Autonomic Resource Management in IEEE 802.11 Open Access Networks

Muslim Elkotob

Computer Communication Group
Division of Systems and Interaction
Department of Computer Science and Electrical Engineering
Luleå University of Technology
To my family
Acknowledgements

I would like to thank my supervisor Evgeny Osipov for his continuous and precious support in the scientific and academic sense. Our numerous discussions have revealed new research challenges and also contributed to rounding up the scope of this licentiate. Thank you, Evgeny, for being such a good scientific advisor. I would like to thank my co-supervisor Christer Åhlund with whom I had a lot of bilateral as well as project consortium-based scientific discussions about various hot topics in networking even before joining LTU. My endless gratitude and best thanks go to my parents who have done all they can and more than they can to support me throughout my academic life. This licentiate is a corner milestone in my career. It allows me to round up the research done so far in my main direction: Quality of Service in Autonomic Networking. I am thankful to the whole team at LTU Systemteknik for the pleasant and productive atmosphere which surrounded me during my work.

Muslim Elkotob
Luleå, November 3, 2008
Abstract

This work is inspired by the autonomic networking paradigm. It investigates the issue of making network resource management mechanisms in the access part of wireless networks self-aware and self-configuring. Open access networks (OAN) have private WLAN cells made available for public use and enable bypassing mobile users to profit from continuous coverage, allowing private and public subscribers to share a common infrastructure. Resources are network parameters, e.g. bandwidth, capacity, that have to be fairly and accurately allocated, distributed, and managed in order to optimize performance and ensure proper and fair operation. Analyzing and exploring existing multimedia session and resource management mechanisms on the wireless part is a crucial step for determining their limitations and shortcomings. A new way of making OANs self-aware is proposed and a self-configuring mechanism which has a competitive advantage in the qualitative sense over already existing mechanisms is designed and developed. The amount of network resources as well as the timing and form of their availability are decisive for enabling service functionality and providing a sufficient quality of service level. Our area of discourse is the “last-mile” part of an 802.11 network; i.e. the part between the access point and the mobile node. In OANs, maintaining and updating rates for different flows used by a single terminal traversing many adjacent APs is a challenge in itself. It is solved by using several key design factors. Monitoring the environment locally and updating the designed information structures leads to network self-awareness. Next, a high-level mechanism that aggregates information from several protocols including CARD (Candidate Access Router Discovery), MIP (Mobile IP), and SIP (Session Initiation Protocol) is defined. It specifies the interactions between the mobile terminal and the access point controller. Then capacity maintenance and rate allocation issues at the two end-points of the last-mile are analyzed. The overhead in OANs required for the launching or reconfiguration of a real-time session can be significantly reduced by a proposed algorithm which performs parameter injection into a multimedia data stream instead of following the classical negotiation path. A solution to change the session characteristics of an application that adapts to varying network conditions was developed, including a QoS solution with SIP and MIP to optimize the session characteristics by taking into account the changed conditions in the network. With this approach, reduced session signaling delay and low jitter values are achieved as implementation results show. The thesis is concluded by considering the general issues: self-aware node architectures, self-configuring resource management.

Keywords
Open Access Networks, Quality of Service, IEEE 802.11, Autonomic Communication, Parameter Injection, Multimedia Communication, Real-time Traffic.
# Contents

## 1 INTRODUCTION
- 1.1 Quality of Service in Open Access Networks ................................................................. 1
- 1.2 Autonomic Networking ..................................................................................................... 3
- 1.3 Motivation and Problem Statement .................................................................................. 4
- 1.4 A New Way of Doing Quality of Service in Autonomic Networking ................................. 6
- 1.5 My Contribution .............................................................................................................. 10
- 1.6 Structure of this Thesis ................................................................................................... 12

## 2 RELATED WORK
- 2.1 QoS Resource Management for Media Streaming with Focus on Self-Awareness and Self-Configuration .............................................................................................................. 13
- 2.2 Context Aware QoS Network Systems ............................................................................ 13
- 2.3 Multimedia Signaling and Session Management in Wireless Networks ......................... 20
- 2.4 Relevant Standardization Efforts .................................................................................... 21

## 3 BACKGROUND INFORMATION
- 3.1 Candidate Access Router Discovery Protocol (CARD) ..................................................... 23
- 3.2 Mobile IP (MIPv4) and Session Initiation Protocol (SIP) .................................................. 27
- 3.3 Session Description Protocol (SDP) ................................................................................. 32

## 4 PROTOCOL DESIGN
- 4.1 Terminal and Access Network Two-Way Signaling ......................................................... 33
- 4.2 Cross-Layer and Inter-Protocol Aspects ......................................................................... 38
- 4.3 Automated Resource Control over Last-Mile 802.11 Networks: Parameter Injection Algorithm and ICSP Protocol ........................................................................................................ 40

## 5 SELF-CONFIGURATION IN PROCESS DESIGN
- 5.1 Self-Configuring Resource Management on the Terminal ............................................... 51
- 5.2 Self-Managing Capacity Allocation in the Access Network ............................................. 58

## 6 DATA DESIGN AND MODEL SEMANTICS
- 6.1 Conclusion ....................................................................................................................... 65

## 7 CONCLUSION AND FUTURE WORK
- 7.1 Conclusion ....................................................................................................................... 71
- 7.2 Future Work: Map of the Road Ahead ............................................................................ 71

## REFERENCES
1 Introduction

1.1 Quality of Service in Open Access Networks

Multimedia streaming is the process of delivering audio-visual content over a telecommunication network in a continuous real-time manner and immediately displaying it at the receiver. It has evolved in parallel with the emergence and growth of IP-based networks, their related protocols, and the spreading network connection availability for end-users. Moreover, due to the increase in available bandwidth, as well as the reduction in network usage costs, media streaming has established its strong position in the industry and the service and consumer markets. Users increasingly access multimedia content through their mobile devices as the demand for mobile content and connectivity grows. Examples include Voice-over-IP, video conferencing, music, and movies.

However, multimedia streaming faces challenges concerning the perceived audio and video quality with changes in network parameters while users traverse several networks when on the move. That leads to flickers in video, noise in audio, and loss of audio-visual information. Therefore, this necessitates the presence of quality of service (QoS) mechanisms that guarantee service quality and give stability and reliability to the system. Currently, the trend of usage and the volume of streaming and multimedia traffic is an increasing one, but the most commonly used architecture remains based on the best effort model provided by the classical Internet. This architecture does not manage quality of resource allocation sufficiently well. In particular, parameters such as session signaling delay, session delay jitter, and packet loss rate should be analyzed and optimized. Providing better bounds for the aforementioned parameters ensures that the desired degree of quality is achieved and maintained.

Currently one strong trend in telecommunications is the expansion of broadband penetration of the “last-mile” part of networks. Most private residences will soon have access to a broadband network over ADSL, VDSL, fiber, cable, or radio. Furthermore, wireless technology will continue to be the main trend for the growing connectivity demands both for residential and business users. In dense metropolitan areas, numerous “micro base stations” will form continuous radio coverage allowing users to roam through a large distance while maintaining their communication sessions. Especially for multimedia and real-time applications, session continuity as well as a high bandwidth is required. This is the challenge posed and solved by deploying appropriate QoS
architectures and solutions on open access networks (OAN).

In OANs, the roaming as well as residential subscribers share the capacity of wireless LANs and access lines according to a general service agreement between all users and the network operator. This shapes the openness feature within the OAN. Due to the fact that such networks are low-cost when it comes to installation and deployment, they are attractive to use and are gaining subscribers every day. Furthermore, the “mobile-broadband” based on 3G still does not deliver the real promised data rate of several megabits per second. Therefore, it is likely that 802.11 will stay as the dominant technology for OANs, and new initiatives with micro-operators, mobile virtual network operators (MVNO)s, closed community open access [FON1], and expanded hotspots and clouds e.g. Boingo [BOI1], Linspot [LIN1], The Cloud [TC1]) will continue to emerge and expand. This positive development was motivating to benefit from the open access network paradigm.

Open access networks, as a concept was first introduced by Battiti et al [RB1]. It contains a business model, service model, and some architectural insights on how such a vision can be realized. Users being able to roam continuously traversing many access points and having the connections shared by residential owners and by-passers is quite an innovative concept that expanded later on. In [ME8] and [ME9], a specific case of an 802.11 OAN is highlighted including architectural details, comparative analyses with other approaches, and technical details on protocol operation.

The mechanisms that exist today for QoS resource management in media streaming lack the required degree of autonomy, proactive behavior, intelligence, self-awareness and self-configurability capabilities. Autonomy of behavior refers to the ability of a system to act and make decisions on resource management in an independent way, without requiring external intervention. This deficit is a challenge and the core of the problem since without sufficient self-awareness and self-configuration in resource management mechanisms, it is very hard to cope with the complexity and condition changes in wireless networks.

There are two major challenges at this point. Firstly, providing the appropriate level of QoS to multimedia applications faces many stringent constraints imposed by the real-time nature of the underlying traffic classes used by audio and video. Secondly, as the volume of multimedia content transferred and exchanged by mobile users grows drastically, new architectures and signaling protocols are constantly being developed to deliver appropriate QoS. However, very complex signaling and a lot of human interaction and control is required for the current solutions to run properly. This motivates the need for a new paradigm in computing and communication called “autonomic networking”, which is the topic of the next section.
1.2 Autonomic Networking

The term “autonomic” in computer science was first clearly highlighted in 2004 when IBM launched the autonomic computing initiative [IBM1]. The focus of the solution is on reducing the need for human interaction in complex computing and communication systems.

Several limited components of this technology are already up and running [ACP1], [ACP2]. Shortly afterwards, the autonomic communication or autonomic networking initiative emerged [MS1]. Autonomic communication and networking is about self-* features: self-awareness, self-healing, self-management, and self-configuration. The conceptual idea behind those features is related to other research work in domains such as: distributed systems, fault-management, and multi-hop ad-hoc, peer-to-peer, sensor and mesh networking. The idea is that communication devices combine local information and intelligence to form joint awareness and act in a self-organizing, coordinated manner.

Furthermore, autonomic communication views the system as composite functional entity. Functionality of this entity is flexible, can be composed dynamically, is evolving and its functions use semantics to model more complex behavior that in turn gives over more control to the system thereby reducing necessary human intervention. Major autonomic features are listed below.

- **Self-monitoring**: a system is self-monitoring when it is capable of measuring the different performance parameters which determine its state in terms of available resources, functioning modules, and general status. For self-monitoring, information collection as well as sensing and event detection are essential.
- **Self-healing**: this principle corresponds to the ability of a system to recover from faults, drops, crashes (mainly software), and deadlocks. This is crucial for nodes on which many other nodes depend. For instance a core router serving many access routers has to possess recovery and self-healing capabilities otherwise all attached access routers will suffer for a long time if no fast recovery after a system fault takes place.
- **Self-organizing**: this principle from autonomic communication can be applied in several contexts. In a self-organizing network, a set of nodes is able to form a network by organizing and arranging itself. A self-organizing system is one which is able to organize its components and modules in such a way that they form consistent services and applications that match the system and user requirements.
- **Self-awareness**: self-awareness is very closely related to self-monitoring. It also involves a lot of self-monitoring and then includes a step further. The further step is about determining the concrete state of the node either from a state space by direct mapping and matching or by applying fuzzy logic and approximations. The important fact about this feature is that it adds intelligence to the behavior of a node.
• **Self-configuration**: this principle is attributed to processes and mechanisms that are able to configure their parameters in an autonomic manner. As the values of different parameters levels vary, a system has to adapt accordingly in an either reactive or proactive manner. Thus processes acquire information using self-monitoring, and perform control operations such as signaling with reduced human intervention, to become self-configuring.

An important question to answer is whether full autonomy is possible at all in the networking world. Looking ahead, the answer for the near future is a definitive “no” because of the complexity of communication systems and their high degree of interactivity. Designing and implementing communication systems that behave autonomously is a growing and popular research direction on the other hand. More autonomy is introduced into the networking and communication area with algorithms and ideas from various domains including: computer science, mathematics, biology, chemistry and physics.

In wireless networks, resources are bottlenecks, therefore they have to be managed and consumed efficiently. Furthermore, the cost and effort for maintaining and administering networks would grow out of reasonable bounds if humans were to entirely handle this task. This is where autonomic networking comes into the picture, to reduce human intervention and solve tasks in a self-contained way as much as possible. Besides cost reduction, the autonomic communication paradigm attempts to: improve system performance by reducing overhead, increase efficiency, lower the degree of interaction between entities unless necessary, and use reasoning and active monitoring in order to take proper decisions in a timely manner. Part of the work done in this thesis is in line with the aforementioned goals of the autonomic communication paradigm.

This thesis applies autonomic principles to the open access network architecture. Now we move on to the next section to motivate the research done and come out with a solid problem statement.

### 1.3 Motivation and Problem Statement

This thesis has two main motivating paradigms as pillars: open access networks (OAN), and autonomic communication (AC). One core goal is to apply autonomic principles to the specific network architecture OAN. Upon applying those principles, several challenges are faced, and there is large potential for feature enhancement and performance optimization.

This section takes a look at the two domains to which this thesis is strongly related: Open Access Networks and Autonomic Communication. Having motivated the need of applying the principles of one of those domains to the other to resolve several upcoming challenges in networking, the focus moves to concisely formulating the problem.
Autonomic communication (AC) is the result of several evolution stages in networking. One of the predecessors is the so-called Intelligent Network, typically stated as its acronym IN, which was intended both for fixed as well as mobile telecom networks. It allows operators to differentiate themselves by providing value-added services in addition to the standard telecom services such as PSTN, ISDN and GSM services on mobile phones. In IN, the intelligence is provided by network nodes owned by telecom operators, as opposed to solutions based on intelligence in the telephone equipment, or in Internet servers. Intelligence and control logic started in a centralized way, where a central node had all the decision power and sent control messages to other nodes of the architecture. Then as computing capabilities of individual nodes improved, and the functionality required from the network architecture grew in complexity, it was time for distributing the intelligence and computational work among nodes. Distributed computing was one of the drivers for promoting autonomic functionality of nodes due to the absence of a central node that controls the whole network.

AC has been constantly evolving since its introduction in 2004. A global research community has been built to endorse this domain, as it tries to solve the upcoming challenges of complexity in networking. As mobility and Quality of Service (QoS) become harder to manage with the increasing number of devices, platforms, access network types, available architectures and signaling protocols, the Autonomic Communication research community tries to find systematic solutions to domain-specific problems.

Autonomic Communication has evolved in its own direction and has not yet succeeded
to provide any ideally fitting generic solution to existing problems except in very domain-specific areas. Moreover in the domain of OAN, solutions that provide easy-to-use broadband connectivity lack the sufficient degree of autonomy and QoS required for satisfying user needs. Therefore, the problem of how to apply the principles of one domain to the other emerged. This problem is handled in this thesis.

