Integrating Context-Awareness in the IP Multimedia Subsystem for Enhanced Session Control and Service Provisioning Capabilities

May El Barachi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

November 2009

Your file Votre référence
ISBN: 978-0-494-67332-4
Our file Notre référence
ISBN: 978-0-494-67332-4

ABSTRACT

Integrating Context-Awareness in the IP Multimedia Subsystem for Enhanced Session Control and Service Provisioning Capabilities

May El Barachi, Ph.D.
Concordia University, 2009

The 3GPP-defined IP Multimedia Subsystem (IMS) is becoming the de-facto standard for IP-based multimedia communication services. It consists of an overlay control and service layer that is deployed on top of IP-based mobile and fixed networks. This layer encompasses a set of common functions (e.g. session control functions allowing the initiation/modification/termination of sessions) and service logics that are needed for the seamless provisioning of IP multimedia services to users, via different access technologies. As it continues to evolve, the IMS still faces several challenges including: the enabling of innovative and personalized services that would appeal to users and increase network operators' revenues; its interaction with other types of networks (e.g. wireless sensor networks) as means to enhance its capabilities; and the support of advanced QoS schemes that would manage the network resources in an efficient and adaptive manner.

The context-awareness concept, which comes from the pervasive computing field, signifies the ability to use situational information (or context) in support to operations and decision making and for the provision of relevant services to the user. Context-awareness is considered to enhance users' experience and is seen as an enabler to adaptability and service personalization - two capabilities that could play important roles in telecommunication environments.

This thesis focuses on the introduction of the context-awareness technology in the IMS, as means to enhance its session control and service provisioning capabilities. It starts by presenting the necessary background information, followed by a derivation of requirements and a review of the related work. To ensure the availability of contextual information within the network, we then propose an architecture for context information acquisition and management in the IMS. This architecture leverages and extends the 3GPP presence framework. Building on the capabilities of this architecture, we demonstrate how the managed information could be integrated in IMS operations, at the control and service levels. Showcasing control level integration, we propose a novel context-aware call differentiation framework as means to offer enhanced QoS support (for sessions/calls) in IMS-based networks. This framework enables the differentiation between different categories of calls at the IMS session control level, via dynamic and adaptive resource allocation, in addition to supporting a specialized charging model. Furthermore, we also propose a framework for enhanced IMS emergency communication services. This framework addresses the limitations of existing IP-based emergency solutions, by offering three main improvements: a QoS-enhanced emergency service; a context-aware personalized emergency service; and a conferencing-enhanced emergency service. We demonstrate the use of context awareness at the IMS service level using two new context-aware IMS applications. Finally, to validate our solutions and evaluate their performance, we build various proof-of-concept prototypes and OPNET simulation models.

ACKNOWLEDGEMENTS

A Ph.D. degree is a personal journey, but one that cannot be completed without a great deal of help and support. On an academic level, I would like to thank my supervisors Dr. Rachida Dssouli and Dr. Roch Glitho for their astute and constructive criticisms that have helped me shape my thesis into its final form, and for their continuous support and encouragement. I would also like to thank my supervisory committee members, Dr. Agarwal, Dr. Debbabi, and Dr. Rilling, for their advices and comments at the different stages of this research.

On a professional level, I would like to thank all my colleagues at Concordia and the TSE research lab, for their moral support and their help and useful suggestions. It has been a great pleasure working with all of you. Moreover, I would like to acknowledge the financial support provided to me by Concordia University, Ericsson Research Canada, and the Natural Sciences and Engineering Research Council of Canada.

On a personal note, I would like to express my gratitude to my parents and my brother who always encouraged me to push my limits and have confidence in myself and my work, and my husband Amir for his patience, understanding, and support throughout this journey. To my kids Fares and Yasmine, you are the light of my life and the motivation that makes me want to go further.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

3G	Third Generation networks
3GPP	Third Generation Partnership Project
AAA	Authentication, Authorization, and Accounting
ABMF	Account Balance Management Function
ACA	ACcounting Answer
ACR	ACcounting Request
AF	Application Function
ALI	Automatic Location Identification
ANI	Automatic Number Identification
AOA	Angle of Arrival
A-RACF	Access – Resource and Admission Control Function
AVP	Attribute-Value Pair
BAA	Bandwidth Adaptation Algorithm
BGCF	Breakout Gateway Control Function
BGF	Border Gateway Function
bpm	Beats per minute
bps	Bits per second
CAC	Connection Admission Control
CACM	Call Admission/Control Module
CAMEL-AS	Customized Application Mobile-Enhanced Logic Application Server
CBP	Call Blocking Probability
CC	Call Category
CCA	Credit Control Answer
CCR	Credit Control Request
CDF	Charging Data Function
CDR	Charging Data Record

CE-ID	Contextual Entity ID
CGF	Charging Gateway Function
CI	Call ID
CIB	Context Information Base
CIPID	Contact Information in Presence Information Data format
COPS	Common Open Policy Service protocol
CRF	Context Retrieval Function
CRM	Context Retrieval Module
CSCF	Call/Session Control Function
CTF	Charging Trigger Function
DD	Displacement Direction
E911	Enhanced 911
EBCF	Event-Based Charging Function
E-CSCF	Emergency-CSCF
ECUR	Event Charging with Unit Reservation
ESQK	Emergency Services Query Key
ESRD	Emergency Services Routing Digits
ESRP	Emergency Service Routing Proxy
ET	Event Type
ETSI	European Telecommunications Standards Institute
FCTP	Forced Call Termination Probability
FSM	Finite State Machine
GA	Guaranteed Audio
GSU	Granted Service Unit
GUP	Generic User Profile
GV	Guaranteed Video
HCDP	Handoff Call Dropping Probability
HSC	Highest Service Class

HSS	Home Subscriber Service
IAM	Information Access Module
IAP	Information Access Policies
I-BCF	Interconnection Border Control Function
ICID	IMS Charging Identifier
ICP	IMS Client Platform
I-CSCF	Interrogating-CSCF
IDE	Integrated Development Environment
IEC	Immediate Event Charging
IETF	Internet Engineering Task Force
IFC	Initial Filter Criteria
IMS	IP Multimedia Subsystem
IMS-GW	IMS GateWay
IMS-GWF	IMS-GateWay Function
ISIM	IP multimedia Services Identity Module
LDAP	Lightweight Directory Access Protocol
LRF	Location Retrieval Function
LS-GW/DCS	Logical Sensors Gateway/Dummy Context Source
MANET	Mobile Ad hoc NETwork
MCN	Multihop Cellular Network
MEGACO	MEdia GAteway COntrol protocol
MGCF	Media Gateway Control Function
MGW	Media GateWay
MLPP	Multi-Level Precedence and Preemption
MMD	Multimedia Domain
mm HG	Millimetres of Mercury
MPC	Mobile Positioning Center
MRFC	Media Resource Function Controller

MRFP	Media Resource Function Processor
MSAG	Master Street Address Guide
MT	Media Types
NCBP	New Call Blocking Probability
NDP	Neighbor Discovery Protocol
NEMO	NEtwork MObility
NGN	Next Generation Network
NS/EP	National Security/Emergency Preparedness
OCF	Online Charging Function
OCS	Online Charging System
OMA	Open Mobile Alliance
OSA-AS	Open Services Architecture Application Server
P-ANI	Pseudo Automatic Number Identifier
PCC	Policy and Charging Control architecture
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function
P-CSCF	Proxy-CSCF
PDE	Position Determination Equipment
PDF	Policy Decision Function
PDP	Policy Decision Point
PEA	Presence External Agent
PEP	Policy Enforcement Point
PIDF	Presence Information Data Format
PLS	Presence List Server
PNA	Presence Network Agent
pps	Packets per second
PS	Presence Server
PSAP	Public Safety Answering Point

PUA	Presence User Agent
P-UAC	Presence User Agent Client
P-UAS	Presence User Agent Server
RACS	Resource and Admission Control Subsystem
RAF	Repository Access Function
RCEF	Resource Control Enforcement Function
RD	Relative Distance
RF	Rating Function
RID	Room ID
RMP	Resource Management Policies
RP	Resource Priority
RPID	Rich Presence Information Data format
RTP	Real Time Transport Protocol
RTCP	RTP Control Protocol
SAA	Stateless Address Auto-configuration
SBCF	Session-Based Charging Function
S-CSCF	Serving-CSCF
SCUR	Session Charging with Unit Reservation
SDG	Service Discovery Gateway
SDS	Service Development Studio
SGW	Signaling GateWay
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SIP-AS	SIP Application Server
SLF	Subscriber Location Function
SLP	Service Location Protocol
SPDF	Service-based Policy Decision Function

SPF	Session Prioritization Function
SPI	Service Parameter Info
SPR	Subscription Profile Repository
SRF	Single Reservation Flow
TDOA	Time Difference of Arrival
TrGW	Transition GateWay
UDDI	Universal Description, Discovery, and Integration
UA	User Agent
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UpnP	Universal plug and Play
WSN	Wireless Sensor Network
XCAP	XML Configuration Access Protocol
XDMS	XML Document Management Server
XMPP	Extensible Messaging and Presence Protocol

Chapter 1

Introduction

This chapter starts by introducing the research area and discussing our motivations for this work, before presenting the problem that will be tackled by this thesis and detailing the thesis objectives. This is followed by a summary of the thesis contributions and a presentation of the thesis structure.

1.1 Research Area

Third Generation (3G) networks combine the strengths of two of the most successful communication technologies: cellular and internet technologies. These networks aim at offering their users ubiquitous access to a multitude of feature-rich, IP-based, multimedia services. The IP Multimedia Subsystem (IMS) is the key component in the 3G architecture, which enables the realization of this vision [1]. Initially specified by the Third Generation Partnership Project (3GPP) [2] for 3G mobile networks, and now embraced by other standard bodies such as the European Telecommunications Standards Institute (ETSI) [3] for fixed networks and PacketCable [4] for cable networks, the IMS consists of a horizontal control and service layer that is deployed on top of IP-based mobile and fixed networks. This control/service layer encompasses a set of common functions and service logics that are needed for the seamless provision of IP multimedia services to users, via different access technologies (i.e. mobile, fixed, and wireless access technologies).

As an important reference service delivery platform for next generation networks, the IMS is expected to offer several benefits to network operators, service providers, and end users. The first benefit is the ability to support the delivery of services combining telecom and datacom flavors, in an access agnostic fashion, over a common IP-based core network.

From the end-user perspective, this implies the availability of a variety of services, such as voice over IP, multiplayer gaming, content and media sharing, presence-based applications, and audio/video conferencing, from any location and over any access network. From network operators' perspective, the reliance on a single network capable of delivering all types of applications, instead of application specific networks, would result in lower capital and operational expenditures. Another important benefit is the support for faster development and deployment of new services by service providers, by offering a set of common functions (acting as service enablers) that can be accessed via standard interfaces and leveraged for the development of applications running on top of the IMS. Examples of the common functions provided by the IMS include: session and service control, subscription and mobility management, access control, and charging. As another merit differentiating it from the best effort Internet technology, the IMS supports basic QoS mechanisms as means to offer 3G users with a predicted experience and a controlled quality, when using IP multimedia services. As a fourth benefit, the IMS offers a flexible charging framework enabling network operators to define different charging models that suit the nature of the various services they offer.

Despite its relative maturity and its deployment in a number of trials [5] and commercial products [6, 7], the IMS continues to evolve and several issues/challenges, related to its operation and architecture, are being investigated. One of these challenges is the enabling of IMS "killer" applications. It is believed that for the IMS to gain wide acceptance, it must go beyond simply providing a fast service creation environment and must foster the creation of innovative and personalized services that would appeal to users and increase network operators' revenues. A potential category of killer applications consists of

intelligent applications that offer personalized services by adapting their behavior according to the users' needs and changing situation. Another open research issue is the interaction of the IMS with other types of networks that would contribute with their additional capabilities (e.g. wireless sensor networks contributing with their data sensing capabilities) to achieve enhanced service provisioning in the IMS. A related challenge, which depends on the availability of sensory data, is the support of advanced QoS schemes that would take into consideration users' needs/preferences and the network situation to manage network resources in an efficient and adaptive manner.

The concepts of context and context-awareness [8] have emerged in the field of pervasive computing [9] that aims at achieving rich, intuitive, and natural interactions between humans and computing devices surrounding them. While context implies situational information related to different entities (e.g. a user's location, a device's battery level, or a network's available capacity), context-awareness signifies the ability to use this contextual information in support to operations and decision making and for the provision of relevant services to the user. The context-awareness lifecycle is a process that consists of three phases, namely: the context acquisition phase during which information is collected from various sources (e.g. physical sensors such as wireless sensor networks or logical sensors such as network probes); the context modeling and processing phase in which information is refined and represented in a proper format that makes it easy to understand; and the context dissemination and usage phase in which the information is either distributed to interested entities or used by the system in support to its own operations and services.

As a main value preposition, context-awareness is considered to enhance users' experience and is seen as an enabler to adaptability and service personalization. These two capabilities

could play important roles in telecommunication environments as follows: On one hand, communication systems could adapt to variations in their execution environment to offer a good QoS to its users. On the other hand, systems could exploit information about the users' context to provide them with personalized services that suit their changing situation. This thesis focuses on the introduction of the context-awareness technology in the IMS, as means to enhance its control operations and service provisioning capabilities.

1.2 Motivations and Work Scope

Due to the availability of low cost sensing technologies, a wealth of contextual information can be collected about the situation of the user and the network, in a telecommunication environment. Integrating this rich set of contextual information in the IMS operation can greatly enhance its functionality and service provisioning capabilities. This integration can be envisioned at two different levels in the IMS, namely: the control level mainly offering session management (i.e. session initiation, modification, and termination) capabilities to applications; and the service level encompassing the logics needed for the provision of value-added services.

On one hand, context-awareness can be introduced at the IMS control level as means to enable the system to proactively manage sessions, in adaptation to the user and the network situation. This proactive session management capability could be translated into an enhanced operation and sophisticated capabilities by the system. Examples of such capabilities include the support of: advanced QoS schemes such as call differentiation schemes enabling users to express the level of priority of their sessions and obtain the needed guarantees via dynamic and adaptive resource allocation by the network; enhanced emergency communications offering preferential treatment, personalized services, and rich

communication models to emergency callers; and context-sensitive charging taking into consideration the context in which the service is rendered (e.g. the events occurring during the session and its level of priority) to enable the effective charging of services while being appealing to users.

On the other hand, context could be considered as an application building block that can be leveraged at the service level for the development of a wide range of novel value-added services. Examples of potential services include: wireless healthcare applications monitoring and interpreting patients' physiological data and offering them with personalized medical assistance under problematic health conditions; pervasive games involving the interaction with physical/virtual objects and characters, and using the game context to adapt the players experience; and lifestyle assistance applications making use of users' situational information to assist them in their daily activities (e.g. training and shopping).

While control level integration focuses on influencing the network's session control behavior based on context, service level integration deals with the adaptation of applications' behavior according to situational information. In this work, both levels of integration are considered, with a focus on control level integration that has the potential of improving the IMS core functionality and solving some of the issues related to its operation. More specifically, two main directions are followed in relation to use of context-awareness for the improvement of the IMS control capabilities, namely: advanced QoS support via context-aware call differentiation; and enhanced emergency services support. Those directions, which touch two of the most important aspects of the IMS operation (i.e. QoS and emergency communications support), were chosen due to the important benefits

they can bring and the challenges they present. As for the introduction of context-awareness at the service level, it is demonstrated via novel applications that could be enabled in a context-aware IMS environment.

1.3 Problem Statement and Thesis Objectives

In this section, we discuss the challenges associated with the introduction of the context-awareness technology in the IMS. We also detail the thesis objectives.

1.3.1 Problem statement

The introduction of the context-awareness technology in the IMS operation entails two main categories of issues. The first category is related to the acquisition and management of contextual information in the IMS. In fact, contextual information must be collected from various sources and effectively managed in the IMS, to ensure its availability for future usage in the network. A first issue related to this problem is how the IMS can interact with different information sources, such as Wireless Sensor Networks (WSNs), to collect the needed information. This raises the need to define a suitable architecture for the integration of WSNs in the IMS. Several issues related to the operation of this architecture should also be addressed, such as: the definition of a business model taking into consideration the WSN operator as a new business player in the 3G environment; dealing with the existence of user and non-user contextual entities (whose information is captured by WSNs) in the IMS in terms of identification and charging; in addition to security, trust, and service discovery issues related to the exchange of potentially sensitive information (e.g. location) between various types of WSNs (with different capabilities) and the IMS. Other technical issues related to information management and dissemination in the IMS consist of: the definition of an appropriate information model facilitating knowledge

representation and sharing; and the definition of suitable information exchange models/protocols enabling information access/dissemination in a flexible and resource efficient manner.

The second category of issues is related to the integration of contextual information in IMS operations. Depending on the nature of the context-aware operations envisioned in the network, different types of challenges can be faced. Since our focus is on advanced QoS support via context-aware call differentiation and on enhanced emergency services support, in addition to the enabling of novel value-added services, challenges related to these areas arise. Issues related to context-aware call differentiation include the following: how to define a suitable call differentiation scheme (i.e. what are the potential classes of calls and their distinguishing factors); how to label sessions and offer flexible QoS negotiation mechanisms to the user; which resource management techniques/policies can be used to achieve dynamic resource allocation in an adaptive and resource efficient manner and which pieces of contextual information are required for their operation; what is the impact of the call differentiation scheme on session management scenarios; and which charging model can be used for the effective charging/billing of the resulting differentiated services.

In terms of enhanced emergency services support, the following questions arise: which pieces of contextual information can be exploited to achieve enhanced emergency services and more efficient emergency operations; which mechanisms can be used to provide preferential treatment to emergency calls and prioritize their access to resources in an efficient and adaptive manner; how to offer personalized emergency services taking into consideration the user situation; and how to utilize richer communication models to improve the efficiency of emergency operations.

As for the use of context as enabler for novel value-added services, it entails the following issues: which application areas could benefit from the availability of contextual information within the network; which pieces of information are needed for their operation; how can applications interact with the context management entities in the network to obtain the needed information; and how to utilize this information to influence the applications' behavior.

1.3.2 Thesis Objectives

As discussed above, this thesis aims at examining the issues and implications associated with the introduction of the context-awareness technology in the IMS. More specifically, the key objectives of this thesis can be summarized as follows:

- Propose an architectural framework enabling the acquisition and management of contextual information in the IMS.
- Suggest mechanisms for using context-awareness to improve the IMS control capabilities. This objective can be refined in two sub-objectives, namely:
    - Proposing mechanisms enabling the use of context-awareness for advanced QoS support in the IMS.
    - Proposing mechanisms enabling the use of context-awareness for the support of enhanced emergency communications in the IMS.
- Present applications demonstrating the use of context at the IMS service level, as a building block for context-aware value-added services.
- Report on experiences related to the implementation and deployment of the solutions proposed and evaluate their performance.

1.4 Thesis Contributions

The main contributions of this thesis, along with the related publications, are summarized as follows:

- **Critical Review of related work:** We have derived requirements related to the problems of information acquisition/management, service differentiation, and emergency service support in IP-based networks, and have evaluated the related work in light of those requirements.
- **A presence-based architecture for context information acquisition and management in the IMS [10, 11, 12, 13, 14]:** To ensure the availability of contextual information within the network, we proposed an architecture for context information acquisition and management in the IMS. This architecture is based on the extension of the 3GPP presence framework, which enables the management and dissemination of users' presence information (a subset of contextual information) in the network. The relation that exists between the concepts of context and presence (context being a generalization of presence) and the extensibility of the presence framework motivated us to use this framework as the basis for our solution. The solution consists of the following elements: An extended presence architecture enabling the IMS interaction with WSNs and network probes for the collection of the needed information - this architecture relies on sensor gateways to enable the interworking between the IMS and various types of physical/logical sensors. Related to the operation of this architecture, an extended presence information model is devised to enable the representation of contextual information (related to different types of entities), and three information exchange models are proposed to achieve flexible and resource efficient information exchange within the network. Furthermore, to enable the practical deployment of the

proposed architecture, a business model taking into consideration the roles of WSN operators and contextual entities is proposed for this sensors-enabled IMS environment, and a two-level identification/charging scheme relying on the novel concept of IMS corporate identities (and their associated dependant identities) is elaborated. Moreover, security, information access control, and service discovery issues are addressed.

- **An IMS call differentiation framework [15, 16, 17]:** We proposed a novel context-aware call differentiation framework, as means to offer enhanced QoS support in IMS-based networks. This framework enables the differentiation between different categories of calls (offering different priorities/guarantees) at the IMS control level, via dynamic and adaptive resource allocation. Several elements constitute this framework. The first element is a call differentiation scheme, enabling the definition of various categories of calls with different QoS profiles – three categories of calls are defined as examples of possible QoS profiles. To enable the support of such profiles, an extended IMS architecture is proposed. This architecture introduces a new resource management entity and two new interfaces to the standard IMS architecture, and brings enhancements to some of the existing IMS functional entities. Related to the operation of this architecture, two adaptive resource management mechanisms are proposed to achieve dynamic resource allocation to sessions. Furthermore, a charging model extending existing IMS online/offline charging mechanisms is proposed to address the charging aspects of multi-grade services. Compared to existing service differentiation solutions, this solution offers several benefits, such as: flexible QoS negotiation mechanisms; control over many communication aspects as means for differentiation; a dynamic and adaptive resource management strategy; and a specialized charging model.

- **A framework for enhanced IMS emergency communication services [11, 18]**: Emergency services represent one of the fundamental and most valued services provided by communication networks, and their support in IP-based networks has recently been investigated. However, the IP-based emergency solutions proposed so far have several limitations. Aiming at addressing those limitations, we proposed an enhanced IMS emergency solution for IP-based networks. This solution focuses on the improvement of three aspects of IMS emergency communications, namely: QoS and resource management; context-awareness and service personalization; and the use of richer emergency communication models. The enhancement of the QoS/resource management aspect of emergency sessions is achieved by generalizing our call differentiation solution (originally tackling the case of regular calls), and applying it to the emergency case. This implies the definition of new QoS profiles for emergency sessions, and the extension of the proposed IMS call differentiation architecture with additional interfaces and interactions with enhanced emergency related components. Unlike the existing IMS emergency service architecture, our QoS-enhanced architecture provides preferential treatment to all categories of emergency communications (including public-initiated emergency calls, mission critical calls, and urgent communications among citizens during major events), and prioritizes their access to resources in an efficient and adaptive manner. The second enhancement aims at exploiting a wide range of contextual information to improve the efficiency of emergency operations and offer personalized emergency services to users. To achieve this goal, a personalized emergency service scenario, demonstrating context-aware routing capabilities and presenting call takers with a wide view about callers' context,

is studied and an extension of the IMS emergency service architecture is proposed to enable its support. As for the third aspect addressed, it tackles the enhancement of the IMS emergency service architecture with multi-party session support capabilities, as means to offer richer forms of communication (e.g. conferencing, sub-conferencing, and automatic switching between sub-conferences) enabling better coordination of rescue efforts.

- **Demonstration of context-aware value added services support in the IMS [10, 12, 14]:** To demonstrate how context can be used as an application building block for the development of context-aware value-added services in the IMS, we investigated three potential application areas, namely: wireless healthcare; pervasive gaming; and lifestyle assistance. Furthermore, we developed two concrete applications (related to these areas) as examples. The first application is a mobile pervasive game called "Fruit Quest", while the second is a personalized call control application that enables the automatic establishment of a call between two colleagues, when they are in their respective offices. These applications demonstrate the application potential that can result from the combination of the IMS and the context-awareness technologies. Furthermore, they show that the use of context as service enabler abstracts applications developers from the details/complexity of context management related operations, thus facilitating the development of context-aware value added services.
- **Implementation and evaluation of the solutions proposed:** We used Ericsson's Service Development Studio (SDS) – an IMS emulated environment – and three implementation technologies (SIP/SIMPLE, COPS, and Diameter) to build proof of concept prototypes of the proposed solutions. SIP and existing SIP extensions were

used for session management and QoS negotiation interactions related to the call differentiation and the enhanced emergency solutions, while an optimized version of SIMPLE was used as a context exchange protocol for the information acquisition/management solution. As for COPS, it was used for the exchange of policy-based resource allocation decisions related to call differentiation, and was extended with a new policy client type and client related objects to achieve this role. Diameter was used for the exchange of charging related interactions and was extended to carry additional information related to multi-grade service charging. All the functional entities proposed in our solutions were built and introduced either as extension components to SDS or as enhancements in the logic of one of its existing components, and different test scenarios were carried to prove the feasibility of the solutions proposed. In terms of performance evaluation, the prototypes were used for the collection of performance measurements about a sub-set of the solutions, while simulations models were built to obtain detailed performance measurements about the call differentiation architecture and the QoS-enhanced IMS emergency service architecture.

1.5 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 gives an overview of the 3GPP IMS architecture and some of its essential functions (i.e. QoS and charging) and advanced services (i.e. presence and emergency service support). Furthermore, it introduces the concepts of context and context-awareness. Chapter 3 presents the identified requirements and critically reviews the related work. Chapter 4 is devoted to the information acquisition/management architecture. Chapters 5 and 6 discuss the usage of context-

awareness for advanced QoS support and enhanced emergency services provisioning in the IMS, by respectively presenting the IMS call differentiation framework and the enhanced IMS emergency framework proposed. Chapter 7 presents the prototypes and demonstrates the introduction of context-awareness at the IMS service level using two context-aware applications. Chapter 8 describes the simulation models and the detailed performance evaluation of the call differentiation and the QoS-enhanced emergency service architectures. Chapter 9 concludes the dissertation and discusses items for future work.

Chapter 2

Background Information

This chapter starts by giving an overview of the 3GPP IP Multimedia Subsystem, including: a description of its architecture and some of the important aspects related to its operation (i.e QoS and charging); and a presentation of some of its advanced services which represent areas of interest in this thesis. This is followed by background information on the concepts of context and context-awareness.

2.1 The 3GPP IP Multimedia Subsystem: Architecture, Essential Functions, and Advanced Services

The IP Multimedia Subsystem is an architectural framework created for the purpose of seamlessly delivering IP multimedia services to end users, with an acceptable QoS at an acceptable price [1]. This framework was initially defined by the 3GPP [2] standards organization as part of its standardization work on a third generation mobile system, called the Universal Mobile Telecommunications System (UMTS). In its original formulation presented in release 5 [19] of the 3GPP specifications (in 2002), the IMS was introduced as a new service layer enabling the support of SIP-based multimedia services in IP-based mobile networks. This vision was enhanced in releases 6 [20] and 7 [21] of the specifications to include additional features such as: presence and emergency service support; interworking with circuit-switched legacy networks and other IP networks; as well as the support for fixed broadband access. Moreover, other standard bodies have embraced the 3GPP-defined IMS as part of their next generation networks standardization activities. Examples include: 3GPP2 [22] that based their Multimedia Domain (MMD) architecture on the 3GPP-IMS, and the ETSI-TISPAN [23] that defined an IMS-based PSTN emulation system as part of its Next Generation Network (NGN) Architecture.

2.1.1 The 3GPP IMS Architecture

Several architectural requirements have led to the design of the 3GPP IMS, namely: the support for *IP multimedia sessions' establishment*; *QoS* support; *interworking* with other types of networks (e.g. CS networks and the Internet); *roaming* support; *service control* via the enforcement of general and user-specific policies; *rapid service creation* via the standardization of service capabilities (instead of services); and the support of *multiple access technologies* [24]. Figure 2.1 shows an overview of the 3GPP IMS architecture [21] that was designed to fulfill those requirements.

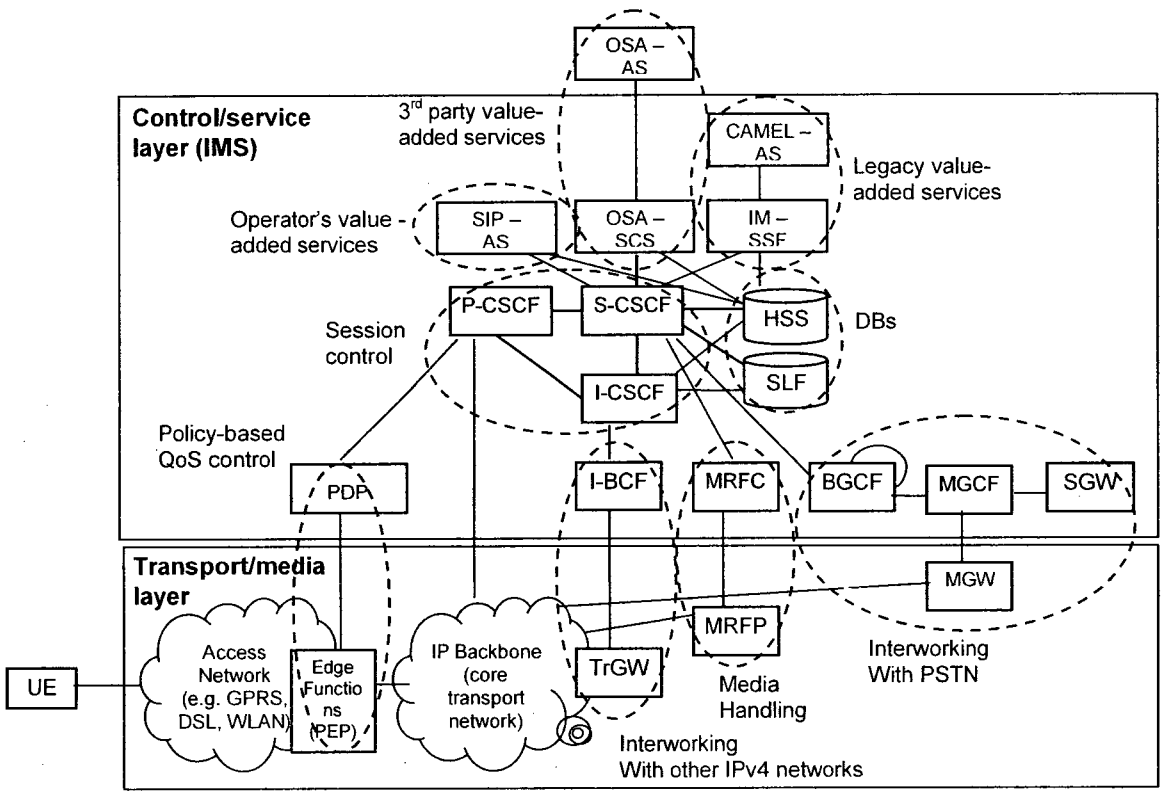


Figure 2.1: An overview of the 3GPP IMS architecture

This architecture can be divided into two main layers: a control/service layer responsible of the management of sessions (i.e. sessions' setup, modification, and tear-down) and the execution of value-added services; and a transport/media layer responsible of IP connectivity and media handling operations (e.g. media mixing and trans-coding). Those

layers encompass a set of functional entities connected via standardized interfaces (also called "reference points"). Each entity could reside on one physical node, or be distributed among several physical nodes. In the coming sub-sections, we describe those entities and the interfaces and protocols they use, and then present a basic session setup scenario to illustrate the IMS operation.

2.1.1.1 Functional entities

As shown in figure 2.1, the IMS architecture consists of six main types of functional entities: databases, session control entities, media handling entities, interworking entities, policy enforcement entities, and entities providing value-added services. The roles of those entities are summarized as follows:

- **Databases:** There are two main types of databases in the IMS: The Home Subscriber Service (HSS) containing all the user-related subscription, authorization, and authentication information; and the Subscriber Location Function (SLF) which maps users' addresses to HSSs (in case more than one exits).
- **Session control entities:** Several SIP servers, collectively known as Call/Session Control Functions (CSCFs), handle signaling operations in the IMS. There are three types of CSCFs: The Proxy-CSCF (P-CSCF), the Serving-CSCF (S-CSCF), and the Interrogating-CSCF (I-CSCF). The P-CSCF is the first point of contact between the User Equipment (UE) and the network, acting as inbound/outbound proxy and performing additional functions such as security associations' establishment, message integrity verification, billing information generation, and media resources authorization. Located at the edge of the administrative domain, the I-CSCF assigns S-CSCFs to users performing SIP registration and routes messages to the appropriate S-CSCF. The S-

CSCF acts as registrar, authenticates users trying to access the network, enforces network and user-related policies, and triggers the appropriate services at the application servers.

- **Media handling entities:** Two entities offer Media handling capabilities (e.g. media streams mixing, and media trans-coding) in the IMS, namely: the Media Resource Function Processor (MRFP), and the Media Resource Function Controller (MRFC). The MRFP is a media plane node implementing the actual media related functions, while the MFPC is the signaling plane node controlling it.
- **Interworking entities:** To enable the interworking between IMS-based networks and circuit-switched networks, a Breakout Gateway Control Function (BGCF) and a PSTN gateway - internally divided into a Media Gateway Control Function (MGCF), a Media Gateway (MGW), and a Signaling Gateway (SGW) - are used. The BGCF performs routing based on telephone numbers, while the PSTN gateway acts as mediator between the two types of networks by performing the needed mappings and protocols conversion. Furthermore, an IMS Gateway (IMS-GW), composed of an Interconnection Border Control Function (I-BCF) and a Transition GateWay (TrGW), is used to enable the interworking of the IMS (relying on IPv6) with other IP-based networks using IPv4.
- **Policy enforcement entities:** By using policies, operators can control access to transport resources and negotiate reasonable QoS parameters. To achieve policy-based QoS control, two entities are used: a Policy Decision Point (PDP), and a Policy Enforcement Point (PEP) residing inside the edge function of the access network. The roles of these two entities will be elaborated in section 2.1.2.1.

- **Entities providing value added services:** Value-added services are services that go beyond two-party calls (e.g. multimedia conferences, call diversion, and call screening). In the IMS, those services are hosted and executed by application servers. Three types of application servers are defined in the IMS architecture: SIP application servers (SIP-AS) executing the IMS operator's value-added services; Open Services Architecture application servers (OSA-AS) executing value added services provided by third parties; and Customized Application Mobile-Enhanced Logic application servers (CAMEL-AS) executing legacy GSM value-added services.

It should be noted that beside those entities, there is another important type of functional entities in the IMS, namely: charging related entities. Those entities, which are not shown in figure 2.1, will be described in section 2.1.2.2.

2.1.1.2 Protocols and interfaces

The IMS relies on existing standard IP protocols with specific profiles and enhancements, to provide the needed functionality. A summary of those protocols is given below:

- **SIP (Session Initiation Protocol):** Session control (or signaling) protocols, which enable session initiation/modification/termination operations, play an important role in any communication system. SIP was selected by 3GPP as the session control protocol for the IMS. It consists of a set of specifications defined by the Internet Engineering Task Force (IETF). The specifications comprise a core part and several extensions. The core part (specified in RFC 3261 [25]) is a signaling protocol for creating, modifying, and terminating multimedia sessions. The extensions provide additional capabilities such as event notification [26] and sessions' preconditions enforcement [27]. There are two main components in the core part of SIP: User agents and network servers. A user agent is an end-system which takes orders from a user and acts on her/his behalf, to

setup and tear down sessions with other user agents. Network servers provide to user agents services such as application level routing. Reference [28] can be consulted for an overview of SIP.

- **Diameter:** Diameter is used as the AAA (Authentication, Authorization, and Accounting) protocol in the IMS. It consists of a base protocol (specified in RFC 3588[29]) and a number of extensions (also known as "Diameter applications") which are defined to suit the needs of particular applications. The Diameter credit control application, used for online charging in the IMS, is an example of such extensions.
- **COPS (Common Open Policy Service):** COPS is an IETF protocol specified in RFC 2748 [30], for the purpose of enabling the exchange of policy-based requests/decisions between policy clients and policy servers. In some releases of the IMS, COPS is used for the exchange of policy-based decisions related to QoS enforcement.
- **H.248:** H.248, also known as the MEGACO (MEdia GAteway COntrol) protocol [31], was jointly developed by IETF and ITU-T for the purpose of enabling signaling nodes to control nodes in the media plane. An example of such use in the IMS is between the MRFC and the MRFP.
- **RTP/RTCP (Real Time Transport Protocol / RTP Control Protocol):** RTP and RTCP are two complementary protocols used for the transportation of real-time media in the IMS. These protocols are specified in RFC 3550 [32].

The protocols presented above are used for the exchange of messages between IMS functional entities, over various interfaces. A summary of the most important IMS interfaces, along with the protocols used over them, is depicted in table 2.1.

Interface name	IMS entities involved	Protocol
Cx	HSS, S/I-CSCF	Diameter
Dx	SLF, S/I-CSCF	Diameter
Gm	P-CSCF, UE (via access network)	SIP
Go	PDP, PEP	COPS
Gq	P-CSCF, PDP	Diameter
ISC	AS, S-CSCF	SIP
Mg	MGCF, I-CSCF	SIP
Mi	S-CSCF, BGCF	SIP
Mj	BGCF, MGCF	SIP
Mn	MGCF, MGW	MEGACO
Mp	MRFC, MRFP	MEGACO
Mr	S-CSCF, MRFC	SIP
Mw	CSCF, CSCF	SIP
Mx	I-BCF, I/S-CSCF	SIP
Sh	HSS, AS	Diameter
Ut	AS, UE (via access network)	HTTP

Table 2.1 Summary of IMS interfaces [33]

2.1.1.3 Basic session setup scenario

To be able to use the services of the IMS, a number of requirements must first be satisfied. The first requirement is the establishment of a service contract between the IMS network operator and the user, after which the user is provided with a smart card hosting a number of applications. One of those applications is the ISIM (IP multimedia Services Identity Module) that provides storage for a collection of parameters, including: private and public user IDs, a home network domain URI, and a long term shared secret. The private user ID is a globally unique identifier used for authentication, authorization, and billing purposes. It takes the form of a network access identifier (e.g. username@operator.com) and is hidden from the user. As for the public user ID, it is known to the user and is used for routing of SIP traffic. Public user IDs can either be SIP URIs (e.g. sip:alice.jones@operator.com) or TEL URLs. The home network domain URI represents the name of the home network administrative domain, while the long term secret is used for security related operations.

As a second requirement, the IMS terminal needs to connect to an IP Connectivity Access Network (e.g. GPRS, ADSL, or WLAN) which will provide IP connectivity to the IMS

core network. Third, the IMS terminal needs to discover the address of the P-CSCF that will act as outbound/inbound SIP proxy server. Finally, the terminal must register at the SIP application level to the IMS network. This IMS level registration allows the network to locate the user (i.e. binding the public user ID to a physical contact address), to authenticate/authorize the user and allocate a S-CSCF to him/her, and to establish the needed security associations between the user equipment and the network. Once the IMS registration procedure completed, the UE can use the IMS services to establish different types of sessions. Figure 2.2 illustrates a basic two-party session setup scenario in the IMS. For simplicity, we assume that both users have subscribed with the same operator and are in their home network.

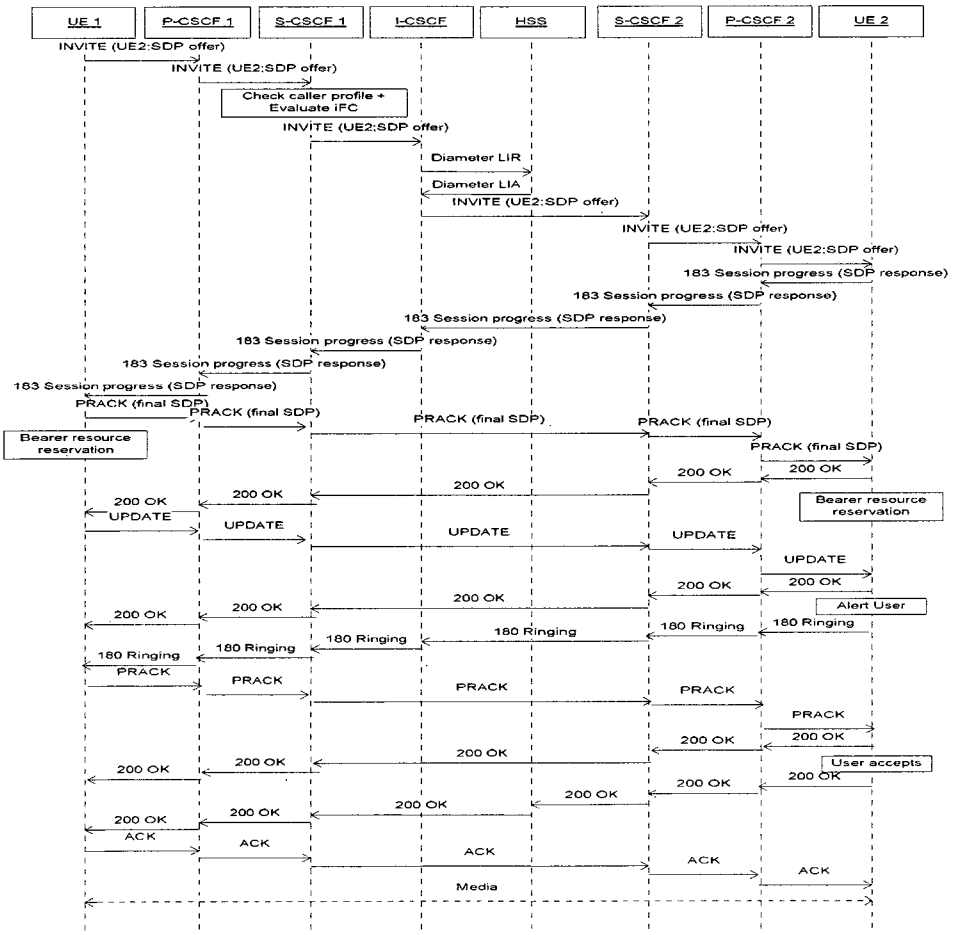


Figure 2.2: IMS basic session setup scenario

In this scenario, UE1 attempts to establish a session with UE2. It sends a SIP INVITE message containing an initial SDP offer (representing the media types/formats it supports) to its P-CSCF. This last forwards the request to the S-CSCF allocated to the user (i.e. S-CSCF1), which checks the caller's profile to ensure that the user is entitled to the service type requested and evaluates the initial filter criteria (IFC) to determine if any services should be triggered at application servers. Assuming that the user has not subscribed to any additional services, S-CSCF1 forwards the request to the local I-CSCF, which queries the HSS to get the address of the S-CSCF assigned to the callee (S-CSCF2 in this case). The I-CSCF then forwards the request to S-CSCF2 that checks the callee's profile, evaluates the IFC, and forwards the request to the next hop (P-CSCF2). This last forwards the request to UE2. To complete the media parameters negotiation procedure, UE2 returns a subset of the media capabilities that it finds acceptable, in a provisional 183 SIP response message traversing the reverse signaling path to UE1. This last then sends a PRACK SIP message (including the mutually agreed upon session parameters) to acknowledge the receipt of the provisional response, and carries a series of interactions for bearer resources reservation. The PRACK message is on its turn acknowledged by a 200 OK SIP response sent by UE2, which also performs bearer resource reservation from its side. After securing resources for the session, UE1 sends an UPDATE SIP message indicating the success of the resource reservation, which is answered by a 200 OK SIP response by UE2. After this stage, the callee is alerted (resulting in 180 Ringing/PRACK/200 OK messages exchange between the two UEs), and then answers the call. This is translated into a 200 OK SIP message forwarded to the caller's UE. This last responds with a SIP ACK message to acknowledge the receipt of the final response, and the media starts flowing between the two end-points.

2.1.2 QoS and Charging in the IMS

QoS provisioning and flexible charging are important aspects that are considered in the IMS. In this section, we highlight the solutions proposed by 3GPP to handle these aspects.

2.1.2.1 QoS support in the IMS

QoS provisioning implies the network's ability to distinguish between different classes of traffic (or service) and provide each class with the appropriate treatment, depending on its needs in terms of QoS parameters. QoS is important for offering 3G users a predictable experience and a controlled quality, when using IP multimedia services. Several end-to-end QoS models [34] can be supported in the IMS. For instance, terminals can rely on link layer resource reservation protocols such as PDP context activations or use RSVP or DiffServ codes directly, while core network elements can use DiffServ or RSVP for QoS enforcement. When cellular terminals are involved (i.e. GPRS access is used), the most common model consists of terminals using link layer protocols to make the necessary resource reservations, while gateway nodes (e.g. GGSN) map link layer resource reservation flows to DiffServ codes in the network.

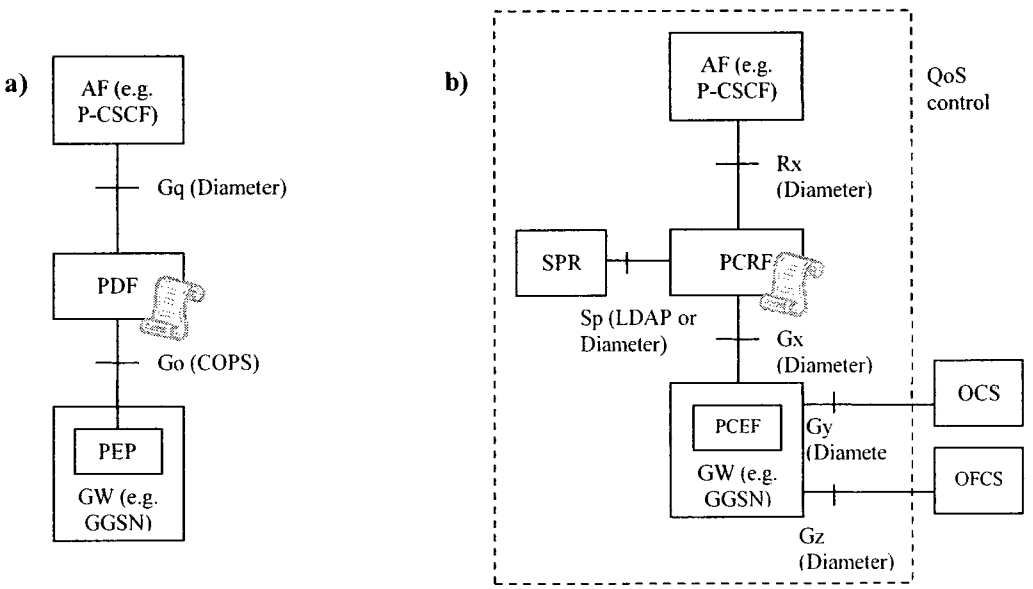


Figure 2.3 IMS QoS architectures: a) Rel. 6 QoS architecture; b) Rel.7 QoS support as part of the PCC architecture

Figure 2.3a illustrates the initial IMS QoS architecture proposed for GPRS access, in release 6 of the specification [34]. This architecture relies on three main functional entities: a Policy Decision Function (PDF), an Application Function (AF), and a Policy Enforcement Point (PEP). The PDF is the point where policy decisions related to QoS control are made. To make those decisions, the PDF uses information obtained from the session signaling (mainly from the media parameters negotiation interactions) and which is conveyed to it by the AF over the Diameter-based G_q interface. The AF is an element offering applications that require the control of IP resources. Examples of AFs are application servers and P-CSCFs. The decisions made by the PDF are enforced by the PEP, which is a logical function residing within the access network gateway (e.g. a GGSN). The PEP communicates with the PDF over the COPS-based G_o interface. It should be noted that in release 7 of the specification [35], policy-based QoS control was integrated with charging control in a single architecture called the Policy and Charging Control (PCC) architecture [36], which is depicted in figure 2.3b. The part of the PCC architecture that is related to QoS control is similar to the release 6 architecture, except that the PDF is now replaced by another entity called the PCRF (Policy and Charging Rules Function), and the PEP is replaced by the PCEF (Policy and Charging Enforcement Function). Unlike the PDF, the PCRF makes policy-based decisions related to both QoS control and charging control. Those decisions are enforced by the PCEF, which communicates with the PCRF over a Diameter-based interface called the G_x interface (instead of the previous COPS-based G_o interface). An additional entity called the SPR (Subscription Profile Repository) was also introduced in the PCC architecture. This entity provides the PCRF with QoS-related information about the user's subscription over the S_p interface.

2.1.2.2 IMS Online and Offline Charging Architectures

Charging constitutes a critical function in any commercial network as it represents the mechanism by which resource consumption is accounted for and revenue is generated. The 3GPP charging framework [37] covers three levels of charging, namely: bearer-level charging (in the circuit switched and packet switched domains); subsystem-level charging (in the IMS); and service-level charging. Furthermore, two charging mechanisms (i.e. offline and online charging) are defined within this framework. In offline charging, the network reports resource usage to the billing domain via a series of functional entities, after the resource consumption. This charging mechanism is often associated with post-paid billing and involves no direct interaction with the service being rendered. In contrast, online charging is used to realize pre-paid billing and involves real-time interaction between the charging process and the service being rendered, as means to achieve credit control. In this case, chargeable events must be authorized (based on the user's available balance and the price of the event) before the actual resources are consumed.

In this section, we focus on subsystem-level charging and give an overview of the IMS offline and online charging architectures specified in [38].

A. The IMS offline charging architecture

Figure 2.4 depicts the IMS offline charging architecture. In this architecture, the different IMS functional entities involved in session establishment (i.e. SIP-ASs, the MRFC, the MGCF, the BGCF, and the P/I/S-CSCFs) participate in offline charging by the means of an integrated charging trigger function (CTF), monitoring the signaling traffic for the occurrence of chargeable events (i.e. messages triggering the generation of charging information). These entities use the Rf interface to report the captured charging information to the charging data function (CDF). This last makes use of the received information to

generate charging data records (CDRs), which are sent to the charging gateway function (CGF) over the Ga interface. The CGF acts as gateway between the 3G network and the billing domain, by pre-processing and storing the received CDRs. Finally, these CDRs are transferred over the Bi interface to the billing domain, in which rating (i.e. price calculation) occurs.

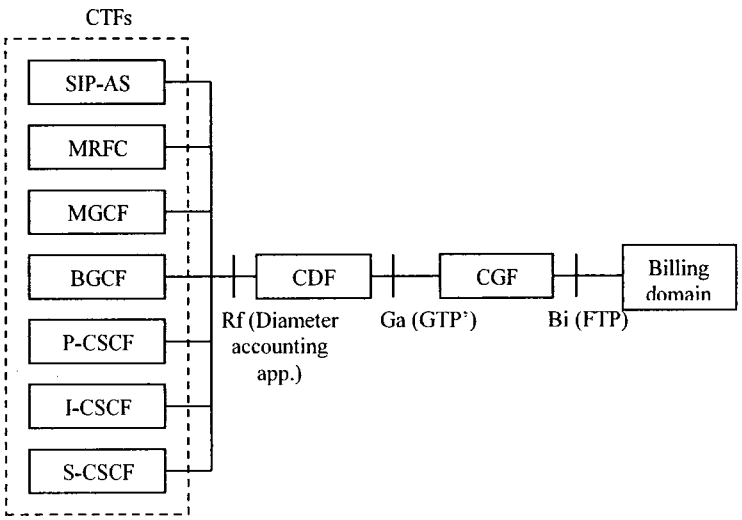


Figure 2.4: The IMS offline charging architecture [38]

In terms of protocols, the accounting part of the Diameter base protocol [29], extended with 3GPP specific attribute-value pairs (AVPs) defined in [39], is used for the Rf interface. Two Diameter messages are used in this case: ACR (accounting request) and ACA (accounting answer). As for the Ga and Bi interfaces, an enhanced GPRS tunneling protocol and a file transfer protocol are employed respectively.

B. The IMS online charging architecture

As shown in figure 2.5, the number of functional entities that may participate in online charging is limited to three: SIP-ASs, the MRFC, and the S-CSCF. The MRFC and SIP-ASs act as online CTFs, monitoring the signaling traffic for the occurrence of chargeable events and reporting them to the OCF (online charging function) – one of the components of the online charging system (OCS) – over the Ro interface. The messages exchanged over that interface are mainly CCRs (credit control requests) and CCAs (credit control

answers), defined as part of the Diameter credit control application [40]. As for the S-CSCF, it employs a special gateway (the IMS-gateway function (IMS-GWF)) for credit control interactions with the OCS. The IMS-GWF acts as online CTF with respect to the OCS and appears as a regular application server to the S-CSCF, with which it interacts using SIP (over the ISC interface). In this case, IFC (initial filter criteria) must be set in users' profile to enable the S-CSCF to determine which messages should be directed to the IMS-GWF and thus be charged using the OCS.

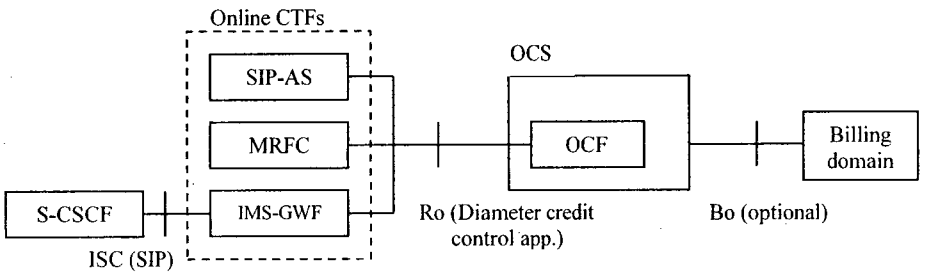


Figure 2.5: The IMS online charging architecture [38]

In terms of interactions between the OCF and CTFs, three credit control cases can be distinguished: immediate event charging (IEC), event charging with unit reservation (ECUR), and session charging with unit reservation (SCUR). For IEC, exchanged messages are of type CCR [event]/CCA and lead to a direct debiting of a certain amount of units from the user's account. On the contrary, both ECUR and SCUR perform a reservation of units from the user's account and carry out the debiting after the service is rendered, as follows: the CTF starts the credit control session by sending a CCR[initial] to request an initial quota, that is either granted/denied using a CCA. For SCUR, several CCR[update]/CCA pairs may be exchanged as the session progresses and resources are consumed. When the session is terminated, a CCR[terminate]/CCA pair is exchanged to stop the credit control session. This results in a debit transaction on the user's account, including a potential release of reserved but non-consumed quota.

2.1.3 IMS Advanced Services: Presence and Emergency Service Support

In addition to the basic services provided by the IMS (e.g. two-party session setup, charging, and QoS support), 3GPP has also standardized some IMS advanced services. Examples of such services include: conferencing, presence, instant messaging, push-to-talk, and emergency service support. In this section, we discuss two of those services, which are of special interest to us. It is important to mention that beside these two levels of services (i.e. basic and advanced services), a third level of services built on top of the IMS is also possible. This third level represents value-added services which are not standardized by 3GPP.

2.1.3.1 The 3GPP Presence Framework

Presence is a widely accepted concept in communication networks. It refers to information conveying users' ability and willingness for communication. The initial presence framework was defined by the IETF, in RFC 2778 [41]. This framework relies on three main types of entities, namely: a presence service that accepts, stores, and distributes presence information; presentities (short for presence entities) that provide presence information to the presence service (i.e. act as information source); and watchers that request information from the presence service (i.e. act as information consumers). In order to represent presence information in a format that facilitates its sharing and understanding, the IETF presence framework uses a standard XML-based information model called the Presence Information Data Format (PIDF) [42]. This PIDF structures presence information in a set of elements called presence tuples, and defines the syntax and semantics of the sub-elements forming these tuples. Figure 2.6 illustrates the structure of the PIDF.

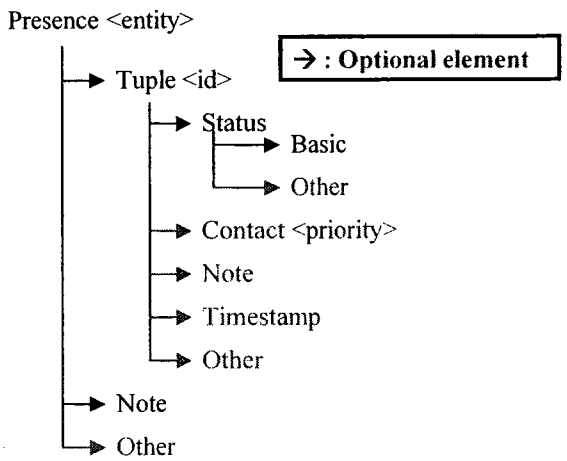


Figure 2.6: The PIDF structure

It should be noted that although the PIDF only defines basic presence information (e.g. status and contact information), it is highly extensible. In fact, several extensions have already been proposed, such as: the RPID (Rich Presence Information Data format) [43] which provides additional presence information about persons and their devices and services (e.g. activities, mood, place type/properties...etc); the CIPID (Contact Information in Presence Information Data format) [44] which provides contact information related extensions; and the GEOPRIV [45] which provides geographical location related extensions. In terms of information exchange, several standard protocols have been proposed to enable the exchange of presence information between the different entities, such as: the SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) protocol [46]; and the XMPP (Extensible Messaging and Presence Protocol) [47].

