.

We proposed mobility management for multimedia sessions across heterogeneous networks using SIP and compared that with the usage of MIP. We noticed large delays when switching to lower-bandwidth networks, and needed to trim the session bandwidth before the switch. Therefore we recommend to send an *about-to-switch* event some time before the switch so that the bandwidth can be reduced before the switch. We also noticed that session control messages were lost in case of network congestion, therefore we also recommend to give priority to control messages such as SIP and MIP signalling.

We compared user-initiated approaches to distribute a multimedia session over multiple devices and proposed a network-initiated method that can be integrated with an existing user-initiated approach. With a prototype we validated that the method works in practice and we expect that a full fledged network-side box that implements this method can offer transfer of the multimedia streams to other devices without compromising the stream continuity too much.

We analysed mobility of WSANs in logistic and person monitoring scenarios. We determined that non-symmetric encryption is desirable to support both overlapping WSANs and controlled movement of nodes between WSANs. Furthermore, early detection of nearby WSANs could be used to prevent severe interference between them.

In the Chapter 4 we will analyse sharing of mobile resources.

Chapter 4

Sharing

In todays networks a number of resources can be shared among applications, namely: network segments such as wireless access; data such as multimedia streams and messages; and management such as configuration and actuation. Sharing among applications is often dynamic in the sense that each application can start and stop utilizing the resource at any time and other factors may constrain the availability of the resource at any time. Because of these dynamics and constraints, there are a number of challenges with respect to sharing. In this chapter we focus on the challenges for sharing media streams and Wireless Sensor and Actuator Networks (WSANs) between devices and applications, within and across network segments.

Dynamic sharing of media streams is constrained by the available bandwidth in each network segment that these media streams cross. This available bandwidth can fluctuate because of the dynamics of the access medium and the dynamics of other traffic that travels in the same network segment. In 2001 [114], see Section 4.1, we proposed a bandwidth distribution mechanism for shared medium networks that utilizes both real-time medium characteristics and feed-forward control mechanisms.

Efficient sharing of media streams from one endpoint to others located in multiple network segments is constrained by the available bandwidth in these network segments and the popularity of media streams across each network segment. Additionally, network segments may support broadcasting, multicasting and unicasting media streams. In 2008 [153], see Section 4.2, we proposed a method to dynamically group media streams in the same network segment to multiple destinations into multicast or broadcast streams depending on the ca-

pabilities and resources in that network segment. This method was initially developed for downstream content, but is agnostic from where the content streams originate. Therefore, it could also be used to efficiently distribute realtime content from an end-user device to other end-users.

Real-time WSAN messaging between WSANs and multiple applications is constrained by the reachability of, and access control on, the WSAN gateway and the applications, and latency along its path. Furthermore, secured transmission is often a requirement for both privacy and remote access reasons, and the sampling rate of each sensor can vary from slow (e.g. temperature every 15 minutes, or only when a threshold of say 30 degrees is reached) to fast (e.g. 200Hz raw 3D motion capturing). In 2011 [36, 34], see Section 4.3, we analysed different methods for shared use of WSANs by multiple remote applications/services while allowing remote configuration and maintenance of its nodes. Example applications are monitoring WSANs, maintaining WSANs, monitoring specific sensor types or specific nodes, and actuation on WSANs nodes. Finally we propose and analyse the efficiency of a method for shared WSANs usage compared to web-based approaches.

4.1 QoS for Broadband Wireless and Wired Access

Wireless LAN (WLAN), Cable, fiber and any of various Digital Subscriber Line technologies (xDSL) are amongst the most popular broadband access technologies in use nowadays. In these technologies, the transport capacity provided at the link layer is non-deterministically shared by multiple streams generated by several applications. These streams may have different or incompatible characteristics (e.g., one stream may contain bursty best-effort traffic generated by file transfer, another stream carries Quality of Service (QoS)-traffic generated by video-on-demand). To accommodate admitted QoS traffic in a fluctuating available bandwidth and to protect it from bursty traffic, all traffic must be regulated. This section describes a bandwidth-distribution mechanism for broadband access technologies that uses real-time characteristics of both the active-medium-sensing and the feed-forward control mechanisms. To validate this mechanism, two prototypes are developed, one based on wireless, the other on wired shared media. These prototypes employ legacy network elements without intrinsic QoS capabilities. Only the wireless prototype is described here, the wired one is available in [114]. We present the results of tests with the wireless

prototype and draw conclusions from our work.

4.1.1 Introduction

When there is sharing of physical and access media and fluctuating bandwidth availability, over-provisioning and prioritization are unsatisfactory. This will be shown in the course of this section. We focus on situations in which:

- Over-provisioning is not an option,
- It is impossible or undesirable to use the layer 2 prioritization mechanisms offered by shared-medium access technologies,
- The QoS mechanisms in use (often distributed prioritisation) cannot guarantee the requested QoS to the applications,
- The stochastic nature of the traffic causes overload in both the access network and the queues of the priority classes of the layer 2 prioritization mechanisms.

The remainder of this section will present our approach to construct the prototype that supports admission of QoS-enabled sessions to broadband access networks based on shared media. The requirements for this approach are based on the idea that the solutions it provides should:

- Allow QoS-enabled services to reserve bandwidth in the access network.
- Protect higher-priority (i.e., QoS) traffic from lower-priority (i.e., best-effort) traffic.
- Provide safeguards against bandwidth fluctuations for higher-priority (i.e., QoS) traffic.
- Support legacy hardware.
- Allow the use of complementary prioritization QoS mechanisms.
- Incorporate appropriate standards into the solution.

4.1.1.1 Overview of different QoS models

Network QoS mechanisms adhere to either the reservation model or the prioritization model. The following two paragraphs offer brief descriptions of these models.

Reservation model for QoS One way to divide scarce resources is to allow the parties that use them to reserve them. The two main network resources that are available for QoS reservation are bandwidth and low-latency data paths. The Integrated Services Working Group of the Internet Engineering Task Force (IETF) has focused on mechanisms for reserving these resources [40]. Some protocols that adhere to the reservation model are the resource reservation protocol [41] (RSVP), multi-protocol label switching [125] (MPLS), Subnet Bandwidth Manager [161] (SBM) and Next Steps in Signaling (NSIS) Signalling Layer Protocol [97] (NSLP).

Prioritization model for QoS Data prioritization is another way of providing QoS [38]. It is complementary to bandwidth reservation in most network contexts. This type of QoS – sometimes referred to as class of service (CoS) QoS, provides QoS by treating higher-priority packets better than lower-priority ones. Because it handles aggregated rather than separate flows, this type of QoS enables looser prioritization of traffic than reservation-based QoS. Some protocols that adhere to the prioritization model are: Differentiated Services [38] (DiffServ), IEEE 802.1D annex H [3], and IEEE 802.11e [148]. There are two different types of prioritization: centrally controlled and stochastically distributed. With centrally controlled prioritization, higher-priority traffic can be allowed to use more timeslots and more bandwidth per end node than lower-priority traffic. With stochastically distributed prioritization, higher-priority traffic has a greater chance of being sent than lower-priority traffic; however, because stochastically distributed prioritization is stochastic, there is no guarantee that higher-priority traffic will always get through first, especially when there is a lot of lower-priority traffic. In both types of prioritization, best-effort traffic usually gets the lowest priority, and there is generally no limit to the amount of best-effort traffic that applications can try to send (i.e., there is no shaping). Furthermore, the relative weights given to higher-priority traffic, optimization of overall throughput, and fairness are generally the same in all end nodes. There are situations (e.g., one endpoint uses high-priority traffic, and 100 others use best-effort traffic) in which such weighting may not be ideal. For prioritization to work under such conditions, each node would have to know how much traffic other nodes generate.

4.1.1.2 Synergy between the Reservation and Prioritization model

The reservation and prioritization QoS mechanisms have primarily been deployed in disparate network segments, as illustrated in Figure 2.2 on page 14.

For example, traffic from access networks is aggregated at the edges, which makes the edge more suitable for prioritization mechanisms (e.g. DiffServ). Typically, higher priority classes are assigned to packets travelling on low latency and low jitter data paths. These aggregated traffic streams travel through the core network using reservation mechanisms (e.g. MPLS in conjunction with RSVP traffic engineering [27] (RSVP-TE)). Neither the statistical aggregation principle employed in the edge, nor the semi-static reservations of aggregate tunnels employed in the core are of use in the access network. In fact, the lower order of granularity of the traffic load in the access network means that any prioritization mechanisms in this network segment must rely heavily on reservation mechanisms; otherwise, the scarce resources in each priority class will be depleted in high-load situations by the overloading of their queues. The rest of this section discusses the use of reservation based QoS mechanisms in broadband shared-media access networks.

4.1.2 QoS in broadband shared media access networks

Before we describe our approach we will discuss the most significant characteristics of the shared media we use as broadband access networks in our prototype.

Reservation and prioritization model QoS protocols have been designed to deliver data with certain QoS characteristics in a high capacity, packet switched, wire-line environment. Employing these protocols on broadband access networks is difficult, because of the physical characteristics of these networks and their use of a shared medium. Nevertheless, because these networks use shared media, appropriate QoS provisions are essential; otherwise, applications that generate high loads of best-effort traffic will consume too much bandwidth and will degrade the service levels of QoS-sensitive applications.

4.1.2.1 Characteristics of shared media

In shared media, numerous terminals and applications (the nodes in Figure 4.1) compete for shared bandwidth. In circuit-switched shared media bandwidth distribution is centrally controlled, but in packet-based shared media most elements of the network infrastructure do not contain the functionality required to collaborate in the end-to-end QoS management that would be necessary to control bandwidth distribution. Unlike the case in packet-switched networks, in which data in excess of a certain threshold can simply be dropped at a switch or router, all data sent on shared media use part of the available bandwidth. Because there are no commonly available bandwidth-distribution mechanisms for

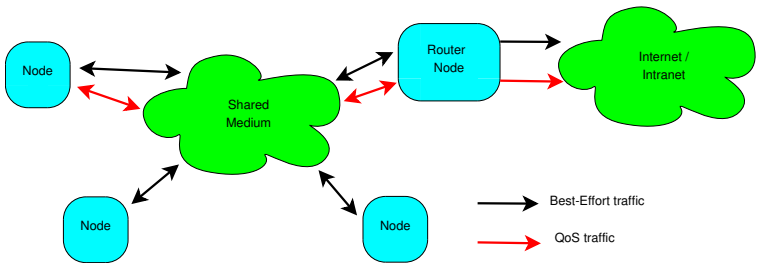


Figure 4.1: Traffic of multiple nodes in a shared medium access network

shared media, it is currently impossible to offer a high guarantee for QoS traffic or to divide the available bandwidth between all nodes accessing the media fairly.

In shared media access networks a router node connects the medium to an external network, such as the Internet or an Intranet. In most situations this router node uses the greater part of the available bandwidth, because most network traffic flows from the edge network to the access network. In a QoS enabled shared medium, this node also functions as a bridge between the QoS mechanisms of its shared medium and the QoS mechanisms of the edge network for which it provides connectivity.

4.1.2.2 Fluctuating available bandwidth

Another important characteristic of shared media is the fluctuation of available bandwidth, which is illustrated in Figure 4.2. Every access network has a theoretically defined amount of available bandwidth (1). Unfortunately, in practice the bandwidth that is actually available is less than this amount. The available bandwidth (3) is calculated as the difference between the theoretical bandwidth (1) minus the unavailable bandwidth part (2). The available bandwidth fluctuates due to:

- The characteristics of the physical medium
  - Electro-magnetic Interference (e,g. crosstalk, Signal to Noise Ratio (SNR), and shared frequency bands)
  - Atmospheric influences (e.g. rain, atmospheric humidity, and lightning)

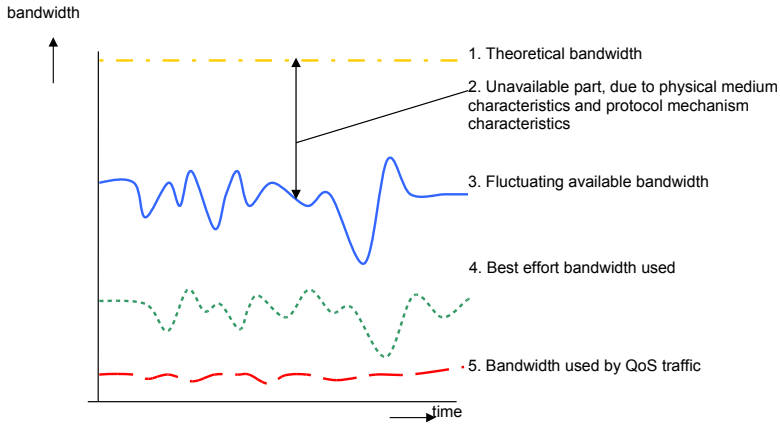


Figure 4.2: Fluctuating available bandwidth at medium level

- Characteristics of the protocol mechanism of shared media
  - Collisions, back-offs, overhead, slow starts
  - Dynamic channel selection, rate adaptation

Because the unavailable bandwidth part is inherently random, the available bandwidth is also random (i.e. it fluctuates unpredictably). This can be disastrous for QoS-sensitive applications. The bandwidth used by QoS traffic (5) is an adaptive and configurable threshold in our QoS-enabled admission control mechanisms. To maximize the probability that QoS bandwidth will be available, the maximum amount of QoS traffic to be admitted should be configured so as to leave a safety margin, as a protection against the minima of the fluctuating available bandwidth. The bandwidth that can be allocated to best effort traffic (4) represents the difference between the fluctuating available bandwidth (3), and the bandwidth used by QoS traffic (5).

The nature and the range of bandwidth fluctuation are determined primarily by the access network technology used. The following subsection discusses the most significant factors contributing to bandwidth fluctuation in the WLAN access technologies.

4.1.2.3 Bandwidth fluctuation in WLAN access technology

A wireless medium has a much higher Bit Error Rate (BER) than a wired medium of comparable bandwidth. The higher BER is caused by the characteristics of the wireless medium (e.g. path loss, shadowing, multipath fading, interference, and hidden terminals).

The fluctuation of the available bandwidth for a terminal in WLAN is related to the SNR measured at the terminal. When the SNR drops, the BER will probably increase and the number of packet retransmissions will also increase, effectively reducing the available bandwidth for all connected terminals. Because the SNR is random, the fluctuations of the available bandwidth will also be random, as illustrated in Figure 4.2 (3).

A back-off mechanism is used within a WLAN to avoid collisions as much as possible. Each collision that does occur wastes bandwidth (2). There are ways to provide central control over the timeslots in which remote nodes are allowed to send data, but for various reasons this mechanism is not widely implemented in WLAN cards.

WLAN networks, with the exception of ad-hoc networks, have a relay-node known as an Wireless LAN Access Point (AP). All traffic sent by the nodes in a WLAN network, including inter-node traffic, must first be sent to the AP. The AP then forwards the traffic to its destination. Because the relay in WLAN is done at the link layer it cannot be controlled without changing the network devices, which would be technology-intrusive. Also, due to the nature of this relay mechanism, inter-node packets occupy the medium twice, which must be taken into account in bandwidth calculations.

IEEE 802.11e proposes a prioritization mechanism for QoS management on WLAN [148]. Assigning different priority classes to different kinds of traffic would solve part of the QoS problem, but QoS still cannot be guaranteed when there is excessive traffic in one of the priority classes.

4.1.3 An admission control mechanism for QoS traffic

Our approach to support QoS in broadband wireless and wired access is to create an access control mechanism that provides an admission control function for QoS traffic. However, the observations we have made concerning the characteristics of shared media lead to the conclusion that there are situations in which it is not possible to guarantee QoS effectively without an access control mechanism for lower-priority (i.e. best-effort) traffic. In other words, the total traffic output of each node, which consists of both admitted QoS traffic and

best-effort traffic, must be controlled by an access control mechanism at the source node. The input to this access control mechanism consists of static and dynamic medium characteristics, Media Access Control (MAC) protocol characteristics, and real-time information about required bandwidth. All parameters can be either measured or provided manually. Furthermore, the system can be configured by parameters that allow several operating preferences, such as the balance between minimal bandwidth waste and a high safety margin for reserved QoS bandwidth, or the degree of fairness in bandwidth distribution among nodes and traffic classes.

We have defined and analysed an access control mechanism and implemented this in a prototype that supports both managed QoS and best-effort traffic in a shared medium. The requirements for this WLAN prototype were:

- Be non-intrusive to the network hardware, the drivers and the operating system
- Not be biased in favour of any network technology vendor
- Not be biased in favour of any Operating System (OS)
- Be able to support legacy applications without modifications to the applications themselves

By integrating and combining existing mechanisms and network devices the prototype provides solutions to the QoS problems we have discussed, while meeting these requirements. In this section we first describe the operation of the admission control mechanism and then its architecture.

4.1.3.1 Operation of the Admission Control mechanism

The QoS-enabled admission control mechanism that we have developed is illustrated in Figure 4.3 and explained below.

Figure 4.3a depicts the available bandwidth for QoS reservations. This available bandwidth is configurable and is mainly determined by the dynamics of the total available shared medium bandwidth. Figure 4.3b shows a high-quality video QoS service being added successfully. In Figure 4.3c and 4.3d, additional services are added. In Figure 4.3e, a node attempts to add another high-quality video service, but is denied permission by the admission control mechanism because of insufficient bandwidth. In Figure 4.3f, a low-quality video QoS service is added successfully, while the admission of the high-quality video QoS service is refused.

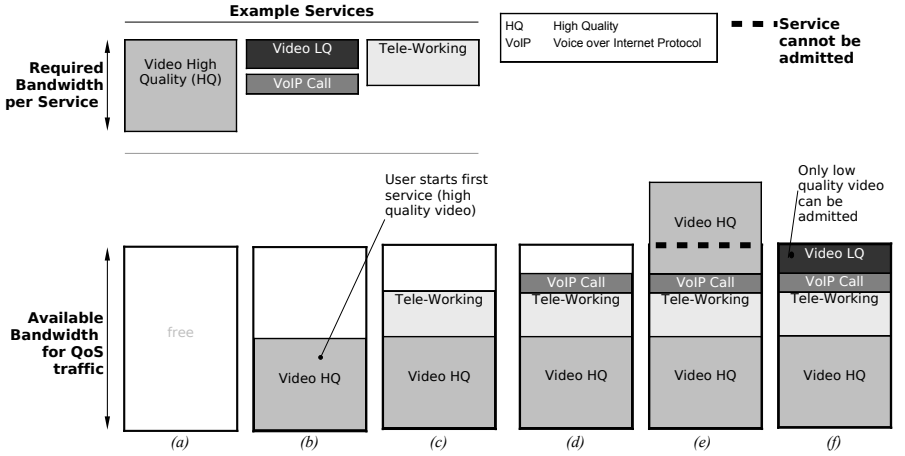


Figure 4.3: QoS enabled admission control

Best-effort traffic is controlled by distributing the remaining shared medium bandwidth among the best-effort traffic sources according to their bandwidth needs. These needs are measured in real-time in the specific WLAN prototype implementation, as will be discussed in the following paragraphs.

4.1.3.2 Architecture of the Admission Control Mechanism

The architecture our admission control mechanism is illustrated in Figures 4.4 and 4.5. Figure 4.4 shows where controller and regulators are positioned in the network: both are technology neutral. Figure 4.5 shows some details of the regulator, which shapes the best effort traffic of a node according to the traffic shaping parameters sent by the controller for this node. The main parameter is the regulator value that bounds the allowed bandwidth for best-effort traffic. It is important to note that our solution does not affect regular QoS packet forwarding. Our solution (with its controller and regulators) can be characterized as a feed-forward control system in which the amount of best-effort traffic occupying the network is controlled. It is based on the following control engineering [53] concepts:

- *Responsiveness* The regulator values should respond quickly to changing circumstances. It should be possible, as in our tests, to configure the responsiveness of the control system in such a manner that the regulators

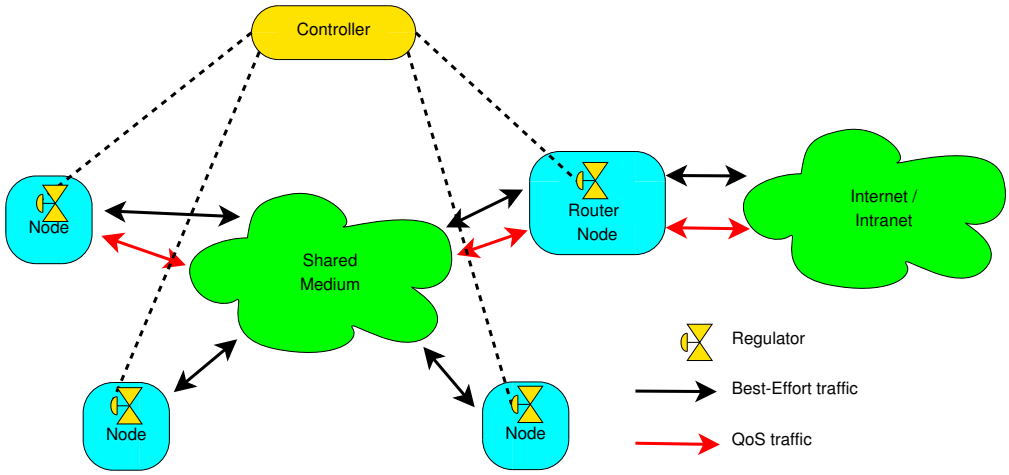


Figure 4.4: A control system for a shared medium access network

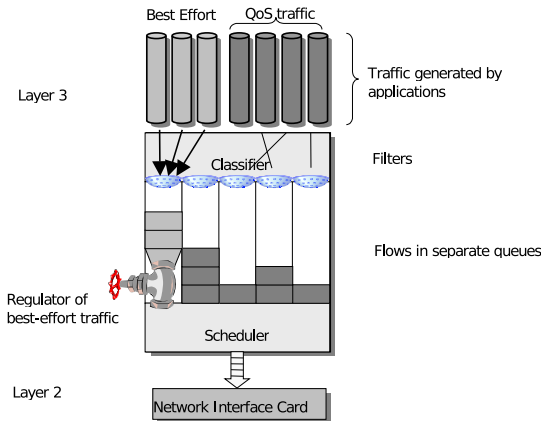


Figure 4.5: A regulator of best-effort traffic

reach a percentage (e.g., 90%) of their target values within 1 second. Doing so validates the subjective experience of the users that the system is indeed responsive. Furthermore, the minimum regulator value should not be zero; rather, it should be set to a small value that allows terminals to send and receive best-effort traffic with small bandwidth requirements immediately. If necessary, the regulator should be able to increase the value, allowing more best-effort traffic to be sent. These conditions enable increased responsiveness.

- *Steadiness* It should be possible to manage the control system. In order to do this, some configurable and adaptive mechanism could be applied (e.g., filters) to compensate for forces that could cause uncontrollable oscillations. This mechanism should be designed in such a way that the impact of regulator dynamics and the responsiveness requirements of the control system are brought into harmony.

- *Fairness* All users should be treated equally (unless otherwise configured); no starvation should be possible (i.e. users that do not get any bandwidth). All users who have the same transmission conditions (e.g. an equal amount of data to send and the same SNR) should have the same regulator value. Furthermore, the architecture of the control system should be able to adopt other definitions of fairness.

The input data for the control system algorithm are the real-time fluctuations of available bandwidth, the minimum regulator value, and the number of source nodes in the shared medium. The amount of bandwidth reserved for QoS traffic is subtracted from the fluctuating available bandwidth. The result is the bandwidth to be distributed for the best-effort traffic of the nodes.

The steps of the control algorithm we use can be described as follows:

1. Each node regularly sends information about (a) the best-effort traffic characteristics, (b) network quality that has been measured during a configurable interval, and (c) active QoS reservations.

2. The controller calculates how much best-effort bandwidth can be allocated for each individual node for the next interval, based on the inputs of step 1, and in accordance with the configured balance of the control-engineering concepts discussed above.

3. This bandwidth is divided by the access control mechanism in accordance with both the users' needs and their experienced network quality, as measured in step 1. The new regulator values are calculated.

4. Regulator values are sent from controller to the users' regulators.

5. Steps (1-4) are repeated endlessly.

The frequency of this control loop is determined by the responsiveness requirements of the deployment situations. In our prototype, the interval was configured to be about 1 second.

4.1.4 WLAN Prototype

Each node in the prototype is equipped with a regulator that controls the amount of best effort traffic that the node is allowed to send. The regulators are positioned between layers 2 and 3 of the protocol stack. The throughput control of a regulator is described by token-bucket parameters. The Microsoft Windows®traffic control application programming interface (API) and a similar package (based on kernel sockets) for the Linux®operating system are used for the throughput control implementations of the regulators in our prototype. Bandwidth for QoS traffic is reserved and handled by RSVP. The regulators are centrally operated by a controller, which has real-time knowledge about the amount of QoS traffic and the fluctuating available network bandwidth. The controller calculates the bandwidth available for best-effort traffic, distributes it among the active nodes, and controls the regulators accordingly. In this process, an active node can acquire all the available bandwidth for best-effort bursts, as long as other nodes are inactive. If multiple nodes are active, the control mechanism will distribute the available bandwidth fairly. The controller takes into account the fact that data sent from one wireless node connected to an AP to another wireless node connected to the same AP travels the wireless medium twice, and so consumes twice the amount of bandwidth. We will call such traffic **internal traffic**, and we will call other traffic to and from the AP **external traffic**.

In order to verify our WLAN approach, we performed a number of tests; we describe a few of them in the following subsections.