![Figure 2: user roaming in an open access network](image)

To more specifically describe the problem, it is essential to take a closer look at the settings for which our research results were developed. Roaming and residential users in dense metropolitan areas share the same IEEE 802.11 infrastructure. Residential users have a backhaul line at home to which an access point is connected. Roaming users bypass several residences and have continuous coverage as Figure 2 shows. Separation of the roaming and residential domains is done by using different Virtual LANs (VLAN) for each category. A roaming subscriber when using real-time and multimedia applications however faces challenges concerning QoS and session consistency. To make things even worse, protocol messaging and signaling for resource management is very complex involving a large negotiation overhead and posing an additional challenge on voice and video sessions in OAN. For this reason, developing better and partially autonomic solution within the OAN and AC domains is necessary. It has to make networks self-aware and resource management processes for session signaling and multimedia flow control self-configuring in OAN. The next section provides insight on how this is done.

1.4 A New Way of Doing Quality of Service in Autonomic Networking

In order to provide not only continuous coverage but also seamless sessions for multimedia communication over 802.11 OANs, fundamental design and algorithmic work has to be done. This requires research in the data design, process modeling, and
protocol optimization areas all within an integrated architecture.

Figure 3 depicts how autonomic networking is used in this thesis. Active sensing and monitoring for information extraction and reasoning using semantics to produce further more meaningful information about the network state paves the way for making networks “self-aware”. Once access networks possess information about their states and resources, they are better capable of endorsing autonomic protocols and algorithms.

In order to depict the variation in resource levels for real-time traffic with different adaptation mechanisms, some measurements were conducted. Figures 4 and 5 reflect the fluctuation of the effective data rate assigned to an audio channel. This channel is built using SIP signaling. The figure also reflects how the system slowly re-adapts to smooth the sharp loss by using different codecs to minimize the disruption in service usage. Figure 5 compared to Figure 4 shows fast adaptation capabilities whereby the fall-back duration is shorter, it is detected earlier and negotiation and adaptation thus takes less time.
The same case is carried out for video (Figures 6 and 7). Despite that very few video codecs exist in the Java Media Framework for video, within the same codec it is possible to vary the data rate to effectively cope with new resource situations. After a sharp drop due to network type change or joining an overloaded cell with a much lower profile, the system uses a codec with better compression rate (as a trade-off to more CPU and battery power use) to deliver a higher effective rate to the user on the audio channel using a lower bandwidth resource level. For instance, from 256 Kbps video a drop to 54 is mildly corrected upwards by the system to reach almost 96 Kbps using a more effective codec. This improves the user experience and stabilizes the profile over the long run and during a service usage cycle. Autonomic communication aims at proactive behavior and fast adaptation mechanisms.
A self-aware QoS system has detailed knowledge of its components, current status, available capacity, and all connections to other systems. Self-configuration is the ability of a system to automatically configure and reconfigure its settings based on the condition changes in the communication environment it runs in. Providing a QoS architecture enriched with semantics and a self-aware, self-configuring resource management mechanism will reduce complexity, lower session signaling delays, allow proactive behavior, and improve the utilization of network resources.

Therefore, starting with semantic descriptions and ontologies, moving on to making networks aware of their resource status and general state, going through allowing resource management mechanisms to profit from network self-awareness paves the way for the desired solution. Additional information embedded in semantics or extracted via one-step reasoning as Figure 3 shows, enriches network nodes and allows protocols to be designed in a way that exploits this information. The lengthy negotiation and signaling
processes used in session management can be vastly improved by exploiting self-awareness information. The AC paradigm thus provides a way of reasoning and a design methodology both of which pay off well when applied to OANs.

1.5 My Contribution

Since autonomic networking is centered on self-* features and functionality, and the first step in that direction is self-awareness, I deal with this issue in [ME3]. In particular I look into and solve the issue of mutual network-service awareness in the future Internet and emerging architectural platforms. I propose a middleware architecture that makes applications and services network-aware in the sense that they utilize network resources and tune their own demands based on what the underlying networks can offer. On the other hand, the middleware makes networks service-aware so they adapt their configuration to service demands.

This thesis deals in depth with two of the five discussed self-aware principles: self-awareness of networks and self-configuration of resource management processes. Often, the different autonomic principles are related. My contribution to the state of the art in this domain is on conceptual, design, and practical levels. The first step of the work is performing data design and semantic modeling, forming ontologies to carry and preserve information, down to designing a state space for network nodes and enabling them to be aware of their state and resource status. The next logical step of the contribution is to take existing protocols for resource management of multimedia traffic and to design new protocols that on the one hand exploit the network self-awareness feature achieved and on the other hand act as autonomously as possible. Autonomous behavior of resource management protocols requires fundamental design that deals with the fine details of session signaling protocols and session description protocols. Furthermore, algorithmic design is required in order to perform the computations of resource values, states, and other parameters required for such a scenario to work. Here I use the Session Initiation Protocol (SIP) for signaling and the Session Description Protocol (SDP) for session specification. The thesis describes a new self-configuring resource management algorithm named the Parameter Injection Algorithm (PIA) and a protocol for aggregating information from different sources called the Integrated Communication and Signaling Protocol (ICSP).

Besides the invited papers, editor-reviewed papers, the technical reports and presentations that reflected my research work done for this licentiate, the following is a list of peer-reviewed publications directly related to the scope of this thesis.


In this paper, my main contribution was to the architecture of the Open Access Network
that provides seamless mobility to roaming users. Using a hybrid Mobile IP and SIP stack as a mobility solution was also a contribution of mine.


My main contributions were: building a test-bed consisting of five tiers and that is equivalent to a real mobile network operator, deploying systems based on different architectural paradigms in the constructed test-bed such as open access networks and Beyond 3G networks, and setting up and running experiments whose measurements allowed to highlight different architectural benefits and evaluate system behavior and potential.


In this paper, my main contribution was the architectural approach for building a real-time multimedia communication service. I performed the analysis of top-down versus bottom-up service construction evaluating the Open Service Access (OSA) modular enabler-component based approach versus the multimedia Application Server (AS) black box approach versus the hybrid approach verified on a real IMS service.


In this paper, my main contribution was the mutual network-service awareness approach and coordination based on a clean-slate network architectural design. I analyzed in detail the self-awareness aspect of a network which is inspired by the Autonomic Communication (AC) paradigm.


This paper is entirely and exclusively my work. It builds upon the experiences and results in the area of Open Access Networks (OAN). It presents a self-configuring resource management mechanism that autonomically handles media streams during mobility.

Glasgow, UK 24-28 June, 2007

In this paper, my contribution was in analyzing the existing signaling on various layers for bandwidth control and in proposing an improvement in the signaling flow using layer 2-layer 4 cross layer design (CLD).


Most of the paper results are my contributions: the architecture, the session management algorithm, the system implementation with a hybrid MIP-SIP stack and the measurements. The metrics used to analyze system behavior and performances are joint work.

1.6 Structure of this Thesis

The thesis is structured as follows. After this introductory chapter that presents the different areas and links them then highlights the problem and the devised solution insights, Chapter 2 extensively reviews the areas of discourse and state of the art work. Chapter 3 provides background information on the various protocols and architectural paradigms used in this thesis. Then Chapter 4 deals with protocol design where high-level protocol aggregating information from different sources is presented. Chapter 5 presents process design with algorithms that are autonomic in nature and which were designed and used in order to improve resource management for multimedia traffic in open access networks. Chapter 6 then describes the data design part of autonomic communication for our resource management scenarios. Chapter 7 concludes the thesis and provides an outlook on future work and the map of the road ahead.
2 Related Work

This chapter extensively surveys the state of the art in the areas: resource management in wireless networks, context aware QoS network systems, and multimedia session setup and management in wireless networks. In addition, the work is positioned with respect to the IEEE 802.11r and 802.21 standardization frameworks. Those categories are handled in consecutive sections. The first section is related to the core contribution of this thesis, namely autonomic, self-configuring resource management. The second section relates more closely to the chapter on data design where context information and semantics play a key role in making networks self-aware. The third section analyzes challenges related to setting up and managing multimedia sessions in different wireless network types. The fourth section shortly overviews work ongoing in standardization bodies. Self-awareness and self-configuration are the two aspects of the Autonomic Communication paradigm that are analyzed in this thesis. The thesis positions itself with respect to existing research work and highlights the delta that was achieved in terms of the two afore-mentioned self-* features. Furthermore, an architectural perspective is given with a comparison to what is available today.

2.1 QoS Resource Management for Media Streaming with Focus on Self-Awareness and Self-Configuration

Active networking is a significant paradigm in the network evolution history and it has partially inspired autonomic communication. Active networking allows highly complex and rapid "real-time" changes to the underlying network operation. This enables features like sending code along with packets of information allowing the data to change its form (code) to match the channel characteristics [AN1].

In this section, the state of the art concerning QoS architectures and mechanisms for media streaming is investigated. We conduct a survey on how far different approaches address the network resource management issue for multimedia and streaming applications. This allows finding out the limitations of existing work, especially concerning the self-awareness and self-configuring capabilities of QoS mechanisms.

Gelenbe et al. [EG1], [EG2] work on user-oriented networks and claim that those will not usually have precise information about the infrastructure at any given instant of time, so their knowledge should be acquired from online observations. Thus, they suggest that user-oriented networks should exploit self-adaptiveness to try to obtain the
best possible QoS for all their connections. They illustrate how self-awareness, through online self-monitoring and measurement, coupled with intelligent adaptive behavior in response to observations, can be used to offer user-oriented QoS. The work is based on ongoing experiments conducted in the “cognitive packet network” test-bed.

The solution provided in this research work is suitable for several domains including sensor, enterprise, home, and military networks. The system aims to support a broad range of QoS requirements imposed by the variety of network access technologies, devices, and applications. The core research issue in this work is intelligence and adaptation to the user needs and the networking environment. This covers routing behavior, self-healing, and security aspects.

Despite the existence of basic adaptation mechanisms for QoS, they are mainly static or based on basic policies exploiting very few parameters and using a limited set of rules. The authors note that learning algorithms and adaptation have very seldom been exploited in networks because of the lack of a practical framework for adaptive control especially for packet networks. The Cognitive Packet Network (CPN) system devised by the authors works on top of IP and interacts with classical networks via IP as well. CPN uses Smart Packets (SP) to collect QoS measurement information as they travel through the network.

In order to investigate the potential of using self-awareness to offer QoS to users, they developed a practical packet-switching architecture that allows a network with an arbitrary topology to observe its state in a distributed manner. These observations are then used by an online algorithm running autonomously at each node to make routing decisions based on an estimate of QoS. However, these routing decisions are restricted to certain “smart” packets, which then inform the source about the paths they have found which offer the best QoS. These paths are then used by the payload carrying packets until a better path is found by the SPs. Thus, CPN is a packet routing protocol which addresses QoS using adaptive techniques based on online measurement.
SPs enable self-observation in a network. Self-observation leads to self-awareness. Awareness covers aspects such as QoS levels of network parameters, topology, flow and path information, and power levels of mobile nodes. In the model used, they assign to each QoS class between a source and a destination pair a QoS goal. This goal corresponds to a function which has to be minimized.

The main difference between this significant state-of-the-art piece of research work and the work pursued in this thesis is as follows. The authors exploit learning algorithms and adaptation mechanisms to add intelligence and awareness to the network behavior. The information they gather is collected via smart packets moving in the network. In this thesis, semantic structures are used to model and represent all information pieces belonging to network elements as well as processes. The process of acquiring semantic information, storing it, exchanging it, and all other types of processing is built into the system in such a way that a high level of self-awareness is achieved. We make processes for resource management and content adaptation for streaming applications on the access point and the client self-aware, self-contained and self-configuring.

Samaan et al. [NS1] present a novel approach for an advanced reservation protocol, NSLP (NSIS Signaling Layer Protocol), to provide seamless real-time services to mobile users in wireless networks. The robustness of the proposed work is the result of a two-fold contribution. The first is the utilization of knowledge about user preferences, goals, and analyzed spatial conceptual maps to predict the user’s future location. The second contribution is a predictive advanced resource reservation protocol for mobile environments. Transforming information into knowledge in the system enables self-awareness. An additional aspect of the intelligent behavior is prediction, which gives a proactive touch to the system.
Patouni et al. [EP1] address the general issue of autonomic computing and communications. They claim that automated functions enhance the intelligence of existing computing and communication systems. This concept forms a new paradigm of systems with self-aware capabilities that will automatically adapt their behavior in relation to the configuration of the drastically changing environment and user preferences. They present a generic architecture for the design and deployment of nodes with self-managing and self-configuring capabilities. They validate the feasibility aspects of the proposed framework by means of a prototype that demonstrates the operation of plug and play solutions for an adaptable component-based protocol. The framework addresses the issue of self-configuration of individual components. Two types of metadata are introduced: metadata for each protocol layer of the protocol stack and metadata for the protocol components.

Bellur et al [UB1] claim that self-configuring systems need a design approach based on behavioral specification and a control layer that can use those to dynamically bind components. Their approach is based on semantic descriptions of components augmented with contextually dependent non-functional requirements for accomplishing the dynamic binding. For this purpose, they model service inter-dependencies as variability points. The control layer dynamically re-configures system behavior by mapping the variability points to components providing the needed functionality. Self-configuration reduces itself to component binding but not to the mechanisms themselves that perform resource management or reservation.

Yu et al. [YU1] developed a network self-configuring routing algorithm, called KTRA (Kernel Tree Routing Algorithm) for mobile ad hoc networks. It converts the actual network topology into a logical tree, thus only partial routing information needs to be maintained, and the update requirement is restricted in the area of a branch. Moreover, KTRA is characterized by low delay and high flexibility compared to that of proactive and reactive routing algorithms. Empirical and simulation results demonstrate that KTRA is convergent and outperforms the traditional networking and routing algorithms. Self-configuration pays off in routing in terms of performance. However, there are other more significant aspects that can still be made self-configuring, namely resource management and allocation.

Krief et al. [FK1], [FK2], [FK3] investigate self-aware management which allows the network to react and to adapt to changes. The architecture they designed is capable of self-aware management of IP networks offering QoS guarantees. This architecture uses policy-based management and multi-agent systems. The originality of the approach lies in the intention to give a real autonomy to the components intervening in the chain of services in terms of internal decisions and configuration. The solution thus enables self-aware behavior, self-provisioning, and self-monitoring of services.

Zhang et al. [ZH1] propose a path-oriented, quota-based QoS brokerage method which aims at increasing the overall call processing capability of the bandwidth broker box.
They rely on path-level admission control and link-level bandwidth allocation. Their approach differs from ours in that their architecture is very centralized, having one box with a single bandwidth broker in the backend; whereas our approach uses many last-mile QoS brokers which are light-weight and optimize capacity based on air-time quota.

Duan et al. propose for the QoS bandwidth brokerage issue to decouple the QoS control plane from the packet forwarding plane [DU1]. They store QoS reservation states and manage them inside the bandwidth broker in addition to performing aggregate and per-flow traffic management. Again their approach lacks the light-weight property and is too centralized.

Krishnamurthy et al. focus in their QoS broker on dynamic path conditions which influence the route between the source and destination involved in a bandwidth brokering scenario [KR1]. They manage admission control requests dynamically to reflect the changes occurring in a topology of moving nodes. The brokering process is then responsible for path tracking for all hops that change in an end-to-end path. Such an approach is not scalable and ceases to perform well as the number of hops between the source and destination increases.