To support the presence service in the IMS, 3GPP adopted the IETF presence model and defined its own presence architecture [48]. Figure 2.7 illustrates this architecture, which distinguishes between three types of presentities: Presence User Agents (PUAs) publishing information provided by users; Presence Network Agents (PNAs) publishing information provided by network entities about the user; and Presence External Agents (PEAs) publishing information provided by external entities/networks about the user. In addition to

presence agents, there are four other types of entities in the 3GPP presence architecture: the presence server, the presence server list, presence proxies, and watchers. The presence server (PS) is responsible for the management of presence information published by agents and the fusion of data from multiple sources into a single presence document, while the presence list server (PLS) is responsible for the management of subscriptions to groups of users. Presence proxies (e.g. presentity/watcher presence proxies) act as inbound/outbound proxies to the presence network, performing routing, security, and charging functions. The roles of presence proxies are assumed by CSCFs, as shown in the figure. As for watcher applications and watcher ASs, they subscribe to presence information of interest, therefore acting as presence information consumers. In this architecture, users are charged for different information related interactions, such as: presence information publication, requesting/accessing information and updates, and access to watchers' information.

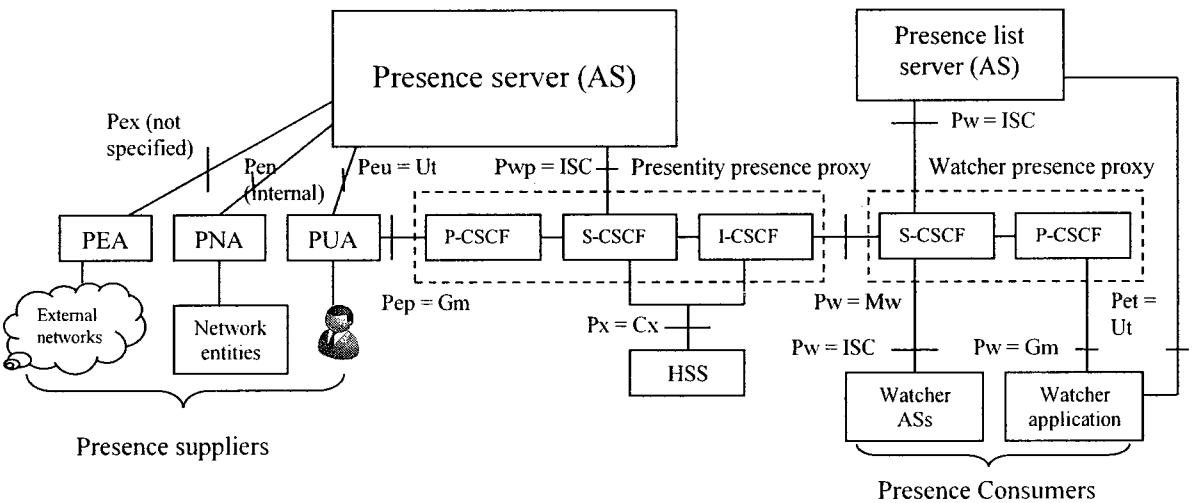


Figure 2.7: The 3GPP presence architecture

Figure 2.7 also shows the different interfaces involved in the 3GPP presence architecture. Most of those interfaces are existing IMS SIP or Diameter interfaces that map to a presence-oriented function. It should be noted that the Pex interface is not fully specified, rather serving as a place holder in the architecture. The same applies to the role of PEA.

2.1.3.2 The IMS Emergency Service Architecture

Emergency services represent one of the fundamental and most valued services provided by communication networks. They enable the public to summon help in case of emergency, and the emergency service agencies to respond quickly in order to minimize life and property losses. An IMS emergency service architecture has been recently proposed by 3GPP [49]. Figure 2.8 illustrates this architecture, which relies on four main functional entities: the UE, the P-CSCF, the E-CSCF (emergency-CSCF), and the LRF (location retrieval function).

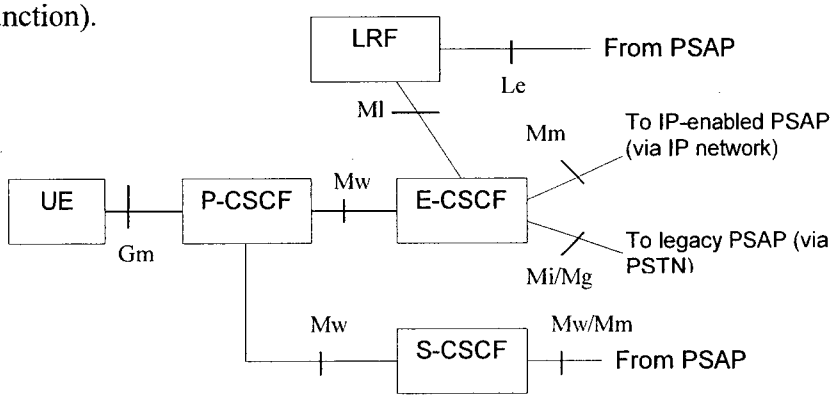


Figure 2.8: The IMS emergency service architecture [49]

The P-CSCF and the E-CSCF are SIP servers, which handle different aspects of emergency sessions' establishment/termination. The P-CSCF is the first point of contact between the UE and the network. It acts as inbound/outbound proxy, and performs authentication/ authorization, emergency session prioritization, and application level routing to the appropriate E-CSCF. The E-CSCF is responsible for acquiring/validating the location of the UE (by interacting with the LRF) and routing the call to the appropriate (IP-enabled or legacy) PSAP (public safety answering point). The LRF interacts with location severs and/or the access network to retrieve the location of the UE that has initiated the session. It may also provide PSAP route determination services.

The UE is the equipment used to make the emergency call. It detects that an emergency session is being established (based on the number dialed), registers with the IMS using a

special emergency public user ID, determines its own location if possible (either using an internal location measurement mechanism or by interacting with the access network), and sends an emergency session establishment request to the P-CSCF with the needed information (e.g. the emergency public user ID and the location information). The P-CSCF then performs authorization/ authentication of the session and the user, prioritizes the emergency session, and forwards the session establishment request to an E-CSCF in the same network. If the location information provided by the UE is insufficient (i.e. missing or not accurate), the E-CSCF interacts with the LRF to acquire/validate the information. After that, the E-CSCF determines the address of an appropriate PSAP (based on this info), and routes the call to this PSAP, thus completing the call establishment. It should be noted that the PSAP is the only entity that can terminate the call - if the caller hangs up or gets disconnected; the PSAP operator initiates a callback to re-establish the call.

2.2 Context and Context-Awareness

The concepts of context and context-awareness have been traditionally used in pervasive computing environments as means to enhance users' experience and as enablers to adaptability and service personalization. In this section, we shed some light on the meaning of these concepts, present a categorization of context and context-awareness features from our perspective, give an overview of the potential sources of contextual information, and describe the different operations related to context management.

2.2.1 Context Definition and Categorization

In order to effectively use context and build context-aware systems, a precise definition of context is needed. A categorization of context is also needed in order to clarify the boundaries between the different types of information used, and identify the different mechanisms needed to deal with these types.

In this work, we adopt Dey's definition of context, which states that "context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is relevant to the interaction between a user and an application, including the user and the application themselves" [50]. Furthermore, we divide contextual information into two broad categories:

- *User-centric information*: This includes information about the user (e.g., user location, preferences, surrounding people/devices, physiological data) and his/her surrounding physical environment (e.g., light, noise, and temperature).
- *Network-centric information*: This includes information about the devices carried by the users (e.g., CPU level, available memory, battery level), and the characteristics of the links existing between those devices (e.g., connectivity, available bandwidth, QoS parameters, security level).

2.2.2 Overview of Context Sources

Contextual information can be automatically collected from various sources. User-centric information, which mainly relates to physical phenomena (e.g. location, temperature, heart rate ...etc), is typically collected using individual sensors or groups of sensors forming Wireless Sensor Networks (WSNs). Sensors are electronic devices that can detect physical phenomena or stimuli from the environment and produce an electric signal. A WSN consists of a set of small sensors equipped with processors, memory, and short range wireless communication capabilities. The sensor nodes collaborate to collect and aggregate data about the phenomena under observation. Figure 2.9 illustrates a typical WSN architecture, which consists of three main types of nodes: sensors, sinks, and gateways. The sensors do the actual sensing, while the sinks collect data from all the sensors, and interact with applications via the gateway. This last has a dual network interface, and acts as a link

between the WSN and the outside world by performing the needed mappings and protocol conversions. Usually, the sink and the gateway are co-located and simply referred to as either sink or gateway. Reference [51] provides a detailed overview of WSNs.

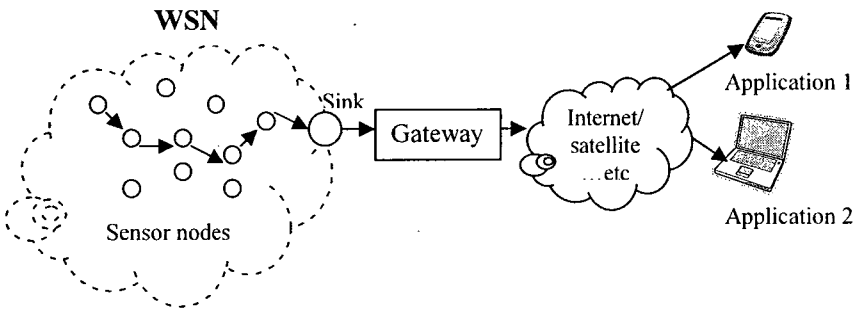


Figure 2.9: A typical wireless sensor network architecture

As for contextual information relating to the logical state of an entity (e.g. a network), it can be collected from functional entities keeping track of this information. In a networking environment, network entities acting as logical sensors (e.g. routers and CSCFs) can be used for the collection of network status information such as network capacity, sessions' status, and QoS parameters.

2.2.3 Context Awareness Definition and Categorization of Context Awareness Features

Context-awareness has been defined as "the ability to use contextual information to provide relevant information and/or services to the user" [50]. From the perspective of autonomic systems, context-awareness can also be defined as *the ability to use contextual information for operations and decision making.* Since our focus is on communication applications, we categorize context-awareness features in terms of the system's proactiveness in supporting different phases of the communication, as follows:

1. *Notification* of caller/callee about situational information prior or during communication: This feature consists of increasing the user's awareness of his surrounding environment and the situation of other communication partners. The most

popular implementation of this feature consists of exposing users' presence information, in order to convey their ability and willingness to communicate. Another type of information which could be useful is the type of communication that can be supported by the network (audio, video, or text), based on the network resources status. A combination of these two types of information would enable the user to determine the most suitable time to establish communication, and the most appropriate communication type to attempt.

2. Proactive *initiation* of communication session: This feature consists of spontaneously initiating a communication session between some participants, when certain conditions are satisfied. For instance, a session could be automatically established between the attendees of a conference to enable them to share files and chat. Beside the automatic selection of communication partners, the system could also support other aspects of the session initiation. For instance, it could automatically direct the call to the most appropriate device to the callee (selection of communication device), or automatically select the media type/format based on the user and the network situation (negotiation).
3. Proactive *modification* of communication session: This feature consists of spontaneously modifying the characteristics of an ongoing session, based on the current situation. Modifications can be made in terms of the media exchanged within a session, or the session's participants, or the devices they use for communication. Some of the possibilities are: the automatic migration of a call to another device (when the user moves); the automatic session re-negotiation to switch to a media format that could be supported with high quality by the network; and automatically moving users between sessions depending on their position in a game space or in a rescue operation.

4. Proactive *termination* of communication session: consists of terminating an ongoing session when a threshold is reached, or in case of unexpected situations. An example could be to automatically terminate an audio session between two users when the system detects that they are collocated in the same room. Another example could be the suppression of unnecessary traffic in crisis situations, by terminating ongoing regular sessions, in order to save resources for more important communications (e.g. mission critical and emergency calls).

2.2.4 Context Management Related Operations

To ensure the availability of contextual information that will be used for the support of different context-aware operations/features in the IMS, several context management-related operations are needed. These operations are summarized as follows:

- Context acquisition: The first operation consists of collecting the needed contextual information from various sources. As mentioned previously, these sources could be physical sensors (such as location and environmental sensors), or logical sensors (such as network probes, agendas, the operating system).
- Context modeling and processing: As a second step in the context management process, any information that has been collected must be represented in a format that makes is easy to use, understand, and share with other entities. Furthermore, since not all information can be acquired directly from sensors (e.g., a user's social situation can't be sensed directly), suitable reasoning/inference mechanisms may be needed for the generation of higher level information from lower level pieces of data.
- Context dissemination: Once the information has been collected and processed, the system can give access to it via different interaction models (e.g., publish/subscribe, polling, tuple spaces).

Chapter 3

Requirements and Related Work Review

This thesis touches several areas related to the introduction of the context-awareness technology in the IMS, namely: context acquisition/management; service differentiation; and emergency service support in IP-based networks. In this chapter, we derive requirements related to these problems, and evaluate the related work in light of these requirements.

3.1 Requirements

3.1.1 Requirements for Context Information Acquisition and Management in 3G Networks

To ensure the availability of contextual information in the network, the first requirement that must be satisfied is the ability to interact with different information sources (i.e. physical and logical sensors) to collect a variety of situational information (e.g. spatial, environmental, physiological, and network status information). The second requirement is that the system should be able to manage information related to different types of contextual entities (e.g. persons, objects, places, and networks), to enable a wide range of context-aware applications/ services. It should be noted that this implies the identification of those contextual entities in the network and the ability to use those high level identifiers in information exchange related interactions (e.g. publishing/subscribing to information related to a certain contextual entity).

As a third requirement, the system should rely on a formal information model to facilitate contextual knowledge representation and sharing. In a 3G networking context, this implies the use of a standard IMS-compatible format for the representation of different types of information provided by physical/logical sensors. Furthermore, to offer refined information

representing different levels of granularity (e.g. spatial information in the form of geographical coordinates, a room identifier, a displacement direction, or a relative distance to an object/person), the system must support different reasoning/inference mechanisms needed for information processing related operations.

As a fifth requirement, the system should support standard IMS protocols for information exchange related interactions on the inbound interface (i.e. the interface residing between the information sources and the 3G network) and the outbound interface (i.e. the interface residing between the 3G network and the information consumers). Moreover, the chosen IMS protocols should enable both synchronous and asynchronous modes of communications to achieve flexible information access/dissemination in the network. Finally, to enable the practical deployment of the system proposed, the solution should rely on a suitable business model accommodating the case in which information sources are owned by the 3G network operator (e.g. network probes deployed by the 3G operator) as well as the one in which information sources are owned by a third party (e.g. WSNs deployed by another operator). The realization of such model implies the need for several support functions regulating the interaction between the 3G network and information sources/consumers. Those functions include: security and access control (to control access to the network resources and protect information integrity and privacy); charging (to account for network resources utilization); and service discovery (to enable the discovery of information sources and information management entities, and their capabilities).

3.1.2 Requirements for Service Differentiation in 3G Networks

To achieve service differentiation in 3G networks, the first requirement that should be satisfied is the ability to distinguish between multiple classes of service in terms of the

treatment they get in the network, taking into consideration two aspects, namely: the traffic requirements in terms of resources; and the importance of the service session from the user's perspective. It should be noted that the consideration of these two aspects (or dimensions) would result in a fine level of granularity in terms of service differentiation, by enabling users to choose the application category they wish to use as well as the priorities/guarantees they wish to obtain on a specific application session.

Second, the system should offer flexible QoS negotiation mechanisms, by giving the user the possibility to select the service class for each call (thus expressing the level of importance of the call when it is established) in addition to the ability to dynamically change the selected class during the session if needed - for instance, a user may wish to upgrade the class of a call to improve its quality, or to protect it from being dropped or downgraded, during an overload or crisis situation.

In order to efficiently utilize the network's resources and appropriately react to unexpected/crisis situations, the system should be able to dynamically allocate resources to (or transfer resources between) different classes of calls, taking into consideration the network situation and the sessions' QoS profiles. Furthermore, the consistency of the treatment offered to a certain class of calls should be guaranteed by offering preferential treatment at the beginning and during sessions. This implies the control of different communication aspects (e.g. controlling when the session starts/ends, the session size, and the used media type/format).

As a fifth requirement, the solution should guarantee that the treatment received by calls belonging to a certain class is uniform across the network, irrespective of the access networks used to establish them (i.e. show access network independence). Moreover, a

suitable charging model should be offered as part of the solution, to enable the effective charging of differentiated services while being attractive to users.

Finally, the solution should introduce reasonably low complexity and overhead to the existing network architecture, and should show satisfactory performance in terms of call setup time, in order to be practically deployable.

3.1.3 Requirements for Enhanced Emergency Communications Support in 3G Networks

Emergency communication systems rely on five main components/steps in their operation: an emergency number (an easy to remember number simplifying access to emergency services); methods for determining the caller's location and phone number to enable the proper handling of the call; a prioritized emergency call handling mechanism (to give emergency calls priorities over regular calls); a location-based routing and call establishment mechanism (to route call to the most appropriate PSAP and establish the call); and a mechanism for presenting the calls taker with the information needed to handle the call (e.g. caller's position and nearest available emergency responders) [52]. To support enhanced emergency communications in 3G networks, several requirements that are directly linked to these five components, are needed. The first is the reliance on a universal identifier, with an international significance, for the identification of emergency calls. Such an identifier is needed to resolve discrepancies between emergency dial-strings used in different countries and to enable different elements to unambiguously identify/handle emergency calls. As a second requirement, the system should be able to determine and manage the caller's contextual information, which could be exploited to enhance emergency operations and increase their efficiency.

Due to the importance of emergency calls, they need to be provided preferential treatment

over regular calls (e.g. faster call setup times and higher probability of completion) and prioritized access to resources, especially when there is a strong contention for scare network resources. Therefore, the third requirement entails the ability to provide preferential treatment to emergency calls and prioritize their access to resources in a resource efficient and adaptive manner.

Furthermore, the system should be able to determine the most appropriate PSAP based on the caller's situation, and route the call to it (i.e. support context-aware routing), in order to offer more efficient and targeted help to the user. An example of a context-aware routing scheme could be to route the call to the nearest emergency-specific PSAP, which supports the user's terminal media capabilities and employs call takers speaking the user's preferred language. Another requirement is to support the establishment of multimedia, multiparty emergency sessions. In fact, the use of multimedia (e.g. audio, video, text messaging) in emergency sessions could lead to richer communications and a better assessment of the caller situation, while multiparty capabilities (e.g. conferencing/sub-conferencing) could enable a better coordination of efforts during emergency situations and the collaboration between different groups for large rescue/relief operations.

The sixth requirement implies the presentation of the call taker with a wide view about the caller's situation (e.g. location, surrounding people/devices, vital signs/biometric data...etc), in order to enable a better assessment of the caller's situation and the provision of better help to the user.

Finally, the system should achieve a satisfactory performance in terms of call setup time in order to cater to the urgent nature of emergency calls and be practically deployable.

3.2 Related Work Review

3.2.1 Information Acquisition and Management Solutions for IP-Based Networks

The related work on information acquisition and management in IP-based networks can be divided in two main categories. The first category encompasses WSN integration solutions that focus on the information acquisition aspects of the problem, by enabling the interaction with WSNs through IP-based networks. As for the second category, it includes solutions enabling the management and dissemination of the collected information in the network.

3.2.1.1 Wireless Sensor Network Integration Solutions

Several solutions have been proposed for the integration of WSNs in the Internet, the most prominent examples being: Alarm-Net [53], TinySIP [54], TinyREST [55], and IP-enabled WSNs [56]. On the other hand, few solutions investigated the integration of WSNs with 3G networks, including the E-Sense framework [57] and the work on service discovery gateways presented in [58]. In this section, we describe these solutions and evaluate their capabilities with respect to our first set of requirements.

The Alarm-Net solution [53] is used for monitoring patients in homes and nursing residencies, by enabling medical applications (running on PCs and PDAs) to access WSNs information via a set of IP-based alarm gates acting as gateways. Figure 3.1 illustrates the Alarm-Net architecture, which relies on a query-based protocol for information exchange. This solution does not rely on standard communication protocols nor does it use a standard data format for information representation, which limits its applicability in a 3G environment. Furthermore, it does not provide any charging and service discovery mechanisms. In fact, WSNs are considered as owned by the IP network operator in this solution, not considering the case of a third-party owned WSN infrastructure. Finally, the

Alarm-Net solution only enables the interaction with physical sensors (but not logical sensors), supports limited information processing capabilities (i.e. the analysis of sensed data for the detection of patients' behavioral patterns), and can only handle information related to persons as contextual entities.

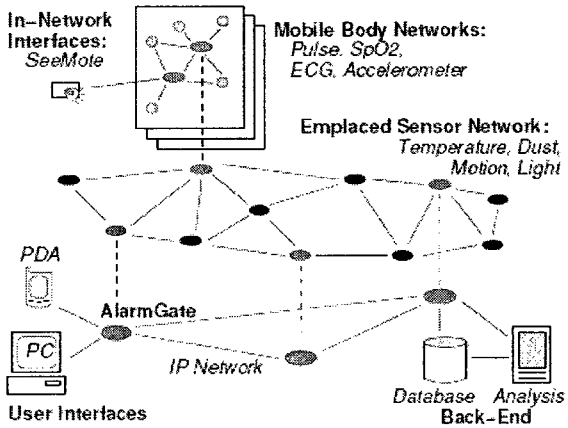


Figure 3.1: The Alarm-Net architecture [53]

Unlike Alarm-Net, TinySIP [54] and TinyREST [55] provide communication abstractions that enable the exchange of information between WSNs and the Internet, using standard IP protocols. TinyREST relies on an HTTP-based mechanism to query/change/subscribe to the state of sensors and actuators in a smart home environment, while TinySIP proposes the use of SIP to achieve more sophisticated interactions with a sensor network. Examples of such interactions include: instant messaging based queries (synchronous mode); session-based polls (asynchronous mode), subscriptions/notifications (asynchronous mode); in addition to the possibility to fork a request to multiple sensors and the possibility to redirect a request to an alternate node. Figure 3.2a depicts the TinyREST architecture, which enables Internet clients to issue HTTP requests to access a URL-identified resource (i.e. a sensor or an actuator) in a WSN. Those requests are conveyed by a home server (offering addressing and application level routing services) to an HTTP-2-TinyREST gateway. This last maps the HTTP requests to TinyOS messages that are broadcasted to the WSN and

responded to by the concerned sensor(s). The TinySIP architecture operates in a similar fashion, by relying on a SIP-TinySIP gateway that enables clients to interact with sensors via SIP-based requests. The main role of this gateway is to map the SIP messages sent by applications to TinySIP messages (a lightweight version of SIP), which it then routes towards the appropriate sensor node (or zone manager). Figure 3.2b illustrates one of the interaction scenarios supported by TinySIP (i.e. the 'publish-subscribe' interaction scenario).

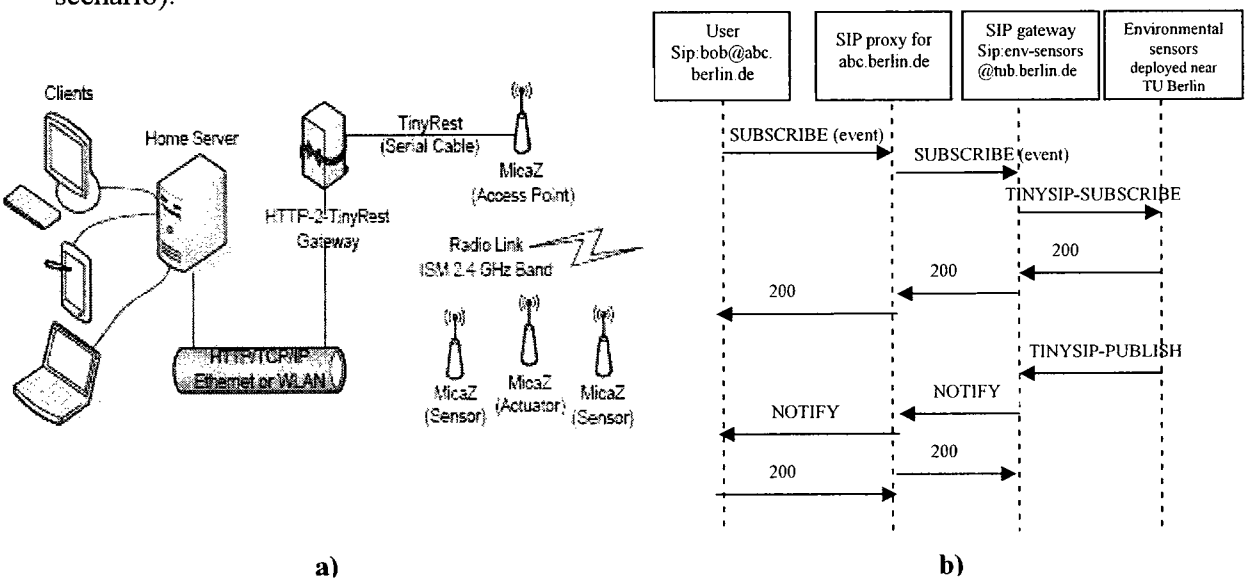


Figure 3.2: The TinyREST and TinySIP solutions: a) The TinyREST architecture; b) TinySIP's Publish-Subscribe interaction scenario

Although these two solutions rely on standard IMS protocols, they only focus on the information exchange aspects of the problem (i.e. on the remote invocation of sensors via the Internet), not tackling other important aspects needed for their adoption in a commercial networking environment (e.g. security and charging). Furthermore, these solutions do not offer information modeling and processing capabilities, and enable the interaction with physical sensors only.

In contrast with the previously presented approaches that rely on an application-level gateway acting as intermediary between WSNs and Internet clients, the IP-enabled WSNs

approach [56] focuses on enabling direct communication between clients and WSNs by deploying an IPv6 stack on sensor nodes. Figure 3.3a depicts the protocol stack of an IPv6-enabled sensor node. It shows how the IP protocol is adapted to the WSN environment by adding a 6lowpan transportation mechanism [59] to achieve interworking between the IP layer and the sensor's link layer technology (IEEE 802.15.4 [60] – also known as Zigbee - in this case). This is in addition to the support of other functions needed to achieve header compression, multi-hop routing (performed using LOAD [61] – a simplified on demand routing protocol based on AODV), and IP address auto-configuration (performed using adapted version of IPv6 auto-configuration mechanisms – the neighbor discovery protocol (NDP) [62] and the stateless address auto-configuration mechanism (SAA) [63]).

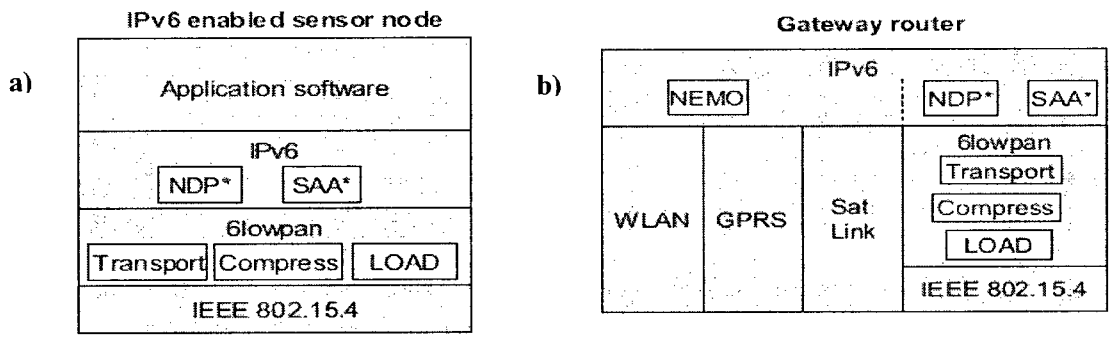


Figure 3.3: Architectures of nodes used in the IP-enabled WSNs architecture: a) IPv6-enabled sensor node's protocol stack; b) Gateway's protocol stack

The IP-enabled WSN architecture also relies on IP-level gateways acting as routers between the sensor nodes and the IP network infrastructure. As shown in figure 3.3b, such gateways may be equipped with several networking interfaces to ensure different types of connectivity between the sensor nodes and the IP backbone, in addition to a NEMO (NEtwork Mobility module) [64] operating at the IP layer to achieve mobility management. As in the case of sensor nodes, the 6lowpan transportation mechanism is used to enable IP/Zigbee interworking, and NDP and SAA are deployed in the gateway's IP layer to enable gateway address discovery.

This solution enables the interaction with physical sensors (i.e. WSNs) only, and does not enable the identification of the contextual entities to which the information relates, since it requires the direct addressing of sensor nodes. Furthermore, it does not offer any information modeling and processing capabilities, although sensor nodes may perform some limited information processing locally. The protocol used for information exchange depends on the sensor-based application software supported, which runs directly on top of IP (unlike standard IMS application-level protocols). Furthermore, this solution lacks support for important functions (e.g. security and charging) needed for its deployment in a secure and controlled 3G environment.

The e-SENSE framework [57] aims at making ambient intelligence available to beyond 3G networks, to enhance their service provisioning capabilities. Figure 3.4 shows an overview of the E-Sense reference architecture, which encompasses three main components: WSNs capturing the ambient intelligence; a core network containing elements that consume the contextual information (e.g. application servers and service platforms); and gateways interfacing WSNs with the core network. In this architecture, a 'publish-subscribe' messaging abstraction is used for the exchange of information between sensor nodes and also within the core network. Furthermore, sensory information is processed in a distributed fashion on sensor nodes and sensor gateways.

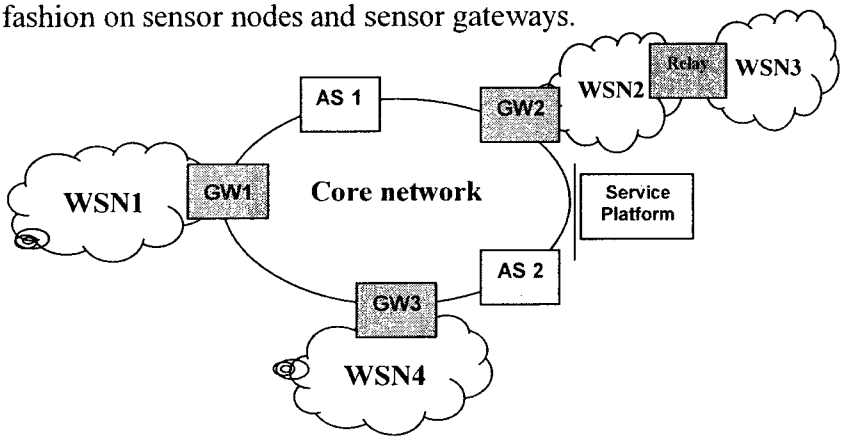


Figure 3.4: The E-Sense reference architecture

This solution focuses on the information acquisition aspects of the problem by addressing issues such as internal communications between sensor nodes (playing different roles), their discovery and coordination, and the collaborative data processing between them. However, issues related to the management and usage of the contextual information in the core network, are not addressed. For instance, this solution does not support the modeling of contextual information using a standard data format, and does not specify a standard protocol enabling information exchange in the network. Furthermore, it does not offer any charging mechanism, and provides limited information processing capabilities since the information is processed locally on sensor nodes (which have limited resources). Moreover, this solution does not enable the identification of the contextual entities concerned within the core network, rather focusing on the addressing of sensor nodes and sensor gateways. Finally, it is important to mention that the E-Sense framework aims at providing a generic solution enabling sensor nodes to interact with external networks. However, the specific integration of WSNs in IMS-based networks is not addressed in this work, although the authors mention the possibility to interface their system with the IMS GUP (Generic User Profile) server as means for the integration.

The work presented in [58] introduces the idea of service discovery gateways as means to connect a local network (with non-IMS devices such as house appliances) to the IMS infrastructure, and proposes three possible deployment scenarios: 1) Leveraging the local network's resources to access IMS services (e.g. routing a video stream to a TV instead of the mobile phone); 2) Using the IMS infrastructure to access the local network's services/resources (e.g. accessing the video feed of home surveillance cameras via the IMS infrastructure); and 3) Using the IMS as an intermediary between two cooperating local

networks (e.g. a user's hotel room network cooperating with his/her home network to access multimedia content stored on a home media server), as illustrated in figure 3.5. All these scenarios rely on service discovery gateways (SDGs), sitting between the local networks and the IMS, to enable the interworking between the disparate service discovery mechanisms employed by the local networks in addition to making the IMS aware of the services provided by those networks. Those SDGs rely on a SIP-based protocol called PIRANHA for remote service invocation sessions establishment, and use the media plane to send service commands to the services/devices to be invoked.

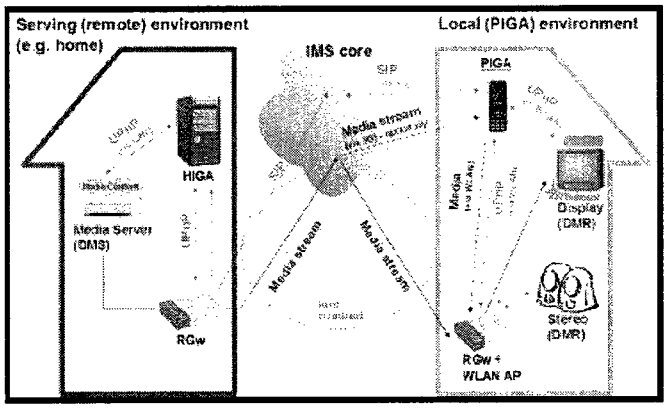


Figure 3.5: Remote service cooperation between two local networks via the IMS [58]

In general, the case of WSNs integration in the IMS could fit within the second deployment scenario proposed by this solution – the WSN being considered as a local network that is accessible via the IMS infrastructure. However, this solution only focuses on service discovery and remote service invocation (which is done over the media plane, using proprietary protocols), not satisfying any of the other requirements (i.e. information modeling/processing, security, charging, and contextual entities identification).

3.2.1.2 Information Management Solutions

Three main information management solutions have been proposed for 3G networks, namely: the presence framework proposed by 3GPP [48], the presence enabler proposed by

OMA (the Open Mobile Alliance) [65, 66], and the Generic User Profile (GUP) architecture proposed by 3GPP [67]. On the other hand, the main information management solution proposed for the Internet is the presence framework defined by the IETF [41].

The 3GPP presence framework [48], which is based on the IETF presence model, has been described in chapter 2. This framework enables the interaction with external networks/entities (e.g. physical sensors) as well as internal network entities (possibly acting as logical sensors), via Presence External Agents and Presence Network Agents. Furthermore, the current presence model only handles information about user entities, although it could be enhanced to handle information about other type of contextual entities due to its extensibility. In terms of information representation, the presence framework relies on standard IMS-compatible information models (the PIDF and the RPID). However, these models only handles a subset of the contextual information that could be provided by physical/logical sensors - for instance, basic spatial information (i.e. location) and rudiments of environmental information (e.g. place properties) are handled while physiological and network status information are not accommodated. To enable synchronous and asynchronous information exchange on the outbound interface (i.e. between the presence server and information consumers), the presence architecture relies on a standard IMS protocol (i.e. SIP) and a subscribe/notify interaction model - the inbound interface between the network and external information sources remaining however undefined. Finally, this framework considers different business models including the one in which un-trusted entities/networks act as information sources, although the needed security, service discovery, and charging mechanisms regulating the interaction of those information sources with the 3G network remain to be defined.

The IETF presence framework [41], which was also described in chapter 2, mostly has the same drawbacks as the 3GPP presence framework (that is derived from it), including: the inability to handle information related to different types of contextual entities (the focus being on user entities); and the reliance on information models that allow the representation of a subset of contextual information only (i.e. basic/rich presence information). As additional limitations, the IETF presence framework does not specify a specific business model for the commercial deployment of presence systems nor does it define some of the important support functions needed (e.g. charging and service discovery).

Beside the IETF and 3GPP, OMA is another standard body working on presence. More specifically, OMA is working on the specification of a presence enabler [65] running on top of the IMS. This enabler (a sort of application building block) follows the 3GPP presence architecture and IETF standards (i.e. presence protocols and data formats) and adds some extensions to fulfill OMA-specific requirements. Among those extensions are new data elements that were introduced to the standard presence information model as means to enrich it with high-level pieces of information facilitating the development of presence-based applications. Examples of such extension elements (which are specified in [66]) include: Application-specific Willingness, Availability, and Media Capabilities.

Beside the OMA-defined presence data model, the OMA solution also described the architectural components needed for the realization of the presence enabler and its interaction with other OMA enablers, while readily relying on IMS basic capabilities (e.g. routing, security, charging). Figure 3.6 illustrates the OMA presence architecture. This architecture integrates some of the 3GPP presence architecture's existing entities (which are marked with a red star in the figure), such as: the presence server; the presence list

server; the watcher; the watcher agent (corresponding to the role of watcher presence proxy in the 3GPP architecture); and the presence source (corresponding to the role of presence agent in the 3GPP architecture). Furthermore, it introduces new entities, such as: the watcher information subscriber that subscribes to retrieve watchers information (this function being previously implemented as part of the presence user agent functionality in the 3GPP architecture); the content server that manages presence related MIME objects by interacting with the PS, presence sources, and watchers; in addition to three XML Document Management Servers (XDMSs), two of which managing XML documents related to the PS and the RLS (namely the presence-XDMS and the RLS-XDMS) while the third (the presence content-XDMS) manages media files for the presence service.

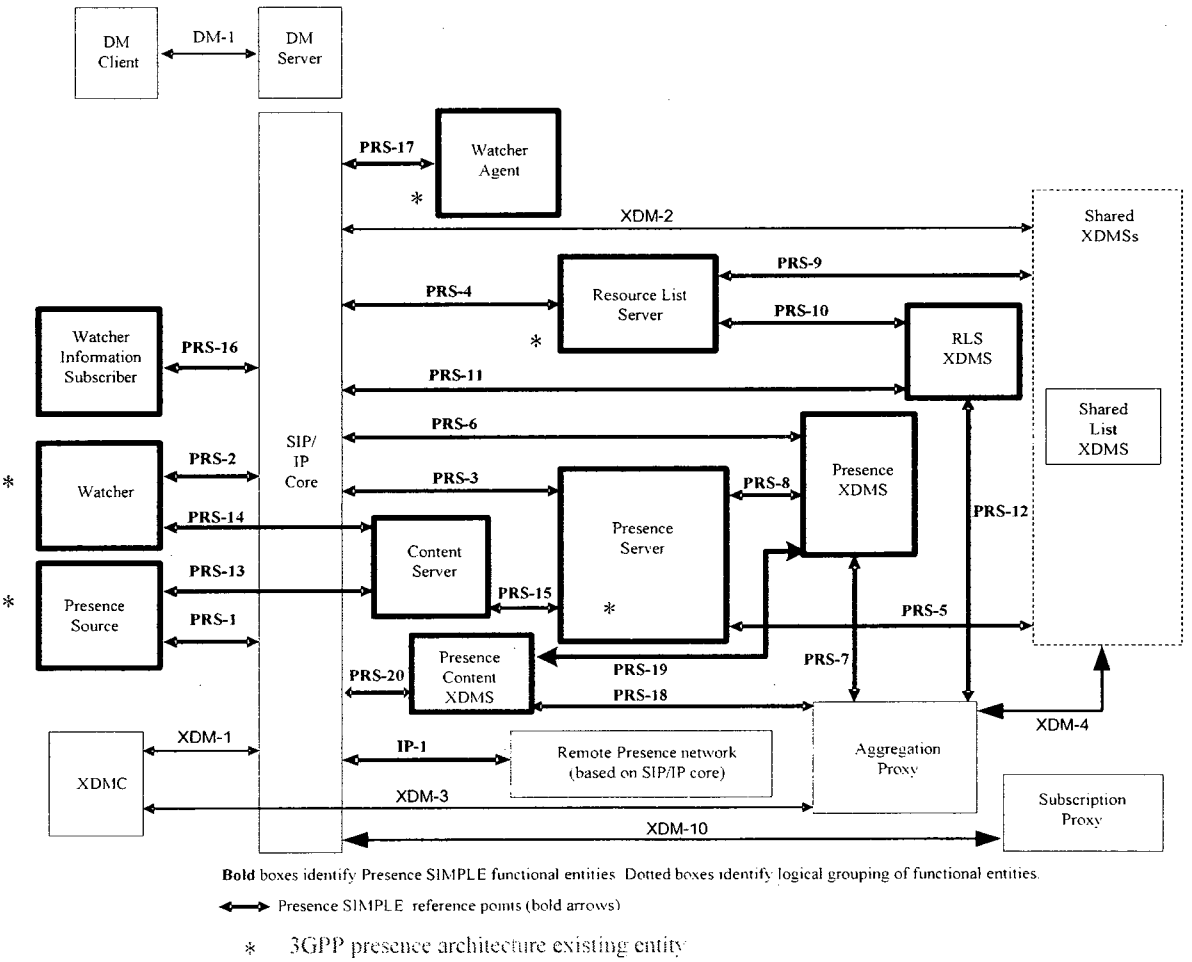


Figure 3.6: The OMA presence architecture [65]

The evaluation of this solution with respect to our requirements is very similar to the (previously presented) evaluation of the 3GPP presence solution, on which it is based, except for the added complexity which is introduced to achieve some OMA-specific additional functionality (e.g. the interaction with other OMA enablers, the dedicated management of XML documents and MIME objects, and the extension of the PIDF to specify application-level presence information semantics). It should be noted that this added complexity does not bring any benefits with respect to the problem of context management, since it focuses on application-level semantics and issues related to the interoperability between application building blocks. For instance, in terms of information modeling, the PIDF OMA extensions remain within the scope of the presence concept that is limited to info conveying ability/willingness for communication, about users and their services and devices (e.g. application specific willingness/availability are related to the "service" that is user is willing/able to use, while the mood and activity data elements are related to the "user" component), not handling additional types of information (e.g. environmental, physiological, and network status information) about other types of entities (e.g. objects and places) that could be relevant for the development of context-aware applications. We note that despite the alignment between the 3GPP and the OMA standardization activities, the reliance on OMA service enablers is not mandatory in the IMS architecture.

The GUP architecture [67], which is depicted in figure 3.7, is another information management solution proposed by 3GPP. The goal of this architecture is to enable harmonized access (including the creation, modification, and reading) to user-related profile information, which is distributed on different entities in the 3G network. The

architecture consists of four functional entities: the GUP server, Repository Access Functions (RAFs), GUP data repositories, and applications. The GUP server provides a single point of access to GUP data, in addition to several other functions (e.g. the location/synchronization of profile components and the authentication/authorization of profile requests). The GUP data repositories act as storage for the profile information, while the RAFs provide a layer of abstraction on top of those repositories by hiding their implementation details and providing uniform access to their data. End user applications and third party applications interact with the GUP server (over the Rg interface) to have access to GUP data, while operator's applications have direct access to this information by interacting with the RAFs (over the Rp interface).

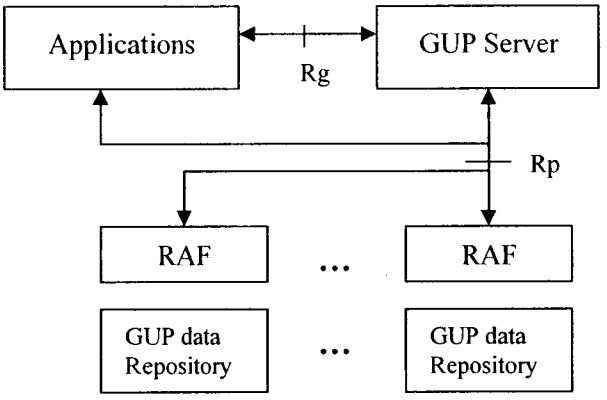


Figure 3.7: The 3GPP GUP architecture [67]

This solution cannot handle any of the information types provided by physical/logical sensors, since it focuses on user profile information management. Furthermore, the information handled is only related to user entities. As for information modeling and processing capabilities, the GUP architecture relies on a generic data model for information representation but does not offer any information processing capabilities. Furthermore, the information exchange protocols are not specified by the current GUP architecture. The same applies to the needed security and charging mechanisms, which remain to be defined.

3.2.1.3 Summary of Evaluation

Table 3.1 presents a summary of the evaluation of the different information acquisition and management solutions presented.

By examining this summary, we notice that all the WSN integration solutions share some common drawbacks, such as: enabling the interaction with physical sensors only (not considering the case of logical sensors); relying on low level identifiers (e.g. addresses of sensor nodes or sensor gateways) for service invocation; lacking support for information modeling and providing limited or no information processing capabilities; in addition to not addressing business issues related to their practical deployment (e.g. charging and security). While WSN integration solutions focus on the information acquisition aspects of the problem, information management solutions complement these approaches by tackling issues related to the management and usage of information in the core network. Nevertheless, the existing information management solutions have some limitations with respect to context information management related requirements. Among those limitations, we mention: their inability to handle information related to different types of contextual entities (the focus being on user entities); their reliance on information models that allow the representation of a subset of contextual information only (i.e. presence information or user profile information); and their focus on outbound interface related interactions (in terms of protocols definition and support functions).

Among all the solutions evaluated, the 3GPP presence framework seems to be the most promising since it satisfies most of our requirements. Another interesting solution is the OMA presence enabler. However, in comparison with the 3GPP framework, this approach introduces additional functionalities/complexities that do not serve the purpose of the problem at hand.

Solution/ Req.	R:1-1		R:1-2	R:1-3	R:1-4	R:1-5			R:1-6		
	Physical sensors	Logical sensors	Mgmt & ID. of info of different Contextual entities	Formal Info. Model	Processing Cap.	Standard IMS Protocols	Synchronous communication	Asynchronous communication	Security	Charging	Service Discovery
Alarm-Net	Yes	No	Only persons (type of info about subject)	No	Ltd.	No	Yes	No	Yes	No	No
TinyREST	Yes	No	Partial (addressing of resources)	No	No	Yes	Yes	Yes	No	No	No
TinySIP	Yes	No	No (addressing of sensor nodes)	No	No	Yes	Yes	Yes	No	No	Yes
IP-enabled WSNs	Yes	No	No (addressing of sensor nodes)	No	No	No	-	-	No	No	No
E-Sense	Yes	No	No (addressing of sensor nodes)	No	Ltd. (locally on SN)	No	No	Yes	Partial (privacy only)	No	Partial (between SN only)
Discovery Gateways	Yes	No	No (addressing of SDGs)	No	No	No (proprietary – over media)	-	-	No	No	Yes (via SDGs)
3GPP Presence framework	Yes - via PEA	Yes - via PNA	Only persons	Yes – handles subset of context info Only	Yes – by PS	Yes (defined over outbound i/f only)	Yes	Yes	Yes (defined over outbound i/f only)	Yes (defined over outbound i/f only)	Yes (defined over outbound i/f only)
IETF Presence framework	possible - via PUA	possible - via PUA	Only persons	Yes – handles subset of context info Only	Yes – by PS	Yes (SIMPLE)	Yes	Yes	Yes – but not fully specified	No	No
OMA Presence enabler	Yes - via Presence src	Yes - via Presence src	Only persons	Yes – focus on application - level info	Yes – by PS	Yes (defined over outbound i/f only)	Yes	Yes	Yes (defined over outbound i/f only)	Yes (defined over outbound i/f only)	Yes (defined over outbound i/f only)
GUP	No	Yes - apps	Only persons	Yes – but doesn't handle any type of context info	No	Not defined	-	-	Yes – but not yet specified	No	No

Table 3.1 Evaluation of information acquisition/management solutions

3.2.2 Service Differentiation Solutions for IP-Based Networks

The existing service differentiation solutions can be divided in two main categories. The first category consists of solutions operating at the access network level, and which are specifically designed for certain access technologies. As for the second category, it includes core network level solutions that support multiple access technologies. Since we are mainly interested in access agnostic solutions, we will start by presenting a brief discussion about the first category of solution, then focus on the detailed evaluation of the second category.

3.2.2.1 Service Differentiation Solutions: Access Level

Several service differentiation solutions have been proposed at the mobile access level [68, 69, 70], while few others have been proposed at the fixed [71] and wireless access levels [72]. All these approaches focus on the management of access networks' resources and offer solutions that are specific to certain access types.

The work presented in [68] proposes an adaptive framework for the provision of connection-level QoS in next generation cellular networks. This framework focuses on giving preferential treatment to different classes of connections based on their importance and resource requirements. In fact, the service differentiation scheme proposed in this work offers two levels of levels of differentiation, namely: the differentiation between different classes of traffic, such as audio, video, and data (i.e. a media-type based model); and the differentiation between new and handoff calls within each traffic class (i.e. a mobility-based model). Furthermore, two differentiation parameters are used to distinguish between the different classes: the New Call Blocking Probability (NCBP) representing a measure of service connectivity; and the Handoff Call Dropping Probability (HCDP) representing a measure of service continuity. To realize the proposed scheme, two algorithms are used

with the goal of maximizing the system's utilization while minimizing the HCDP, namely: a connection admission control (CAC) algorithm; and a bandwidth adaptation algorithm (BAA). Although this work offers a fine level of granularity in terms of service differentiation by proposing a hybrid service differentiation scheme (i.e. a scheme combining media type and mobility based models) and relies on a dynamic/adaptive resource allocations strategy, it has some shortcomings: it does not offer flexible QoS negotiation mechanisms to the user as the negotiation is done implicitly by the system; it offers limited control over the different communication aspects since it is operating at the connectivity level (for instance, aspects such as session size and media type/format cannot be controlled at this level); and it is specific to mobile access networks (thus not being applicable to other types of access). Furthermore, no charging model is proposed as part of the solution and its performance in terms of call setup time has not been evaluated.

The work presented in [69] promotes the idea of service-classes based QoS over air interfaces in 4G networks. The main goal of this work is to enable the sharing of radio resources among multiple users with different QoS requirements, in a fair and efficient manner. To achieve this goal, an Olympic model based scheme, enabling the distinction between different classes of service (gold, silver, and premium), is proposed. The only differentiating factor used in this case is the allocated B.W. guarantee, and the algorithm used for the realization of the scheme is a B.W. scheduling algorithm. This solution offers a fine level of service differentiation granularity since it introduces an Olympic-based model to the existing application category based models usually supported on air interfaces. Furthermore, it has the potential of offering flexible QoS interactions with the user (although no details are provided on how these interactions can be practically realized) and

relies on an adaptive resource allocation strategy. Nevertheless, it remains specific to CDMA air interfaces (thus not being applicable to other types of access technologies); offers preferential treatment at the beginning of sessions only (not at the beginning of sessions); and offers limited control over communication aspects (the focus being on the control of allocated bandwidth). Moreover, it does not provide a suitable charging scheme.

The solution presented in [70] focuses on prioritized resource allocation in stressed networks, by dynamically allocating radio resources to important connections during stress situations via distributed ticket servers and an upper limit connection admission policy. The differentiation scheme proposed in this work follows a stress-based model, in which the differentiation is made between classes of connections based on their levels of importance (e.g. high priority vs. low priority). The main differentiating factor used in this scheme is the NCBP and the algorithm used for its realization is a connection admission control algorithm. Like the previous solution, this solution offers a fine level of service differentiation granularity (stress and application category based models), provides flexible negotiation mechanisms to the user (via ticket servers), and relies on a dynamic admission control strategy (although this strategy does not adapt to changing network situations). In terms of limitations, this solution focuses on offering preferential treatment at the beginning of session only, provides limited control over the different communication aspects (mainly connection admission), and remains specific to mobile access technologies, in addition to not providing any specialized charging model. Furthermore, its performance in terms of session setup time is not discussed by the authors.

Tackling the issue of IP traffic differentiation in broadband wireline access networks, the work presented in [71] aims at minimizing the delay of real-time traffic as well as the

packet loss of P2P traffic. The differentiation scheme proposed in this work follows the application category based model in which traffic is classified based on the application type (i.e. VoIP, video, WWW, or FTP applications). The main differentiating factor used in this scheme is the traffic delay sensitivity and the mechanism used for its realization is multi-class queuing and scheduling. This solution is very preliminary and does not satisfy most of our requirements. In fact, it only tackles one service differentiation dimension (i.e. traffic req. in terms of resources), does not involve the user in QoS negotiation that is performed implicitly by the system, relies on a basic (non-dynamic) traffic scheduling mechanism that uses predefined classes priorities, offers preferential treatment during sessions only, and allows the control of a single communication aspect (i.e. routing delay). Furthermore, it is specific to fixed access networks and does not provide a specialized charging model.

The work presented in [72] examines the issue of MAC-level QoS provisioning in IEEE 802.11e WLANs. It presents a MAC simulation model illustrating the effect of adjusting and combining two of the 802.11e parameters (namely, the AIFS and the CW_{min}) on QoS provisioning. Moreover, it proposes a set of rules for establishing a service differentiation scheme comprising three classes (gold, silver, and bronze). This solution focuses on the second dimension of service differentiation (i.e. session importance level) and does not offer flexible QoS negotiation mechanisms to the user. Furthermore, it relies on a static resource allocation strategy (fixed parameters for each class) and provides preferential treatment during sessions only, in addition to being specific to fixed access networks and not providing a suitable charging model.

3.2.2.2 Service Differentiation Solutions: Core Network Level

Existing core network level service differentiation solutions can be divided in two categories, namely: solutions operating at the routing/transport level including the 3GPP

IMS QoS architecture [34] and the TISPAN RACS architecture [73]; and solutions operating at the signaling/control level such as the 3GPP IMS emergency solution [49], the work on SIP traffic prioritization [74] proposed by Lucent, and the multi-level precedence and preemption (MLPP) solution [75] proposed by the ITU.

The 3GPP IMS QoS architecture [34] has been described in chapter 2. This architecture focuses on GPRS mobile access scenarios and differentiates between four classes of traffic, namely: the conversational class (for audio/video call applications); the streaming class (for audio/video streaming applications); the interactive class (for web browsing applications); and the background class (for the background delivery of e-mails). These traffic classes are distinguished by their delay sensitivity (conversational traffic being the most delay sensitive and background traffic being the least), and several QoS parameters are defined as differentiating factors (e.g. max. bit rate, delivery order, and transfer delay). In this architecture, four main functional entities (the UE, the P-CSCF, the GGSN, and the PDF), using two mechanisms (PDP contexts and DiffServ) and enforcing an authorize-reserve-commit resource management model, collaborate for the realization of this service differentiation scheme.

Although this solution enables preferential treatment at the beginning of sessions (via resource authorization) and during sessions (via resource commitment), and introduces reasonably low complexity and overhead to the existing network architecture, it still has several drawbacks: it does not offer a fine level of service differentiation granularity since it only tackles one service differentiation dimension (i.e. traffic reqs. in terms of resources); it does not offer flexible QoS mechanisms to the user as the negotiation is performed implicitly by the system (as part of the interactions between the P-CSCF and the UE); it

relies on a static resource management strategy that is based on pre-defined parameters/rules; and it offers limited control over the different communication aspects (the focus being on traffic delay and allocated B.W.). Furthermore, although the resource management mechanisms used in the core part of the network are access agnostic, the QoS negotiation mechanism used (i.e. PDP contexts) is specific to GPRS access networks, thus not being applicable to other types of access. Finally, in terms of charging, the IMS QoS architecture does not define any specific charging model, rather relying on the basic IMS charging model.

Building on the 3GPP IMS QoS architecture, the TISPAN RACS (Resource and Admission Control Subsystem) architecture [73] incorporates the fixed access scenario and proposes two QoS models, namely: the guaranteed QoS model ensuring service delivery with absolute bounds on some or all of the QoS parameters via admission control, throughput control, and traffic policing; and the relative QoS model implying traffic aggregate differentiation using appropriate QoS mechanisms at the IP network edge. As shown in figure 3.8, the RACS architecture bears some similarities to the IMS QoS architecture and consists of four main functional entities: the SPDF (Service-based Policy Decision Function), the A-RACF (Access – Resource and Admission Control Function), the BGF (Border Gateway Function), and the RCEF (Resource Control Enforcement Function). The SPDF is the equivalent of the PDF in the 3GPP architecture. It enforces local policies on resource usage by collaborating with an application function (e.g. the P-CSCF) and communicates with the A-RACF to obtain call admission decisions related to the access network. As for the BGF (a core border node) and the RCEF (a core edge node), they act as policy enforcement points, respectively enforcing the SPDF and the A-RACF

decisions by means of gate control, packet marking, and traffic policing.