4.1.4.1 WLAN Test setup

All tests were performed using the basic configuration shown in Figure 4.6. The number of terminals varied in testing, the minimum number was 2. All terminals have a regulator and the terminal physically connected to the AP hosts the controller. We implemented the controller software in the Java programming

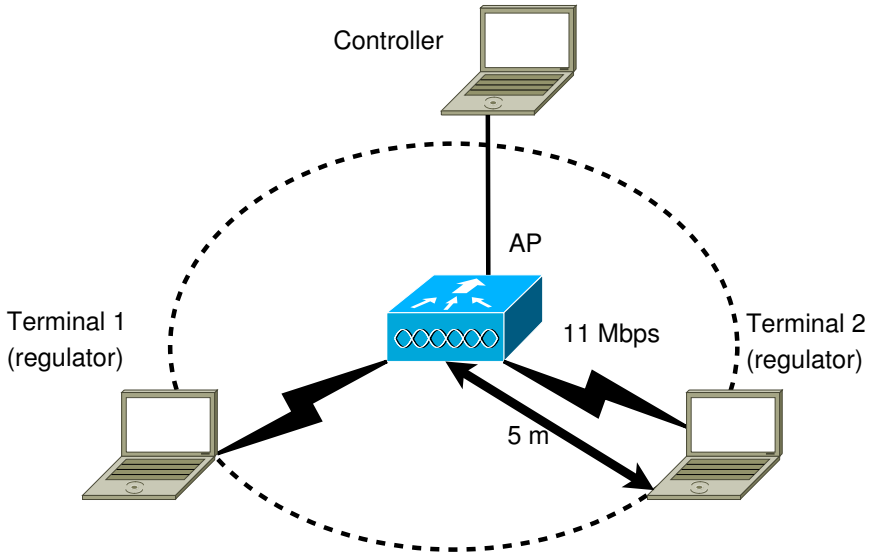


Figure 4.6: Wireless LAN prototype

language and the regulators in C++. If not otherwise specified, the distance between each laptop and the AP was about 5 meters.

4.1.4.2 WLAN Test results

All the tests depend on knowing how much of the theoretical maximum bandwidth of 11 megabits per second (Mbps) is available. Terminals sent as much data as possible and the throughput was measured. The bandwidth varied around 800 Kilobytes per second (KBps) depending on the number of used terminals in the experiment. In case of two terminals, the measured throughput was around 830 KBps, which is 60% of the theoretical maximum. We then used this figure as the maximum available bandwidth throughout the tests. Therefore, in all the tests, the maximum available bandwidth, which must be split between QoS and best-effort traffic, is equal to 60% of the theoretical bandwidth. The exact relationship between the number of terminals and the available bandwidth can be found in [133].

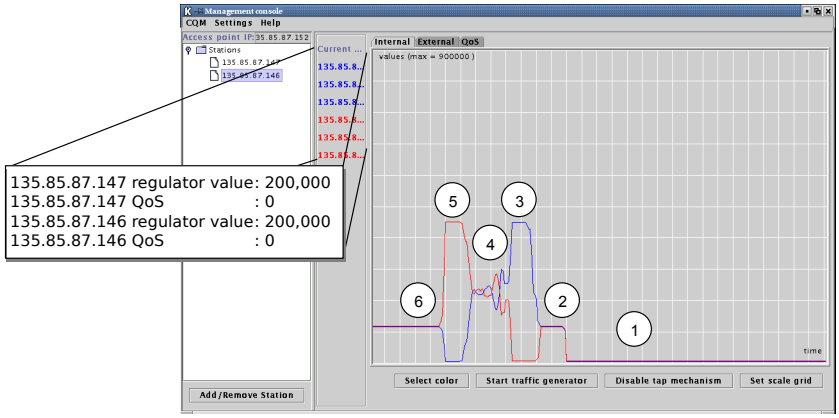


Figure 4.7: WLAN test 1 results

Test 1 This test evaluates the responsiveness, steadiness and fairness requirements. When two terminals try to send best-effort traffic with the same characteristics, and the SNR measured at the terminals is approximately equal, then the available best-effort bandwidth is divided equally between them.

Figure 4.7 shows a screenshot of the Graphical User Interface (GUI) of the system. The graph shows the regulator values for internal best-effort traffic. The horizontal grid lines are one tenth of the maximum, which is 900 KBps. The vertical grid lines are 5 seconds apart. The dark green graph line shows the regulator values for terminal 1, while the light green line shows the regulator values for terminal 2. The time line in the graph is from right to left, so the part of the graph numbered 1 represents the oldest regulator values.

The following events can be distinguished in Figure 4.7:

1. The control system is disabled, which means the prototype runs in a standard best-effort traffic configuration. In this situation, video streams will be disrupted by high-load best-effort traffic.

2. The control system is enabled. No traffic is transmitted by the terminals. This causes the control system to distribute the available best-effort bandwidth to the regulators. Both regulators are configured to a quarter of the maximum available bandwidth for internal traffic and a quarter of the maximum available bandwidth for external traffic. This accounts for the total amount of available best-effort bandwidth.

3. Terminal 1 starts transmitting internal User Datagram Protocol (UDP) traffic. The controller configures the internal traffic regulator for terminal 1 to the maximum amount of available best-effort bandwidth, increasing the traffic by a factor of 4. The internal traffic regulator for terminal 2 is configured to zero. The external traffic regulators for both terminals are also configured to zero. The responsiveness of the system is around 1 second, as can be seen in the figure.

4. Terminal 2 also starts transmitting internal UDP traffic. The available best-effort bandwidth is divided equally between the internal traffic regulators for the two terminals. The regulator values become steady in a few seconds. In contrast to step 1, video streams are no longer disrupted by high-load best-effort traffic.

5. Terminal 1 stops transmitting. Terminal 2 gets all the available best-effort bandwidth for its internal UDP traffic.

6. Terminal 2 stops transmitting as well. Because neither terminal is transmitting, all regulators have the same value. The system is in the same state it was in in step 2.

WLAN Test 2 This test, the results of which are depicted in Figure 4.8, shows what happens when two terminals transmit different amounts of traffic. As in the previous test, the regulator values for terminal 1 are indicated by the dark green line, the regulator values for terminal 2 by the light green line.

The following events can be distinguished in Figure 4.8:

1. The control system is disabled (see step 1 of test 1).

2. The control system is enabled, no traffic is transmitted. All regulators have the same value.

3. Terminal 1 starts to transmit external UDP traffic at a rate of 256 KBps.

4. Terminal 2 starts to transmit external UDP traffic at a rate of 128 KBps. Terminal 2 gets less bandwidth than terminal 1. However, the regulators are not configured to the precise bandwidth requirements of the terminals, which totals 384 KBps. Since more best-effort bandwidth is available (i.e. 900 KBps, the controller distributes the remaining 516 KBps best-effort bandwidth evenly over the regulators as can be seen in Figure 4.8.

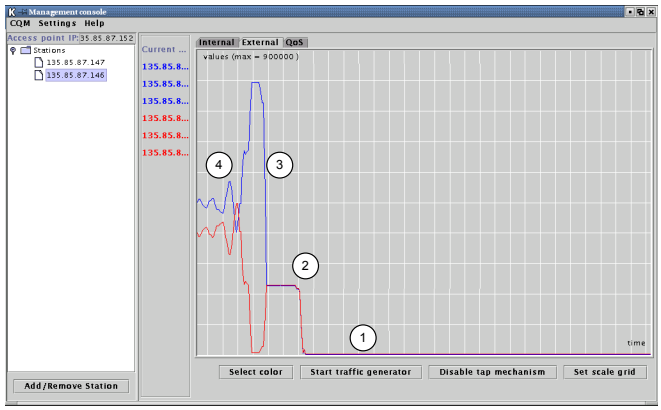


Figure 4.8: WLAN test 2 results

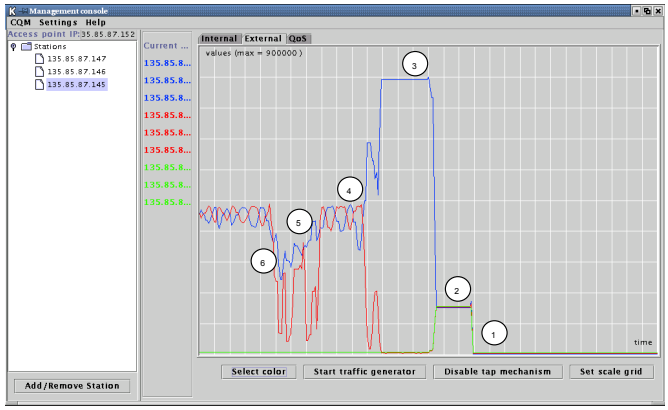


Figure 4.9: WLAN test 3 results

WLAN Test 3 The third test, the results of which are depicted in Figure 4.9, shows the impact of the distance from a terminal to the AP on the available bandwidth for that terminal. As in the previous test, the regulator values for terminal 1 are indicated by the dark green line, the regulator values for terminal 2 by the light green line. When a terminal moves away from the AP, its SNR decreases. This means that it is more difficult to separate the received signal from the background noise. Also, as the distance increases, the BER increases, more retransmissions are necessary at the MAC level [2], and the throughput at the Internet Protocol (IP) level drops. The results of extensive testing of the relationship between the SNR and the regulator value can be found in [133].

The following events can be distinguished in Figure 4.9:

1. The control system is disabled (as in step 1 of tests 1 and 2).
2. The control system is enabled. No traffic is transmitted. All regulators have the same value
3. The first terminal starts to transmit external UDP traffic.
4. The second terminal starts to transmit external UDP traffic.
5. The second terminal is placed 20 meter from the AP, while the first terminal is moved slowly from its initial position 5 meters from the AP to a position 10 meters from the AP.
6. Both terminals are moved back to their initial positions.

In the WLAN tests, the minimum amount of best-effort traffic that could be transmitted by a node (i.e. the minimum regulator value) is set to 4000 Bytes per second (Bps). So a terminal could send up to 4000 Bps of best effort traffic, without the value of the regulator having to be increased. This further improved the responsiveness of the system.

4.1.5 Conclusions

This section has addressed the lack of QoS in broadband shared-media access networks. Our approach supports QoS for multiple sessions per node by incorporating a QoS mechanism that uses admission control based on reservations. Our prototype tests have shown a successful implementation of this mechanism in a WLAN environment. In our prototype, QoS session setup is denied when the session cannot be accommodated in the available bandwidth. Furthermore,

services with different traffic characteristics can be used simultaneously, without quality degradation. We have also seen that the shaping functionality of the regulators in the WLAN source nodes protect active QoS sessions from otherwise uncontrolled best-effort traffic. Combined with edge- and core-network QoS capabilities, the presented reservation model QoS solution offers a straightforward mechanism that can be applied to any transport protocol, as long as the start-up and termination of all QoS sessions in the access network can be determined by the controller. Furthermore, this approach is complementary to prioritization model QoS mechanisms in the same broadband shared-medium access network.

4.2 Efficient Personalized Content Distribution

Services like telecommunication, messaging, radio/television broadcast, and web browsing are increasingly using the same IP based transport technology. Converged IP networks will enable easier blending and personalization of those services.

For popular content, technologies like caching, broadcasting and multicasting aim to increase network efficiency, but also introduce extra complexity and bandwidth overhead. Additionally, the sheer amount of available content, and the increasing flexibility for the end user to request content at any time, at any place, makes it increasingly difficult to determine content popularity in order to predict the efficiency of using content distribution techniques.

This section describes a model for wireless networks to use network characteristics and a prediction of the popularity distribution to calculate the optimal combination of unicast and broadcast techniques when offering a number of media channels to the end user. Furthermore an approach is described to optimize personalized content distribution efficiency in converged IP networks utilizing efficiency calculations with this model.

4.2.1 Introduction

Content distribution describes the delivery of digital media **content** such as audio or video over a delivery medium such as broadcasting or the Internet. For distributing popular content to multiple endpoints, content distribution techniques such as multicast, broadcast, caching and peer-to-peer [24] can improve network efficiency by reducing the used network capacity when compared with multiple connections from the content source [11, 12, 142].

Personalization and customization tailors services based on personal details or characteristics [98]. Future television-like services [98] are likely to offer personalization by enhancing or replacing parts of commonly distributed content (Television (TV) channels) with additional content aimed specific at a particular user or user group, such as: added overlay elements such as text, graphics, pictures, animations or video clips and adding or replacing audio- and/or video-specific segments.

Content distribution techniques mainly improve efficiency for popular content. However, personalization and almost infinite content choice make it difficult to determine content popularity at a specific moment in time. This section describes a content distribution efficiency model for wireless networks, and proposes a dynamic mechanism to provide efficient personalized content distribution in each network segment of converged IP networks, i.e. the mechanism will dynamically switch individual content streams that become more popular in a certain network segment to broadcast modes. Since the personalized part of a session is less likely to be popular in one network segment, it will likely be unicasted, while a generic content stream in the same session is more likely to be popular enough to be switched to broadcast or multicast mode. The described efficiency model can be used to determine when to switch.

4.2.2 Efficiency of content distribution in wireless networks

For several wireless technologies, special techniques are available for content distribution. Examples are broadcast for WLAN, and Multimedia Broadcast Multicast Service (MBMS) for Universal Mobile Telecommunications System (UMTS) networks. These broadcast or multicast techniques offer different capacity characteristics than their unicast variants in order to ensure reasonable signal reception at end points with the worst reception characteristics. For instance MBMS cannot use specific characteristics of a point-point connection to optimize the offered data rate like in High-Speed Downlink Packet Access independent, optimized personal services (HSDPA).

Table 4.1 gives an overview of standardized wireless technologies in terms of capacity and cell size. The given capabilities [5, 85, 90, 4, 6, 61] are generalized and theoretical: In practice there are many interdependencies. The cell capacity is often related to the desired cell diameter, using higher transmission power or lower transmission rates needed to serve more distant user equipment. Furthermore, the level of urbanization and the number of active users in a cell results in additional transmission overhead per user. Also the cell topology will have

	WLAN	UMTS	HSDPA	WiMAX	DVB-H
Typical cell diameter	150m	0.5 km	0.4-0.8 km	1-5 km, up to 25 km	up to 40 km (8K mode)
Typical cell capacity	35 Mb/s (802.11g)	1 Mb/s	10 Mb/s	15-40 Mb/s	5-11 Mb/s

Table 4.1: Overview of standardized wireless access technologies (2008)

an influence. In this scope a basic cell topology with non-overlapping cells is considered, but also other lay-outs are possible, including using overlapping cells of the same or different size. For instance, when smaller cells are primarily used for unicasts and larger cells primarily for broadcast, the efficiency may improve compared to usage of non-overlapping cells only. Also hybrid solutions which combine different complementary technologies (such as Digital Video Broadcast - Handheld (DVB-H) and HSDPA) can be considered.

In the next paragraphs we describe the popularity of content and an efficiency model to use stream popularity to determine how content can be transferred more efficiently.

4.2.2.1 Popularity of content

When the end user can choose content from several sources (for instance, web sites, or TV channels) it is unavoidable that there is a difference in popularity between different content items. Today the end user can choose between an almost infinite number of content sources, and the 'long tail' of less popular content becomes more important [23]. The Zipf's law is widely used to model the popularity ranking of words in text, popularity of books, or even television channels [140], and is used in our model.

4.2.2.2 Efficiency model

For less popular channels optimizing the network efficiency is mainly a challenge near the content source. The more popular a channel gets, the more performance can be gained near the end points. The current version of the efficiency model applies only to wireless access networks. In other parts of the network, such as the core network, dual streams with equal content already justifies the application of content distribution techniques, allowing a much simpler efficiency model. An example is switching to broadcast mode whenever there are two or more equal content streams in that network segment.

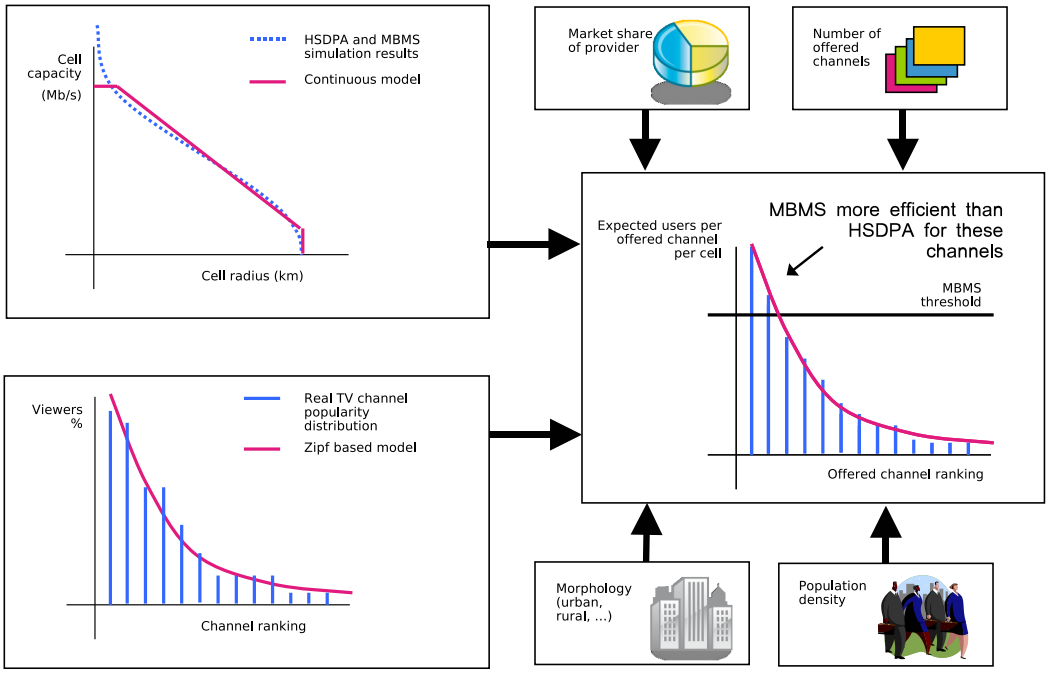


Figure 4.10: Overview of the efficiency model

To predict the required cell capacity for introducing new content channels in a wireless network, an efficiency model was made that calculates the cell capacity for the most optimal combination of MBMS and HSDPA transmission channels for a given number of channels at a certain data rate. This model may later be extended for other technologies, like DVB-H and Worldwide Interoperability for Microwave Access (WiMAX).

Figure 4.10 shows a simplified overview of the efficiency model, which currently consists of:

- A popularity distribution model based on enhanced Zipf-Mandelbrot approximation of popularity distribution of a number of offered channels such as television channels. The constants in the formula are currently se-

lected such that the model correlates with a selected real life distribution of viewers to 21 Dutch television channels [149] based on viewer ratings on an average evening in June 2007[1]. This forms the basis for modelling popularity for a given number of offered channels at peak hour. Other approximations can be envisioned, for instance for modelling popularity distribution for mobile radio.

- Models for calculating the relation between the cell size and the cell capacity for HSDPA and MBMS. These models are simplified continuous approximations of simulation results, tailored for 4 different morphologies (dense urban, urban, sub urban and rural). The used simulation results were existing material from simulations performed at Alcatel-Lucent on a multi-user, multi-cell system simulator, which uses detailed modelling of the MAC and physical layer behaviour of the network and the User Equipments (UEs).

- Additional input factors are: population density, service provider market share, fraction of video-enabled UEs, a multiplication factor describing service pricing and appeal compared to television, network capacity usage expected besides the offered channels. The model assumes a uniform distribution of users.

Based on the input parameters (offered number of channels at certain data rate, cell radius, morphology, and calculated number of potential viewers in a cell) the model calculates how many viewers are needed for an MBMS channel to be more efficient in terms of capacity then offering duplicate HSDPA connections for each viewer. Also the number of different channels likely to be active in a cell and the total capacity for the service per cell are calculated.

The model can potentially be used for feasibility studies to determine possible dimensioning of current and future networks and group services, and can provide information to optimize the Broadcast Multicast Service Centre (BM-SC), the decision function in MBMS which authorizes and initiates MBMS Bearer Services and can be used to schedule and deliver MBMS transmissions [10].

Table 4.11 shows an example of output of the model calculating different situations for an operator with considerable market share offering a number of 128kb/s mobile TV channels using HSDPA and MBMS. Calculating optimized combinations of HSDPA and MBMS, the table shows that MBMS is only considered efficient in rare occasions: only when few channels are offered in more rural areas, MBMS seems to increase efficiency over using multiple HSDPA channels.

Number of offered channels: 4

	Dense Urban			Urban			Sub Urban			Rural		
Population Density (people/km²)	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA
100	0.98	0	4	1.09	0	4	1.26	0	4	1.26	0	4
200	0.77	0	4	0.84	0	4	0.89	0	4	0.89	1	3
1000	0.40	0	4	0.38	0	4	0.40	1	3	0.43	1	3
5000	0.18	0	4	0.18	1	3	0.19	1	3	0.19	1	3

Number of offered channels: 16

	Dense Urban			Urban			Sub Urban			Rural		
Population Density (people/km²)	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA
100	0.97	0	9	1.06	0	11	1.14	0	14	1.14	0	14
200	0.75	0	11	0.77	0	13	0.80	0	14	0.80	0	14
1000	0.35	0	14	0.36	0	14	0.36	0	14	0.36	0	14
5000	0.16	0	13	0.16	0	14	0.16	0	14	0.16	0	14

Number of offered channels: 64

	Dense Urban			Urban			Sub Urban			Rural		
Population Density (people/km²)	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA	cell radius (km)	Different channels distributed in cell MBMS	HSDPA
100				1.19	0	10	1.32	0	14	1.32	0	14
200	0.84	0	10	0.89	0	12	0.93	0	14	0.93	0	14
1000	0.41	0	14	0.41	0	14	0.41	0	14	0.41	0	14
5000	0.18	0	14	0.18	0	15	0.18	0	13	0.18	0	13

Figure 4.11: Sample model output for operator offering 128 kbps mobile TV channels using HSDPA and MBMS

In other occasions, due to optimization of HSDPA using the specific characteristics of a point-point connection, using multiple HSDPA connections instead of one MBMS channel is more efficient, even for the most popular offered media channel. Note that the cells are now considered to be for TV distribution only, in case of offering more services in the same cell (dedicated voice traffic, web browsing), the cells will need to be smaller, and MBMS will even be less likely to improve efficiency.

4.2.3 Application in converged IP networks

IP Multimedia Subsystem (IMS) and Internet Protocol television (IPTV) are gaining ground in the operator space for converged networks, however they are still operating separately. In IMS, the Session Initiation Protocol [31, 123] (SIP) is the main signalling protocol used for setting up, modifying and ending multimedia sessions [8]. This section shows how SIP in IMS could be used to form an efficient personalized content distribution overlay that can utilize multicast where useful and available in the network.

4.2.3.1 SIP to provide multimedia content

SIP is particularly suited for setting up multimedia sessions, and could therefore easily be used to set up multimedia sessions to content providers, complementary to the Real Time Streaming Protocol [136] (RTSP) that is used nowadays by content providers. A big benefit of using SIP [142] for setting up and controlling these sessions is the relative ease to modify content streams within them by exchanging updated session descriptions (via the Session Description Protocol [127] (SDP) [128]). This can be done both by session endpoints and by Back-to-back User Agents (B2BUAs) in the signalling path. These modifications allow among others:

- changing a stream from unicast to IP multicast,
- changing the transmission endpoint on a multi-homed device (e.g. switching a stream from UMTS to WLAN interface on the device) [124], and
- changing the endpoint of transmission (e.g. from mobile device to wall display) [13].
- changing the encoding of multimedia data.

SIP Application Servers (ASs) can be used in the SIP signalling path to provide services for sessions that are being set up or running. In our case we want to use such an AS to dynamically collapse equal session streams in a network segment into one in order to reduce network load.

4.2.3.2 Multimedia broadcast convergence

For converging IMS with broadcast media, an Application Server for Multimedia Broadcast Convergence (MBC-AS) is introduced, which is typically located in the IMS core. The role of the MBC-AS is to reduce network load by collapsing and/or caching equal content streams that travel through same network segment. For this purpose, the MBC-AS can re-direct content streams via SIP-enabled relays located in different network segments. These relays can be used to combine equal streams to different users into one where possible, cache streams for users that tune in later, and to transcode streams for users that do not support the codec. The MBC-AS could optionally be closer to or in the access network, i.e. near the Proxy CSCF (P-CSCF), to reduce switching time.

The MBC-AS can make network segment-specific decisions while content streams become more popular and as such control the content distribution overlay enabled by the relays. Moreover, this MBC-AS can enable switching a stream from unicast to IP multicast when useful (see the efficiency model in Section 4.2.2.2) and supported in a network segment by specifying a multicast address for that stream in the session description (in short SDP). Furthermore, this MBC-AS can potentially utilize all network paths through which a multi-homed terminal is reachable in order to optimize bandwidth and latency in a combination of network segments (relays would not add much latency since they do simple address translation).

So, the actual distribution of a stream can be based on: the network location(s) and number of receivers, the efficiency/cost of multicast/broadcast versus unicast per network segment, and context information from the network (e.g. predicted/inferred content popularity or network usage). The concept also reduces signalling to the content provider, i.e. MBC-AS could handle INVITEs that would utilize an existing relay on behalf of a content provider. The MBC-AS supports dynamic changes of individual streams in a multimedia session from unicast to multicast on the same or other access technologies and vice-versa using SIP re-INVITE messages.

The MBC-AS assumes future enhancement of IP multicast support in the answer/offer model for SIP to allow signalling between multicast senders and multicast receivers and vice-versa [33]. As a fall-back each SDP would have to

specify sending plus receiving for multicast streams, i.e. specify stream attribute "a=sendrecv" or nothing (since sendrecv is the default) in the SDP for that stream.

4.2.3.3 Architecture

Figure 4.12 shows two operator networks, an all-IP network with IMS core 1 on the right and an UMTS network with IMS core 2 on the left. Both operator networks have an MBC-AS in their IMS core, and a relay (or a proxy when caching non-realtime content) on the data plane. The figure shows the situation where three terminals (UE1, UE2 and UE3) receive the same real-time video stream in an optimized manner while receiving a personalized audio stream directly from a content provider: UE1 receives the video stream multicasted or unicasted by relay 1, UE2 and UE3 receive the video stream multicasted or unicasted by relay 2 which receives the video stream multicasted or unicasted from relay 1. The SDPs in the figure show the original for UEs and the content provider (providing ads, generic or specific content), as well as the modified SDPs that both MBC-ASs made dynamically to the multimedia sessions (utilizing the efficiency model and multicast availability) while more UEs requested the same video stream. Since MBC-AS is a B2BUA in the SIP signalling stream, relay 2 has an SDP for both relay 1 and the Content Provider side and the Content Provider has an SDP for both relay 2 and the UE(s) side. Note that the BM-SC could potentially make local decisions to use unicast or MBMS to the terminal(s), or be instructed by the MBC-AS based on the efficiency model. For IP network segments simpler efficiency models like more than N (e.g. 2) listeners are envisaged.