Nahrstedt et al. integrate the resource reservation functionality with their resource broker module [KN1]. We on the other hand rather use policing and discarding traffic which violates the agreement between the served station and the broker on the last mile. Nevertheless, the approach used in [KN1] has several similarities to our resource management scheme. For instance, they use a client scheduler and client information table, a module on the terminal which communicates with entities in the network for optimizing admission control and resource utilization. We use a local resource management module on the client functionally similar to an operating system and it is responsible for resource partitioning among traffic types; that is to say, creating flows out of one flow. Nahrstedt et al. also use event-based triggering to perform updates on the allocated resources by a broker to a terminal giving a dynamic touch to the brokering mechanism.

From what has been surveyed, we observe that decision making in the area of resource management has been analyzed and solved from the architectural and signaling perspectives. However, there is barely any approach that views resource management as a self-contained process on each node interacting minimally with peer processes on other nodes. The thesis on the other hand, presents a process-based approach to solve QoS resource management whereby both the terminal and the access point are regarded as self-aware network nodes. Each of the nodes runs a self-configuring process for resource management of multimedia traffic. What is different in the approach followed in the thesis is that it increases the amount of local computations but reduces signaling overhead, thereby optimizing session management. The Session Initiation Protocol (SIP) sessions in turn handle resources of real-time traffic. The processes used combine information from OSI/ISO layers three, five, and seven.
2.2 Context Aware QoS Network Systems

Paper [WD1] by Dargie et al. summarizes different techniques and probabilistic schemes used by context-aware systems. The paper covers fuzzy logic, hidden Markov Models, Bayesian Networks, and the Dempster-Schafer theory of evidence. It surveys various approaches including logic and rule-based schemes, and probabilistic schemes. Then a conceptual architecture is presented that shows an optimal framework for computing context while using a smart combination of different techniques previewed before in the paper. Components for fuzzy logic, primitive context acquisition, aggregation, belief, and reasoning are integral parts of the conceptual architecture depicted.

Yorig et al. [YO1] work on network centric context aware services over heterogeneous networks. Context awareness is used as a key issue for making decisions on how to adapt automatically to changing conditions. They present a policy-based context model which takes into account the real implementation of context-aware services in underlying networks. Their policy-based context information model is designed based on the IETF PCIM (Policy Core Information Model) and its extensions. The core focus of the work is on context modeling which faces the challenge of how to represent contextual information in such a way that can help bridge the gap between applications that use context awareness and the implementation of context awareness itself. They use an object oriented (OO) context information model. Context information is expressed in the policy conditions in the form of elementary Boolean expressions of the form: <Context Variable> MATCH <Context Value>. Using context classification, every context entity is classified into four basic features: identity, location, activity, and time. The Context Aware System (CAS) under question can adapt itself dynamically to complex environmental changes according to the policies resulting from customer-provider contract and rules given by the network administrator.

Khedr et al. [MK1] present the design and system architecture of Ad hoc Context Aware Networks (ACAN) in a wireless environment with no pre-configuration and with spontaneous applications running according to the contextual situation. ACAN is regarded as a new trend in context-aware wireless networks and it consists of: sensors that capture the entities in the environment, the surrounding users, and a context manager agent who interprets the sensor-captured information and passes it on to a higher level for use by applications. This data is used to minimize the user interaction while maximizing the relevance of the information provided. This is automated context awareness. ACAN targets the network layer and the application layer and introduces new mechanisms for network configuration, QoS provisioning, and more dynamic adaptable and flexible applications. Most importantly, ACAN contains a new protocol stack and a Context Aware Service Discovery Protocol (CASDP) which are both innovative contributions in this area. The authors presented the ACAN architecture showing how it merges the technology of ad hoc networking with context-aware systems to provide a better, dynamic, and user-friendly network.
Chou et al. [CH1] provide an overview of the BlueSpace project at the IBM T. J. Watson Research Center. They investigate issues associated with next-generation workspaces in which integrated electronic transducers, displays, and network appliances are used to augment personalization and context awareness. To achieve this goal, they constructed a full-scale prototype office and implemented several applications, which include situation-triggered environmental adjustments and context-aware event notification. These applications use a set of infrastructure services based on XML for sharing contextual information. The prototype is viewed as a sandbox to which additional technologies and interesting applications can be added and evaluated in the future.

Mamei et al. [MM1] present TOTA (Tuples On The Air), a novel middleware for supporting adaptive context-aware activities in dynamic network scenarios. The key idea is to use spatially distributed tuples for representing contextual information and supporting uncoupled and adaptive interactions between application components. On one hand, the middleware sends and receives tuples across a network on the basis of application-specific patterns, and adaptively re-shapes the resulting distributed structures according to changes in the network scenario. On the other hand, application components can locally “sense” these fields and can rely on them for both acquiring contextual information and carrying out complex coordination activities adaptively. The key objectives of TOTA, only partially achieved by similar proposals in the area, are to promote uncoupled and adaptive interactions by locally providing application components with simple yet highly expressive contextual information and to actively support adaptivity by discharging application components from the duty of dealing with network and application dynamics. TOTA relies on spatially distributed tuples, to be injected in the network and propagated accordingly to application-specific patterns. Tuple propagation patterns are dynamically re-shaped by the TOTA middleware to implicitly reflect network and application dynamics, as well as to reflect the evolution of coordination activities.

In [MM2], the authors analyze field-based coordination and claim that it is a promising approach for a wide range of application scenarios in modern dynamic networks. To implement it, they rely on distributed tuples injected in a network and propagated to form a distributed data structure to be sensed by application agents. However, to gain the full benefits from such a coordination approach, it is important to enable the distributed tuples to preserve their structures despite the dynamics of the network. They show how a variety of self-maintained distributed tuples for field-based coordination are programmed in the TOTA middleware. Several examples clarify the approach, and performance data is presented to verify its effectiveness. Nevertheless, it is worth mentioning that the definition of such a methodology still lacks the maturity, systematic and modular style, and the wide adoption and applicability to a variety of domains that require distributed coordination with the help of context information.

Context awareness is a well evolved and very mature field of research that relates to other scientific fields. When it comes to networking, context awareness has added
intelligence, semantic richness, and functional capabilities to networks. Autonomic communication, which is the paradigm backing this thesis, is heavily based on active monitoring. Monitoring parameters in a network is analogous to network context information collection. Once a network has enough information pieces, it can process and evaluate the information to deduce meaningful conclusions about its own state (e.g. congestion).

2.3 Multimedia Signaling and Session Management in Wireless Networks

This section analyzes multimedia signaling and session management in wireless networks. In particular, it looks at which challenges the Session Initiation Protocol (SIP) faces when it comes to session construction or updating in a wireless error-prone environment.

Different wireless technologies have different access mechanisms such as point-to-point with dynamic Time Division Multiple Access (dTDMA) (e.g. WiMax), or Carrier Sense Multiple Access with Collision Avoidance (e.g. 802.11), or Code Division Multiple Access (CDMA) such as 3G networks. This difference creates variation in terms of how multimedia session setup, update, and management in general perform over each technology.

Although this thesis deals with the particular case of multimedia resource management over 802.11 open access networks, the challenges faced by this domain in other access technologies are highlighted. Furthermore, the using composite metrics to assess system performance is demonstrated [ME2].

A wide range of papers has been analyzed in parallel to conducting our work. In fact, it was observed that the SIP protocol is adopted in various access technologies and it faces a challenge in each technology it is deployed in. VoIP over 3G is thoroughly handled by Prasad et al in [RP1]. The authors analyze the different phases in setting up a SIP-based voice session and explore the different cases via numerical analysis in order to obtain an estimate on what it would cost in terms of time and resources to establish an appropriate VoWLAN or video-over-WLAN session in 3G networks.

The authors in [MJ1] explore statistical multiplexing and try to suppress the large overhead in VoWLAN. They perform voice multiplexing using a polling mechanism in the contention-free period and a deterministic priority access for voice traffic in the contention period. They also work on reducing the overhead for voice traffic.

In [MG1] McGovern et al. address the problem of link adaptation using media codecs in 802.11. Different codecs are believed to result in different levels of congestion, and thus the system the authors propose switches back and forth between codecs to reduce the measured or perceived congestion level.

Bacciu et al. [DB1] presents a fuzzy logic approach to determine a soft admission control mechanism for Voice-over-IP services over Wireless LANs. A framework is defined where
the provider may express the network status and the client their preferences by the means of an approach based on Fuzzy Set Theory.

Brännström et al. [RB2] proposes a mobility support system integrating the benefits of application-layer SIP mobility with network-layer MIP mobility where a cross-layer information system exchanges context for mobility adaptation.

Åhlund et al. [CA1] proposes a Running Variance Metric performance metric for wireless local area networks used to predict relative traffic loads of available access points at the network layer.

The different approaches in this section either take a step in introducing combined metrics in mobility scenarios or try to solve the congestion and bad coupling of voice traffic and IEEE 802.11.

When it comes to signaling, the interactive handshake mechanisms of the protocols on different layers incur significant overhead. Cross-layer analysis and optimization is necessary in this case especially in wireless networks as this thesis deals with 802.11. In what has been surveyed, some articles present theoretical work on session setup analysis or hard-wired L3-L5 solutions tailored for specific architectures. However, there has been no focus on integrating signaling messages from both MIP and SIP and seeing them as part of the same control plane. The metrics for decision-making are computed locally in the thesis, reducing thereby the interaction among nodes when a multimedia session is set up or updated. Other hard-coded approaches have been seen in the literature that statically link e.g. particular codecs to particular congestion levels, leaving out the session management optimization aspect. We however follow the approach of step-wise resource distribution. In other words, nodes get resources as a budget which is then broken down by the node itself and distributed among different running or pending applications, streams, and channels. This has been chosen as an approach because it better suits the principles of autonomic communication.

2.4 Relevant Standardization Efforts

IEEE 802.11r

The group IEEE 802.11r [802.11r] has as main purpose finding solutions that minimize or optimize the time required for 802.11 Basic Service Set (BSS) to support real-time traffic sessions. Preparing the next candidate access point (AP) in advance during roaming in a proactive manner improves real-time and multimedia application experience over 802.11 by reducing handover time. In 802.11r, there is an AP database as a prerequisite for the algorithm to work. It is the key technical element providing the link between the current AP and the next one. There is a scanning phase to detect available access points and check their capabilities, and then a network selection phase follows to execute a fast roaming association with the next access point. We alternatively consider the Candidate Access Router Discovery Protocol (CARD) as a mechanism for determining the next AP to join. IEEE 802.11r operates on protocol layers two and
three, just as CARD does. They both need to gather information on available APs, performing reasoning on which best AP to select as the next candidate network. Even though 802.11 fast roaming (802.11r) was designed to mainly serve real-time applications and improve user-perceived quality and session continuity, there is still a substantial signaling overhead and room for optimization on higher layers. This thesis deals with this issue. In particular, distributing resources among users, applications, sessions, and channels while also reducing overhead in node interaction is a key goal of our work.

IEEE 802.21

IEEE 802.21 [802.21], also known as media independent handover (MIH), is a working group dealing with optimizing handover in heterogeneous access networks. On the other hand, the scope of this thesis is open access networks that are based on 802.11. OAN are homogeneous, based on a single technology, and only use additional access technologies for backhaul and coverage extension scenarios. Due to the fact that this thesis optimizes sessions on layer five, and coordinates network and application layer behavior when it comes to resource management, MIH is a complementary setting. It certainly is beneficial to optimize and simplify handovers even when it comes to rare cases with vertical handoffs in the absence of coverage for the main access technology (802.11).
3  Background Information

As network resource management in general and in this thesis in particular is a complex process, it requires information from several sources. This chapter discusses different protocols such as Candidate Access Router Discovery (CARD), Mobile IP version 4 (MIP), Session Initiation Protocol (SIP), and Session Description Protocol (SDP).

3.1  Candidate Access Router Discovery Protocol (CARD)

We use the intermesh and intra-mesh mobility scenario to explain the benefit of the CARD [CARD] protocol for Open Access Networks. It is especially useful in network selection and assisting resource management in becoming autonomic by supplying necessary information. One important intention of building mesh networks is to let the network appear as one AP, all Mesh Access Points (MAPs) of one mesh operate at the same SSID. Based on this principle, Mesh1 and Mesh2 appear to clients as the conventional wireless LAN provided by residential gateways (RGW) RGW1 and RGW2, the AP controllers in the network. A residential gateway in an OAN controls the access point and its traffic. This abstraction makes it less complex to specify the process of inter- and intra-mesh mobility within the Open Access Network architecture.
Client1 is located in Mesh1 and has to move from one AP to another, both situated in Mesh1, using the same SSID. The mesh is not self-configured but the links are set up statically, such that the mobility broker (MB) is aware of all link states. The MB links all meshes together. When the mesh is installed and operated for the first time, the IP addresses from the participating APs are set and for each AP an entry in a table of the MB is created with the following fields, where the path variable specifies a vector of intermediate APs on the path to destination AP:

<table>
<thead>
<tr>
<th>MAC address</th>
<th>IP address</th>
<th>Mesh ID</th>
<th>Path to the AP</th>
</tr>
</thead>
</table>

This allows the MB to have a global overview of the topology and to have a lookup table for calculating handover metrics along a path to a candidate AP. As the client stays in the same subnet, the fastest solution is a Layer2 handover, without changing the terminal IP address.

![Mobility Broker](image)

Figure 10: Inter- and intra-mesh handover scenarios

Intra-mesh mobility scenario in an OAN (Figure 11):

- Client1 is connected to mesh access point MAPB3 and authenticated in the mesh of RGWB. It scans for other neighboring MAPs every period t. It caches information related to these neighboring MAPs. After having compiled a list of neighboring MAPs, it sends a CARD request to MAPB3, which is relayed to RGWB, specifying the MAC address of each of the MAPs it wants to connect to. Furthermore, a QoS request related to resources can be piggy-backed on the same message or sent in a subsequent one;
- RGWB, in this case, is not aware of the whole path to MAPB2 (which is in the same mesh) and cannot resolve the MAC address of MAPB2. It sends a CARD request to the MB to get this information;
• The MB maps the MAC address of MAPB2 to the corresponding IP address and returns this information to RGWB;
• In order to be able to predict the conditions along the path to MAPB2, the MB sends a vector including all IP addresses on the path to MAPB2 back to RGWB;
• RGWB forwards the CARD request sent by the terminal to MAPB2;
• MAPB2 and all intermediate MAPs reply with a CARD reply, including their current QoS capabilities. RGWB waits until it has seen all replies from the MAPs and sends back a CARD reply, with the best of all available QoS levels;
• Based on the information received, the terminal may reiterate the process several times, or decide to switch to the candidate MAP. For instance if MAPB2 has better QoS availability than MAPB3, it may be advantageous to switch to MAPB2;
• A MIPv4 procedure is applied for AP roaming, providing the client with the IP address he requires.