Like the 3GPP architecture, the TISPAN architecture tackles only one service differentiation dimension by focusing on traffic resource requirements, does not offer the user any control over QoS negotiation in which traffic classes are assigned by the network (based on administrative policies), offers a limited control over communication aspects, and involves certain aspects that are specific to the fixed access type (namely, resource management policies installation/manipulation in transport layer nodes). However, unlike the 3GPP solution, the TISPAN solution offers the possibility of dynamically configuring QoS policies in edge nodes, and specifies some of the details related to the needed charging model (namely, the specific charging information required).

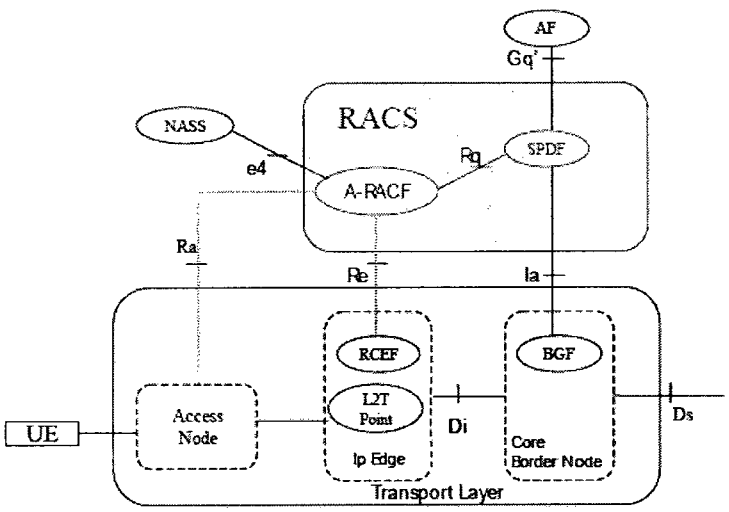


Figure 3.8: The TISPAN RACS architecture [73]

In contrast with the previously presented transport level solutions that focused on the differentiation between classes of traffic, signaling level solutions add another level of differentiation by enabling the distinction between classes of sessions within the same traffic class. One of these signaling-level solutions is the IMS emergency solution [49] that has been presented in chapter 2. This solution enables the support of emergency communications in the IMS by proposing a stress oriented service differentiation scheme,

which distinguishes between two classes of calls (i.e. regular and emergency calls). The main differentiation factor in this case is the signaling messages' processing delay and the mechanism used for the realization of the scheme is a priority queuing mechanism. Although this solution is access network independent and relies on a specialized charging model (emergency calls not involving any charges and being potentially used by non-subscribed users), it does not involve the user in QoS negotiation since the class of the call is automatically assigned by the system based on the dialed number, in addition to relying on a basic (non-dynamic) queuing mechanism that uses predefined classes priorities. Furthermore, it offers preferential treatment at the beginning of sessions only, and allows the control of a single communication aspect (i.e. session establishment delay).

Another signaling level differentiation solution has been proposed by Lucent for the prioritization of signaling traffic in SIP proxy servers [74]. The goal of this solution is to assign appropriate priorities to different SIP messages and use these priorities to determine the relative order in which the messages are processed in SIP servers. Several potential classification schemes are proposed in this work, namely: classification based on the urgency of the request (e.g. regular vs. emergency call); classification based on the request/service type (e.g. conversational vs. presence vs. instant messaging applications); and classification based on the request initiator (e.g. regular vs. privileged user). The main factor differentiating the different classes of signaling traffic is the message processing delay and the resource management mechanisms implemented by SIP servers consist of prioritized queuing and message rejection.

This solution offers a fine level of service differentiation granularity since it allows the support of different classification schemes. Furthermore, it is access network independent

and allows the dynamic update of resource management policies and their potential adaptation to changing network situations (although details on how this adaptation can be achieved are not provided), in addition to introducing low complexity and overhead to the existing network architecture. However, the QoS negotiation mechanisms it offers to the user remain non flexible since in some classification schemes (e.g. schemes based on urgency and service type), priorities are automatically assigned by the system, while in other schemes (i.e. the scheme based on the user type), the differentiation between users is based on their subscription type (i.e. all calls made by a certain user will belong to a certain category and no change of this category is allowed during the session). Finally, it offers limited control over the different communication aspects (the focus being on session establishment delay) and does not specify a specialized charging model.

Beside the previously presented solutions that have been designed for the packet switched domain, the MLPP solution [75] proposed by the ITU was designed to enable prioritized call handling in the circuit switched domain. Five priority levels are defined by this solution and hard preemption is used for resource re-allocation. This solution offers a fine level of service differentiation granularity and allows the user to explicitly choose the service category on a per call basis, in addition to relying on a dynamic resource allocation strategy (i.e. a preemptive strategy). On the other hand, the solution does not allow the change of the service category during an ongoing session, enables preferential treatment at the beginning of sessions only and offers limited control over communication aspects (i.e. when the session begins/ends), in addition to not specifying any charging model.

3.2.2.3 Summary of Evaluation

Table 3.2 presents a summary of the evaluation of the different service differentiation solutions presented.

Solution/ Req.	R:2-1	R:2-2	R:2-3	R:2-4		R:2-5	R:2-6	R:2-7
	Diff. scheme offering fine level of granularity	Flexible QoS negotiation	Dynamic & adaptive resource allocation	Preferential treatment at beginning/during	Control of different comm. aspects	Access technology independent	Suitable charging model	Practicality & satisfactory performance
Adaptive framework for connection-level QoS	Yes – media type & mobility based models	No – done implicitly by system	Yes – UL CAC + adaptive BAA	Yes – admission & B.W. adaptation	Ltd.	No – mobile access only	-	Partial – Performance not fully evaluated
Class-based diff. on air interfaces	Yes – Olympic & app. category based models	Yes (potentially)	Yes – B.W. scheduling	No – during sessions only	Ltd.	No – CDMA only	-	Yes
Prioritized resource allocation in stressed networks	Yes – stress & app. category based models	Yes (via ticket servers)	Dynamic CAC – but not adaptive	No – at beginning only	Ltd.	No – mobile access only	-	Partial – Performance not fully evaluated
Traffic diff. in broadband networks	No – one serv. Diff. dimension	No – done implicitly by system	No – basic scheduling	No – during sessions only	Ltd.	No – fixed access only	-	Yes
Traffic diff. in WLANs	No – one serv. Diff. dimension	No – done implicitly by system	No	No – during sessions only	Ltd.	No – wireless access only	-	Partial – Performance not fully evaluated
3GPP IMS QoS architecture	No – one serv. Diff. dimension	No – done implicitly by system	No	Yes	Ltd.	Partial – GPRS resource reservation mech.	Basic IMS charging	Yes
TISPAN RACS architecture	No – one serv. Diff. dimension	No – done implicitly by system	Partially-dynamic config. of policies	Yes	Ltd.	Partial – fixed access policies manipulation	Partial - charging info	Yes
3GPP IMS emergency solution	Partially (coarse grained)	No – based on dialed number	No – basic queuing	No – at beginning only	Ltd.	Yes	Yes	Yes
Lucent SIP message prioritization	Possible	Ltd. – based on subscription type + no category change	Possible	Yes	Ltd.	Yes	-	Yes
ITU MLPP	Yes	Partially – no change of category	Yes - preemption	No – at beginning only	Ltd.	No – ISDN based	-	Performance not evaluated

Table 3.2 Evaluation of service differentiation solutions

This summary shows that all access level service differentiation solutions share some common drawbacks, namely: they are all access technology dependant, do not support a specialized charging model, and most of them lack consistency of preferential treatment and flexibility in terms of QoS negotiation interactions, in addition to offering limited control over the different communication aspects that could be used as means for differentiation (focusing mainly on connection admission and allocated B.W).

As for core network level service differentiation solutions, although certain aspects of their operation are access agnostic, they still lack flexibility in terms of QoS negotiation and offer limited control over the different communication aspects used for differentiation (the focus being on traffic delay/B.W. and session admission/preemption). Moreover, most of these solutions offer a coarse grained service differentiation scheme, and do not employ dynamic/adaptive resource management mechanisms nor a specialized charging model.

3.2.3 Emergency Solutions for IP-Based Networks

In this section, we discuss two categories of emergency solutions, namely: emergency solutions proposed for circuit-switched legacy networks; and emergency solutions that are being developed for IP-based networks.

3.2.3.1 Legacy Emergency Solutions

There are two main legacy emergency service architectures, namely: the wireline E911 (Enhanced 911) architecture enabling the support of emergency communications in the PSTN, and the wireless E911 architecture rendering emergency services possible in traditional mobile networks.

In the E911 architecture, 911 calls are segregated from the PSTN and switched to a network equipped with dedicated point-to-point circuits (i.e. the wireline E911 network)

whose sole function is to transmit a 911 call to a PSAP associated with the geographic location of the calling party. The E911 network also enables a PSAP to automatically correlate and display the caller's phone number with his/her associated street address – a capability that is referred to as Automatic Number Identification/Automatic Location Identification (ANI/ALI). To achieve these functions, the E911 architecture relies on five main functional entities, namely: an ALI database that maps a phone number of an address; a Master Street Address Guide (MSAG) database acting as companion to the ALI DB by specifying the address information model (i.e. spelling of street names, street number ranges, and addresses formatting standards) used; a 911 selective router acting as a central switch intelligently distributing calls to PSAPs; a PSAP which is a call center responsible for answering emergency calls; and a user making the emergency call.

This solution, which was introduced in the 1970's, has several important limitations, namely: it relies on an emergency dial-string with local significance (i.e. differing from country to country and sometimes from service to service within the same country); has limited awareness of the user situation (the information collected being limited to civic location and phone number obtained using a reverse directory mechanism); and lacks efficiency and adaptability in terms of QoS and resource management since it relies on a dedicated/redundant network for the support of emergency calls – a solution that is not efficient since dedicated resources are not used all the time, nor adaptive since extra resources cannot be obtained when needed. Furthermore, it uses a simple form of context-aware routing that takes into consideration location information only, offers limited means/forms of communication (i.e. 2-party voice calls and conventional text telephony), and introduces long call setup delays due to its the old CAMA trunk technology.

The counterpart of the wireline E911 architecture in the mobile world is the wireless E911 architecture. This architecture faces the challenge of users' mobility that renders the use of permanent addresses useless and requires the conveyance of real-time location updates to PSAPs. This issue is solved with the use of Pseudo-ANI (P-ANI) – the ESQK (Emergency Services Query Key) and the ESRD (Emergency Services Routing Digits) being forms of P-ANI. In fact, PSAPs equipped to handle P-ANIs can distinguish wireline from wireless calls and can use those P-ANIs to query the ALI DB for non traditional location information (e.g. ID of cell servicing the user, and users' geographic coordinates). It should be noted that two new entities - the Position Determination Equipment (PDE) and the Mobile Positioning Center (MPC) - are used for the collection and formatting of geographic location information using triangulation techniques (e.g. the Angle of Arrival (AOA) and Time Difference of Arrival (TDOA) techniques) and its update in the ALI database. This architecture mostly has the same drawbacks as the wireline E911 architecture, including: the lack of efficiency/adaptability in terms of resource management (due to the reliance on fixed dedicated resources); context-awareness that is limited to location and radiolocation mechanisms; and the support of poor means/forms of communication that are limited to two-party voice calls. However, unlike its wireline counterpart, this architecture uses an international emergency calling number ('112' for GSM networks) and achieves better performance in terms of call setup time due to its reliance on a newer technology (i.e. SS7).

3.2.3.2 Emerging IP-based Emergency Solutions

The support of emergency services in IP-based networks has been recently investigated and some IP-based emergency solutions have been proposed, including: the IETF ECRIT

framework [52] aiming at enabling emergency communications support in IP telephony; and the 3GPP IMS emergency service architecture [49] focusing on emergency services support in IP-based mobile networks.

Figure 3.9 illustrates the ECRIT architecture [52] that relies on five main functional entities: the caller's User Agent (UA) used to make the emergency call; a configuration server providing the UA with its address and other configuration information (e.g. location information) and possibly acting as SIP registrar; an ESRP (Emergency Service Routing Proxy) server making routing decisions based on PSAP state and location of caller; a LoST server processing requests for location to PSAP URI mapping; and a PSAP constituting the destination of emergency calls.

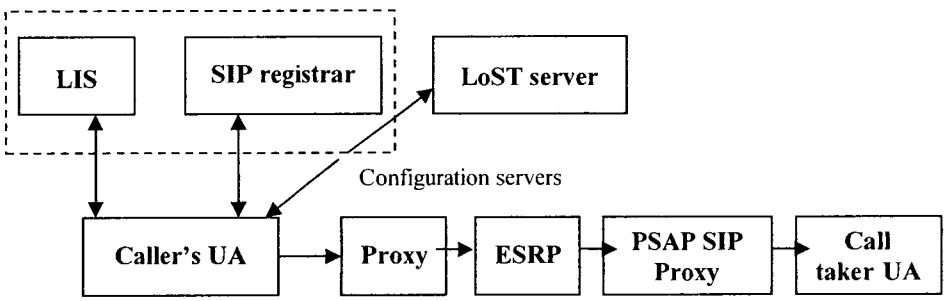


Figure 3.9: The ECRIT architecture

In this architecture, the UA generally either has location information configured manually, has an integral location measurement mechanism (e.g. GPS), or runs a location configuration protocol (e.g. DHCP) to obtain location information from the access network. This UA performs an initial LoST location to PSAP URI query to learn a backup URI, and when the user decides to make an emergency call, it will attempt to refresh its location as well as the PSAP URI it has. After that, an INVITE (containing the location information) is forwarded to its proxy that forwards it to the ESRP. This last routes it to the most appropriate PSAP based on the caller's location, the PSAP state, and other state information. Finally, a proxy in the PSAP chooses an available call taker and extends the

call to its UA.

This architecture relies on an emergency URN called 'SOS' that is defined as part of the solution, takes into consideration different pieces of information (e.g. location, service type, and terminal's media capabilities) to influence routing decisions, and support multimedia emergency calls. Nevertheless, the context-awareness aspects of this solution are still limited to location information (determined using different mechanisms – manual entry, end-system measured, access network measured), in addition to the QoS and resource management aspects of emergency calls that are not tackled in this architecture.

The IMS emergency service architecture [49] has been presented in chapter 2. Like the other solutions, this solution has several limitations. In fact, it has a restricted view of the caller's situation that is limited to location information (mainly geo-location and location ID obtained from the access network), relies on a simple form of queuing (i.e. a static resource management strategy) at the P-CSCF level to achieve emergency sessions prioritization as discussed in section 3.2.2.2; uses a simple form of context-aware routing (i.e. location-aware routing); in addition to presenting the call taker with a single piece of contextual information about the caller (i.e. his/her location). Finally, in terms of means/forms of communication, this solution offers multimedia support but is limited to two-party sessions - no specific solution for the support of IMS multiparty emergency sessions existing.

3.2.3.3 Summary of Evaluation

Table 3.3 presents a summary of the evaluation of the different emergency solutions presented.

Solution/ Req.	R:3-1	R:3-2	R:3-3	R:3-4	R:3-5	R:3-6	R:3-7
	Universal emergency calling number	Acquisition/mgmt of caller's context information	Preferential treatment of emergency calls via dynamic/adaptive resource mgmt	Context-aware routing	Rich emergency communication models	A wide view about the caller's situation	Satisfactory call setup time
Emergency services in PSTN	No – number with local significance	Ltd. to civic location & phone #	No – dedicated fixed resources	Partially – location based	No – 2-party voice & text	Ltd. to caller's address & phone #	No – old CAMA tech.
Emergency services in traditional mobile networks	Yes – international emergency number	Ltd. to spatial location & phone #	No – dedicated fixed resources	Partially – location based	No – 2-party voice	Ltd. to caller's location & phone #	Yes – SS7 tech.
Emergency services in IP telephony	Yes – emergency URN	Ltd. to location info	-	Yes	Partially – multimedia sessions	Ltd. to location info	Not yet studied
Emergency services in IP-based mobile networks	Yes – emergency service identifiers	Ltd. to location info	No – basic queuing strategy	Partially – location based	Partially – multimedia sessions	Ltd. to location info	Not yet studied

Table 3.3 Evaluation of emergency solutions

This summary shows that all the existing emergency solutions present limitations with respect to three aspects, namely: QoS and resource management; context-awareness and service personalization; and the emergency communication models used.

In terms of QoS and resource management, none of these solutions relies on a dynamic/adaptive resource allocation strategy, thus lacking efficiency and adaptability. Therefore, this aspect should be better investigated to define appropriate QoS profiles for emergency sessions and elaborate needed resource management techniques providing emergency sessions with preferential treatment, in a resource efficient and adaptive manner.

Concerning context-awareness and service personalization, currently a limited range of contextual information (mainly location) is used to support emergency operations.

Exploiting a richer set of contextual information could lead to enhanced emergency services and more efficient emergency operations. For instance, presenting the call taker with a wider view about the caller's situation could enable a better assessment of the situation and the provision of better help to the user, while context-aware routing (i.e. routing based on the user's situation) could lead to more efficient and targeted help. Finally, the means/forms of communications supported in current solutions are limited (mainly two-party voice calls). Richer and more sophisticated forms of communication (i.e. multimedia multiparty communications) could be supported to help in situation assessment and better coordination of rescue efforts.

3.3 Conclusions

In this chapter, we derived requirements related to the issues of context acquisition/management, service differentiation, and enhanced emergency support in 3G networks, and reviewed the related work in light of these requirements.

Two categories of information acquisition/management solutions were reviewed: WSN integration solutions focusing on the information acquisition aspects of the problem and information management solutions complementing these approaches by addressing issues related to the management and usage of contextual information in the core network. The evaluation highlighted the limitations of these two categories of solutions and showed that the 3GPP presence framework is the most promising of these solutions since it satisfies most of our requirements.

Related to service differentiation in 3G networks, two groups of solutions were evaluated, namely: solutions operating at the access network level; and solutions operating at the core network level. The evaluation of the access-level solutions showed that they have several

limitations such as access technology dependence, the lack of consistency of preferential treatment and flexibility in terms of QoS negotiation, in addition to limited control over the different communication aspects used for differentiation. As for core network level solutions, they also lacked flexibility in terms of QoS negotiation and offered limited control over the different communication aspects, in addition to mostly offering coarse-grained service differentiation schemes and not employing dynamic/adaptive resource management strategies nor specialized charging models.

Finally, in relation to enhanced emergency communication support in 3G networks, we reviewed both legacy emergency solutions and emergency IP-based emergency solutions. The evaluation showed that all these solutions present limitations with respect to three main aspects, namely: QoS and resource management; context-awareness and service personalization; and the emergency communication models used.

Chapter 4

A Presence-Based Approach for Context Information Acquisition and Management in the IMS

This chapter proposes a solution for context information management in the IMS, as means to ensure its availability for future usage in the 3G network. This solution leverages and extends the 3GPP presence framework. The chapter starts by discussing our motivations for using a presence-based approach. This is followed by a presentation of the proposed architecture, and an elaboration of the different information management and business issues related to it. We end the chapter with a presentation of information exchange scenarios illustrating the system's operation, before drawing our conclusions.

4.1 Motivations for a Presence-Based Approach

There are two possible approaches for tackling the problem of context information management in the IMS. The first is the revolutionary approach, which necessitates the design of a new information management framework that is specialized in the handling of contextual information. The approach implies the introduction of new functional entities to the IMS architecture and their design from scratch (including the definition of the needed information model to represent contextual information, as well as the protocols to be used on the inbound and outbound interfaces, in addition to the support functions needed for information exchange). The second approach is the evolutionary approach in which we take an existing 3G information management framework and extend/refine it to enable the management of contextual information. In this work, we chose the evolutionary approach since it enables the reuse of existing components and concepts, and builds on the installed basis thus reducing the cost of migration to the new solution.

The existing framework that will be extended should be flexible and should be able to satisfy the requirements related to the problem of context management in 3G networks. Among the existing information management frameworks discussed in chapter 3, the 3GPP presence framework seemed the most promising since it is flexible/extensible and satisfied most of our requirements. Furthermore, we note that the concept of context represents a generalization of the concept of presence, which it extends in two aspects: the scope of the information handled and the types of entities concerned. While presence is restricted to information conveying the ability and willingness for communication, context includes any type of relevant information. Moreover, context applies to any type of entity, unlike presence information that only applies to users (and their services and devices). This relationship between the two concepts and the flexibility and potential of the 3GPP presence framework makes it a prime candidate for the management and dissemination of contextual information within the IMS.

4.2 An IMS Context Management Architecture

Ensuring the availability of contextual information in the 3G network involves two main aspects, namely: enabling the interaction of the IMS with physical/logical sensors to collect the needed information; and effectively managing and disseminating this information (to interested entities) within the network. In this section, we present the architecture we propose to handle these aspects. We start by describing the functional entities and interfaces forming this architecture, and then present an example of a business model that we propose to enable the practical deployment of the solution.

4.2.1 Functional Entities

Figure 4.1 depicts the proposed IMS context management architecture. In this architecture, we leverage an existing role (i.e. the PEA role) that was defined as a place holder in the

3GPP presence architecture, and assign this role to the WSN gateway acting as interworking unit between WSNs (i.e. external information sources) and the 3G core network. To achieve this role, the WSN gateway conveys properly formatted sensory information (captured by WSNs) to the PS, via the presentity presence proxy. Furthermore, it interacts with the PS to manage subscription policies. Those policies are used by the PS to install filters that determine which watchers are allowed to access the information related to a certain contextual entity, thus preserving information privacy. In addition to information management functions, the WSN gateway carries other support functions needed to achieve WSNs/IMS interworking, such as: IDs and Protocols mapping, IMS registration, security, and capabilities publication.

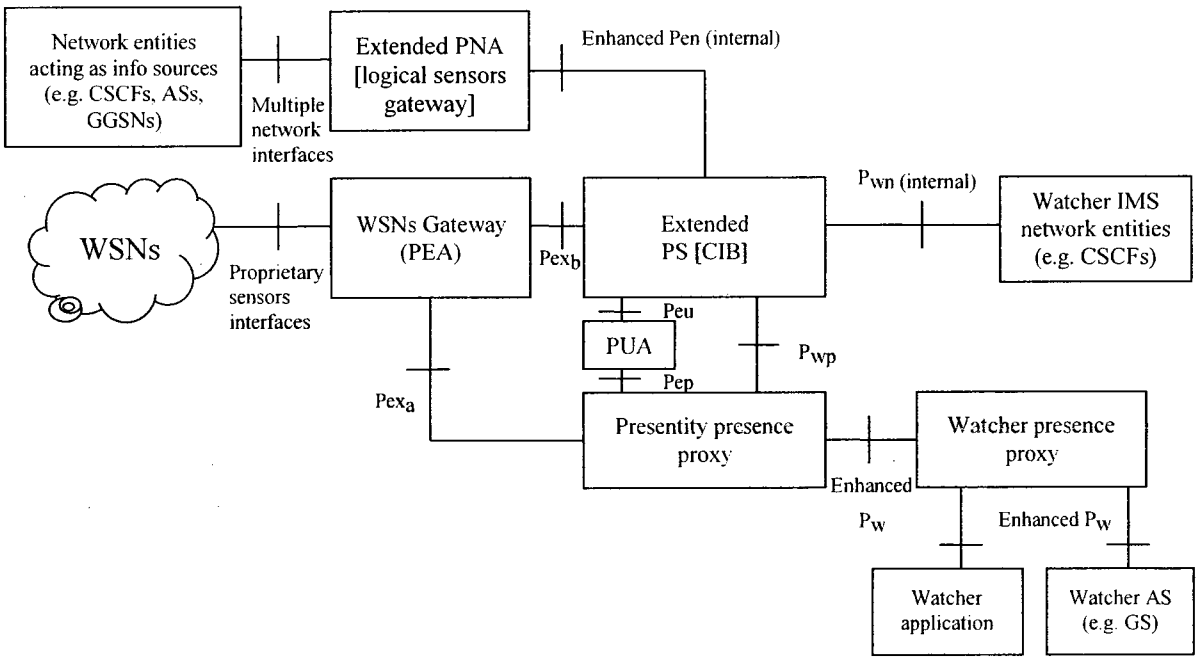


Figure 4.1: A Presence-based architecture for context management in the IMS

Moreover, some enhancements are made to existing entities, namely: the PNA, the PS, and presence proxies. The PNA is evolved into a logical sensors gateway enabling the interaction with various core network entities (e.g. CSCFs, ASs, edge routers, resource mgmt nodes, and network mgmt nodes) for the collection of network status information

(e.g. state of ongoing sessions and network capacity) and user status information (e.g. IMS registration status) that is conveyed to the PS. Figure 4.2 illustrates the relationship between the extended PNA and different network entities that could act as information sources, and pinpoints the interfaces/protocols that can be used for their interactions. As shown in the figure, the PNA and the PS may be co-located since their relationship involves local exchange of network status information.

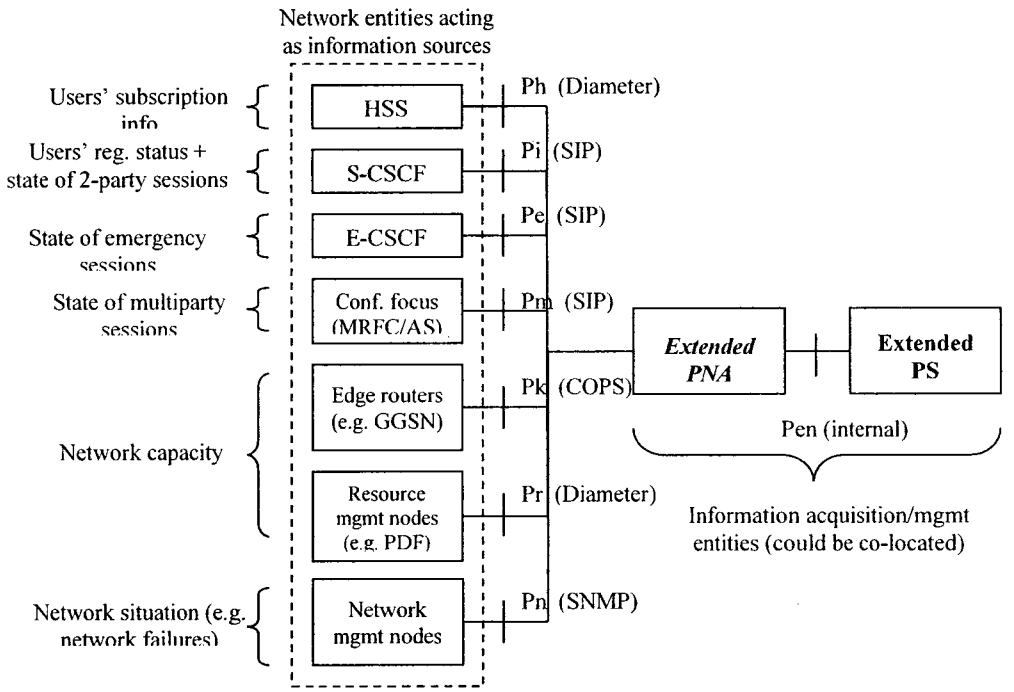


Figure 4.2: Core network entities acting as context information sources

As for the PS, it is now extended to play the role of a Context Information Base (CIB) managing a wide range of contextual information provided by physical/logical sensors as well as end users. To achieve this role, the extended PS supports a set of functions, including: the reception of contextual information that is published by presence user/network/external agents and the triggering of information publication (when needed); the processing and formatting of the received information; the management of subscriptions from internal (trusted) information watchers and external information watchers; as well as the control of access to information via the enforcement of

subscription authorization policies. Finally, some enhancements were also made to the presence proxies, such as: the identification and charging of different types of contextual entities and the routing of information exchange traffic based on their identities; in addition to supporting publication triggers and capability publications related interactions.

4.2.2 Interfaces

In terms of inbound interfaces (i.e. interfaces between information providers and the extended PS), two interfaces are used: the Pex and the Pen interfaces. We divide the Pex interface into two sub-interfaces: Pex_a and Pex_b. Pex_a is used for the indirect exchange of sensory information between the WSN gateway and the extended PS, via a presentity presence proxy. This indirect interaction between the WSN gateway and the PS is motivated by the fact that several support functions (e.g. identification, charging, and security) are needed for information exchange. Some of these functions are already supported by the presentity presence proxy, and therefore could be leveraged by including the proxy as intermediary node between the gateway and the PS. As for the Pex_b interface, it is used for direct interactions between the gateway and the PS, in relation to the management of subscription policies (for information access control). The Pen intra-operator interface, which represents an enhancement of the existing 3GPP interface, enables the direct exchange of network status information and user registration status information between the extended PNA and the extended PS.

In addition to the inbound interfaces, figure 4.1 depicts two outbound interfaces (interfaces between the PS and watchers): the Pw and the Pwn interfaces. Pw is an enhancement of the existing 3GPP interface that enables end user applications and IMS application servers to access context information managed by the PS, via presence proxies. We note that the Pw interface existing between watcher applications and the watcher presence proxy

corresponds to the IMS Gm interface, and is thus used for IMS registration and security related functions in addition to information access interactions with the PS. In contrast, the Pw interface between trusted applications servers (acting as watchers) and the PS corresponds to the IMS ISC interface and is only used for information exchange related operations. The interactions on both interfaces trigger the generation of charging information related to information exchange (by presence proxies). In the case of core network entities acting as watchers, security/trust and charging issues are not relevant, since those entities are owned and trusted by the network and would not be charged for access to information used in their operation. We give as example an E-CSCF leveraging user contextual information to route an emergency call to the most appropriate PSAP. This E-CSCF would not need to be authenticated/authorized by the network and won't be charged for its access to information. This motivated the definition of the Pwn interface, as a new intra-operator interface, enabling network entities acting as watchers to get direct access to context information from the PS (without triggering the generation of charging records).

4.2.3 The Proposed Business Model

A business model describes the different parties (or business roles) involved in service provisioning and their relationships/interactions [76]. In the 3G networking environment, the IMS business model defines three main business roles: the end-user; the service provider; and the network operator. The end-user is an entity that has a service usage agreement with the network operator and which owns the UE needed to invoke/use services. The service provider owns the application servers (ASs) hosting and executing value-added services; while the network operator owns the 3G network infrastructure on which ASs rely for session control and traffic transportation.

To enable the integration of physical and logical sensors in the IMS, we propose the extension of the basic IMS business model with new roles and interactions. Figure 4.3 illustrates the proposed model, in which the new roles/interactions are shown in italics and dotted lines.

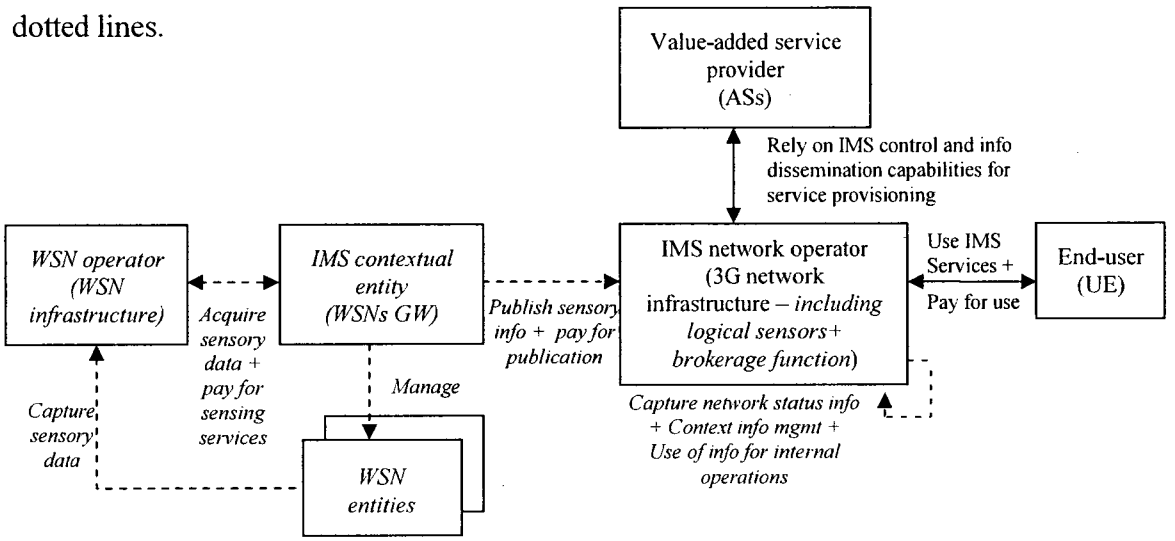


Figure 4.3: Proposed business model for sensors-enabled IMS environment

To enable the interaction of the IMS with WSNs (i.e. physical sensors), three new roles are proposed, namely the roles of: WSN operator; WSN entity; and IMS contextual entity. The WSN operator is the owner of the WSN infrastructure and the provider of information sensing services. WSN entities are the entities whose information is being captured by the WSN infrastructure and conveyed to the IMS. Those entities could be objects, persons, or places, and the sensors (forming a part of the WSN infrastructure) are attached to them or placed in their proximity (e.g. RFIDs attached to merchandize, biometric sensors attached to persons, and environmental sensors spread in a location). Since WSN entities may not be legally liable entities (e.g. objects and places whose information is being captured) or may not be 3G network subscribers (e.g. children whose location is being tracked by their parents), their interaction with the IMS becomes difficult. To solve this issue, we introduced the role of IMS contextual entity as a legally liable entity acting as intermediary between the IMS network operator and the WSN entities. Each IMS contextual entity

manages one or several WSN entities (e.g. an organization managing a set of locations, equipment, and employees; or a family head managing a household, children, and house supplies), and publishes (and pays for the publication of) information on their behalf in the IMS. To achieve this role, the IMS contextual entity owns one or several WSNs gateways and has separate service agreements (and charging relations) with both the WSN operator and the IMS network operator. Those WSNs gateways are used for the collection of the captured sensory information from (certain elements in) the WSN infrastructure and the publication of this information (after proper processing and formatting) in the IMS. To enable the dynamic discovery of available WSNs gateways by the IMS, the 3G network infrastructure is enhanced with a capability publication/discovery mechanism, thus implicitly acting as information broker between the WSN and the IMS network operators. Moreover, to account for the interaction of the IMS with logical sensors supplying it with network status information, the function of the 3G network infrastructure (owned by the IMS network operator) is enhanced to include: the local capturing of network status information; the management of this information, in addition to sensory information captured by WSNs; the usage of this information for internal network operations; and the dissemination of this information to other interested entities (e.g. user applications or third-party owned ASs). Similar to all business models, one entity could play several business roles at the same time. For instance, the IMS operator could also play the role of WSN operator (i.e. the IMS operator owning the WSN infrastructure). However, in the rest of this thesis, we assume that each role is played by a separate entity.

4.3 Identification and Charging Issues

The ability to identity users/entities accessing and using the network resources is central to any telecommunication system. In fact, users/entities identification is important for two

reasons: 1) to be able to route traffic to its correct destination based on those identities (e.g. directing a call to the appropriate user); and 2) to be able to charge concerned users/entities for the utilization of the network resources.

In this section, we propose identification/charging schemes for two cases: the first involving the IMS interaction with external entities (i.e. WSN gateways) conveying sensory information captured by WSNs (i.e. the WSNs/IMS integration case); while the second involves local interactions between IMS network entities for the collection of network status information (i.e. the logical sensors/IMS integration case). Those schemes were specifically designed the support the operation of the proposed business model.

4.3.1 An Identification Scheme for the WSNs/IMS Integration Case

To deal with the identification of IMS contextual entities and WSN entities in the IMS, we propose a two-level identification scheme that is illustrated in figure 4.4.

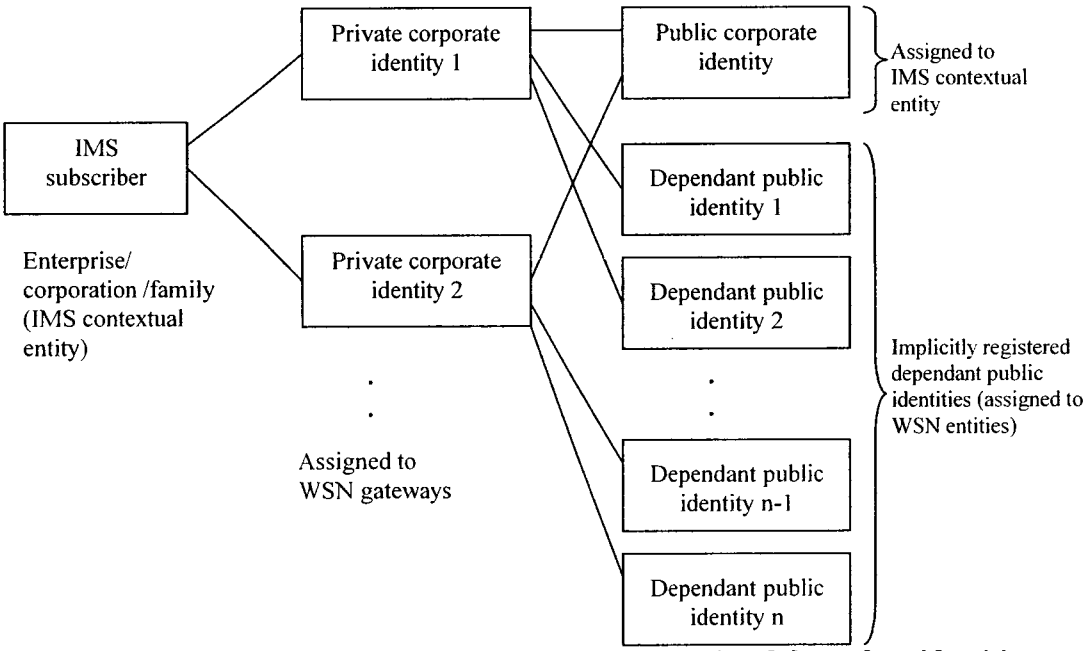


Figure 4.4: IMS corporate identity and its associated dependant identities

This scheme is based on the novel concept of IMS corporate identity and its associated dependant identities, which are respectively allocated to the IMS contextual entity and its

managed WSN entities as follows: IMS contextual entities (e.g. game providers, enterprises, and corporations) would have IMS corporate accounts, and thus be known as IMS subscribers - each corporation/enterprise would then be assigned one public corporate identity and one or more private corporate identities. Furthermore, separate (but dependant) public identities are created for the WSN entities managed by the IMS corporate client. We also assume the existence of a corporate registration process (implemented by the WSN gateway owned by the corporate user), which is responsible for registering the IMS corporate client and all its dependant identities with the network. This process could be optimized using the existing concept of implicitly registered public identities, which requires only one registration using the main identity and results in the registration of this identity and all its associated identities.

It should be noted that corporate public identities are only used during the IMS registration phase (during which the dependant identities associated to those corporate identities are implicitly registered), while the actual WSN entities identities are used for information exchange related interactions. Figure 4.5 illustrates a possible SIP URI scheme that can be used for the representation of public corporate and dependant identities. As for private identities, they are used for identification and authentication purposes only, and are not know by the corporate user. Private identities take the form of network access identifiers, such as username@operator.com, and each private identity is associated with a SIM card on which it is stored. Since there may be several WSN gateways publishing different types of information about the objects/places/persons managed by each organization/family, it may be necessary to allocate several private identities for each corporate client. In this case, each WSN gateway will be assigned a SIM card (associated with one of the private

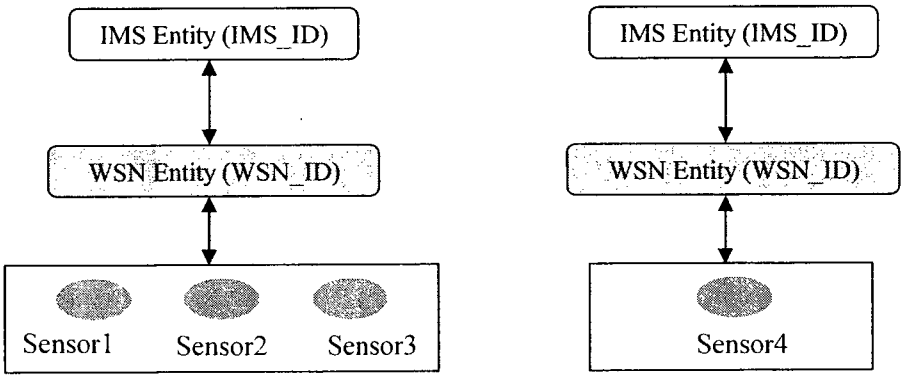
corporate identities) for the purpose of its identification and registration to the IMS. This is also needed for the authentication/authorization of WSN gateways by the network, and the establishment of the needed security associations, as we will elaborate in the section 4.3.1.

Public corporate identity:
CompanyName.Branch.Country@operator.com or
FamilyName.City.Country@operator.com

Public dependant identity:
EntityID.CompanyName.Branch.Country@operator.com or
EntityID.FamilyName.City.Country@operator.com
Examples:
- Hospital room status:
  ER201.RoyalVicHospital.Montreal.Canada@operator.com
- Inflatable object in a pervasive game:
  IFO100.GameProviderX.MontrealDownTown.Canada@operator.com
- Child's location:
  INF-John.SmithFamily.Westmount.Canada@operator.com

Figure 4.5: SIP URI scheme for public corporate identities and dependant WSN entities identities

Since the information related to a certain WSN entity can be captured by one or several physical sensors, there is a need to map the sensor IDs to the IMS IDs, at the WSN gateway level. Figure 4.6a illustrates this mapping in which each sensor or group of sensors is associated with a WSN-ID, identifying the entity (whose information is being captured) in the WSN world. This WSN-ID is further mapped to an IMS ID (identifying the entity in the IMS world). Examples of mapping tables illustrating these two levels of mappings are shown in figures 4.6b and 4.6c.



a)

WSN_ID	IMS_ID	Entity type	Entity Description
we1	IFO100.GameProvider.Montreal.Canada@operator.com	object	Inflatable game object
we2	Room4060.Ericsson.Montreal.Canada@operator.com	place	Ericsson Office Room 4060
we3	Player1.GameProvider.Montreal.Canada@operator.com	person	Game player
we4	Player2.GameProvider.Montreal.Canada@operator.com	person	Game player

b)

WSN_ID	SENSOR_ID	LOCATION	TEMPERATURE	LIGHT	SOUND_LEVEL
we3	sensor1	TRUE	FALSE	FALSE	FALSE
we3	sensor2	TRUE	FALSE	FALSE	FALSE
we3	sensor3	TRUE	FALSE	FALSE	FALSE
we2	sensor4	FALSE	TRUE	FALSE	FALSE

c)

Figure 4.6: Mapping physical sensors IDs to IMS IDs: a) Two-level mapping scheme; b) one-to-one mapping table from WSN-ID to IMS-ID; c) one-to-many mapping table from Sensor-ID to WSN-ID

4.3.2 An Identification Scheme for the Logical Sensors/IMS Integration Case

To deal with the identification of the contextual entities whose information is being captured by logical sensors (the network and its users in this case), we propose a simple identification scheme that is illustrated in figure 4.7. In this scheme, the network and its users (acting as contextual entities) are considered to be managed by the IMS network operator, and their identities are used for information exchange related interactions (e.g. publications and subscriptions).

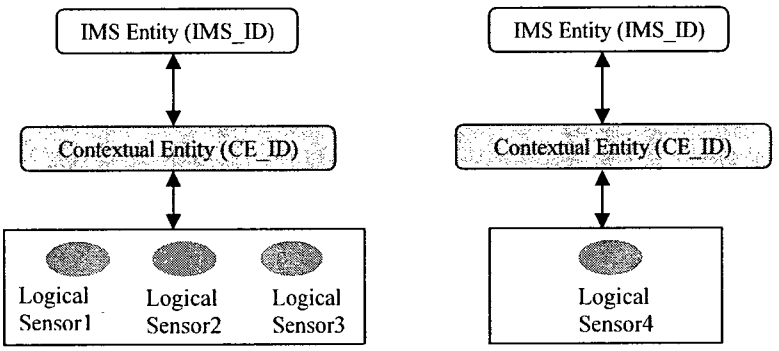
Identity of network acting as contextual entity:
NetworkID.DeploymentArea.Country@operator.com
Example:
- Net202.MontrealArea.Canada@operator.com

Identity of network user acting as contextual entity:
UserID.HomeNetworkDeploymentArea.Country@operator.com
Example:
- Alice_Jones.TorontoArea.Canada@operator.com

Figure 4.7 Identification scheme for networks and users acting as contextual entities in a logical sensors/IMS integrated environment

Similar to the case of WSN entities, each contextual entity in the logical sensors/IMS environment is associated with one or more logical sensor capturing its information, and is

assigned a CE-ID (Contextual Entity ID) identifying it in the logical sensing world. This CE-ID is then mapped to an IMS-ID, as shown in figure 4.8.



a)

CE_ID	IMS_ID	Entity type	Entity Description
Ce1	Alice_Jones.MontrealArea.Canada@operator.com	person	Network user
Ce2	NET202.TorontoArea.Canada@operator.com	network	Telecom network

b)

CE_ID	LOG_SENSOR_ID	REG_STATUS	SESSION_STATUS	NET_CAP	NET_SITUATION
Ce1	S-CSCF1	TRUE	FALSE	FALSE	FALSE
Ce2	S-CSCF2	FALSE	TRUE	FALSE	FALSE
Ce2	AS101	FALSE	TRUE	FALSE	FALSE
Ce2	E-CSCF1	FALSE	TRUE	FALSE	FALSE
Ce2	GGSN3	FALSE	FALSE	TRUE	FALSE

c)

Figure 4.8: Mapping logical sensors IDs to IMS IDs: a) Two-level mapping scheme; b) one-to-one mapping table from CE-ID to IMS-ID; c) one-to-many mapping table from Logical-Sensor-ID to CE-ID

4.3.3 Charging for Context Information Management in the IMS

In the 3GPP presence framework, charging is performed on the inbound interface (i.e. interface between information sources and the PS) for information publication, as well as on the outbound interface (between the PS and information consumers) for access to published information/information updates, and access to watchers information. For information publication, it is the entity whose information is being published that is responsible for paying for the publication. For instance, if a presence user agent publishes user status information, it is the user who is billed for that. For subscriptions/notifications.

it is the entity who subscribed to the information (e.g. an application user subscribing to the presence information of another user) who settles the bill. In this work, we followed a similar philosophy as illustrated in figure 4.9, such that in the case of WSN/IMS integration in which the context information published is related to WSN entities, it is the corporate user to which these entities are linked that should pay for the publication of their related information. We assume that there is no charging relation between the WSN and the IMS operators.

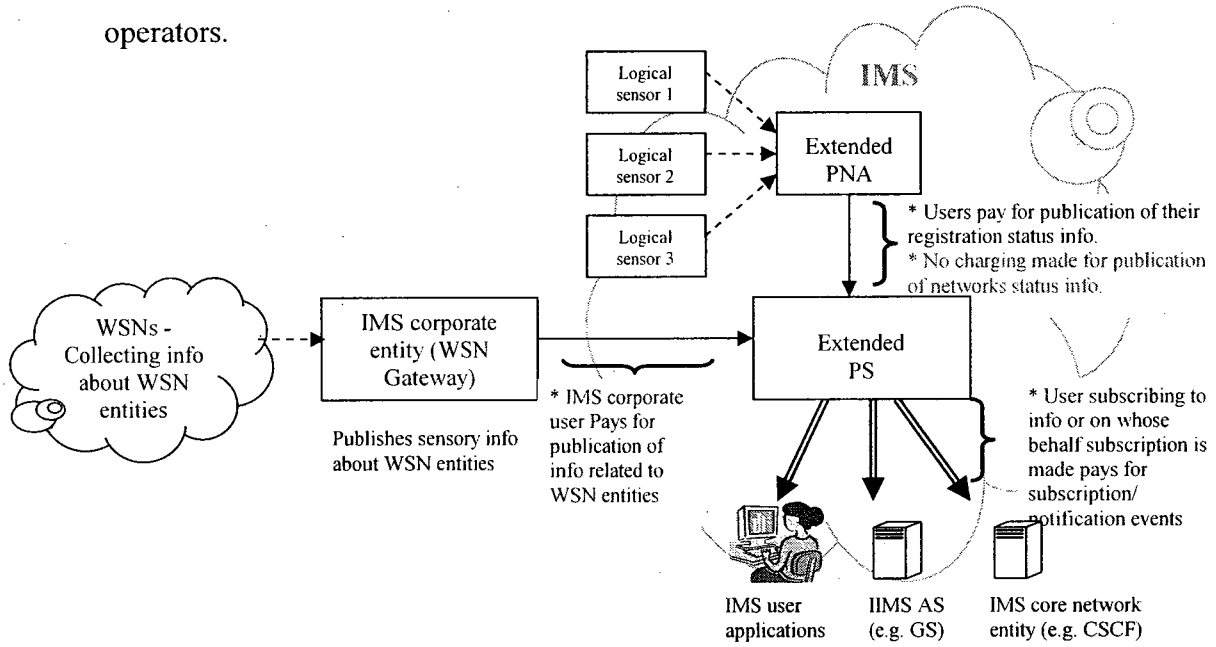


Figure 4.9: Charging for context information management in the IMS

In the case of logical sensors/IMS integration, if the information published by the PNA is related to a user (e.g. a user's registration status); it is the user who is charged for the publication of this information. The only exception is in the case where network status information is being locally published in the network. In this case, no charging is needed since the entity to whom this information relates (i.e. the network) is owned by the network operator. In terms of outbound interface related interactions (i.e. subscriptions/ notifications), it is the user who subscribed to the information or on whose behalf the subscription is made, that is charged for subscriptions/notifications events. For instance, in

the case of an end-user application acting as watcher, it is the user of this application that is charged for information consumption. In the case of a watcher AS subscribing on behalf of a user (to whom it is providing a service), it is the user who is responsible for the bill.

4.4 Security, Information Access Control, and Service Discovery

In this section, we discuss other support functions that are needed to regulate the interaction between the 3G network and information sources/consumers, namely: security; information access control; and service discovery.

4.4.1 Security and Information Access Control

Two levels of security are used in the IMS, namely: access security [77] dealing with the authentication/authorization of end users and the protection of the traffic exchanged between IMS terminals and the network; and network security [78] dealing with inter-domain and intra-domain traffic protection between network nodes.

The existing inter-domain IMS security mechanism [78] can be directly used to support the IMS interaction with logical sensors, since this case does not introduce new roles but rather involves local information exchange between network entities. On the other hand, since the IMS interaction with external information sources (i.e. WSNs) introduces new roles in the IMS business model, this necessitates the refinement of the existing IMS access security mechanisms.

In this section, we focus on this refinement that is achieved as follows: Each WSN gateway (owned by a corporate user) is assigned a SIM card, storing the following information: the public corporate identity; one of the private corporate identities; the public identities of the WSN entities managed by the corporation (i.e. the dependant identities); the home network domain URI; and a long term secret that is shared offline between the corporate user and

the operator and is simultaneously stored in the gateway's SIM card and the HSS. This information is used during the IMS registration phase to carry three security related operations (i.e. the authorization of the IMS corporate user, the mutual authentication between the WSN gateway and the network, and the establishment of security associations between them), in addition to other registration related operations such as the binding of the public/private ID pair to a physical contact address (the gateway's address) and the implicit registration of dependant public IDs. Figure 4.10 gives an overview of the different operations performed during WSN gateways' registration to the IMS.

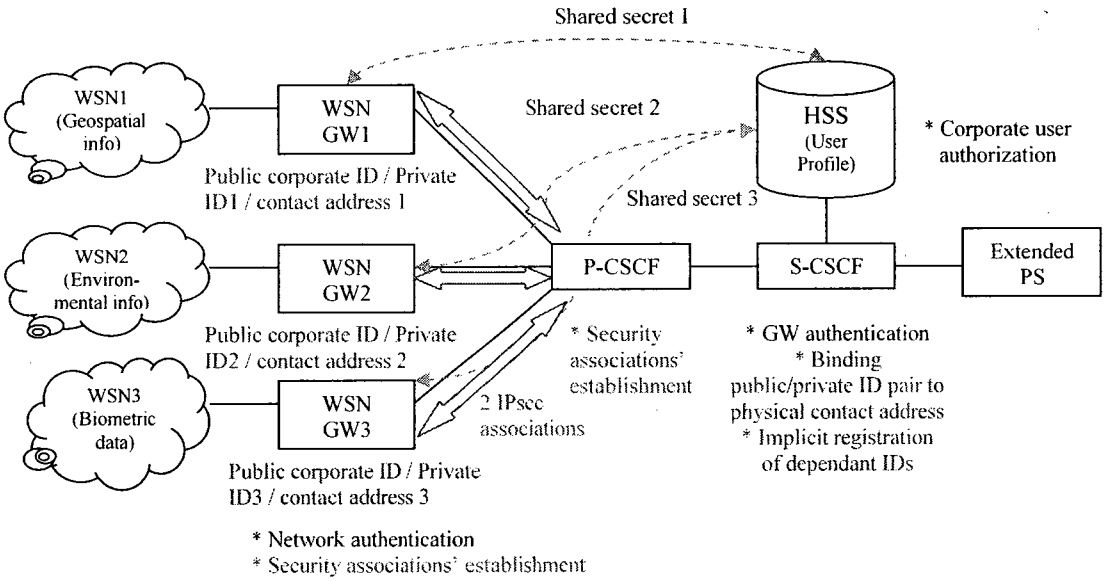


Figure 4.10: Overview of operations performed during WSN gateways registration to the IMS

Figure 4.11, details how the different operations presented in the previous figure are concretely realized, by presenting an IMS corporate user registration scenario. The scenario starts when the gateway (owned by the corporate user) sends a SIP REGISTER message (containing the public/private corporate IDs and its contact address) to its P-CSCF that was discovered previously. The P-CSCF forwards the message to an I-CSCF in the corporate user's home network. The I-CSCF contacts the HSS, which authorizes the corporate user to access the network resources and returns the address of the S-CSCF allocated to the user, to

which the REGISTER message is forwarded. After receiving the REGISTER message and downloading the user authentication vectors from the HSS, the S-CSCF generates a challenge (based on the shared key and some other parameters extracted from the authentication vectors) and sends it in a 401 SIP message to the gateway. Based on parameters extracted from the challenge, the gateway is able to authenticate the network, and generates on its turn a response to the challenge which it includes in a second SIP REGISTER message routed to the S-CSCF. The S-CSCF then compares this response to the expected answer contained in the authentication vectors, and if they match, it authenticates the gateway and completes the registration by binding the public/private corporate user ID to the provided contact address. Afterwards, the S-CSCF sends a Diameter SAR to the HSS to inform it of the user registration status, and to download the rest of the user profile, containing among other things the list of implicitly registered dependant public IDs, which are then automatically registered by the S-CSCF. Finally, the S-CSCF sends a SIP 200 OK message indicating the success of the registration to the gateway. It should be noted that the P-CSCF and the gateway also establish two IPSec security associations enabling the bi-directional exchange of secure traffic between them. The parameters used for those associations (i.e. the encryption mechanisms supported, the associations' identifiers, and the ports used) are negotiated during the initial SIP REGISTER message and its subsequent SIP 401 response. It should also be mentioned that although several WSN gateways may simultaneously register using the same public corporate ID, the (de)registration of one gateway does not affect the registrations status of the other gateways, since each registration relates to a particular private ID/contact address pair that remains unique among different gateways, as shown in figure 4.10.