4.2.3.4 Personalized multimedia and targeted ads

Personalization can be offered directly by a content provider or by a separate application (e.g. a SIP AS). Since unicasted and multicasted streams from different sources can be combined in multimedia sessions, interactivity can be supported with an electronic menu guide, that is integrated within a multimedia stream or as a separately one, in which menu selection triggers adding, deleting or adapting streams in multimedia session. Additionally, the user profile (preferences, account settings) and user context (location, environment, activity, role, etc.) can trigger user-tailored set up and dynamic change of a multimedia session in order to integrate both targeted advertisements and content into a seamless multimedia experience while at the same time streaming the generic parts effi-

Figure 4.12: Multimedia broadcast convergence

ciently to multiple recipients and specific personalized parts directly using the MBC-AS described above. Such a specific personalized part can also be the last (or generated) infra-frame and subsequent frames of a generic Moving Picture Experts Group (MPEG) stream, such that the user can start enjoying content directly before being able to synchronize with the generic stream. So, the MBC-AS is an enabler for a whole range of efficient personalized content streaming services.

The different parts (specifically personalized, generic, and targeted ads) in the multimedia session can be synchronized since the same overall session ties all the parts together. Each change in the session (like insertion of targeted ad, or switch of stream from unicast to multicast) will invoke a re-INVITE to the SIP User Agent in the terminal and as such provide synchronization points. Optimizing the client-side streaming buffers in combination with specific start times per stream in the SDP is left for future study.

4.2.4 Conclusions and Recommendations

A model for content distribution in wireless access networks was described that uses popularity distribution functions, functions relating cell size and cell capacity and a number of input parameters such as the number of users per square km. Applications of the model include feasibility studies and network decision functions.

In the use case discussed, using MBMS only appears to increase efficiency over using only HSDPA in rural areas, and only when offering a limited number of 128kb/s TV channels. A hybrid approach with distribution via DVB-Terrestrial (DVB-T) combined with HSDPA seems in this case more economical, since large cells are more efficient for broadcasting popular content and small cells are more efficient for other traffic. Future improvements such as HSDPA broadcast, or 3rd Generation Partnership Project [8] (3GPP) Long Term Evolution (LTE) improvements to MBMS may enable substantially higher throughput for broadcast traffic over UMTS based access networks, and prove less inefficient compared to multiple unicast connections.

The amount of point to point traffic increases due to trends such as addition of personalization to distributed content, the long tail phenomena, peer to peer traffic and increasing complexity and size of web pages, and make it less obvious to decide which content distribution technique to use where, especially when considering the high efficiency of point-to-point HSDPA connections in contrast to broadcast overhead.

A flexible approach using SIP that blends general and targeted content

streams in multimedia sessions while enabling efficient distribution of streams per network segment could form a feasible solution utilizing the described efficiency model for the wireless access and simpler rules in the other parts. The described approach is primarily focused on real-time streams, handling on-demand and transcoding of streams is future work.

Future extensions of the efficiency model may include modelling the efficiency in other technologies like DVB-H and WiMAX and for hybrid approaches combining access networks of different technologies.

4.3 Shared usage of WSANs

This section analyses how WSANs can be remotely monitored, maintained, configured and used in applications.

A number of issues related to shared WSAN usage were described by Shu et all [139], and some solutions have been proposed for sharing WSANs [95, 73]. The purpose of the analysis of shared WSANs usage is to determine which sharing scheme can best be used with different numbers of attached IP applications, where both applications and WSANs can be mobile.

Section 4.3.1 describes and analyses different schemes for handling sharing of mobile WSANs, and determine the most promising scheme to be used for shared mobile WSANs. Subsequently, we will analyse the efficiency of the Ambient middleware compared to other messaging protocols.

4.3.1 Schemes for shared usage of mobile WSANs

In this Section schemes are analysed where multiple applications use the same WSAN data [95, 73, 139], while the IP gateway (IPG) of both the application and the WSAN can change IP address while moving. For handling mobility of a WSAN and its connected applications a number of options can be considered. The following properties are used for comparing them, a number of them originate from the scenarios analysis, others from deployment and complexity concerns:

- **multi-move**: is simultaneous moving of source and destination supported?
- **smart buffering**: can intelligent buffering be done for applications when connections fail? Alarm messages and recent measurements can better be sent first since they usually have higher priority than older measurements.
- **overhead**: is there inherent overhead in the approach?

- **duplication node**: where are messages destined for multiple recipients duplicated (or broadcasted)? Obviously, closer to the recipients is more efficient, especially when different recipients require different data rates [107]. Options are end-node, gateway, server, router, proxy or relay.
- **maturity**: is the scheme still in research or is it already available?
- **deployment needs**: what is necessary to deploy this on the current Internet?
- **access control**: who checks whether a destination is allowed to get the content? Depending on the number of destinations, the source may need to be taken out of the loop.
- **request method**: can application requests like configuration and actuation be transferred to the source using the methods of the mobility scheme or is an additional method required?

The combination of the properties overhead, duplication node, and access control give an indication of the scalability of a scheme. For instance when a scheme has much overhead, and duplication and access control are done at the source is it not very scalable. The scalability increase when access control and duplication can be done closer to the destinations, and when the overhead decreases.

The next paragraphs, we analyse the sharing support for different schemes for shared use of mobile WSANs. Some of these schemes originate from the mobility perspective, others are more geared towards sharing.

4.3.1.1 Sharing using IPv6 mobility

IPv6 over Low power Wireless Personal Access Networks [106] (6LoWPAN) turns the WSAN into an Internet Protocol version 6 (IPv6) network and addresses mobility of nodes with Mobile IP [116, 77] (MIP). This maintains reachability of all nodes in the WSAN when they move inside or across WSANs. However, WSAN nodes are often not reachable since they are sleeping to save energy. The 6LoWPAN gateway may then send additional information when a connection is broken because of sleeping duty-cycle. Furthermore, 6LoWPAN assumes the application will handle re-sending to each individual node in case of failure. 6LoWPAN uses Network Mobility [48, 115] (NEMO) for mobility of the complete WSAN. This means that the whole WSAN can change its point of attachment, since the network prefix of the WSAN has MIP support.

There are a number of issues with 6LoWPAN for WSANs:

- Traditionally WSAN nodes just send their readings towards the gateway, and an application can connect to the gateway to receive the sensor readings and for configuration. In 6LoWPAN, the gateway is an IPv6 router, and an IP application that is interested in the readings, needs to register its IP address with each individual node (unless multicast can be used as destination, and applications can join the multicast group). This makes the binding between the application and nodes very tight which hinders scalability.
- The burden of reaching sleeping nodes is placed on the IP application(s) that use them. Since the time window for sending messages to a WSAN node can be very small, this may be infeasible from remote application locations because of unpredictable latency on the path towards the WSAN. It is therefore advised to let the WSAN gateway handle reachability of nodes.
- 6LoWPAN requires both IPv6, and a Home Agent (HA) with support for NEMO. Neither of those are currently widely deployed.
- Every WSAN node will need to do access control for configuration and actuation from applications.
- When security is required, every WSAN node will need to do network or application layer encryption to secure the path towards the IP application, independent of data link layer security that may already be in place.
- When multiple applications require sensor information from the same node, that node needs to sent the information twice (unless there is multicast support), which doubles bandwidth both within the WSAN and its uplink.
- Transmission Control Protocol (TCP) connections are a bad match with dynamic WSAN nodes that are often sleeping and since packets may also be dropped because of congestion or because the node battery drained or the node moved outside range. It is often better to send a new measurement than to retry an old measurement that got dropped because of collisions.
- There are still numerous challenges related to security in 6LoWPAN [111], not to mention combining security with nodes that move between WSANs.

There are a number of things that the gateway could potentially handle transparently on behalf of the nodes when it uses packet inspection to pre-process requests towards nodes and responses from nodes:

- Access control on behalf of nodes.
- Buffer requests to sleeping nodes until they wake up.
- Handling interest of an application, for instance by using IP multicast.
- Replication of sensor readings to multiple applications
- Converting TCP connection towards a node to UDP packets, and injecting UDP packets from the node to an existing TCP connection towards an application.

Most of these options turn the gateway from a simple IPv6 router to a stateful router that requires deep packet inspection and making realtime packet modifications. Moreover, transparent network layer security with nodes will make many of these options impossible without sharing key material between nodes and the multiple WSAN gateways they need to attach to.

Because of all these issues and complicated solutions, it currently makes more sense to look for a WSAN mobility scheme that does not require full IP access to individual WSAN nodes and allows efficient usage by multiple applications. The results for 6LoWPAN are summarized in Table 4.2.

4.3.1.2 Sharing using instant messaging

Communication between an IP application and WSAN can be seen as instant messaging over IP, it can make use of existing instant messaging solutions. Since these solutions have either a publicly reachable server or distributed ones, both the WSAN and IP application can move while sending messages. Most instant messaging approaches offer encryption of the connection to the messaging server or the messages themselves. Only a limited number of instant messaging protocols are suitable for integration in applications (i.e. are an open standard [74]), popular ones are Internet Relay Chat [82] (IRC), Protocol for SYnchronous Conferencing [152] (PSYC), SIP for Instant Messaging and Presence Leveraging Extensions [71] (SIMPLE) and Extensible Messaging and Presence Protocol [132] (XMPP). Most of these protocols are not designed for reliability, but reachability is good for all of them since they all provide one or more ways to traverse through firewalls. The messages in these protocols are quite large because they are text-based, especially in SIMPLE and XMPP.

Unfortunately, only few instant messaging solutions (e.g. PSYC) offer efficient ways to send to multiple recipients. The results for instant messaging are summarized in Table 4.3.

4.3.1.3 Sharing using mobile stream endpoints

Communication between an IP application and a WSAN can be seen as a bidirectional message stream. A number of mobility schemes can be envisaged using this approach. The results for mobile stream endpoints are summarized in Table 4.4.

Sharing using transparent mobility : MIP could be used to transparently support mobility for both sides of this bidirectional stream. A drawback of this approach is that the WSAN needs to duplicate its sensor messages to each application, and that there is no good support for intelligent buffering when there is a connection outage, since MIP transparently keeps connections open even when there is temporarily no Internet connection.

Sharing using nomadic mobility : A bi-directional connection could be set-up between the WSAN and each application. The overhead can be low when a compact asynchronous protocol is used, or high when a synchronous protocol with verbose messages is used (such as Simple Object Access Protocol [105] (SOAP)). In cases of connection-loss, the WSAN would queue the messages that could not be sent and re-sent them in another order when the connection is re-established later (possibly from a new IP address). Big drawbacks of nomadic mobility are:

- the WSAN and application may not be able to find one another when they move at the same time.
- communication is duplicated when multiple applications use one WSAN
- the WSAN will need to do access control for every application.
- bi-directional messaging does not work very well with web services when only one communication endpoint is publicly reachable. This would involve some sort of polling to get the requests from the other direction.

Sharing using nomadic mobility with public server : Nomadic mobility can be enhanced using a publicly reachable server towards which both WSANs and applications set-up a bi-directional stream. This enables both WSANs and applications to be mobile and at the same time reduces messaging that would otherwise be duplicated at the source, i.e. the WSAN only has to sent sensor info once and the server duplicates it to all connected applications. An example is the asynchronous Ambient middleware [36]. With web services, the bi-directional messaging drawback worsens, since all interested applications will have to poll for updated sensor data and the WSAN will have to poll for configuration and actuation requests.

Sharing using session mobility : A session could be set up between the IP application and the WSAN, for instance with SIP. This session will contain the bi-directional messaging connection between the WSAN and the IP application. A SIP re-INVITE can be used to move the end-points on either side to another IP address. Just like in nomadic mobility, messages during connection outage can be queued and sent in a different order when the connection is re-established. The WSAN gateway is expected to handle sending messages to sleeping nodes and will forward all messaging from the WSAN to the application. Since the WSAN gateway is the IP endpoint of communication with applications, it can also easily support network layer security mechanisms such as Virtual Private Network (VPN) and Internet Protocol Security [83] (IPsec). There are a number of issues with this approach:

- Each WSAN will need to do access control for every application.
- The WSAN needs to replicate messages for every attached application, wasting uplink bandwidth.
- Connection setup must be supported at both network ends (possibly private or protected networks).

4.3.1.4 Sharing using WSAN as content source

When communication between an IP application and WSAN is seen as a content stream from the WSAN to all interested applications, the messaging could be optimized by bundling communication to groups of applications. The results for mobile content sources are summarized in Table 4.5.

Sharing using IP multicast : IP multicast by the sender enables sending information to multiple recipients that can join the stream. IP multicast has a number of issues:

- Mobility of the content source has only recently become a research topic, and would typically involve context transfer between routers.
- Configuration and actuation messages towards the source would have to use a different protocol.
- IP multicast is mainly deployed in content-distribution networks for predefined sources, and a lot of routers in the Internet do not yet support or allow it.

Sharing using content-based routing : With content-based routing [28, 60, 55], routing is done based on elements of the content body instead of the destination. Interested applications can subscribe for different content. Content-based routing has the following issues:

- Mobility of the source has only recently become a research topic.
- Routing is often implemented on the application layer, inheriting application protocol overhead.
- Configuration and actuation requires reverse traffic, but WSAN could subscribe for these events.
- Messages are not cached for unconnected clients, however a subscription proxy [147] could be used.

Sharing using cache-and-forward routing : In cache-and-forward routing [113], interested applications can subscribe to content via a local post-office which will look up the source post office via a naming service. The content is sent by the source via cache-and-forward routers towards the destination(s). It allows efficiently sending content by the (mobile) source to multiple recipients. Both the sender and the receivers can be mobile.

- Cache-and-forward routing is a future Internet research topic and would require deployment of a number of network elements.
- Configuration and actuation messages towards to source would have to use a different protocol.

Sharing using partial session mobility with relays : When the session between the WSAN and applications is split in sub-session between the WSAN and sub-sessions between the relay and each application, mobility can be supported for both the WSAN and the applications without duplication at the source (but at the relay instead). The duplication can be further reduced by adding additional relays in different network segments. With SIP, a SIP application server can be used to automate splitting the sessions [13, 153]. The INVITE from an application towards the WSAN is therefore picked up by a SIP application server and split into sub-sessions:

- One sub-session between the WSAN and the relay. This sub-session is typically set-up when the first application subscribes to the WSAN messages using a SIP INVITE.
- Other sub-sessions between the relay and each application. These sub-sessions are set-up for each application that subscribes.
- To further reduce duplicate message streams, a sub-session between a relay and another relay can be set-up when multiple applications in the same network segment subscribe to the same WSAN stream. A SIP re-INVITE can be used to split the sub-session between the initial relay and the application(s) to one: between the relays and others between the new relay and each application.

A further advantage of splitting the streams, is that private and protected networks are less of an issue, since no connection needs to be set up directly between these sort of networks, because a relay will be used that is reachable by both endpoints. Configuration and actuation requests towards the WSAN can likewise be intercepted and be transformed into a configuration or actuation message towards the WSAN after access control is checked and when there is no conflict between multiple applications. For example when one applications requires a temperature update every 5 minutes instead of the default 15 minutes, the WSAN nodes can be configured to sent it every 5 minutes and the relay would forward it in this pace to the requesting application, and keep forwarding it every 15 minutes to the other applications.

4.3.2 Reflection

Currently, only few reasonably mature sharing schemes offer buffering, low overhead and don't need an additional protocol for requests. These are Nomadic

Table 4.2: 6LoWPAN Mobility

	multi-move	smart buffer	overhead	duplic. node	maturity	deployment needs	access control	request method
6LoWPAN	ok	–	low	end-node	–	IPV6, HA + NEMO	end-node	same

Table 4.3: Mobility using instant messaging

	multi-move	smart buffer	overhead	duplic. node	maturity	deployment needs	access control	request method
SIMPLE, XMMP, IRC	ok	–	medium-high	gateway	++	client API server	server	same
PSYC	ok	–	low	server	+/–	client API server	server	same

Table 4.4: Mobility using stream endpoints

	multi-move	smart buffer	overhead	duplic. node	maturity	deployment needs	access control	request method
MobileIP	ok	–	low	gateway	+/–	HA	gateway	same
Nomadic	-	+/–	low-high	gateway	++	self-contained	gateway	same
Nomadic with server	ok	+/–	low-high	server	+/–	self-contained	server	via server
Session mobility	ok	+/–	low	gateway	+/–	SIP server	gateway	SIP message

Table 4.5: Mobility of content sources

	multi-move	smart buffer	overhead	duplic. node	maturity	deployment needs	access control	request method
IP multi-cast	–	–	low	router	++	router(s)	router	separate
Content-based routing	–	++ with proxy	low-medium	server	++	client API, server(s)	server	WSAN sub-scribes
Cache& forward routing	ok	++	low	proxy	–	multiple IP tunnels	proxy	separate
Partial sessions with relays	ok	+	low	relay	–	SIP server and relays	appl. server	SIP message

with server when a compact asynchronous protocol is used, session mobility and content-based routing with a proxy. However, session mobility does not scale well since it does access control at the gateway for each application and duplicates messages for multiple applications at the gateway, wasting precious uplink bandwidth. Content-based routing with a proxy does scale well, but does not guarantee reliable communication and source mobility is still a research topic. So, the nomadic with public server with a compact asynchronous protocol provides the best current option. The partial sessions with relays scheme forms a good, but non-mature, alternative when more applications use the WSAN data, since it can add relays in different network segments on the fly. This option is similar in efficiency to cache&forward routing although it uses a session approach instead of post-offices and can use its own protocols for sending messages towards the source. Cache&forward routing could also provide a solution for some of the shortcomings of 6LoWPAN, since it makes it possible to cache&forward the sensor messages for interested applications instead of direct IP connections.

4.3.3 Requirements

Important requirements on this shared messaging are: minimal message loss, support for high message rates and occasional large messages, mobility, firewall traversal, and limited overhead. The requirements for sharing can be divided in general, management and application related requirements. The general requirements are:

- support high message rates from WSAN, for instance 1000 nodes with 3 sensors (e.g. temperature, humidity and tilt) sending a message every 5 minutes yields an average rate of 10 messages per second. For inertial measurements, a sampling rate of 100Hz or higher from multiple nodes is quite common.
- minimal message loss (preferably none)
- enable WSAN behind firewall and in private networks behind Network Address Translation (NAT) router
- work with fixed-IP and dynamic (Dynamic Host Configuration Protocol [51] (DHCP)) addresses both in Internet Protocol version 4 (IPv4) and IPv6
- enable mobility of WSANs (terminal mobility)

- support logging of messages while disconnected, and flushing unsent messages when connected again
- support encryption
- support large message size (e.g. for file up/down-load)
- minimal bandwidth, e.g. should work over General Packet Radio Service (GPRS) link

Additional requirements for WSAN management:

- enable remote maintenance of WSANs
- make realtime network information available remotely
- inspect and manipulate network queues
- enable remote configuration and commands on all nodes in a WSAN
- support firmware upload and dissemination to nodes

Additional requirements for applications:

- multi-computer language support (At least Java, C(++), and C#/.Net bindings)
- make real-time sensor information available to multiple applications
- enable actuation from multiple applications
- support filtering and/or compression to reduce required bandwidth and handling in applications
- Messages should be easily creatable and processable and clearly distinguish the message type (command request/response or sensor data), and clearly separate header and message parameters.

4.3.4 Ambient middleware

Ambient Systems [22] has created a middleware architecture to enable customers to easily integrate their applications and to enable remote monitoring and maintenance. The interaction between the different components is depicted

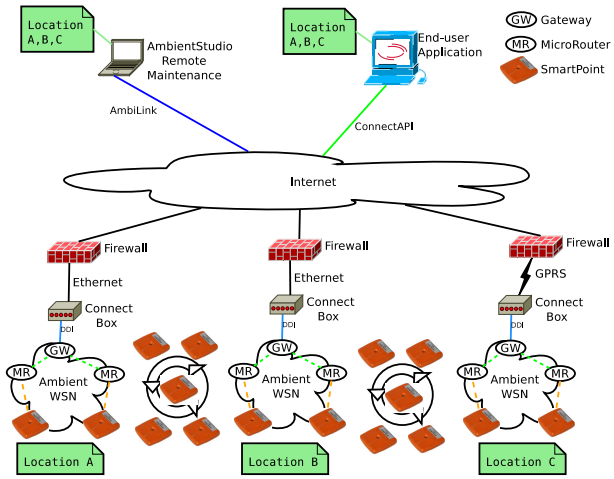


Figure 4.13: Ambient connect framework

in Figure 4.13. Note that AmbientStudio and the ConnectBox share the same Ambient middleware (AmbientMW).

One or more Gateways can be connected via RS232 using the AmbientMW in a ConnectBox device or AmbientStudio on a PC. The ConnectBox device is an embedded Linux device that offers Ethernet connectivity towards the wireless nodes from eXtensible Markup Language (XML) applications, AmbientStudio, or other AmbientMW instantiations.

The AmbientMW offers the ConnectAPI to ease integration with third-party applications using asynchronous XML messages over a TCP/IP connection (optionally encrypted with Secure Socket Layer [63] (SSL)). The XML messages are similar to Device Driver Interface (DDI) messages but fully parsed so they can be easily used in an application utilizing its XML schema (which enables code generation in for instance Java and C#). When required, a pass filter can be configured to reduce the type of DDI messages that are forwarded over the ConnectAPI.

To offer flexibility, the ConnectAPI can be started as client and server:

- ConnectAPI server allows multiple local or remote applications to connect.
- ConnectAPI client allows connecting to a remote host, automatic re-

connects, and automatic logging of messages while disconnected and flushing when the connection is re-established.

The AmbientMW also offers AmbiLink (compared to ConnectAPI in Table 4.7) to ease remote monitoring and maintenance of sensor networks using asynchronous binary messages over optionally SSL encrypted TCP/IP connections. Similar to the ConnectAPI, both AmbiLink client and server can be started. This offers the flexibility to monitor and maintain multiple ConnectBoxes with one or more AmbientStudio instances without loosing messages when client connections are disrupted. Additionally to DDI messages, also management messages can be sent over both ConnectAPI and AmbiLink for configuring, opening and closing serial ports and remote connections. New message types can easily be added, for instance, for fetching historical data or changing DDI message filters. Another message type could be introduced for file exchange (for instance firmware) with the WSAN, such that the WSAN can use its own pace and protocol for exchanging it with the involved node(s). AmbiLink also supports merging sensor information from all connected nodes via multiple ConnectBox or AmbientStudio instances, and routing messages to them. It can then provide the merged data to multiple applications using the ConnectAPI. Wildcards can be used in message destinations, which enables addressing multiple AmbientMW instances at once.

For both AmbiLink and ConnectAPI, conversion between DDI and respectively their binary and XML counterpart was automated. Logging and flushing is implemented in the middelware for both protocols in order to cache messages that cannot be sent by the client during connection outage. Server logging and flushing is not implemented, since sensor messages are usually towards a server and there is no guarantee that a client will ever reconnect to the server. To reduce message loss, the TCP connections were set up such that small messages were sent without delay and the last sent message is logged until it is possible to sent the next message. This removes the need for a special acknowledgement scheme on top of TCP (which already has its own acknowledgements), since a new message cannot be sent unless the previous one was successfully sent.

4.3.5 Messaging efficiency

In this section the efficiency of ConnectAPI and AmbiLink is compared with existing messaging protocols.

4.3.5.1 Comparing existing methods

Existing methods for messaging over the Internet are the email protocol Simple Message Transfer Protocol [86] (SMTP) for sending/receiving email, and Instant Messaging (IM) protocols like IRC, PSYC, SIP/SIMPLE and XMPP. Also web-services like SOAP and Representational State Transfer (REST), and Peer to Peer (P2P) messaging like P2P SIP can be used for message exchange over the internet. These messaging protocols can be used over a variety of transport protocols like TCP, UDP, Stream Control Transmission Protocol [145] (SCTP), Datagram Congestion Control Protocol [87] (DCCP), and multicast, and use a variety of security protocols like IPsec, SSL and Transport Layer Security [50] (TLS). Most of the protocols can also provide nomadicity (i.e. reconnection after connection loss), MIP can be used to provide seamless connectivity when switching networks. Each of these protocols has its strong and weak points, which will be described in this subsection.

Criteria for comparing the existing methods are the following:

- **Availability**: are the required elements widely deployed, or are can they be easily deployed? Availability is positive when the protocol is generally supported in the endpoints and intermediate routers, negative when is is hardly supported on the endpoints and routers. For instance MIP and multicast are not widely deployed, application-level protocols can often be easily deployed.

- **Impact**: The impact is high when the routers along the path must be equipped to support the protocol (denoted as "dr" for dedicated router), or when the firewall must be updated to support incoming traffic (denoted as "df"). The impact is also high when dedicated clients (denoted as "dc") or a dedicated server (denoted as "ds") is required. The impact is less when a library can be used for clients (denoted as "lc"), and servers (denoted as "ls"). Using for instance XML messages, usually requires a library for parsing it.

- **Latency**, i.e. are messages forwarded in real-time, or are there inherent delays? For instance request-based mechanisms like web-services require higher bandwidth and processing time and double that with the required return messages. Messaging protocols like XMPP and SIMPLE add latency by one or more intermediate proxies between the clients and have a verbose message format that takes some parsing time. The SMTP servers usually cache the messages on disk before forwarding, and clients usually poll for new messages with intervals in de order of minutes.

- **Reliability**: is message loss prevented, or is there a mechanism to prevent losing messages?
- **Reachability**: can the WSAN be reached remotely when there is an Internet connection? For instance (company) firewalls often block all incoming ports and are not keen on clear-text protocols, a default NAT router blocks all incoming connections unless configured with specific forwarding rules.
- **Bandwidth**: can the protocol work across a limited bandwidth link such as GPRS? For instance verbose messaging like SOAP could add much overhead and other associated costs across a wireless link such as GPRS.
- **Security**: can others inject or obtain messages, or disrupt the service? Can the protocol easily be encrypted?

The connection-oriented transport protocols TCP, SCTP, and DCCP allow messages travelling in the opposite direction once a connection has been established. Once a connection-oriented server is set up such that it is reachable, most clients can connect to it (unless they are behind a firewall that blocks outgoing connections). In general the connection-less protocols like UDP and multicast require assistance for sending packets in both directions over the internet when one of the parties is behind a firewall and/or in a private networks. These issues can be overcome with protocols like Session Traversal Utilities for NAT [131] (STUN) and Interactive Connectivity Establishment [126] (ICE) that require a third reachable party on the internet in order to enable communication in both directions. The connection-less and connection-oriented protocols described above do provide encryption of the exchanged data. The SSL protocol works on top of TCP, TLS can also work on top of the other transport protocols. IPsec works in the Internet layer and can both be routed (transport mode) and tunnelled (tunnel mode) over TCP and UDP. IPsec requires support in the network that is being connected to. Table 4.6 compares the transport protocols and popular security protocols.