![Diagram](image)

**Figure 11: Intra-mesh handover process**

The following steps are performed for an inter mesh handover (Figure 12):
• The terminal is connected to MAPB3. It scans for other neighboring APs. After having compiled a list of neighboring APs, it sends a CARD request to MAPB3, which is relayed to RGWB, specifying the MAC address of each of the APs it wants to connect to. RGWB cannot map the MAC address of MAPA3 (which is located in the mesh of RGWA) to an IP address. It sends a CARD request to the MB to get this information;
• The MB realizes that MAPA3 is in a different subnet than RGWB. As the meshes are hidden behind the RGWs, RGWB has no route to MAPA3. Therefore, the MB returns the IP address of RGWA;
• RGWB sends the CARD request to RGWA, which is not aware of the whole path to MAPA3 and cannot resolve the MAC address of MAPA3. It sends a CARD request to the MB to get this information;
• The MB maps the MAC address of MAPA3 to the corresponding IP address and is aware of the whole path to the candidate AP. It therefore returns a vector of all intermediate MAP addresses to RGWA. RGWA forwards the CARD request sent by the terminal to MAPA3 and all intermediate MAPs;
• All MAPs reply including their current resource status. RGWA waits for the last reply and then calculates the best available QoS along the path, including its own. It then generates a CARD reply, which is sent via RGWB to the terminal;
• MIP registration is done afterwards. When this procedure is completed the handover process ends.

Figure 12: Inter-mesh handover process

A single criterion, the 802.11 signal strength, is often used for selecting a new AP. This can result in a sub-optimal decision to select a particular network. A cost function in the terminal is optimized to decide to which access point the terminal should perform the handover. Input to this cost function includes not only signal strength but also:
• Parameters related to the traffic from/to the device, e.g. data rate, packet size;
• System status related parameters, e.g. estimated bit error rate;
• User or operator preferences, e.g. cost per bit;
- Terminal related, e.g. available battery power.

The information mentioned above needs to be gathered, either locally within the terminal or with help from the network. The handover is composed of three phases: information gathering, decision, and actual handover phase. The first two phases occur before the handover while the other terminal is still connected to the network.

3.2 Mobile IP (MIPv4) and Session Initiation Protocol (SIP)

Mobile IP version 4 [MIP] is the basic mobility protocol used in Open Access Networks (OAN) due to its ability to perform fast handovers. Furthermore, on the same stack which is hybrid, a SIP module takes care of multimedia session management. This part discusses Mobile IP in context with CARD for network selection and in a scenario where there are many access points available and capable of seeing each other. Currently, the performance of seamless handover using Mobile IP is not adequate for streaming applications like VoIP, therefore the interoperability of several protocols is required. MIPv4 is considered jointly with SIP and CARD and in an integrated manner within the overall architecture rather than as a standalone module.

Due to using hybrid mobility in this thesis, with a mixed MIP-SIP stack, it is necessary to discuss SIP [SIP] and how it relates to resource management. Session Initiation Protocol (SIP) is an application level signaling protocol for managing multimedia sessions over IP. SIP supports the basic operations: session setup, tearing down, and modification of attributes. SIP provides strong and flexible session adaptation capabilities especially in terms of media settings, formats, and rates. Moreover, it enables mechanisms that take care of user presence and context awareness. SIP is of particular interest for many reasons such as: making use of location-based and context information for achieving user presence and adapting the session parameters of SIP-based multimedia sessions (VoIP, telephony, and conferencing) in terms of media formats, data rates, clock rates, and resource usage to the current conditions. In compound capability as Figure 13 depicts, the SIP UA is registered as a listener on the MIP Client API, to be informed about the changing of network connection parameters. This happens after a performed handover.

![Figure 13: SIP-MIP interfacing](image-url)
The MIP client is capable of detecting the loss and gain of a connection over a certain interface and automatically performs a handover to the new mobile IP interface. The SIP UA is informed about the handover by an event fired by the MIP Client API, and adapts the session stream parameters (media types and codecs) to the new network connection conditions.

The network connection parameters provided by the MIP client to the SIP client:

- Available bandwidth;
- Type of the current network (802.11, LAN, OFDM, UMTS, GPRS);
- Connection addresses (HA and Care of Address).

Figure 14 shows the overall process of SIP signaling initiated by the performed handover. Additionally the SIP UA can inform the remote SIP UA about its newly assigned Care-of-Address (instead of putting the Home-Address in the session description) within the re-negotiation process. This results in new established RTP [RTP] media streams taking a shorter way bypassing the home agent.

Based on the highlighted architectural logic whereby SIP mobility resides on top of Mobile IP, several aspects have to be highlighted which result from or are influenced by the dynamics of the system. The bottleneck or benchmark for handover delays fully depends on Mobile IP, whereby fast handover is the central focus of research session updates using SIP re-INVITE messages can be used in different ways and in different situations: upon performing a Mobile IP handover, everything is tunneled via the HA to the CoA (care-of address), so SIP users do not notice the tunneling except in cases where large propagation delays occur when moving to another cell which most probably has other QoS resource constraints than the original one.

In order to avoid long tunneling, it is necessary that the SIP session is updated within the re-INVITE message with the corresponding new location. This requires providing the new connection address of the peer in the session description part and in the contact-header of the SIP message. Additionally to this mid-session mobility binding update mechanism, pre-session mobility can be realized by updating the location of the SIP device in the SIP registrar by sending a REGISTER request carrying the new location. So tunneling can be avoided for the signaling of subsequent SIP session establishment. When used jointly with MIP, this is a cross-layer action.
Figure 14: Only session media parameter update with SIP + MIPv4
MIP is the actual delay bottleneck during handovers (since SIP is placed on top of MIP). SIP updates can be performed independently from MIP handovers (unsynchronized to avoid long cascaded delays). We still would like to give an overview of what delay figures SIP mobility or session adaptation causes:

- **Measurements:** signaling time for re-negotiation (SIP re-INVITE)
  - Peer to Peer SIP communication case: all messages are end-to-end
  - Communication with traversal of a SIP proxy

- **Negotiation:**
  - Offer is packed in SDP as body of the SIP re-INVITE request;
  - Answer in SDP in SIP 200 OK response.
Figure 17: SIP communication via a proxy

Figure 18 outlines the individual segments over which we measured the delay times for the SIP part of communication. Table 1 outlines the individual values averaged statistically (after taking 10 measurement samples). Each column in Table 1 corresponds to a single SIP message, as shown in Figure 18.

Figure 18: Delay segments for SIP performance table

Figure 18 explains why values in the 2nd column in Table 1 are larger for the Ethereal measurement case than for the stack case measured with soft timers.

Table 1: SIP performance measurements

<table>
<thead>
<tr>
<th></th>
<th>1./1.+2. re-INVITE</th>
<th>Local delay on the answerer side:</th>
<th>2./3.+4. 200 OK:</th>
</tr>
</thead>
<tbody>
<tr>
<td>peer-to-peer</td>
<td>28,3 ms (SIP stack, L7)</td>
<td>49,1 ms (SIP stack, L7)</td>
<td>27,9 ms (SIP stack, L7)</td>
</tr>
<tr>
<td>UA-proxy-UA</td>
<td>46,7 ms (SIP stack, L7)</td>
<td>39,4 ms (SIP stack, L7)</td>
<td>31,3 ms (SIP stack, L7)</td>
</tr>
<tr>
<td></td>
<td>27,3 ms (Ethereal, L2/L3)</td>
<td>57,3 ms (Ethereal, L2/L3)</td>
<td>4 ms (Ethereal, L2/L3)</td>
</tr>
</tbody>
</table>
3.3 Session Description Protocol (SDP)

We take into account the fact that many types of traffic may be running between a terminal and an access point. We focus on real-time traffic that uses SIP for signaling and the Session Description Protocol (SDP) for configuration. An SDP [SDP] frame is an exact descriptor of the Real-time Protocol (RTP) streams for a particular session.

SDP includes information about: the session name and purpose, durations when a session is active, media types comprising the session, and logistical information for session recipients such as formats, ports, and address. Furthermore, resources necessary to participate in a session may be limited, so additional information is possible to include within SDP, e.g.: information about the bandwidth to be used by the conference, and contact information for the person responsible for the session. We outline below the media related fields that comprise part of an SDP frame together with their explanations:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>“m”</td>
<td>media name and transport address</td>
</tr>
<tr>
<td>“i”</td>
<td>media title</td>
</tr>
<tr>
<td>“c”</td>
<td>connection information</td>
</tr>
<tr>
<td>“b”</td>
<td>bandwidth information</td>
</tr>
<tr>
<td>“k”</td>
<td>encryption key</td>
</tr>
<tr>
<td>“a”</td>
<td>zero or more media attribute lines</td>
</tr>
</tbody>
</table>

One important point is that a lot of negotiation can take place with SDP frames going back and forth between peers till:

1. It is insured that enough resources are available for a session;
2. Both parties are satisfied with the parameters: codec, rate, QoS tags [RFC3312].
4 Protocol Design

This chapter discusses protocol design. Protocols are at a higher conceptual level than processes. Having a self-managing process on the terminal and one on the access point controller necessitates communication between the two in order to form a comprehensive protocol. This high-level protocol is called the integrated communication and signaling protocol (ICSP). It is based on the parameter injection algorithm (PIA) [ME4] that automates resource control for real-time traffic in open access networks. Since information from several OSI layers is used by the individual processes and atomic standardized protocols, a cross layer and inter-protocol analysis is carried out. Performance evaluation based on metrics is also performed.

4.1 Terminal and Access Network Two-Way Signaling

The aim of this part is to present a new methodology for automatically configuring the parameters of the session to initiate multimedia channels with well defined settings. This should be done in a way which insures that real-time traffic gets the most appropriate resource allocation dynamically.

With two self-configuring processes, one being on the terminal and the other on the access point controller, it is important to coordinate their behavior and integrate them into a higher level protocol. The result would be highly automated resource management for real-time traffic and multimedia applications.

Resource control and management over the wireless part is solved in this thesis using individual autonomic processes on each of the end-points of the wireless last-mile and then by integrating those two processes into a high-level protocol (Figure 19). Because our resource management process involves information pertaining to several signaling protocols from different domains, we call it the Integrated Communication and Signaling Protocol (ICSP). ICSP includes the communication between the resource management entities on the two ends (which is the focus of this section) plus control messages from other protocols CARD, MIP, and SIP.
When two parties are involved in a multimedia communication session for e.g. VoIP telephony, real-time video, or conferencing, then in the classical case, the offer-answer model is used with lots of negotiation going back and forth until the session is set. Furthermore, during such a session, when either of the nodes moves, or the dynamics of the resource situation change (e.g. more nodes join the access point) capacity redistribution is necessary to cope with the new load).

As network conditions change quite often, frequent resource negotiation distorts the perceived quality of real-time multimedia sessions. For instance, voice is sensitive to jitter more than to any other network quality of service (QoS) parameter. For mobile OAN users, the quality of e.g. voice over 802.11 (VoWLAN) depends on the service level delivered by the visited access point. The performance of that particular AP then depends on the operator or service provider to whom it belongs and also on the load it experiences. For mobility scenarios, where the user roams in 802.11 OAN, there is a frequent handover taking place using MIP and session adaptation at OSI/ISO Layer 5 using SIP. This scenario for voice and video over open access networks based on 802.11 is the main setting for this thesis and forms its technical scope.

To portray the impact of frequent handovers in multi-hop networks and open access networks, an analysis on session setup delays was conducted. During a single voice or video session, when several access points are traversed, a few factors become important.

Among those are:

- Pattern of joining, utilizing, and quitting an access point;
- Tradeoff between session optimization overhead and the time spent with the acquired session properties;
- Tradeoff between network resources for QoS and the obtained quality of experience (QoE);
- Degree to which session management can be made autonomic and how network context information can be exploited to improve the process.

For a mobile pedestrian user, having an average speed of 4-5 km/hr or 1.3 m/sec, he would traverse a distance of 50 meters in 38 seconds and a distance of 100 m in 76 seconds. There is plenty of time for a mobile terminal to perform a handoff from one access point to another. However, for a vehicular speed of 50 km/h or 13.9 m/s, an
access point coverage range of 50 meters would be traversed in about 3.6 seconds and one of about 100 m in approximately 7.2 seconds. So the higher the mobility speed, the harder it is to perform a handoff and the less time it is desirable that a session setup or update consumes on the radio link. Real-time applications in particular have strict requirements on network QoS parameters. Thus the pattern of motion analysis allows optimizing the design of session adaptation mechanisms.

To solve the time limitation challenges, it is necessary to design an architecture that enables different signaling paths for improved session management. It has to be ensured that all parameters needed to build or update a multimedia session (e.g. voice) are available in real-time. Session setup time in multimedia using SIP is affected by the quality of the used link, and SIP uses a retransmission timer that doubles in size upon the occurrence of errors [RP1]. Therefore, it makes sense to model session signaling in such a way that the overall setup or update time is smaller (parameter injection algorithm [ME4]), reducing the probability of an error occurring. This involves cross-layer aspects which are analyzed in the next section.

As previously mentioned, OANs enable mobility within a so called distribution system where access points are connected e.g. via a virtual operator to provide continuous network coverage that enables the user to seamlessly roam from one AP to another. Now bearing in mind that 802.11b and 802.11g cells are larger than those of 802.11a, but still relatively small, high speed mobility is hard to support. For this reason, we analyzed the cases with pedestrian speed mobility and low to moderate speed vehicular mobility, up to 50 km/h.

Pedestrian speed is for a user roaming with his terminal and traversing many 802.11 access points in the process. Low and moderate speed vehicular motion is for certain types of transportation which have speed limitations due to technical or regulatory reasons.

Table 4: Motion speeds and average durations of stay within an 802.11 access point

<table>
<thead>
<tr>
<th>Case</th>
<th>km/h</th>
<th>m/s</th>
<th>T100 (sec)</th>
<th>T50(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>5</td>
<td>1.38</td>
<td>76.92</td>
<td>38.46</td>
</tr>
<tr>
<td>Riding</td>
<td>15</td>
<td>4.17</td>
<td>23.98</td>
<td>11.99</td>
</tr>
<tr>
<td>Slow Vehicular</td>
<td>20</td>
<td>5.56</td>
<td>17.99</td>
<td>8.99</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>6.94</td>
<td>14.41</td>
<td>7.20</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>8.33</td>
<td>12.00</td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>10.29</td>
<td>10.29</td>
<td>5.14</td>
</tr>
<tr>
<td>Moderate Vehicular</td>
<td>40</td>
<td>11.11</td>
<td>9.00</td>
<td>4.50</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>12.50</td>
<td>8.0</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>13.89</td>
<td>7.20</td>
<td>3.60</td>
</tr>
</tbody>
</table>

In Table 4:
- The first column presents mobility speed in kilometers per hour;
- The second column presents mobility speed in meters per second;
• The third column shows T100, the time in seconds it takes a terminal to traverse an access point with coverage diameter 100 meters;
• The fourth column shows T50, the time in seconds it takes a terminal to traverse an access point with coverage diameter 50 meters.

As Table 4 shows, there is enough time-room for a station to perform a handover, but the session update or setup delay can become comparable with the station’s duration of stay within an access point if the error rate on the wireless part is too high [RP1].

Theoretical calculations in Table 5 show that the session setup delay for SIP in 3G networks used with UDP is about 4.61 seconds when using the Radio Link Protocol (RLP) and a transmission rate of 9.6 Kbps, and 2.9 seconds when using a transmission rate of 19.2 Kbps. The setup delay time for a SIP-based session is a few seconds over a narrow-band rate. For 802.11 OAN (broadband), this setup time can vary but would still be in the range of a 100-200 milliseconds. User-perceived QoS also known as quality of experience (QoE) would still be affected by hundreds of milliseconds.