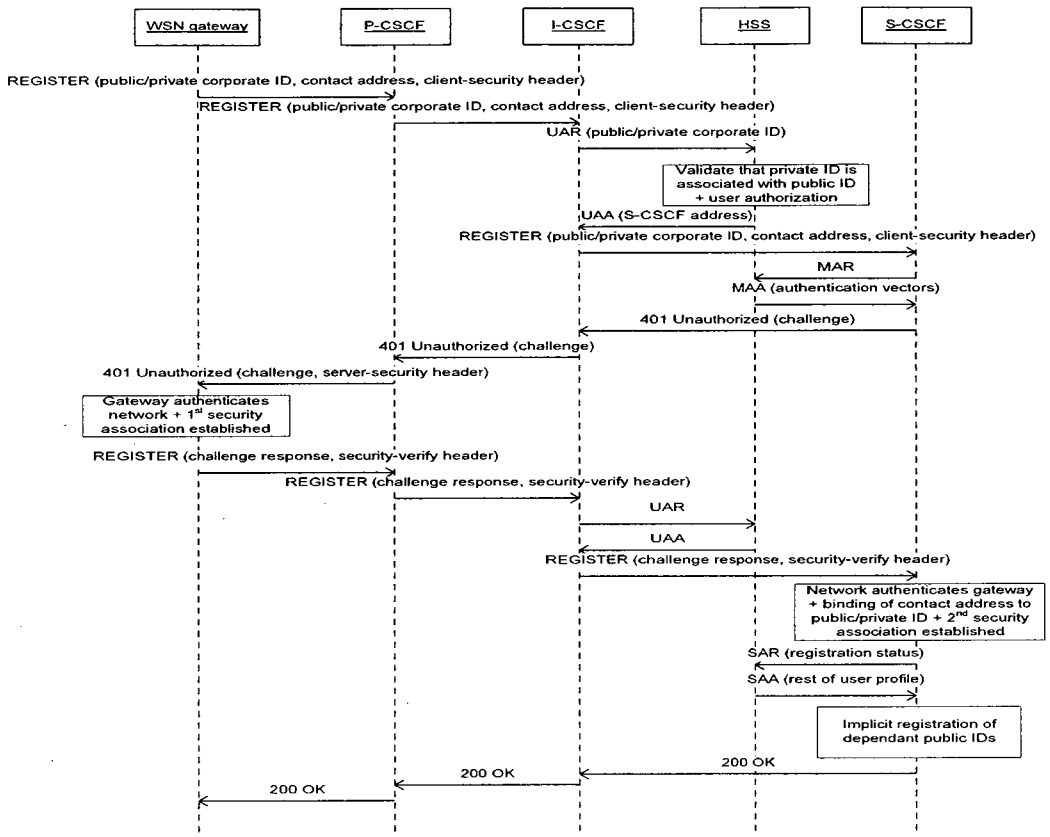
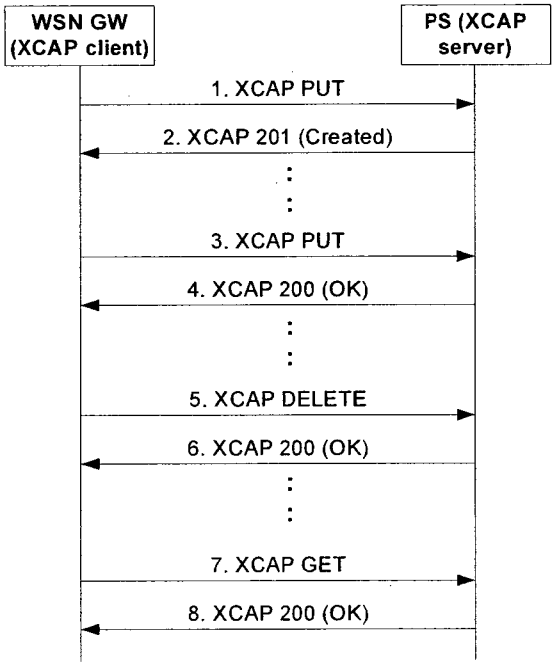


Figure 4.11: IMS corporate user registration scenario performed by WSN gateway

In addition to information confidentiality, information privacy is an important concern especially when dealing with contextual information which may be of sensitive nature (e.g. location info, biometric data). As mentioned previously, the 3GPP presence architecture preserves information privacy by the means of subscription authorization policies which are set by presence agents, in the PS. The XML Configuration Access Protocol (XCAP) [79] (a list manipulation protocol) is proposed to enable presence agents to manipulate subscription authorization policies. We use the same protocol over the Pex_b and the Pen interfaces to control access to contextual information. Figure 4.12a shows examples of interactions between a WSN gateway (acting as XCAP client) and the PS (acting as XCAP server) over the Pex_b interface. for the manipulation of subscription authorization policies related to the access of sensory information, while figure 4.12b shows the contents of the

first message used for the creation of the policy. In this policy, two medical specialists (doctor1 and nurse2) are allowed access to Bob's physiological data.



a)

```
PUT http://xcap.home1.net/services/pres-rules/users/Bob/ ps.xml
HTTP/1.1
User-Agent: WSN Gateway
Date: Sun, 08 Mar 2009 10:20:00 GMT
Content-Type:application/auth-policy+xml
Content-Length: (...)
<?xml version="1.0" encoding="UTF-8"?>
  <ruleset xmlns="urn:ietf:params:xml:ns:common-policy"
    xmlns:sc="urn:ietf:params:xml:ns:pidf:caps"
    xmlns:pr="urn:ietf:params:xml:ns:pres-rules"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:ietf:params:xml:ns:pres-rules">
    <rule id="klbds565778">
      <conditions>
        <identity>
        <id entity="Doctor1.MontrealArea.Canada@home1.net"/>
        </identity>
        <identity>
        <id entity="Nurse2.MontrealArea.Canada@home1.net"/>
        </identity>
      </conditions>
      <actions>
      <pr:sub-handling>allow</pr:sub-handling>
      </actions>
      <transformations>
        <pr:provide-information>
          <pr:all-physiological-info/>
        </pr:provide-information>
      </transformations>
</rule> </ruleset>
```

b)

Figure 4.12: Use of XCAP for the manipulation of subscription authorization policies: a) a WSN gateway manipulating context authorization policy on a PS; b) Contents of the context authorization policy

4.4.2 Context Sources Discovery Alternatives

In an evolving sensors-enabled IMS environment, various physical/logical sensors, providing different types of information (with different levels of accuracy and granularity), may be continuously deployed. Therefore, there is a need for a mechanism enabling the discovery of available gateways and their capabilities by the PS. This would permit the PS to choose the most suitable information sources for the consumers' needs.

The PS can discover the available gateways either dynamically or via static configuration. In the static configuration case, the PS could be pre-configured with the capability and contact information of the available gateways. This solution may be suitable when the sensors infrastructure is owned and deployed by the network operator, such as in the case

of logical sensors/IMS integration. However, for third party deployed WSNs, dynamic discovery may be a more practical solution.

Several approaches could be envisaged to enable the dynamic discovery of available WSN gateways, by the PS. The approach to be used should fit/be consistent with the proposed business model, and should introduce minimal changes to the existing architecture as well as generate reasonably low network load overhead. We now evaluate three potential approaches with respect to these requirements and select the most suitable one.

The first possibility is to introduce a new IMS entity, acting as context broker, between the WSN gateways and the PS, as depicted in figure 4.13a. In this case, WSN gateways would publish their capabilities to this broker, with which the PS will interact to discover those capabilities. After that, the PS could interact with the chosen WSN gateway for the acquisition of the needed information. Various service location protocols could be used to support the operation of such system, such as: the Lightweight Directory Access Protocol (LDAP) [80]; the Universal Description, Discovery, and Integration standard (UDDI) [81]; JXTA [82]; the Universal Plug and Play standard (UpnP) [83]; and the Service Location Protocol (SLP) [84]. The drawbacks of this approach are that it introduces important changes to the existing architecture (i.e. a new entity and new interfaces/protocols), and generates additional network load in relation to the capabilities publication/discovery interactions that are carried using a separated service location protocol.

Another potential approach could be to enhance an existing IMS entity with the needed capabilities publication/discovery mechanism. Figure 4.13b shows how this approach could be realized via the enhancement of the S-CSCF functionality. In this case, WSN gateways could include in the bodies of the SIP registration messages, a description of their

capabilities (e.g. type/accuracy of information they can provide). Upon the acceptance of the registrations, the S-CSCF (now enhanced with a context brokerage function) will save the gateways capabilities, which can be requested later on by the PS, using SIP OPTIONS messages (including queries indicating the criteria that should be satisfied). The S-CSCF will then map the identifier of the piece of contextual information requested to the contact information of relevant gateways, and return the list of the matching gateways (if any) and their capabilities in a 200 OK SIP message. In comparison to the first approach, this approach generates less network load since capabilities publication interactions are embedded in the basic SIP registration operation. Nevertheless, it still generates some additional traffic related to the capabilities discovery interactions (i.e. the OPTIONS/OK messages) and requires some changes to the existing architecture (i.e. the enhancement of the S-CSCF functionality with context brokerage capabilities and the extension of the PS with service discovery capabilities).

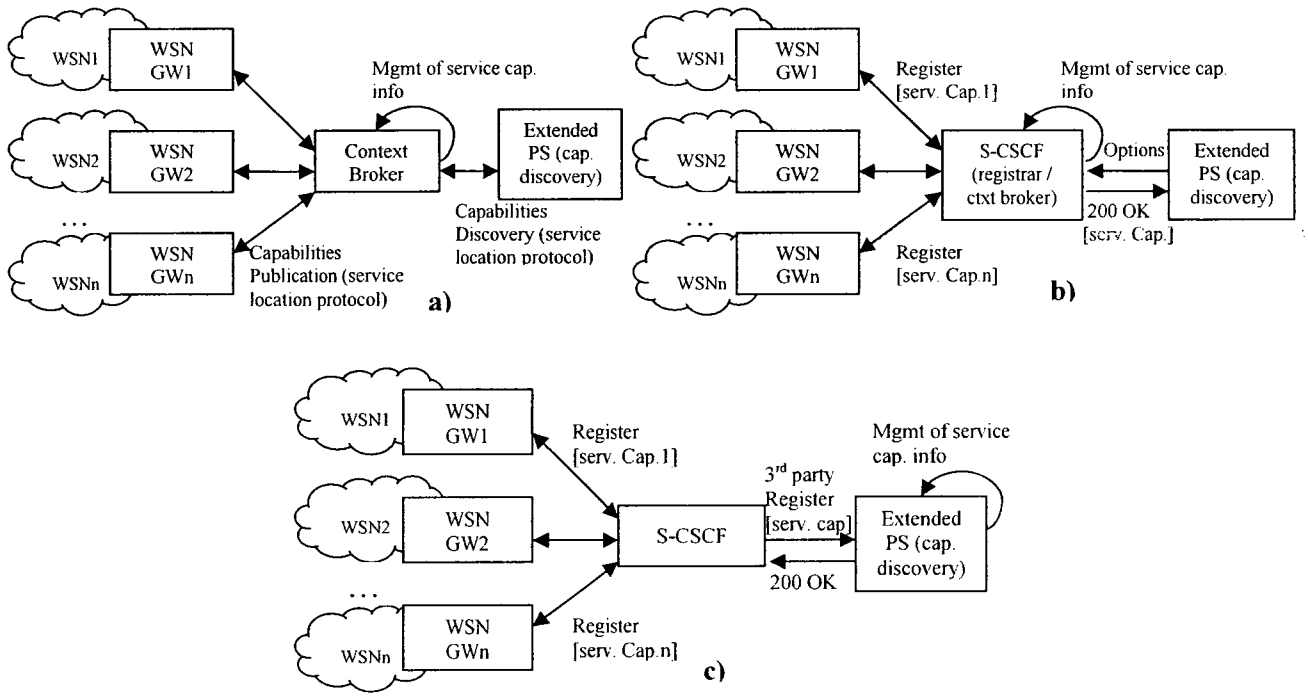


Figure 4.13: Possible approaches for the dynamic discovery of WSN gateways' capabilities: a) Relying on a new IMS entity acting as context broker; b) Enhancing the S-CSCF with context brokerage functionality; c) Leveraging the existing 3rd party registration mechanism

A third possible approach, which is illustrated in figure 14.3c, is to leverage the existing 3rd party registration mechanism to enable gateways' capabilities publication. This approach is similar to the second one, except that instead of enhancing the S-CSCF functionality, this CSCF will simply use the existing 3rd party registration mechanism (it already supports) to convey gateways' capability information to the PS, as follows: WSN gateways will include their XML-formatted service capability information in the bodies of SIP REGISTER messages sent to the S-CSCF during IMS registration. The S-CSCF then builds a 3rd party REGISTER message (containing the capability info of several gateways) and send it to the PS. This last extracts and stores the service capability information that it can use later on to choose the best information sources suiting its clients' needs. The benefits of this approach is that it introduces minimal changes to the existing architecture (only enhancements to the PS functionality are needed), and generates reasonable network load since capabilities publication interactions are embedded in the basic 3rd party registration procedures. Furthermore, it is consistent with our business model. For those reasons, we opt for the last approach to enable the dynamic discovery of WSN gateways' capabilities in our architecture.

4.5 The Extended Presence Information Model

Information modeling consists of knowledge representation in a standard format that makes it easy to use, understand, and share with other entities. The 3GPP presence architecture mandates the use of the XML-based PIDF [42], defined by the IETF, as presence information model. As mentioned previously, The PIDF only defines basic status and contact information, but can be extended. The RPID [43], the CIPID [44], the GEOPRIV [45], and the OMA [66] information models are among the existing extensions.

To enable the management of contextual information in the IMS, other extensions need to

be defined to accommodate the additional types of information captured by physical/logical sensors (i.e. spatial, physiological, environmental, and logical status data). Furthermore, the information model should enable the distinction between the different types of contextual entities to which the information relates. Figure 4.14 highlights the extension elements and attributes we introduced to the standard PIDF, to address these aspects.

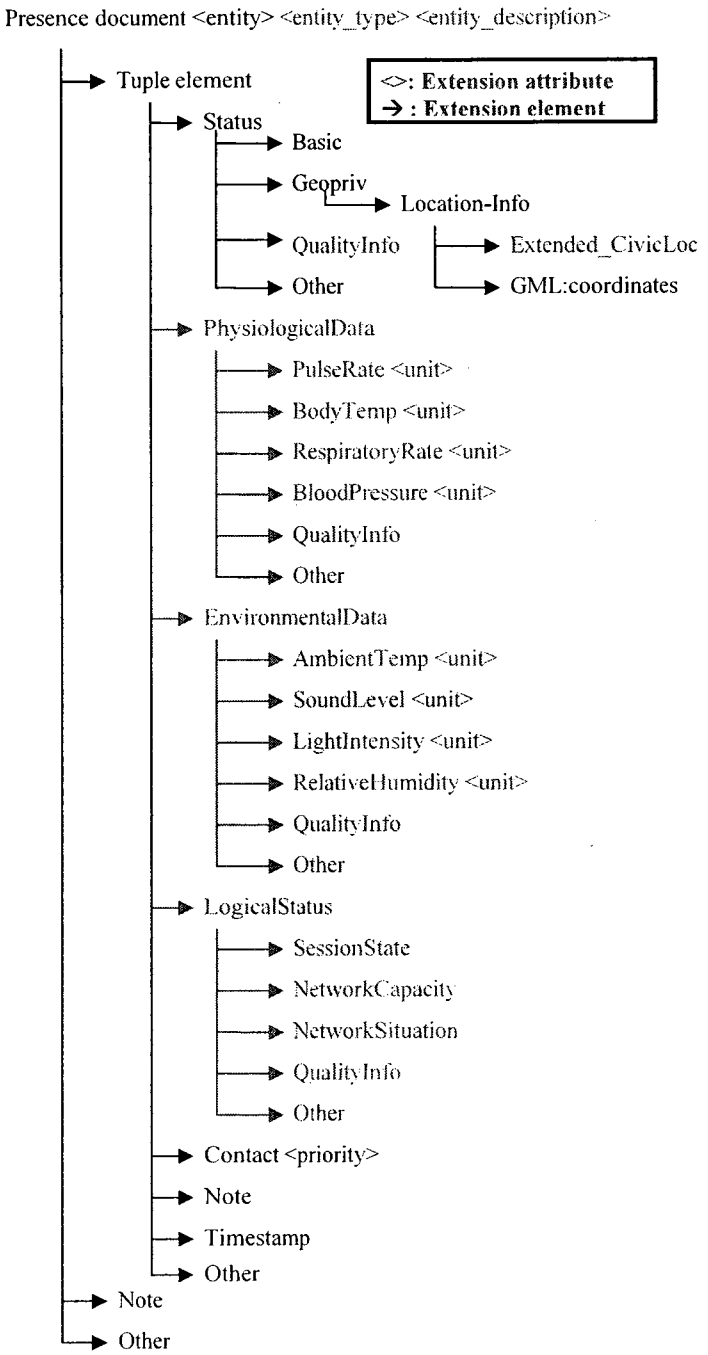


Figure 4.14: The extended PIDF structure

At the root of any presence document, we find a "presence" data element, which has an "entity" attribute specifying the URL of the presentity to which the information belongs, and a mandatory child element "tuple" constituting an umbrella element used to structure the presence document into identifiable sections of information. To allow the encapsulation of physiological, environmental, and logical status data within a presence document, three new optional sub-elements ("physiologicalData", "environmentalData", and "logicalStatusData") are added to the existing "tuple" element. Each of these sub-elements is further divided into other sub-elements. The complex element "physiologicalData" is divided into "pulseRate", "bodyTemp", "respiratoryRate", and "bloodPressure" sub-elements, while the "environmentalData" element is divided into "ambientTemperature", "soundLevel", "lightIntensity", and "relativeHumidity" sub-elements. As for the "logicalStatus" complex element, it is divided into "sessionState", "networkCapacity", and "networkSituation" sub-elements. In addition to these sub-elements, each of the three new complex data elements also includes a "qualityInfo" sub-element and any number of optional extension sub-elements. For spatial information, we leverage the existing GEOPRIV extension data element [45] for its representation, and extend one of its child elements (namely "civicLoc") with refined location information such as room ID, displacement direction, and relative distance to other.

Finally, to enable the distinction between different types of entities (i.e. person, object, place, or network) to which the information may relate, we added two new mandatory attributes "entityType" and "entityDescription" to the existing "presence" element. In the coming sub-sections, we discuss the syntax and semantics of the new extension elements and attributes in details.

4.5.1 Physiological Data Related Extension

Four pieces of information were used for the modeling of physiological data, which could be useful for wireless healthcare applications and enhanced emergency services. The first piece of information is the "pulseRate", which represents the rate at which the heart beats. The "pulseRate" is a complex data element consisting of a mandatory attribute called "unit" and two child elements, one of which is mandatory (the "pulseValue") while the other is optional (the "pulseDescription"). The "unit" attribute is a string used to specify a unit of measurement for the pulse rate, and which is set to the value "bpm" (i.e. beats per minute) per default. The "pulseValue" that is used for specifying the value of the pulse rate is an unsigned integer ranging between 50 and 150 beats per minute – these values representing the normal upper and lower bound for human heart rates. The "pulseDescription" is an optional string field that is used to provide a description of the pulse rate condition. It is limited to three possible values: "normalRange" (signifying a rate between 60 and 100 bpm); "tachycardia" (signifying a fast rate that is > 100 bpm); and "bradycardia" (signifying a slow rate that is < 60 bpm).

The second child element defined for physiological data is "bodyTemperature", which gives an indication of a human body temperature. Like the pulse rate element, the body temperature element also consists of a mandatory "unit" attribute, a mandatory "temperatureValue" child element, and an optional "temperatureDescription" child element. The string unit attribute is set to "Celsius" by default in this case, while the "temperatureValue" is limited to the range of 30 to 42 degrees Celsius. As for the "temperatureDescription", it is also limited to three possible string values: "normalRange" (lying between 35.5° C and 37.2° C); "hypothermia" (signifying low body temperature that is < 35.5° C); and "fever" (signifying high body temperature that is > 38.9° C).

Following the same structure, the last two elements (i.e. "respirationRate" and "bloodPressure") also encompass a mandatory string unit attribute; a mandatory integer field representing the numeric value of the parameter; and an optional string field giving a textual description about the biometric parameter's condition. In the case of the "respiratoryRate" element that represents the rate at which a person inhales/exhales, the unit attribute is set to "resp. per minute" per default. As for the "respirationRate" value, it is limited to the range of 15 to 40 respirations per minute, while the "respirationDescription" is limited to the following values: "normalRange" (signifying a range of 12 to 20 resp. per minute for adults; or 20 to 30 resp. per minute for preschool children; or 20 to 40 resp. per minute for infants); "hypoventilation" (signifying a slow breathing rate that is < 12 resp. per minute); and "hyperventilation" (signifying a rapid breathing rate that is > 20 resp. per minute).

Finally, in the case of the "bloodPressure" element that represents the pressure exerted by the blood against the walls of the blood vessels, the unit attribute is set to "mm HG" (or millimetres of mercury). The "pressureValue" is divided in two sub-elements: systolic (highest pressure during a heart beat measured when the heart contracts) which ranges between 100 and 200 mm HG; and diastolic (lowest pressure during a heart beat measured when the heart fills with blood) which ranges between 60 and 100 mm HG. The "pressureDescription" element expresses one of three possible values: "normalRange" (average blood pressure being 120/80 mm HG); "hyperTension" (representing high blood pressure); "hypoTension" (representing low blood pressure). Figure 4.15 shows the portion of the defined XML schema that relates to the "physiologicalData" element definition.

```
<xs:complexType name="tuple">
    <xs:sequence>
      <xs:element name="status" type="tns:status"/>
      <xs:element name="contact" type="tns:contact" minOccurs="0"/>
      <xs:element name="note" type="tns:note" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="physiologicalData" type="tns:physiology"
minOccurs="0"/>
      <xs:element name="environmentalData" type="tns:environment"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
  </xs:complexType>
  <xs:complexType name="physiology">
    <xs:sequence>
      <xs:element name="pulseRate" type="tns:pulse" minOccurs="0"/>
      <xs:element name="bodyTemperature" type="tns:bodyTemperature"
minOccurs="0"/>
      <xs:element name="respiratoryRate" type="tns:respiration"
minOccurs="0"/>
      <xs:element name="bloodPressure" type="tns:pressure"
minOccurs="0"/>
      <xs:element name="qualityinfo" type="tns:qualityinfo" minOccurs="0"
maxOccurs="1" />
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
 <xs:complexType name="pulse">
    <xs:sequence>
      <xs:element name="pulseValue" type="tns:pulseValue"
minOccurs="1"/>
      <xs:element name="pulseDescription" type="tns:pulseDescription"
minOccurs="0"/>
    </xs:sequence>
      <xs:attribute name="unit" type="xs:string" use="required"
fixed="bpm"/>
  </xs:complexType>
  <xs:simpleType name="pulseValue">
    <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="50"/>
      <xs:maxInclusive value="150"/>
    </xs:restriction></xs:simpleType>
  <xs:simpleType name="pulseDescription">
    <xs:restriction base="xs:string">
      <xs:enumeration value="normalRange"/>
      <xs:enumeration value="tachycardia"/>
      <xs:enumeration value="bradycardia"/>
    </xs:restriction> </xs:simpleType>
  <xs:complexType name="bodyTemperature">
    <xs:sequence>
      <xs:element name="temperatureValue" type="tns:tempValue"
minOccurs="1"/>
      <xs:element name="tempDescription" type="tns:tempDescription"
minOccurs="0"/>
    </xs:sequence>
      <xs:attribute name="unit" type="xs:string" use="required"
fixed="Celsius"/>
  </xs:complexType>
  <xs:simpleType name="tempValue">
    <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="30"/>
      <xs:maxInclusive value="42"/>
    </xs:restriction>
  </xs:simpleType>
```

```
  <xs:simpleType name="tempDescription">
    <xs:restriction base="xs:string">
      <xs:enumeration value="normalRange"/>
      <xs:enumeration value="hypothermia"/>
      <xs:enumeration value="fever"/>
    </xs:restriction>
  </xs:simpleType>
<xs:complexType name="respiration">
    <xs:sequence>
      <xs:element name="respirationRate" type="tns:respirationRate"
minOccurs="1"/>
      <xs:element name="respirationDescription"
type="tns:respirationDescription" minOccurs="0"/>
    </xs:sequence>
      <xs:attribute name="unit" type="xs:string" use="required"
fixed="resp. per minute"/>
  </xs:complexType>
  <xs:simpleType name="respirationRate">
    <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="15"/>
      <xs:maxInclusive value="40"/>  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="respirationDescription">
    <xs:restriction base="xs:string">
      <xs:enumeration value="normalRange"/>
      <xs:enumeration value="hypoventilation"/>
      <xs:enumeration value="hyperventilation"/>
</xs:restriction> </xs:simpleType>
    <xs:complexType name="pressure">
    <xs:sequence>
      <xs:element name="pressureValue" type="tns:pressureValue"
minOccurs="1" />
      <xs:element name="pressureDescription"
type="tns:pressureDescription" minOccurs="0" />
    </xs:sequence>
      <xs:attribute name="unit" type="xs:string" use="required"
fixed="mm HG" />
  </xs:complexType>
  <xs:complexType name="PressureValue">
  <xs:sequence>
    <xs:element name="systolic" type="tns:systolic" nimOccurs="1" />
    <xs:element name="diastolic" type="tns:diastolic" nimOccurs="1" />
  </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="systolic">
    <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="100" />
      <xs:maxInclusive value="200" />
    </xs:restriction>
  </xs:simpleType>
 <xs:simpleType name="diastolic">
    <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="60" />
      <xs:maxInclusive value="100" />
    </xs:restriction>
  </xs:simpleType>
 <xs:simpleType name="pressureDescription">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normalRange" />
    <xs:enumeration value="hyperTension" />
    <xs:enumeration value="hypoTension" />
  </xs:restriction>
 </xs:simpleType>
```

Figure 4.15: XML schema definition for the physiological data related extension

It should be noted that the four data element described where chosen as representatives of physiological data because they correspond to the four basic vital signs that are normally used for the assessment of a person's physical condition. However, other elements (e.g. blood sugar level) could be added to the defined XML schema, via the "other" sub-element provided. Furthermore, since this information could be collected from various sources offering different levels of accuracy/granularity, a "qualityInfo" child element is appended to the physiological data element, to give an indication about the quality of the information conveyed. Figure 4.16 shows the definition of this "qualityInfo" element, which consists of the following quality attributes: "freshness" (representing a timestamp of when the info was produced); "granularity" (a string indicating the level of granularity of the info using one of two possible values: raw and derived); and "accuracy" (a decimal number between 0 and 1 representing the probability of error of the info – thus the info source reliability).

```
<xs:complexType name="qualityinfo">
<xs:sequence>
 <xs:element name="freshness" type="xs:dateTime" />
 <xs:element name="granularity" type="tns:granularity" />
 <xs:element name="accuracy" type="tns:accuracy" minOccurs="0" maxOccurs="1" />
 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
 </xs:sequence>
 </xs:complexType>
<xs:simpleType name="granularity">
<xs:restriction base="xs:string">
 <xs:enumeration value="raw" />
 <xs:enumeration value="derived" />
 </xs:restriction>
 </xs:simpleType>
<xs:simpleType name="accuracy">
<xs:restriction base="xs:decimal">
 <xs:pattern value="0(.|0-9]{0,3})?" />
 <xs:pattern value="1(.0{0,3})?" />
 </xs:restriction>
</xs:simpleType>
```

Figure 4.16: Definition of the quality information data element

4.5.2 Environmental Data Related Extension

As shown in figure 4.17, four data elements were chosen as representatives of environmental information (i.e. "ambientTemperature", "soundLevel", "lightIntensity", and "relativeHumidity"), in addition to the quality information element presented previously

and the "other" element added for extensibility purposes. This type of information could be useful for pervasive gaming and interest-based applications.

The "ambientTemperature" element, which represents the temperature of the surrounding physical environment, encompasses a mandatory child element "tempValue" providing an integer value of the measured temperature and a mandatory "unit" attribute specifying the unit of measurement as one of three possible values: "Celsius"; "Fahrenheit;" and "Kelvin". Conveying sound level information, the "soundLevel" element consists of a mandatory integer child element "soundlevel" conveying the value of the sound measurement and a mandatory string attribute "unit" fixing the unit of the measurement at "decibels" per default. Similarly, the "lightIntensity" element consists of an integer child element called "intensity" and a string "unit" attribute initialized to "candelas". Finally, the "relativeHumidity" element conveys humidity information via an unsigned integer mandatory sub-element called "humidityLevel" that is limited to the range of 30 to 100 % (i.e. the typical relative humidity percentages), and a "unit" attribute set at the value "%".

```
<xs:complexType name="environment">
 <xs:sequence>
  <xs:element name="ambientTemperature" type="tns:temperature"
minOccurs="0" />
  <xs:element name="soundLevel" type="tns:sound" minOccurs="0" />
  <xs:element name="lightIntensity" type="tns:light" minOccurs="0" />
  <xs:element name="relativeHumidity" type="tns:humidity"
minOccurs="0" />
  <xs:element name="qualityinfo" type="tns:qualityinfo" minOccurs="0"
maxOccurs="1" />
  <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
 </xs:sequence>
</xs:complexType>
<xs:complexType name="temperature">
  <xs:sequence>
    <xs:element name="tempValue" type="xs:int" minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="unit" type="tns:tempUnit" use="required" />
 </xs:complexType>
<xs:simpleType name="tempUnit">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Celsius" />
    <xs:enumeration value="Fahrenheit" />
    <xs:enumeration value="Kelvin" />
  </xs:restriction>
</xs:simpleType>
```

```
<xs:complexType name="sound">
  <xs:sequence>
    <xs:element name="soundlevel" type="xs:int" minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="unit" type="xs:string" use="required"
fixed="decibels" />
</xs:complexType>
<xs:complexType name="light">
  <xs:sequence>
    <xs:element name="intensity" type="xs:int" minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="unit" type="xs:string" use="required"
fixed="candelas" />
</xs:complexType>
<xs:complexType name="humidity">
  <xs:sequence>
    <xs:element name="humidityLevel" type="tns:humidityLevel"
minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="unit" type="xs:string" use="required" fixed="%" />
</xs:complexType>
<xs:simpleType name="humidityLevel">
  <xs:restriction base="xs:unsignedShort">
    <xs:minInclusive value="30" />
    <xs:maxInclusive value="100" />
  </xs:restriction>
</xs:simpleType>
```

Figure 4.17: XML schema definition for the environmental data related extension

4.5.3 Refined Location Information Related Extension

The existing GEOPRIV extension model [45] defines a data element called "location-info" that allows the representation of different types of location information related to presentities. This specification proposes the use of one of two formats for the representation of this data element, namely: the GML feature.xds schema specified in [85] and the "civicLoc" format defined as part of the GEOPRIV specification [45]. The GML format is geared towards low level geographic coordinates type of information, while the civicLoc format supports higher level civic addressing information (e.g. country, city, and street names). However, other refined types of location information (e.g. displacement direction and relative distance to objects/persons) may be needed for some context-aware applications such as interest-based services.

To represent this type of information, we chose to extend the civicLoc format' since it already focuses on high level location information. More specifically, three new child elements were added to the civicLoc's "civicAddress" data element namely: "RID" (Room ID) which is a string conveying the ID of the room the contextual entity is currently located in (e.g. office-4060); "DD" (Displacement Direction) which is a string representing the displacement direction of an entity as one of eight possible values (north, south, east, west, north-east, north-west, south-east, and south-west); and "RD" (Relative Distance) which consists of a mandatory attribute "other-entity" specifying the URL of the other entity relative to which the distance is measured, and an integer child element called "distance" carrying the numeric value of the relative distance. Figure 4.18 shows the extended civicLoc schema proposed.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:pidf:geopriv10:civicLoc"
xmlns:tns="urn:ietf:params:xml:ns:pidf:geopriv10:civicLoc"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:complexType name="civicAddress">
 <xs:sequence>
  <xs:element name="country" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="A1" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="A2" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="A3" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="A4" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="A5" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="A6" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="PRD" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="POD" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="STS" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="HNO" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="HNS" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="LMK" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="LOC" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="FLR" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="NAM" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="PC" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="RID" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="DD" type="tns:displacementDirection" minOccurs="0"
maxOccurs="1" />
  <xs:element name="RD" type="tns:relativeDistance" minOccurs="0" maxOccurs="1" />
  <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
 </xs:sequence>
 </xs:complexType>
```

```
 <xs:simpleType name="displacementDirection">
 <xs:restriction base="xs:string">
  <xs:enumeration value="north" />
  <xs:enumeration value="south" />
  <xs:enumeration value="east" />
  <xs:enumeration value="west" />
  <xs:enumeration value="north-east" />
  <xs:enumeration value="north-west" />
  <xs:enumeration value="south-east" />
  <xs:enumeration value="south-west" />
 </xs:restriction>
 </xs:simpleType>
<xs:complexType name="relativeDistance">
 <xs:sequence>
  <xs:element name="distance" type="xs:int" minOccurs="1" />
 </xs:sequence>
 <xs:attribute name="otherEntity" type="xs:anyURI"
use="required" />
 </xs:complexType>
 </xs:schema>
```

Figure 4.18: Extended civicLoc schema

4.5.4 Network Status Information Related Extension

As shown in figure 4.19, three data elements were chosen to represent network logical status information (i.e. "sessionsState", "networkCapacity", and "networkSituation"), in addition to the "qualityinfo" element and the "other" element enabling the extensibility of the model. This type of information could be useful for enhanced network services, such as advanced QoS schemes and enhanced emergency communications.

The "sessionState" element, which represents the state information of an ongoing session, consists of three mandatory sub-elements and a mandatory attribute. The first sub-element "category" is a string describing the category of the session as one of five possible values: "silver"; "gold"; "platinum"; "emergency-public"; and "emergency-authority" – these values being related to the advanced QoS scheme that will be presented in the coming chapter. The second sub-element "ParticipantsNum" is an integer indicating the number of participants in the session, while the third sub-element "mediaParameters" describes the

parameters of the media used in the session. This last sub-element consists of two mandatory child elements, namely: "mediaType" a string specifying the type of media as one of three possible values ("audio", "video", or "text"); and "mediaFormat" a string specifying the media format as one of four possible values ("AMR", "AMR-WB", "H.263", or "T.140") that represent the most common media codecs used in the IMS. As for the mandatory attribute called "ID", it consists of a string representing the ID of the call that is extracted from the signalling messages.

Conveying network capacity related information, the "networkCapacity" element consists of a mandatory sub-element ("averageThroughput") and two optional sub-elements ("averageLinksUtilization" and "capacityDescription"). The "averageThroughput" sub-element carrying information about the total average throughput currently achieved in the network, consists of a mandatory integer sub-element "throughputValue" indicating the numeric value of the throughput and a mandatory string attribute "unit" indicating the unit of the measurement as one of two possible values: "bps" (bits per second) and "pps" (packets per second). Similarly, the "averageLinksUtilization" element, giving an indication about the B.W. utilization efficiency, consists of a mandatory integer child element "linksUtlizationValue" indicating the % of the total average links utilization and a mandatory string attribute "unit" fixed at the value "%" per default.

As for the "capacityDescription" sub-element, it gives a textual description about the network capacity situation as one of four possible values: "lightLoad" (signifying that current throughput is < 25% of max. achievable throughput); "regularLoad" (signifying that current throughput lies between 25% and 69% of max. achievable throughput); "overloadThreshold" (signifying that current throughput equals 70% of max. achievable

throughput); and "overload" (signifying that current throughput is > 70% of max. achievable throughput).

As for the "networkSituation" element giving a high level indication of network situation, it consists of a mandatory string sub-element "situation" describing the network situation as one of two possible values (i.e. "regular" or "crisis"), and an optional integer sub-element "911callsinfo" providing information about the number of 911 calls currently serviced in the network.

```
<xs:complexType name="logicstatus">
  <xs:sequence>
    <xs:element name="sessionState" type="tns:stateinfo" minOccurs="0"/>
    <xs:element name="networkCapacity" type="tns:networkcapacity"
minOccurs="0" maxOccurs="1" />
    <xs:element name="networkSituation" type="tns: operationinfo"
minOccurs="0" maxOccurs="1" />
    <xs:element name="qualityinfo" type="tns:qualityinfo" minOccurs="0"
maxOccurs="1" />
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
<xs:complexType name="stateinfo">
  <xs:sequence>
    <xs:element name="category" type="tns:category" minOccurs="1" />
    <xs:element name="ParticipantsNum" type="xs:nonNegativeInteger"
minOccurs="1" />
    <xs:element name="mediaParameters" type="tns:mediaParameters"
minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="ID" type="xs:string" use="required" />
</xs:complexType>
<xs:simpleType name="category">
  <xs:restriction base="xs:string">
    <xs:enumeration value="silver" />
    <xs:enumeration value="gold" />
    <xs:enumeration value="platinum" />
    <xs:enumeration value="emergency-public" />
    <xs:enumeration value="emergency-authority" />
  </xs:restriction>
 </xs:simpleType>
<xs:complexType name="mediaParameters">
  <xs:sequence>
    <xs:element name="type" type="tns:mediaType" minOccurs="1" />
    <xs:element name="format" type="tns:mediaFormat" minOccurs="1" />
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="mediaType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="audio" />
    <xs:enumeration value="video" />
    <xs:enumeration value="text" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="mediaFormat">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AMR" />
    <xs:enumeration value="AMR-WB" />
    <xs:enumeration value="H.263" />
    <xs:enumeration value="T.140" />
  </xs:restriction> </xs:simpleType>
```

```
<xs:complexType name="networkcapacity">
  <xs:sequence>
    <xs:element name="averageThroughput" type="tns:throughput"
minOccurs="1"/>
    <xs:element name="averageLinksUtilization" type="tns:linksUtilization"
minOccurs="0"/>
    <xs:element name="capacityDescription" type="tns:capacitydescription"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="throughput">
  <xs:sequence>
    <xs:element name="throughputValue" type="xs:int" minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="unit" type="tns:throughputUnit" use="required" />
 </xs:complexType>
<xs:simpleType name="throughputUnit">
  <xs:restriction base="xs:string">
    <xs:enumeration value="bps" />
    <xs:enumeration value="pps" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="linksUtilization">
  <xs:sequence>
    <xs:element name="linksUtlizationValue" type="xs:int" minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="unit" type="xs:string" use="required" fixed="%" />
</xs:complexType>
<xs:simpleType name="capacityDescription">
  <xs:restriction base="xs:string">
    <xs:enumeration value="lightLoad" />
    <xs:enumeration value="regularLoad" />
    <xs:enumeration value="overloadThreshold" />
    <xs:enumeration value="overload" />
  </xs:restriction> </xs:simpleType>
<xs:complexType name="operationinfo">
  <xs:sequence>
    <xs:element name="situation" type="tns:situation" maxOccurs="1"/>
    <xs:element name="911callsInfo" type="xs:nonNegativeInteger"
minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="situation">
  <xs:restriction base="xs:string">
    <xs:enumeration value="regular" />
    <xs:enumeration value="crisis" />
  </xs:restriction>
</xs:simpleType>
```

Figure 4.19: XML schema definition for the network status information related extension

4.5.5 Extension for Distinction between Different Types of Contextual Entities

To enable the distinction between the different types of contextual entities to which the information may relate, two new attributes were added to the existing "presence" root element, as shown in figure 4.20. The first attribute "entityType" is a mandatory string specifying the type of contextual entity to which the presence document relates as one of four possible values: "person"; "object"; "place"; or "network". As for the second attribute "entityDescription", it is an optional string giving a more detailed description of the concerned entity (e.g. inflatable game object, game player, hospital room, or telecom. network).

```
<xs:complexType name="presence">
 <xs:sequence>
   <xs:element name="tuple" type="tns:tuple" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="note" type="tns:note" minOccurs="0" maxOccurs="unbounded" />
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
 </xs:sequence>
 <xs:attribute name="entity" type="xs:anyURI" use="required"/>
 <xs:attribute name="entityType" type="tns:entityType" use="required"/>
 <xs:attribute name="entityDescription" type="xs:string"/>
 </xs:complexType>
   <xs:simpleType name="entityType">
    <xs:restriction base="xs:string">
     <xs:enumeration value="person"/>
     <xs:enumeration value="object"/>
     <xs:enumeration value="place"/>
     <xs:enumeration value="network"/>
    </xs:restriction>
  </xs:simpleType>
```

Figure 4.20: Extension for the distinction between different types of contextual entities

4.6 Information Exchange Protocol and Models

In this section, we present the protocol we are proposing as context exchange protocol, and discuss the different information exchange models supported by our architecture.

4.6.1 SIMPLE as Context Exchange Protocol

The information exchange protocol to be used on the inbound and outbound interfaces must satisfy a set of requirements defined by 3GPP in [48], namely: it must not impose any limits on the size of the information transported; it should support full and partial update notifications; and should support the transport of information formatted according to the

PIDF. The IETF-defined SIMPLE [46] protocol suite (mainly offering a publish/subscribe/notify mechanism) satisfies all those requirements. Therefore, we believe it can be used for the transport of contextual information.

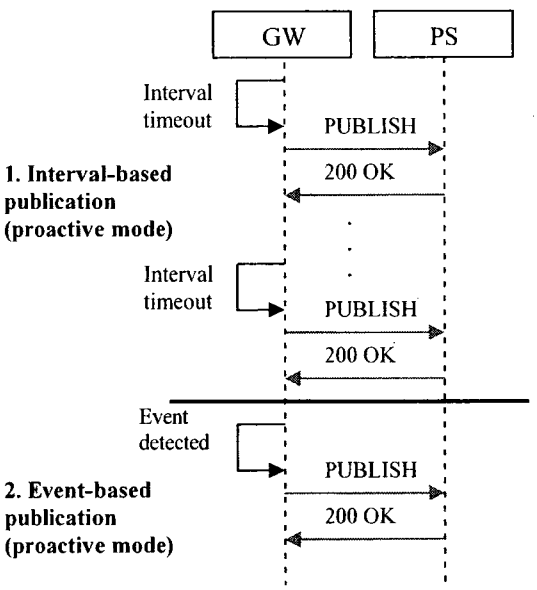
A main issue with this protocol is that it can generate a high signaling load on the network, due to the frequent information exchange interactions and the large amount of information transported. To solve this issue, several optimizations, such as partial events notifications [86]; event notification filters [87]; and event throttling [88], have been proposed. All these optimizations focus on improving the efficiency of the interactions over the outbound interface (i.e. the subscribe/notify interface).

Here, we propose another optimization targeting the inbound interface (i.e. the publish interface) related interactions. It consists of a 'publish-upon-request' mechanism, which can be used by the PS to trigger a certain gateway to publish information, only when requested. This mechanism can be useful for controlling the amount of information generated by certain types of sensors (e.g. environmental sensors), which typically generate large amounts of information that can overload the network. Furthermore, this reactive mode of publication can be helpful to obtain the value of a certain contextual attribute that is not yet published (or is outdated) in the network, thus complementing the current proactive mode of publications in which presentities actively publish information when they wish. Unlike the existing optimizations that aim at controlling the amount of information disseminated within the network, our optimization controls the amount of information that is entering the network.

4.6.2 Information Exchange Models

The proactive mode of operation, which is currently supported in the 3GPP architecture, is translated in two types of information publications. namely: interval-based publication in

which information is published on regular time intervals (e.g. every 15 sec); and event-based publication in which information is published when certain events are detected (e.g. when temperature is above 30 °C). These two types of publication are realized using regular PUBLISH/OK SIP interactions between the PS and the GW, as shown in figure 2.21a. We complement this mode of operation with the idea of 'publication policies', which are configured at the gateway level, in order to specify the mode/type publication that should be supported for different types of information. Figure 2.21b shows an example of a publication policy.



a)

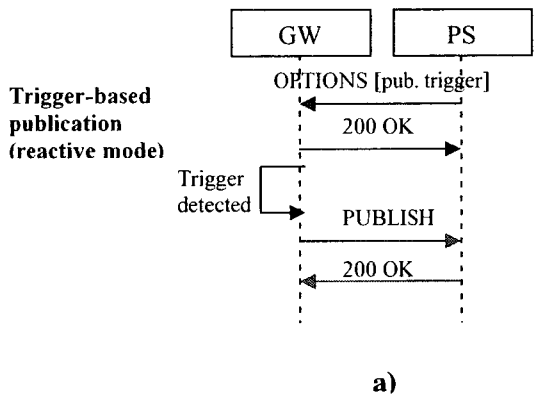
```
<?xml version="1.0" encoding="UTF-8"?>
<PublicationPolicies>
<WSNGW id="wsngw1.ericsson.canada@home1.net">
<Policy id = "ab25fv6">
   <EntityId>user1.Concordia.SGWcampus.Canada@operator.com
      </EntityId>
   <InfoType>LOC</InfoType>
   <Rule>
      <Pub_Mode>Proactive</Pub_Mode>
      <Pub_Type>
        <Type> Interval </Type>
         <Value> 15 </Value>
      </Pub_Type>
   </Rule>
<Policy>
<Policy id = "ab25fv7">
   <EntityId> Ev139.Concordia.SGWcampus.Canada@operator.com
      </EntityId>
   <InfoType>TEMP</InfoType>
   <Rule>
      <Pub_Mode>Proactive</Pub_Mode>
      <Pub_Type>
         <Type> Event </Type>
         <Condition> GT </Condition>
         <Value> 30 </Value>
      </Pub_Type>
   </Rule>
<Policy>
</PolicySet">
</WSNGW> </PublicationPolicies>
```

b)

Figure 4.21: Proactive mode of publication: a) Interval-based and event-based publications; b) Contents of a publication policy

The reactive mode of operation is a new mode that is introduced by our solution. It is translated into a third type of publications (i.e. trigger-based publications), which are achieved using SIP OPTIONS messages sent by the PS to the gateway (to trigger

information publication), as shown in figure 2.22a. The body of an OPTIONS message contains an XML document specifying the identifier(s) of the requested piece(s) of information and the entity/entities to which it related, as shown in figure 2.22b. Upon the receipt of the publication trigger by the gateway, it consults the publication policies to determine whether this type of publication is supported for the specified pieces of information. If so, the gateway accepts the trigger using a 200 OK SIP message, and then publishes the needed information using a SIP PUBLISH message.



a)