The web-service protocols XML-Remote Procedure Call (RPC) and its successor SOAP use XML documents for messaging. REST can use both text, XML and other representations (for a request an URL could suffice). These web-services all use the request/response model of HTTP. The WebSockets [59, 67] protocol that is drafted for Revision 5 of HTML (HTML5) allows bidirectional communication of user-defined messages. JavaScript Object Notation (JSON)-RPC uses a compact representation and is one of the few web-service protocols that can also be used bi-directionally over a socket, i.e. it allows requests, responses and notifications to be sent asynchronously in each direction over the

Table 4.6: Comparison of transport and security protocols

Protocol	Availability	Impact	Reliability	Reachability
MIP	-	ds+dc	+/-	two-way
TCP	+	ls+lc	+	two-way
UDP	+	ls+lc	-	issues
SCTP	-	ls+lc+dr	+	two-way
DCCP	-	ls+lc+dr	+/-	two-way
Multicast	-	ls+lc+dr	+/-	issues
SSL/TLS	+	ls+lc	+	two-way
IPsec	+/-	ds+dc	+	needs support

same connection. When behind a firewall the other protocols require either opening a firewall port, tunnelling or polling on a reachable server to receive messages (SOAP could also be used over SMTP with associated high latency, but then it would not act as a web-service). Using HTTP Secure (HTTPS) for security increases the latency of the first message, since the connection needs not only to be set up for each request but also the security association. The reliability of web-services is generally ok. Multiple libraries are available for all protocols, however there is no cross-platform C++ library available for JSON-RPC (JsonRpc-Cpp is GPLv3 licensed which requires opening all linked source when releasing). Table 4.8 compares the popular web-service protocols.

For messaging over the Internet, a great number of protocols exist. Only a limited number of these protocols are suitable for integration in applications (i.e. are an open standard [74]). Most of these protocols are not designed for reliability, but reachability is good for all of them since they all provide one or more ways to traverse through firewalls. The messages in these protocols are quite large because they are text-based, especially SIMPLE and XMPP. Table 4.9 compares the popular open messaging protocols.

4.3.5.2 Comparing with Ambient middleware

The AmbiLink users binary DDI and ConnectAPI uses DDI in XML format for messaging, for both messaging is asynchronous, meaning that no response is required like in web services. When an AmbiLink or ConnectAPI client is behind a firewall, it can still reach its related server on the Internet without having to reconfigure the firewall. Both AmbiLink and ConnectAPI can be secured with SSL with the added delay of setting up the security association.

The reliability of the Ambient middleware is ok, it logs and flushes messages when the connection is temporarily unavailable. AmbiLink only works as part of the Ambient middleware, ConnectAPI can be used from any program that can send XML documents over a socket. Table 4.10 compares the Ambient middleware protocols.

Table 4.11 compares the number of messages and bandwidth for a number of protocols in more detail[1]. Typical HTTP header size is 256 bytes, the size of XML and JSON documents are comparable when XML attributes are used instead of tags (else XML is about 30% larger), a typical size of such a message is 1024 bytes. A typical SOAP envelope adds 172 bytes. Typical WebSocket messages are expected to contain JSON payload since that is easily handled in Javascript. Typical AmbiLink binary sensor messages are approximately 250 bytes long, typical ConnectAPI messages are approximately 900 bytes long. ConnectAPI messages make heavy use of XML attributes instead of tags, which make them comparable in size to JSON messages.

The table clearly shows that the asynchronous messaging of JSON-RPC, WebSockets, AmbiLink and ConnectAPI saves the return-trip messaging as well as the HTTP headers. Depending on the setup of server and client, the HTTP keep-alive can keep the TCP connection open for a long time. However, usually the keep-alive timeout is less than a minute, which means more connection setups (and associated higher latency) for low-frequency messaging over HTTP. Note that the typical SOAP messages are around 1500 bytes, so a slight increase would require an additional TCP packet. WebSockets make polling schemes unnecessary by offering bidirectional messaging through a webserver.

4.3.5.3 Bandwidth optimizations

The aim is to use the Ambient middleware protocols across low bandwidth links like GPRS, in which the download bandwidth varies between 9 and 52 kbit/s, and upload is usually limited to 18 kbit/s. It is envisaged that also large sites may want to use GPRS to be independent of Ethernet infrastructure which could be owned or managed by another party or simply be unavailable in a storage area. For instance 1000 nodes with 3 sensors (e.g. temperature, humidity and

[1]TCP uses 3-way handshake for setup and teardown, the set-up ACK can already contain part of the message, HTTP1.1 can use keep-alive which reduces the number of required TCP connects, TCP message header is 24 bytes, The latency of messages doubles when there is an explicit response for each message. The table assumes that each TCP message is acknowledged, where it practice the acknowledgement can be for a number of them (depending on the rate of transmission). IP header is 24 bytes

Table 4.7: AmbiLink versus ConnectAPI features

Protocol	Usage	Transport	Security	Format	Filter	Destination	Merged WSANs
ConnectAPI	3^{rd}-party applications	TCP/IP	SSL option	XML	header fields	Broadcast to all applications	Using multiple clients
AmbiLink	Monitoring & maintenance	TCP/IP	SSL option	binary	per WSAN	Route to AmbiLink instances	At client or server

Table 4.8: Comparison of web service protocols

Protocol	Availability	Impact	Latency	Reachability	Bandwidth	Security
XML-RPC	+	ls+lc	medium	issues	medium	HTTPS
SOAP	+	ls+lc	medium	issues	high[78]	HTTPS
REST	+	ls+lc	medium	issues	depends	HTTPS
JSON-RPC	+/-	ls+lc	low	two-way	low/medium	SSL/TLS
WebSockets	+/-	ls+lc	low	two-way	low/medium	SSL/TLS

Table 4.9: Comparison of open messaging protocols

Protocol	Availability	Impact	Latency	Reliability	Bandwidth	Security
SMTP	+	ds+dc+lc	high	+/-	medium	-
IRC	+	ds+dc+lc	low	+/-	medium	SSL
PSYC	-	ds+dc	low	+/-	medium	TLS/SSL
SIMPLE	+/-	ds+dc+lc	medium	+/-	high	TLS
XMPP	+	ds+dc+lc	medium	+/-	high	TLS

Table 4.10: Comparison of Ambient middleware protocols

Protocol	Availability	Impact	Latency	Reliability	Bandwidth	Security
AmbiLink	+	ds+dc	low	+	low	SSL
ConnectAPI	+	ds+dc+lc	low	+	medium	SSL

Table 4.11: Comparison of Bandwidth (in bytes) & latency for N message exchanges, and bandwidth for N=10

Protocol	TCP/IP headers 48 bytes	HTTP headers 256 bytes	request messages	response messages	typical message size	bandwidth N=10 & 1 TCP connect
XML-RPC	5+4N..9N	N*2	N*XML	N*XML	XML=1024	27760
SOAP	5+4N..9N	N*2	N*(env.+XML)	N*(env.+XML)	envelope=172	36790
REST	5+4N..9N	N*2	N*XML	N*OK	URL|OK|other=100	19000
JSON-RPC	5+(2N..4N)	0	N*JSON	(0..N)*JSON	JSON=900	10200..20160
AmbiLink	5+2N	0	N*AmbiLink	0	AmbiLink=250	3700
ConnectAPI	5+2N	0	N*ConnectAPI	0	ConnectAPI=900	10200
WebSockets	9+(2N..4N)	2	N*(WS+4)	(0..N)*(WS+2)	WS=900	10944..20924

tilt) sending a message every 5 minutes yields an average rate of 10 messages per second[2].

For 10 AmbiLink messages per second[3] that would yield a bandwidth of 2500 bytes/s = 20 kbit/s. So, also AmbiLink could certainly use compression for bigger sensor networks over GPRS. A simple gzip [47] on a binary message gives a compression factor of 1.6 on AmbiLink messages. Compressing a group of messages, e.g. 4 at a time gives compression rate of 4, 25 at a time gives a compression rate of 8. So it would make sense to compress a group messages (e.g. all messages to be send in a second) when possible, this would also reduce the overhead on the TCP/IP level, but will increase the message latency. AmbiLink messages could also be reduced in size by shortening them or using a generic compressing on string values in these messages that are now sent as UTF-8. Huffman coding [69] would be a candidate for this, an alternative would be a look-up table for commonly used attribute names.

Sending 10 ConnectAPI messages per second would require a bandwidth of 70 kbit/s. Compression of these XML messages would thus be required for using ConnectAPI across GPRS with bigger networks. Compression with gzip of a temperature message achieves a compression factor of 1.8. Compressing a group of 4 messages yields a compression factor of 3, compressing a group of 25 messages yields a compression factor of 18. Some more can be saved by stripping redundant information from the ConnectAPI messages and shortening the XML tag and attribute names. A large part of these attribute and tag names come from the DDI descriptors, so shortening them in these descriptors will reduce the bandwidth.

4.3.5.4 Measurements

We have measured the number of messages we can send using different protocols, namely native TCP sockets (such as for instance AmbiLink uses), websockets, REST and XMPP. The setup is as follows: a client machine (Android tablet) has a Java application that has client libraries for each protocol and files with newline separated JSON strings. For each protocol it sets up a connection to a server and sends the messages, for subsequent runs, the connection is setup again. The measurements were done both with an high-speed Internet connection (20Mbit up/down) and with constrained bandwidth similar to that of

[2]note that these message rates are only required when full sensing history is required, else it is more practical to configure alarms in the SmartPoint on specific sensing conditions

[3]The amount of messaging depends on the number of SmartPoints, its reporting period or alarm thresholds, and scale of the network (current maximum is 64 infrastructure nodes)

Table 4.12: Client-side measurements with JSON messages of 214 bytes

	Fixed internet timings (s), mean ≥ 50 runs				18kbit over WiFi timings (s),mean ≥ 50 runs			
Protocol	10 msgs	100 msgs	2279 msgs	msgs/s	10 msgs	100 msgs	2279 msgs	msgs/s
TCP socket	0.0169	0.0342	0.374	6093.6	1.56	9.19	68.08	33.48
jWebSocket	0.0493	0.0614	0.492	4632.1	2.86	7.07	64.61	35.27
Restlet	0.4260	4.8030	106.924	21.3	3.23	34.05	641.31	3.55
XMPP asmack	1.6259	13.0529	288.425	7.9	5.66	23.85	350.88	6.50

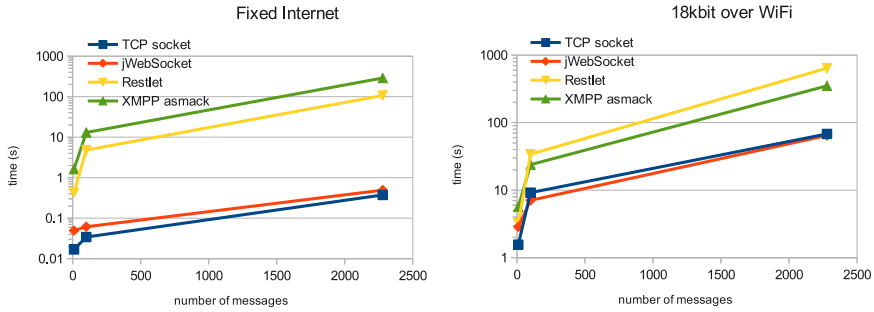


Figure 4.14: Client-side messaging speed for JSON messages of 214 bytes

GPRS over WLAN using traffic shaping. The native Java sockets are used for the TCP socket, the net extension of Restlet [49] (version 2.1.0) is used for REST, the Java client-side from jWebSocket [79] (version 1.0) is used for websockets, the asmack [150] library (version May 2010) is used for XMPP. At the serverside, a Restlet server (version 2.1.0) was used for REST, a Java server socket was used for native sockets, jWebSocket (version 1.0) was used for websockets, and the talk.google.com server was used for XMPP. Note that both the Restlet and Java socket servers actually process the data, the jWebSocket server just receives the data (debug logging was changed to info to maximize throughput), and talk.google.com just forwards the data when there is a connected destination and otherwise stores it in the email box.

The measurements are depicted in Table 4.12 and Figure 4.14. The XMPP protocol performed worst over the high-speed connection, this is explained by rate limiting in XMPP servers that allow only 8 messages per second per user in the default configuration. XMPP degraded less when the bandwidth was constrained, since the number of messages per second was already constrained. Restlet also performed badly, since a TCP connection was setup for each indi-

vidual message, HyperText Transport Protocol (HTTP) keep-alive would help at the high rates, but was not supported in our setup. The Restlet performance decreased even more with an increased number of messages over the constrained connection. The native sockets performed slightly better than the websockets over the high-speed connection, which is understandable since websockets need an HTTP exchange for setting up and a 4 bytes per client-side message. The websockets performed better over the constrained connection with 100 messages per connection and higher. Taking a closer look at the measurements revealed that there where more cases that the connection setup was delayed for native sockets than for websockets. Messaging over asynchronous socket connections obviously perform much better than REST and XMPP, and XMPP outperforms REST on a constrained mobile connection like GPRS.

4.3.6 Example deployment

Antaris Solutions [25] offers the SmartView web application to monitor temperature, humidity and/or shelf-life of static location or cargo with an attached SmartPoint. The SmartView Bridge uses the ConnectAPI to receive sensor measurements from AmbientStudio or ConnectBoxes and forwards the samples to SmartView. This system is deployed for Panalpina for monitoring the quality of pharmaceuticals in storage and in transit, see figure 4.15. The figure clearly shows the use of the system across Intranet boundaries. Since the ConnectAPI connections are all initiated from within an Intranet towards the SmartView server, there is usually no need for firewall configuration as long as the SmartView server is reachable across the Internet. The same holds for any customer application that uses the ConnectAPI.

During deployment we met a number of challenges:

- The gateway and ConnectBox could not be placed in its ideal position (middle of the network) because of power outlet and Ethernet availability and/or GPRS coverage.

- In order to cover also refrigerators, MicroRouters needed to be placed inside or close to their door. This was very challenging with the maximum of 32 routers at the time, currently we support 64 of them.

- Permission to use the Intranet often takes time or meets resistance, therefore the GPRS-enabled ConnectBox is sometimes also used in storage areas.

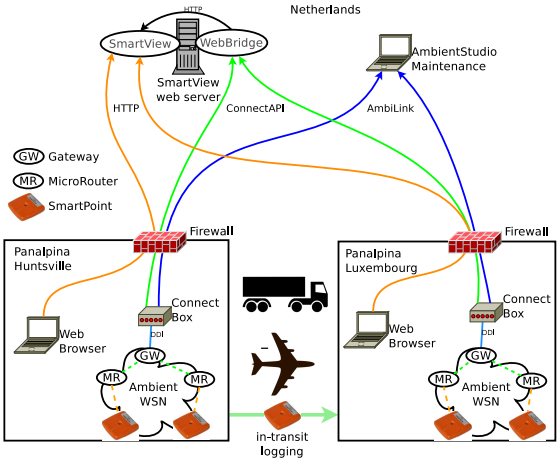


Figure 4.15: Panalpina remote sensing and maintenance

- AmbiLink is only required for occasional maintenance, but acquiring permission to open a server port is often not possible. Therefore we made it possible to configure and start an AmbiLink client via the ConnectAPI interface, or alternatively to configure the AmbiLink client to periodically connect to a specific maintenance server.

4.3.7 Conclusions

This section analysed scenarios in which different WSAN and application movements take place. Moving end-nodes between different WSANs can be supported by compatible WSANs, but does not allow controlling when the movement takes place. With different encryption in each WSANs, the end-nodes can associate with the other WSAN at a convenient moment.

To reduce interference between WSANs, the moving gateway can turn off its gateway or switch to intermediate node mode to make end-nodes communicate with the other WSAN. To prevent interference from overlapping WSANs, it is advisable to adapt the wireless resources before the overlap occurs, for instance by detecting similarity in Global Positioning System (GPS) coordinates of the WSANs. With different WSAN types, data of overlapping WSANs can best be merged at the application layer. In order to support coexistence of WSANs

using the same wireless resources, WSAN protocols should be robust against foreign protocol messaging. When privacy is required, as is often the case in body sensor networks, messages can better be encrypted with the public key of the receiving gateway (or middleware), which can in turn sent it encrypted to one or more applications.

For sharing WSANs among few applications, nomadic mobility with a server using compact asynchronous messaging has the best properties. When the number of applications increases, schemes that bundle traffic towards groups of applications become more attractive.

We proposed a middleware layer to support real-time monitoring, remote maintenance and application integration of WSANs in logistic scenarios across the Internet via wired and mobile wireless network access technologies. Its messaging efficiency was compared with that of well known web protocols and recommendations have been made to further increase the messaging efficiency. Additionally, messaging speed measurements confirmed that using asynchronous communication over (web)sockets greatly increases the number of messages that can be sent over high speed and constrained connections such as GPRS.

4.4 Conclusions

In this chapter we proposed and analysed methods for sharing media streams and WSANs across devices and applications.

We proposed a bandwidth-distribution mechanism for WLAN that uses real-time characteristics of the network medium and feed-forward control mechanisms to regulate the bandwidth distribution. With our prototype we showed that centrally controlled traffic regulators can successfully be used to protect QoS sessions from otherwise uncontrolled best-effort traffic. The mechanism does however require support in all terminals and in the router that connects them with other networks. The mechanism itself cannot protect against non-conforming endpoints, and their potential traffic would reduce the overall available bandwidth. Those terminals would of course be denied access by the access point when they do not have the right credentials.

We proposed a flexible approach using SIP to blend general and targeted content streams in multimedia sessions while enabling efficient distribution of streams per network segment utilising dynamic switching between unicast, multicast and broadcast streams. We noticed that MBMS only appears to increase efficiency over HSDPA in rural areas and only when delivering a limited number of TV channels. Using even larger cells such as DVB-T is expected to be more

economical to transfer popular content. Since simultaneous use of the same content stream changes per network segment and in time we propose to group those content streams per network segment dynamically as the group grows.

We analysed the mobility and sharing of sensor networks in logistic and person monitoring scenarios. We determined that nomadic connections from WSANs to a public server with a compact asynchronous protocol is the best current option to support mobility and sharing of the measured data and remote configuration. When the number of applications interested in the measurements increases, a SIP approach with partial sessions and relays (similar to the efficient content stream distribution above) is expected to be more efficient.

We proposed and analysed the efficiency of a middleware layer for real-time monitoring, remote maintenance and application integration of WSANs in dynamic logistic scenarios across the Internet using asynchronous communications.

In Chapter 5 we will analyse pervasive architectures that make use of shared mobile resources.

Chapter 5

Pervasive service and system architectures

This chapter describes architectures that support pervasive services and systems in heterogeneous multi-operator networks. It first describes the Daidalos approach that supports pervasive services in an open way on heterogeneous and context-aware networks (see Section 5.1). The Daidalos project ran from 2003 to 2008, and we were responsible for the service provisioning architecture, and focused on multimedia session management and broadcast integration. Our work on partial session mobility (see Section 3.2) and sharing of multimedia streams (see Section 4.2) has been performed in this context. Next, this chapter compares architectures for pervasive systems and services and gives guidelines for flexible pervasive system architectures (see Section 5.2). This work was performed between 2010 and 2011 in the SENSEI project [45].

5.1 A pervasive service platform architecture

In this section we describe a communication infrastructure that enables personalized, context-aware, composite services for mobile users of next generation heterogeneous networks. The Daidalos platform is an example of a pervasive service platform as defined in Section 2.4 that supports mobility, QoS and sharing of among others multimedia sessions and context information. Its communication infrastructure allows federation of operators that implement (parts of) the Daidalos platform, remote access and composition of of both platform and

third-party services, integrated mobility, security, virtual identity for users, and resource management. This work was done in the context of the EU funded Daidalos project (IST-2002-506997 and its successor IST-2006-026943) between 2003 and 2007. The integration of the Daidalos concepts have been validated by means of specification, implementation, and integration in a large research testbed.

5.1.1 The 5 key concepts

The Daidalos approach can be categorized into five key concepts that together enable a seamless and context-aware user experience across federated operators:

1. **MARQS**: MARQS makes network usage transparent despite heterogeneity, providing authorized users seamless access, it encompasses Mobility Management, AAA, Auditing and Charging (A4C), Resource Management, Quality of Service (QoS) and Security. The solution must at least be on par with existing solutions, such as at 3GPP, while offering flexibility and openness.

2. **VID**: Virtual IDentitys (VIDs) [134] make users independent of their own or public devices, support privacy and the provisioning of services to users independent of what device they use and whether they own them. Such VIDs are assigned to a user independent of a specific device and contain the profile of services and networks used and may be used for pseudonymous access.

3. **USP**: Ubiquitous and Seamless Pervasiveness makes services, networks and content ubiquitous and seamless. USP enables pervasiveness across fixed, portable and embedded devices and adapts to changing contexts and movement, as well as user requests.

4. **SIB**: Seamless Integration of Broadcast integrates entertainment, such as via TV or radio services, with information and communication services. Integration is needed at both the service level, e.g. for movies, and at technology levels, e.g. integrating Digital Video Broadcast (DVB) and Wireless LAN (WLAN), to separate the concern of what will be delivered from how it will be delivered.

5. **Federation**: Enable business players to enter and leave an area of business in a dynamic manner, such as to offer a new network, a new network service

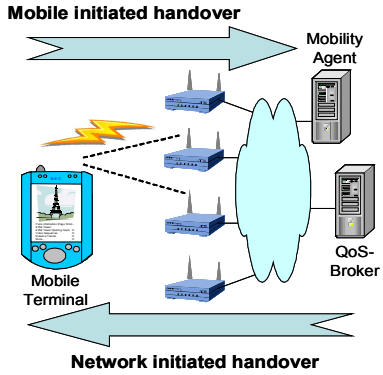


Figure 5.1: Mobile and network initiated handovers

or a new information service or content, and cater for both existing and new, large and small network operators and service providers. Such a flexible business environment on the whole will benefit all, incumbents and newcomers, whether large or small.

The following subsections further detail these 5 concepts.

5.1.1.1 The MARQS concept

MARQS brings together 5 critical dimensions of ubiquitous networks, and extends the three-dimensional approach covering mobility, A4C and QoS with resource management and security.

Seamless mobility has been enhanced by improving fast handover to cover both terminal-initiated and network initiated handover (see Figure 5.1). Handover by the network can be useful when it is required to balance load across access points, thus supporting resource management. Such handover covers several access technologies, and handover has been demonstrated for WLAN, Ethernet and TD-CDMA. Additionally, we support soft handover with more than one interface (multi-homing) carrying data during the handover phase. Quality of Service in Daidalos supports the negotiation and management of network resources at the IP level for both legacy and multimedia services, while guaranteeing mobility. For security and A4C, we developed a flexible access control mechanism for the heterogeneous and mobility-enabled environment with accounting and charging mechanisms. This can be both session or flow-based,

as well as pre- and post-paid. The Daidalos network facilitates the provisioning of basic network services including multimedia through its platform that also supports service discovery. Content delivery to heterogeneous clients is made possible using content adaptation schemes signalled with Session Initiation Protocol [31, 123] (SIP). Several flavours of mobility are thus supported, ranging from terminal via session to user mobility. To discover the identities and capabilities of candidate Access Routers, Daidalos has specified and implemented the Candidate Access Router Discovery [92] (CARD) protocol and integrated this with functions that allow network-initiated fast handover based on performance measurements. Full QoS support at Layers 2 and 3 including the QoS mapping between the layers and for end-to-end delivery were specified and implemented. The Key Management infrastructure supports the Daidalos Security Architecture and provides flexible access for a heterogeneous and mobile environment. Operational prototypes show power-up and registration followed by terminal-initiated or network-initiated handover supporting mobility. Additionally, seamless handover between infrastructure, Mobile Ad-hoc NETworks (MANETs) and moving networks (mobile router, see Section 2.1.1) are supported, as well as mobility of separate communication flows between network interfaces (see Section 2.4.3).

Regarding multimedia, an enhanced version of session mobility as described in Section 3.1 is supported. Partial session mobility from Section 3.2 was developed as an network-initiated alternative to a user-initiated approach.

5.1.1.2 Virtual Identities in Daidalos

VIDs are the Daidalos solution to the requirements for flexible identity management and anonymity. VIDs separate users from their device and represent user-controlled user related attributes within the system. The flexible scheme supports both privacy and personalization. The VID contains a pseudonym together with additional information such as a profile, credentials and usage trace. A VID is a view that somebody in the system has on the user. Several VIDs can be associated with an operator subscription. To enable authorization across domains with an ID-Token, a pseudonym is constructed from a unique identifier and the name of the home domain: identifier@domain. The ID-Token is a combination of the pseudonym and authentication information (see Figure 5.2) and is created by the Security Assertions Markup Language (SAML) authority, which enables the same authentication mechanism for both web and network services. The ID-Token contains the following elements: the pseudonym in plain text, another part encrypted with receiver's public key, containing the following:

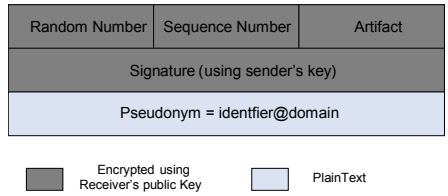


Figure 5.2: VID data structure

a random number to make the ID-token different each time it is sent, a sequence number to avoid replay, which is checked by SAML Authority, an artefact that references the appropriate SAML assertion referring to the subscription of the user with a network operator, a digital signature using the sender's private key via the ID-token excluding its pseudonym.