The same parameter sequence is necessary when updating a session due to mobility or upon the need to change session parameters (e.g. codec, resources). The only difference between a session update and a session initiation is that the former uses a SIP RE-INVITE whereas the latter uses SIP INVITE. Therefore, we make the observation that even a session update in 802.11 which involves pure signaling can substantially disrupt or degrade a voice call when traversing many cells. In VoWLAN, it is necessary to consider different signaling alternatives that reduce signaling delay and provide more bounded jitter with fewer spikes and a lower jitter on the average, yielding a much more acceptable perceived audio quality. In the thesis we analyzed classical negotiation versus parameter injection with self-configuring resource management on the terminal and AP.

<table>
<thead>
<tr>
<th>Messages</th>
<th>Payload size(bytes)</th>
<th>Message size(bytes)</th>
<th># frames, 9.6 Kbps</th>
<th># frames, 19.2 Kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP (RE)INVITE</td>
<td>700</td>
<td>728</td>
<td>37</td>
<td>19</td>
</tr>
<tr>
<td>SIP 183</td>
<td>835</td>
<td>863</td>
<td>44</td>
<td>23</td>
</tr>
<tr>
<td>SIP PRACK</td>
<td>538</td>
<td>586</td>
<td>30</td>
<td>16</td>
</tr>
<tr>
<td>SIP 200 OK</td>
<td>545</td>
<td>573</td>
<td>29</td>
<td>15</td>
</tr>
<tr>
<td>SIP 180</td>
<td>349</td>
<td>377</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>SIP ACK</td>
<td>300</td>
<td>328</td>
<td>17</td>
<td>9</td>
</tr>
</tbody>
</table>

The total payload size in bytes for a SIP session setup or update is: $700+835+538+545+349+300 = 3267$ bytes. This is a lot for a lightweight signaling protocol. Calculating the size of the total data required, which includes payload and overhead yields: $728+863+586+573+377+328 = 3455$ bytes.
Then it is obvious that a modified signaling scheme that spares several of the messages listed in Table 5 would significantly reduce session setup time and also the probability of errors occurring during updating the session when roaming between APs.

The higher the used rate, the fewer frames are needed to transmit the required information. As the rate in Table 5 is doubled being first 9.6 and then 19.2 Kbps respectively, the number of frames needed shrinks to about a half. Therefore as the rate is increased, the number of frames decreases proportionally. 802.11 links are error prone. Thus for a particular frame error rate (FER) e.g. 0.01%, the probability of getting an error becomes smaller as the number of transmitted frames decreases. This makes session signaling delay cycles for sessions with higher rates lower.

![Figure 20: Basic architecture and delay components](image)

Now using Table 5 and Figure 20 we perform the following calculations. For one end-to-end signaling message in terms of resources and delay

\[
delay_{\text{total}} = 2(\text{proc}_d + \text{prop}_d + \text{prop}_c) + \text{prop}_b
\]  

With proc\_dc being the processing delay on the client, prop\_dc being the propagation delay towards the AP, prop\_dw is the processing delay on the 802.11 AP controller, and prop\_db the propagation delay on the backend, respectively. The value delay\_total stands for one cycle of E2E delay between two clients using SIP for a VoWLAN session. The processing delay for various messages depends on the type of the message received or sent but take pretty close values.
For the classical case of negotiating session parameters either when setting up an initial session or upon updating the session settings due to mobility or resource reallocation, six cycles are needed, one for each of the rows in Table 5. Hence, the entire signaling cost is approximately six times the delay_total variable.

In order to measure this value for a session setup, we used Java timers within our SIP-stack based on the JAIN-NIST base version. The timers, which are based on threads, are started when a frame processing starts and stopped when the other party (correspondent client) finishes its processing. This way, the E2E delay in a real-implementation of the system under question is measured.

This negotiation can also take place during an active session due to the need to change resource usage or due to circumstances such as handover decisions. For instance, when the signal strength measured at a terminal keeps getting weaker and a stronger signal is in view, then a handover has to take place and then SDP has to be reconfigured based on the new circumstances, the most important of which being the assigned bandwidth.

4.2 Cross-Layer and Inter-Protocol Aspects

In this thesis, the concept of hybrid SIP MIP joint mobility has been addressed in the context of OANs where advanced resource management techniques are needed for application session adaptation to changing network conditions. We used a hybrid stack as a mobility solution for multimedia. MIP is used to perform fast handover and association to a new access point and to acquire a new IP address. Then we used with the Birdstep [BS1] MIP client a specially modified API that sends updates and event notifications upwards. In this way, the SIP mobility module which operates on logical layer five in the ISO/OSI model can then adapt a multimedia session (either voice or video) to the new network conditions and after MIP has handled the basic mobility part.

The basic idea is to combine the strengths of MIP for doing fast handovers and the strengths of SIP for powerful session adaptation capabilities. The reason for this is the fact that multimedia is often handled with SIP or peer protocols and 802.11 environments have relatively small cells, so when traversing them and holding a VoIP call, the session update delay upon incurring handovers is a significant parameter. We analyze this value and show how real measurements of an existing prototype depict the improvement in performance.

The major message of cross-layer design in this thesis is two-fold:

- Making applications and services fully network-aware in the sense that they utilize network resources and tune their own demands based on what the underlying networks can offer. An in-depth architectural analysis on this aspect is provided in [ME6];
• Making networks service aware in the sense that networks also adapt their configuration to service demands. Test-beds tailored for e.g. operator-specific needs is an example [ME7].

The core point for this thesis is achieving mutual network-service awareness and coordination (Figure 21) [ME3].

The system monitors the state of the communication on different layers. Monitoring involves the collecting information from underlying networks about their status in terms of available resources and low-level network parameters. The collected network related information is provisioned to the related applications in a standard and structured format. Based on the changes in these collected network information pieces, the adaptation of the application as an answer to changes in the network layer takes place. Yet the system core logic can be loaded with policies, which makes adaptation possible for network agnostic applications.

Cross-layer control logic also provides information towards the network protocols, if they have a channel to receive this information. These information pieces include service classes and their requirements. By making use of service data, and also using network context data organized in a structured way and made easily accessible, the network node can be self aware and realize various self-* aspects. These include: self-monitoring, self-healing, self-organizing, and self-awareness. Optimization and smoothing of behavior for improved performance has been handled in different cross-layer approaches also within the scope of AC. In [ME5], TCP acknowledgement suppression in joint L2-L4 cross-layer design is suggested in 802.11 networks to boost the goodput. Reducing the volume of
overhead bytes using more intelligent signaling that requires fewer ACKs pays off. This thesis focuses in a more in-depth manner on L3-L5 improved signaling as analyzed and discussed in [ME1], [ME2], [ME4], and [ME8].

SIP and MIP from an architectural perspective can be used in a cross layer manner in different contexts within OAN as the autonomic communication paradigm encourages:

- Multimedia application (using SIP) knows from other applications e.g. GPS or CARD that mobility will take place; therefore it signals this towards MIP downwards [ME3];
- MIP signals towards SIP upwards that network changes have occurred [ME2, ME4, ME8]. MIP-SIP combined mobility on the same stack has several choices:
  - Tunnel SIP over MIP keeping the same multimedia session;
  - Performing re-invite with the new location;
  - Performing full update and re-invite for new resource settings and location update.

Depending on the context (mobility speed, duration of stay in a cell, admission control mechanism, resource scarcity), different options of the ones listed are applied.

4.3 Automated Resource Control over Last-mile 802.11 Networks: Parameter Injection Algorithm and ICSP Protocol

After viewing cross-layer aspects of resource management in OAN and autonomic process modeling on the terminal as well as the access point, we now move on to discuss ICSP (Integrated Communication and Signaling Protocol) and PIA (Parameter Injection Algorithm).

Computations of new multimedia session settings are done using a client-based self-managing process. Parameter injection is an algorithmic process that represents the core of autonomic resource management and of mutual network-service awareness. Injection is about automatically adjusting session parameters for resource management avoiding as much negotiation between communicating peers as possible.

In ICSP, the session update part consists of a computation phase and an injection phase. The computation phase is the core logic of the self-managing terminal process. The injection phase itself corresponds to calling a session with the newly computed parameters for reassigning the session parameters to obtain a new RTP stream with the updated configuration. This dynamic feature is supported already in many SIP stacks and media frameworks, but still may be subject to buggy stack performance (e.g. NIST) quite frequently.
Moreover, the availability of protocols such as CARD, which provide information in a proactive fashion and the availability of information on coverage, encourage profiting from this type of knowledge to reduce session signaling delay. We configure the session and perform an update on it automatically and then inject the new calculated parameters into the RTP stream using the stack on the client which is an end point of the stream. Therefore, terminals which use SIP-based real-time applications shall exchange their complete list of RTP supported payload types, once at the beginning. This delivered information is enough to determine the next session configuration when one endpoint is subject to changing channel conditions, a handover, or a resource update.
The preconditions for performing a direct injection of parameters for RTP into a running stream are listed below:

- **Availability of enough information** to compute realistic session parameters for the update; this is available thanks to certain entities within the architecture, located on the client and on the last mile;
- **Standard conformity**: this has been investigated and it is indeed the case. As stated in RFC 3264 by Rosenberg and Schulzrinne: “An Offer/Answer Model Session Description Protocol”, section 5.1 on Unicast Streams. Bypassing negotiation and changing settings in the middle of a stream is allowed;
- **Feasibility**: current RTP protocol implementations, SIP stacks, and signaling frameworks can cope with this. This is also the case, as we were able to successfully conduct parameter injection into running RTP streams bypassing renegotiation with minor difficulties.

Our MIP used for the experiments used Eager Cell Switching (ECS) as an Agent...
Discovery and Motion Detection (ADMD) method. ECS is designed to trigger immediate handover to the new foreign agent (FA) as soon as the mobile node (MN) receives an Agent Advertisement (ADV) message from that FA. Mobile IP is used on both cases of the experimental MIP-SIP stack, so the delta achieved in delay reduction and jitter bounding is due to improved signaling in the SIP part using the parameter injection algorithm.

After establishing an initial session between two peers A and B engaged in a multimedia (video, voice, or both) session, then when one node moves, it is allowed by RFC 3264 to modify the data stream and piggyback the new session settings with the new stream. This signaling procedure is depicted in Figure 23, the “session update” part.

We used four access points placed 50 meters apart as depicted in Figure 24. Human testers were walking and traversing the different 802.11b APs while the data was being recorded. Windows XP equipped laptops were used as test clients with a hybrid MIP-SIP mobility client. When the signal strength of the next AP was stronger than that of the one the client was associated with, a handover was triggered (pure MIP handover). After performing the hard handover in Mobile IP and the handoff is complete, the Mobile IP API sends a trigger to activate the "REINVITE" message on the SIP part of the stack. The QoS controller on the access point assigns a new resource pair to the client upon joining that AP. Average Packet Size (APS) and Average Packet Rate (APR).
After a client switches APs, MIP performs an update and then the new network parameters are fed into the SIP module via the AP QoS controller as well as upwards by the MIP API. "The RE-INVITE" is launched with the newly computed parameters. For a multimedia session the SIP proxy on each controller is updated with the new location (IP address) of the callee. As the user continues to move, at every cell change, there is a MIP-handover followed by a SIPS-SDP re-invite. The MIP handover delay is represented logically by the following time span: let the time-stamp of the last received packet (can be traced via ICMP) on the first interface be $t_1$, and the time-stamp of the first received packet on the new interface be $t_2$, then the handover cycle delay is $t_2-t_1$. SIP session signaling delay is measured via thread based timers. After the trigger from the MIP API reaches the SIP-part, a SIPS-SDP session update cycle is launched. In this cycle, there is a computation part and session re-launch part with new parameters. Once the session is re-launched with a data stream obtained on the stack with updated session parameters, the timer is stopped. Sample system input data is in Table 6.

The different table columns for the algorithm supply input data for the algorithm’s configuration module as follows.
A: duration of stay of mobile station in association with an AP predicted using info from CARD, as well as motion speed and AP coverage range
B: average packet size in bytes as allocated by the capacity manager on the AP
C: average packet rate acquired within same profile as average packet size
D: fraction as percent allocated by local resource manager of overall network resources to real-time traffic
E: total number of real-time applications contending for resources on the terminal

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12000</td>
<td>64</td>
<td>62</td>
<td>80</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>30000</td>
<td>64</td>
<td>62</td>
<td>70</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>20000</td>
<td>128</td>
<td>107</td>
<td>80</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>15000</td>
<td>128</td>
<td>107</td>
<td>90</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>20000</td>
<td>128</td>
<td>125</td>
<td>80</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>24000</td>
<td>128</td>
<td>125</td>
<td>75</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25000</td>
<td>512</td>
<td>125</td>
<td>80</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>30000</td>
<td>512</td>
<td>125</td>
<td>85</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>20000</td>
<td>256</td>
<td>64</td>
<td>90</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>256</td>
<td>64</td>
<td>90</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>64</td>
<td>63</td>
<td>90</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>64</td>
<td>63</td>
<td>80</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>64</td>
<td>63</td>
<td>80</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>64</td>
<td>63</td>
<td>60</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>128</td>
<td>125</td>
<td>60</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>20000</td>
<td>128</td>
<td>125</td>
<td>70</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>15000</td>
<td>256</td>
<td>125</td>
<td>70</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>256</td>
<td>125</td>
<td>70</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>64</td>
<td>61</td>
<td>80</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>13000</td>
<td>64</td>
<td>62</td>
<td>90</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>64</td>
<td>62</td>
<td>50</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>64</td>
<td>62</td>
<td>50</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>13000</td>
<td>64</td>
<td>61</td>
<td>40</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>64</td>
<td>61</td>
<td>40</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Input data to parameter injection algorithm
The negotiation algorithm interacts more with remote entities, whereas the injection scheme requires a slightly longer local computation but fewer remote interactions.

Figure 25: Delay profiles of negotiation versus injection for just the SIP part

Figure 25 shows the session update cycle delay profiles for long runs with negotiation and short runs with injection (PIA) in excellent channel conditions with stationary nodes and manually induced condition changes fed from an input file, forcing the nodes to re-adapt.

We make the following observations:

- Convergence towards a statistical pattern as verified with up to 10000 runs is observed in Figure 26 whereby the session signaling delay average stabilizes at around 38 milliseconds when using the parameter injection algorithm. Session signaling reduction by as much as a factor of 2.5 is achieved for small runs with 300 handovers or parameter update cycles and a factor of 94 milliseconds/38 milliseconds = 2.47 times is achieved for large runs which is still a significant reduction on session signaling delay;

- Figure 27 shows that the jitter profile, which does not vary from large to small runs, has a low jitter value on the average of around 10 to 15 milliseconds. This is very good for real-time traffic especially for the jitter-sensitive voice traffic category.
Figure 26: Parameter injection performance

Figure 27: Jitter profile for the injection algorithm

Table 7: Quality levels and network performance parameters

<table>
<thead>
<tr>
<th>Parameter/Level</th>
<th>Good</th>
<th>Acceptable</th>
<th>Poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>0-150 msec</td>
<td>150-300 msec</td>
<td>&gt; 300 msec</td>
</tr>
<tr>
<td>Jitter</td>
<td>0-20 msec</td>
<td>20-50 msec</td>
<td>&gt; 50 msec</td>
</tr>
<tr>
<td>Packet Loss Rate</td>
<td>0-0.5%</td>
<td>0.5-1.5%</td>
<td>&gt; 1.5%</td>
</tr>
</tbody>
</table>
Values for delay, jitter and packet loss rate (network QoS atomic parameters) can be grouped into three main ranges for quality good, acceptable and poor. The negotiation algorithm falls on the border between good and acceptable in terms of delays under good network conditions and in terms of jitter, it falls within the acceptable range, being subject to distortions and perturbations for VoIP users in sensitive wireless environments. On the other hand, the proposed injection approach provides very low jitter values and also drastically shorter session update times which never even get out of the first half of the good interval.