```
OPTIONS wsngw1.concordia.canada@home1.net  SIP/2.0
Via: SIP/2.0/UDP gwhost.concordia.canada@home1.net:5060_; comp=sigcomp;
        branch=z9hG4bK9h9ab
Max-Forwards: 70
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>,
        <sip:orig@scscf1.home1.net;lr>
From: < sip:.PS.montrealarea.canada@home1.net >;tag=31415
To: < sip:wsngw1.concordia.canada@home1.net >
Contact: < sip:.PS.montrealarea.canada@home1.net >
Call-ID: a84b4c76e66710
CSeq: 6 OPTIONS
Content-type: application/pub_trigger+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<PublicationTriggers>
<PRESENTITY>
  <EntityId>user1.Concordia.SGWcampus.Canada@operator.com </EntityId>
        <InfoType>LOC</InfoType>
</PRESENTITY>
<PRESENTITY>
  <EntityId>EvBuild.Concordia.SGWcampus.Canada@operator.com</EntityId>
        <InfoType>TEMP</InfoType>
</PRESENTITY>
</PublicationTriggers>
```

b)

Figure 4.22: Reactive mode of publication: a) Trigger-based publication; b) Contents of a SIP OPTIONS message

4.7 Information Exchange Scenarios

In this section, we detail some of the information publication and subscription scenarios in order to illustrate the operation of the architecture proposed.

4.7.1 Publication on Behalf of a Contextual Entity

Figure 4.23 depicts an information publication scenario carried by a WSN gateway, on behalf of a WSN entity.

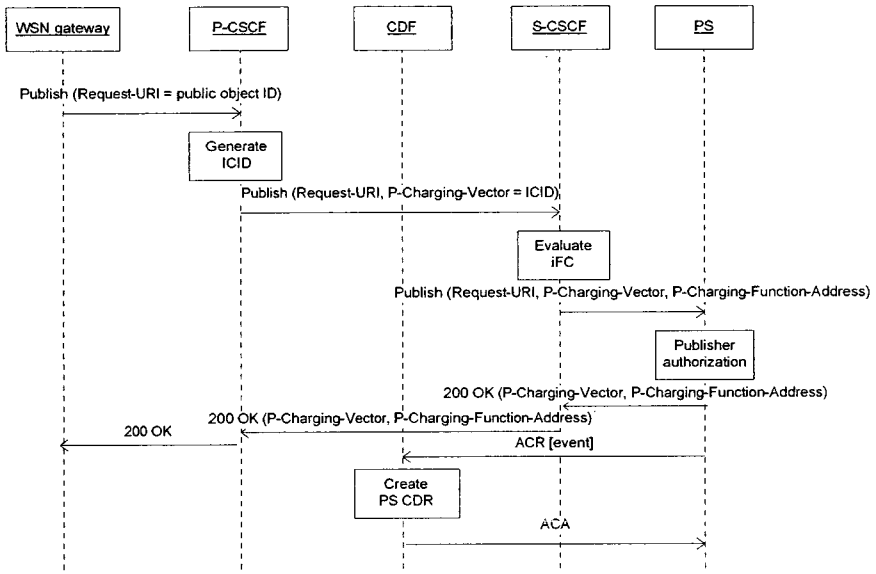


Figure 4.23: Information publication scenario performed by WSN gateway on behalf of a WSN entity

The scenario starts when the gateway sends a SIP PUBLISH message carrying in its request URI the public WSN entity ID, to its P-CSCF. The P-CSCF then generates an IMS Charging Identifier (ICID), which represents a common identifier used to uniquely identify the session (for charging purposes) between all involved elements, and inserts it in the message's P-Charging-Vector header. The message is then forwarded to the S-CSCF assigned to the corporate user (managing the WSN entity). After evaluating the iFC, the S-CSCF forwards the message to the PS (allocated to the user), which authorizes the publisher to make the publication, saves the published information, then responds with a SIP 200 OK message that is returned to the gateway following the reverse signaling path. After this response, the PS reports the information publication as a chargeable event to the CDF, via a pair of ACR[event]/ACR Diameter messages. The CDF makes use of the received information to create a PS-related CDR, which is sent to the billing domain. It should be noted that although the ICID created by the P-CSCF identifies a certain session related to a certain entity (the dependant WSN entity in this case), the processing of the

generated CDR in the billing domain will use the correlation between the corporate ID and the dependant WSN entity ID to issue a bill in the name of the corporate user, for the publication interaction.

4.7.2 Subscription by a Watcher Application

Figure 4.24 presents an information subscription scenario performed by an end-user application, acting as watcher.

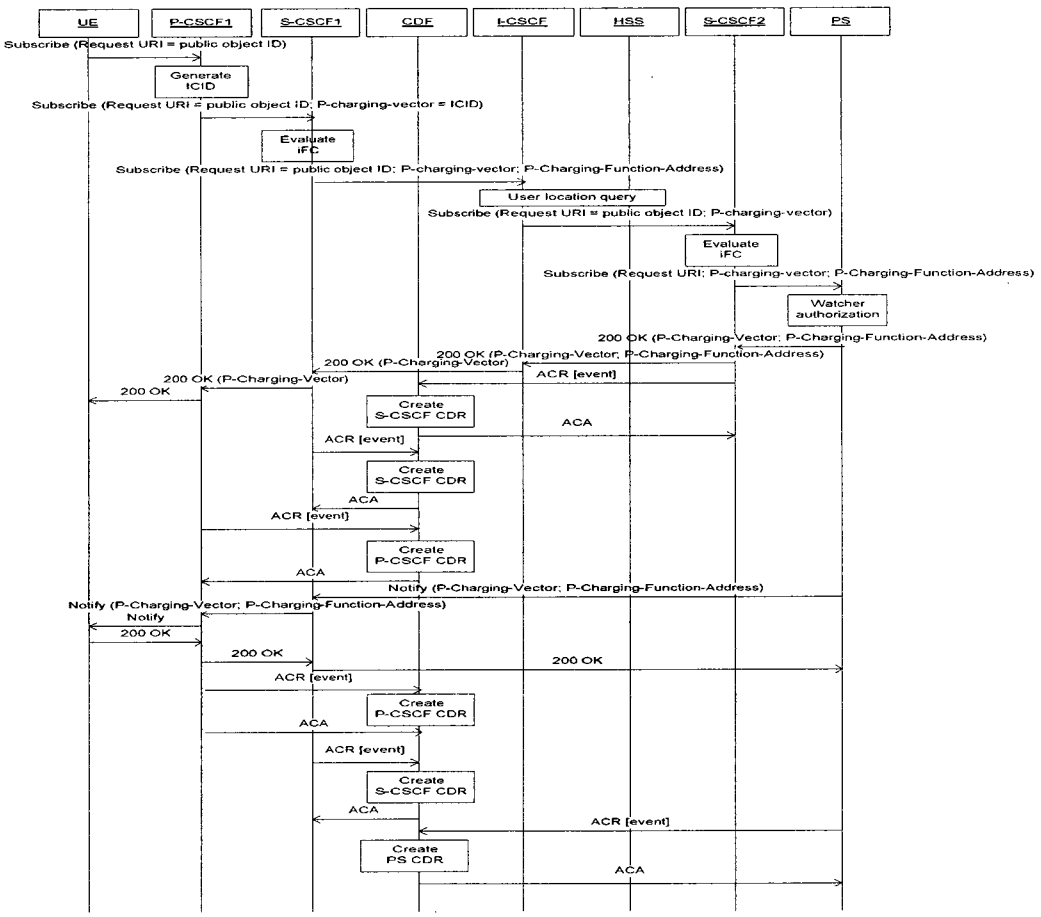


Figure 4.24: Information subscription scenario performed by a watcher end-user application

In this case, the UE (hosting the application) sends a SIP SUBSCRIBE message, carrying the identity of the entity (a WSN entity in this case) to whose information the watcher is subscribing in the request URI and the identity of the watcher (the application user in this case) in the P-Preferred-Identity header, to P-CSCF1. After generating an ICID for the

session and inserting this information in the message, P-CSCF1 forwards this message to the watcher's S-CSCF (i.e. S-CSCF1) which relays it (via I-CSCF and S-CSCF2) to the PS assigned to the corporate user managing the WSN of interest. This last performs watcher's authorization for information access, accepts the subscription using a SIP 200 OK message, then sends a SIP NOTIFY message containing the requested information to the watcher's UE (via it's P/S-CSCFs). It should be noted that the entity subscribing to the information (the application user in this case) is the one charged for subscription/notification events. Those events are reported as chargeable events by P/S-CSCFs after receiving their 200 OK related messages. Furthermore, notifications are reported as chargeable events by the PS.

4.7.3 Subscription by a Watcher Application Server

Figure 4.25 depicts an information subscription scenario performed by an application server, acting as watcher.

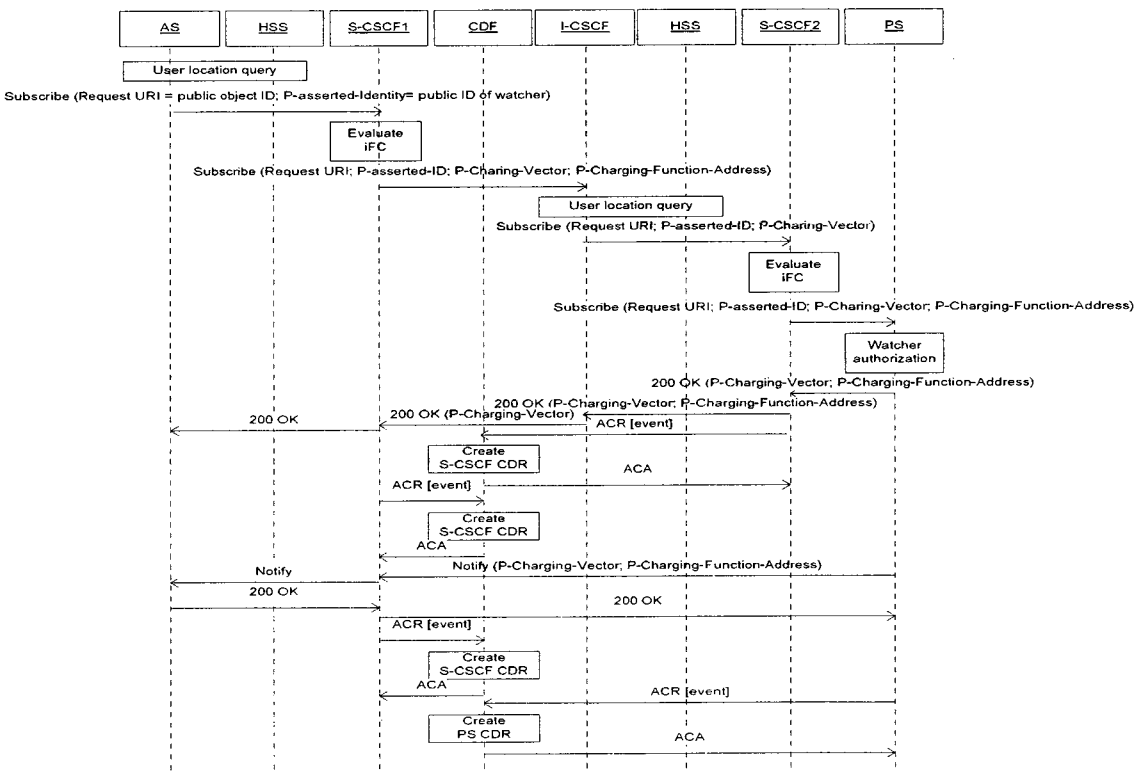


Figure 4.25: Information subscription scenario performed by a watcher application server

This scenario is similar to the previous one, except that application servers do not interface with P-CSCFs and directly interact with the IMS network via S-CSCFs (since they are trusted by the network). This implies that the watcher application server must first query the HSS to obtain the address of the S-CSCF assigned to the user (on whose behalf the subscription is made), then send the SUBSCRIBE message to this S-CSCF. This last being the first IMS node invoked in the session will generate an ICID for the session, and propagate it along the signaling path. In this case, charging/billing for the subscription/notification interactions is made for the watcher on whose behalf the subscription is made (i.e. the user of an application hosted by the AS).

4.7.4 Subscription by a Watcher Core Network Entity

The simplest information subscription scenario is the one involving a core network entity acting as watcher. Since such watcher entity is owned and trusted by the network (thus rendering security/charging issues irrelevant), it can directly interact with the PS to access the needed contextual information via an intra-operator interface (the Pwn interface). Figure 4.26 illustrate such this direct interaction between a watcher core network entity and the PS. In this case, we assume that the watcher network entity is configured with the PS address, or can perform PS discovery by interacting with the HSS and evaluating the iFC.

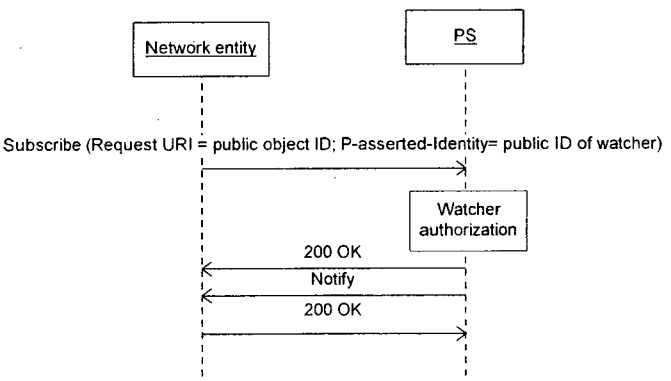


Figure 4.26: Information subscription scenario performed by a watcher core network entity

4.8 Conclusions

In this chapter, we have proposed a solution for context information acquisition and management in the IMS. This solution, which leverages and extends the 3GPP presence framework, consists of an IMS context management architecture and several components related to its operation.

The proposed architecture introduced new and enhanced entities/interfaces to the existing 3GPP presence architecture, in order to enable the IMS interaction with physical/logical sensors for the collection of a variety of contextual information in addition to enabling the effective management and dissemination of this information in the network. Two main information management issues related to the operation of this architecture were addressed, namely: the extension of the presence information model to enable the representation of additional types of information provided by sensors; and the elaboration of three information publication models enabling flexible and resource efficient information exchange over the inbound interfaces. To enable the practical deployment of the proposed architecture, a new business model along with suitable identification/charging schemes were proposed for this sensors-enabled IMS environment. Furthermore, support functions needed for regulating the interaction between the 3G network and information sources/consumers (i.e. security, info access control, and service discovery) were discussed, and different information exchange scenarios illustrating the system's operation were detailed.

Building on the capabilities of this solution, which ensures the availability of contextual information in the IMS, we will demonstrate in the next chapters how this information could be integrated in IMS operations, at the control and service levels.

Chapter 5

Use of Context-Awareness for Advanced QoS Support in the IMS

In this chapter, we propose a context-aware call differentiation solution, as means to offer enhanced QoS support in 3G networks. This solution enables the differentiation between different categories of calls at the IMS control level, via dynamic and adaptive resource allocation, in addition to supporting a specialized charging model enabling the effective charging of the resulting differentiated services.

The chapter starts by clarifying the work scope and presenting the proposed call differentiation scheme that enables the definition of various categories of calls. This is followed by the architectural framework and the resource management techniques and policies proposed to enable the support of this scheme in the IMS. The specialized charging model is then presented, followed by session management scenarios illustrating the system's operation and a presentation of our conclusions.

5.1 Introduction

The terms service differentiation and QoS provisioning are used interchangeably to signify the network's ability to distinguish between different classes of traffic/service and provide each class with the appropriate treatment, depending on its needs in terms of QoS parameters. There are two main service differentiation dimensions (or criteria): the traffic requirements in terms of resources; and the importance of the service session from the user perspective (irrespective of the traffic exchanged). In the first case for instance, one could differentiate between audio, video, and data traffic (based on their B.W. requirements); while in the second case, we could differentiate between regular and premium calls.

Combining these two dimensions offers a finer level of granularity in terms of service differentiation (e.g., premium audio, regular video ...etc). Figure 5.1 illustrates the two service differentiation dimensions, giving examples of applications that could fit at their different intersection points.

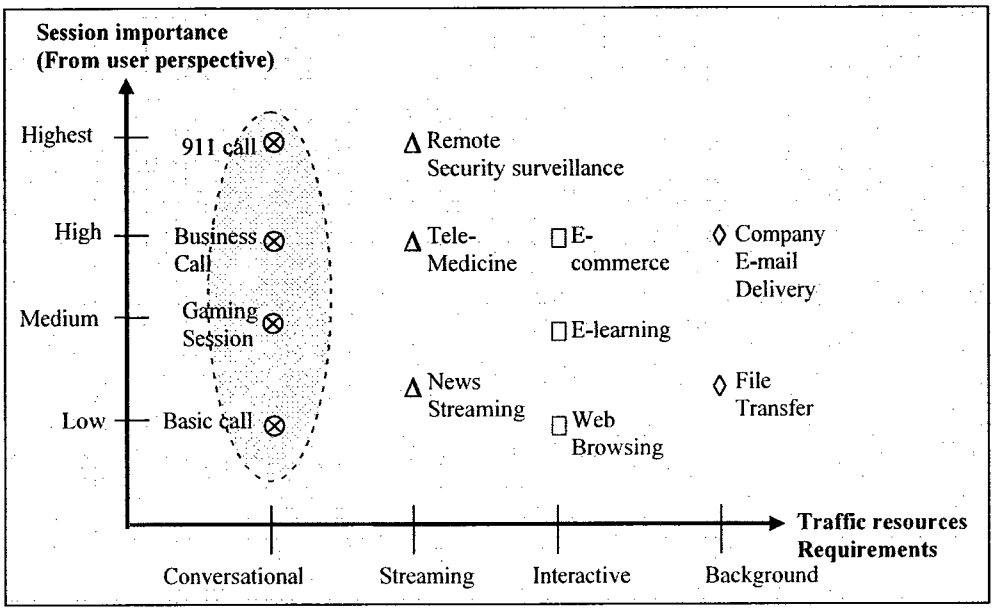


Figure 5.1: Service differentiation dimensions

Beside the service differentiation criteria, there are two other aspects that are related to the problem of service differentiation, namely: the resource management strategy used to allocate resources to the different classes based on their QoS requirements – a strategy that could be static or dynamic; and the level at which the solution is operating including the access/connectivity level, the routing level, or the control level.

The solution that we are proposing in this chapter focuses on the second service differentiation dimension by enabling the differentiation between different categories of sessions/calls within the conversational traffic class, as highlighted in figure 5.1. To achieve this scheme, the solution operates at the IMS control level and relies on a dynamic and adaptive resource allocation strategy. The different components of this solution will be elaborated in the coming sections.

5.2 A Call Differentiation Scheme for 3G Networks

The first component of the solution proposed is a call differentiation scheme, enabling the definition of various categories of calls with different QoS profiles. This scheme is described in the coming sub-sections.

5.2.1 Differentiating Factors

In our scheme, five differentiating factors are used to distinguish between the different classes, namely:

- **Call blocking probability (CBP)**: The CBP is the probability that a new call is blocked and not allowed admission to the core network. This factor reflects the priority a call receives (based on its importance) when being admitted to the network.
- **Forced call termination probability (FCTP)**: The FCTP is the probability that an ongoing call is terminated by the core network. This factor, which controls when a session ends, illustrates the possibility of preempting some ongoing sessions in order to free resources for new (more important) ones.
- **Multiparty session ability to grow**: This factor represents the limit a multiparty session can reach in terms of number of participants. In fact, some sessions may be allowed unlimited growth, while others may be subject to restrictions in terms of the number of participants (i.e. subject to session size control).
- **Media type guarantee**: This factor represents the ability to sustain a call with a certain media type, without downgrade to another type (e.g. dropping from video to voice). For instance, some sessions may offer guaranteed audio and video communications (i.e. Guaranteed Video - GV). while in others; video streams could be subject to downgrade (i.e. only offering Guaranteed Audio - GA).

- **User perceived media quality**: This factor represents the quality of the media transmission as perceived by the user. Some sessions could offer a quality that varies (var.) with the network conditions, while others could offer a sustained (sus.) quality (i.e. no freezing, interruptions…etc).

It should be noted that these factors enable the control of several communication aspects as means for differentiation, including: the control of when the session starts/ends; the session size; and the used media type/format.

5.2.2 QoS Profiles

Based on the differentiating factors introduced, we defined three classes of calls (Silver, Gold, and Platinum), as examples of possible QoS profiles. These classes were designed to accommodate the different needs that a user may have, in terms of priorities/guarantees, while making different types of calls (e.g. a regular call, an entertainment/gaming session, or a business call). Table 5.1 presents the proposed classes. It should be noted that these classes focus on "regular" calls, not addressing the case of emergency calls. In fact, emergency calls have specific requirements which will be addressed in the coming chapter.

Factor / Class	CBP			FCTP			Multiparty session ability to grow		Media type guarantee		User-perceived media quality	
	H	M	L	H	M	L	Ltd.	Ultd.	GA	GV	Var.	Sus.
Silver	●			●			● (L1)		●		●	
Gold		●			●		● (L2)		●			●
Platinum			●			●		●	●	●		●

Table 5.1: The three classes of service

As shown in table 5.1, the silver class offers the lowest QoS profile, with high call blocking and forced call termination probabilities, size-limited multiparty sessions, variable media quality, and non-guaranteed video communications. This class may be suitable for calls requiring limited/basic guarantees (e.g., routine or basic calls). Compared to the silver

class, the gold class brings improvements in terms of call blocking and forced call termination probabilities, as well as the media quality that is sustained by the network for this service class. As for gold multiparty sessions' size, there is still a limit, but this limit is higher than for the silver class (L2>L1). These enhancements could be useful for power users that have more demands from wireless communications (e.g. game players). Finally, the best priorities/guarantees can be obtained by using the platinum service class, which offers low call blocking and forced call termination probabilities, unlimited size for multiparty sessions, guaranteed audio and video communications, as well as sustained media quality. This category could be suitable for business calls and urgent communications among citizens. We note that these classes involve two forms of preemption: soft-preemption (i.e. network-initiated session downgrade from one type of media to another) and hard preemption (i.e. network-initiated session termination). Furthermore, the use of these classes depends on the user's subscription which should indicate the Highest Service Class (HSC) that the user is allowed to utilize. Any class up to and including the HSC can be chosen by the user for each call, and the selected class can be dynamically changed by the user during the session.

5.3 An IMS Call Differentiation Architecture

Figure 5.2 depicts the architecture we propose to achieve call differentiation in the IMS. This architecture introduces two new functional entities to the standard IMS architecture, namely: the Session Prioritization Function (SPF) and the Context Information Base (CIB). The CIB is a support entity that is responsible for the management of the contextual information needed for the operation of the SPF - the role of the CIB being realized by the extended PS that was presented in the previous chapter. The SPF is the entity that makes

resource allocation/re-allocation decisions, taking in consideration the contextual information it receives from the CIB, the sessions' QoS profiles, and the Resource Management Policies (RMP) that are set by network operators.

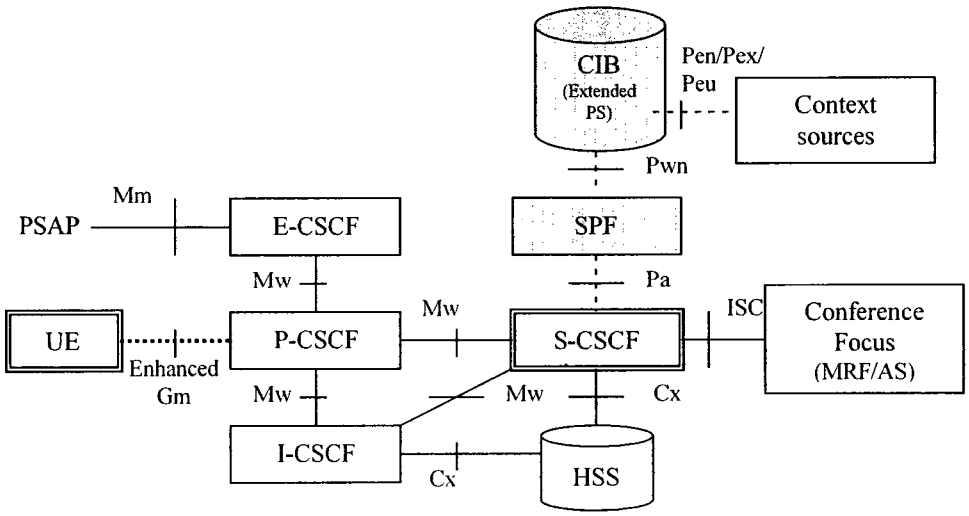


Figure 5.2: The IMS call differentiation architecture

In addition to the newly introduced entities, enhancements are made to two of the existing IMS entities, namely: the UE and the S-CSCF. The UE is enhanced with QoS negotiation capabilities including the ability to label session initiation requests with the appropriate session category and the ability to issue a session category change request. As for the S-CSCF, it is enhanced with the ability to communicate with the SPF for resource allocation decisions and the ability to take appropriate actions upon the receipt of triggers concerning the control of ongoing sessions (e.g. sessions' preemption and media parameters re-negotiation). Furthermore, some of the mechanisms previously supported by the S-CSCF are refined, such as the ability to authorize the service class requested by the user by checking his/her profile, and the ability to include information related to the call category and the occurring preemptive events as part of the charging information it sends to the charging related functions.

To enable the interaction between the new and the existing entities, two new interfaces are

introduced: the Pwn and the Pa interfaces. The Pwn interface is an outbound information exchange interface that is used for contextual information exchange (using queries and subscriptions/notifications) between the CIB and the SPF. As previously mentioned in chapter 4, SIMPLE [46] is the protocol used on this interface. The Pa interface, on the other hand, is used for the exchange of information related to resource allocation/re-allocation decisions between the SPF and the S-CSCF. The COPS protocol [30] is used on this interface since it is already supported in the IMS, is extensible, and provides a policy enforcement framework. According to the COPS framework, the S-CSCF acts as a policy enforcement point (PEP) and the SPF as a policy decision point (PDP), operating in the outsourcing mode. In addition to the interactions carried out via the new interfaces, other interactions related to QoS negotiation occur between the UE and the network over the enhanced Gm interface, using SIP and SIP extensions. In fact, since the differentiation occurs at the signaling level, the same protocol (SIP) is used for session control and QoS negotiation, which are combined together.

Figure 5.3 illustrates the architecture's general operation, in which call differentiation is achieved as follows: When the user initiates a call of certain category, the UE maps it to the appropriate service class then forwards the request (including the service class info) to the P-CSCF. This last forwards the request to the S-CSCF (allocated to the user), which checks the user profile to determine if he/she is entitled to the service type and service class requested. If the request is not authorized, it is rejected. Otherwise, the S-CSCF attempts to admit the call into the network by communicating with the SPF. If resources are available, the SPF renders a positive decision and the call is established normally as shown in figure 5.3a. If no resources are available, the SPF either rejects the call admission request

(resulting in the blocking of the call), or triggers one or several S-CSCFs to downgrade and/or preempt one or more ongoing calls as shown in figure 5.3b, in order to free resources for the new call that is admitted afterwards. After session establishment and depending on the session class, the SPF may trigger the S-CSCF to re-negotiate the session parameters (i.e. the media format used) in order to sustain its perceived quality. It should be noted the actual downgrade/preemption/renegotiation of a session is achieved by opening a referral session between the S-CSCF and one of the session's UE, which is instructed to contact the other UE to re-negotiate or terminate the session. A notification is sent about the referral event status to the S-CSCF, after the successful re-negotiation/termination of the session.

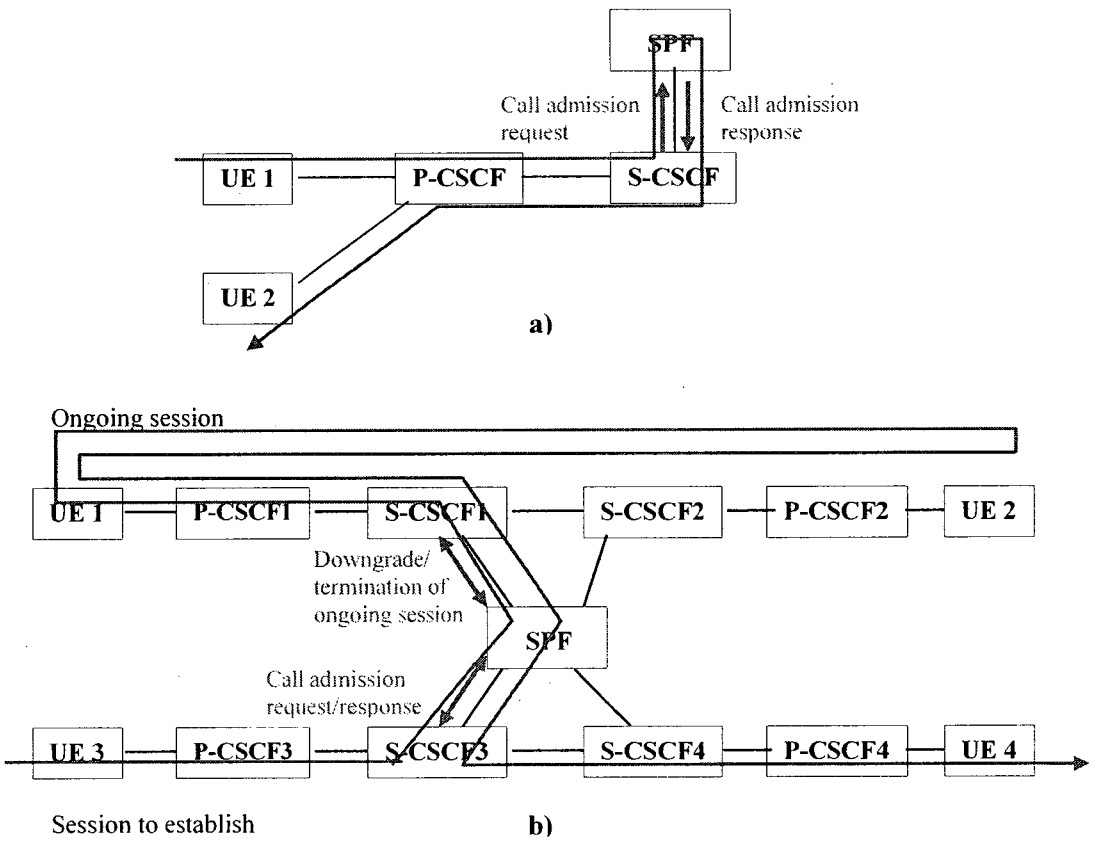


Figure 5.3: Illustration of the IMS call differentiation architecture's operation: a) Session initiation without control of ongoing sessions; b) Session initiation after downgrade/termination of an ongoing session

In the case of an attempt to join an ongoing multiparty session, the same procedure is followed. The only difference is that, in the case of a lack of resources, the SPF first checks the session size before attempting to free resources for the new request. If the size limit has been reached, the admission request is rejected. We note that this limit is a soft limit that is imposed in case of high load only. As for the case of a call category change, it is special in the sense that resources have already been allocated to the call, but a change in the guarantees on those resources is requested. In this situation, if the user is entitled to the new service class requested, the request is accepted and the session information is updated at the SPF and the S-CSCF.

The resource management techniques and charging mechanisms used in this architecture are presented in the next sections, followed by detailed session management scenarios illustrating the system's operation.

5.4 Resource Management Techniques

In this section, we present the resource management strategy used in our architecture and then detail the resource management techniques used to achieve this strategy.

5.4.1 The Resource Management Strategy

At the heart of any service differentiation solution, we find resource management techniques. Those techniques are used to allocate resources to the different classes, based on their QoS requirements. Examples of resource management techniques that have been proposed in the past include: admission control; rate/power control; queuing/scheduling, as well as policing/shaping. Among those techniques, call admission control is of particular interest since it can be used for signaling-level access control and sessions' prioritization. This section briefly discusses the existing call admission control strategies and pinpoints the strategy that will be used in this work.

Reference [89] presents an extensive survey of the call admission control schemes proposed for wireless networks. They are categorized based on the objectives they seek to achieve (e.g., signal quality control, dropping probability control, and revenue maximization). Reference [90] presents a more general discussion about the different resource allocation strategies/policies that can be implemented by admission control schemes, including: the complete sharing policy; the complete partitioning policy; the trunk reservation policy; the guaranteed minimum policy; and the upper limit policy. In general, these policies limit the access to resources by low priority classes in order to protect those resources for the high priority ones. The main concern about these approaches is that they reduce the utilization and revenue generation potential of networks by rejecting traffic from low priority users.

A radically different approach which is now being investigated is preemption [91]. A preemptive policy admits users to the network whenever resources are available, then interrupts the flows of low priority users (hard-preemption) or reduces the quality of their sessions (soft-preemption), to free resources for high priority users when there is no room to accommodate them. It has been shown that an optimal preemptive policy can use capacity more efficiently while being able to adapt to important variations in load by transferring/re-assigning resources between different classes [91]. The main issue with preemption is the irritation of low priority users that are preempted. However, this problem can be offset with the right incentives such as credits and lower subscription rates.

In this work, we combine the benefits of a conservative policy (the upper limit policy) and a preemptive policy to achieve call admission control. In fact, two resource management techniques are used in our architecture to enable preferential treatment at the beginning and

during sessions – *call admission control and media parameter control.* New calls (or requests to join ongoing calls) are granted/denied access to the core network by the call admission control mechanism, based on the call admission policy. The media parameter control mechanism is used to control the format of the media streams exchanged in (some of) the admitted sessions, in order to sustain their perceived media quality. The objective of these techniques is to maximize the resources utilization while satisfying the sessions' QoS profiles. Both techniques will be detailed in the coming sub-sections.

5.4.2 Call Admission Control Mechanism

The call admission control mechanism we propose implements a combination of the upper limit policy and the preemptive policy along with an overload prevention mechanism, as follows: as long as the network load remains within its planned value (i.e. light to regular load), the upper limit policy is used to limit the amount of resources that can be accessed by each class. This is accomplished by putting a threshold over the amount of resources that can be accessed by each class. When the load exceeds its planned value (i.e., in high load or in crisis situations), session size control is first attempted for overload prevention/ reduction, then a mix of soft and hard preemption is used to enable the adaptation to this high load condition by transferring resources between different classes. It should be noted that, only silver and gold sessions are subject to session size control in high loading conditions. Furthermore, preemption (either hard or soft) is carried in a prioritized manner. This implies that lower priority calls are preempted first, while high priority calls are preempted last; noting that a session can trigger the preemption of one or more lower priority sessions (e.g. one platinum multiparty session triggering the downgrade of 2 silver sessions and 1 gold session, as well as the termination of 1 silver session). Within the same

category of calls, sessions of small size are preempted first and larger sessions last, in order to free only as much resources as needed (not more) and minimize the disturbance of participants to large sessions – this implying that larger sessions are considered more important than smaller sessions in our scheme.

In the case of call category change request, if the load is within its planned value, a check is made to ensure that the maximum amount of resources allocated for the new service class won't be exceeded (due to the call category change). However, in high loading conditions or crisis situation, only category downgrades (e.g., from gold to silver) are permitted.

Three algorithms are used as part of the mechanism described, namely: a threshold-based admission algorithm, a preemption-based admission algorithm, and a session size control algorithm. Figure 5.4 details these algorithms and illustrate their roles in the call admission control mechanism.

Def.: S_{IT} = A session S of class I and type T;
Where: I: 1 = platinum; 2=gold; 3= silver;
T: 1 = audio; 2=audio/video; 3 = text; 4 = audio/text; 5=audio/video/text;
L_I = Size limit of sessions of class I
b_{x_y} = B.W. required by session of class x and type y
n_{x_y} = Number of active sessions of class x and type y
C = Network capacity (in terms of B.W.)
$Thresh_I$ = limit on amount of B.W. that can be used by sessions of class I
Δb_{x_y} = Amount of released B.W. due to downgrade of session of class x & initial type y

```
if (Current_load < Threshold_high-load)
        then Apply_upperLimitPolicy (S_IT);
        else {
            if ((I =2 || I=3) && (current_size (S_IT)+1>L_I)) //Session size control algorithm
                then Reject (S_IT);
              else Apply_preemptivePolicy (S_IT);
          }
Apply_upperLimitPolicy (S_IT) //Threshold-based admission algorithm
{
if ((\sum_{I=1}^{3}\sum_{T=1}^{5} b_{I_T} n_{I_T} + b_{I_T} \le C) && (\sum_{T=1}^{5} b_{I_T} n_{I_T} + b_{I_T} \le Thresh_I))
  then  Accept (S_IT);
  else  Reject (S_IT);
} // End Apply_upperLimitPolicy
```

```
Apply_preemptivePolicy (S_IT) //Preemption-based admission algorithm
{
Status = pending;
Potential_downgrade_resouces = 0 ;
Potential_preemption_resouces = 0 ;
List_downgradable_sessions = empty;
List_preemtable_sessions = empty;

for (K=3; K ≥ I+1; K=K-1) //Prioritized soft-preemption
{
    for (S = 2; S ≤ max_session_size_inNet ; S = S+1)
  {
  If ( ∃ 1 or n calls of class K and size S that can be downgraded such that
        Σ_{j=1}^{n} Σ_{t=1}^{5} Δb_{K,,} + potential_downgrade_resources ≥ b_{I_T} )
    then {
            Downgrade (selected calls + List_downgradable_sessions);
          Accept (S_IT);
          Status = accepted;
          Break;
          }
         else {
               Update (List_downgradable_sessions);
               Update (Potential_downgrade_resouces);
              } //end else
      }
}
If (Status = pending)   // Soft preemption resources are not sufficient, attempt hard preemption
as well
for (K=3; K ≥ I+1; K=K-1) //Prioritized hard-preemption
{
    for (S = 2; S ≤ max_session_size_inNet; S = S+1)
  {
  If ( ∃ 1 or n calls of class K and size S that can be terminated such that
        Σ_{j=1}^{n} Σ_{t=1}^{5} b_{K,,} + potential_preemption_resources + potential_downgrade_resources ≥ b_{I_1} )
    then {
            Terminate (selected calls + List_preemtable_sessions) && downgrade
       (List_downgradable_sessions);
          Accept (S_IT);
          Status = accepted;
          Break;
          }
         else {
               Update (List_preemtable_sessions);
               Update (Potential_preemption_resouces);
              } //end else
      }
   }
If (Status = pending) then Reject (S_IT); //Both soft and hard preemption are insufficient
} // end Apply_preemptivePolicy
```

Figure 5.4: The call admission control mechanism

5.4.3 Media Parameter Control Mechanism

Media parameter control consists of modifying the characteristics of some of the ongoing sessions (in terms of media format/codec used) in response to important variations in the network capacity, by triggering the re-negotiation of the session parameters between the involved parties. Figure 5.5 shows the details of the media parameter control mechanism, which operates as follows: When an important drop in the network capacity is detected, a format downgrade procedure is invoked to trigger the re-negotiation of the session parameters to a media format that suits the network situation (i.e., a format that consumes fewer resources). The goal of this procedure is to prevent the degradation of the session's performance in terms of user perceived quality, reducing by the same token the session's load on the network. Upon a re-increase in the network capacity, a format upgrade procedure is invoked to restore the initial session characteristics. It should be noted that, several degrees of downgrade/upgrade may be used, depending on the media codecs supported by the terminals. Contrarily to soft and hard preemption events, format upgrade/downgrade events are performed transparently to end-users, to avoid their disturbance by the frequent upgrade/downgrade notifications.

```
Def.: List_downgraded_sessions = empty;