Different VIDs can be created to provide privacy by using an alternative name or pseudonym. The user has an option to make only part of its real identity available by revealing only the part needed for a service. A VID without any personal information can be used to provide anonymity. The VIDs can be used to access both basic services like Network Access and Multimedia Services, and more advanced services like (3rd party) applications. To enforce privacy, it should be difficult, if not impossible to correlate VID usage of the same user. To ensure this the following measures are taken: Only one component in an operator domain holds the subscription information of users including the available VIDs in this subscription. This Daidalos component is the A4C server, which embeds the SAML authority. The user has a secured store, from which he can select VIDs to use for a certain service. By limiting the accessible user information in the different VIDs and making this information disjoint, the user can make it more difficult for third parties to correlate VIDs. The user presents an IDToken with a certain VID to the service he wants to use (for instance multimedia), and the service can check its validity with the A4C for authentication purposes. By referencing the pseudonym, attributes and profiles can be stored in a distributed manner (e.g. close to where they are used) without revealing the user's identity. In order to prevent correlation of VIDs from information in the communication layers, such as IP or MAC addresses, we introduced IP addresses and virtual MAC addresses per VID. An open issue is restricting web service subscriptions to a specific VID, which could probably be facilitated by using a web subscription profile per VID.

5.1.1.3 Bringing Pervasiveness to Networks

Daidalos is filling an important gap in pervasive computing research and development. Traditional pervasive computing research has focused on user interaction aspects of pervasive computing, such as proper design of applications and devices that considers psychological and sociological aspects. Daidalos is building on this important research, and adds scalability with its federated approach. Adapting services to user needs requires knowledge about the user in the form of preferences and context information. This information is typically collected through a heterogeneous technological infrastructure, and can be represented in various formats. The information can be used for services provisioning as well as to help the user select or optimize the service. Context information affects not only single services, such as those residing on the various devices the user is using, but also influences the whole process of discovery, selection and composition of advanced services. Due to the dynamic context of the user, changing rapidly and in unpredicted ways, a composed service will have to adapt dynamically and will have to continuously reconfigure itself. This adaptation can be a source of information for better personalization and adaptation to user preferences. Each user will have his or her individual needs and will require a specific configuration of network resources for his/her devices and services from various service providers. Putting users into focus brings fundamental challenges. Daidalos is unique in that it integrates services, including 3rd party services, into the combined network and service infrastructure. Traditional network architectures are indifferent about user contexts and needs, and often divide the service layer from the lower network layers. Services become pervasive through a personalized and adaptable network. Based on the assumption that a magnitude of services will be available everywhere in a personalized manner, Daidalos divides pervasiveness into two interrelated areas of functionality:

- Everywhere access to services: The service that a user subscribes to or wants to use has to be available to the user when needed regardless of network technology, device technology, service type or user location. This is done in close interaction with device and user mobility, AAA, QoS and security. We have paid particular attention to scalability, user identity and privacy covering any device anywhere in the network. The Daidalos architecture supports discovery and composition of services and session management.

- Context-aware access to services and access to context: Not only should services be accessible everywhere, they should also be customized to the

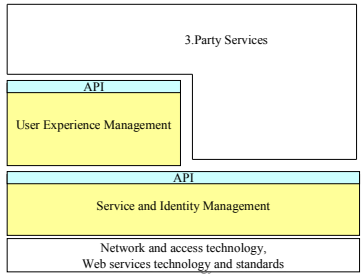


Figure 5.3: Pervasive service architecture

user's context and preferences. We develop mechanisms for collecting relevant information from the network infrastructure, sensors and services. APIs support 3rd party applications that can access and provide information in a controlled manner. This way, Daidalos platforms become enablers for service providers, as they can context-enable their services and provide additional context. Quality-assurance of this raw context information and its refinement into accurate and suitable information for services is central to Daidalos.

An overall architecture for the pervasive service platform is shown in Figure 5.3. It provides a set of APIs towards 3rd party service providers, which are mostly based on open but controlled web standards, and relate to the two areas of functionality described above. The Service and Identity Management API provides functionality for VID management and personalization, service discovery and composition, dynamic session management and deployment. The User Experience Management API provides access to functionality such as learning user preferences, negotiation of privacy when accessing services, and refined context provisioning. By adopting this layered approach Daidalos provides an attractive transition path to 3rd party service providers who have already invested in web standards.

As a comprehensive approach covering several layers, the network in Daidalos supports pervasive services by providing network-level context, such as on availability or even geographic position. This enables the user to access an abstracted network context that helps him or her make appropriate network and service choices. We further integrate and provide sensor, terminal and network based context information. These can trigger or support resource management or initiate and change multimedia sessions (such as in partial session mobility,

see Section 3.2). Such information could also be used as a trigger for service re-composition, such as by noting that someone is close to a billboard and use that event to present personalized content.

Shared access to context, including that of sensor networks (see Section 4.3), is offered by distributed context managers.

5.1.1.4 Broadcast Integration

A seamless integration of broadcast functionality into an overall communication environment has to take multiple aspects into account, in particular: full integration of broadcast radio, allowing both users and service providers to take advantage of specifics of broadcast technologies without having to consider the underlying technology (as far as possible); support for point-to-multipoint transmission in the network layer, to mimic the capabilities of traditional broadcast networks in every environment; provisioning of well-known broadcast services (such as TV or carousel) independent of the underlying communication network. We have addressed the broadcast modes of multiple radio technologies, specifically W-CDMA (Multimedia Broadcast Multicast Service (MBMS)), WiMax (IEEE 802.16), WiFi/WLAN (IEEE 802.11), DVB-Terrestrial (DVB-T), Digital Video Broadcast - Handheld (DVB-H), and DVB-Satelite (DVB-S). Each has very specific capabilities that need specific treatment to enable seamless integration. For W-CDMA, a radio access point with direct IPv6 interface had been developed in one of the predecessor projects. In Daidalos broadcast capabilities of the MBMS are added to the prototype and the mapping of IP multicast to broadcast bearers was developed. In WiMax and WiFi, the multicast channel at the radio layer imposes specific limitations such as reduced bandwidth that needs to be taken into account, in particular by the QoS management. DVB-T - and usually DVB-S - is unidirectional by nature. This has multiple implications for interaction. Interaction can be done in different ways, namely:

- with an electronic program guide, but this only offers selection within the already transmitted streams;
- with a virtual return channel based on unidirectional link routing, leading to additional requirements to the mobility system, QoS, and A4C, and the need to cater for the unavailability of a return channel;
- with a separate bi-directional connection for interactivity, while the big data chunks are received over the broadcast medium. This will move the channel selection from the data level to the service level and would not put additional requirements on QoS and A4C .

Broadcast radio networks are typically used to provide point-to-multipoint services. Therefore, multicast communication has to be provided by the communication layers. New requirements on the multicast protocols for the seamless integration of broadcast include seamless handover for multicast groups, including inter-technology handover, local repair to cope with variations on the different radio links, and support for temporary unavailability of a return channel due to the support for mobile environments and even mobile networks. Daidalos has included high-level services that are traditionally only provided in the radio part of broadcast networks to the IPv6 based network. is also developed mechanisms for dynamic modification of the transmission mode (Unicast vs. multicast), the guaranteeing of appropriate QoS, and the adaptation of broadcast content for some fraction of the receiving user group. The latter supports many users wanting to receive a soccer video at the same time, but only some of them are willing to pay for the HDTV version. Implementation of the key concept also includes the support for broadband distribution of multimedia content to huge groups of users, as well as personalised interactive broadband multimedia sessions. We attempt to design a system, which is capable to select the transmission media (non-broadcast and broadcast technologies) to reach the user, and to select the transmission mode (unicast, multicast, or broadcast) according to the service, and the set of users.

Efficient personalized content distribution (see Section 4.2) was developed in the context of this key concept and can offer application level multicast to a multitude of users. Sharing of Wireless Sensor and Actuator Networks (WSANs) among multiple applications can utilize application level broadcast in a similar manner.

5.1.1.5 Federation

The Daidalos design strongly relies on Federation concepts (see Section 2.3 and 2.4). Daidalos business approach allows for a multitude of business roles: network providers, service providers, aggregators and platform providers. In a real environment, these functional roles may be covered by different companies. Different business scenarios can be described by presenting different interrelationships (like ownership) between these roles, from monolithic operators that execute most of these roles to small micro-operators that provide only network access and rely on other operators for other functions. In Daidalos, the federation concept is the glue for this flexibility of scenarios.

Daidalos defines 'federation' as the existence of a trust and responsibility delegation between different entities in the network. These may be different

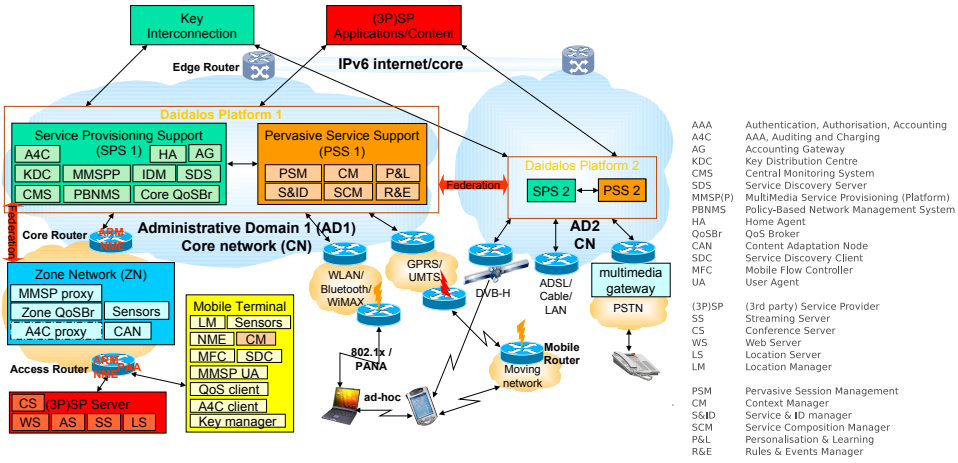
parts of the same operator infrastructure, or parts under the administrative ownership of other operators. One example of the former is the management delegation that exists between an Authentication server (part of A4C) and a network manager (QoS Broker), where the server trusts the manager to perform the network-level control functions required for a specific user, as registered in the authentication server. Daidalos assumes this type of trust relationships exist between the equipments owned by the same operator - and naturally a secure management infrastructure should be in place to assure this.

A wide range of cases from closely-federated to non-federated has to be supported, as relationships will cover different degrees and types of information. Daidalos enables companies with different levels of trust to federate with each other in a controlled way. A federation framework allows entities to dynamically establish agreements with each other. This diversity of federation relationships may exist between traditional operators (horizontal federations) and between operators and service providers (vertical federations).

This leads to a horizontalization of service provisioning with several categories of providers, such as access operators, core operators and service aggregators including mobile virtual operators and Value-added Service Providers (VASPs). These providers will specialize in their specific markets, and establish service relationships with complementary operators in order to be able to provide a complete service solution for their customers. With a service negotiation infrastructure in place, this environment can be fairly dynamic, providing facilities for the operators to exploit market competition to reduce total costs. A reliable trust infrastructure has to be in place - potentially by an independent, federated, operator - including key distribution mechanisms, and Service Discovery Servers. In general, federation does not mean the exchange of all information or control, but only the essential elements required for an integrated service solution across operators to be provided to a specific user. Bringing pervasive services into play means that even more information can be shared (for instance user context and preferences), and a truly cooperative environment will be required between potentially different types of operators and service providers. Regarding peering agreements, such as those used to enable roaming support, Daidalos users will be able to handover their network connectivity and sessions across operators given the existence of adequate federation agreements between those operators. These handovers will necessarily be slower in execution, given the required inter-operator trust checks, but can be often prepared in advance, and thus can be realized for real-time communications.

5.1.2 The Daidalos architecture

There is a lot of related work in the future networks area. 3GPP aims to evolve the 3G network to an All-IP network concept, but does not address pervasiveness, broadcast and universal identity concepts. The IETF addresses several protocols that solve some issues addressed above, but does not provide an overall architecture. Some EU projects, such as Ambient Networks, take a network centric view, while not integrating their networking concepts with overall service needs of the user. On the other hand, other projects are even more specialized. For instance, the EU IST MAGNET Project focuses on personal networking, while others are primarily concerned with security aspects, such as the PRIME project. We believe Daidalos is the first project developing a comprehensive solution for the future scenario described above, covering several layers and viewpoints and making services easy and pervasive for the user.



AAA	Authentication, Authorisation, Accounting
A4C	AAA, Auditing and Charging
AG	Accounting Gateway
KDC	Key Distribution Centre
CMS	Central Monitoring System
SDS	Service Discovery Server
MMSP(P)	MultiMedia Service Provisioning (Platform)
PBNMS	Policy-Based Network Management System
HA	Home Agent
QoSBr	QoS Broker
CAN	Content Adaptation Node
SDC	Service Discovery Client
MFC	Mobile Flow Controller
UA	User Agent
(3P)SP	(3rd party) Service Provider
SS	Streaming Server
CS	Conference Server
WS	Web Server
LS	Location Server
LM	Location Manager
PSM	Pervasive Session Management
CM	Context Manager
S&ID	Service & ID manager
SCM	Service Composition Manager
P&L	Personalisation & Learning
R&E	Rules & Events Manager

(a) Architecture (b) Acronyms

Figure 5.4: Daidalos network and service architecture

Figure 5.4a shows the Daidalos architecture, which has embedded the five key concepts at different levels in its components. In contrast to 3G networks, Daidalos is a pure IP solution, with routers and servers replacing or complementing specialized 3G equipment. At the bottom the Daidalos architecture

shows the (mobile) devices, for both end-users and service providers. On top of that it offers multiple access technologies via access routers. The Daidalos platform above that offers service provisioning and support for pervasiveness across those access networks and terminals. The service provisioning includes identity, QoS, service discovery and multimedia enablers for pervasive services and third-party services. The pervasive service support offers context management and reasoning, service composition, personalisation and learning, and pervasive session management. The Daidalos platform provides horizontal federation between different administrative domains, and vertical federation between different operators (for access and core network, and the Daidalos platform) and service providers. Service providers can be either parties that offer services in the lower layers, like providing context such as location and those from sensor networks, in the middle layers with for instance SIP application servers, and in the higher layers with context-aware web services, and multimedia content.

The major impact of the Daidalos concepts and integrated architecture is working towards more open and flexible architectures for next-generation context-aware heterogeneous networks that enable both competition and cooperation of network, platform and service providers. It further offers context-awareness in all architecture layers, enabling a pervasive user experience to its users.

5.2 Reuse of pervasive systems

A great number of pervasive systems have emerged over the last decade, most of which serve dedicated application areas such as environmental monitoring, supply chain monitoring and remote health monitoring [64, 36, 26, 99]. Some generic architectures have been developed in research projects that unleash the potential of pervasive systems to multiple applications. Unfortunately, none of these generic architectures have matured into a widely accepted architecture yet.

Our main objective is to provide a mechanism for shared usage of pervasive systems in multiple applications. This is because for deployment of existing systems in a new context and developing new systems, the effective use of legacy systems turns out to be tedious. It is difficult to exploit opportunities of reuse and to identify and implement effective measures for modifications. Recently, technologies like web services and Service Oriented Architecture (SOA) have been adopted in pervasive systems architectures for service discovery and service composition. In some cases [45] these technologies have even been used within

clusters of sensor nodes in a pervasive system. Although these technologies offer great flexibility, they can jeopardise the efficiency of these systems (see Section 5.2.3.7).

In general, a common reasoning framework in which existing pervasive systems architectures can be assessed in multiple dimensions is missing. Such a framework would allow amongst others for: (i) comparing existing systems; (ii) effective integration of heterogeneous systems and modification of existing systems; (iii) easy system deployment in new contexts; (iv) identification of and fixing weak spots.

Unified Modelling Language [110] (UML), a well-known and widely used modelling language, unfortunately does not yet provide a view in which the important properties of pervasive systems can be modelled together in a comprehensive way. Furthermore its precise notation adds a level of detail that is not required for the conceptual framework advantages listed above, and can therefore be distracting.

To this end, it is our objective to create a simple conceptual framework for reasoning about pervasive systems, as well as for comparing and integrating such systems. For doing so, it is obviously essential to derive the key entities and principles of pervasive system architectures and their modules. Therefore, we identify generic entities, called *resources*, and interactions between them found at different levels of detail in pervasive systems. The general applicability of these resources is validated by decomposing a number of generic pervasive systems into their *resource compositions* in section 5.2.3. A resource composition gives a quick overview of the resource interaction at different levels, to which we refer as the *communication view*.

In addition to the communication view of each pervasive system, we analyse the required properties for scalability, efficiency and pervasiveness in different application areas. These properties are also used to compare the support of the analysed pervasive systems. From these comparisons and observations, the disadvantages of these systems become apparent, which helps us to define a guideline towards an overall architecture that diminishes the identified weaknesses. In section 5.2.4, an improved pervasive system is created from existing ones to show the strength of our reasoning framework, and to identify the integration points.

To summarise, the contributions are threefold: (i) we propose a conceptual framework for pervasive systems taking its principles and entities into account (Section 5.2.2) and detail the properties of pervasive systems in different application areas, (ii) we use this framework to decompose and compare a number of existing pervasive system architectures on the required properties of differ-

ent application areas (Section 5.2.3), (iii) we use this framework to combine existing pervasive systems into improved ones and indicate integration points (Section 5.2.4). We conclude with directions towards a flexible architecture for shared use of pervasive systems (Section 5.2.6).

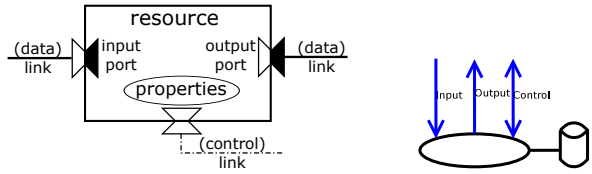
5.2.1 Related work

Middleware design for WSANs helps separate the concerns of applications, low-level networking, and heterogeneous hardware. A recent survey [156] highlights advantages of WSAN middleware: providing system abstraction, shared and deployable functionality, and resource management. As noted by the authors, the main WSAN challenges, i.e. context-awareness and data-centricity, have not yet properly been addressed. Recently, the WSAN middleware design direction shifted towards web-based and service oriented architectures [81, 112, 84].

Another survey [103] addresses existing pervasive architectures and evaluates them in terms of context abstraction level, communication model, reasoning system, extensibility, and reusability. The authors conclude that most of the current architectures are centralised and application-dependent, and suffer from lack of generality and a single point of failure. Here, we analyse more generic pervasive architectures and explicitly model the communication type and make it explicit where combined systems can interface. Gajjar et al. [64] give design directions for WSANs in different application areas. Here we focus on architectures using WSAN and extend their work with architecture requirements per application area.

Our proposed conceptual reasoning framework builds on modelling concepts and model-driven design. UML provides a good starting point for modelling pervasive systems. Unfortunately, it only provides limited support for modelling behavioural patterns like synchronous and asynchronous communication, push and pull. A recent modelling paper [80] proposes to use stereotypes to express these architectural primitives, an idea that we adopt in our framework.

Another UML-based modelling language for pervasive systems is PervML [76]. It is used to visually and platform-independently model a pervasive system from the point of view of both system analysts and system developers. It was later extended to also allow conceptual representation of context and context inference in real-time [56] as well as automatic generation of system code [57]. Since PervML targets the generation of Open Services Gateway Initiative (OSGi) code and only two example models are available, its generality to model any kind of pervasive system is not obvious.



(a) Main Entities (b) Interactions

Figure 5.5: Main entities and interactions of resource.

5.2.2 Flexible pervasive systems architecture

We propose a flexible conceptual framework that allows: (i) assessment of pervasive systems using a common notation, and (ii) identification of how to incorporate different modules into an existing system. In what follows, we explain the concept and building blocks of our conceptual framework as well as their types and properties, and analyse them in different application areas.

5.2.2.1 Conceptual framework

Our conceptual framework builds on the concept of *entities.* An entity is identifiable and consequently it has a set of properties. Some properties of an entity can be good in one context and may be bad in another. The main entities that can be identified in pervasive systems (and communication systems in general) are: *resource, port,* and *link.* A basic resource has an input port and output port and transforms input events or streams into output events or streams. Links can carry data or control signals. Data links connect output ports to input ports, whereas control links are used to schedule actions or access the properties of an entity.

Figure 5.5a illustrates the principal entities of a resource. One may notice that not all details of this model are always equally relevant. Therefore, in the remainder of this chapter, we will use a compressed form as shown in Figure 5.5b. The model implicitly holds properties in a container, which are made explicit whenever appropriate.

Resources may be compound or basic. A compound resource can contain other resources that are connected via links. Figure 5.6b provides an example of a compound resource. When a compound resource uses other external ports than the ones contained resources require or provide, a transformation is required. A dedicated resource called *transformer* is introduced that takes care

of this transformation and makes it more explicit. To distinguish the transformer from other resources, it is drawn as a rectangle instead of an ellipse.

The operation of a compound resource is coordinated by a *manager* , which is a dedicated resource that can manage the configuration of the contained resources, their execution, resource scheduling and maintenance. Note that there can only be one manager per compound resource. In embedded devices, the manager is usually the operating system or scheduler. In bigger software systems, an execution environment, event loop or scheduler are also candidate managers. To distinguish the manager from other resources, it is depicted as a rectangle with rounded edges.

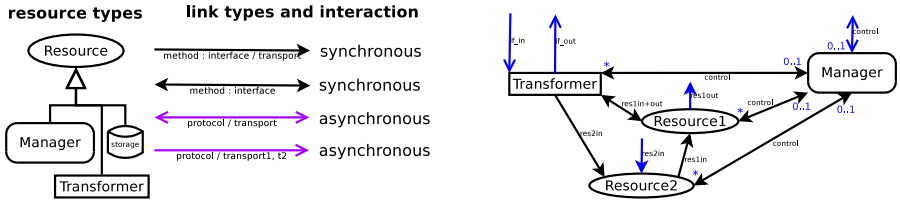
Figure 5.6a provides an overview of all the resource and link types. It also shows the inheritance relation between the different resource types. Storage is added to denote a resource dedicated to information storage and retrieval. The link types and notation of interaction protocols and interfaces are further detailed below.

In order to make architecture comparisons presented in the following sections easier, we will use our compressed notation and borrow notations like multiplicity and inheritance from UML where appropriate (appropriate notations can come from different types of UML diagrams). Multiplicity is denoted by numbers on either end of the communication link between resources (like UML associations in a class diagram) and specifies how many instances can exist on each side of the relation. Note that the **communication link** between resources are drawn as in UML sequence diagrams:

- an **open arrow head** denotes an asynchronous message. Asynchronous messages are normally used to push information (including requests). The "pull" stereotype [80] can be added to the link to explicitly specify information pull.
- a **solid arrow head** denotes a synchronous message, the corresponding return message is not drawn. Synchronous messages are mostly used to pull information and can contain information in the request message. The "push" stereotype [80] can be used to explicitly specify information push.

The **interaction *type*** is specified as a label on the communication link between resources, using the following EBNF notation:

$$type = [M\text{“:”}]P[\text{“/”}T]$$
$$M = (method|message)[\text{“,”}(method|message)]*$$
$$P = (protocol|interface)[\text{“,”}(protocol|interface)]*$$
$$T = transport[\text{“,”}transport]*$$



(a) Communication view notation

(b) Compound resource

Figure 5.6: Notation and compound resource example

A *method* is usually a synchronous call to a resource according to an *interface* specification. A *message* is usually an asynchronous *protocol* message. *Protocol*s and *interface*s can be used over different *transport*s, such as TCP, SSL, HTTP, Simple Object Access Protocol [105] (SOAP) and RS232.

In Figure 5.6b, the "0..1" on the *Manager* side specifies that *Resource1*, *Resource2* and *Interface* can only have 0 or 1 managers, just like associations in UML class diagrams. The *Manager*, on the other hand, can control any number of *Resource1*, *Resource2* and *Transformer*, denoted by "*" on the connecting arrow. Inheritance is denoted by a directed hollow arrowhead as shown in Figure 5.6a), just like in UML class diagrams. Stereotypes can be added to the arrow to specify the type of inheritance. From now on we will only explicitly model the multiplicity where this adds value, e.g. when there are many-many or one-many relations. From here onwards, the compressed notation is called the communication view.

Since different groupings of resources into compound resources are possible, a similar grouping should be made to allow for comparison and integration of models. The models in this chapter therefore use a grouping that at the highest level identifies components that communicate over TCP/IP, serial lines or via radio links.

5.2.2.2 Resource types

Pervasive systems tend to have recurring and hierarchical categories of resources. The main categories include: Sensors, Actuators, Repositories, and Services. Sensors observe physical signals such as light, temperature and movement, whereas Actuators control these physical signals. Repositories are used to store parameters, services, programmes, and other data. Finally, a Service is

a (possible) combination of sensors, actuators, or repositories.

Transformers are often bi-directional and may be in the form of Software–Software, Hardware–Hardware, Hardware–Software, and Human–Machine. A software–software transformer transforms data from one domain to another, for example a software module. Hardware–hardware transformers are merely physical channels, whereas hardware–software transformers can for instance be a transition between analogue continuous domain to digital quantised domain; sensors and actuators fall in this category. A human-machine example is a Graphical User Interface (GUI) that shows a graphical representation on a screen or captures user actions from input devices like a mouse and a keyboard.

5.2.2.3 Properties of pervasive systems

To date, pervasive systems have been used in various contexts and still have the potential to be deployed in many more. These systems present different properties and attributes, which make them suitable for specific application areas or contexts. Based on our own experience and literature study (e.g., [135, 88, 80, 157]), we identify the following groups of properties (scalability, efficiency, pervasiveness, and maintainability) that characterize pervasive systems:

Scalability Scalability of a system is high when it can grow while still being manageable and responsive. Scalability is low otherwise. For scalability we define the following properties and parameters (inspired by [135] amongst others):

- **Resource binding**: Explicit binding between applications and resources is discouraged, since it requires a WSAN node to signal and process on a per application basis. This is related to the notion of localised scalability [135]. The binding can be *tightly* coupled, *loosely* coupled, or *uncoupled*. A looser binding enhances scalability.

- **Node configuration**: Examples of WSAN resource configuration are setting sampling and reporting periods, and alarm thresholds. The most static case is that it requires *reprogramming* of the node. *Online* configuration using an open protocol is considered better. The most scalable is *controlled* configuration, in which authentication and authorisation are checked and concurrent requests are managed. This is related to the notion of Effective Use of Smart Spaces [135]. When possible, also the typical frequency of re-configurations is shown.

- **IP Reachability**: IP Reachability refers to being able to work through firewall and private network boundaries (and associated Network Address Translation (NAT) traversal) to reach a WSAN (e.g. for remote node configuration or actuation). When components are not reachable via IP, additional measures are required per installation (for instance adding firewall rules, or STUN) to make the system work across the Internet, which makes the system less scalable. When possible, also the typical frequency of remote maintenance is shown.