Figure 28: Delay components in the architecture used

Figure 28 shows, there is a delay component due to Mobile IP handovers and then on the upper part of the hybrid stack, there is the SIP delay. On each client, MIP and SIP interact with each other via an API.
The fact that a lower session update time is obtained in the second case that modifies the data stream avoiding negotiation automatically means that the user will experience a milder perturbation in the VoWLAN session. Perceived audio quality is furthermore more sensitive to session jitter, or delay variation. Using the delay values in Figure 29, we compute the jitter values and plot them in Figure 30.
Network QoS parameter values have been grouped into intervals based on the perceived quality by end-users [PC1]. Jitter values between 0-20 milliseconds show a good performance and yield a “good” perceived audio quality. Values between 20-50 milliseconds yield an “acceptable” audio quality and values above 50 milliseconds indicate a poor performance and poor perceived voice quality.

Figure 30 depicts the jitter mean value for the different session update mechanisms. The mean value is around 23 milliseconds in the classical case and 11.5 in the modified case. As a result, using the classification of jitter into intervals based on the perceived audio quality, the classical signaling case falls into the “acceptable” range whereas the modified scheme which bypasses negotiation and updates the data stream for VoWLAN on each handover directly falls within the “good” range, even when considering overall MIP plus SIP session delay. Surveyed research work as in [CA1], [RB2], and [ME2] suggests forming composite metrics adding the delay parameter to the square of the jitter or to the jitter value scaled by a linear factor. The purpose is to capture the system behavior in terms of performance and pattern of QoE and smoothness of degradation.

Using the metric \( m1 = delay + k\cdot jitter \) with \( k = 5 \), we obtain Figure 31. Both signaling schemes (negotiation and injection) used with the hybrid SIP-MIP stack show a lot of similarity. This similarity is then well reflected by the parallel curves using Metric 1 in Figure 31.

![sorted plots for Metric 1 = delay + k.jitter, with k = 5](image)

**Figure 31: Metric1 is linear in terms of delay and jitter**

Analyzing session setup and update delay cycle times in the two different cases of negotiation and injection, we observe from the available measured data set that:
Where $k_i$ is a linear constant and where all $k$ values are close to each other. $\varepsilon$ is a small correction variable that indicates the delta spread or variance. The two above equations are also valid for the mean values of the session delays and jitters.

Using the metric $m_2 = \text{delay} + \text{jitter}^2$ whereas the second case (Figure 32) shows a diversion between the curves, and does not say much about two schemes except that the injection mechanism outperforms the negotiation mechanism with hybrid mobility. The behavioral relationship is nevertheless is better seen using metric $m_1$.

![sorted plots for Metric 2 = delay + jitter^2](image)

**Figure 32: Metric2 sums linear delay and quadratic jitter**

We hence observe that for each particular pattern or scenario use-case, a specific composite metric is suitable. In our case $m_1$ turns out to be more suitable then $m_2$ due to the resemblance and correlation between the two data patterns for session setup delay. We have seen that parameter injection as an algorithm brings on qualitative and quantitative benefits when using self-managing resource management processes on the terminal and access points.
5 Self-Configuration in Process Design

This chapter discusses process design. In autonomic communications, one of the key goals is to make network nodes as autonomous as possible reducing their dependence on external interactions and processes. This largely depends on the processes that run on and control the nodes in question. Process design is thus a key aspect of autonomic communication and needs to be addressed at this point.

Since the scope of the thesis is the last-mile part of open access networks, we consider process design on the two end-points of this range, namely: the mobile user terminal and the 802.11 AP. Self-managing processes on the aforementioned end-points are discussed in Sections 5.1 and 5.2 respectively.

The main challenge for resource management in open access networks is to support a controllable and manageable quality of service level for moving users that traverse several adjacent APs. As Section 5.2 of this chapter points out, bit rate is not sufficient to properly characterize 802.11 rates and thus we introduce QoS profiles that solve this issue. The quality of service delivered by the network results in a quality experienced (QoE) individually by each user. A solution to change the session characteristics that adapts to changing network conditions was developed for the terminal, combining the capacity resource management QoS solution with both Session Initiation Protocol (SIP) and Mobile IP (MIP) to optimize the session characteristics.

Changing conditions in 802.11 channels as well as user mobility make the redistribution of network resources inevitable. The challenge is to optimally deal with the changing conditions and to think of solutions how the desired QoS level can be restored fast enough to cope with the incurring changes.

5.1 Self-Configuring Resource Management on the Terminal

The mobile terminal should be able to act as an autonomous entity. In particular, what concerns resource management and session adaptation, the terminal which hosts a peer of a multimedia session has to make sure that sufficient resources for the user on his different channels are present and are fairly distributed and updated.

The amount of available network resources (especially bandwidth) changes vastly in mobile networks. The most sensitive traffic and application category to this kind of
change is the real-time class. This chapter presents a novel methodology for allocating the appropriate amount of resources dynamically to real-time traffic in wireless networks. The methodology is tailored for deployment in OANs. Bandwidth allocation to individual users is performed by the network operator (Section 5.2) and then a self-managing session adaptation process deployed on terminals, systematically computes the exact configuration for real-time traffic channels and provides this information as part of the data flow of the parameter injection algorithm. This configuration is embedded in the Session Description Protocol (SDP) as media payload types. We use RFC specifications and apply table permutation and sorting operations to obtain the proper media types and resource configuration for real-time sessions at runtime.

In metropolitan areas with OANs, it is technically feasible to use 802.11 in an almost continuous fashion without losing coverage. Structurally, such a network is topologically analogous to a homogeneous mesh network.

The focus of this section is on the process running on the client and its self-configuring aspects. Negotiation can also take place during an active session (voice or video call) due to the need to change resource levels. For instance, when the signal strength measured at a terminal keeps getting weaker and a stronger signal is in view, then a handover has to take place and then SDP has to be reconfigured based on the new settings, the most important of which being the assigned bandwidth (average packet rate, average packet size). We distinguish three key components which are essential for our algorithm to run, namely: the bandwidth broker, the SIP broker for real-time applications, and the mobile terminals which stands for two key elements: device profile and operating system.

The self-adaptation process requires re-computing resources for a terminal in order to make it reconfigurable and prone to fluctuations in network conditions during mobility. Now having briefly seen the signaling protocol SIP and the session specification protocol SDP, it is now essential to describe the dynamics of resource reconfiguration from a client device perspective.

Consider the following scenario. A user is engaged in a VoIP call, based on SIP and the session continues as he traverses several 802.11 access points. As the user moves from one access point to another, the available resources on the newly joined access point may vastly differ due to several factors, e.g. load, roaming agreements, or simply the used policy. Therefore, it is very likely that the access point controller that admits the roaming user offers a very different bandwidth resource value in the form of packet size and packet rate product.

Semantics are very important in shaping network behavior. This is also a case when this comes in handy. Knowing the semantics of the scenario allows performing the necessary computations and transformation on a node in order to redistribute its resources optimally for the new conditions in the newly joined AP.
There are several important issues at this point. First of all, real-time traffic is sensitive to bandwidth fluctuations more than other traffic classes are. Therefore, it is good to minimize the effect of fluctuations and degradation by taking this into account when designing the self-configuration process itself. Second, there are many parameters that need to be fed into the process. Since the SDP protocol uses so-called payload types (PT) which are codes for media including raw format and encoding and playback rate, this is a value that has to be computed. Moreover, each PT has a different aggregate average rate, and this is not provided explicitly in available RFC specifications. Third, a terminal has to fairly manage the bandwidth budget it gets in order to distribute it among its running applications (a part of all applications is multimedia and real-time, and the rest is non-real-time traffic like web, FTP, or mail) and different channels of various applications (e.g. voice and video in a conference call). Fourth, it is desirable to reduce negotiation messages between peers during mobility unless necessary. Last, there are different codecs in either software or hardware that an operating system supports.

In order to highlight the different aspects of the resource self-managing process, a step-wise formulation is presented in what follows. The resource management process which computes and allocates resources to real-time streams on the terminal requires information from several different input sources as seen in Figure 33:

- The average packet size (APS) and average packet rate (APR) from the QoS controller on the AP. The APS and APR together form the airtime allocated to a station and their product corresponds to the network resources granted to a terminal.
- The device profile signifying which media formats are supported by a terminal (codecs in software and hardware).
- Information on the terminal from a local resource manager which determines what fraction of the overall network resources goes to each traffic class. The aim is to indicate how much of the overall bandwidth or airtime owned by a terminal is allocated to real-time traffic.
Another potential useful piece of information is the estimated duration of stay of a mobile station within the vicinity of an access point without being subject to major reduction in assigned airtime. This is predicted using parameters such as motion speed of the terminal, coverage size (range) of the cell to be visited, and any other supplementary information supplied to the terminal via the CARD protocol or other sources depending on the architecture used.

Table 8: Extended PT table from statistical measurements

<table>
<thead>
<tr>
<th>Encoding (PT)</th>
<th>Media Type</th>
<th>Clock Rate (KHz)</th>
<th>Data Rate (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PCMU</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>3</td>
<td>GSM</td>
<td>8</td>
<td>13.2</td>
</tr>
<tr>
<td>4</td>
<td>G723</td>
<td>8</td>
<td>5.3, 6.3</td>
</tr>
<tr>
<td>5</td>
<td>DVI4</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>DVI4</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>LPC</td>
<td>8</td>
<td>5.6</td>
</tr>
</tbody>
</table>

Table 8 is an extension of the regular SDP payload type table with one added column on the right which depicts typical data rates for various payload types.
Swapping the table columns leads to what is seen in Table 9. The process has to map a data rate pair (as APS and APR) into a corresponding PT for each audio and each video channel of each running real-time application. This is exactly what we are doing here. For the aforementioned reasons, the transformation from the RFC table to Table 8 takes place via extension and averaging (sampling) whereas that from Table 8 to Table 9 is via swapping the first and last columns. Rotation is important because in OANs, the capacity manager delivers a budget and it is up to the client to break it down and allocate resources to different sessions. Therefore, the primary field of the input is the rate and the resulting parameter set includes PTs. The order of columns in the table simply reflects the time sequence in which parameters are available.

### Table 9: Reversed table for PT lookup based on data rates

<table>
<thead>
<tr>
<th>Data Rate (Kbps)</th>
<th>Clock Rate (KHz)</th>
<th>Media Type</th>
<th>Encoding (PT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>8</td>
<td>PCMU</td>
<td>0</td>
</tr>
<tr>
<td>13.2</td>
<td>8</td>
<td>GSM</td>
<td>3</td>
</tr>
<tr>
<td>5.3, 6.3</td>
<td>8</td>
<td>G723</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>DVI4</td>
<td>5</td>
</tr>
<tr>
<td>64</td>
<td>16</td>
<td>DVI4</td>
<td>6</td>
</tr>
<tr>
<td>5.6</td>
<td>8</td>
<td>LPC</td>
<td>7</td>
</tr>
</tbody>
</table>

This scheme performs a sequence of simple operations such as table column extensions, rotations, sorting, and row-merging. The process on the terminal interacts with a capacity management process on the last mile in the residential domain. The broker allocates resources in the form of airtime budget periodically to the clients. Then an algorithm is needed to run on the client side and interact with the last mile broker in order to read the new budget and then manage it locally. The overall solution performs autonomic resource management for different media channels belonging to real-time applications.

Table 10 shows the mapping table for audio; based on RFC 3551 [RFC3551], PT (Payload Type) values for various bandwidth ranges. It is obtained from by aggregating several rows from Table 9 into ranges after sorting the rows using the data rate in the first column as a primary field. Entries in the second column of the table are filtered based on the device profile which should contain a list of software and hardware-based supported formats and codec groups. For instance, a client gets assigned for the audio channel on the uplink 6 Kbps; therefore we end up in the 1st row of the table in Table 10 within the 2nd column with the formats LPC which corresponds to PT code 7, G723 PT which maps to code 4 and QCELP corresponding to PT code 12.
Table 10: Resulting mapping table for audio PTs

<table>
<thead>
<tr>
<th>Bandwidth Range Audio Channel</th>
<th>Corresponding payload types and encodings</th>
<th>Filtering Stage (Profile dependent)</th>
<th>Final PT Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 8 Kbps</td>
<td>7(LPC), 4(G723[5.3/6.3]), 12(QCELP)</td>
<td>Each element in this column is a subset of its left neighbour; filtering based on device profile</td>
<td>7, 4, 12</td>
</tr>
<tr>
<td>8 - 16 Kbps</td>
<td>18(G729), 3(GSM)</td>
<td></td>
<td>18, 3</td>
</tr>
<tr>
<td>16-32 Kbps</td>
<td>15(G728), G726-16, G726-24</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>32-48 Kbps</td>
<td>5(DVI,8KHz), 16(DVI,11KHz), G726-32, G726-40</td>
<td></td>
<td>5, 16</td>
</tr>
<tr>
<td>48-64 Kbps</td>
<td>-</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>64-128 Kbps</td>
<td>8(PCMA), 9(G722), 6(DVI,16KHz), 17(DVI,22KHz)</td>
<td></td>
<td>8, 9, 6, 17</td>
</tr>
<tr>
<td>&gt; 128 Kbps</td>
<td>14(MPA), 1011(L16)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Resulting mapping table for video PTs

<table>
<thead>
<tr>
<th>Bandwidth Range for Video Channel</th>
<th>Corresponding payload types and encodings</th>
<th>Filtering Stage (Profile dependent)</th>
<th>Final PT Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-64 Kbps</td>
<td>34(H263QCIF (352x288)), 32(H261QCIF (352x288))</td>
<td>Each element in this column is a subset of its left neighbour; filtering based on device profile</td>
<td>34, 32</td>
</tr>
<tr>
<td>64-256 Kbps</td>
<td>34(H263CIF (176x144)), 32(H261CIF (176x144))</td>
<td></td>
<td>34, 32</td>
</tr>
<tr>
<td>&gt;256 Kbps</td>
<td>26(JPEG), 32(MPV)</td>
<td></td>
<td>26</td>
</tr>
</tbody>
</table>

Assuming the device profile supports only the LPC and QCELP CODECs, only SDP PT parameter values “7” and “12” will be assigned to the audio part of the media channel for the uplink for the particular node in question. Our algorithm then follows the same procedure for video media channels to configure the PT codes for the SDP offer to be sent out. It is worth noting at this point that ongoing research also attempts at refining the bandwidth or data rate range bounds for finer granularity and better performance, especially concerning video encodings and formats. The same procedure is done for video payload types so as to obtain the appropriate mapping table.