Format_Downgrade () //Single degree of format downgrade due to drop in network capacity
{
If ( ∃ sessions of class gold || platinum that can be downgraded)
    then {
    Perform 1 degree of downgrade to sessions;
    update (List_downgraded_sessions, downgrade_degree);
    }
}
Format_Upgrade () //Single degree of format upgrade after restoration of network capacity
{
If (List_downgraded_sessions != empty)
    then {
    Perform 1 degree of upgrade to List_downgraded_sessions;
    update (List_downgraded_sessions, downgrade_degree);
    }
}
```

Figure 5.5: The media parameter control mechanism

5.5 Charging Aspects

Charging for multi-grade differentiated services involves two aspects: allocating suitable prices to calls of different grades/classes; and considering the effect of preemptive events (i.e. lower priority sessions being downgraded or terminated to free resources for high priority sessions) on the charging model. In the coming sub-sections, we present the approach we propose to deal with these aspects.

5.5.1 The Proposed Charging Model

In terms of pricing, all the service classes presented are assigned different charging rates, depending on the priorities/guarantees they offer. This price differentiation would provide the required incentive to users to select for each call the service class that suits their true needs, leading by the same token to a more efficient utilization of network resources and a more uniform distribution of calls among the different classes. Furthermore, this differentiation can be translated in an increase in network operator's revenues, since resources are sold for higher prices to users with more urgent needs. To achieve this price differentiation, information about the *service class* must be conveyed to the charging system to enable the calculation of the appropriate price for each session.

Two different approaches can be used to deal with preemptive events. The first approach is to embed the effect of preemption in the price per minute. This implies that an average minute rate that takes into consideration the potential preemptive events that could occur, is calculated and applied to all the calls in a certain category. This approach requires a suitable mathematical model to calculate the optimal price per minute, under different loading conditions and taking into consideration predicted preemption patterns/trends. The second approach is a more refined approach in which callers with preempted calls (i.e.

downgraded or abruptly terminated calls) pay a bit less that other callers (in the same category) with non-preempted calls. This can be achieved by giving a small compensation credit (that can be redeemed later on) for each preemptive event, thus reducing the total price for the session. In this case, the following pieces of information must be conveyed to the charging system to enable the calculation of the number of compensation credits that will be allocated to the user: the *service class*; the type of *event triggering the credit* (i.e. soft/hard preemption); the *session duration before preemption*; and the *caller and callee's identities*. A mathematical formula can be used to calculate the exact number of credits, based on this information, such as the following:

$$Credit = Cf * PPM_{SC} * PEff * TPE \quad (1)$$

Where: Cf ≡ Compensation Factor (a constant representing a % of the price per minute)

SC ≡ Service Class [Platinum = 1; Gold = 2; Silver = 3]

PPM_{SC} ≡ Price per minute for a certain service class

TPE ≡ Type of preemptive event [soft-preemption = 1; hard-preemption = 1.5] (a weight for each type of preemptive event)

TCD ≡ Total call duration in minutes

ACD ≡ Average calls duration in minutes

DBP ≡ Call duration before preemption in minutes

PEff ≡ Preemption effect (an estimation of the # of minutes affected by preemption)

$$= \begin{cases} TCD - DBP & \text{if TPE} = 1 \\ |ACD - DBP| \bmod ACD & \text{if TPE} = 1.5 \end{cases}$$

Although this second approach adds some communication overhead (i.e. more information to convey and a mechanism for credit allocation), it achieves fairness among users of the same category (since callers of preempted calls pay less than callers with non-preempted calls), and gives an additional incentive for users to choose lower service classes despite their higher preemption probability. In the remaining part of the chapter, we will focus on this second approach.

5.5.2 Charging Model Realization in the IMS

In the IMS, the pricing of sessions and the allocation of compensation credits to users are carried as follows: For offline charging scenarios, the regular offline charging mechanism, augmented with information about the service class, is used for session initiation, modification, and termination. Furthermore, referral sessions triggering sessions' downgrade/termination, are pre-configured as chargeable events in the S-CSCF. These chargeable events trigger the generation of CDRs containing all the necessary information about the preemptive events, to be able to credit the users in the billing domain. For online charging scenarios, the SCUR model (enhanced with information about the service class) is used for credit control interactions related to session initiation, modification, and termination. As for preemptive events, iFC are set in the users' profiles to indicate to the S-CSCF that the IMS-GWF should be put on the signaling path for referral sessions. Moreover, the IMS-GWF is pre-configured to consider the notification message (indicating the success of the referral session) as chargeable event that triggers the interactions needed to refund the user account with the calculated number of compensation credits. The IEC model is used to perform a direct refund operation on the user account, since the preemptive event can be considered as a single event occurring within the ongoing session, and which does not require the maintenance of a credit control state.

It should be noted that we leveraged existing 3GPP/Diameter AVPs and CDR fields to carry the additional information needed for price differentiation and compensation credits calculation. This information is formatted according to the standard AVPs definitions [92, 93] and CDR format [94]. In fact, the *service class* information (needed for price differentiation) is inserted in the Diameter *Service-Parameter-Info(SPI)* AVP of ACR

and CCR messages, as in the following example: SPI{[Service-Parameter-Type = service class]; [Service-Parameter-Value = gold]}. For offline charging scenarios, this information is mapped into a corresponding value and inserted in the *Record-extensions* field of generated CDRs (e.g. Records extensions {service class = gold}). As for preemptive events related interactions, in addition to the *service class*, the *SPI* AVP *a*lso includes information about the *service duration* (e.g. SPI{[Service-Parameter-Type=service class]; [Service-Parameter-Value=gold]; {[Service-Parameter-Type=service duration]; [Service-Parameter-Value=180 sec]}. Furthermore, the 3GPP E*vent-Type (ET), Called-Party-Address,* and *Calling-Party-Address* AVPs respectively carry information about the *event triggering the credit* (e.g. ET=soft_preemption), *the caller, and callee's identities* (e.g. CPA=sip:alice@home.net). For offline scenarios, these parameters are mapped onto corresponding CDR fields as follows: the *service class* and *duration* are inserted in the *Records-extensions* field; the *Event-Type* AVP is mapped into the *Event* field; while the *Called-Party-Address* and *Calling-Party-Address* AVPs are mapped onto the *List-of-called-party-address* and *Requested-Party-Address* fields respectively.

5.6 Session Management Scenarios

The scenarios presented in the section illustrate how SIP and COPS can be used to achieve signaling level access control and call differentiation within a single administrative domain. They also demonstrate how Diameter can be used for the charging/billing of differentiated services.

We start with two-party sessions related scenarios, and then tackle the multiparty case. The main difference between these two cases is that two-party sessions are established

between two UEs, while multiparty sessions (also know as conferences) are established between three or more participants, by the intermediary of a centralized control point (the conference focus). In both cases, the S-CSCFs servicing the different UEs (engaged in either two-party or multiparty sessions) communicate with the SPF for call admission decisions, and receive triggers concerning the control of ongoing sessions. We end the section with the charging related scenarios.

5.6.1 Two-Party Sessions Scenarios

5.6.1.1 Session Initiation without Control of Ongoing Sessions

Figure 5.6 depicts a successful two-party session initiation scenario that is carried without any attempt to control ongoing sessions. In this scenario, UE1 attempts to establish a session of a certain category with UE2. It sends a SIP INVITE message with a resource-priority header set according to the session category to its P-CSCF (i.e. P-CSCF1). The resource-priority header is a SIP extension that has been proposed to indicate sessions' priority in terms of access to SIP-signaled resources [95]. In our case, sessions are assigned the following priority values according to the Q.735 namespace: *platinum=Q735.1; gold= Q735.2; and silver= Q735.3*. The P-CSCF ignores the resource-priority header and forwards the request to the assigned S-CSCF (i.e., S-CSCF1), which checks the user profile to ensure his/her entitlement to the service requested. Afterwards, S-CSCF1 sends the SPF a call admission request using a COPS REQ message (including the session info). In this case, the call is admitted and an "install" decision is sent to S-CSCF1 (using a COPS DEC message). This last inserts a tag in the record-route header indicating the address of the SPF that has admitted the request (e.g. Record-Route: <sip:scscf1.home1.net; lr; admitted_by=spf.home1.net>,

<sip:pcscf1.home1.net;lr>), then forwards the INVITE to the callee's S-CSCF (i.e., S-CSCF2), via the local I-CSCF. After checking the admission tag, S-CSCF2 determines that the call has already been admitted in this domain (i.e., call originating and terminating in the same network), stripes off this tag to prevent the leakage of sensitive information, and carries the session establishment procedure normally – Noting that in the case where the call originates and terminates in different networks, the terminating S-CSCF should contact its local SPF for call admission in its domain. Following the session establishment, S-CSCF1 returns a COPS RPT message indicating that it has enforced the SPF decision.

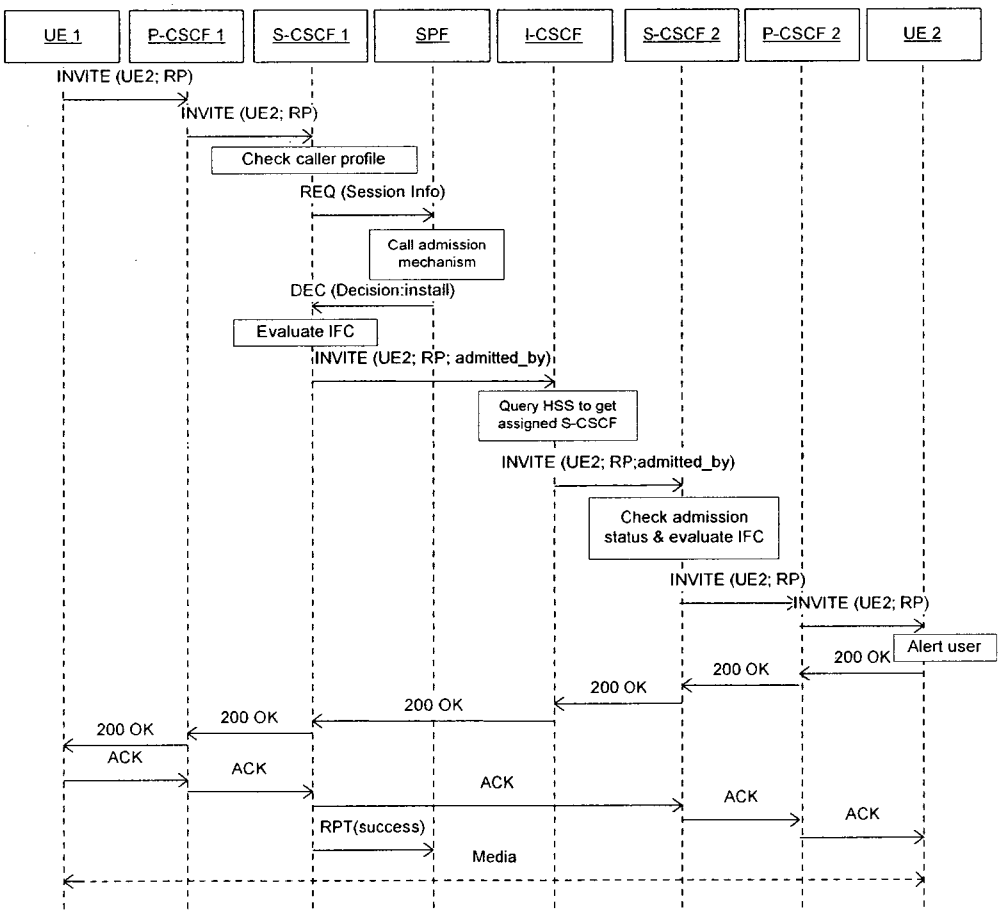
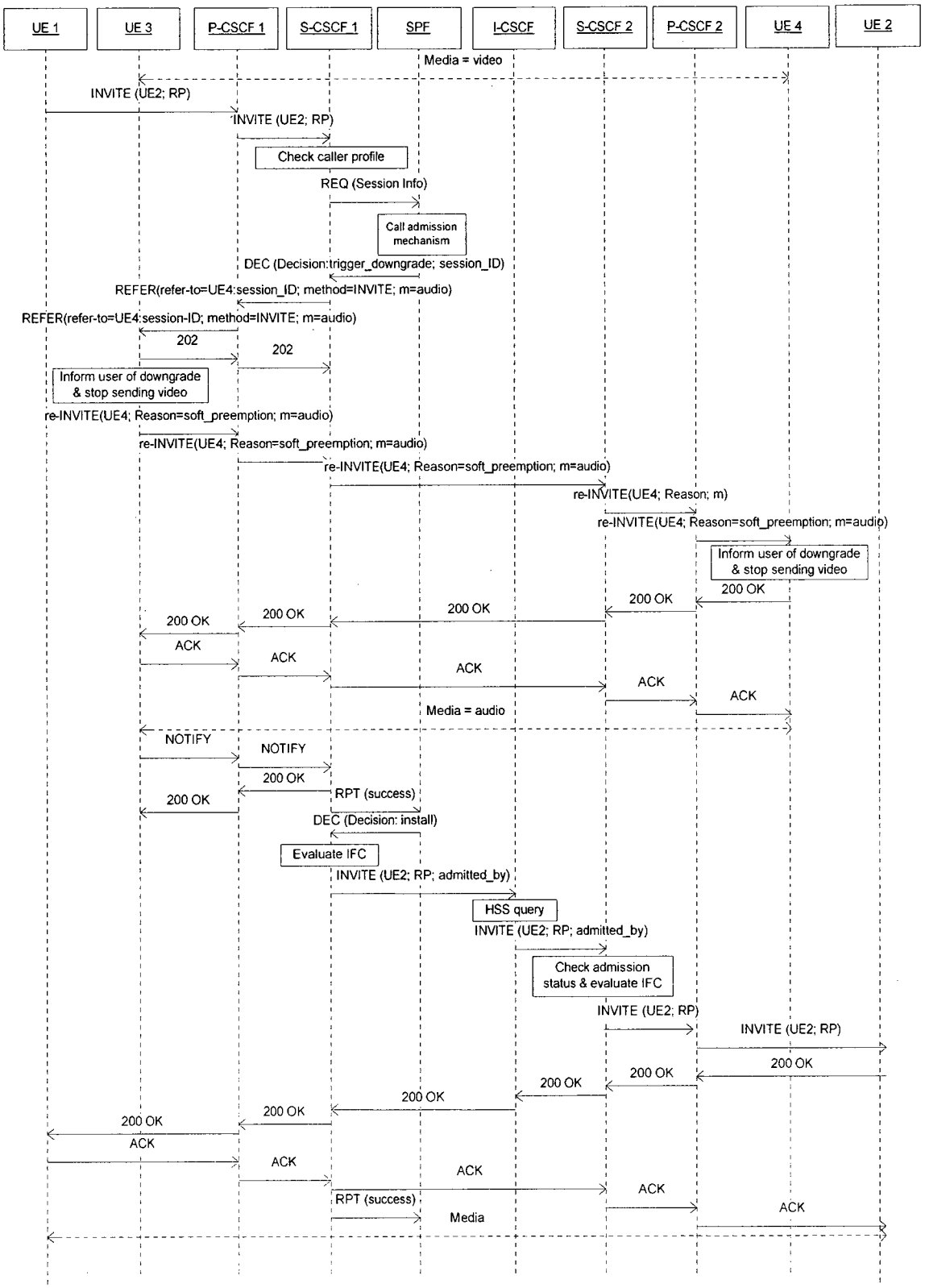


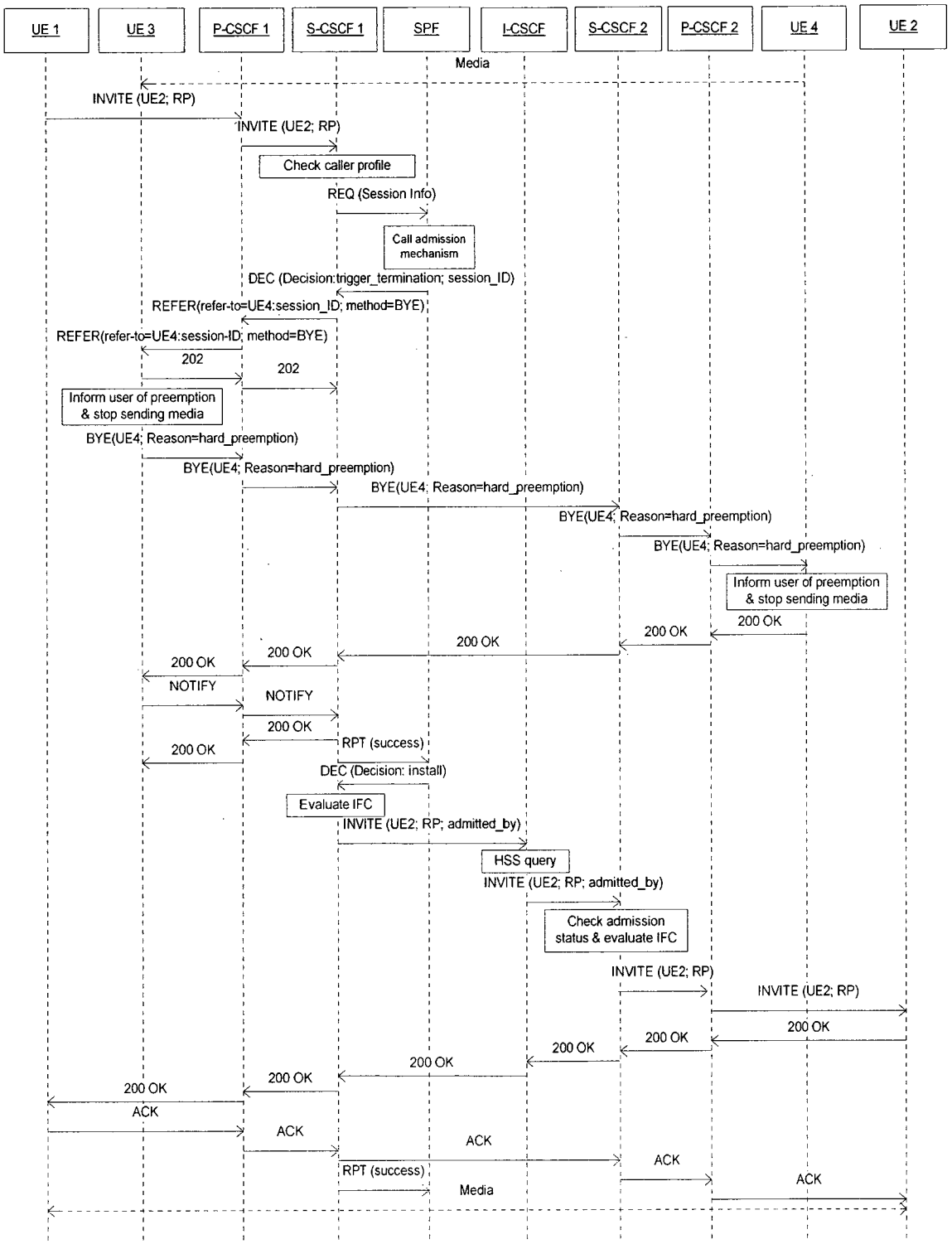
Figure 5.6: Successful two-party session initiation without attempt to control ongoing sessions

5.6.1.2 Session Initiation after Downgrade/Termination of an Ongoing Session

Figure 5.7a illustrates the case where an ongoing session (a video session between UE3 and UE4 in this case) must be downgraded in order to free resources for the new session to be established (between UE1 and UE2). We assume that UE1 and UE3 are serviced by P-CSCF1 and S-CSCF1, while UE2 and UE4 are serviced by P-CSCF2 and S-CSFC2. In this case, before admitting the new session, the SPF sends a "trigger downgrade" decision to S-CSCF1 in order to modify the decision made about the (previously admitted) session between UE3 and UE4. After receiving this decision, S-CSCF1 sends a SIP REFER message instructing UE3 to contact UE4 in order to renegotiate the parameters of the session (from video to audio). UE3 carries this instruction by sending a SIP re-INVITE to UE4, containing "audio" as new media type, and a "reason" header indicating soft preemption as a reason for the re-negotiation. An extension of the reason header has been proposed to indicate preemptive events, in reference [96]. After the session is re-negotiated successfully, UE3 sends a notification to its S-CSCF (using a SIP NOTIFY message). This last informs the SPF, which authorizes the admission of the new call. The case of hard preemption is similar as shown in figure 5.7b, except that an ongoing session is terminated to allow the establishment of the new session.


UE 1
UE 3
P-CSCF 1
S-CSCF 1
SPF
I-CSCF
S-CSCF 2
P-CSCF 2
UE 4
UE 2
Media = video
INVITE (UE2; RP)
INVITE (UE2; RP)
Check caller profile
REQ (Session Info)
Call admission mechanism
DEC (Decision:trigger_downgrade; session_ID)
REFER(refer-to=UE4:session_ID; method=INVITE; m=audio)
REFER(refer-to=UE4:session-ID; method=INVITE; m=audio)
202
202
Inform user of downgrade & stop sending video
re-INVITE(UE4; Reason=soft_preemption; m=audio)
re-INVITE(UE4; Reason=soft_preemption; m=audio)
re-INVITE(UE4; Reason=soft_preemption; m=audio)
re-INVITE(UE4; Reason; m)
re-INVITE(UE4; Reason=soft_preemption; m=audio)
Inform user of downgrade & stop sending video
200 OK
200 OK
200 OK
200 OK
200 OK
ACK
ACK
ACK
ACK
ACK
Media = audio
NOTIFY
NOTIFY
200 OK
200 OK
RPT (success)
DEC (Decision: install)
Evaluate IFC
INVITE (UE2; RP; admitted_by)
HSS query
INVITE (UE2; RP; admitted_by)
Check admission status & evaluate IFC
INVITE (UE2; RP)
INVITE (UE2; RP)
200 OK
200 OK
200 OK
200 OK
200 OK
ACK
ACK
ACK
ACK
ACK
RPT (success)
Media

a)



b)

Figure 5.7: Successful two-party session initiation scenarios after control of an ongoing session: a) Session initiation after downgrade of an ongoing session; b) Session initiation after termination of an ongoing session

5.6.1.3 Session Category Change

The case of successful session category change is carried like the first scenario. The only difference is that instead of issuing a new INVITE request, the UE requesting a call category change issues a re-INVITE with the new call category, to re-negotiate the parameters of the session (from "category 1" to "category 2"). Figure 5.8 illustrates this scenario.

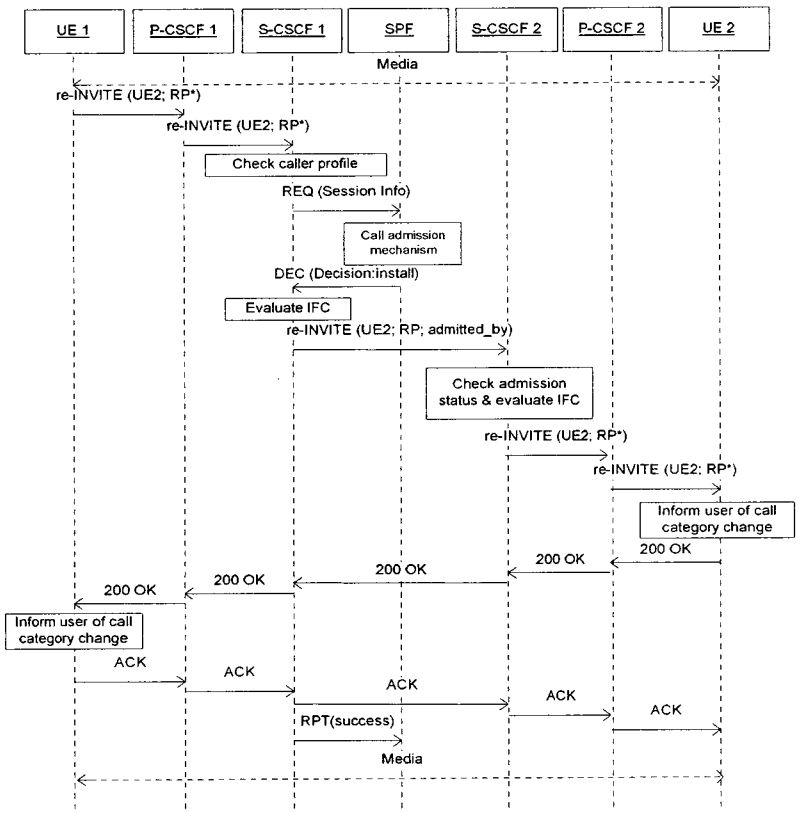


Figure 5.8: Successful two-party session category change scenario

5.6.1.4 Exception Scenarios

There are several unsuccessful session initiation scenarios. The first scenario is when the SPF determines that there are not enough resources and that no alternative actions can be taken to free some resources for the new session. In this case, a "remove" decision with the appropriate flag is sent to the S-CSCF, which sends a 488 "Not acceptable here"

response message with an "insufficient resources" warning header to the UE (via the P-CSCF). After failure of the operation, the UE may wait for an arbitrary amount of time then try again. The second case is when a UE tries to establish a session that violates his/her subscription profile (e.g., a higher service class than what was subscribed for). In this case, the S-CSCF sends a 403 "Forbidden" response which is relayed to the UE. After receiving this response, the UE should give up and not try again. We should note that the SPF is not contacted in this scenario, since a "forbidden" session request should not be considered for admission. The third scenario is when the S-CSCF does not recognize the resource priority value (or namespace). A 417 "unknown resource priority" response carrying the list of the acceptable priority values (in the accept resource priority header) is retuned to the UE, in this case. This last could try to re-initiate the session, with one of the acceptable priority values. The last failure scenarios occur when the callee doesn't answer or is busy with another call. Regular SIP procedures are followed in these cases. However, in order to maximize the chances of priority call establishment, we suggest the following: If the callee doesn't answer but has subscribed to a call redirection service, the call should be diverted to an alternate party. If the callee is busy with a lower priority call, this call could be preempted by the callee's UE in order to allow the establishment of the new call. However, this necessitates the implementation of a preemption mechanism at the UE level.

5.6.2 Multi-Party Sessions Scenarios

The standard IMS conferencing architecture [97] follows a centralized model in which the conference focus acts as conference manager, maintaining a signaling link with each participant. At the beginning, one of the participants creates a conference room, which

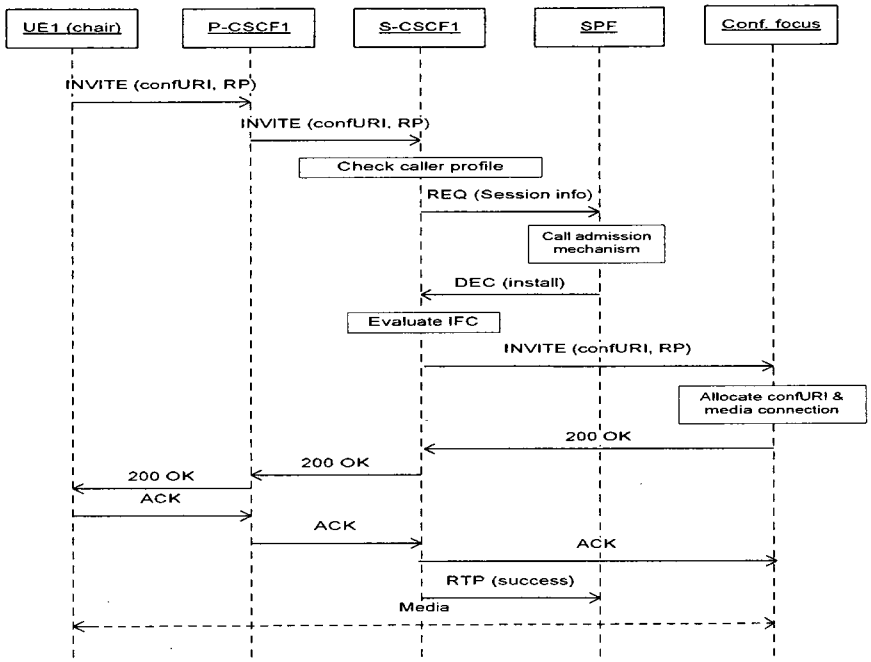
can be joined by other participants afterwards. Users can join by sending an invitation to the conference focus (either directly, or after being referred by one of the participants). They can also subscribe to the conference notification service provided by the focus (to learn about changes in the conference state), and may leave the conference when they wish. Furthermore, the focus can terminate or re-negotiate a dialog with a participant based on the instructions of authorized users (e.g., a conference chair).

We will now present some scenarios illustrating how our call differentiation scheme can be applied to conferencing. In these scenarios, we assume that there is a conference chair in each conference, although this role could be played by any participant with enough privileges (as specified in the conference policy). We also assume that the chair is the one creating the conference room.

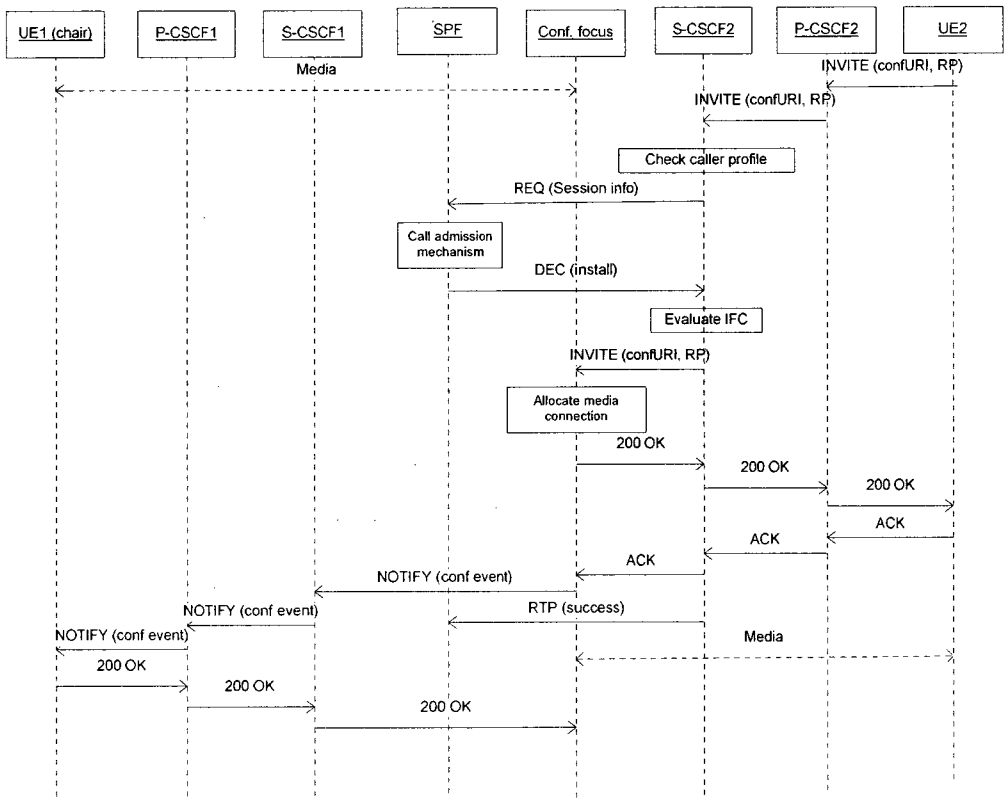
5.6.2.1 Conference Room Creation and Joining of an Ongoing Conference

In the case of conference room creation that is depicted in figure 5.9a, the chair sends an INVITE message specifying the session category (in the resource priority header) to its P-CSCF. The P-CSCF forwards the message to the S-CSCF, which tries to admit the request by communicating with the SPF, after checking the caller profile. Assuming that the request is admitted, the S-CSCF forwards the INVITE to the conference focus, which carries the rest of the procedure normally. Finally, a COPS RPT message is sent to the SPF (by the S-CSCF) to indicate the success of the operation.

The case where a user tries to join the conference is carried in the same fashion as shown in figure 5.9b; noting that the same category must be used for all participants. This category can only be changed by the chair (or an authorized user), during the session.



a)



b)

Figure 5.9: Multi-party sessions related scenarios: a) Conference room creation scenario; b) joining of an ongoing conference scenario

5.6.2.2 Conference Category Change by Chair

Figure 5.10 depicts the case of successfully session category change, by the conference chair. This last sends a SIP re-INVITE message indicating the new session category requested, in the resource-priority header. This message is relayed to its S-CSCF (i.e., S-CSCF1), which checks the caller's profile, then contacts the SPF for admission decision. In this case, the request is authorized, and the re-INVITE message is relayed by S-CSCF1 to the conference focus. After receiving this message, the focus updates the session state, returns a 200 OK message to the chair, then sends notifications to the rest of the participants (using SIP NOTIFY messages) indicating an update in the conference category.

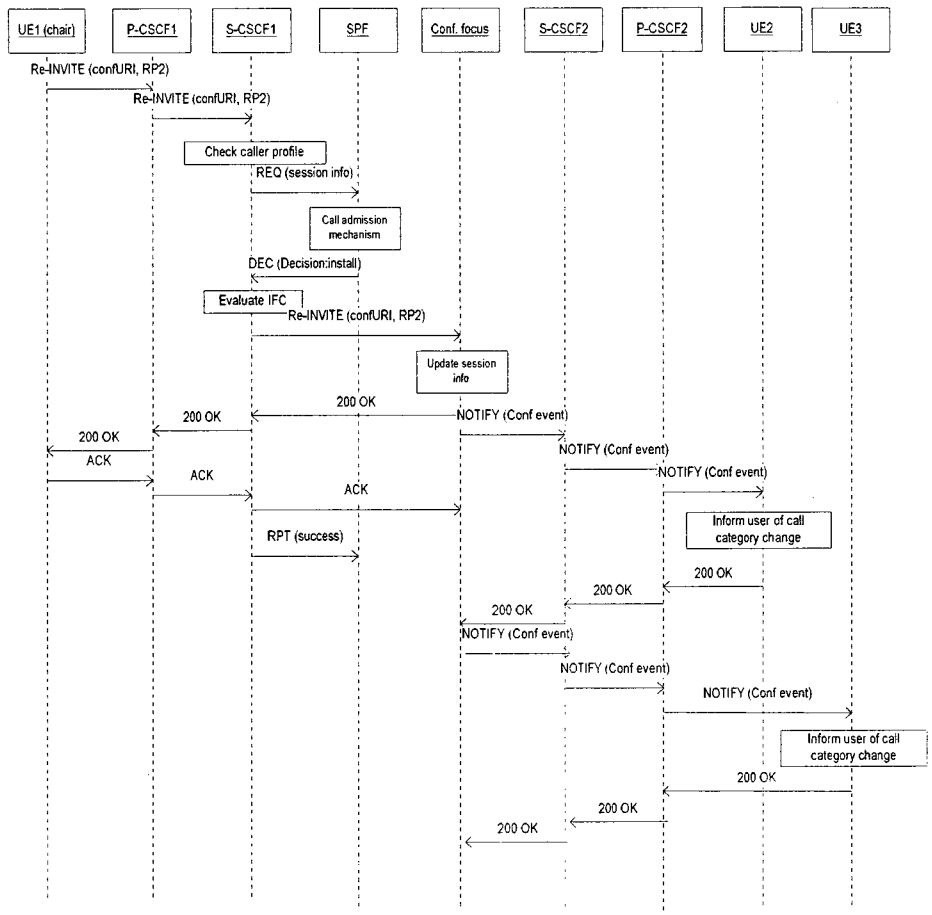


Figure 5.10: Successful conference category change scenario by chair

5.6.2.3 Network-Initiated Downgrade/Termination of an Ongoing Conference

A conference may be downgraded or terminated (either partially or fully), to free resources for a new session to be established. Figure 5.11 illustrates the case where an ongoing conference (a conference between UE1 (which acts as chair), UE2, and UE3) is fully terminated in order to free resources for a new session. In this scenario, a call admission request is received by the SPF, triggering the termination of the conference. As a result, the SPF sends a "trigger_termination" decision specifying the confURI, and the scope of the preemption (full-session in this case) to the S-CSCF servicing the conference chair (i.e., S-CSCF1). After receiving this message, S-CSCF1 sends a SIP REFER message to the conference chair, instructing it to contact the conference focus in order to terminate the session. The conference chair sends a 202 SIP response indicating its acceptance of the referral event, then sends another SIP REFER message to the conference focus, instructing it to terminate the signaling links its has with all the other participants. The focus carries this instruction by sending BYE SIP messages to UE2 and UE3, containing reason headers indicating "hard-preemption" as reason for the termination. After the termination of each signaling link, the focus informs the chair using a SIP NOTIFY message. After receiving those notifications, the chair terminates the link it has with the conference focus (using a SIP BYE message), then sends s SIP NOTFIY message to S-CSCF1, with the results of the preemption operation. Finally, S-CSCF1 returns a COPS RPT message indicating that is has complied with the SPF decision. The case of soft preemption is similar, except that the session parameters are re-negotiated (e.g., from video to audio) with each of the participants.

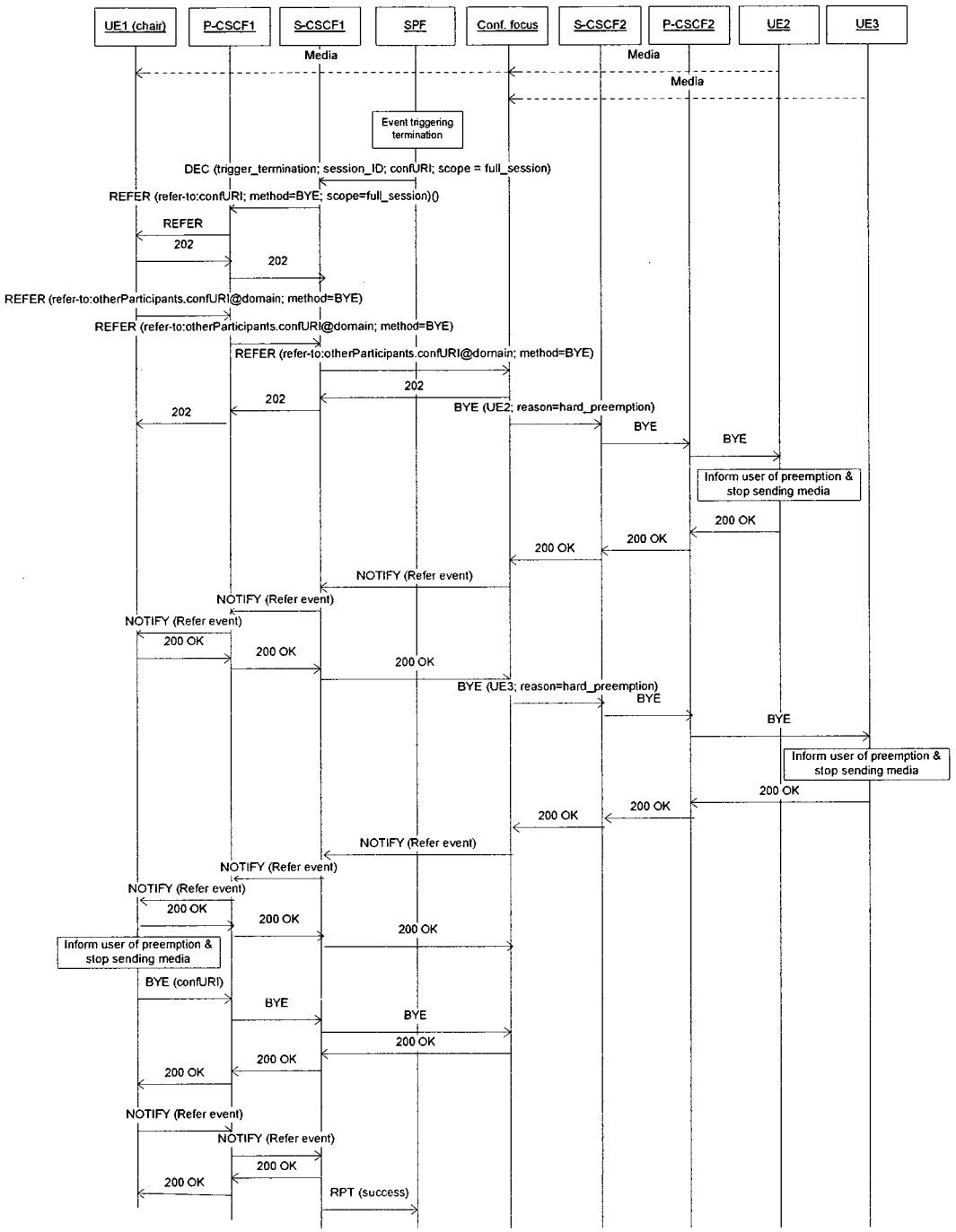


Figure 5.11: Network initiated termination of ongoing conference – full scope

5.6.2.4 Exception Scenarios

All the exception scenarios described in section 5.6.1.4 apply to the case of multiparty sessions, in addition to two other scenarios which are specific to the multiparty case. The

first scenario occurs when a user tries to join an ongoing multiparty session, while its size limit has been reached. In this case, a "remove" decision with the appropriate flag is sent to the S-CSCF, which sends a 488 "Not acceptable here" response message with a "size limit reached" warning header to the UE (via the P-CSCF). In this case, the UE may wait a certain amount of time before trying again. The second case is when a non-authorized participant tries to modify the session category. In this case, a "remove" decision with the appropriate flag is sent to the S-CSCF, which sends a 403 "Forbidden" response to the UE. After receiving this response, the UE should give up and not try again.

5.6.3 Charging Related Scenarios

In this section, we detail some of the offline and online charging scenarios in order to illustrate the operation of charging model proposed.

5.6.3.1 Offline Charging Scenarios

Figure 5.12 depicts three phases on an offline-charged video session, namely: session establishment (without control of ongoing sessions); network-initiated downgrade of the session; and UE-initiated session release.

In the first phase, UE1 attempts to initiate a video session with UE2 by sending a SIP INVITE message, carrying a resource priority (RP) header, to its P-CSCF (i.e. P-CSCF1). This last forwards this message to S-CSCF1 that sends the SPF a call admission request. In this case, the call is admitted and the invitation is sent to UE2, going through the I-CSCF, S-CSCF2, and P-CSCF2. After the call is accepted by the callee, a 200 OK SIP message (also including the RP header) is sent back to UE1, going through the same signaling path. This 200 OK message is recognized as a chargeable event by the two proxy and the two

serving CSCFs, which each send an ACR of type [start] carrying the service class information (extracted from the RP header of the 200 OK message) in its SPI AVP, to the CDF, to report this charging information. This information is used by the CDF for the generation of the corresponding CDRs, encapsulating the service class information (in their Records-Extension fields) that will be used latter on for the calculation of the correct price for the session, based on its category. Upon the receipt of the 200 OK message, UE1 returns a SIP ACK message to UE2, thus completing the session establishment.

The second phase starts when the SPF determines that the previously established session (between UE1 and UE2) must be downgraded, to free resources for another session. In this case, the SPF sends a "trigger_downgrade" decision to S-CSCF1 that instructs UE1 to downgrade the session. UE1 carries the session downgrade by sending a SIP re-INVITE message to UE2, to renegotiate the session parameters (from video to audio). After relaying the 200 OK message acknowledging this re-INVITE to the next node, each proxy and serving CSCF sends an ACR of type [update] (carrying the service class in its SPI AVP) to the CDF, to report this modification of the session. After the successful downgrade of the session, UE1 sends a notification to S-CSCF1, which sends an ACR of type [event] reporting the occurrence of this preemptive event, to the CDF. This ACR[event] carries five pieces of information needed for the calculation of the compensation credit (i.e the service class/duration (in the SPI AVP); the preemptive event type (in the ET AVP); and the calling/called party addresses (in the Calling-Party_address and Called-Party_address AVPs)), which are mapped into corresponding fields in the new S-CSCF CDR that will be created. This CDR will be processed latter on to allocate a compensation credit to the user.

In the third phase, UE1 terminates the session by sending a SIP BYE message (including

the RP header) to UE2. The receipt of this message triggers the proxy and serving CSCFs to send ACRs [stop, SPI[service class]] to the CDF, which closes the corresponding CDRs.

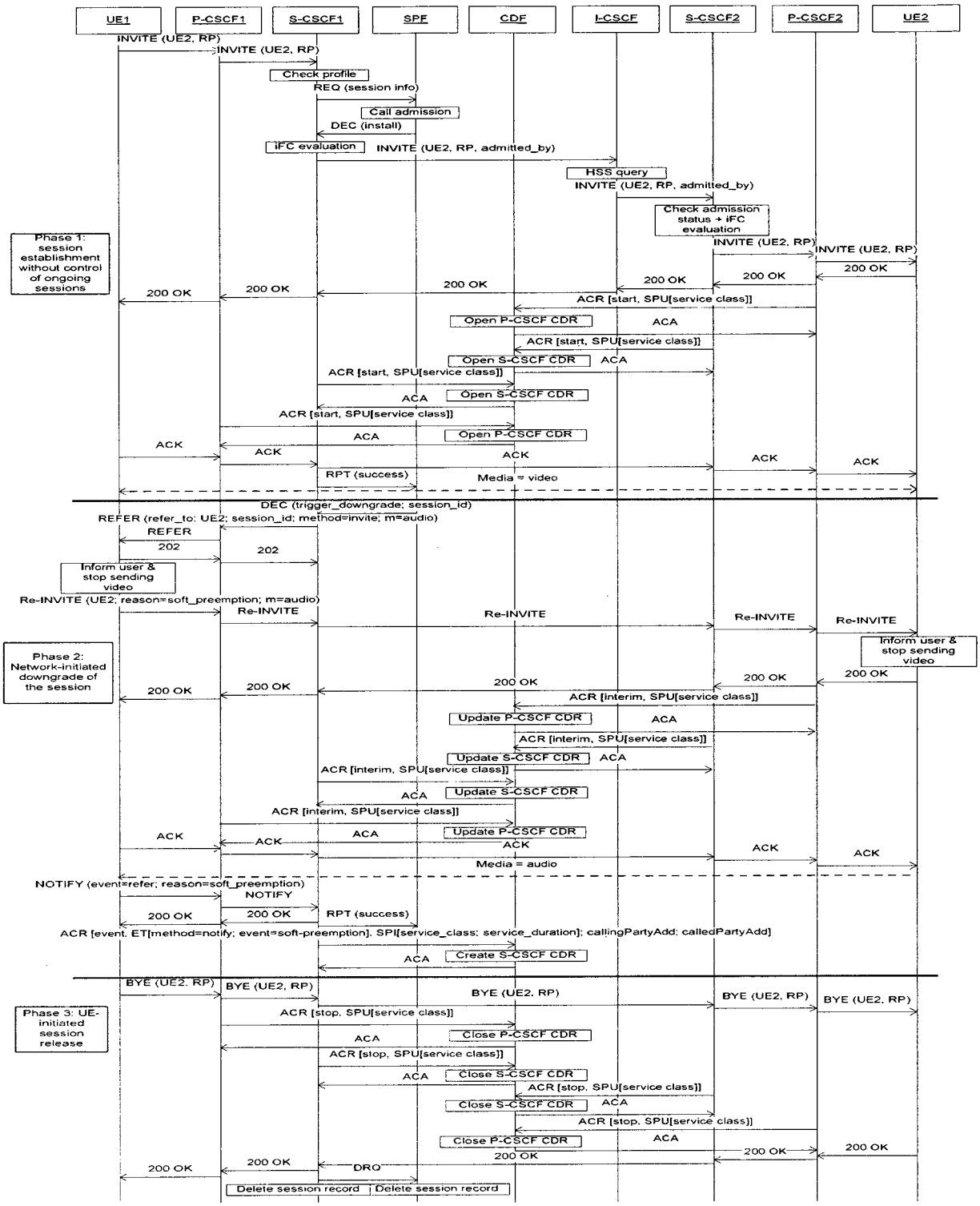


Figure 5.12: Offline charging scenarios

5.6.3.2 Online Charging Scenarios

Figure 5.13 shows the initiation and network-triggered termination of an online-charged session.

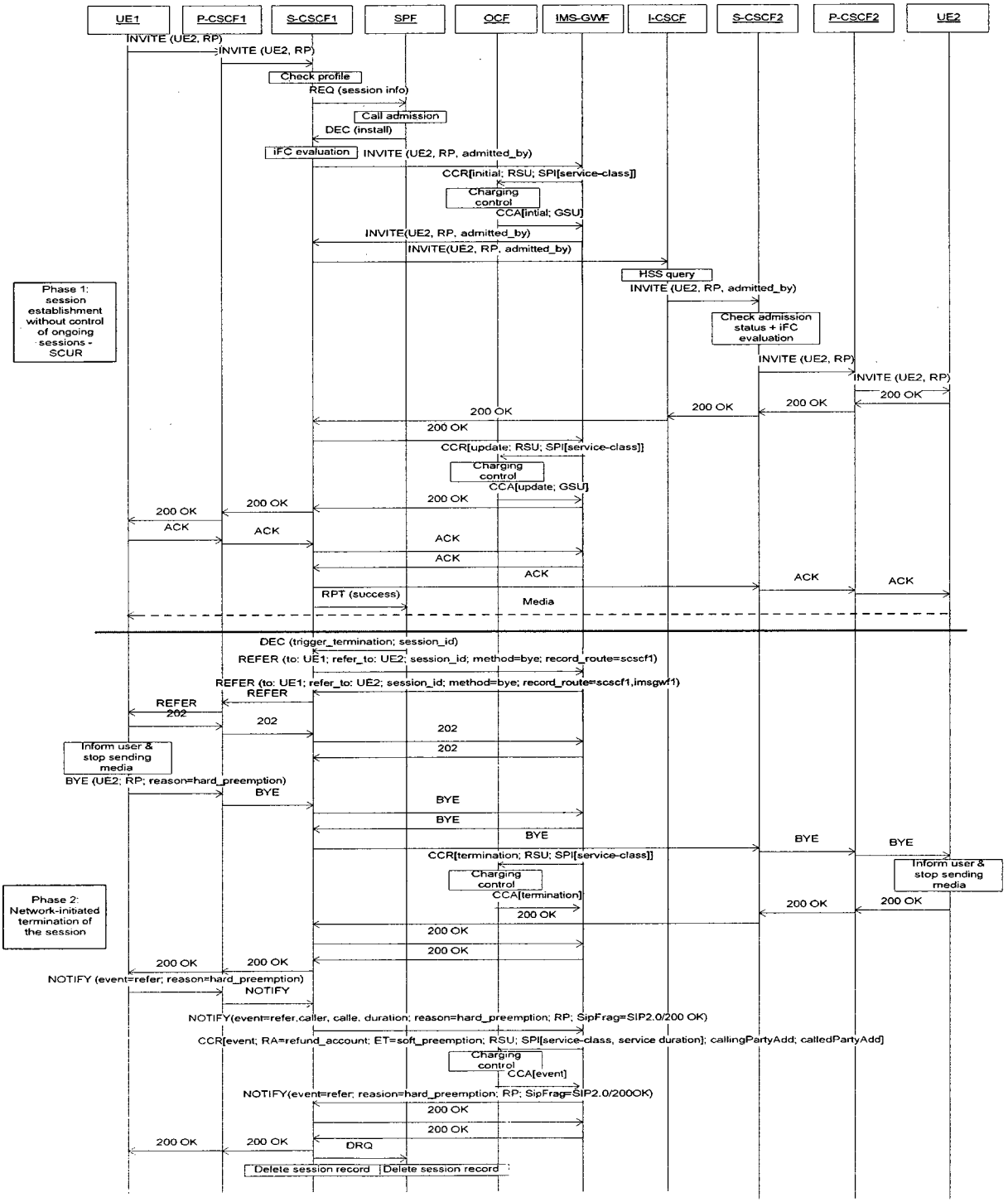


Figure 5.13: Online charging scenarios

Unlike the offline charging case, all the signaling messages received/generated by S-CSCF1 are forwarded to the IMS-GWF as means to put it on the signaling path of the initial and the referral session. For the main session, the initial INVITE message, its 200 OK acknowledgement, and the final BYE message trigger credit control interactions between the IMS-GWF and the OCF. These interactions are in the form of CCR/CCA pairs; the CCRs being of type [initial], [update], and [termination] respectively. Similarly to ACRs, CCRs also carry the service class information encapsulated in an SPI AVP. Concerning the referral session triggering the termination of the main session, we note that the S-CSCF includes in the notification message (received from UE1) additional information related to the preemption event, before forwarding the message to the IMS-GWF. For instance, the service duration, and the caller/callee identities are inserted as parameters in the event header, while a reason header indicates "hard preemption" as reason for the session termination. This information is used by the IMS-GWF to create a CCR of type [event] with the requested-action AVP set to "REFUND ACCOUNT". When this CCR is received by the OCF, it calculates the number of credits to be refunded to the user and carries a direct crediting operation on the user balance.

5.7 Conclusions

In this chapter, we have proposed a context-aware call differentiation solution as means to offer advanced QoS support in the IMS, thus enhancing its session control capabilities. This solution consisted of a novel call differentiation scheme, along with an architectural framework, two dynamic/adaptive resource management techniques, and a specialized charging model to support this scheme in the IMS. In the coming chapter, we will demonstrate how context-awareness can be used to offer enhanced emergency communications in the IMS.

Chapter 6

Use of Context-Awareness for the Provision of Enhanced Emergency Services in the IMS

In this chapter, we demonstrate how context-awareness can be integrated at the IMS control level as means to support enhanced IMS emergency communication services. The enhanced IMS emergency solution proposed addresses the limitations of existing IP-based emergency solutions, by supporting three main improvements, namely: a QoS-enhanced emergency service; a context-aware personalized emergency service; and a conferencing-enhanced emergency service.

The chapter starts by presenting some background information on emergency services. This is followed by an elaboration of the enhancements proposed to the 3GPP IMS emergency service architecture, and a presentation of our conclusions.

6.1 Introduction

Emergency services enable the public to summon help in case of emergency, and the emergency service agencies to respond quickly in order to minimize life and property losses. There are four main categories of emergency communications: *citizen to authority* (used by the public to report problems and ask for help); *authority to authority* (used by authorities for coordinating efforts during emergency/disaster relief and mitigation operations); *authority to citizen* (used by government agencies to notify the public when disasters occur); *and citizen to citizen* (used by the public to learn the state of relatives and property in case of major events) [98]. All these categories rely on synchronous (or session-based) communications, except the third category that relies on broadcasting. This chapter focuses on session-based emergency services.

Moreover, emergency calling systems rely on five main components/steps in their operation, as illustrated in figure 6.1. These steps are summarized as follows:

- **Emergency call identifier:** An easy to remember number (e.g. '911' in North America and '112' in parts of Europe) to simplify access to emergency services and enable the proper handling of emergency calls.
- **Methods for determination of caller's location and phone number:** Location information is central to the operation of emergency services, since it is used by the system to route the call to the appropriate PSAP (serving the concerned geographic area), and employed by the call taker to dispatch responders to the caller's location. Since it is frequently the case that the user is unable to provide a unique valid address, automatic location of users is the norm in emergency systems. As for the caller's phone number, it allows call centers to call the person back if they get disconnected, limit prank calls, and log calls for evidence
- **Prioritized emergency call handling mechanism:** Due to the importance of emergency calls, they need to be provided preferential treatment over regular calls (e.g. faster call setup times and higher probability of completion) and prioritized access to resources. This is especially important when there is a strong contention for scarce network resources.
- **Location-based routing to most appropriate PSAP and call establishment:** This implies the determination of the most appropriate PSAP based on the caller's location (i.e. location to PSAP address mapping) and the routing of the call to it. It should be noted that the PSAP is the only entity that can terminate the call once it is established - if the caller hangs up or gets disconnected; the PSAP operator initiates a callback.

- **Presentation of call and needed info to call taker:** This implies providing the call taker with an on screen street map that highlights the caller's position and the nearest available emergency responders.

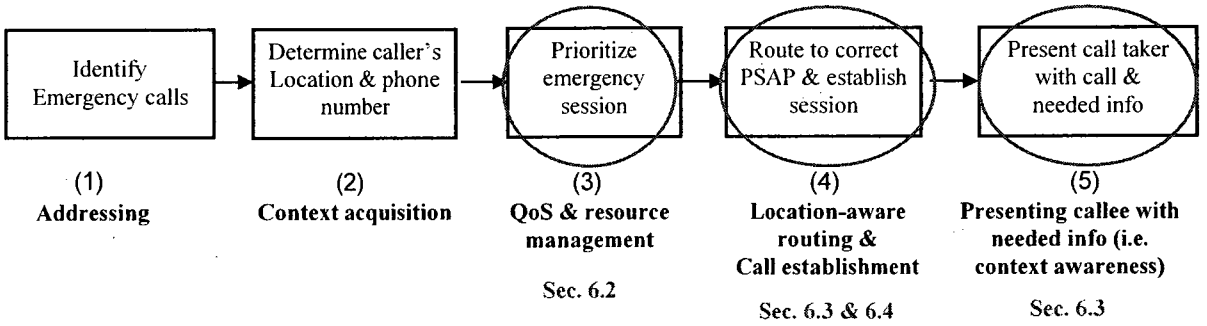


Figure 6.1: Emergency call handling steps

In the coming sub-sections, we focus on the improvement of three aspects of emergency communications that are highlighted (in red) in the figure, namely: the QoS and resource management aspect; the context-awareness and service personalization aspect; and the emergency communication models used.

6.2 Enhancing the QoS and Resource Management Aspects of the IMS Emergency Service Architecture

In this section, we enhance the QoS and resource management aspects of the IMS emergency service architecture by proposing new QoS profiles for emergency sessions (reflecting their needs in terms of QoS parameters), and the architecture and mechanisms needed for their realization. This solution is a generalization of our call differentiation solution that was presented in the previous chapter.

6.2.1 QoS Profiles for Emergency Sessions

We have previously proposed in section 5.2 a call differentiation scheme for 3G networks. This scheme enables the definition of various categories of calls, with different QoS profiles. Three QoS profiles were defined for regular calls as examples (silver, gold, and platinum). We now define new QoS profiles for emergency sessions.

Our definition of the new QoS profiles is based on the needs of emergency communication services, in terms of QoS guarantees. For the first category of emergency communications (i.e. public to authority), the CBP and the FCTP should be as low as possible to guarantee a high probability of call completion and a low probability of call interruption. A guarantee on the media type used (either audio or video) is also needed in order to avoid affecting the communication quality. Furthermore, the user perceived quality should be sustained to guarantee intelligibility. As for the session size, it is rather limited in this case (the session potentially including the personnel involved in a limited rescue operation). The second category of emergency communications (i.e. authority to authority) has similar needs, except for the session size required. In this case, the size could be large depending on the size of the mission and the involved rescue teams (e.g. national authorities, international organizations, and non profit organizations). The last category of emergency calls (i.e. citizen to citizen) can be considered as "urgent" regular calls, and therefore could be supported using the highest profile defined for regular calls (i.e. the platinum profile).

Based on this analysis, we define two possible profiles for the first two categories of emergency communications (i.e. the *emergency-public* and the *emergency-authority* profiles). These profiles are presented in table 6.1, along with the three profiles previously defined for regular calls (i.e. *silver, gold,* and *platinum*).

Factor / Class	CBP				FCTP				Multiparty session ability to grow		Media type guarantee		User-perceived media quality	
	H	M	L	Nil	H	M	L	Nil	Ltd.	Ultd.	GA	GV	Var.	Sus.
Silver	●				●				● (L1)		●		●	
Gold		●				●			● (L2)		●			●
Platinum			●				●			●	●	●		●
Emergency-public				●				●	● (L3)		●	●		●
Emergency-authority				●				●		●	●	●		●

Table 6.1: The different QoS profiles

It should be noted that the first three classes (silver, gold, and platinum) are related to the user subscription, which should indicate the HSC allowed for each user. Any class up to and including the HSC can be chosen by the user on a per call basis, and the selected class can be changed by the user during the session. The fourth class (emergency-public) is special, as it is not related to the user subscription (i.e. involves no charges). In fact, it can be used even by non-subscribed users. Furthermore, calls made using an emergency dial-string (e.g. 911) should be automatically mapped to the emergency-public class identifier. As for the fifth class (emergency-authority), it is also related to the user subscription, but the subscription for this class should be reserved for persons in NS/EP (national security/emergency preparedness) leadership positions (e.g. emergency centers coordinators, senior command levels of law enforcement, fire and public safety functions...etc). This last class could also be used for PSAP callbacks.

6.2.2 A QoS-Enhanced IMS Emergency Service Architecture

To support the defined emergency QoS profiles, we extend our IMS call differentiation architecture (originally tackling the case of regular calls) to cater to the emergency case, as shown in figure 6.2. Our previous architecture (presented in section 5.3) introduced two new functional entities (the SPF acting as resource management node and the CIB acting as context management node) and two new interfaces (the Pa interface used for the exchange of policy-based resource allocation decisions and the Pwn interface used for contextual information exchange) to the standard IMS architecture.

In order to handle emergency calls, we now introduce an additional COPS-based Pa interface between the SPF and the E-CSCF, and an additional SIMPLE-based Pwn interface between the CIB and the LRF. Furthermore, we make enhancements to the UE, the E-CSCF, and the LRF. The UE is enhanced with the ability to map public-initiated

emergency calls to the appropriate category. If the emergency session is not detected by the UE, this mapping should be performed by the P-CSCF. As for the E-CSCF, it is enhanced with the ability to communicate with the SPF for resource allocation decisions and the ability to receive triggers (concerning the re-negotiation of emergency session parameters) and take the necessary actions. The LRF is enhanced with the ability to interact with the CIB, in order to obtain more accurate location information. It should also be noted that the Gm interface and the Mw interface (existing between the P-CSCF and the E-CSCF) are both enhanced to support QoS negotiation interactions related the operation of the architecture. As for the Ml interface, it is enhanced to support the exchange of refined location information (e.g. ID of a room in a building) between the LRF and the E-CSCF.

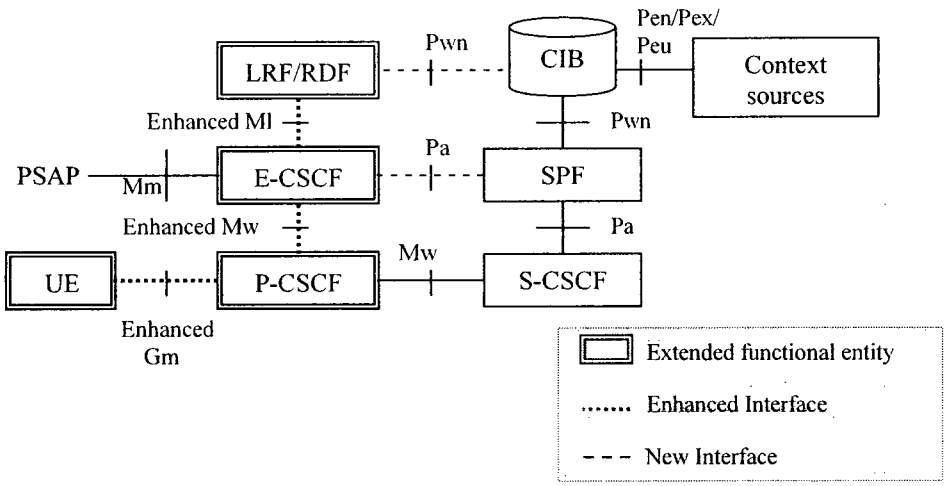


Figure 6.2: The QoS-enhanced IMS emergency service architecture

Figure 6.3 illustrates the architecture's general operation, in which call differentiation is achieved as follows: For public initiated emergency calls (e.g. 911 calls), when the user initiates the emergency call, the UE detects this, maps the call to the emergency-public service class, and forwards the session initiation request (including the service class) to the P-CSCF. The P-CSCF sends the request to the E-CSCF, which communicates with the SPF in order to allocate resources to the call. If resources are available, the SPF renders a

positive decision and the call is established normally. If no resources are available, the SPF triggers one or several S-CSCFs to downgrade and/or preempt one or more ongoing (regular) calls, in order to free resources for the emergency call, which is admitted afterwards. After session establishment, the SPF triggers the E-CSCF to re-negotiate the session parameters in order to sustain its user perceived quality. Furthermore, a limit is imposed on the session size in case of lack of resources. Meanwhile, the LRF may consider the CIB as location server and interact with it to obtain more accurate location information, or location information obtained from alternate sources (e.g. wireless sensor networks).

For mission critical calls (i.e. calls initiated by authorities), emergency callbacks (initiated by PSAP operators), and regular calls, a similar procedure is followed, except that the call goes through a S-CSCF that communicates with the SPF for resource allocation decisions. In these cases, the call category is explicitly chosen by the user when the session is established. Furthermore, in the case of regular calls, depending on the call category and its CBP, the call may be rejected if there are not enough resources and no additional resources can be freed. This is not the case for the two categories of emergency calls, which have a CBP of zero.

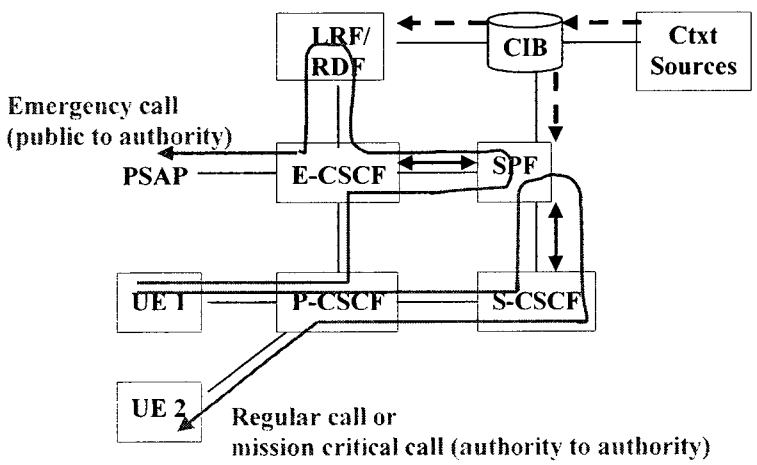


Figure 6.3: Illustration of the QoS-enhanced IMS emergency service architecture 's operation

As shown in the figure, whether the call is a regular call or an emergency call initiated by a regular user; a mission critical user; or a PSAP operator, it receives the appropriate treatment due to the interaction of E-CSCFs and S-CSCFs with the SPF, which dynamically allocates resources to sessions, based on their QoS profiles.

In terms of resource management strategy, the two resource management mechanisms presented in section 5.4 (i.e. call admission control and media parameter control) are used to enable preferential treatment at the beginning and during sessions. The media parameter control mechanism previously presented in reused without modifications, while the call admission control mechanism is adjusted to take into consideration the two newly defined emergency classes as follows: In light to regular loading conditions, upper limits are imposed on regular call classes (i.e. silver, gold, and platinum) so that their overloads do not affect emergency classes. As for the two emergency classes, no thresholds are imposed. In fact, if these classes exceed their engineered loads, they may use any additional capacity available. However, in high loading conditions or crisis situations, a more aggressive approach (i.e preemption) is used to adapt to this situation by transferring resources between classes (e.g. from regular classes to emergency ones) when needed. Figure 6.4 illustrates the modified call admission strategy.

Figure 6.4: The call admission control strategy taking into consideration emergency calls

As for billing/charging, the third class (i.e. the emergency-public class) involves no charges since it represents a non-subscription service, while the fourth class (i.e. the emergency-