- **IP Mobility**: IP Mobility controls the changes of network attachment of devices and their contained applications, services, and context sources. Mobility is an integral part of pervasive systems [135]. Scalability increases when devices can remain connected while moving around or reconnect without much deployment or communication overhead. When possible, also the typical number of mobility changes is specified.

- **WSAN Mobility**: WSAN Mobility refers to changes of attachment of sensor and actuator nodes within and across WSANs. Scalability increases when nodes can move freely within the WSAN coverage area while still being reachable and able to send measurements. Scalability increases even more when the WSAN supports multiple hops and when nodes can also move between WSANs. When possible, also the typical number of mobility changes is specified.

- **#Domains**: Estimated number of supported peering domains or clusters. Peering enables applications to transparently use resources from another domain or cluster, and increases scalability.

- **#WSANs/domain**: Estimated number of WSANs supported per domain or cluster (or in the overall system when peering is not supported).

- **#Nodes/WSAN**: Estimated number of nodes supported, per WSAN when applicable, else in the overall system.

- **#Apps/node**: Supported number of concurrent applications using sensor information from one or more nodes. This is typically 1 for dedicated pervasive systems. Increasing this number may limit the efficiency of the interactions (see below).

Efficiency Efficiency of a system increases when latency and bandwidth overhead in communication decrease. With respect to efficiency we define the following properties (inspired by [156, 157] amongst others):

- **Application type**: The application type can be *generic* or *dedicated* and can run close to the WSAN (*local*) or elsewhere. A dedicated pervasive application can be fine-tuned to be more efficient than a generic pervasive system that has support for many different applications running in parallel. When possible also the typical number of applications is shown per application type.

- **Dependencies**: Systems may be stand-alone or depend on another component, like *WSAN*, *WAN*/*LAN* connectivity, and applications (*apps*). A stand-alone system can be more efficient since it does not need to interact.

- **Interaction complexity**: The order of complexity of the interactions is defined in terms of number and size of messaging required. Lower interaction complexity leads to improved efficiency.

- **WSAN link**: This refers to the interaction protocol that is used among WSAN nodes (including the gateway). Messages are sent either with or without a response (acknowledgement) message. Bidirectional asynchronous messaging is the preferred type of interaction, since it allows reaching dormant WSAN nodes statelessly and thus reduces the overhead in WSAN nodes. When undetermined, *proprietary*, *de facto* or *open* is used to indicate the typical level of standardization.

- **Application link**: This refers to the interaction protocol between the application and the WSAN or device. Asynchronous messaging is a natural extension of WSAN messaging, whereas synchronous messaging, like request/response in web services, involves additional logic (processing of responses) and associated energy consumption and bandwidth. When undetermined, *proprietary*, *de facto*, or *open* is used to indicate the typical level of standardization.

- **WSAN multicast**: This refers to the ability to efficiently send messages to a group or to all nodes in a WSAN. Link layer broadcast and/or multicast support for messages towards WSAN nodes makes communication in WSANs much more efficient in terms of energy consumption and bandwidth. When undetermined, the *beneficial* effect of multicast can be indicated.

Pervasiveness Availability of the following properties make the system more pervasive [88, 103] (note that mobility is already covered under scalability):

- **Sensing**: Specifies whether sensing is supported.
- **Actuation**: Specifies support for actuation. Actuation may also be controlled (authenticated, authorised and management of multiple requests).
- **Reasoning**: Reasoning refers to aggregation, merging of sensor data or context. Reasoning can be done through algorithms, programs, rules or inferencing in different parts of the system.

Maintainability Maintainability properties relate to the changeability of components, and the maturity/stability of the system and its components:

- **Changeability**: Specifies the minimum level of interchangeability and stability of hardware, ranging from prototype, specific, various (i.e. a mix of specific and standardized) to standardized hardware.
- **Maturity**: A pervasive system can be a model, a simulation, a prototype, or a commercially available product.

5.2.2.4 Typical application area requirements

This section describes the typical needs of different application areas with respect to pervasive systems, using the properties described in Section 5.2.2.3. In Section 2.1.2 we defined the application areas cool chain logistics, surveillance, smart spaces, and remote eHealth.

Since the use of sensor networks in most application areas is still in the early stages of deployment, the requirements on the different properties is an educated guess based on a number of papers [64, 36, 26, 99],

The typical items used in the application areas are shown in Table 2.1. The typical application area requirements are depicted in Table 5.1. The specified values for scalability, efficiency, pervasiveness and maintainability properties depict the minimum workable option in each application area. For scalability, four new parameters have been added for a WSAN. The *IP transfer period* indicates how often sensor updates are made available to applications. The *Sampling period* indicates how often WSAN nodes take measurements. The *WSAN IP messages per second* equals *Number of nodes per WSAN* / (60 * *Sampling period)*, multiplied by the *Number of WSANs per domain* this gives the *Domain IP messages per second*.

Table 5.1: Typical requirements for different application areas

Framework /property	Cool chain logistics	Environmental monitoring	Surveillance	Smart spaces	Remote eHealth
Resource binding	loosely	none	loosely	loosely	loosely
Node config	monthly	monthly	yearly	per minute	weekly
IP Mobility	#reconnects	#reconnects	#reconnects	#reconnects	#reconnects
WSAN Mobility	#movements	#movements	#movements	#movements	#movements
IP Reachability	monthly	monthly	hourly	per minute	weekly
IP Transf period	per minute	per day	per minute	per minute	per hour
#domains	$\leqslant$100	$\leqslant$1000	$\leqslant$100	$\leqslant 10^6$	$\leqslant$1000
#WSANs/domain	$\leqslant$1000	$\leqslant$1000	$\leqslant$10	$\leqslant$10	$\leqslant$10000
#Nodes/WSAN	$\leqslant$2000	$\leqslant$10000	$\leqslant$100	$\leqslant$100	$\leqslant$10
#Apps/node	$\leqslant$10	$\leqslant$100	$\leqslant$10	$\leqslant$100	$\leqslant$10
Sampling period	10 minutes	30 minutes	1 minute	1 minute	10 minutes
WSAN IP mesg/s	$\leqslant$3.333	$\leqslant$5.556	$\leqslant$1.667	$\leqslant$1.667	$\leqslant$0.01667
Domain IP mesg/s	$\leqslant$3333	$\leqslant$5556	$\leqslant$16.67	$\leqslant$16.67	$\leqslant$166.7
Scalability	medium-high	high	low-medium	medium	medium
Application type	dedicated, optional local	$\leqslant$10 dedicated	dedicated local, optional central	$\leqslant$100 generic	dedicated, optional local
Dependencies	all	all	all	all	all
Inter.complexity	low-medium	low	medium	medium	medium
Application link	proprietary	standard	proprietary	standard	standard
WSAN link	proprietary	proprietary	proprietary	proprietary	proprietary
WSAN multicast	benefits	benefits	no	benefits	no
Efficiency	medium/high	high	medium	medium	low/medium
Sensing	+	+	+	+	+
Actuation	–	–	+/–	+	+
Reasoning	–	+/–	+	+	+
Pervasiveness	low	medium	high	high	high
Changeability	standardized	various	specific	various	specific
Maturity	product	prototype	product	prototype	product
Maintenance	high	low	medium	low	medium

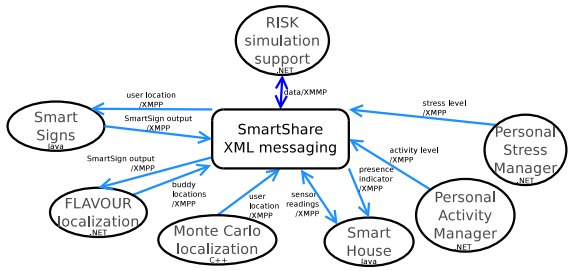


Figure 5.7: Smart Surroundings communication view

5.2.3 Generic pervasive systems

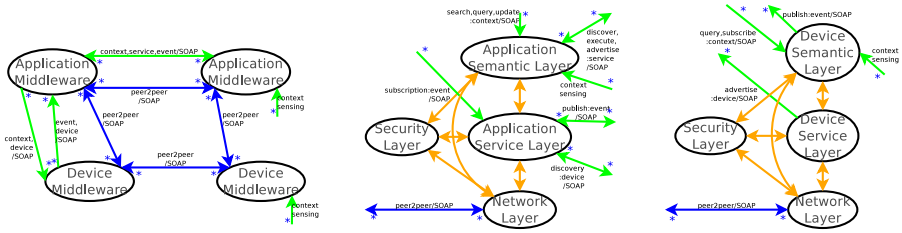
This section introduces a number of generic pervasive system architectures. It uses the conceptual framework using a similar grouping and the properties from Section 5.2.2 to decompose and compare them.

5.2.3.1 Smart Surroundings

One of the main goals of the Smart Surroundings project [141] was to support many diverse ubiquitous and context-aware applications concurrently and remain open towards unforeseen use cases. The architecture design aimed at enabling execution of many software components developed in different languages and running on different platforms across distributed and embedded devices and technology platforms. Its integration technique has two steps, i.e., (i) defining an eXtensible Markup Language (XML)-like message exchange interface on top of TCP/IP, and (ii) designing a publish-subscribe mechanism on top of Extensible Messaging and Presence Protocol [132] (XMPP). Result of this project is an integrated demonstrator composed of multiple heterogeneous prototypes and systems developed within the project that exchange their real-time data. Figure 5.7 illustrates the communication view of Smart Surroundings.

5.2.3.2 Hydra

The goal of the Hydra [70] project is to develop a middleware that enables any device to be detectable and usable from Hydra applications and to develop tools for solution providers of ambient intelligent applications using such devices. A complementary goal is developing tools for device producers to enable their devices to be part of an ambient intelligence environment.



(a) Hydra interactions (b) Application middleware (c) Device middleware

Figure 5.8: Hydra communication view

The Hydra architecture is divided into middleware[1] for applications and devices. This middleware has three stacked layers: network, service, semantic, and one cross-layer for security. The middleware uses web services based on SOA and the semantic web. The service and semantic layers make devices available as services that advertise their capabilities and properties using a device ontology. Applications can then search and use these device services remotely and context-aware applications and workflows can be composed of application and device services. Additionally, applications can subscribe to context or other middleware events related to a specific topic, independently from the application or device that generates them. The communication view in Figure 5.8 depicts how application and device middleware resources in Hydra interact.

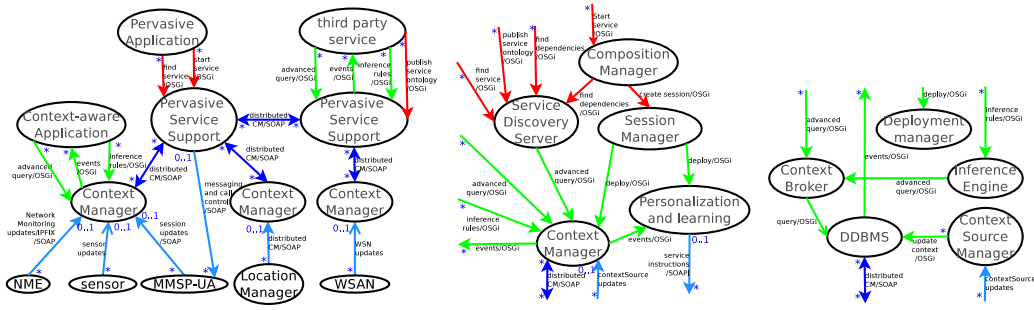
5.2.3.3 Daidalos

The goal of Daidalos [16] (see Section 5.1 was to enable seamless pervasive access to content and services via heterogeneous networks that support user preferences and context. Daidalos demonstrated its integrated concepts in December 2008, in which a prototype from the UbiSec&Sens [151] project was used as sensor network.

The Daidalos architecture enables federation between multiple domains; each domain hosts a Daidalos platform that consists of service provisioning and pervasive service support. The corresponding communication view is depicted in Figure 5.9a.

Context information can have various forms in Daidalos, i.e., it can be from

[1]Hydra middleware has been renamed to LinkSmart middleware.



(a) Daidalos interactions (b) Pervasive Service Support (c) ContextManager

Figure 5.9: Daidalos context related communication view

a Network Monitoring Entity (NME) in the Mobile Terminal or in one of the routers, from the SIP user agent (MMSP-UA) or sensors in the terminal, it can also be sensor information from a device in the Access Network. Additionally, a complete WSAN can be attached to one of the Access Routers.

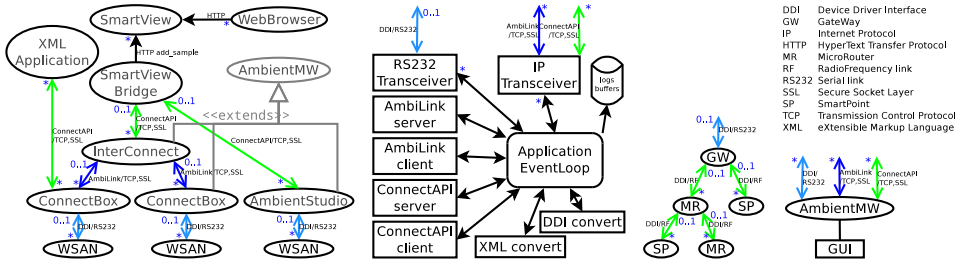
Context Managers (CMs) in both Mobile Terminal, Access and Core Network provide a means to stream context information from all context sources to a distributed database. Also higher-level context information can be stored or inferred. In these Context Managers, context is stored using ontologies. Applications can query and obtain context information from any of the Context Managers when they are permitted based on their credentials.

5.2.3.4 Ambient integration middleware

Recently Ambient middleware [36] was developed to enable easy integration with applications and to enable remote monitoring and maintenance. The communication view in Figure 5.10a shows the interaction between the different components. Note that AmbientStudio, the ConnectBox and Interconnect all share the same Ambient middleware (AmbientMW).

Multiple WSAN Gateways (GW) can be connected via RS232 using the AmbientMW in a ConnectBox device or AmbientStudio on a PC. This makes GWs, MicroRouters (MR) and SmartPoints (SP) remotely available.

The AmbientMW offers the ConnectAPI to ease integration with third-party applications using asynchronous XML messages over a TCP/IP connection (op-



(a) Ambient Middleware interaction (b) Ambient middleware (AmbientMW) (c) WSAN (d) AmbientStudio

Figure 5.10: Ambient Middleware communication view

tionally encrypted with Secure Socket Layer [63] (SSL)).

The AmbientMW also offers AmbiLink to ease remote monitoring and maintenance of sensor networks using compact asynchronous binary messages over optionally SSL encrypted TCP/IP connections.

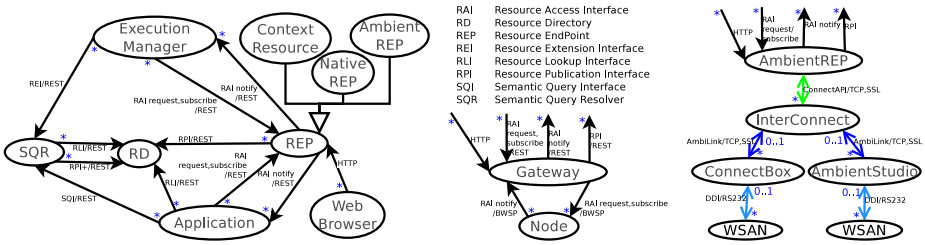
The SmartView web application enables monitoring temperature, humidity and/or shelf-life of static location or cargo with an attached SmartPoint. The SmartView Bridge uses the ConnectAPI to receive sensor measurements from AmbientStudio or ConnectBoxes and forwards the samples to SmartView.

5.2.3.5 SENSEI

The Integrated European project SENSEI [45] developed a common framework to make WSANs available to services and applications. SENSEI can handle sensor, actuator, and context resources and is able to create compound resources from basic ones. It also provides a common interface for integrating existing sensor platforms.

SENSEI uses a Representational State Transfer (REST)-style architecture over HTTP, which allows sending requests and returning a response. Requests and responses can be simple text, but also complete XML documents. Resources are identified with a URL, which can be used to request the resource information, or to post a command.

The interaction between the different SENSEI components is depicted in Figure 5.11a. The project created prototypes for most components and has demonstrated their integration.



(a) SENSEI interactions (b) Native REP (c) Ambient REP

Figure 5.11: SENSEI communication view

5.2.3.6 Native SENSEI

In a native SENSEI sensor network, a similar REST-style communication is used between the SENSEI Gateway and the Nodes in the sensor network using IPv6 over Low power Wireless Personal Access Networks [106] (6LoWPAN). The SENSEI Gateway translates REST requests/responses to/from the Binary Web Service Protocol (BWSP), which is used inside the sensor network. BWSP uses User Datagram Protocol (UDP) as transport and can optionally disable generation of response messages from the gateway. XML messages are greatly reduced in size using Efficient XML Interchange [155] (EXI). As a future option, the SENSEI Gateway may also mediate when multiple subscriptions are done for the same Node resource. The interaction between the Native SENSEI Gateway and Node is depicted in Figure 5.11b.

AmbientREP The Ambient Resource EndPoint (AmbientREP) connects Ambient sensor networks with the SENSEI framework via the AmbientMW (see Section 5.2.3.4). So far, temperature, humidity, battery level, and location have been integrated in the AmbientREP using a subset of the wide variety of Device Driver Interface (DDI) drivers available on SmartPoints, MicroRouters, and Gateway. Also LEDs can be turned on the MicroRouters and Gateway. The interaction between the different components is depicted in Figure 5.11c.

Since the underlying Ambient network provides asynchronous updates of sensors, the AmbientREP simply remembers the latest measurement and requests a new sample when no sample is available yet. For more frequent sensor updates, an application can subscribe for sensor updates and will be notified

when a matching DDI message is received by AmbientREP.

5.2.3.7 Reflection and observations

This section reflects on how the generic pervasive systems perform on the properties defined in Section 5.2.2.3. Furthermore it describes the observations that can be made when comparing these systems.

Daidalos Daidalos is very scalable, both horizontally between domains and vertically between different access technologies, networks, and third party application providers. Daidalos treats WSANs as just another context source and supports distributing this context via a chain of context managers within and across domains. It does not yet consider actuators in the WSAN, although it does send feedback to network management entities. With respect to reasoning, there is context inference from low-level context to more complex notions describing the environment. Daidalos requires creation of a Context Source Manager to send sensor/context to a Context Manager which allows various WSANs to be integrated. An application can both subscribe to context events and request specific context. The order of interaction complexity of the Daidalos system is low, as sensor information is just fed into the context managers without delay. Interested applications will then get an (optionally inferred) context update. There is no direct coupling between context producers and context consumers and configuration/management of WSANs is not yet considered.

Smart Surroundings Application mobility is supported in the sense that application can run anywhere and on any type of platform as long as it can connect to the Smart Surroundings framework. Since XMPP supports a chain of servers between clients to exchange messages over several hops, private networks with NAT and firewalls can be easily traversed. Simple asynchronous messaging in Smart Surroundings through XMPP offers low interaction complexity. However there is a tight binding between context producers and applications using the context. In addition parallel execution of applications is not supported. Therefore the scalability is low. Reasoning occurs at the application level and is not supported as a feature of the systems itself.

Hydra Subscribing for types of notifications is powerful and unbinds producers from consumers. The distribution of events in Hydra is done at the application layer and therefore multiplies the involved SOAP messages. Hydra

supports most kinds of federation, but does not yet consider sharing of WSAN data between peers. Hydra therefore offers medium scalability; no evaluation results have been found with multiple applications that use a variety of devices concurrently, and WSANs are not yet considered. The order of interaction complexity of Hydra is high, each application and each device needs to publish itself as a web service, all communication between applications and devices is based on SOAP, which adds considerable processing and bandwidth overhead.

Ambient middleware There is no binding between context producers (sensor resources) and context consumers (applications). The mobility support depends on whether the client or server role is used. In principle, an AmbiLink server supports mobility of ConnectBoxes and associated WSANs, while the ConnectAPI client and server offer mobility towards applications. Since a great number of Ambient WSANs can be merged with AmbiLink, made available to multiple applications, and managed remotely, the scalability is medium. The order of interaction complexity is low, as each sensor update goes asynchronously to application(s) without much delay. The communication via AmbiLink adds small latency, since there is an additional hop between application and WSAN, and improves scalability. The communication via ConnectAPI adds some latency by converting to XML and improves flexibility for connecting with applications.

Native SENSEI Since each sensor update needs either arequest to fetch it from the sensor resource or a subscription to the specific sensor resource (i.e. the sensor node), the interaction complexity is high. Context producers and consumers are therefore tightly coupled. The execution manager makes this binding less strong by working on behalf of applications. Mobility is covered across WSANs, this means that moving a node between WSANs does not update subscribers with the new location (e.g. for updating/stopping the subscription). Reachability can be a concern for SENSEI when parts are behind firewalls or in private networks. On the positive side, native SENSEI is able to connect to a number of WSANs and be used by multiple applications and a number of framework components can peer across domains (most notably the resource directory). All things considered, the scalability of native SENSEI is low/medium.

AmbientREP The order of interaction complexity of AmbientREP is medium, since the requests and subscriptions for sensor updates are handled within the

AmbientREP and do not reach the WSAN nodes. Therefore the context producers and consumers are loosely coupled. The scalability of AmbientREP is medium, i.e. a multitude of Ambient WSANs can feed into one AmbientREP and they become available to multiple SENSEI applications, and its resources are locatable across domains via the peering Resource Directory. Other properties are inherited from Ambient middleware and SENSEI.

Comparison and observations The generic pervasive systems architectures comparison are listed in Table 5.2 for scalability, efficiency, pervasiveness and maintainability properties. Comparing these systems yield the following observations:

- **Explicit resource binding**: Two pervasive systems, i.e., Smart Surroundings and native SENSEI, make an explicit **binding between applications and sensor** resources. Such a tight binding is not so bad for small systems, but as the system grows and sensors are used by multiple applications this will impact energy consumption of sensor nodes and consumed bandwidth in both the WSAN and the IP network, and thus hinders scalability.

- **Web services for applications**: Protocols like REST and SOAP offer great flexibility at the application level. Choosing one over the other usually depends on bandwidth and processing constraints, and protocols that are already in use. Requests and subscriptions require reachability of the WSAN that normally sends a stream of measurements. This can be impractical when that WSAN is within a private networks or behind a restrictive firewall, e.g. when the WSAN is in a vehicle. Therefore, a mobile web-enabled WSAN usually polls for reconfiguration and actuation and needs a server for subscriptions and requests. Websockets [67], as proposed in Revision 5 of HTML (HTML5), may alleviate most of these concerns since it offers asynchronous messaging between web services through a web server.

- **Web services within WSANs**: Energy and bandwidth limitations in the WSAN make a poor match with the added complexity of web services. Precious bandwidth and energy is used to handle acknowledgements for sensor measurements and to notify multiple subscribed receivers. Moreover, latency is added by per-message connection set-ups. Protocols like BWSP help reduce the web services overhead in the WSAN and make the

Table 5.2: Comparison of generic pervasive system architectures

Framework /property	Smart Surround.	Hydra	Daidalos	Ambient MW	Native SENSEI	Ambient REP
Resource binding	tightly	loosely	none	none	tightly	loosely
Node config	–	–	–	+	+/–	+ (AmbientMW)
IP Mobility	+	+	++	+/–	+/–	+/–
WSAN Mobility	n.a.	n.a.	external	++	++	++
IP Reachability	overlay	overlay	MobileIPv6	overlay	issues	issues
#Domains	n.a.	n.a.	⩽10	n.a.	⩽10	⩽10
#WSANs/domain	n.a.	n.a.	⩽10	⩽10	⩽10	⩽10
#Nodes/WSAN	⩽10	⩽10	external	⩽2000	⩽100	⩽2000
#Apps/node	⩽10	⩽10	⩽10	⩽100	⩽10	⩽100
Scalability	low	medium	med./high	medium	low/med.	medium
Application type	generic	generic	generic	generic	generic	generic
Dependencies	none	WSAN	WSAN,appl.	WSAN,appl.	appl.	appl,AmbientMW
Inter.complexity	low	medium	low	low	high	medium
Application link	XMPP	SOAP	OSGi	DDI/XML	REST/XML	REST/XML
WSAN link	–	–	various	DDI	BWSP	DDI
WSAN multicast	–	–	–	+	–	+
Efficiency	medium	medium	high	high	low/med.	medium
Sensing	+	+	+	+	+	+
Actuation	+/–	+/–	–	+/–	+	+/–
Reasoning	none	rules	inference	none	rules	rules
Pervasiveness	low	medium	medium	medium	med./high	med./high
Changeability	various	various	various	Ambient	various	Ambient
Maturity	prototype	prototype	prototype	product	prototype	prototype
Maintenance	low	low	low	medium	low	low

communication asynchronous. However, combined with its high interaction complexity, the efficiency of Native SENSEI is still barely enough to support the considered application areas.

- **Instant messaging for applications**: XMPP and SIP for Instant Messaging and Presence Leveraging Extensions [71] (SIMPLE) are standardised and open candidates for conveying **presence information** (which can be any context) across the Internet. However, efficient one-to-many messaging still poses a great scalability problem for both. On the other hand, distributed Context Managers in Daidalos, Eventing in Hydra, and Subscription in SENSEI offer more tailored messaging, driven by context inference or subscriptions. Note that the size of XMPP and SIMPLE messages are quite high, which makes their usage in energy and bandwidth constrained WSAN nodes impractical.
- **Bandwidth limitations**: The uplink from a WSAN towards the application can have limited (and often costly) bandwidth, for instance a General Packet Radio Service (GPRS) link has typical upload between 9 and 18 kilobit/s (download upto 52 kilobit/s), which severely limits the size and amount of messages that can be sent. GPRS may therefore be too limited to use instant messaging and web protocols with bigger WSANs [36]. Of course a lot can be gained by compression of aggregated messages, this would be practical for messages that are not time-critical. The communication view helps to make the protocol type and the number of messages explicit using the multiplicity on links from sender to receiver. This makes it easier to estimate the required bandwidth in each communication direction.
- **Asynchronous communication**: Only Ambient middleware explicitly uses asynchronous communication between the WSAN, other middleware instances, and applications. On the application link, both OSGi, ConnectAPI, and XMPP provide asynchronous communication, which aids the system scalability and reduces latency since message providers do not need to wait for responses, the provider resources are not necessary tied to its consumers, and messages can more easily be sent to multiple consumers without impacting the producer. Content-based routing [28] could improve efficiency with multiple interested applications.
- **Quality of information**: The importance of accurate information and reliable message transfer differs per application area. Asynchronous protocols could offer the flexibility to only sent acknowledgements for messages

that require it, or sent one acknowledgement for a group of messages. With REST, retries are sent when an acknowledgement does not arrive while the message may have arrived. Compared to REST, SOAP offers additional security features, atomic transactions, and reliable messaging. The stronger reliability is usually only necessary for online purchases, business processes and surveillance. They may however also become important for remote configuration and actuation.