The process starts with an attribute pair (average packet size, average packet rate), and performs all the necessary computations to obtain the proper target PT codes for its media data. The second part of this process is at a higher level where a node has to distribute resources among contending applications as well as different media channels within the same application.

Negotiation of session parameters has to take place at the beginning of a call setup.
Muslim Elkotob, Autonomic Resource Management in IEEE 802.11 Open Access Networks

It is important that not only parameters for a particular session start are exchanged, but also all capabilities of both end-points are shared within the first SDP frame. It is worth noting that a node usually sends to its peer the relevant codecs to the current session. Then every time there is a change in the session, resending of codecs is necessary based on the settings of the new session. To avoid this, and make the process more self-configuring, we propose to exchange all the capabilities upon session setup, and then let the process choose automatically a new codec, format and rate for resuming the session in the newly joined access point.

During mobility, communicating peers have the full list of each others’ capabilities. Therefore, terminals would not need to negotiate for a new session update because the resource information is available from the QoS broker (next section) and the full Payload Type (PT) and CODEC list is obtained in the first cycle.

Furthermore, other information is obtained by local modules on the terminal such as operating system and driver. The process of updating a session upon resource reallocation or a handover is thus composed of two main phases: a computation and information gathering phase followed by the phase where the computed parameters are then injected into the running RTP stream.

Multiplying the average packet size by the average packet rate yields the effective airtime data rate. Then this has to be multiplied by 8 to convert bytes to bits. The result has then to be multiplied by the fraction of overall bandwidth which is allocated to real-time traffic; this corresponds to the fraction (percent) variable divided by 100. Finally, the intermediate result which is the data rate allocated to real-time traffic has to be divided by the number of multimedia applications. The division by a factor of 1000 is to obtain a result in Kbps not bps. Then we assign one fourth of the bandwidth to the audio channel and the rest to the video channel within a single multimedia application (4), (5), (6).

\[
\text{Net data rate for real-time traffic per application on a terminal} = \frac{\text{DataRate}}{100 \times \text{ApplCount} \times 1000}
\]

\[\text{AudioDataRate} = \frac{\text{DataRate}}{4} \]

\[\text{VideoDataRate} = \frac{3 \times \text{DataRate}}{4} \]

Assumptions made at this point are:

- The operating system scheduler has a partitioning scheme for network resources with a proportion depending on the traffic class involved as in [WIL1]. For instance, if a local network is partitioned to have a proportion of 30 percent for
file transfer traffic and 40 percent for real-time, and the rest for background, a node can use a similar scheme that is dynamically adjustable in order to keep a reasonable balance between how much each traffic class consumes out of its total bandwidth;

- After obtaining the net rate for real-time traffic on a node from the previous step, it is assumed that bandwidth is evenly divided among real-time applications, then from experience, one fourth of the rate is given to the audio channel and three quarters to the video channel.

5.2 Self-Managing Capacity Allocation in the Access Network

This section discusses a capacity allocation self-managing process that resides on the access point and takes care of coping with resources and distributing them among different terminals. The key point is that all control logic attributed to capacity management is based on are self-managing and autonomous processes, interacting minimally with other entities in the network.

The process for capacity distribution has two main restrictions: meeting the user QoS requirements and the 802.11 wireless medium physical constraints. It is important that the access point controller fairly allocates bandwidth to users who are in the access point vicinity. Long term consistency of real-time traffic is more critical than that of non-real-time traffic. The capacity management module on the access point is subject to changes and constraints due to several factors. Among those are: fluctuations of the bandwidth due to jamming, variation of the rates on the physical layer based on the distance between the access point and the mobile terminal, and the scarcity of the resources which experience bottlenecks especially with real-time applications whose demands are usually high.

The capacity of the medium can be considered in two different ways:

- The maximum capacity of the medium $C_{\text{max}}$; it is the capacity, expressed in gross rate, which can be offered by the AP under optimal conditions. For example: 11 Mbps with 802.11b or 54 Mbps with 802.11g;
- The instantaneous capacity $C_{\text{ins}}$; this capacity is dependent on the constraints being exerted by the propagation in the 802.11 medium. The value of $C_{\text{ins}}$ is calculated by the AP on a short interval of time and is updated periodically.

The parameters defining the user capacity profile are:

- $R_{\text{max}}$, it is the maximum rate which the customer will be able to reach when the conditions allow it but that he will not be able to exceed. Moreover, $R_{\text{max}}$ does not profit from any guarantee.
- $R_{\text{min}}$, it is a minimum rate almost guaranteed which must be maintained for a particular priority level. This rate is dimensioned for real-time (RT) applications based on media types and user requirements.
and $R_{\text{min}}$ can be broken down based on the traffic direction, upstream and downstream. This breakdown is more complicated for $R_{\text{max}}$ than for $R_{\text{min}}$ since $R_{\text{min}}$ handles equally bidirectional traffic.

A classification of the associated terminals is carried out according to a level of priority, in order to be able to discard certain terminals when the capacity becomes insufficient to satisfy the needs. In the case of reduction in the capacity available, or in the case of a new request which cannot be satisfied, admission control discard actions are necessary. This classification is preserved in a data table in the access point QoS controller and updated periodically.

Table 12: Link budget management table sample for a wireless operator

<table>
<thead>
<tr>
<th>Service Class</th>
<th>APS in Byte/P</th>
<th>Uplink APR PperSec</th>
<th>Budget Kbps</th>
<th>APS in Byte/P</th>
<th>Downlink APR PperSec</th>
<th>Budget Kbps</th>
<th>Total Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silver</td>
<td>64</td>
<td>175</td>
<td>90</td>
<td>64</td>
<td>70</td>
<td>360</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>87</td>
<td>90</td>
<td>128</td>
<td>35</td>
<td>380</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>22</td>
<td>90</td>
<td>512</td>
<td>8</td>
<td>380</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>11</td>
<td>90</td>
<td>1024</td>
<td>4</td>
<td>380</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>7</td>
<td>90</td>
<td>1500</td>
<td>4</td>
<td>360</td>
<td>450</td>
</tr>
<tr>
<td>Gold</td>
<td>64</td>
<td>273</td>
<td>140</td>
<td>64</td>
<td>109</td>
<td>560</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>136</td>
<td>140</td>
<td>128</td>
<td>54</td>
<td>580</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>34</td>
<td>140</td>
<td>512</td>
<td>13</td>
<td>580</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>17</td>
<td>140</td>
<td>1024</td>
<td>6</td>
<td>580</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>11</td>
<td>140</td>
<td>1500</td>
<td>4</td>
<td>560</td>
<td>700</td>
</tr>
<tr>
<td>Premium</td>
<td>64</td>
<td>390</td>
<td>200</td>
<td>64</td>
<td>156</td>
<td>800</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>195</td>
<td>200</td>
<td>128</td>
<td>78</td>
<td>800</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>48</td>
<td>200</td>
<td>512</td>
<td>39</td>
<td>800</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>24</td>
<td>200</td>
<td>1024</td>
<td>9</td>
<td>800</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>16</td>
<td>200</td>
<td>1500</td>
<td>6</td>
<td>800</td>
<td>1000</td>
</tr>
</tbody>
</table>

In open access networks, we have two categories of users, residential users (RU) and visiting users (VU).

- RU: the ones who power up the access point and own it;
- VU: the ones who subscribe for roaming services, to use OAN access point available capacity as they roam.

The capacity management process (CMP) is used at two instances of the access phase.

- Case 1: CMP is processed at the same time as the network access authorization of the terminal;
- Case 2: CMP is processed after the authorization of the terminal and is triggered by a specific QoS request.

Case 1:
$\sum R_{\text{max}}$ and $\sum R_{\text{min}}$ integrate the residential user values RU ($R_{\text{min}}, R_{\text{max}}$) and a margin of provisioning which could be used to satisfy the needs for handover. The values $R_{\text{max}}, C_{\text{max}}$ and $C_{\text{min}}$ are expressed as brut rate values. $R_{\text{min}}$ is expressed in application net data rate;
Case 2:
In this process, the requested rate $R_{req}$ comes from a negotiation between the terminal and access point QoS controller modules. The cost of the request is computed by the AP and is compared with the free capacity before allocation takes place.

A terminal can upgrade or downgrade its allocation with a new negotiation. The AP contains a table with the subscriber profiles of the terminals and maintains a table with the negotiated profiles. Initially the CMP verifies that the sum of the requests $R_{req}$ is lower than the value $R_{min}$ subscribed.
The capacity management process (CMP) must be defined independently of the mechanisms of the MAC layer to be applicable in all the cases. However it is possible to consider a time sharing of the bandwidth between different categories: a portion for signaling, a portion reserved for RT traffic and the remaining portion for the non-real-time (NRT) traffic.

This division makes it possible to ensure a better guarantee for RT traffic and an instantaneous knowledge of the state and distribution of capacity when using e.g. polling. The capacity allocated by the CMP following the requests of the terminals and to match the RT traffic needs is called $C_{res}$. $C_{res}$ or reserved must be guaranteed and
given high priority. The reserved capacity is the sum of all made capacity allocations.

The remainder of available capacity is divided between the terminals starting from an access mode in contention. This capacity is free of reservation and is called $C_{\text{free}}$. With $C_{\text{tins}}$, the total instant capacity available, then: $C_{\text{free}} = C_{\text{tins}} - C_{\text{rsv}}$

$C_{\text{free}}$ must be kept higher than a minimal value $C_{\text{free,thr}}$ at any time. $C_{\text{free,thr}}$ must include several capacity provisioning cases:

- provision for the Residential User
- provision for mobile terminals having needs for handover between two AP
- capacity for new terminal that wants to join the AP

The profile values make it possible to determine the maximum capacities which a terminal can claim. One distinguishes the maximum rate $R_{\text{max}}$ and the maximum rate guaranteed $R_{\text{rsv}}$, which consists of the allocations carried out by the CMP for requests of RT traffic. The terminal profile is defined by these two parameters ($R_{\text{rsv}}$, $R_{\text{max}}$). With $R_{\text{max}}$ having two components, $R_{\text{rsv}}$ for guaranteed traffic and $R_{\text{nrsv}}$ for the elastic traffic with no guarantee: $R_{\text{max}} = R_{\text{rsv}} + R_{\text{nrsv}}$.

These rates are the maximum values that the terminal can reach when it is in the optimum zone of each corresponding physical rate. If an application consumes 100 Kbps with an 11Mbps PHY rate, it will consume 1100 Kbps with a 1 Mbps PHY rate. That can result in prohibiting PHY rates compatibility with profiles.

The values $R_{\text{max}}$ and $R_{\text{rsv}}$ must be defined for the two directions: uplink and downlink. A numerical example for profiling is given in what follows:

<table>
<thead>
<tr>
<th></th>
<th>$R_{\text{max}}$ down</th>
<th>$R_{\text{max}}$ up</th>
<th>$R_{\text{rsv}}$ down</th>
<th>$R_{\text{rsv}}$ up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>4 Mbps</td>
<td>1 Mbps</td>
<td>500 kbps</td>
<td>500 kbps</td>
</tr>
<tr>
<td>Silver</td>
<td>2 Mbps</td>
<td>500 kbps</td>
<td>250 kbps</td>
<td>250 kbps</td>
</tr>
<tr>
<td>Bronze</td>
<td>1 Mbps</td>
<td>250 kbps</td>
<td>125 kbps</td>
<td>125 kbps</td>
</tr>
</tbody>
</table>

The request must be transmitted in a format which can describe RT application requirements. Two essential elements are used: the size of each packet of data and the periodicity of the transmission. Thus the request is expressed in the following format: $\text{rq}(\text{period}, \text{size})$. The CMP will use these parameters to calculate the rate of the allocation it will grant to a terminal.

The distribution and the maintenance of capacity are actions based upon solely on the parameters that define the relationship between the capacity and the load which is to balance. Two instruments are necessary for the solution to work: a capacity sniffer or estimator and a local profile database.
The database is the memory of the CMP. For all new operations, the CMP refers to it and registers information which results from the modifications of the plan that the CMP computed. The database is updated constantly. Terminals granted allocations are classified in decreasing priority order. The Residential User gets the highest priority. For each terminal, the rate characteristics are added up and summed with all current allocations. Then the CMP is able to check that capacity limitations are not exceeded. Current PHY rate is also stored in the database. The sum of uplink and downlink allocations is checked simultaneously to ensure that no capacity overshooting has happened.

The database contains also the description of allocations carried out by the CMP, following requests sent by the terminals. The parameters are preserved as expressed in the request of the terminal, i.e. the periodicity and the size of the data to be transmitted. Thus the CMP can find these parameters in the database and use them to re-compute the cost of an allocation in terms of bandwidth.

The sniffer listens on the medium to the data transmissions coming from terminals. It can thus note the use of the channel and calculate the capacity really used. In particular, it can detect the collisions without knowing their origin. It also senses changes of transmission PHY rates. The sniffer operates jointly at layer one and two. When it is necessary, the sniffer sends a report about the problem inside an alarm intended for the CMP.

The principal events that trigger or activate the CMP are:

- New request of a terminal with priority for the handover requests;
- Cancellation of a request or detection of a terminal inactivity;
- Change of rate for a connection AP-terminal detected by the Sniffer;
- Diminishing of the capacity of the bandwidth after observation by the sniffer.

Having discussed protocol and process design in a top down sequence, we move on to discuss data design and semantics in the subsequent chapter.
6 Data Design and Model Semantics

The approach followed in this thesis is to identify the aspects within the problem definition which are not well addressed in the state of the art and to work on designing and implementing them, for proof of concept purposes. Due to the fact that semantics, reasoning, and aggregate information are key elements for autonomic behavior, very good data design is required as a key step to enable this.

Figure 36: High-level conceptual model

Figure 36 provides a high level architectural overview of a node in a small-scale solution for autonomic resource management. It includes a set of showing the semantic model, the components for adaptive QoS resource management, and the self-configuring and self-aware modules in which the control logic is embedded.

For the semantic model part, the resource management functionality for media streaming is decomposed into smaller processes and that are included in an ontology. Having the representations for our network elements and processes would make our
ontology tree complete. The atomic elements of the tree are placeholders for pieces of information such as network parameters which change over time. Those information pieces form the basis for a self-aware system.

The self-configuring resource management mechanism for media streaming has been developed using control logic. This control logic gathers information from the system and also performs local operations that lead to predictive, situation-aware decisions on how to configure and re-configure real-time sessions in a self-* manner. This means that the mechanism for resource management and session configuration has a high level of awareness and be able to autonomously take its decisions in order to allocate the right amount of resources to flows, perform media tuning, and do admission control.

Analyzing Figure 36 bottom up, we see a conceptual architectural model of a node comprising in the following order: network elements, processes, and services. There are three levels of conceptual entities in terms of compositional hierarchy as well as complexity. The first level contains the network elements, node types such as a terminal (mobile device), access network element such as an Access Point (AP) or Base Station (BS). The second level corresponds to basic processes related to mobility management and QoS. Those processes are the building blocks of resource management examples include media transcoding, buffering, and network selection based on resource availability. Finally, the third level is comprised of services in the area of media streaming and interactive real-time communications such as Voice over IP (VoIP), video conferencing or telephony, and video on demand (VoD). The different conceptual levels in this section are covered one by one and the atomic parameters which are incorporated into the semantic model used to make networks self-aware and resource management processes self-configuring are listed.