authority class) is assigned a specific charging rate like the first three classes. To apply this rate, information about the call's *service class* must be conveyed to the charging system, as described in sections 5.5.1 and 5.5.3. Moreover, since the two emergency classes do not involve any forms of preemption (i.e. FCTP equal zero and no drop from video to audio), the allocation of compensation credits to users is not needed in these cases.

6.2.3 Emergency Sessions Management Scenarios

There are several emergency sessions' management scenarios related to our architecture, including: Successful emergency session initiation without control of ongoing sessions; Successful emergency session initiation after downgrade/termination of an ongoing two-party or multi-party session; in addition to exception scenarios (e.g. unrecognized/missing priority value and violation of user profile). In this section, we present one of these scenarios as example.

Figure 6.5 illustrates the case where an ongoing regular session (a session between UE2 and UE3) must be terminated in order to free resources for an emergency session (initiated by UE1) to be established. In this scenario, we assume that UE1 has already registered with the IMS and that the destination PSAP is IP-enabled. The scenario begins when the user operating UE1 attempts to establish an emergency call. UE1 then sends a SIP INVITE message with a resource-priority header set according to the session category (Q735.0 in this case) to the P-CSCF assigned to the user (i.e. P-CSCF1). P-CSCF1 ignores the resource-priority header and forwards the request to an E-CSCF within the same network. The E-CSCF sends the SPF a call admission request using a COPS REQ message (including the session info). In this case, the SPF determines that an ongoing session must be terminated in order to free resources, and sends a "trigger termination" decision to S-CSCF1, in order to modify the decision made about the (previously admitted) session

between UE2 and UE3. After receiving this decision, S-CSCF1 sends a SIP REFER message instructing UE2 to contact UE3 in order to terminate the session established between them. UE2 then sends a SIP BYE message to UE3, containing a reason header that indicates "hard preemption" as reason for the termination. After the session is terminated, UE2 sends a notification to its S-CSCF, using a SIP NOTIFY message. This S-CSCF informs the SPF (using a COPS RPT message), which sends an "install" decision (using a COPS DEC message) to the E-CSCF to authorize the admission of the emergency call. The E-CSCF then carries the rest of the emergency session establishment procedure normally, and sends a COPS RPT message indicating that it has complied with the SPF decision.

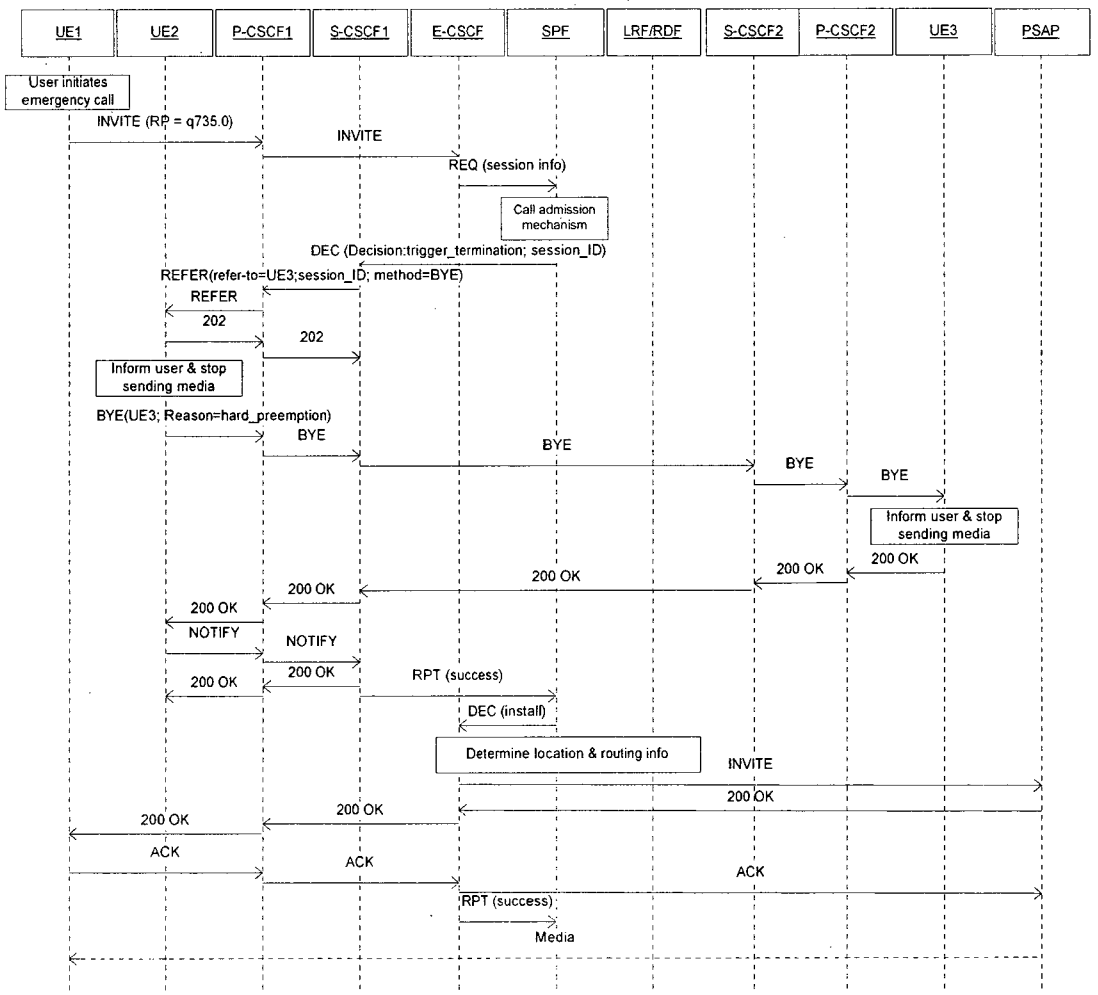


Figure 6.5: Successful emergency session establishment, after the termination of an ongoing session

6.3 Offering Personalized, Context-Aware Emergency Services in the IMS

Emergency services are considered as context-aware services, since they rely on contextual information in their operation. However, the range of contextual information used today to support emergency operations is limited to the caller's location and phone number. Exploiting a richer set of contextual information could lead to enhanced, personalized, emergency services and more efficient emergency operations. In this section, we propose an extension of the 3GPP IMS emergency service architecture as means to provide personalized, context-aware emergency services to 3G users. We start by describing a motivating scenario, before presenting the architecture proposed and elaborating the personalized emergency service operation in the IMS.

6.3.1 Motivating Scenario

In this section, we describe an enhanced emergency service scenario to illustrate the benefits of using a rich set of contextual information in support to emergency operations. Figure 6.6 depicts this scenario.

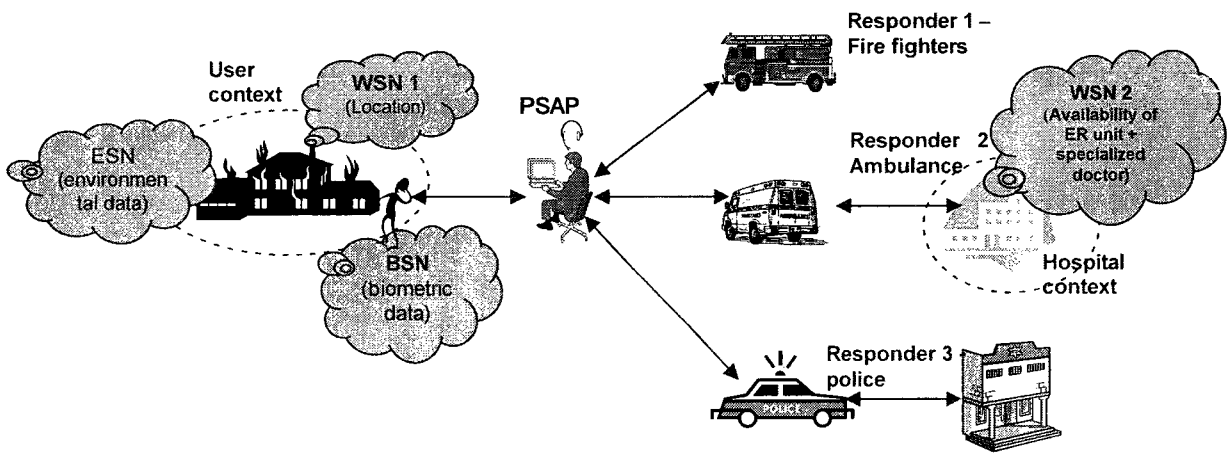


Figure 6.6: Illustration of enhanced emergency service scenario

The scenario describes a fire incident occurring at night, in a place with a defective fire detector. The victim (an elderly person, mainly speaking Spanish) is trapped in a burning

apartment building, and calls 911 using her 3G terminal. WSNs are used to detect and convey the user's contextual information (i.e. spatial, physiological, and environmental data) to the network.

Upon receipt of the 911 call establishment request, the user's contextual information is queried by the system, and four pieces of information are used to determine the most appropriate PSAP to which the call should be routed, namely: the *user's location (address and room ID)*; the *service requested* (e.g. fire fighters); the *terminal's media capabilities* (e.g. audio and video); and the *user's language preferences*. The user's location is used to determine the PSAP servicing the concerned geographic area and to dispatch responders to the exact room where the user is located (for improved response time). It should be noted that this location information (i.e. the ID of a room in a building) is a refinement of the location information used today in mobile emergency services (mainly consisting of a cell ID determined using triangulation techniques). The information about the service requested may be used to route the call to an emergency-specific PSAP (if this is the case according to the jurisdiction), while the information about the media capabilities could help directing the call to a PSAP handling a certain media type (e.g video or text). As for the language preferences, they are used to direct the call to a PSAP call taker speaking the user's preferred language (Spanish in this case).

After the call establishment, the PSAP call taker assesses the situation and dispatches fire fighters, the police, and an ambulance to the person's location. The fire fighters and the ambulance personnel are added to the call (conferencing) by the PSAP call taker, to give further instructions to the user. Furthermore, environmental data (e.g. temperature, humidity. noise level) is conveyed to the fire fighters to enable a better assessment of the

situation and the tracking of the fire progress. Similarly, the user's vital signs are monitored and communicated to the ambulance. While waiting for the arrival of the responders, information about the user's surrounding people/devices is used by the PSAP call taker to guide the user towards the nearest first aid kit/oxygen mask/fire extinguisher in the building, or even a person with medical training that could provide temporary assistance. Upon the responders' arrival, the conference is terminated by the PSAP call taker, and the nearest hospital with an available ER unit and a specialized doctor is queried by the ambulance personnel, who establishes a call with the hospital and transfers to it the patient's medical information (e.g. heart rate, temperature, respiratory rate, and blood pressure).

6.3.2 A Context-Aware IMS Emergency Service Architecture

Offering context-aware emergency services entails two types of issues, namely: the management of the contextual information (captured by WSNs) by the system, and the integration of this information in emergency service operations. Figure 6.7 depicts the proposed architecture tackling those issues.

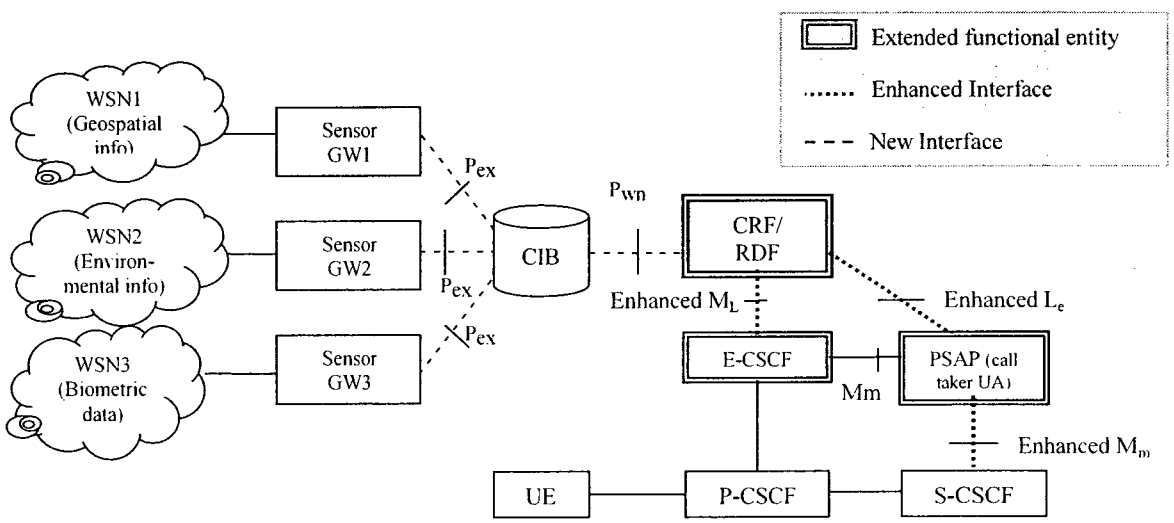


Figure 6.7: The Context-aware IMS emergency service architecture

This architecture extends the 3GPP IMS emergency service architecture with context management entities, namely: sensor gateways (GWs) acting as interworking units between WSNs and the 3G core network; and a CIB responsible of the management/dissemination of the contextual information provided by sensor GWs to interested parties. These entities were already introduced as part of the context acquisition/management solution proposed in chapter 4.

In addition to the newly introduced entities, some enhancements are made to existing entities, namely: the LRF, the E-CSCF, and the PSAP. The LRF (initially responsible of retrieving the location of the UE initiating the emergency call) is evolved into a Context Retrieval Function (CRF) allowing the retrieval of different pieces of contextual information related to the user, from the CIB. As for the E-CSCF and the PSAP, they are enhanced with the ability to request different pieces of information from the evolved LRF, via the enhanced M_L and L_e interfaces. The M_m interface existing between the PSAP and the S-CSCF is also enhanced to convey contextual information updates sent by the PSAP call taker to responders involved in the rescue operation (e.g. fire fighters and ambulance personnel). In addition to the enhanced interfaces, two new SIMPLE-based interfaces (the Pex and the Pwn interfaces) are introduced as inbound and outbound interfaces related to contextual information exchange within the network. In the coming sub-section, we detail a session management scenario illustrating the system's operation.

6.3.3 Personalized Emergency Service Operation

Figure 6.8 details the enhanced emergency service scenario presented in section 6.3.1. In this case, we assume that all clients (i.e. the UE, the PSAP, the fire fighters, the ambulance personnel, and the hospital) have already registered to the IMS, and that the PSAP is IP-enabled.

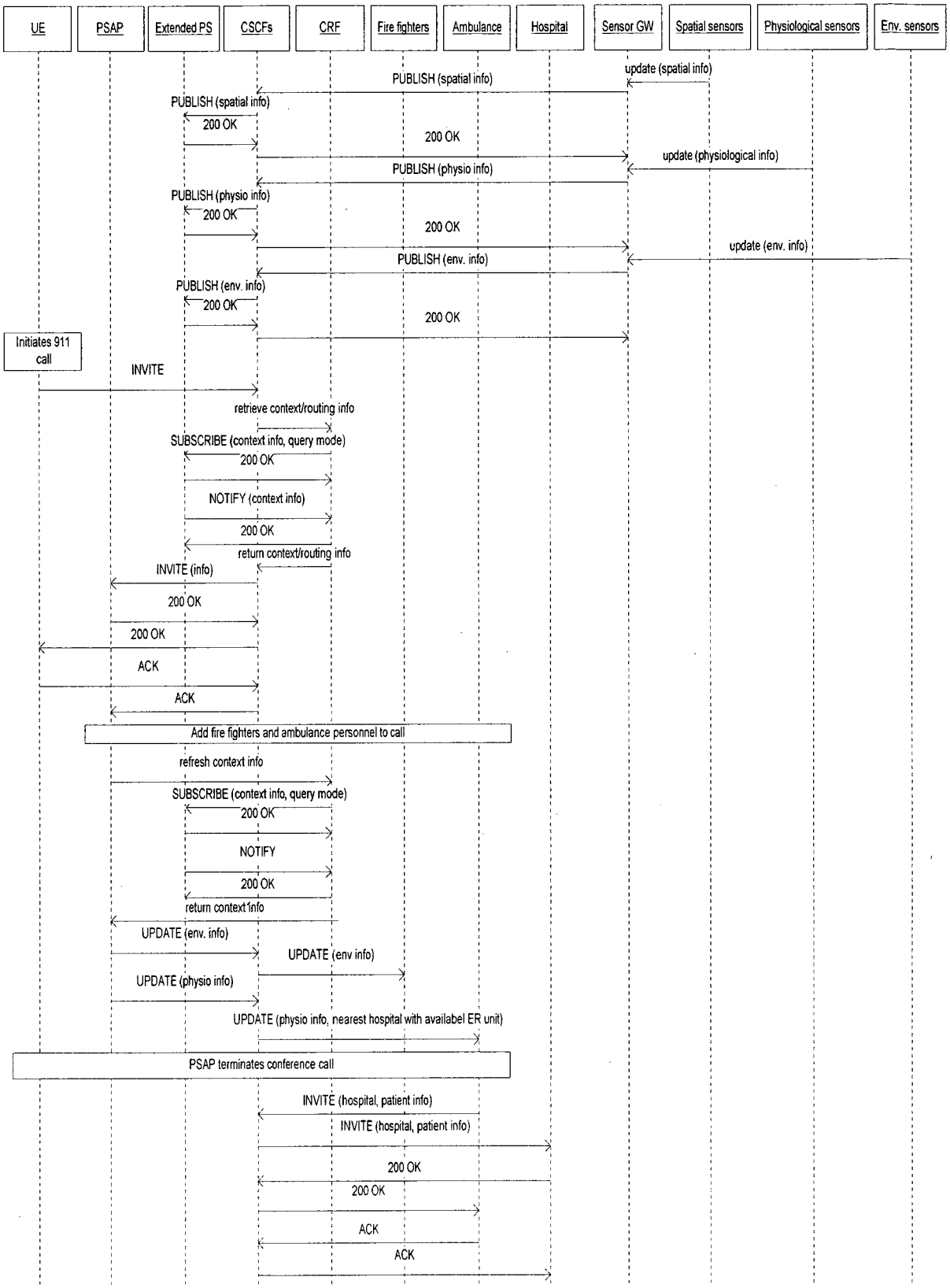


Figure 6.8: Personalized emergency service scenario

Before the initiation of the emergency call, WSNs collect contextual information (i.e. spatial, physiological, and environmental info) about the user, and convey it to the sensor gateway, which publishes it to the extended PS, going through a set of CSCFs (acting as pesentity presence proxy). When the user initiates the emergency call, his/her UE sends a SIP INVITE message to the P-CSCF assigned to the user. The P-CSCF forwards the request to the E-CSCF which interacts with the CRF to retrieve the user context information. The CRF queries the PS (using a SIP SUBSCRIBE message) concerning the needed information that is conveyed afterwards to the E-CSCF. Based on this information, the E-CSCF routes the call to the appropriate PSAP.

After the session establishment, the PSAP call taker dispatches responders to the user location, and adds the fire fighters and the ambulance personnel to the call. Furthermore, the PSAP call taker uses his/her terminal to send an information refresh request to the CRF, which returns the fresh information after interacting with the PS. Using SIP UPDATE messages, the PSAP UA sends fresh environmental information (related to the user) to the fire fighters, as well as fresh physiological information and the location of the nearest hospital with an available ER unit, to the ambulance personnel. Furthermore, the PSAP call taker uses the refreshed information to guide the user towards a nearby first aid kit.

Upon the arrival of the responders, the PSAP call taker terminates the conference. Afterwards, the ambulance personnel establish a call with the hospital, via a SIP INVITE message encapsulating the patient's medical information.

6.4 Offering Multi-Party Emergency Services in the IMS

Despite the critical nature of emergency calls, the means/forms of emergency communications supported in current solutions are rather limited (mainly to two-party

voice calls). Richer and more sophisticated forms of communication, such as multimedia multi-party communications, could be supported to help in situation assessment and better coordination of rescue efforts. In this section, we enhance the IMS emergency service architecture with multi-party session support capabilities, as means to offer richer forms of emergency communications (e.g. conferencing, sub-conferencing, and automatic switching between sub-conferences) to 3G users. We start by some background information on multimedia conferencing, before presenting the architecture proposed and elaborating the different multi-party emergency sessions' scenarios.

6.4.1 Background on Multimedia Conferencing

Multimedia conferencing (also known as multimedia multi-party sessions) can be defined as the conversational exchange of multimedia content (i.e. audio, video, and text) between multiple parties. Conferences can be classified using several schemes. An example is floor control, which allows users of networked multimedia applications to utilize and share resources (e.g., media channels) without access conflict [99]. Another classification scheme is how the conference starts. *Pre-arranged* conferences start at a pre-determined time and have a known conference identifier, while *ad hoc* conferences start spontaneously when the first two users start communicating. Yet another scheme is whether participants can join without invitation - a conference being *closed* (or private) if it is not possible to join without being invited by one of the participants, while an *open* (or public) conference allows participants to join when they wish. A fourth scheme is whether the conference has *sub-conferencing capabilities* – a sub-conference representing a kind of private room within the main conference, in which participants can hear/see each other without being heard/seen by the other participants. A last scheme is the *topology used for signaling and*

media handling, the four main existing topologies being: end-system mixing; full mesh; multicast; and centralized [100]. Among these topologies, the centralized model is the one used in 3G networks. In this model, a conferencing server does the mixing for all the end-systems, and there is signaling link between every end-system and this server.

6.4.2 Multi-Party Emergency Session Support in the IMS: A Case Study

In this section, we present a case study on the support of multi-party emergency sessions in the IMS. We start by discussing the conferencing models we propose for different types of emergency sessions, then present the architecture enabling the realization of these models and detail some session management scenarios.

6.4.2.1 Conferencing Models for Emergency Sessions

Several conferencing models can be devised to support multi-party emergency sessions based on the combination of the different conferencing criteria (i.e. how the conference starts/ends, the resources handling options, the rules that should be respected by the users and the sub-conferencing capabilities). In this work, we propose two potential conferencing models suiting the needs of public initiated emergency calls (i.e. public to authority communications) and mission critical calls (i.e. authority to authority communications), and discuss some of the context-awareness features that can be used as part of their operation.

For both types of emergency sessions, we chose centralized, ad-hoc conferencing models with basic floor control capabilities. The centralized topology was chosen since it is the only topology that is supported in 3G networks. As for the ad-hoc nature of the models (i.e. sessions starting spontaneously without a pre-defined time), it is needed since emergencies represent unforeseen events that do not have a pre-determined time. A basic form of floor

control, in which a chair conducts the conference and orchestrates access to the shared resources, is used in both models.

For public-initiated multiparty sessions, each session could be conducted by the PSAP call taker, with the possibility for one of the responders (e.g. fire fighters or paramedics) to act as successor if the call taker leaves the call. In this case, the chair starts and ends the conference when he /she wishes, and only a chair can invite other users to the conference. This implies that such conferences are closed – i.e. users can only join them when invited by the chair. Finally, since the sizes of such conferences are normally limited to a small number of responders involved in a small rescue operation, only simple sub-conferencing capabilities (such as the creation/termination of sub-groups) may be needed. Such capabilities could be supported in a context-sensitive manner to enable more efficient and personalized rescue operation. For instance, the choice of the communication partner involved in a sub-session with one of the responders could be based on the caller's location and his/her health condition (e.g. initiation of a sub-conference between a paramedic and the nearest hospital with an available ER unit and specialized doctor).

Similarly, multi-party sessions related to disaster relief operations should be conducted by a chair, such as a chief of operations responsible of the commandment of troupes and the control of the mission - with the possibility of passing the control of the sessions between chiefs when needed. However, since such sessions may involve several rescue teams (e.g. national authorities, international organizations, and non profit organizations), sophisticated sub-conferencing capabilities (such as the creation of sub-groups, the moving of users between sub-groups, the splitting/merging of sub-groups, and the automatic switching between sub-groups based on the user location) may be needed to help the coordination of

rescue efforts. In this case, conferences are initiated by the chair and are automatically terminated when the last two participants leave a certain conference. Furthermore, new participants are only allowed to join when invited by other users that are already participating in the session. An interesting context-awareness feature that could be supported in this case is the possibility to track the chief location and automatically move him/her between relevant sub-conferences to be able to supervise different groups working on different rescue sites. Table 6.2 summarizes the features proposed for public-initiated and mission critical multiparty sessions.

Conferencing model for public initiated emergency communications	Conferencing model for mission critical communications
▪ Ad hoc ▪ Conducted by chair (call taker as main chair & possibility for responders to act as secondary chairs) ▪ Simple sub-conferencing capabilities (creation/termination of sub-conferences) ▪ Closed ▪ Join allowed ▪ Only a chair can invite users. Users can't join. ▪ Chair starts conference and ends it as he/she wishes. ▪ Context-awareness feature: Automatic selection of sub-conferences communication partners based on caller situation	▪ Ad hoc ▪ Conducted by chair (chief of operations & possibility to pass control to other chiefs) ▪ Sophisticated sub-conferencing capabilities (creation/termination/splitting/ merging of sub-conferences, moving users between sub-conferences) ▪ Closed ▪ Join allowed ▪ Participants invite other users ▪ Chair starts conference. Conference ends when the last two users leave. ▪ Context-awareness feature: Automatic moving of chief between sub-conferences based on his/her location

Table 6.2: Conferencing models for emergency communications

6.4.2.2 An Enhanced IMS Emergency Service Architecture for Multi-Party Session Support

Figure 6.9 depicts the architecture proposed to realize the conferencing models presented in the previous section. This architecture brings enhancements to some of the existing IMS emergency service architecture's functional entities, namely: the conference focus; the PSAP caller taker's UE; the rescuer UE; and the mission critical caller's UE.

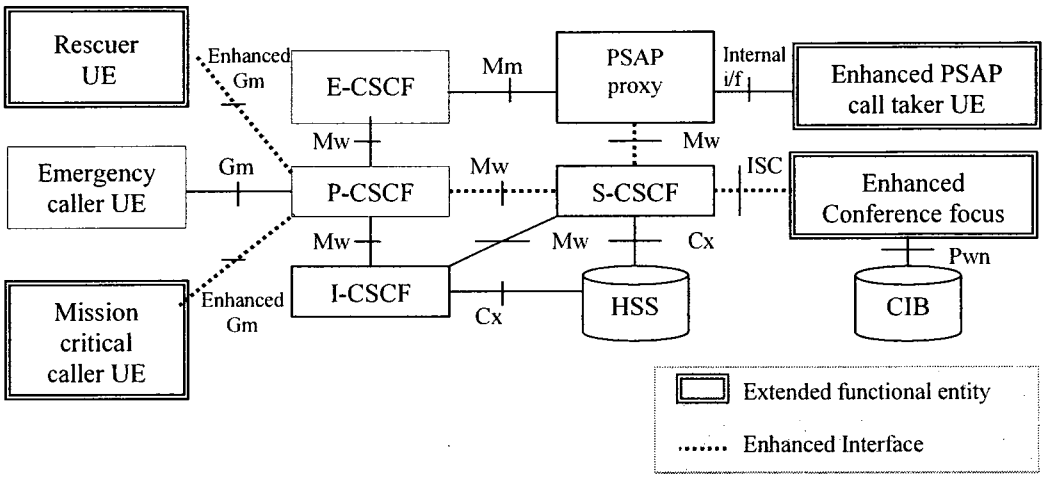


Figure 6.9: The conferencing-enhanced IMS emergency service architecture

The conference focus is enhanced with sub-conferencing capabilities, such as the ability to carry operations related to the creation/termination of sub-conferences, the moving of users between sub-conferences (based on instructions from authorized parties), and the automatic switching of users between sub-conferences based on their location. The PSAP call taker's UE (initially handling two-party emergency calls only) is enhanced with basic conferencing and sub-conferencing capabilities such as the ability to create a conference for each emergency call (by combining existing active sessions), the ability to add/remove users to/from existing conferences, the ability to terminate conferences, and the ability to create/terminate sub-conferences. As for the mission critical caller acting as chief of operations, his/her UE is enhanced with conferencing and sub-conferencing capabilities in order to request the creation of conferences between different rescue team members, as well as the creation of sub-conferences (or private rooms) within the main conference and move users between sub-conferences or the main conference and sub-conferences (and vice versa), in addition to the termination of conferences/sub-conferences. Similarly, the rescuer's UE acting as secondary chair is enhanced with basic conferencing and sub-conferencing capabilities. In terms of interfaces, the Gm interface residing between UEs

and their P-CSCFs is enhanced to support sub-conferencing related interactions. As for the ISC interface (residing between the conference focus and the S-CSCF) and the Mw interfaces (residing between CSCFs), they are also enhanced to support sub-conferencing related interactions in addition to interactions related to the passage of control of sessions between session chairs - both types of interactions being performed using regular SIP operations as shown in the coming sub-section. It should be noted that since floor control is enforced by chairs who are controlling manually the access to session resources (i.e. who is added/removed from session, who is moved to a sub-session), no additional floor control protocol is needed on these interfaces.

Figures 6.10 and 6.11 illustrate the architecture's operation. For the case of public-initiated emergency session shown in figure 6.10a, the session is first established normally between the caller and the PSAP call taker, going through the P/E-CSCFs and the PSAP proxy. After the call establishment, the call taker assesses the situation and dispatches the needed responders (e.g. fire fighters, the police, paramedics) to the caller's location. During the call, the call taker may decide to add some of the dispatched responders to the call (thus creating a conference), to give further assistance to the user. This can be achieved by joining some of the ongoing two-party sessions in a three-way session, as follows: first, the call taker interacts with the 3G conference focus to create a conference. Then, it sends a referral message to each of the parties to be included in the conference, instructing them to join the conference. Each party must then contact the focus to join the conference and send a notification to the call taker about the join operation result. The call taker can then terminate the original two-party sessions it has with these parties, and treat the three-way session as a regular conference. Figure 6.10b shows the relationships existing between the

call taker and the different parties, at the initial/final stages of the conference creation.

It should be noted that instead of relying on the conference focus residing in the 3G network, the call taker could have contacted a local conferencing server hosted by the PSAP to achieve such conferencing scenario. However, this would necessitate some changes to the PSAP architecture. Furthermore, the design could be modified so that a conference is created from the beginning of each call, by relying on a PSAP call controller that is responsible of the creation of a conference for each incoming call and the addition of the caller and the appropriate call taker to this conference. This solution would necessitate important changes to existing PSAPs and could have some impact on emergency sessions establishment time, since first the conference needs to be created and the caller added to it, then the call taker is added. This is in contrast with our proposed three-way session solution that establishes the two-party call normally at the beginning (to achieve a short call setup time), then performs additional interactions during the session (when needed) to join different active sessions into a conference.

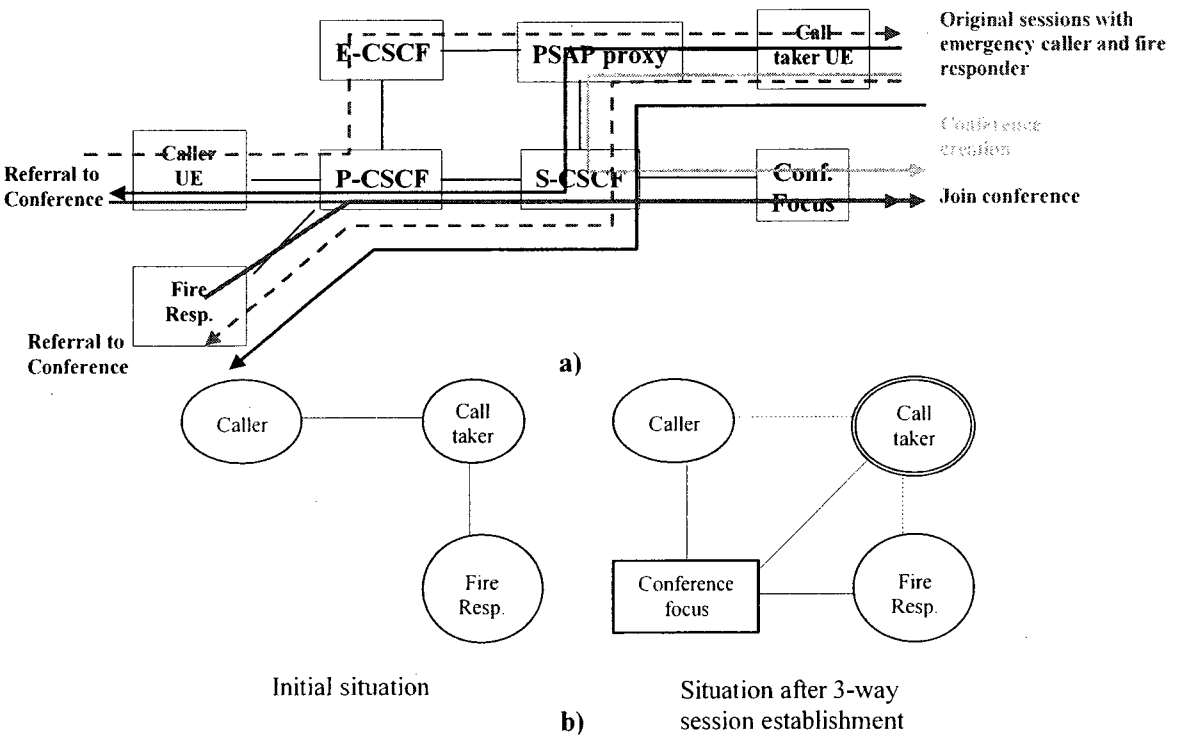


Figure 6.10: Conferencing for public-initiated emergency calls: a) Conference establishment related interactions; b) Relation between call taker and other entities

For the case of mission critical calls shown in figure 6.11, the conference starts when the chief of operations triggers the 3G conference focus to create a conference and add some participants to it. The focus will then send invitations to the specified parties and the conference is started. During the session, the chief may decide to add/remove some users from the main conference, by sending triggers to the focus that will take the necessary actions. Furthermore, the chief may create sub-conferences (i.e. private rooms) within the main conference and move users to/between them to enable a better coordination of rescue efforts and the sharing of information with some specific participants (e.g. giving certain order to a specific group of rescuers). Meanwhile, the focus can keep track of the location of the chief (by interacting with the CIB) and move him/her between relevant sub-conferences based on his location, to be able to command different groups working in different sites.

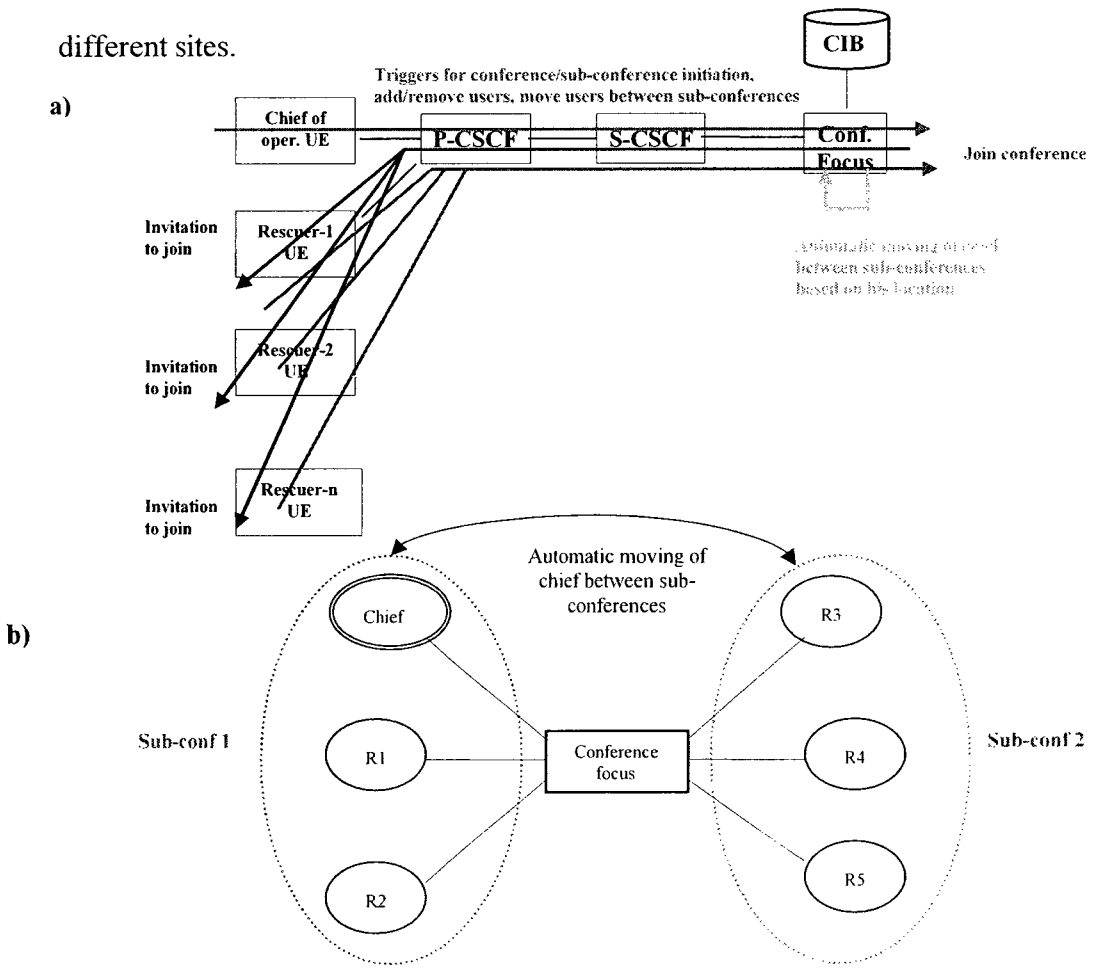


Figure 6.11: Conferencing for mission critical calls: a) Conference related interactions; b) Relation between chief of operations and other entities

6.4.3 Multi-Party Emergency Sessions Scenarios

There are several multi-party emergency sessions scenarios related to our architecture, including: the creation of a three-way session in the case of public-initiated emergency calls; the creation of a conference by the chief of operations for mission critical communications; the creation of sub-conferences within a main conference; and the automatic moving of the chief of operations between sub-conferences. In this section, we present some of these scenarios as examples.

6.4.3.1 Multi-Party Public Initiated Emergency Session

Figure 6.12 shows the different interactions related to the establishment of a multi-party public initiated emergency session. In this scenario, we assume that the call taker has already established (separate) two-party sessions with the caller and a fire fighter responder, and wants to join these sessions in a conference. To achieve this operation, the call taker sends a SIP INVITE message to the conference focus to create a conference. Upon the receipt of this message, the conference focus creates a unique URI for the conference and allocates to it the needed media resources. Afterwards, the call taker instructs the caller and the fire responder to join the conference, by sending to each of them a SIP REFER message with a 'refer-to' header set to the confURI and a 'method' attribute set to invite. Upon the acceptance of this referral (using SIP 202 messages), each of the caller and the fire responder send a SIP INVITE message to the conference focus to join the conference, then notify the call taker about the results of the referral. At this point, all three parties are part of the conference and can exchange media via the conference server. Afterwards, the call taker terminates the original two-party sessions it has with the caller/fire responder by sending a SIP BYE message to each of them.

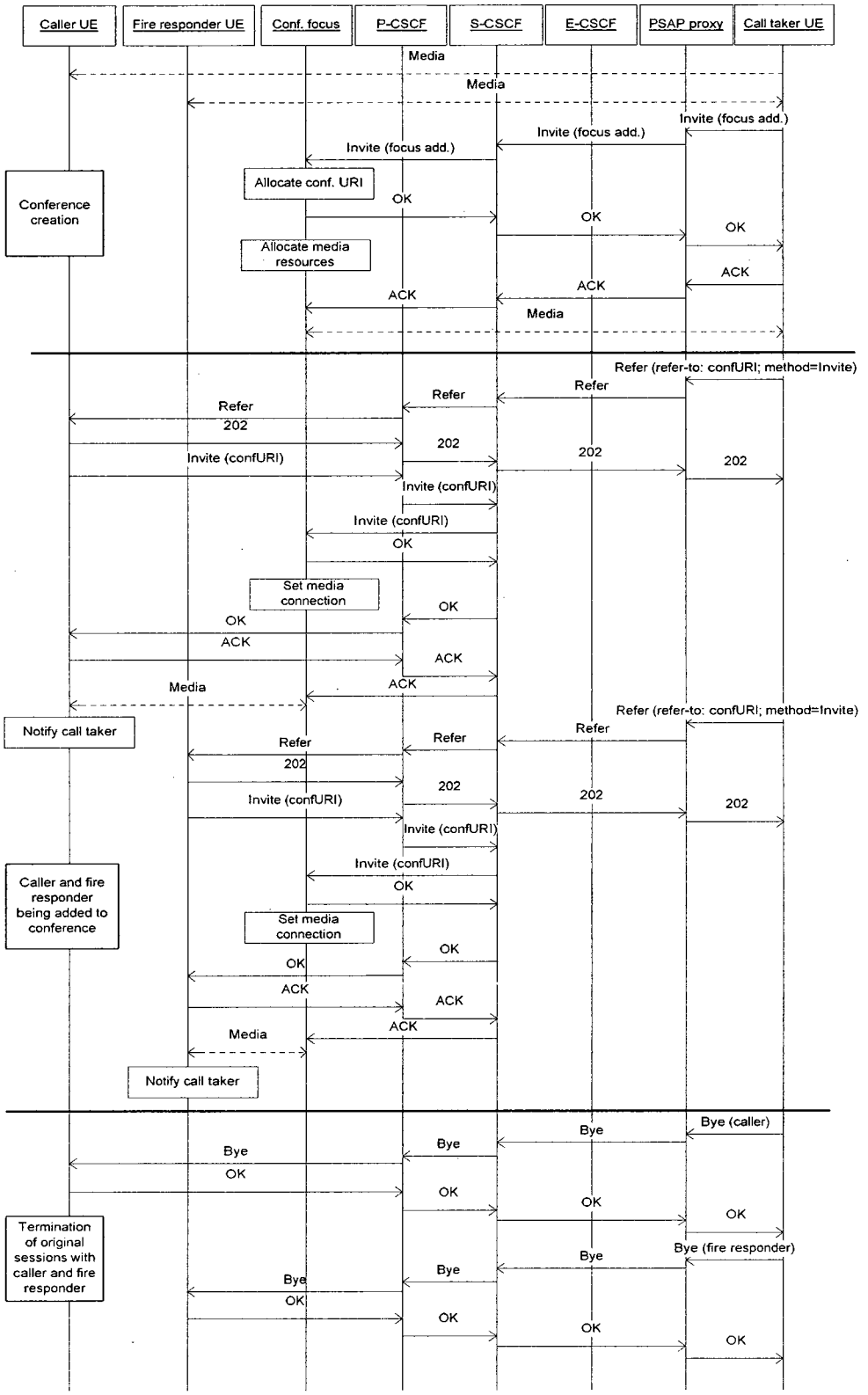


Figure 6.12: Multi-party public initiated emergency session related interactions

6.4.3.2 Multi-Party Session for Disaster Relief Operation

Figure 6.13 shows the different conferencing interactions related to the establishment of a mission critical multi-party session.

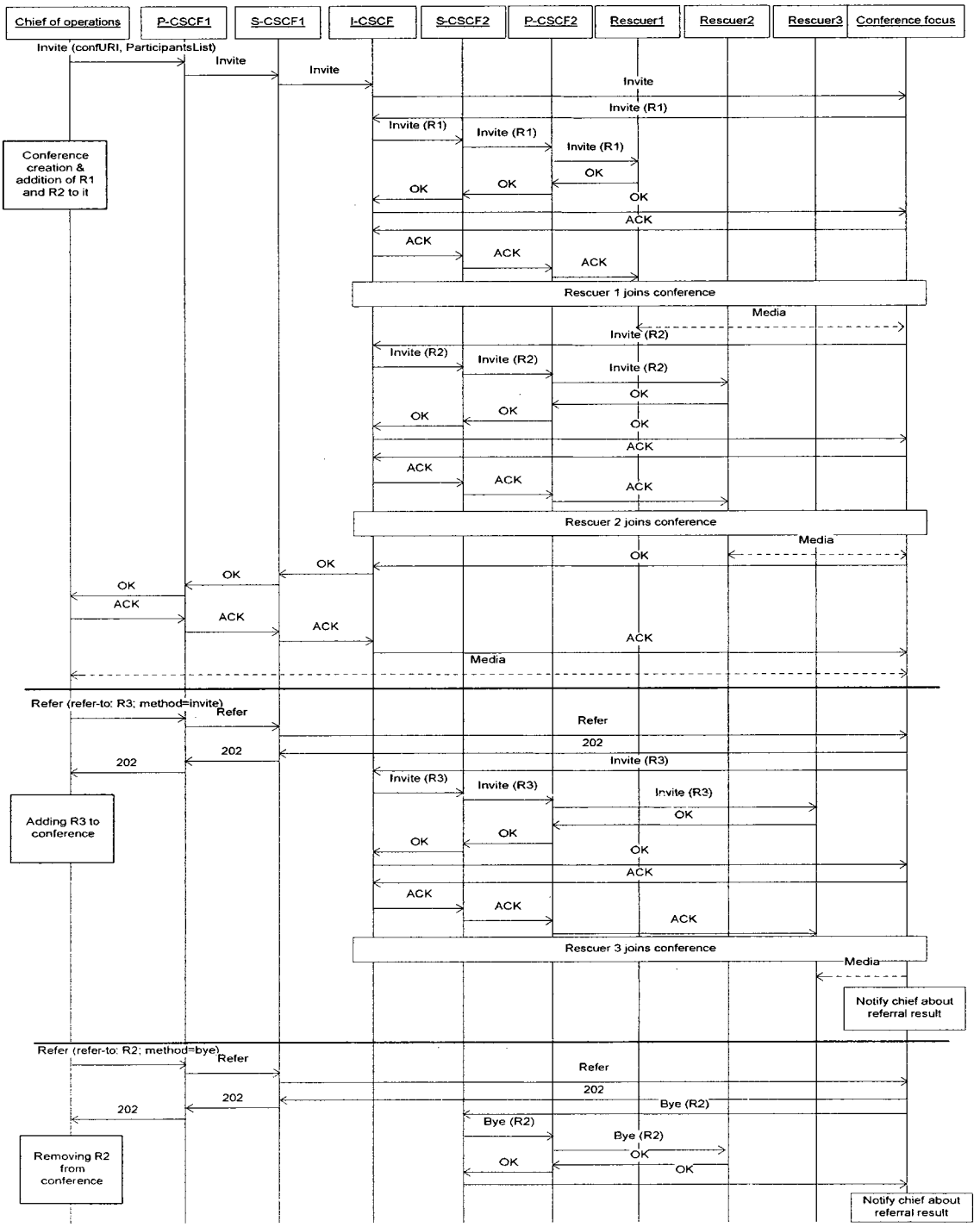


Figure 6.13: Mission critical multi-party session – conferencing interactions

In the first phase, the chief of operations sends a SIP INVITE message, carrying in its body the list of participants, to the conference focus in order to create the conference with the specified users. Upon the receipt of this message, the focus sends a separate SIP INVITE message to each participant, which joins the conference. After the successful joining of all the specified users, the focus responds with a SIP OK message that is relayed to the chief of operations. At this point, the chief and all the specified users are part of the conference and can communicate via the conference focus. As a second step, the chief instructs the focus to add a new participant to the conference, by sending to it a SIP REFER message. This last accepts the referral, sends a SIP INVITE message to the new participant that subsequently joins the conference, and notifies the chief about the referral result. The third stage that implies the removal of a participant from the conference is similar, except that the REFER message carries the method 'bye', and that the focus sends a BYE message to the participant to be removed.

Figure 6.14 details some sub-conferencing related interactions. In the first stage, the chief sends a SIP INFO message, carrying the needed info (i.e. action, users' addresses, and sub conference ID) in its message body, to the focus in order to create a sub-conference a move some of the participants to it. Upon the receipt of this message, the focus moves the users' streams from the main conference stream pool to the sub conference stream pool, and sends a notification about the sub-conference initiation result to the chief. We note that no signaling action is needed here, since signaling relations already exists with all the users (due to the main conference initiation). The second stage that involves the moving of a user between two sub-conferences is similar, except that the INFO message carries as action 'move user' in addition to the user address, the main conference ID, and the origin/

destination sub conferences IDs. In this case, the focus also moves the user stream to the appropriate stream pool. Finally, the third stage shows the automatic switching of the chief between different sub-conferences based on his/her location. In this case, no triggers are received by the focus, which automatically tracks the chief's location (by interacting with the CIB) and moves his/her stream to the appropriate pool based on this information.

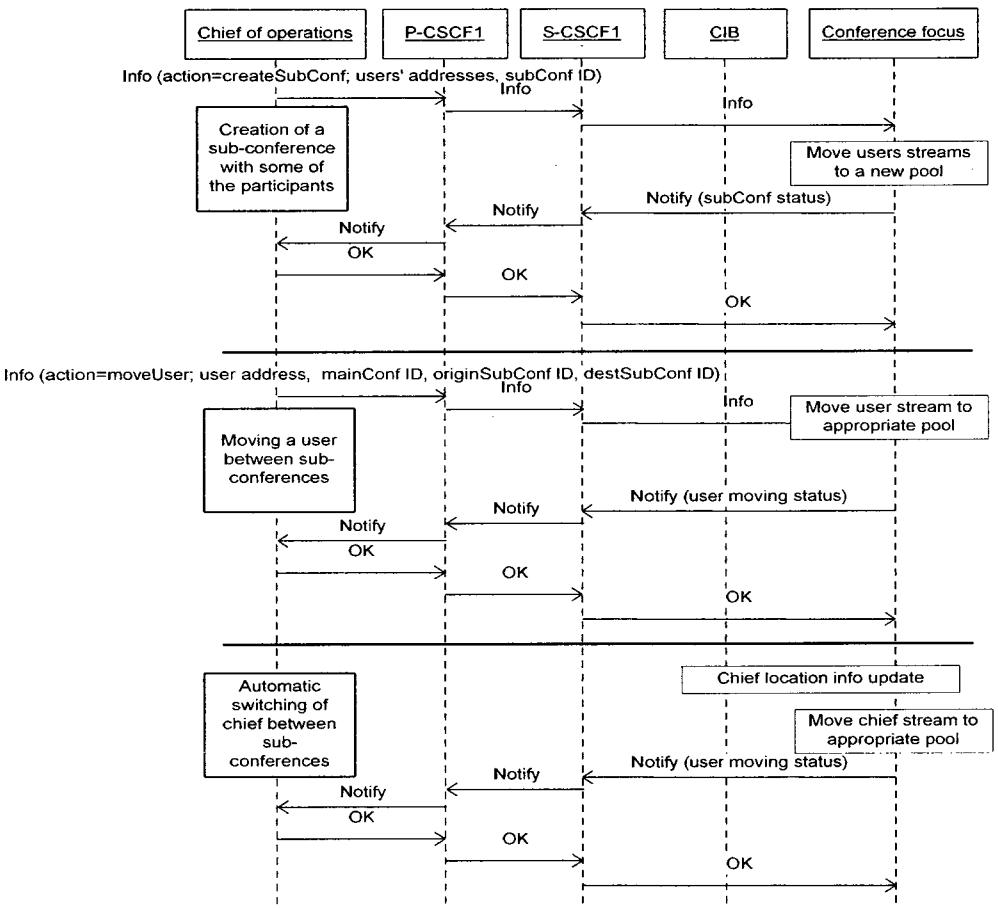


Figure 6.14: Mission critical multi-party session – sub conferencing interactions

6.5 Conclusions

In this chapter, we demonstrated the use of context awareness at the IMS control level as means to support enhanced emergency communications. The solution that we proposed focused on the improvement of three aspects of IMS emergency communications, namely:

QoS and resource management; context-awareness and service personalization; and the use of richer emergency communication models. The enhancement of the QoS/resource management aspect of emergency sessions was achieved by generalizing our previously proposed call differentiation solution. This generalization implied the definition of new QoS profiles for emergency sessions, the extension of the proposed IMS call differentiation architecture, the adjustment of the used resource management strategy, and the elaboration of the related emergency session management scenarios. Unlike the existing IMS emergency service architecture, our QoS-enhanced architecture provides preferential treatment to all categories of emergency communications, and prioritizes their access to resources in an efficient and adaptive manner.

The second enhancement focused on the exploitation of a wide range of contextual information as means to improve the efficiency of emergency operations and offer personalized emergency services to users. This was achieved by the elaboration of a personalized, context-aware, emergency service scenario, and the extension of the IMS emergency service architecture to enable its support.

Finally, the third enhancement tackled the improvement of the IMS emergency service architecture with multi-party session support capabilities, as means to offer richer forms of emergency communications to 3G users. This was achieved via a case study including the definition of potential conferencing models for public-initiated emergency calls and mission critical calls, and the extension of the IMS emergency architecture to enable the support of these models. Furthermore, different multi-party emergency sessions' related scenarios were elaborated.

Chapter 7

Design and Implementation of Proof-of-Concept Prototypes and Applications

In this chapter, we present the different proof-of-concept prototypes used for the validation of the solutions proposed in the previous chapters. We also present two context-aware applications demonstrating the use of context at the IMS service level.

7.1 WSN/IMS Integrated Architecture Prototype and Applications[1]

In order to validate our IMS context management solution (presented in chapter 4), we developed a proof-of-concept prototype focusing on the case of WSN/IMS integration (since it was the most challenging one), and used it for the collection of performance measurements. Furthermore, to illustrate how new applications can be built using the capabilities of our system, a pervasive game called 'Fruit Quest' and a personalized call control application 'Sense Call' were developed.

In the coming sub-sections, we start by presenting the prototype architecture and the architectural components' design. This is followed by a presentation of the application scenarios and the setups used to test them, along with the performance evaluation.

7.1.1 Prototype Architecture

To build a proof-of-concept prototype of our architecture, we used Ericsson's Service Development Studio (SDS) [102] as implementation platform and extended the existing JAIN presence server [103] to satisfy our requirements. As for the WSN/IMS gateway, it was implemented from scratch, while the two applications were used to test the system's

[1] The implementation of this prototype and the collection of its related performance measurements were conducted by a master's student working in our laboratory, as detailed in [101].

operation. Figure 7.1 illustrates the different prototype components.

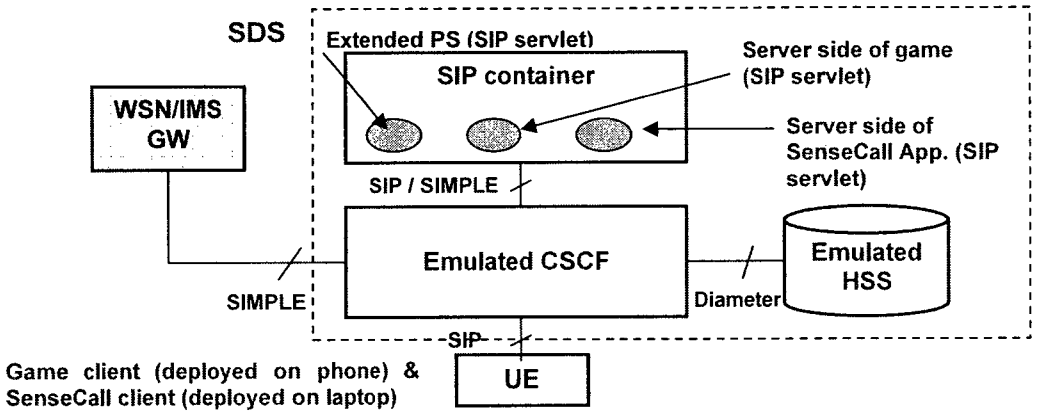


Figure 7.1: The WSN/IMS integrated architecture prototype components

SDS is an Eclipse-based design and execution environment where IMS applications can be designed, deployed, and tested [102]. Several features are provided by SDS, such as: an Integrated Development Environment (IDE); a set of service APIs facilitating the development of client/server side applications; and an IMS simulated environment simulating CSCFs, an HSS, and an application server acting as container for the deployment of SIP servlet based services. In the prototype, the JAIN PS [103] (originally relying on a JAIN SIP stack for communication) was remodeled as a SIP servlet to enable its deployment in the SDS application server. Furthermore, the presence server's XML schema was extended with the additional data elements, and its logic was enhanced with the publication trigger mechanism. The server side of the gaming application was implemented as a SIP servlet and deployed in SDS application server, while the game clients were developed using SDS IMS client platform and installed on P990 Sony Ericsson phones. The sever side of Sense Call [104] (originally developed as a SIP-based standalone Java application) was remodeled as a SIP servlet and ported to SDS. Furthermore, the application logic was modified to communicate with the PS (instead of direct communication with the web service based gateway on which it relied) to obtain the

required information. In this prototype, two types of sensors were used: MIT Cricket location sensors [105] accessible via the Cricket API; and the MTS300/Mica2 environmental sensor [106] accessible via the Crossbow API.

As for the WSN/IMS gateway, it was implemented as a JAVA-based extended presence agent relying on a Microsoft access database and a set of APIs (namely, the JAIN SIP API, the Cricket API, and the Crossbow API) in its operation. In the coming section, we describe the designs of the WSN/IMS gateway and the extended presence server in details.

7.1.2 New Components' Design

7.1.2.1 The WSN/IMS Gateway Architecture

The WSN/IMS gateway plays a key role in our architecture, by acting as intermediary between WSNs and the 3G network. Figure 7.2 depicts the proposed gateway architecture, which consists of two layers: a connectivity layer and an abstraction layer.

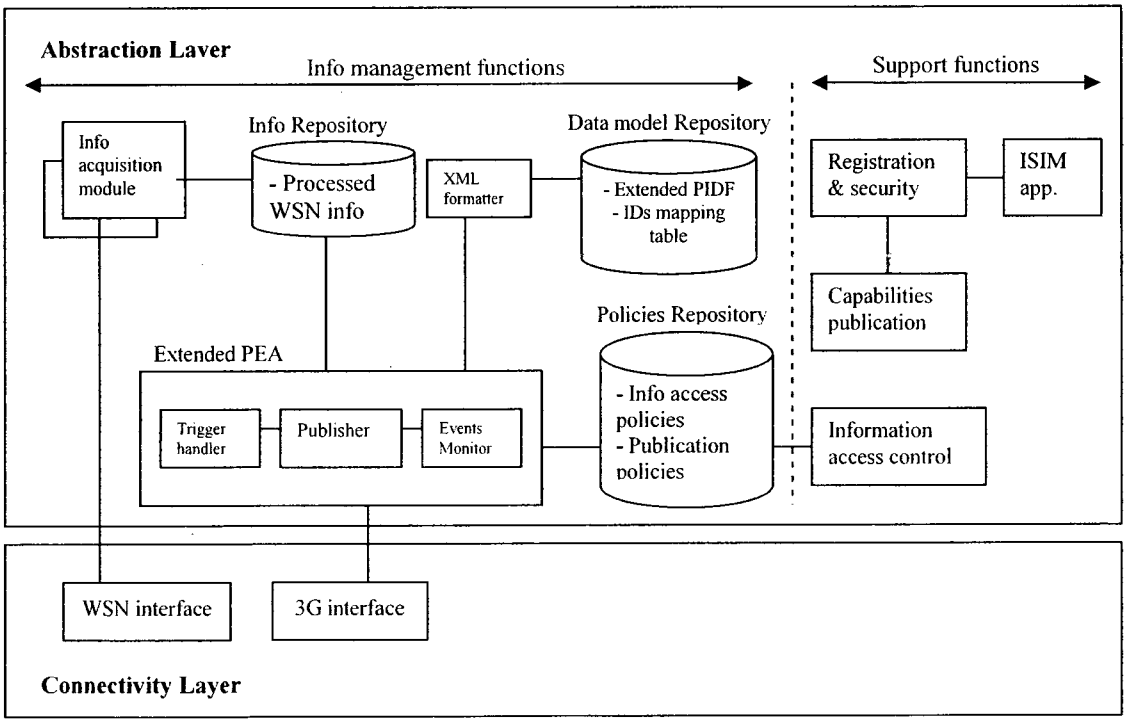


Figure 7.2: The WSN/IMS gateway architecture

The connectivity layer includes a dual networking interface ensuring the connectivity to both WSNs and the 3G network. The abstraction layer is responsible of conveying the information captured by WSNs to the IMS, after the proper processing and formatting. It consists of two types of functions: information management functions; and support functions.

The support functions are realized by the following modules: the registration/security module; the capability publication module; and the information access control module. The registration/security module is the first module invoked when the gateway is put in service. It interacts with the ISIM application (contained in the gateway's SIM card) to get the needed information for IMS registration and security association establishment (e.g. public/private identities, and long term secret), builds the first SIP REGISTER message, and interacts with the capabilities publication module that inserts the gateway capabilities information in the message body. The registration module then carries the rest of the IMS registration procedure as described in section 4.4.1. After the registration phase, the information access control module communicates with the PS to set the needed subscription authorization policies. These policies are pre-configured in the gateway's policies repository, which also contains publication policies indicating the types of information that should be published within regular time intervals; and the ones that are published based on events' detection. The subscription policies are used by the PS to install filters that determine which watchers are allowed to access the information related to a certain entity. It should be noted that among the different support functions, only the registration and security related functions were implemented in our prototype, while the capability publication and the access control functions were omitted, for simplicity.

As for the information management functions, they are carried by a set of information acquisition modules, an XML formatter, an extended PEA, and three repositories. The first repository (the data model repository) contains the extended PIDF we defined as information model and mapping tables correlating IMS entities' IDs to sensor IDs, while the second (the information repository) contains the processed WSN information that is persistently stored in the gateway for future publications. The third repository is the policies repository presented previously. As for the information acquisition modules, they are specialized components enabling the interaction with various WSNs. There is one acquisition module per WSN type. Such module is capable of extracting sensor-specific data from WSN messages, and pre-processing this information (by performing data fusion and consistency checking), before storing it in the gateway's information repository. The extended PEA represents the heart of the WSN gateway. It publishes WSNs information to the IMS, based on the publication policies defined in the gateway. Two modes of publications are supported by the extended PEA: the proactive mode in which information is actively published by the gateway on regular time intervals or when certain events are detected; and the reactive mode in which information is only published upon the receipt of a trigger from the PS. These two modes of publication are realized by the PEA sub-modules as follows: Based on the publication policies, the publisher saves a list of information that should be proactively published within regular time intervals, and following those intervals, it consults the information repository to get the needed information, which is passed to the XML formatter. This last consults the IDs mapping tables and the extended PIDF to represent the processed information in a standard format, and then returns the resulting XML document to the publisher that publishes it to the PS.

Similarly, the events monitor saves a list of information to be proactively published upon the detection of events (e.g. publish temperature when above 30 oC), and keeps interacting with the information repository to detect the occurrence of those events. Once an event is detected, the events monitor interacts with the publisher that will fetch the needed information and publish it (after proper formatting) to the PS. As for the trigger handler, it does not actively publish any information. However, once it receives a publication trigger (from the PS), it contacts the publisher that will convey the needed information to the PS. Figure 7.3 illustrates the interactions between the different information management components.