- **Shared control**: Although a number of pervasive systems support **remote actuation and configuration** of WSAN nodes, only SENSEI started defining specific modules that resolve issues when multiple parties are controlling/configuring the same sensor/actuator. Additionally, most pervasive systems lack proper authentication and authority checks to make sure that WSAN configuration and actuation are only invocable with proper credentials. Only SENSEI defines a security mechanism to support fine-grained authentication, authorisation, and accounting. So, only the analysed SENSEI-based systems offer the required pervasiveness properties for surveillance, smart spaces, health and well-being. Smart surroundings and Ambient MW lack the required reasoning for environmental monitoring, but an outlier-detection module [165] may prove enough to satisfy this requirement.

- **Application area scalability**: The generic pervasive systems with tight resource binding are clearly not scalable enough, and only Ambient middleware supports remote configureability without reachability issues but lacks reasoning. Therefore, it makes sense to combine the strengths of pervasive systems (see Section 5.2.4).

- **Mobility of WSANs and their nodes**: Multiple WSANs should be able to coexist in the same area [34]. For sharing sensor information in different applications, compact asynchronous messaging has the best efficiency properties [34].

- **Maintainability**: All analysed generic pervasive systems offer a sufficient platform for deployment in environmental monitoring and smart spaces. Only the Ambient MW provides a sufficient platform for deployment in all considered application areas, except for the required standardization of the hardware platform.

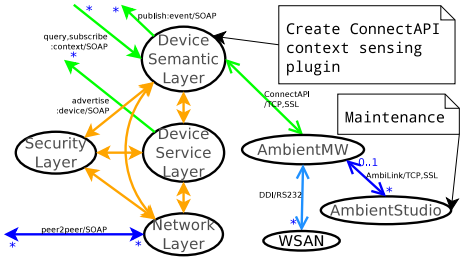


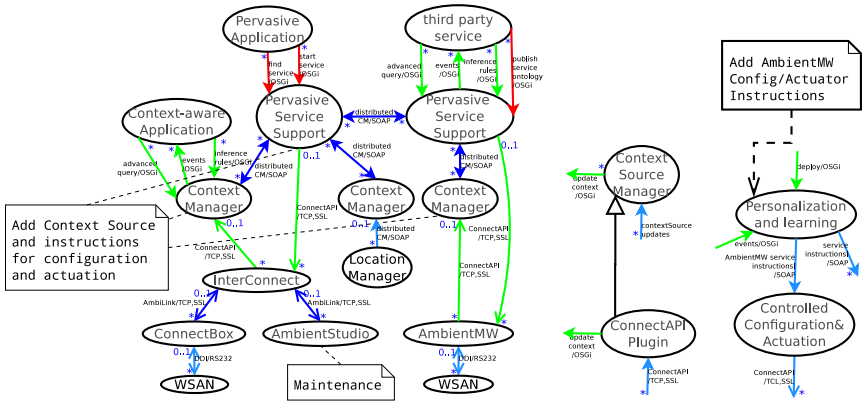
Figure 5.12: Combined Hydra/Ambient device middleware

5.2.4 Combining pervasive systems

This section combines pervasive systems into new ones with combined strength. For integrating different systems it is important that these systems are modelled on the same level of abstraction at the point where interaction between the systems is expected. This means that components involved in this interaction and their candidate links must be explicitly modelled. This will make it easier to analyse if and where transformations need to be done between the systems. The combination of different pervasive systems is done at a conceptual level. For actual integration, the details of protocols and interfaces may need to be modelled as well. Note further that the AmbientREP in Section 5.2.3.5 already is a combination of SENSEI with Ambient middleware.

5.2.4.1 Combining Hydra and Ambient middleware

When combining Hydra with Ambient middleware, the Hydra Device Semantic Layer should be adapted to understand ConnectAPI messages from the AmbientMW as context messages. This most likely requires a plugin in the Device Semantic Layer. The Device Semantic layer can then make this context available for queries and sent events to subscribed applications. How these systems can be combined is shown in Figure 5.12. Since there is no facility defined in Hydra middleware for managing sensor networks, AmbientStudio could be used to maintain them. Alternatively, and more in the peer-to-peer style of Hydra, a capability could be added for controlled and authorised configuration and actuation (combined with authentication and authorisation from the Security Layer) and advertise this capability via the service layer.



(a) Ambient/Daidalos interactions (b) Context source (c) Shared control

Figure 5.13: Combined Daidalos/Ambient communication view

5.2.4.2 Combining Daidalos and Ambient middleware

When combining Daidalos with Ambient middleware, a Context Source Manager instantiation is needed (see Figure 5.9c) to convert ConnectAPI messages to the applicable context type, and let the ContextManager store and/or inference from them. A way to combine these systems is shown in Figure 5.13.

Since Daidalos does not support sensor network configuration, AmbientStudio can be used to maintain them as depicted in the figure. However, it would be better to add functionality for controlled and authorised configuration and actuation to the Personalisation and Learning module of the Pervasive Service Platform using the virtual identity mechanisms of Daidalos and let it expose an interface for actuation and configuration. Therefore, a controlled configuration and actuation component (see Figure 5.13c) is added to Pervasive Service Support (see Figure 5.9b), in order to control service interaction for multiple simulteneous users and translate the service instructions from SOAP to the ConnectAPI protocol.

Since Ambient middleware focuses mostly on remote configuration of and sharing information from mobile WSANs, it is complementary to the Daidalos scalability and efficiency properties. With respect to pervasiveness, Daidalos adds context inferencing and Ambient middleware adds limited support for ac-

tuation. The combined system allows thousands of nodes per WSAN but not enough WSANs per domain to support all the application areas. This is mainly limited by the context manager. In order to support upto 5556 sensor readings per second per domain (see Table 5.1), the scalability can be increased by:

- Distributing the WSANs over multiple context managers. But this will increase operational costs.
- Using node configuration and reasoning within the WSAN to send only meaningful events [164, 26].
- Using context inferencing in the Daidalos context manager to derive meaningful events for applications, such as temperature variations during transport, while (off)loading and during storage.

5.2.5 Recommendations

Although our conceptual framework has been developed for identifying points of integration and modification as well as to compare instantiations of pervasive systems, it can also serve as a source of inspiration to arrive at a global pervasive systems architecture. However, before we can reach that point a number progresses need to be made. To give an impression, here is a list of recommendations:

- Provide an easy way to use context in applications while limiting the binding between context resources. Preferably the context resource should not be aware of applications using it.
- Keep messaging asynchronous as long as possible, preferably extended into the application layer. The intermediate components mentioned for web services should preferably be placed at a globally reachable Internet address, such that bandwidth-constrained uplink of the WSAN and firewalls can be traversed with more light-weight protocols.
- Reduce messaging whenever possible; candidates to do this are: data fusion, filtering, adding intelligence like outlier detection [164] in the WSAN nodes, adding intelligence in the system like inference, event, and activity detection [26].
- Support context history, enabling, among others, detection of slow and rapid changes over time; distributed context storage with a good retention policy is a likely candidate for this.

- Make sure the system also works when part of it is in private networks or behind restrictive firewalls.

- Support any type of context source, actuator, WSANs nodes, simple devices, and applications. Allow remote configuration of them.

- Create a generic reasoning engine that supports flexible ontologies. This reasoning engine needs to be flexible enough to add ontologies for various context sources and inferred context, such as situation, activity, and environment.

5.2.6 Conclusion

We introduced a conceptual framework with flexible building blocks for composing pervasive systems. These building blocks provide an effective way to compare different pervasive systems and modules. We exemplified the use of this conceptual framework by decomposing various pervasive systems. The mapping turned out to be straightforward for the selected pervasive systems. We also showed how easily models can be combined into an integrated model and identified where integration work is required.

Over the last years, good progress has been made in a number of areas with respect to the properties of pervasive systems. Dedicated pervasive systems show excellent properties for their task. The structured approach of our conceptual framework helps identify and evaluate these properties. Further, it clearly indicates opportunities for integration and as such exploits individual qualities of existing systems in merged ones.

Despite the good progress in different application areas, there is not yet a candidate architecture that prevails on all properties. The identified weaknesses in the compared architectures are:

- Synchronous messaging is not very efficient for WSANs. An intermediate component can be used to distribute a stream of asynchronous messages from WSANs to multiple applications (as a web service or by messaging protocols like XMPP or SIMPLE) without having the energy and bandwidth impact on the WSANs and its nodes.

- A number of pervasive systems lack reasoning support, only few support remote WSAN configuration, and most lack support for shared but controlled actuation and configuration.

- Message reduction in size and number is an effective technique to limit resource utilisation. Unfortunately only a few systems have native support for these techniques, which is essential for the scalability of WSANs and its remote usage.
- Most of the pervasive systems are still in prototype stage or dedicated to a specific task. They are often limited to specific sensor data and context. Interoperability between different WSANs is usually not possible.
- Binding sensor resources to applications hinders scalability, and should be avoided.

Progress on these weaknesses and interoperability between pervasive system architectures will enable efficient and scalable architectures for shared use of pervasive systems in multiple applications.

5.3 Conclusions

First we proposed a communication infrastructure for next-generation networks, called Daidalos, that enables personalized, context-aware, composite services to mobile users. The basis for this infrastructure is federation between operators who create a pervasive environment for service provisioning to users and groups, integrated support for mobility and security, virtual identities for users, and resource management. This communication architecture and its components have been modelled in UML, prototyped and integrated in testbeds, and validated using scenarios that utilize its binding concepts.

Next we proposed a conceptual reasoning framework for comparing and integrating pervasive system architectures. This framework enables decomposition of an architecture in its building blocks and makes its interactions explicit. Additionally the required scalability, efficiency, pervasiveness and maintainability properties of a number of architectures are compared. We made a number of observations and created the following recommendation for pervasive system architectures where WSANs are used by multiple applications:

- Do not impose synchronous and verbose protocols on WSAN nodes, but rather translate between WSAN and application protocols near or in the WSAN gateway.
- There is a need for shared and controlled configuration and actuation, which can best be done near or in the WSAN gateway.

- For scalability, messages should be limited in number and size. Interpretation of raw data, data fusion and filtering can already be done within the WSAN resulting in less and to the point messages towards applications.
- Do not bind WSAN nodes to specific applications, this hinders scalability.

Chapter 6

Conclusions

Today's connected networks enable feature-rich applications that adapt according to situational and environmental changes, and the networks and objects that are encountered. However it is a challenge to keep these so-called pervasive applications perform with increasing number of users, increased mobility and captured situational dynamics, and more sensor-enabled environments.

This thesis focuses on federated middleware for efficient sharing of multimedia and sensor information among mobile applications over the Internet. Our objective for mobility is to support transfer of multimedia stream endpoints across devices and mobility of devices, multimedia control and stream endpoints across networks. Our objective for sharing is to support efficient sharing of: a wireless network by applications running on mobile devices, Wireless Sensor and Actuator Networks (WSANs) by multiple applications over the Internet, of multimedia streams by multiple mobile devices. For federated middleware the main focus has been on scalability and efficiency of the TCP/IP interactions. We have defined the following research question:

> *Analyse the tradeoffs in federated middleware for mobility and efficient sharing of networks, multimedia, and WSANs among applications on a multitude of mobile devices.*

In relation to this research question, this thesis proposes and analyses methods for mobility and efficient sharing of networks, multimedia and WSANs, and integrates them in federated pervasive service platforms to enable usage in pervasive applications.

6.1 Contributions

The contributions of this thesis are the following.

1. **QoS support in shared networks** [114], see Section 4.1: We proposed a bandwidth-distribution mechanism for Wireless LAN (WLAN) that uses real-time characteristics of the network medium and feed-forward control mechanisms to regulate the bandwidth distribution. A prototype is developed for WLAN that uses legacy network elements without Quality of Service (QoS) capabilities, with which this mechanism is verified.

2. **Mobility and sharing of wireless sensor networks** [36, 34], see Section 3.3 and 4.3: We analyse the mobility and sharing of wireless sensor networks in logistic and person monitoring scenarios. We provide guidelines for dealing with mobile and overlapping wireless sensor networks and the most promising scheme for sharing wireless sensor networks in applications. A middleware layer is designed and created to support real-time remote monitoring and maintenance of wireless sensor networks in logistic scenarios. Its middleware messaging efficiency is compared with web protocols and improvements are proposed.

3. **Mobility and sharing of multimedia** [29, 13, 153], see Section 3.1, 3.2 and 4.2: We propose a seamless roaming experience in multimedia applications using SIP across heterogeneous networks. We use terminal intelligence to detect and select access networks from federated network operators. We compare MobileIP, SIP and their combination, and share the issues we encountered with our prototype. We propose a network initiated method to distribute a multimedia session over multiple devices in proximity to the user. Its applicability is verified with a prototype, and the combination with a terminal-initiated method is described. We propose to dynamically group multimedia streams from the same origin per network segment based on network characteristics and stream popularity, using relaying or multicast/broadcast when available.

4. **Pervasive service platforms** [29, 15, 17], see Section 2.3 and 5.1: We design, develop and validate a framework for next generation mobility-enabled networks offering seamless roaming with quality guarantees for multimedia sessions, broadcast integration, privacy and anonymity. We propose operator federations to enable personalized, context-aware, composite services to mobile users.

5. **Reuse of pervasive system architectures** [37], see Section 5.2: We propose a conceptual reasoning framework for comparing and integrating pervasive system architectures. This framework enables decomposition of an architecture in its building blocks and makes its interactions explicit. Additionally, the required scalability, efficiency, pervasive and maintainability properties of a number of architectures are compared. We make observations and identify weak spots in current architectures and give recommendations towards flexible pervasive system architectures to be used by multiple applications.

Our hypothesis was that none of the current trends (the Internet of things, Web 2.0 and cloud computing) can satisfy our challenges by themselves, and that in fact a combination of centralisation and decentralisation is required to address the research question.

Regarding the Internet of Things, IP is not necessary for small devices like sensor nodes, since they do not need to be addressable by applications, and the nodes would most likely be sleeping when a request is sent. Authorisation and control for remote configuration and actuation can better be done outside wireless sensor networks than within each resource-constrained node, especially when multiple applications want to use them simultaneously.

Regarding the web service protocols used in Web 2.0, they are still synchronous and quite bandwidth intensive. They form a poor match with sensor nodes that have limited processing and bandwidth, and want to sleep to save energy. Current web protocols were also underperforming for transporting sensor network data over a low-bandwidth connection such as General Packet Radio Service (GPRS). The WebSockets [59, 67] proposed in Revision 5 of HTML (HTML5) are expected to improve this significantly.

Cloud computing alone does not yet offer efficient realtime sharing, since the cloud servers are concentrated in server farms instead of being distributed along the network path between users. This currently means point-to-point connections with the same content from the cloud farm(s) to multiple users. This makes the bandwidth and processing needs proportional to the number of users that use the same content. It is more efficient to distribute the data in a multicast fashion (either IP multicast or an application-layer variant). Cloud computing could aid realtime mobile sharing in terms of storage of multimedia content and context data and hosting services for reasoning based on context from various sources for great numbers of users. Complete service platforms could even be run in the cloud.

Mobility and efficient realtime sharing however, require an interplay between devices, services and the network infrastructure. Both local and remote storage and processing are needed, since mobile devices can be out of network reach or have limited bandwidth or battery. The device still needs to operate when unconnected, for gathering sensor network info, buffering audio/video, and for discovering and selecting the next network connection to use, etc. With limited bandwidth or battery level a smart choice should be made for what to transmit, what to process remotely and what to do on the device.

Acceptable performance can be achieved for realtime mobile sharing by a combination of centralisation and decentralisation, i.e. centralisation of services for groups of users in service platforms and federations between these platforms to allow service usage in the coverage area of other service platforms. Furthermore, the distribution of content and context needs to be decentralized to allow for efficient realtime content and context sharing.

6.2 Future research directions

A number of challenges remain for mobility and sharing of networks: Efficient and cost-effective use of multiple available networks, QoS guarantees with high number of users in a shared medium like WLAN, context-awareness in network elements to dynamically adapt to provider/consumer needs.

For mobility and sharing of WSANs, a number of challenges remain: solutions for controlled configuration and actuation of WSANs by multiple users, coexistence and adaptability of WSANs that overlap.

A number of challenges remain for mobility and sharing of multimedia: compare the efficiency and adaptability of application-level multicast versus IP multicast, validate combined use of user- and network-initiated partial session mobility.

For (pervasive) service platforms some of the open challenges are: Make-before-break QoS reservations and security while roaming, performance when the number of users, amount of mobility and sharing increases, create a context distribution framework that scales. standardize the conceptual reasoning framework as an Unified Modelling Language [110] (UML) view. improve the conceptual reasoning framework to enable generation of executable models.

Glossary

3GPP	3rd Generation Partnership Project [8]. 20, 56, 117
6LoWPAN	IPv6 over Low power Wireless Personal Access Networks [106]. 79, 80, 119–121, 127, 169
A4C	AAA, Auditing and Charging. 144, 145, 147, 150, 152
AmbientMW	Ambient middleware. 129, 130, 167–169, 176
AmbientREP	Ambient Resource EndPoint. 169
AP	Wireless LAN Access Point. 96, 101, 102, 106
AS	Application Server. 56, 114, 115
ASM	Any Source Multicast [46]. 15
ATSC	Advanced Television Systems Committee. 14, 15
B2BUA	Back-to-back User Agent. 61, 63, 113, 115
BER	Bit Error Rate. 96, 106
BM-SC	Broadcast Multicast Service Centre. 111, 115
Bps	Bytes per second. 106
BSN	Body Sensor Network. 16, 71–73, 75, 80, 83–86
BWSP	Binary Web Service Protocol. 169, 172
CARD	Candidate Access Router Discovery [92]. 146
CDMA	Code Division Multiple Access. 15
CM	Context Manager. 167
CN	Corresponding Node. 58–64, 67–69
CoA	Care-off-Address. 13, 41

CoS	class of service. 92
CRC	Cyclic Redundancy Check. 80
CSMA	Carrier Sense Multiple Access. 74, 79
CTP	Collection Tree Protocol [65]. 74
DCCP	Datagram Congestion Control Protocol [87]. 131–133
DDI	Device Driver Interface. 129, 130, 133, 136, 169, 170
DHCP	Dynamic Host Configuration Protocol [51]. 127
DiffServ	Differentiated Services [38]. 92, 93
DVB	Digital Video Broadcast. 14, 15, 28, 144
DVB-C	DVB-Cable. 15
DVB-H	Digital Video Broadcast - Handheld. 15, 109, 110, 118, 150
DVB-S	DVB-Satelite. 15, 150
DVB-T	DVB-Terrestrial. 15, 117, 140, 150
ESN	Environmental Sensor Network. 16
EXI	Efficient XML Interchange [155]. 169
FA	Foreign Agent. 13
FTP	File Transfer Protocol [122]. 29
GPRS	General Packet Radio Service. 17, 45, 51–53, 72, 80, 83, 84, 128, 132, 134, 136–138, 140, 174, 185
GPS	Global Positioning System. 17, 72, 73, 81, 82, 84–86, 139
GSM	Global System for Mobile Communication. 17, 22
GUI	Graphical User Interface. 103, 160
HA	Home Agent. 12, 13, 51, 120, 126
HSDPA	High-Speed Downlink Packet Access independent, optimized personal services. 108–111, 113, 117, 140
HTML5	Revision 5 of HTML. 19, 20, 132, 172, 185
HTTP	HyperText Transport Protocol. 19, 138

HTTPS	HTTP Secure. 133
ICE	Interactive Connectivity Establishment [126]. 132
IETF	Internet Engineering Task Force. 19, 92
IM	Instant Messaging. 131
IMS	IP Multimedia Subsystem. 20, 56, 113–115
IP	Internet Protocol. 74, 106–108, 113–115, 120–123
IPG	IP gateway. 72, 74–76, 78, 80–86, 118
IPsec	Internet Protocol Security [83]. 13, 123, 131–133
IPTV	Internet Protocol television. 15, 18, 113
IPv4	Internet Protocol version 4. 127
IPv6	Internet Protocol version 6. 119–121, 127
IRC	Internet Relay Chat [82]. 121, 131, 135
ISDN	Integrated Services Digital Network. 20, 22
ITU-T	International Telecommunications Union Telecommunications Sector. 19, 20
JMF	Java Media Framework [146]. 50
JSON	JavaScript Object Notation. 132–136
KBps	Kilobytes per second. 102–104
LAN	Local Area Network. 51–53
LED	Light Emitting Diode. 16
LN	Local Node. 58–62
LTE	Long Term Evolution. 117
MAC	Media Access Control. 79, 97, 106
MANET	Mobile Ad-hoc NETwork. 146
MBC-AS	Application Server for Multimedia Broadcast Convergence. 114, 115, 117
MBMS	Multimedia Broadcast Multicast Service. 15, 108, 110, 111, 113, 115, 117, 140, 150
Mbps	megabits per second. 102
MDSM	Multi-Device System Manager. 58, 60

MIP	Mobile IP [116, 77]. 12, 20, 21, 51–53, 75, 76, 80, 87, 119, 122, 131, 133
MN	Mobile Node. 12, 13, 58–64, 67–69
MPEG	Moving Picture Experts Group. 15, 117
MPLS	multi-protocol label switching [125]. 92, 93
NAT	Network Address Translation. 127, 132, 161, 170
NEMO	Network Mobility [48, 115]. 13, 119, 120
NME	Network Monitoring Entity. 167
NSLP	Next Steps in Signaling (NSIS) Signalling Layer Protocol [97]. 92
OS	Operating System. 97
OSGi	Open Services Gateway Initiative. 156, 173, 174
P-CSCF	Proxy CSCF. 114
P2P	Peer to Peer. 20, 131
PGP	Pretty Good Privacy [43]. 13
PSN	Participatory Sensor Network. 17
PSTN	Public Switched Telephone Network. 20, 22
PSYC	Protocol for SYnchronous Conferencing [152]. 121, 122, 131, 135
QoS	Quality of Service. 4, 6, 8, 10, 13, 14, 28, 43, 90–98, 100–102, 106, 107, 144, 145, 150, 184
REST	Representational State Transfer. 131, 132, 135–138, 168, 169, 172, 175
RPC	Remote Procedure Call. 132–135
RSVP	resource reservation protocol [41]. 92, 101
RSVP-TE	RSVP traffic engineering [27]. 93
RTSP	Real Time Streaming Protocol [136]. 18–20, 113
SAML	Security Assertions Markup Language. 146, 147
SBM	Subnet Bandwidth Manager [161]. 92
SCTP	Stream Control Transmission Protocol [145]. 13, 131–133

SDP	Session Description Protocol [127]. 43, 45, 58, 113–115, 117
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions [71]. 121, 131, 133, 135, 174, 179
SIP	Session Initiation Protocol [31, 123]. 19–21, 23, 43–45, 47, 48, 50–56, 58, 62, 67, 70, 71, 87, 113–115, 117, 123, 125, 131, 146, 167
SMTP	Simple Message Transfer Protocol [86]. 131, 133, 135
SNR	Signal to Noise Ratio. 94, 96, 100, 103, 106
SOA	Service Oriented Architecture. 154, 166
SOAP	Simple Object Access Protocol [105]. 122, 131–135, 159, 170–173, 175, 177
SSC	subsession controller [13]. 59, 61, 63, 64, 67–70
SSH	Secure Shell [162]. 29, 39
SSL	Secure Socket Layer [63]. 13, 129–133, 168
SSM	Source Specific Multicast [32]. 15
SSN	Structure Sensor Network. 16, 17, 71–73, 81–86
STUN	Session Traversal Utilities for NAT [131]. 132
TCP	Transmission Control Protocol. 13, 21, 120, 121, 131–133
TDMA	Time Division Multiple Access. 74, 79, 80
TLS	Transport Layer Security [50]. 13, 131–133
TSN	Transport Sensor Network. 17
TV	Television. 14, 108, 109, 111, 113, 117
UA	User Agent. 58, 62, 70
UDLR	Unidirectional Link Routing. 28
UDP	User Datagram Protocol. 13, 104, 106, 121, 131–133, 169
UE	User Equipment. 111, 115
UML	Unified Modelling Language [110]. 155, 156, 158, 159, 180, 186
UMTS	Universal Mobile Telecommunications System. 13, 15, 28, 83, 108, 113, 115, 117
VASP	Value-added Service Provider. 152

VID	Virtual IDentity. 144, 146, 147, 149
VOD	Video on Demand. 18
VoIP	Voice over IP. 18, 22
VPN	Virtual Private Network. 12, 123
VSN	Vehicle Sensor Network. 17, 71, 72, 75, 81–83, 86
WiMAX	Worldwide Interoperability for Microwave Access. 110, 118
WLAN	Wireless LAN. 3, 6, 8, 12, 27, 45, 51–53, 72, 73, 80, 83–85, 90, 95–98, 101, 106–108, 113, 137, 140, 144, 145, 150, 184, 186
WSAN	Wireless Sensor and Actuator Network. 2–7, 10, 11, 15–18, 35, 37, 39, 51, 71–83, 85–87, 89, 90, 118–128, 130, 132, 139–141, 151, 156, 160–164, 167, 168, 170–175, 177–181, 183, 186
xDSL	any of various Digital Subscriber Line technologies. 14, 15, 90
XML	eXtensible Markup Language. 129, 130, 132–136, 165, 167, 169, 171
XMPP	Extensible Messaging and Presence Protocol [132]. 19, 121, 131, 133, 135–138, 165, 170, 174, 179

Bibliography

[1] Viewer ratings of television channels in the netherlands. Technical report, Stichting KijkOnderzoek (SKO).