Level 1: Network Elements

Mobile Device/Terminal: This is the user equipment or client side of a network corresponding to the front end or fore-most tier. Information can be grouped into profiles on the client tier, based on logical category. For instance:

- User Profile: This group contains information mostly about dynamic user preferences related to media streaming such as media settings in connection with desired thresholds for quality, customized presentation issues for video and voice traffic, and call control options related to media session setup. Furthermore, this profile contains information entered in the subscriber contract of the user. It includes information such as connection quality and default rates on the uplink and downlink for various traffic classes;
- Device Profile: This group includes all properties of a terminal that relate to the hardware and software properties of a device. Properties such as supported media types, codecs, display size, buffers, computational capabilities, and available media stacks belong to this category;
• Connection Profile: This is a more dynamic category pertaining to the set of flows, especially media flows in the scope of this thesis that are active at a particular time instance. A connection is modeled as a budget of packet size and packet rate pairs. The product of each pair yields the rate of a particular flow. This better reflects the resource consumption especially in terms of airtime. Furthermore, it is the duty of a network node to manage its flows and distribute resources available in a common budget to different media flows fairly and in compliance with the network settings and optimization goal. This issue is dealt with in details in Chapter 4 and Chapter 5.

Access Network Element: This is the closest tier to the client. An Access Network Element (ANE) and a user terminal jointly form the “last-mile” of a larger scale multi-tier operator network. This element also depends on the underlying network technology since many of those have a standardized architecture as specified either by the IEEE or 3GPP and 3GPP2. In UMTS for instance, the access network element is the Base Station (BS), whereas in IEEE 802.11, the ANE is the Access Point (AP) which can be seen as a micro base station but still differs vastly from its 3G counterpart. The ANE includes the following information elements that are of interest to resource management.

• Technology specific channel properties: Depending on whether the network is based on e.g. UMTS, 802.11, or WiMax, a different attribute set characterizing the physical channel as well as the channel access mechanism is needed. This is related to:
  • Hardware properties of the ANE;
  • MAC protocol used;
  • Functional and architectural primitives specified in the standard of each technology.
QoS and resource management parameters related to this category usually relate to the lower layers.

• Session, flow, and capacity management modules: The ANE should have a mechanism for managing not only low-level channel access issues at the MAC layer but also higher layer media flow management. The main protocol for media streaming and real-time traffic is the IETF standardized Real-time Protocol (RTP) with its acquaintances RTCP (Real-time Control Protocol) and RTSP (Real-time Streaming Protocol).

Other network element types closer to the core network and further from the front-end also have their own semantic information useful for resource management but they are out of the scope of this thesis.

Level 2: Processes

Our semantic model includes structures to represent at the first level elements, at the
second level processes (which are designated as dotted lines in Figure 37, and at the third level services in the area of multimedia streaming and real-time communication.

The bandwidth (resource) allocation models known today have be modified to efficiently provide QoS to distributed interactive multimedia applications. When performing QoS renegotiation, global architectural support with flexible connection management and a suitable binding architecture which allows tailoring communication services to applications requirements are needed.

The use of a fixed Usage Parameter Control (UPC) [SYC1] set does not allow QoS support and high channel utilization since Variable Bit Rate (VBR) multimedia traffic varies significantly over different time-scales. This becomes an even more critical issue in the mobile wireless multimedia scenario where it is difficult to commit fixed resources to VBR traffic for the duration of a connection while maintaining high network utilization.

On top of the network elements discussed above, basic network processes and logical modules reside. Those have a higher position in the semantic hierarchy and carry functional features. Due to the fact that this thesis focuses on the two autonomic features self-awareness and self-configuration, we design a functional module for each and then decompose it further into finer functionality.

Figure 37: Component view of self-* modules with functional decomposition

The functional breakdown for the two main modules is as follows.

- For a network node: self-awareness a module having the following building blocks:
  - Self-monitoring process module: monitors information on: current flows, available resources, and load;
Resource evaluation module: collects monitored parameters and generates further data which is then jointly evaluated to generate output information;
State recognition/identification: maps the outcome of the resource evaluation module to a discrete multi-dimensional state space.

- For a resource management process: self-configuration module having components:
  - Information collection: from different protocols such as MIP, CARD, and SIP;
  - Information inference and processing: Linking of information from various protocols, filtering it, and performing reasoning;
  - Resource configuration: trans-coding, buffering, admission control, distribution.

The capacity maintenance process on access point controllers is a complex self-configuring mechanism requiring a large set of parameters in order to operate properly. Those can be packed into a semantic information profile.

**Level 3: Services**

Services correspond to end-user modules such as voice over IP, streamed media such as video on demand, or video telephony and conferencing. They form the uppermost level in the semantic hierarchy and logically contain the process modules.
7 Conclusion and Future Work

7.1 Conclusion

In thesis, we considered network resource management backed by the autonomic communication principles: self-awareness and self-configuration. We applied those principles to the open access network architecture and devised two mechanisms for self-configuring resource management, one on the client side and one on the 802.11 access point. We understood that as the complexity of networks grows, resources become bottlenecks so more automation in the mechanisms that manage those resources is required. We suggested a way to deal with that which brings on some quantitative as well as qualitative improvement. We conclude that applying autonomic principles to resource management as well as other aspects like mobility can solve some of the challenges faced by networking today. However, we also conclude that only limited improvements can be brought on at a cost and facing tradeoffs, therefore, the vision of full autonomy and optimal network behavior remains out of near sight.

This research work investigated the issue of making network resource management mechanisms in the access part of wireless networks self-aware and self-configuring. To achieve our aim, we broke down this afore-mentioned high level goal into smaller ones. We determined the limitations and shortcomings of current brokerage mechanisms, parameter negotiation procedures, and general QoS handling in broadband wireless cellular networks (such as 802.11) both qualitatively and quantitatively. We came up with a way of making our open access networks self-aware and devise a self-configuring mechanism which has a competitive advantage in the qualitative sense over already existing mechanisms.

The licentiate thesis, that comprises the first part of the PhD research, dealt with the problem of autonomic resource management in IEEE 802.11 open access networks. In particular, it focused on making networks self-aware and resource management processes self-configuring (with minimal human interaction).

7.2 Future Work: Map of the Road Ahead

Earlier in this thesis, Chapter 3 discussed semantics and data design and Chapters 4 and 5 discussed network resource management processes and protocols respectively that act in an autonomic manner. This chapter looks at two key aspects of autonomic
communication: self-awareness and self-configuration just as the thesis as a whole does, but from a more holistic and systematic point of view. The roadmap for subsequent research is the subject of this chapter. It includes a big-picture overview of the different research aspects within the autonomic communication area, a discussion on the reconfigurability challenges faced when wiring several protocols, an overview of meta-protocols, a methodology for guided cross-layering, and a multiple stakeholder perspective for autonomic communication services.

This section looks globally at autonomic networking and introduces the different sequential steps and aspects necessary for achieving self-aware node architecture and self-configuring resource management.

The subsequent part of the research shall deeper analyze the autonomic resource management topic. The research shall follow a systematic and scientific approach towards achieving self-aware node architectures and self-configuring resource management. Individual steps in this research shall include: data, state space, and semantic design, active monitoring, reasoning (individual one-step and join reasoning), high-level meta-protocol specification, and protocol formation.

Part One: Self-Aware Node Architecture

Self-aware node architectures comprise several aspects. The definitions of those are as follows:

- **Data design**: includes mainly attributes pertaining to network elements and protocol atomic data; semantics, and ontologies that carry meaningful information about the network.
  - **State-space design**: the purpose of state space design is to mathematically model and capture the behavior of resource management processes in open access networks. In computer science, a state space is a description of a configuration of discrete states used as a simple model of machines.
  - Utility functions to represent resources;
  - State machines to represent outputs, inputs, and processes;
  - Cost functions to represent stakes of stakeholders.

- **Active monitoring**: regular acquisition of parameters modeled in the step before from the environment. Active monitoring means that the system proactively retrieves parameters in real-time and evaluates what is retrieved in order to potentially take necessary action.

- **1-step reasoning**: when not enough parameters or “knowledge pieces” are available, then available information is processed to obtain new parameters (using suitable utility functions). As an example, when combining monitored information

- **Joint reasoning**: collect output state information of single nodes, and join them, using meta-state and meta-utility functions, to obtain global network state information. For example a congested state of a single network node can or
cannot mean that the whole network containing this node is congested depending on the topology, architecture, and relationships of the nodes to the service or process (e.g. resource management) in question.

Part Two: Self-Configuring Resource Management involves several stages:

- **High-level specification**: when using several protocols as is also done in this thesis, dependencies become more complex. High-level specification as pseudo-code, high-level description instruments like functional diagrams, message sequence charts, and other available primitives are necessary. This specification captures the behavior. Behavior modeling is a very significant aspect of complex systems, and it is crucial for autonomic networking for understanding system dynamics and core logic.
- **Protocol processing**: transform protocol descriptions into intermediate formats where invariants are visible, as well as variant parts, and the notation is suitable for processing;
- **Protocol wiring**: this process takes place systematically guided by ontologies that classify protocols and allow linking key invariant states of different protocols;
- **Evaluation**:
  - **Conformance run**: a verification toolbox responsible for standard conformance and consistency testing of a designed meta-protocol;
  - **Performance run**: is a test of a meta-protocol where behavior is assessed quantitatively based on certain metrics and criteria.

Autonomic Communication is all about self-* aspects and their realization. So far, however, those aspects have been treated independently. A two-phase approach in this thesis for realizing autonomic behavior is proposed. The first phase is to make network nodes individually as well as collectively self-aware. The second phase deals with the dynamic behavior. The aim is to make resource management processes reconfigurable/self-configuring. For the self-awareness aspect, a state-space model where nodes are capable of situated reasoning that is either predictive or definitive is required.

Individual known network protocols are targeted to a specific layer or set of layers and handle very specific functionality. On the other hand, a communication service often requires several functional (e.g. connectivity) and non-functional (e.g. QoS) features to be there in order to operate properly. Thus, a service (logical or functional) flow involves several protocols making the design choices numerous but also posing a challenge. A **meta-protocol** within subsequent research work refers to a protocol description that captures the flow logic of a particular service and clearly indicates how the wiring of individual (standardized and available) protocol modules forms reconfigurable behavior. A systematic methodology is required for such work. ICSP, presented in this thesis, is a pre-mature version of a meta-protocol.
References


[ME2] Analysis and measurement of session setup delay and jitter in VoWLAN using composite metrics, Muslim Elkotob and Karl Andersson; International ACM Conference on Mobile and Ubiquitous Multimedia - In cooperation with ACM SIGMOBILE; December 3-5, Umeå, Sweden.


[SDP] IETF RFC 3264: An Offer/Answer Model with Session Description Protocol (SDP); J. Rosenberg and H. Schulzrinne, 2002


[RFC3558] IETF RFC 3558 - RTP Payload Format for Enhanced Variable Rate Codecs (EVRC) and Selectable Mode Vocoder (SMV)


[RFC3551] IETF RFC 3551: RTP Profile for Audio and Video Conferences with Minimal Control; H. Schulzrinne and S. Casner, July 2003

[WIT1] Willamette Integrated Technology Services, on resources and bandwidth usage: www.willamette.edu/wits/resources/netiquette/bandwidth.htm


[MS1] Autonomic Communication whitepaper: https://www.fokus.fraunhofer.de/de/motion/_docs/aclab_200411_Autonomic-Communication.pdf; Michael Smirnov


[PM1] McGovern, P.; Murphy, S.; Murphy, L., “Addressing the Link Adaptation Problem for VoWLAN using Codec Adaptation”, IEEE GLOBECOM '06

[JS1] An Optimized Handoff Scheme for IP Mobility Support in IEEE 802.11 802.11S: Joanna Samprakou, Christos J. Bouras, Theodore Karoubalis
[CA1] Running Variance Metric for evaluating performance of Wireless IP Networks in the MobileCity Test-bed, Christer Åhlund, Robert Brännström, and Arkady Zaslavsky


[LS] LinSpot website: www.linspot.com


Glossary of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>Third Generation Partnership Project</td>
</tr>
<tr>
<td>802.11</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>AC</td>
<td>Autonomic Communication</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>ADMD</td>
<td>Agent Discovery and Motion Detection</td>
</tr>
<tr>
<td>ADV</td>
<td>Agent Advertisement</td>
</tr>
<tr>
<td>AN</td>
<td>Autonomic Networking</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CARD</td>
<td>Candidate Access Router Discovery</td>
</tr>
<tr>
<td>CMP</td>
<td>Capacity Management Process</td>
</tr>
<tr>
<td>CPIM</td>
<td>Common Presence and Instant Messaging</td>
</tr>
<tr>
<td>CPN</td>
<td>Cognitive Packet Network</td>
</tr>
<tr>
<td>CSMA</td>
<td>Carrier Sense Multiple Access</td>
</tr>
<tr>
<td>ECS</td>
<td>Eager Cell Switching</td>
</tr>
<tr>
<td>FA</td>
<td>Foreign Agent</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messaging</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Subsystem</td>
</tr>
<tr>
<td>INAP</td>
<td>Intelligent Network Application Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Science, Medical</td>
</tr>
<tr>
<td>ISUP</td>
<td>ISDN User Part</td>
</tr>
<tr>
<td>JAIN</td>
<td>Java API for Integrated Networks</td>
</tr>
<tr>
<td>JCC</td>
<td>Java Call Control</td>
</tr>
<tr>
<td>JCP</td>
<td>Java Community Process</td>
</tr>
<tr>
<td>JMF</td>
<td>Java Media Framework</td>
</tr>
<tr>
<td>JSR</td>
<td>Java Specification Requests</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>LRE</td>
<td>Limited Relative Error</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>MIP</td>
<td>Mobile IP</td>
</tr>
<tr>
<td>MN</td>
<td>Mobile Node</td>
</tr>
<tr>
<td>MSRP</td>
<td>Message Session Relay Protocol</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transfer Unit</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OAN</td>
<td>Open Access Network</td>
</tr>
<tr>
<td>PA</td>
<td>Presence Agent</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>PT</td>
<td>Payload Type</td>
</tr>
<tr>
<td>PUA</td>
<td>Presence User Agent</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Remote Authentication Dial In User Service</td>
</tr>
<tr>
<td>RU</td>
<td>Residential User</td>
</tr>
<tr>
<td>RFC</td>
<td>Requests for Comments</td>
</tr>
<tr>
<td>RLS</td>
<td>Resource List Server</td>
</tr>
<tr>
<td>RTCP</td>
<td>Real-Time Control Protocol</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Protocol</td>
</tr>
<tr>
<td>RTS/CTS</td>
<td>Request to Send/Clear to Send</td>
</tr>
<tr>
<td>RTSP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>SDP</td>
<td>Session Description Protocol</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>SIP for Instant Messaging and Presence Leveraging Extensions</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>S/MIME</td>
<td>Secure/Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UA</td>
<td>User Agent</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifiers</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual LAN</td>
</tr>
<tr>
<td>VU</td>
<td>Visiting User</td>
</tr>
</tbody>
</table>