[2] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, ISO/IEC 8802-11:1999(E), ANSI/IEEE std 802.11. IEEE standards publication, Institute of Electrical and Electronics Engineers, 1999.

[3] 802.1d - MAC bridges. IEEE standards publication, Institute of Electrical and Electronics Engineers, 2004.

[4] UMTS factsheet. Technical Report version 2.2, Federal Office for Communication, November 2004.

[5] Capacity coverage & deployment considerations for IEEE 802.11g. White paper, Cisco Systems, 2005.

[6] HSPD indoor deployment aspects. Technical Report 80-W0976-! Rev. A, Qualcomm, Engineering Services Group (ESG), September 2006.

[7] 3GPP. Combining Circuit Switched (CS) and IP Multimedia Subsystem (IMS) Services, Stage 2 (Release7). TS 23.279, 3rd Generation Partnership Project (3GPP).

[8] 3GPP. IP Multimedia (IM) session handling; IM call model; Stage 2. TS 23.218, 3rd Generation Partnership Project (3GPP).

[9] 3GPP. IP Multimedia Subsystem (IMS); Stage 2. TS 23.228, 3rd Generation Partnership Project (3GPP).

[10] 3GPP. Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description (Stage 2). TS 23.246, 3rd Generation Partnership Project (3GPP).

[11] Samuli Aalto, Janne Aaltonen, and Jouni Karvo. Quantitative performance comparison of different content distribution modes. *Performance Evaluation*, 63(4-5):395 – 422, 2006.

[12] Janne Aaltonen, Jouni Karvo, and Samuli Aalto. Multicasting vs. unicasting in mobile communication systems. In *WOWMOM '02: Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, pages 104–108, New York, NY, USA, 2002. ACM.

[13] Jasper Aartse Tuijn and Dennis Bijwaard. Spanning a multimedia session across multiple devices. *Bell Labs Technical Journal*, Volume 12, Issue 4:179–193, Winter 2006.

[14] R. Aguiar, D. Bijwaard, B. Farschian, and A. Sarma. Pervasive services for next generation heterogeneous networks. In *Proc. World Telecommun. Congress (WTC'06)*, Budapest, Hungary, 2006.

[15] R. Aguiar, D. Bijwaard, J. Jaehnert, P. Christ, and H. Einsiedler. Designing networks for the delivery of advanced flexible personal services: the daidalos approach. In *IST Mobile and Wireless Communication Summit*, 2004.

[16] R. L. Aguiar, A. Sarma, D. Bijwaard, L. Marchetti, P. Pacyna, and R. Pascotto. Pervasiveness in a competitive multi-operator environment: the Daidalos project. *IEEE Communications Magazine*, pages 22–27, October 2007.

[17] R.L. Aguiar, A. Sarma, D. Bijwaard, L. Marchetti, and P. Pacyna. Pervasiveness in a competitive multi-operator environment:the daidalos project. *Communications Magazine, IEEE*, 45(10):22–26, October 2007.

[18] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunaliolu, and W. Wang. Mobility management in current and future communication networks. *IEEE Network Magazine*, August 1998.

[19] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunaliolu, and W. Wang. Mobility management in next generation wireless systems. Technical report, Georgia Institute of Technology, 1999.

[20] M. Ali, T. Suleman, and Z.A. Uzmi. MMAC: a mobility-adaptive, collision-free MAC protocol for wireless sensor networks. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pages 401 – 407, april 2005.

[21] Open Mobile Alliance. Common definitions for OMA RESTful Network APIs, OMA-TS-REST_NetAPI_Common-V1_0. `http://www.openmobilealliance.org/`, Last visited July 2012.

[22] Ambient systems. ambient-systems.net, Last visited August 2011.

[23] Chris Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[24] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[25] Antaris solutions. `http://www.antaris-solutions.net`, Last visited August 2011.

[26] A. Avci, S. Bosch, M. Marin-Perianu, R. S. Marin-Perianu, and P. J. M. Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *23th International Conference on Architecture of Computing Systems, ARCS 2010, Hannover, Germany*, pages 167–176, Berlin, February 2010. VDE Verlag.

[27] Daniel Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: extensions to RSVP for LSP tunnels. RFC 3209, IETF, December 2001.

[28] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R.E. Strom, and D.C. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*, pages 262 –272, 1999.

[29] M. S. Bargh, D. Bijwaard, H. Zandbelt, E. Meeuwissen, and A. Peddemors. Mobility management in beyond 3g environments. In *Proceedings of Wireless World Research Forum 9, WWRF9*, July 2003.

[30] M. S. Bargh, H. Zandbelt, and A. Peddemors. Managing mobility in 4g environments with federating service platforms (an overview). In *Proceedings of EVOLUTE Workshop, Evolute'03*, 2003.

[31] A. Berger and D. Romascanu. Power ethernet MIB. RFC 3621, IETF, December 2003.

[32] Santanu Bhattacharyya and I. Ed. An overview of Source-Specific multicast (SSM). RFC 3569, IETF, July 2003.

[33] D. Bijwaard. Requirements for group sessions using multicast. Internet Draft 00, IETF, August 2007.

[34] D. J. A. Bijwaard, P. J. M. Havinga, and E. H. Eertink. Analysis of mobility and sharing of wsns by ip applications. *International Journal of Distributed Sensor Networks*, 2012:923594, October 2011.

[35] Dennis J.A. Bijwaard, Henk Eertink, and Paul J.M. Havinga. Challenges in efficient realtime mobile sharing. 2012. To appear.

[36] Dennis J.A. Bijwaard, Wouter A.P. Kleunen, Paul J.M. Havinga, Leon Kleiboer, and Mark J.J. Bijl. Industry: Using dynamic WSNs in smart logistics for fruits and pharmacy. In *Proceedings of SenSys11, November 14, 2011, Seattle, WA, USA.*, SenSys '11, pages 218–231, New York, NY, USA, 2011. ACM.

[37] Dennis J.A. Bijwaard, Berend Jan van der Zwaag, Nirvana Meratia, Hylke W. van Dijk, Henk Eertink, and Paul Havinga. Reuse of pervasive architectures. *Ubiquitous Envionments*, 2012.

[38] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service. RFC 2475, IETF, December 1998.

[39] S. Bosch, R. S. Marin-Perianu, P. J. M. Havinga, M. Marin-Perianu, A. Horst, and A. Vasilescu. Automatic recognition of object use based on wireless motion sensors. In *International Symposium on Wearable Computers 2010, Seoul, South Korea*, pages 143–150, USA, October 2010. IEEE Computer Society.

[40] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. RFC 1633, IETF, June 1994.

[41] R. Braden, L. Zhang, and S. Berson. Resource ReSerVation protocol (RSVP) – version 1 functional specification. RFC 2205, IETF, September 1997.

[42] BSD. Tinyos, operating system designed for low-power wireless devices.

[43] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer. Openpgp message format. RFC 4880, IETF, November 2007.

[44] G. Camarillo, A. Niemi, M. Isomaki, M. Garcia-Martin, and H. Khartabil. Referring to multiple resources in the session initiation protocol (SIP). RFC 5368, IETF, October 2008.

[45] SENSEI consortium. Integrating the physical with the digital world of the network of the future. `www.sensei-project.eu`, Last visited Jan 2011.

[46] S. Deering. Host extensions for IP multicasting. RFC 1112, IETF, August 1989.

[47] P. Deutsch. GZIP file format specification version 4.3. RFC 1952, IETF, May 1996.

[48] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network mobility (NEMO) basic support protocol. RFC 3963, IETF, January 2005.

[49] Restlet developers. Restfull web services. `http://www.restlet.org`, Last visited October 2012.

[50] T. Dierks. The transport layer security (tls) protocol version 1.2. RFC 5246, IETF, August 2008.

[51] R. Droms. Dynamic host configuration protocol. RFC 2131, IETF, March 1997.

[52] Ashutosh Dutta, Ravi Jain, Daniel Wong, James Burns, Ken Young, and Henning Schulzrinne. Multilayered mobility management for survivable network. In *Milcom*, Vienna, Virginia, November 2001.

[53] Ken Dutton, Steve Thompson, and B. Barraclough. *The Art of Control Engineering*. Addison-Wesley,, 1997.

[54] M. Eisenhauer, P. Rosengren, and P. Antolin. A development platform for integrating wireless devices and sensors into ambient intelligence systems. In *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops '09. 6th Annual IEEE Communications Society Conference on*, pages 1 –3, 2009.

[55] Elvin. `http://www.elvin.org`, Last visited November 2011.

[56] Serral Estefanía, Valderas Pedro, and Pelechano Vicente. A model driven development method for developing context-aware pervasive systems. In *Proceedings of the 5th international conference on Ubiquitous Intelligence and Computing*, UIC '08, pages 662–676, Berlin, Heidelberg, 2008. Springer-Verlag.

[57] Serral Estefanía, Pedro Valderas, Javier Munoz, and Vicente Pelechano. *Developing Ambient Intelligence*, chapter Towards a Model Driven Development of Context-aware Systems for AmI Environments, pages 114–124. Springer-Verlag, Berlin, Germany, 2008.

[58] L. Evers, M. J. J. Bijl, M. Marin-Perianu, R. S. Marin-Perianu, and P. J. M. Havinga. Wireless sensor networks and beyond: A case study on transport and logistics. Technical Report TR-CTIT-05-26, Centre for Telematics and Information Technology University of Twente, Enschede, June 2005.

[59] I. Fette and A. Melnikov. The websocket protocol. RFC 6455, IETF, December 2011.

[60] Ludger Fiege, Felix Gartner, Oliver Kasten, and Andreas Zeidler. Supporting mobility in content-based publish/subscribe middleware. In Markus Endler and Douglas Schmidt, editors, *Middleware 2003*, volume 2672 of *Lecture Notes in Computer Science*, pages 998–998. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-44892-6_6.

[61] WiMAX forum. WiMAX forum.

[62] XMPP Standards Foundation and Google. Xmpp technologies: Jingle.

[63] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL protocol version 3.0. `http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt`.

[64] S.H. Gajjar, S.N. Pradhan, and K.S. Dasgupta. Wireless sensor network: Application led research perspective. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, pages 025 –030, sept. 2011.

[65] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, New York, NY, USA, 2009. ACM.

[66] Gong Haigang, Liu Ming, Wang Xiaomin, Chen Lijun, and Xie Li. An interference free cluster-based TDMA protocol for wireless sensor networks. In Xiuzhen Cheng, Wei Li, and Taieb Znati, editors, *Wireless Algorithms, Systems, and Applications*, volume 4138 of *Lecture Notes in Computer Science*, pages 217–227. Springer Berlin / Heidelberg, 2006. 10.1007/11814856_22.

[67] Ian Hickson. The websocket API. `http://www.w3.org/html5/websockets/`.

[68] T.J. Hofmeijer, S.O. Dulman, P.G. Jansen, and P.J.M. Havinga. AmbientRT - real time system software support for data centric sensor networks. In *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 61–66. IEEE Computer Society Press, 2004.

[69] D. A. Huffman. A method for the construction of minimum redundancy codes. In *Proc. IRE 40*, pages 1098–1101, 1952.

[70] Hydra. Middleware for networked embedded systems. `www.hydramiddleware.eu`, Last visited May 2010.

[71] IETF. The SIMPLE working group charter. `http://datatracker.ietf.org/wg/simple/charter/`.

[72] Inertia technology. `http://inertia-technology.com`, Last visited September 2011.

[73] M. Isomura, T. Riedel, C. Decker, M. Beigl, and H. Horiuchi. Sharing sensor networks. In *Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on*, page 61, july 2006.

[74] ITU-T. Open standard. `http://www.itu.int/en/ITU-T/ipr/Pages/open.aspx`.

[75] ITU-T. Packet-based multimedia communication systems.

[76] Carlos Cetina Javier Munoz, Vicente Pelechano. Implementing a pervasive meetings room: A model driven approach. In *Proccedings of the International Workshop on Ubiquitous Computing (IWUC 2006)*, pages 13–20, 2006.

[77] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. RFC 3775, IETF, June 2004.

[78] Matjaz B. Juric, Ivan Rozman, Bostjan Brumen, Matjaz Colnaric, and Marjan Hericko. Comparison of performance of web services, WS-security, RMI, and RMI-SSL. *Journal of Systems and Software*, 79(5):689 – 700, 2006. Quality Software.

[79] jWebSocket team. Java/javascript websocket library. `http://websocket.org`, Last visited October 2012.

[80] Ahmad Kamal and Paris Avgeriou. Modeling architectural patterns' behavior using architectural primitives. In Ron Morrison, Dharini Balasubramaniam, and Katrina Falkner, editors, *Software Architecture*, volume 5292 of *Lecture Notes in Computer Science*, pages 164–179. Springer Berlin / Heidelberg, 2008.

[81] A. Kansal, S. Nath, J. Liu, and F. Zhao. Senseweb: An infrastructure for shared sensing. *IEEE MultiMedia*, 14:8–13, 2007.

[82] W. Kantrowitz. Network questionnaires. RFC 459, IETF, February 1973.

[83] S. Kent and K. Seo. Security architecture for the internet protocol. RFC 4301, IETF, December 2005.

[84] K.K. Khedo and R.K. Subramanian. A service-oriented component-based middleware architecture for wireless sensor networks. *IJCSNS International Journal of Computer Science and Network Security*, 9(3):174–182, 2009.

[85] Eino Kivisaari. Mobile TV technology: T-109.4300 network services and business models. Technical report, Telecommunications Software and Multimedia Laboratory (TML), Feb 2007.

[86] J. Klensin. Simple mail transfer protocol. RFC 5321, IETF, October 2008.

[87] E. Kohler, M. Handley, and S. Floyd. Datagram congestion control protocol (dccp). RFC 4340, IETF, March 2006.

[88] L. Kolos-Mazuryk, G. J. Poulisse, and P. A. T. van Eck. Requirements engineering for pervasive services. In *Second Workshop on Building Software for Pervasive Computing. Position Papers., San Diego, California, USA*, pages 18–22, October 2005.

[89] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. RFC 4919, IETF, August 2007.

[90] P. Laine, C. Boscher, D. Boettle, and L. Feijt. WiMAX, making ubiquitous high-speed data services a reality. Strategy white paper, Alcatel-Lucent, June 2004.

[91] Yee Wei Law and P.J.M. Havinga. How to secure a wireless sensor network. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pages 89 – 95, dec. 2005.

[92] M. Liebsch, A. Singh, H. Chaskar, D. Funato, and E. Shim. Candidate access router discovery (card). RFC 4066, IETF, July 2005.

[93] Javier Lopez. Unleashing public-key cryptography in wireless sensor networks. *J. Comput. Secur.*, 14:469–482, September 2006.

[94] Mrio Macedo, Antnio Grilo, and Mrio Nunes. Distributed latency-energy minimization and interference avoidance in TDMA wireless sensor networks. *Computer Networks*, 53(5):569 – 582, 2009.

[95] A. Malatras, A. Asgari, and T. Bauge. Web enabled wireless sensor networks for facilities management. *Systems Journal, IEEE*, 2(4):500 –512, dec. 2008.

[96] Mehdi Mani and Noel Crespi. Session mobility between heterogeneous accesses with the existence of IMS as the service control overlay. In *10th IEEE International Conference on Communication Systems 2006*, Singapore, Singapore, October 2006.

[97] Karagiannis G. Manner, J. and A. McDonald. NSIS signaling layer protocol (NSLP) for quality-of-service signaling. RFC 5974, IETF, October 2010.

[98] E. Marilly, G. Delegue, O. Martinot, and S. Betge-Brezetz. Adaptation and personalization of interactive mobile TV services. In *ICIN 2006 conference*, 2006.

[99] N. Meratnia, B. J. van der Zwaag, H. W. van Dijk, D. J. A. Bijwaard, and P. J. M. Havinga. Sensor networks in the low lands. *Sensors*, 10(9):8504–8525, September 2010.

[100] Nirvana Meratnia, Berend Jan van der Zwaag, Hylke W. van Dijk, Dennis Bijwaard, and Paul J.M. Havinga. Sensor networks in the low lands. *Sensors*, 10(9):8504–8525, 2010.

[101] A.C Milonas. Enterprise networking for the new millenium. *Bell Labs Technical Journal*, Volume 5, Issue 1:73–94, 2000.

[102] Ren-Hung Hwang Min-Xiou Chen, Chen-Jui Peng. SSIP: Split a SIP session over multiple devices. *Computer Standards & Interfaces*, 29(5):531–545, July 2007.

[103] M. Miraoui, C. Tadj, and C. Ben Amar. Architectural survey of context-aware systems in pervasive computing environment. *Ubiquitous Computing and Communication Journal*, 3(3), 2008.

[104] A. Misra, S. Das, and P. Agrawal. Application-centric analysis of IP-based mobility management techniques. *Journal of Wireless Communications and Mobile Computing*, Volume 1, Issue 3, August 2001.

[105] Nilo Mitra and Yves Lafon. SOAP specificiations. `http://www.w3.org/TR/soap/`.

[106] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 packets over IEEE 802.15.4 networks. RFC 4944, IETF, September 2007.

[107] Rene Muller and Gustavo Alonso. Efficient sharing of sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 109 –118, oct. 2006.

[108] NIST. NIST SIP stack incorporating JAIN SIP 1.1.

[109] University of Strasbourg and community. Jitsi (SIP communicator).

[110] OMG. Unified modeling language (uml). `www.uml.org`, Last visited January 2011.

[111] S. Park, K.Kim, W. Haddad, S. Chakrabarti, and J. Laganier. IPv6 over low power WPAN security analysis. Internet Draft 05, IETF, March 2011.

[112] C. Pastrone, M.A. Spirito, R. Tomasi, and F. Rizzo. A jabber-based management framework for heterogeneous sensor network applications. *International Journal of Software Engineering and Its Applications*, 2(3):9–24, 2008.

[113] S. Paul, R. Yates, D. Raychaudhuri, and J. Kurose. The cache-and-forward network architecture for efficient mobile content delivery services in the future internet. In *Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference*, pages 367 –374, may 2008.

[114] Bastien Peelen, Miroslav Zivkovic, Dennis Bijwaard, and Harold Teunissen. Supporting qos in broadband wireless and wired access. *Bell Labs Technical Journal*, Volume 8, Issue 2:65–81, Summer 2003.

[115] Eranga Perera, Vijay Sivaraman, and Aruna Seneviratne. Survey on network mobility support. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8:7–19, April 2004.

[116] C. Perkins. IP mobility support for IPv4, revised. RFC 5944, IETF, November 2010.

[117] Charles Perkins. IP mobility support for IPv4. RFC 3344, IETF, August 2002.

[118] Charles Perkins and David B. Johnson. Mobility support in IPv6. In *second annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom '96*, pages 27–37, 1996.

[119] H. Pham and S. Jha. An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC). In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 558 – 560, oct. 2004.

[120] D. J. Plas and W. Romijn. Online charging in a converged network environment: An overview and results from an initial implementation. In *Proc. 10th Internat. Conf. on Convergence in Services, Media and Networks (ICIN'06)*, Bordeaux, France, 2006.

[121] Christos Politis, Kar Ann Chew, and Rahim Tafaz. Multilayer mobility management for all-IP networks: Pure SIP vs. hybrid SIP/mobile IP. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, 2003.

[122] J. Postel and J. Reynolds. File transfer protocol. RFC 959, IETF, October 1985.

[123] Adam Roach. Session initiation protocol (SIP)-Specific event notification. RFC 3265, IETF, June 2002.

[124] Willem A. Romijn, Dirk-Jaap Plas, Dennis Bijwaard, Erik Meeuwissen, and Gijs van Ooijen. Mobility management for SIP sessions in a heterogeneous network environment. *Bell Labs Technical Journal*, Volume 9, Issue 3:237–253, Autumn 2004.

[125] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. RFC 3031, IETF, January 2001.

[126] J. Rosenberg. Interactive connectivity establishment (ice): A protocol for network address translator (nat) traversal for offer/answer protocols. RFC 5245, IETF, April 2010.

[127] J. Rosenberg, J. Peterson, Henning Schulzrinne, and G. Camarillo. Best current practices for third party call control (3pcc) in the session initiation protocol (SIP). RFC 3725, IETF, April 2004.

[128] J. Rosenberg and Henning Schulzrinne. An Offer/Answer model with session description protocol (SDP). RFC 3264, IETF, June 2002.

[129] J. Rosenberg and Henning Schulzrinne. Session initiation protocol (SIP): locating SIP servers. RFC 3263, IETF, June 2002.

[130] J. Rosenberg, Henning Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, IETF, June 2002.

[131] Jonathan Rosenberg, J. Weinberger, Christian Huitema, and R. Mahy. STUN - simple traversal of user datagram protocol (UDP) through network address translators (NATs). RFC 3489, IETF, March 2003.

[132] Peter Saint-Andre. Extensible messaging and presence protocol (XMPP): core. RFC 3920, IETF, October 2004.

[133] J.P. Sanderman. A traffic control algorithm for wireless LANs. Master thesis, University of Twente, 2001.

[134] Amardeo Sarma, Alfredo Matos, Joo Giro, and Rui Aguiar. Virtual identity framework for telecom infrastructures. *Wireless Personal Communications*, 45:521–543, 2008. 10.1007/s11277-008-9475-4.

[135] M. Satyanarayanan. Pervasive computing: vision and challenges. *Personal Communications, IEEE*, 8(4):10 –17, aug 2001.

[136] Henning Schulzrinne, Asha Rao, and R. Lanphier. Real time streaming protocol (RTSP). RFC 2326, IETF, April 1998.

[137] Henning Schulzrinne and Elin Wedlund. Application-Layer mobility using SIP. *Mobile Computing and Communications Review (MC2R)*, 4(3):47–57, July 2000.

[138] Ron Shacham, Henning Schulzrinne, Srisakul Thakolsri, and Wolfgang Kellerer. Session initiation protocol (SIP) session mobility. RFC 5631, IETF, October 2009.

[139] Lei Shu, M. Hauswirth, Long Cheng, Jian Ma, V. Reynolds, and Lin Zhang. Sharing worldwide sensor network. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 189 –192, 28 2008-aug. 1 2008.

[140] N. Sinha and R. Oz. The statistics of switched broadcast. In *Proceedings of SCTE 2005 Conference on Emerging Technologies*, 2005.

[141] Smart Surroundings. Future ambient systems. `www.Smart-Surroundings.org`, Last visited April 2010.

[142] Yee Jiun Song, Venugopalan Ramasubramanian, and Emin Gun Sirer. Optimal resource utilization in content distribution networks. Technical Report TR 2005-2004, Cornell University, Comput. and Inform. Sci, November 2005.

[143] R. Sparks. The session initiation protocol (SIP) refer method. RFC 3515, IETF, April 2003.

[144] R. Sparks. The session initiation protocol (SIP) Referred-By mechanism. RFC 3892, IETF, September 2004.

[145] R. Stewart. Stream control transmission protocol. RFC 4960, IETF, September 2007.

[146] SUN. Java Media Framework.

[147] Peter Sutton, Rhys Arkins, and Bill Segall. Supporting disconnectedness-transparent information delivery for mobile and invisible computing. In *Proceedings of the 1st International Symposium on Cluster Computing and the Grid*, CCGRID '01, pages 277–, Washington, DC, USA, 2001. IEEE Computer Society.

[148] P802.11 task group E. MAC enhancements for quality of service. IEEE standards publication, Institute of Electrical and Electronics Engineers, 2005.

[149] A Teslyuk, S. Krashakov, and L. Schchur. On the universality of rank distributions of website popularity. Technical report, Landau Institute for Theoretical Physics, 2003.

[150] Ren Treffer and Till Klocke. Android xmpp (jabber) client library in java. `http://code.google.com/p/asmack/`, Last visited October 2012.

[151] UbiSec&Sens. Ubiquitous sensing and security in the European homeland. `www.ist-ubisecsens.org`, 2006.

[152] Carlo v. Loesch. Whitepaper on PSYC. `http://www.psyc.eu/whitepaper/`.

[153] Sietse van der Gaast and Dennis Bijwaard. Efficiency of personalized content distribution. *Bell Labs Technical Journal*, Volume 13, Issue 2:135–145, 2008.

[154] R. van Eijk, J. Brok, J. van Bemmel, and B. Busropan. Access network selection in a 4G environment and the roles of terminal and service platform. In *Proc. of Wireless World Research Forum 10, WWRF 10*, 2003.

[155] W3C. Efficient xml interchange(exi) primer. `www.w3.org/TR/exi-primer`, Last visited Jan 2011.

[156] Miao-Miao Wang, Jian-Nong Cao, Jing Li, and Sajal K. Das. Middleware for wireless sensor networks: A survey. *Journal of Computer Science and Technology*, 23(3):305–326, May 2008.

[157] Yuanli Wang, Xianghui Liu, and Jianping Yin. Requirements of quality of service in wireless sensor network. In *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*, page 116, april 2006.

[158] Elin Wedlund and Henning Schulzrinne. Mobility support using SIP. In *2nd ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM)*, Seattle, Washington, August 1999.

[159] Tao Wu and Subir Biswas. Reducing inter-cluster TDMA interference by adaptive MAC allocation in sensor networks. In *Proceedings of the First International IEEE WoWMoM Workshop on Autonomic Communications and Computing (ACC'05) - Volume 02*, pages 507–511, Washington, DC, USA, 2005. IEEE Computer Society.

[160] Antaris solutions. `http://xmpp.org/extensions/xep-0251.html`, Last visited July 2012.

[161] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, and M. Speer. SBM (subnet bandwidth manager): A protocol for RSVP-based admission control over IEEE 802-style networks. RFC 2814, IETF, May 2000.

[162] Tatu Ylonen and Chris Lonvick. The secure shell (SSH) protocol architecture. RFC 4251, IETF, January 2006.

[163] Dalong Zhang, Qing Li, Xiaoyi Zhang, and Xiaomei Wang. DE-ASS: An adaptive MAC algorithm based on mobility evaluation for wireless sensor networks. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1 –5, sept. 2010.

[164] Y. Zhang, N. Meratnia, and P. J. M. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 12(2):159–170, 2010.

[165] Y. Zhang, N. Meratnia, and P.J.M. Havinga. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In *Proceedings of the IEEE 23rd International Conference on Advanced Information Networking and Applications Workshops*, pages 990–995, Bradford, United Kingdom, May 2009. IEEE Computer Society Press.