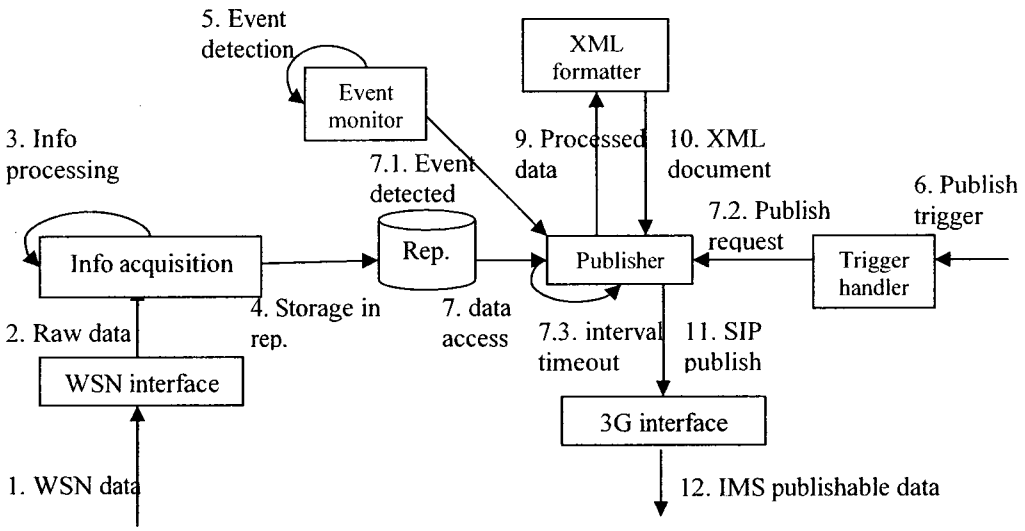


Figure 7.3: Interactions between the gateway's information management components

7.1.2.2 The Extended Presence Server Architecture

Figure 7.4 depicts the software architecture of the extended presence server used in our prototype. This architecture consists of protocol and service related components. The protocol supported in this case is the SIMPLE protocol, which is accessible via the SIP servlets API v.1.0 [107]. As for the service component, it consists of a presence service logic module implementing the logic of the PS engine. This module relies on several sub-modules in its operation, namely: a publication/subscription manager that handles

information publications and subscriptions from presentities (i.e. presence user/network/external agents) and information watchers; a notifier that creates and sends information notification messages based on received subscriptions (these notifications could be sent following regular time intervals or upon the detection of events); an events monitor that monitors the collected information and detects the occurrence of events possibly leading to information notifications; a trigger generator that generates publication triggers to prompt the publication of information that is missing or not fresh enough in the network; and an authentication and authorization module that is responsible of the authentication of publishers/watchers and the enforcement of subscription authorization policies (needed for info access control).

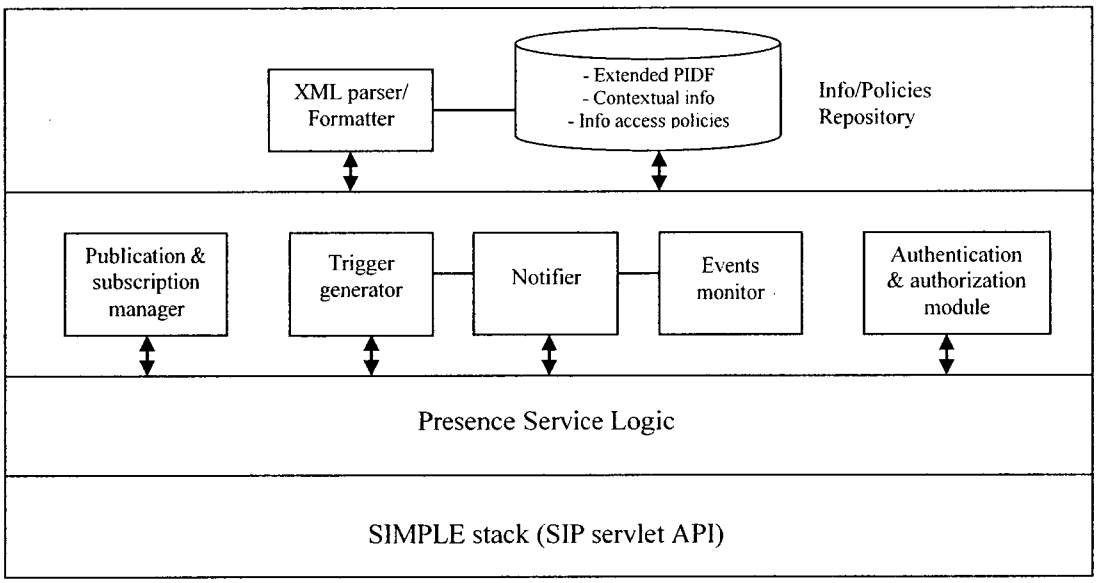


Figure 7.4: The extended presence server architecture

It should be noted that these sub-modules rely on an XML parser/formatter for the extraction of information from received messages and for the XML formatting of information to be inserted in newly created messages; and they rely on an information/policies repository for the storage of: the extended PIDF, the information access policies, and the contextual information that is stored for future notifications.

7.1.3 Context-Aware IMS Applications and Prototype Setups

In this section, we show how new context-aware IMS applications can be built using the capabilities of our architecture. We start by describing how context can play the role of an application building block in the IMS, and then discuss some of the application areas that could benefit from the availability of contextual information in the network. This is followed by the description of two concrete application scenarios related to these areas and the setups used to test them.

7.1.3.1 Context as Application Building Block and Potential Application Areas

Introducing context acquisition/management components as part of the IMS infrastructure and enabling the access of their capabilities via standard IMS interfaces, can abstract applications developers from the details/complexity of context management related operations, thus facilitating the development of context-aware value added services. In fact, developers do not need to use any proprietary APIs or sophisticated information processing operations when developing new applications/services to obtain and process the needed contextual information, since this information is already available in the IMS and can be accessed (as any other IMS capability) via standard interfaces.

Figure 7.5 illustrates how context can play the role of an application building block (or service enabler) in the IMS architecture and shows the three levels of services that can be offered by the IMS. The first level consists of basic services offered by the IMS infrastructure, such as session control, subscription management, and charging. The second level includes more advanced services running on top of the IMS, and which can serve as service enablers. The context management service can be considered as one of such services. Another example is the OMA service enablers. As for the third level, it consists of value-added services that are not standardized by 3GPP but could be offered by third-party

service providers. Such services can rely on the capabilities of service enablers and/or the capabilities of the IMS infrastructure in their operation.

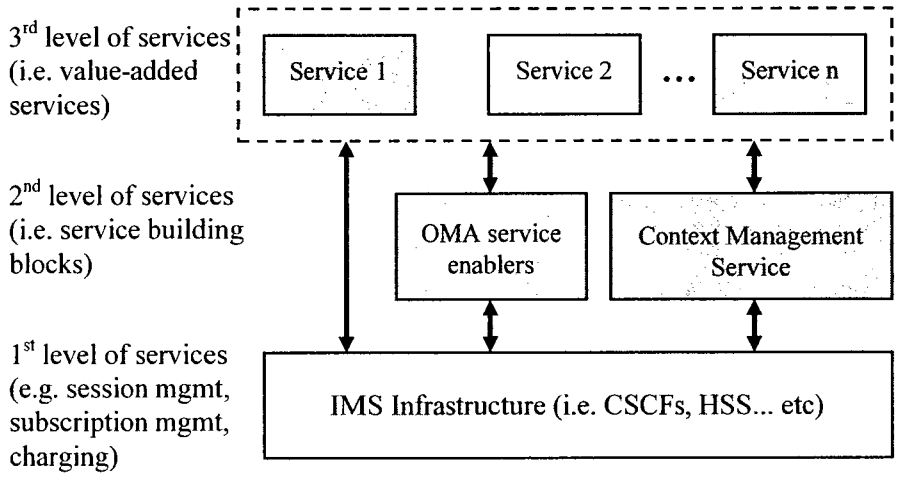


Figure 7.5: Context serving as an application building block in the IMS

Several application areas can benefit from the availability of contextual information in the network. Examples of such areas include:

- **Wireless Healthcare:** With the increase in life expectancy and the number of aging populations, there is a significant interest in wireless healthcare applications that could improve the quality of life of elderly and chronically sick people. By monitoring and interpreting patients' physiological data, these applications could offer personalized medical assistance under problematic health conditions, and increase the efficiency of the health care system by reducing its costs while maintaining a high level of service performance. Consider for instance an application that enables the real-time monitoring of Alzheimer's patients, and their assistance by offering certain recommendations via their mobile terminals using IMS messaging (e.g. take your medicine, go to the doctor, take left to return home). In addition, the collected contextual information could be conveyed to care givers or relatives in case of danger, and can be used by doctors for offline diagnosis and progress assessment. Another potential application could track

heart patients' health conditions, and automatically calls for help (e.g. call an ambulance), upon the detection of an incoming stroke.

- **Pervasive Gaming:** Pervasive games are an emerging type of interactive games that use the real world as a platform and involve multiple types of media and virtual game elements. Players typically interact with physical and virtual objects and characters in the game to achieve a certain goal. This form of gaming is gaining a lot of popularity and could be offered in a sensors-enabled IMS environment, in which sensors capture the game context and use it to adapt the gaming experience of each player.
- **Personalized Lifestyle Assistance:** Many context-aware applications could be provided to 3G users, to assist them in their daily activities. Examples of such applications include: interest-based services, real-time training support, and smart shopping applications. Consider a training assistance application that can monitor the physiological conditions of sports amateurs and professional athletes, and generate automatic recommendations and visual demonstrations concerning the adaptation of their training programs (e.g. changes in training exercises and visual demonstration on how to perform the new exercises) using multimedia messaging. As for interest-based services, they could range from targeted multimedia advertisement about the services that could be of interest to the user based on his situation (e.g. advertising the nearest restaurants when the user hasn't eaten for five hours), to the automatic (pre-booked) establishment of a call between two colleagues, when they are in their respective offices. Finally, RFID-tagged merchandize could enable smart shopping, by guiding the user in the store towards the items he/she wishes to purchases and advertising discounts on his/her preferred products.

7.1.3.2 The Fruit Quest Game and the Sense Call Application

To illustrate the pervasive gaming applications category, we used a strategic pervasive game designed in our lab, called 'Fruit Quest'. In this game, players are assigned plantation zones, in addition to some virtual game objects (e.g. fruits, walls, bombs, and virtual money). WSNs are used to detect and convey the location of players and their presence in zones to the network, and this information is used to adapt the players' gaming experience. The game scenario can be described as follows: players physically move between plantation zones within the game area, and as they move, they see the zones appearing on their terminals and get notifications about game events. They can also communicate with each others using IMS instant messaging. When players are in their plantation zones, they can plant fruits and add defensive walls for protection. When in rivals' zones, players can pick fruits and attack the zones using bombs. When all defensive walls in a zone are destroyed, the zone can be occupied by rivals. When the time of the game ends, the player with the highest number of zones and fruits wins the game.

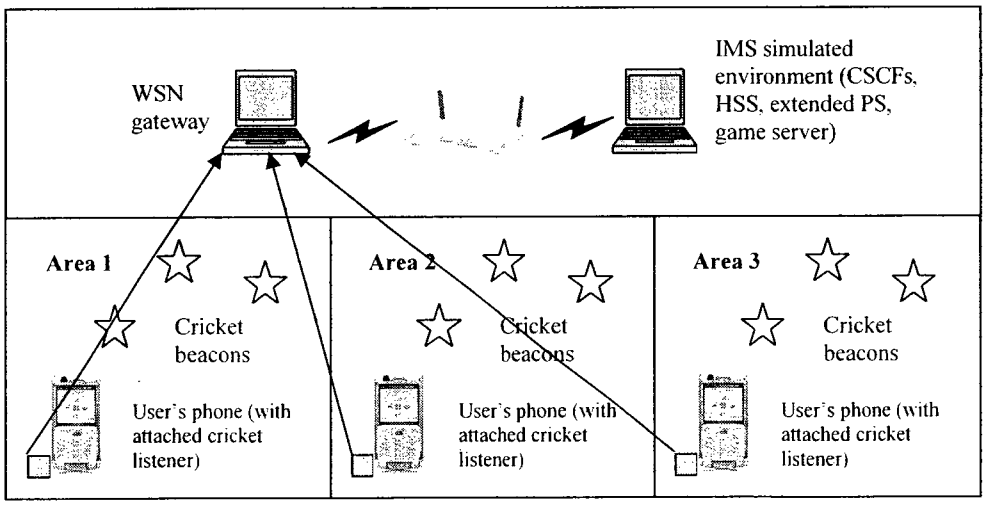
'Sense Call' [104] is a personalized call control application previously developed in our lab as part of another project. This application monitors users' locations and enables the automatic (pre-booked) establishment of a call between two colleagues when they are in their respective offices. To illustrate the capabilities of our system, Sense Call was remodeled and deployed in our WSN/IMS integrated environment.

7.1.3.3 The Prototype Setups

A shown in figure 7.6a, the Fruit Quest game setup consisted of two laptops and three phones, forming a WLAN, in addition to a set of MIT Cricket location sensors. The game clients were running on the phones, while one of the laptops represented the IMS simulated

environment (including the game server) and the other laptop represented the WSN gateway. The following interactions related to the pervasive gaming scenario were successfully tested: first, the WSN gateway was registered as an IMS corporate user (using the identity of the game provider). Then three players, each carrying a phone with an attached cricket listener, started moving between three game zones. Cricket beacons mounted to the ceiling were used in conjunction with the cricket listeners attached to the phones to determine the location of the players. This information was conveyed (by cricket software running on phones) to the WSN gateway, using TCP/IP communication. The gateway monitored the information received and when it determined that a player has moved to another game zone, it published this event (using a SIP PUBLISH message) to the extended PS, which notified (using a SIP NOTIFY message) the game server. This last then sent the appropriate game updates (based on the received information) to the game clients hosted by the players UEs, which updated the game display. Figure 7.6b shows a sequence diagram detailing the game operation.

a)



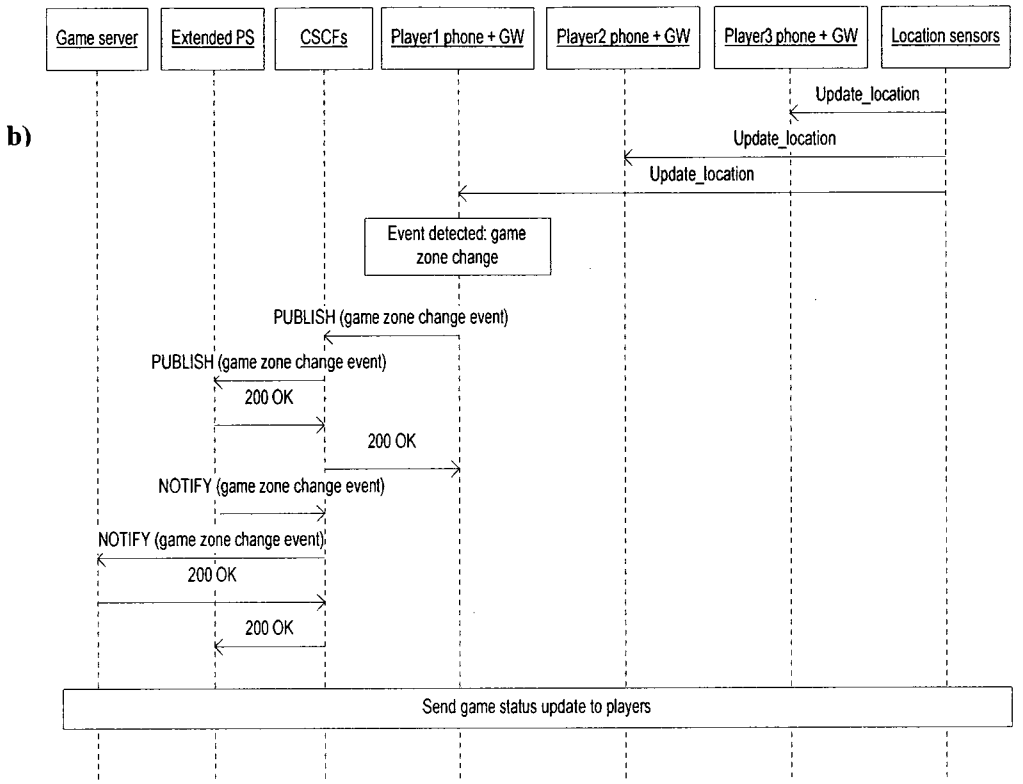


Figure 7.6: The Fruit Quest Prototype: a) Fruit Quest game setup; b) Fruit Quest game scenario

A similar setup was used for the Sense Call application, as depicted in figure 7.7a In this case, the application clients were installed on two laptops, while the IMS simulated environment hosted the server side of the application. Then, the following interactions were successfully tested: first, the two clients were registered as IMS users and the server side of the application was used to schedule a call between them. Then, the users carrying their laptops (with attached cricket listeners), started moving in the office space and their location information was conveyed to the gateway. Upon the detection of a location change, the gateway published this event (using a SIP PUBLISH message) to the PS, which notified the server side of the application (using a SIP NOTIFY message). When the application detected that two users were in their respective offices, it established a third-party controlled call between them by sending a SIP REFER message to one of the users'

UE. This last accepted the referral using a SIP 202 message, then sent a SIP INVITE message to UE2. When the call was established successfully, UE1 notified the Sense Call application about the result of the referral event, as shown in figure 7.7b.

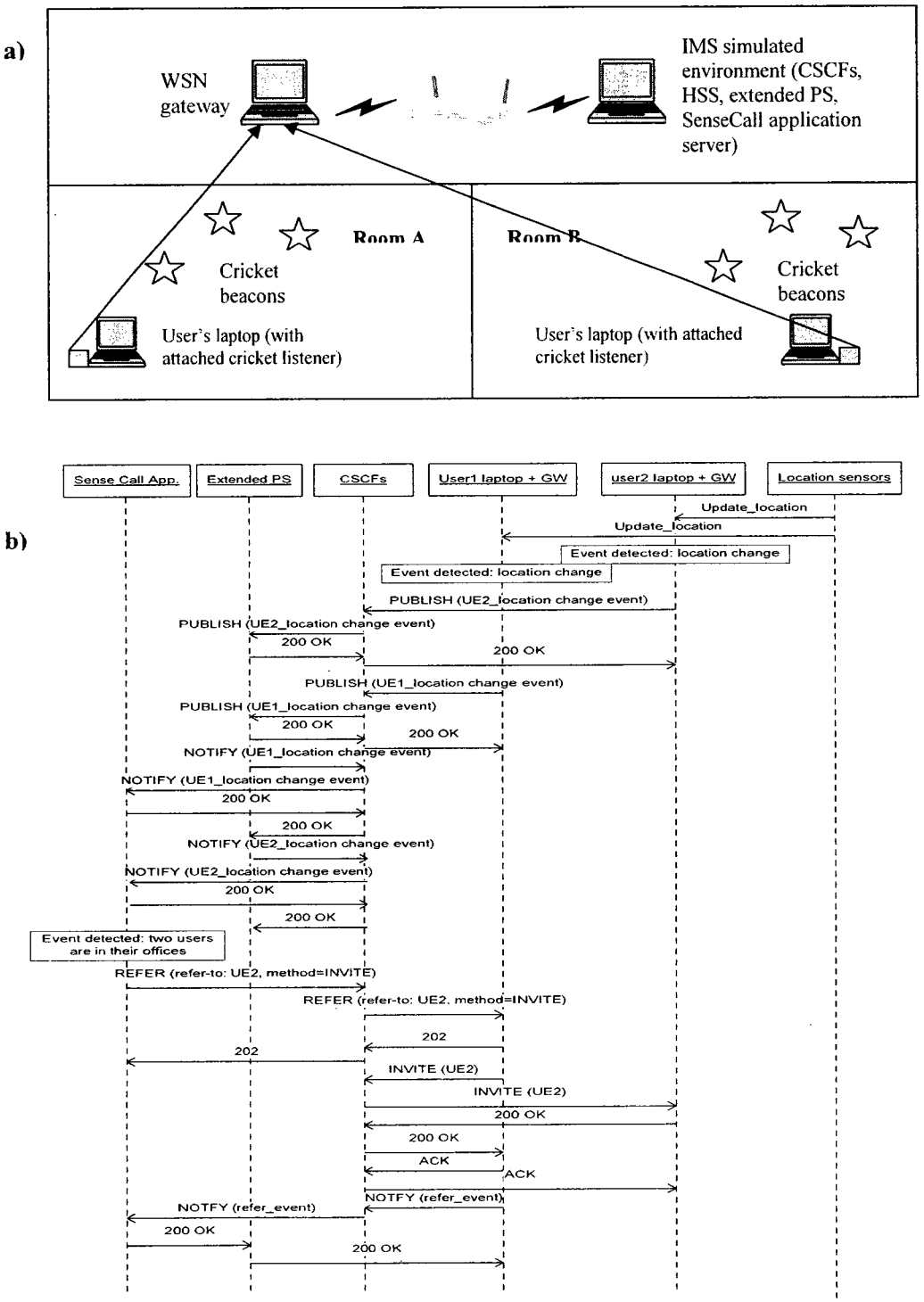


Figure 7.7: The Sense Call application prototype: a) Sense Call application setup; b) Sense Call application scenario

7.1.4 Performance Evaluation

To evaluate the performance of our system, we used the Fruit Quest prototype to collect some measurements, focusing on the publication interactions (between the GW and the PS) and the notification interactions (between the PS and the game server). Spatial (i.e. location) and environmental (i.e. light/temp) data was collected, and two performance metrics were used: the response time (in ms) and the network load (in bytes).

In addition to the sensors, the test bed consisted of the following: the WSN/IMS gateway running on a Pentium 4/2.5 GHz laptop, with 512 MB RAM and the Windows XP platform. The laptop was attached with an MIB510 sink node allowing it to communicate and collect data from sensor nodes – this data was monitored using a MoteView application which is installed on the laptop. A second laptop with a similar configuration (i.e. 1.6 GHz Intel Pentium Duo with 1 GB RAM, running Windows XP) hosted the IMS simulated environment (i.e. the CSCFs, HSS, and the extended PS), while a third laptop with an identical configuration hosted a second instance of the IMS environment in which the Fruit Quest game server was deployed. It should be noted that the game server's logic was slightly modified (for testing purposes) to subscribe/accept environmental information from the PS, in addition to the location information it originally used. Furthermore, three Sony Ericsson P990 phones, with attached Cricket listeners and running the Symbian operating system and the IMS client platform (provided with SDS), hosted the game clients. Table 7.1 shows some of the measurements collected using this test bed. These values are average measurements over 20 trials.

Operation	Scenario	Response time (ms)	Network load (bytes)
Publication	Proactive – location info	205	1139
	Proactive – Environmental info	178	1067

	Reactive – location info	228	2371
	Reactive – Environmental info	214	2300
Notification	Location info	224	1335
	Environmental info	164	1241

Table 7.1 Network load and response time measurements for the WSN/IMS integrated architecture prototype

In the presented measurements, the response time for proactive publications is calculated at the gateway level, as the time duration between the moment when information is accessed by the gateway's publisher module (from the info rep.) and the message is created/sent, until a successful publication response is received from the PS. For reactive publications, the response time (also measured at the gateway level) is calculated from the moment a publication trigger (i.e. SIP OPTIONS message) is received from the PS, acknowledged and responded to by a PUBLISH message, until a successful publication response is returned by the PS. As for notifications, the response time is measured at the PS, from the moment the information is internally accessed and the message is created/sent, until a successful notification response is received from the game server. Figure 7.8 illustrates how these measurements are calculated.

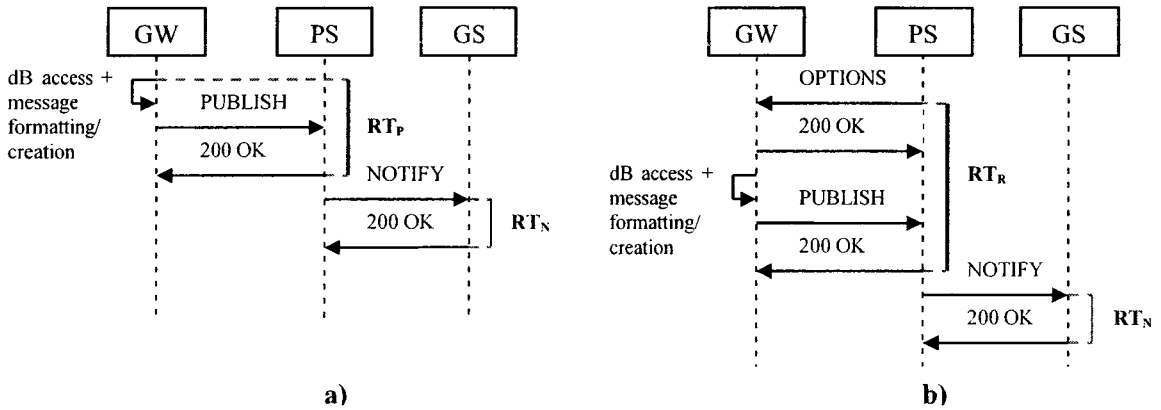


Figure 7.8: Response time calculation: a) For proactive mode; b) For reactive mode

Several types of comparative analysis were made by examining the collected measurements. The first analysis was made by comparing the performance of the two

modes of publication for the same type of information (e.g. proactive-location vs. reactive-location), in order to calculate the overhead introduced in the case of the reactive mode. This overhead is caused by the exchange of an additional pair of SIP messages (i.e. OPTIONS and OK messages) to trigger the publication, and by the processing of the publication triggers contained in the OPTIONS message body. The average overhead, in terms of response time, ranges between 23 ms (for location info) and 36 ms (for environmental info) per operation, which can be considered as non-significant since its effect will be barely felt by the end-user. The penalty in terms of network load is nevertheless signification (an increase of 1.2 Kbytes/operation, for both types of information). However, this penalty will only be incurred occasionally since the reactive mode is a secondary mode of operation which is only used when the required contextual information is not available (or not fresh enough) in the network.

By comparing the performance of one mode of publication for two different types of information (e.g. proactive-location vs. proactive-environmental), we can see how the type of information exchanged can affect the performance. The same type of comparison can be made for notifications of different types of information. In general, we notice that the publications/notifications of environmental data achieve better response times and induce less load in comparison to location information related interactions (e.g. a decrease of 27 ms and 72 bytes for proactive-environmental publication in comparison to proactive-location publication). This is due to the fact the number of XML fields/tags required to represent location information is bigger than the one needed to model environmental data, thus requiring more time for XML formatting and generating bigger message payloads. The performance of location information related interactions could therefore be improved

by using another modeling schema necessitating a smaller number of tags for the representation of this type of information.

We also examined the factors affecting the response time and network load. Several operations contributed to the response time achieved for different operations, such as: the dB access time; messages' creation/processing time; and messages' sending/receiving time. To illustrate the weight of these different operations, we take the proactive-location information publication case, in which the dB access time and the PUBLISH message creation time accounts for 20% of the total response time, while the sending/receiving of the publish/OK message accounts for the remaining 80%. As for the network load, it mainly depends on the size of the messages exchanged (i.e. the size of the headers and payloads they are composed of). In the conducted measurements, the payload sizes varied between 428 and 500 bytes (i.e. around 40% of PUBLISH messages size and 35% of NOTIFY message size) depending on the type of data carried.

7.2 Prototype for IMS Call Differentiation Architecture

In this section, we present the proof-of-concept prototype we developed to validate our IMS call differentiation solution, which was presented in chapter 5. We start by describing the different prototype components, and then detail the software architecture of the new components, along with the extensions made to existing protocol stacks. This is followed by a presentation of the prototype setup and the different test scenarios executed.

7.2.1 Prototype Architecture

We extended the prototype presented in the previous section to demonstrate the feasibility of the IMS call differentiation solution proposed. As depicted in figure 7.9, several new

components were added to the previous prototype, namely: a SPF responsible of resource management and call admission control; a combined Logical Sensors Gateway/Dummy Context Source (LS-GW/DCS) generating and conveying network status information to the extended PS (acting as CIB); a conference focus acting as centralized control point responsible of the management of multiparty sessions; in addition to a combined CDF/CGF, an IMS-GWF, and an OCS representing charging related components.

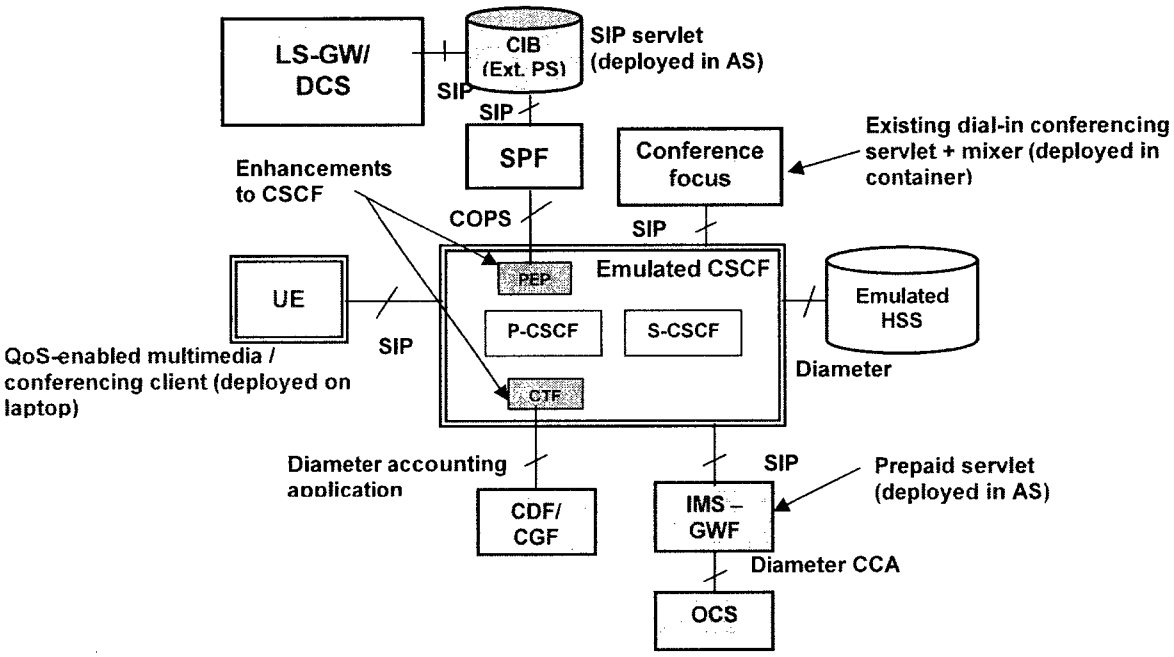


Figure 7.9: The IMS call differentiation architecture prototype components

Beside the conference focus and the IMS-GWF that were implemented as SIP servlets and deployed in SDS application server, all the other components mentioned were developed as Eclipse plug-ins and integrated with SDS. The integration was performed at the level of the CSCF, which was enhanced with two new modules, namely: a PEP module allowing the interaction with the SPF (using COPS) for resource allocation/re-allocation decisions; and a CTF enabling the reporting of offline chargeable events to the CDF (using Diameter).

We note that, in the case of online charging, iFC are set in the users' profiles (defined in the HSS) to specify the messages that should be directed by the CSCF to the IMS-GWF.

Furthermore, the sequence of modules involved in the CSCF operation is specified in an XML-based configuration file. Several modes of operations were defined in this file, including: with/without call prioritization; and with/without charging. The second and the fourth modes (i.e. without call prioritization and without charging) allow the bypassing of the PEP and the CTF modules added to the original CSCF design.

In terms of protocols, an open source implementation of the COPS protocol stack [108] was used for the implementation, in addition to the SIP and the Diameter protocol stacks employed by SDS. The COPS protocol was extended with a new policy client type and client related objects, while SIP was extended with two headers needed for QoS negotiation related interactions. As for the Diameter protocol, it was extended to carry additional information related to multi-grade service charging. On the client side, we developed a simple multimedia client, enhanced with the ability to label/re-label sessions as well as conferencing capabilities. In the coming sub-sections, we highlight the software architecture of the new components; describe the enhancements made to the existing protocol stacks; then present the prototype setup and test scenarios.

7.2.2 New Components' Design

7.2.2.1 Resource and Context Management Components

Figure 7.10 depicts the software architecture of the resource management component (i.e. the SPF) and the context management related components (i.e. the CIB and the combined LS-GW/dummy context source). As shown in part A of the figure, the SPF is composed of a policy repository and two modules: The Call Admission/Control Module (CACM); and the Information Access Module (IAM). The CACM acts as a PDP, making decisions about the admission of new calls and the control of ongoing calls, according to the call admission control mechanism and the media parameter control mechanism presented in chapter 5.

The decisions made by the CACM take into consideration the contextual information it collects from the CIB (via the IAM) using queries and subscriptions/notifications, as well as the resource management policies it consults by accessing the policies repository. As for the CIB, it acts a support entity that is responsible for the management of the contextual information needed for the operation of the SPF. As mentioned previously, the role of the CIB is realized by the extended PS whose architecture was detailed in the section 7.1.2.2.

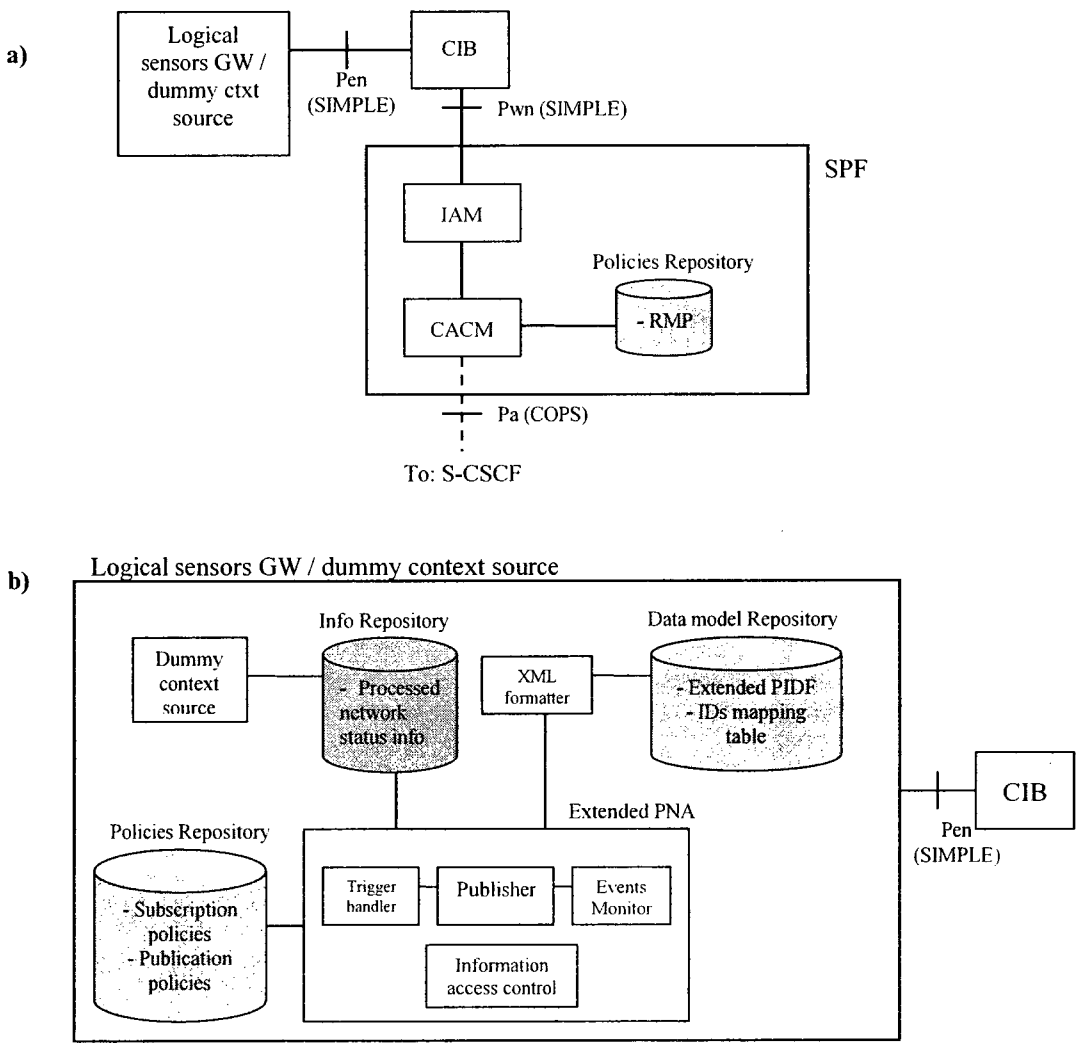


Figure 7.10: Software architecture of the resource and context management related components: a) SPF software architecture; b) Combined logical sensors GW/ dummy context source software architecture

Part B of the figure depicts the architecture of the combined LS-GW/DCS, which encompasses the following components: A data model repository containing the extended PIDF used as information model and mapping tables correlating IMS entities' IDs to logical sensor IDs; an information repository containing the processed network status information that is persistently stored for future publications; a policies repository containing publication and subscription policies; an XML formatter used for the representation of the processed information in a standard XML format; a dummy context source used for the generation of different contextual values (representing different network conditions); and an extended PNA publishing the processed/formatted network status information to the CIB, in addition to setting the needed subscription authorization policies controlling access to the published information. It should be noted that this architecture represents a simplification of the WSN/IMS gateway architecture (presented in section 7.1.2.1), since both gateways play similar roles with the exception of some of the support functions (e.g. security, registration, and capabilities publication) that are not required in the logical sensors gateway case. It should also be mentioned that we relied on a dummy context source for the generation of network status information due to the impossibility to communicate with lower layer nodes (e.g. routers) for the collection of this information in our prototype. Furthermore, the access control function was not implemented in our prototype, for simplicity.

7.2.2.2 Charging Components

Figure 7.11a depicts the software architecture of the IMS-GWF. This last was implemented as a SIP servlet which was deployed in the SIP container. The servlet consists of a back to back user agent, implementing pre-paid service logic and also acting as diameter credit control client. Moreover, iFC are set in pre-paid users' profiles to instruct the S-CSCF to

forward appropriate SIP messages to the IMS-GWF. The IMS-GWF operates as follows: when a SIP message is received by the container, it activates the pre-paid servlet. The servlet examines the message to determine if it is a triggering message. If so, it extracts the needed information from it and creates an instance of the diameter CC client, to which those parameters are passed. These parameters are used to build the appropriate AVPs inserted in the CCR, which is sent to the OCS. After receipt of a CCA, the CC client returns the control to the servlet that forwards the SIP message back to the S-CSCF (or creates an error or BYE message). The granted service unit (GSU) AVP extracted from the CCA is used by the CC client for unit supervision. It should be noted that, in addition to instantiating the CC client, the servlet also starts a listener to listen for feedback from the client during the session (e.g. to terminate the session if credits are over). Simple commands are sent to this listener over a TCP socket (e.g. Notify: no credit left).

Figure 7.11b illustrates the OCS architecture. In this architecture, the OCS acting a credit control server, is composed of the following modules: the session-based charging function (SBCF) responsible of the credit control of sessions; the event-based charging function (EBCF) responsible of the credit control of events; the rating function (RF) calculating the price of the service; the account balance management function (ABMF) managing users accounts' balances; and the accounts repository storing users' accounts information. It should be noted that the OCF (a component of the OCS that is composed of the SBCF and the EBCF) interacts with the IMS-GWF's CC client as follows: when it receives a CCR, it extracts the needed information, checks the user account balance (by interacting with the ABMF and the RF) and authorizes a certain amount of credit based on the available credit and the rating of the of the chargeable event.

The combined CDF/CGF is shown in figure 7.11c. This entity gets ACR messages from the CTF (embedded in the CSCF), and generated appropriate CDRs that are locally stored to be sent latter the billing domain (using FTP).

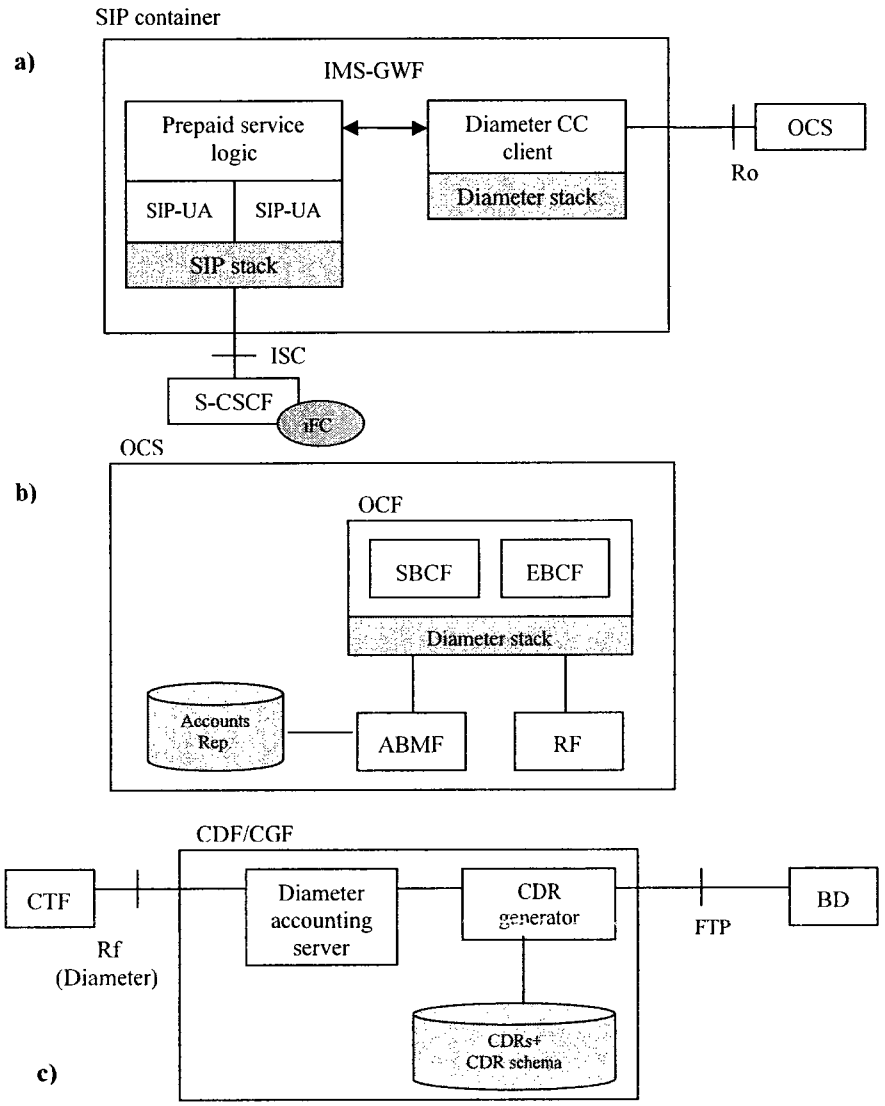


Figure 7.11: Software architecture of the charging related components: a) IMS-GWF software architecture; b) OCS software architecture; c) Combined CDF/CGF architecture

7.2.2.3 Multiparty Session Management Component

The IMS simulated environment provided by SDS lacks support for conferences due to the absence of the conference focus and the MRFP components as part of its architecture. In order to test multiparty sessions related scenarios, we reused an existing dial-in

conferencing servlet [109] and an existing mixer [110] previously developed in our group, in order to simulate the operation of the missing conferencing components. The conferencing servlet, which was deployed in SDS SIP container, played the role of a simplified conference focus by enabling the creation/joining/leaving of dial-in conferences – other functions such as policy and floor control not being supported. As for the mixer, it played the role of a simplified MRFP.

Figure 7.12 depicts the architecture of the conferencing components. In this architecture, the dial-in servlet maintains signaling links with all the conference participants, and interacts with a local mixer for the mixing of media streams and the management of media connections. We note that the capabilities of the dial-in servlet are accessible via the SIP servlet API it supports. Furthermore, the interaction between the signaling and the media control units is performed using a Megaco-based API provided by the media unit, and which supports methods such as: addStream(), substractStream(), and moveStream().

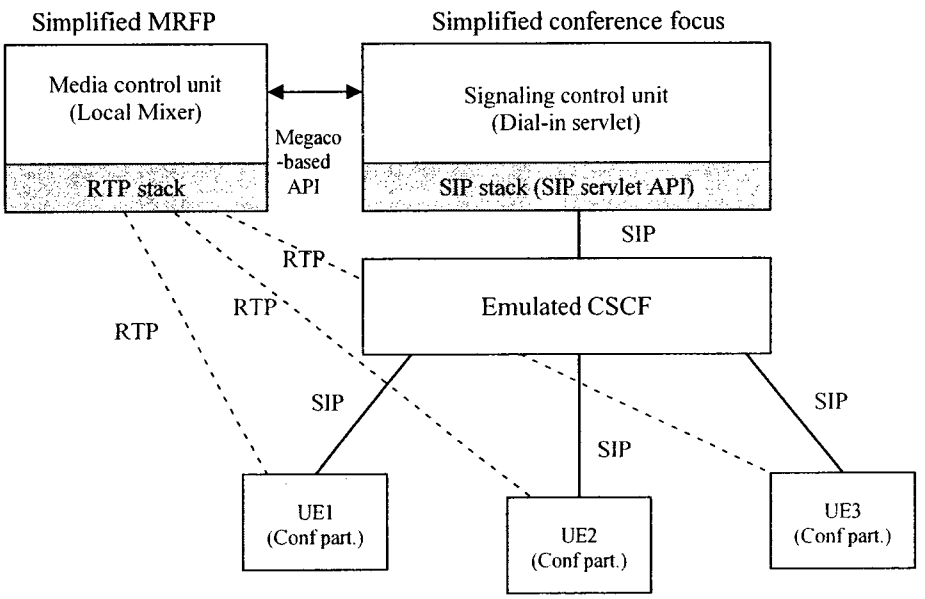


Figure 7.12: Software architecture of the conferencing components

7.2.3 Protocols Extensions

In order to support the operation of our IMS call differentiation architecture, we made some extensions to the three protocol stacks used in our prototype. These extensions are described in the coming sub-sections.

A. COPS Extensions

COPS is used in our architecture to enable the exchange of policy-based resource allocation/re-allocation decisions between the CSCF and the SPF. To achieve this role, we extended COPS with a new policy client type representing the CSCF (with code 0x800e) and client related objects (i.e., new clientSI data fields, decision commands, and context flags). Table 7.2 summarizes these extensions and their usage in different COPS messages.

Field name	**Encapsulating element**	**Message types employing info**	**Field description**
Client-type	Common header	All COPS messages	Identifies policy client involved in session. The code 0x800e was chosen to identify the CSCF as a new policy client type
CI	ClientSI object	REQ messages	2 bytes field indicating the call ID extracted from SIP messages (to bind a specific admission request to a certain SIP session)
CC	ClientSI object	REQ messages	1 byte field specifying the request service class (mapped from info extracted from resource-priority header in SIP message). This field can have one these 5 values: 1: Platinum (mapped from RP=Q.735.1) 2: Gold (mapped from RP=Q.735.2) 3: Silver (mapped from RP=Q.735.3) 4: Emergency-public (mapped from RP=Q.735.0.1) 5: Emergency-authority (mapped from RP=Q.735.0.2)
MT	ClientSI object	REQ messages	1 byte field specifying the media type(s) to be used in the session (mapped from the session description info extracted from the body of SIP messages). This field can have one these 5 values: 1: Audio 2: Audio/Video 3: Text 4: Audio/Text 5: Audio/Video/Text

M-Type	Context object	REQ and DEC messages	Specifies the type of request that triggered the query, with possible values: 1: New-2Party-Request 2: Join-Multiparty-Request 3: ReferTo-Multiparty-Request 4: CategoryChange-2PartyCall 5: CategoryChange-MultipartyCall
Command-Code	Decision object	DEC messages	Specifies type of decision made by PDP, as one of 5 possible values: 0: Null decision 1: Install (admit request) 2: Remove (reject request) 31: Trigger_downgrade 32: Trigger_termination

-----: Field enhanced with new values -----: Newly defined field

Table 7.2: COPS extensions

B. SIP Extensions

To achieve QoS negotiation between the user and the network, we enhanced the SIP stack (provided with SDS) with two existing extension headers, namely: the RP header [95] and the enhanced reason header [96]. In our case, sessions were assigned the following priority values according to the Q.735 namespace: platinum=*Q735.1*; gold= *Q735.2*; silver= *Q735.3*; emergency-public=*Q735.0.1*; and emergency-authority=*Q735.0.2*. For each session, one of these values is included (depending on the service class chosen by the user) in the RP header of SIP INVITE messages (used for the creation, the joining, or the re-negotiation of sessions). As for the enhanced reason header, it was used to indicate the different preemption events triggering certain SIP requests. In our case, two values were defined: *hard-preemption* (to indicate a termination event) and *soft-preemption* (to indicate a downgrade event). This header is included in SIP BYE messages (used for sessions' termination) and SIP INVITE messages (used for sessions' re-negotiation).

C. Diameter Extensions

To deal with the charging issues of the solution, the Diameter stack was extended to carry the additional information needed for price differentiation and compensation credits calculation. Instead of defining new AVPs, we leveraged existing 3GPP/Diameter AVPs to carry this additional information. Table 7.3 summarizes these extensions.

Field name	**Encapsulating AVP**	**Charging mechanism & Message types employing info**	**Field description**
Service class	*Service-Parameter-Info(SPI)* AVP	Offline charging: ACR[start], [interim], [stop], and [event] Online charging: CCR[initial], [update], [termination], and [event]	The service class information extracted from the RP header of SIP messages is inserted in the SPI AVP such as following: SPI{[Service-Parameter-Type = service class]; [Service-Parameter-Value = gold]}.
Service duration	*Service-Parameter-Info(SPI)* AVP	Offline charging: ACR[event] Online charging: CCR[event]	For preemptive events, in addition to the service class, the SPI AVP also includes information about the service duration before the preemption, such as: SPI{[Service-Parameter-Type=service class]; [Service-Parameter-Value=gold]; {[Service-Parameter-Type=service duration]; [Service-Parameter-Value=180 sec]}.
Credit triggering event	E*vent-Type (ET)* AVP	Offline charging: ACR[event] Online charging: CCR[event]	Also, for preemptive events, information about the type of event triggering the credit allocation is included in the ET AVP, such as: ET=soft_preemption
Caller and callee's identities	*Called-Party-Address,* and *Calling-Party-Address* AVPs	Offline charging: ACR[event] Online charging: CCR[event]	To allow the calculation of the compensation credits, information about the caller and the callee's identities is inserted in the *Called-Party-Address,* and *Calling-Party-Address* AVPs (e.g. CPA=sip:alice@home.net).

Table 7.3: Diameter extensions

7.2.4 Prototype Setups and Test Scenarios

Three different prototype setups were used to respectively test: two-party scenarios without charging; multiparty scenarios without charging; and two-party scenarios with charging. Figure 7.13 illustrates the test bed used to test two-party and multiparty call differentiation scenarios without charging capabilities (the CTF being bypassed in the CSCF operation). It

consisted of eight clients using the services of the enhanced IMS simulated environment. The clients were running on laptops and the enhanced IMS simulated environment was installed on a regular PC. The machines (forming a WLAN) were equipped with IEEE 802.11g wireless cards, had Pentium 4 processors and 512 MB RAM, and ran Windows XP.

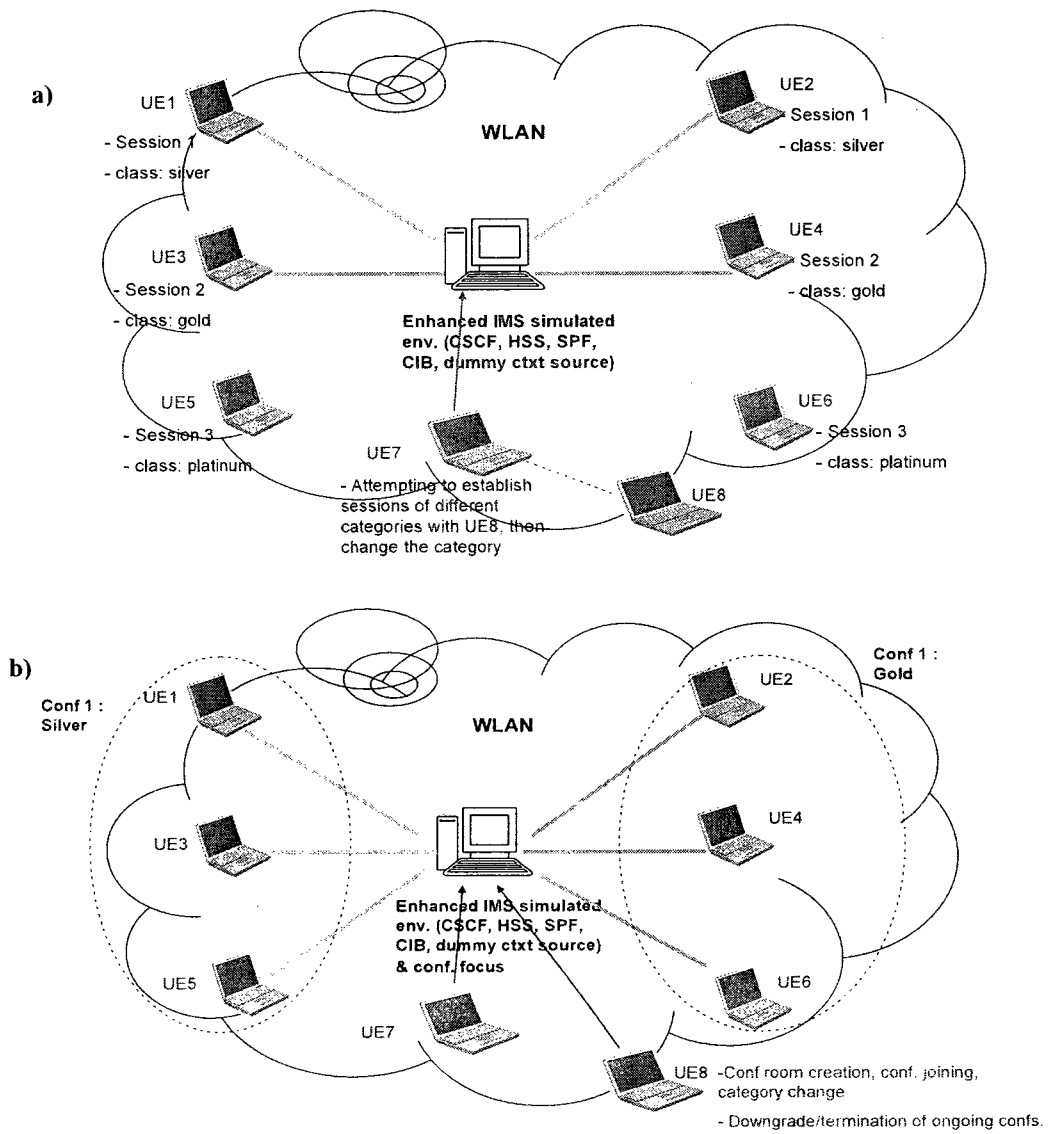


Figure 7.13: Prototype settings for call differentiation architecture without charging capabilities: a) Two-party scenarios test bed; b) Multi-party scenarios test bed

In order to test two-party scenarios, three pairs of clients were used to simulate an environment with three ongoing two-party calls with different categories, and the dummy

context source was configured to generate different contextual values, representing different loading conditions. Then, the remaining two clients were used to test the two-party session initiation and session category change procedures for the three categories of calls (under these conditions).

To test the multiparty case, the first six clients were instead used to simulate two ongoing conferences (of three participants each), of categories silver and gold. Then, different loading conditions were simulated and the remaining two clients were used to test the conference room creation procedure, the conference join procedure, and the conference category change procedure. They were also used to establish a new two-party call of class platinum, triggering the downgrade and then the termination of each of the ongoing conferences. The tests conducted using this prototype show that the system works well for each of these scenarios, therefore demonstrating the feasibility of the proposed call differentiation solution.

A similar setup was used to test charging related scenarios, as shown in figure 7.14. In this case, the 3G environment was simulated using six laptops and one PC forming a WLAN. The clients were running on the laptops, while the PC hosted the extended IMS simulated environment. Then the context source was configured to simulated high loading conditions, and these steps were followed to test the different charging scenarios: First, an offline-charged video session of type silver was established between UE1 and UE2 (only the CDF address/port were set in the users' profiles). Then, a second (audio) session of type gold was established between UE3 and UE4, triggering the downgrade of the first session. The charging mechanism used for UE3 and UE4 was online charging (i.e only the OCS address/port was set in the users' profiles). As a third step, UE1 terminated the first session.

Afterwards, a third offline-charged audio session of type platinum was established between the remaining two clients (UE5 and UE6). This third session triggered the termination of the session between UE3 and UE4. Finally, the third session was terminated by UE5. Following those steps, we found that the P-CSCF and S-CSCF related CDRs were successfully generated and updated for the first and third session. Moreover, the account balances for the pre-paid users (UE3 and UE4) were correctly updated for the second session and the compensation credit was allocated, thus demonstrating the feasibility of the multi-grade service charging solution proposed.

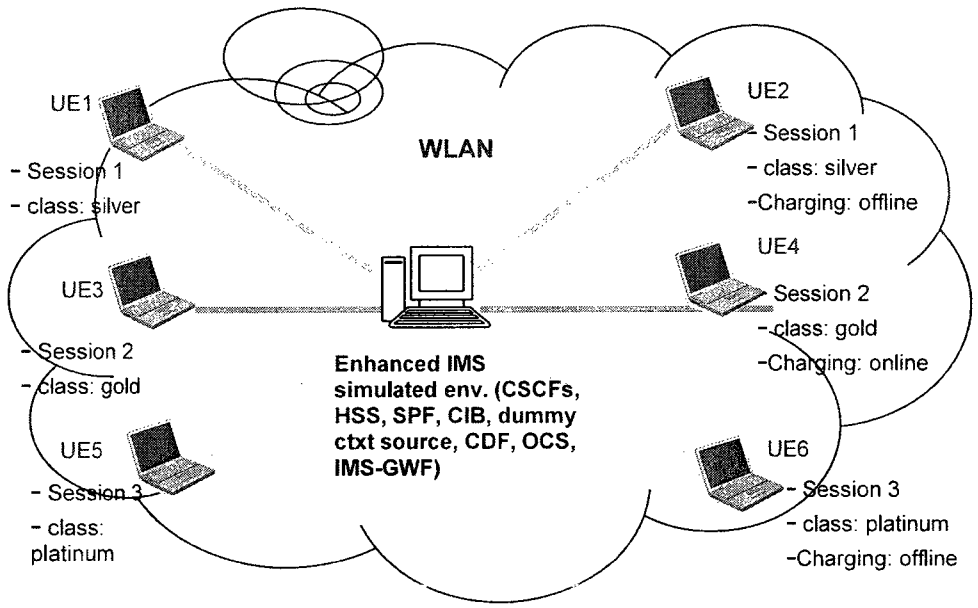


Figure 7.14: Prototype settings for call differentiation architecture with charging capabilities

7.3 Prototype for Enhanced Emergency Service Architectures

In this section, we present the proof-of-concept prototype we developed to validate our IMS enhanced emergency solution, which was presented in chapter 6.

7.3.1 Prototype Architecture

We extended the prototype presented in the previous section with emergency related components to demonstrate the feasibility of the IMS enhanced emergency solution

proposed. As shown in figure 7.15, two new components were added to the previous prototype, namely: a CRF responsible of context information retrieval from the CIB; and an E-CSCF module that was introduced to the CSCF to enable the routing of emergency calls to their correct destination (i.e. an appropriate PSAP). Furthermore, an existing dial-out servlet [109] was enhanced and added as part of the conference focus functionality to enable multi-party emergency sessions' scenarios.

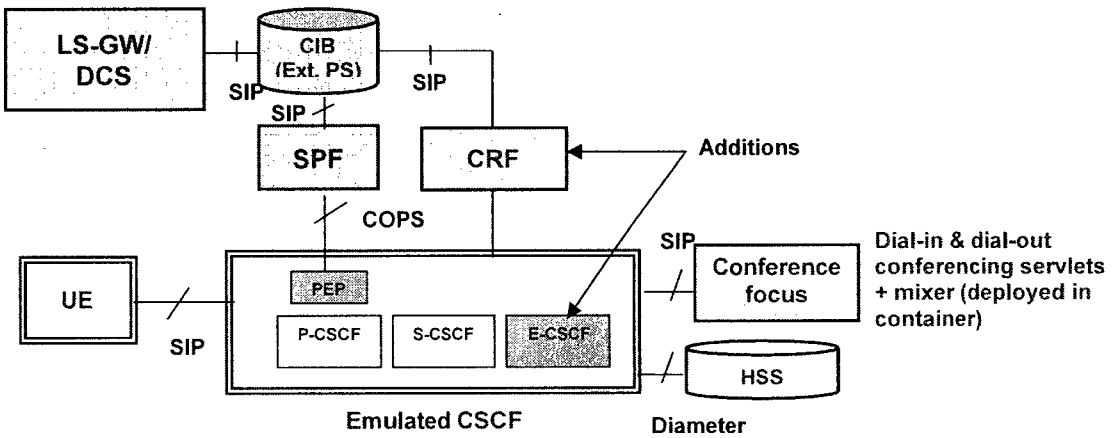


Figure 7.15: The IMS enhanced emergency service architecture prototype components

As depicted in figure 7.16, the CRF consisted of three sub-modules, namely: a Context Retrieval Module (CRM) used to respond to information queries from authorized emergency related entities (e.g. an E-CSCF or a PSAP); an Information Access Module (IAM) used for the interaction with the CIB for the collection of the needed information via subscriptions/notifications; and an Information Access Policies (IAP) repository that specifies which entities have access to which piece(s) of information related to the emergency caller.

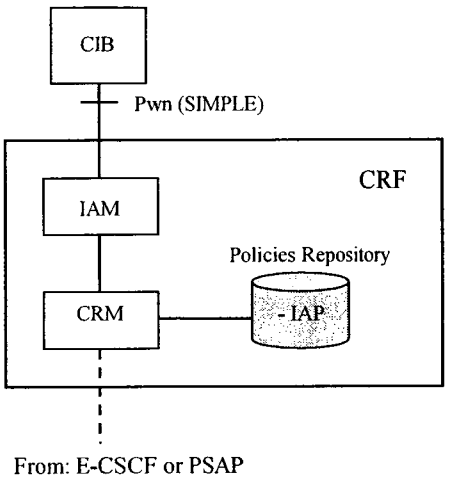


Figure 7.16: CRF software architecture

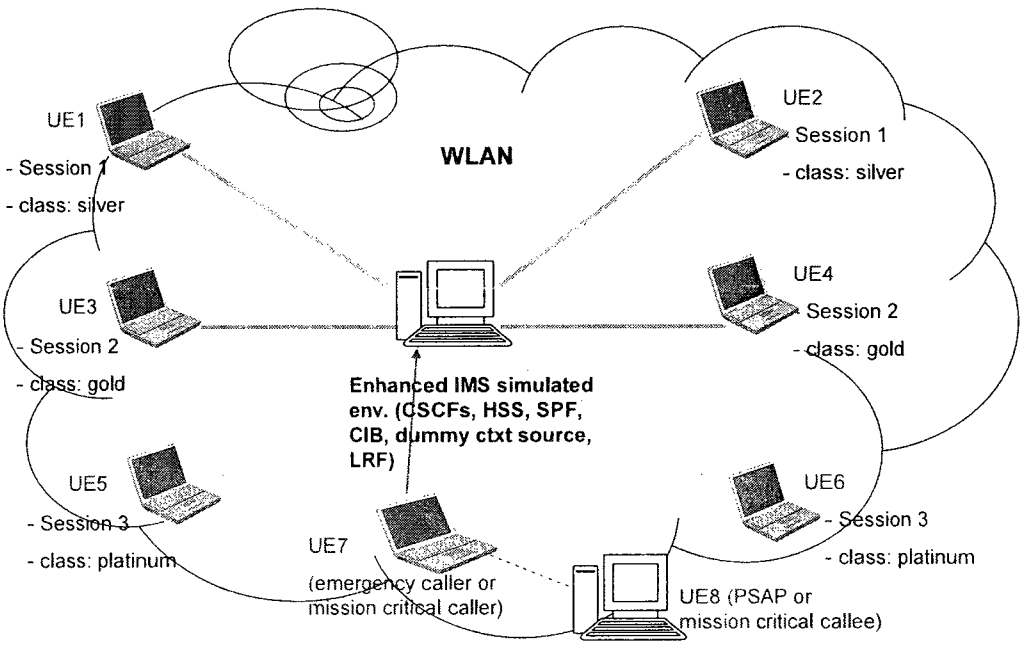
As for the E-CSCF functionality, it was achieved using a PSAP routing module that was introduced to the CSCF architecture. The role of this module is to route emergency calls to an appropriate PSAP, based on the contextual information it obtains from the CRF.

To enable the establishment of multi-party emergency sessions, an existing dial-out servlet [109] was added as part of the simplified conference focus described in section 7.2.2.3. This servlet, which initially enabled the initiation/termination of dial-out conferences, the addition/removal of users to/from conferences, the initiation/termination of sub-conferences and the moving of users between sub-conferences, was enhanced to enable the automatic moving of a user between relevant sub-conferences based on his/her location information obtained from the CIB.

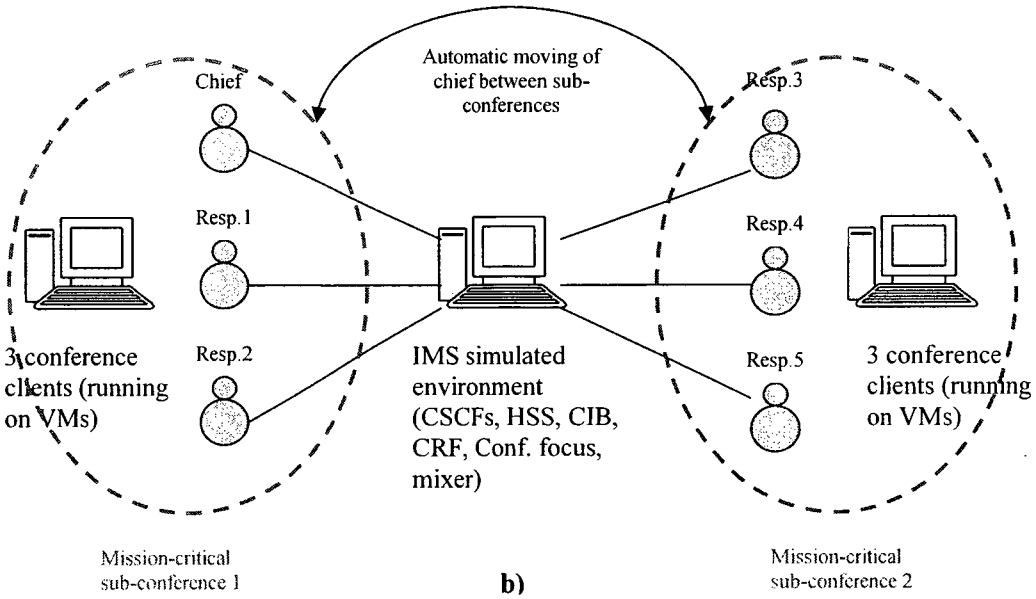
7.3.2 Prototype Setups and Test Scenarios

Three different prototype setups were used to test the QoS-enhanced, the conferencing-enhanced, and the context-aware emergency service architectures' operations. Figure 7.17a illustrates the test bed used to test the QoS-enhanced emergency service operation. It consisted of seven laptops and two PCs, forming a WLAN. The clients were running on

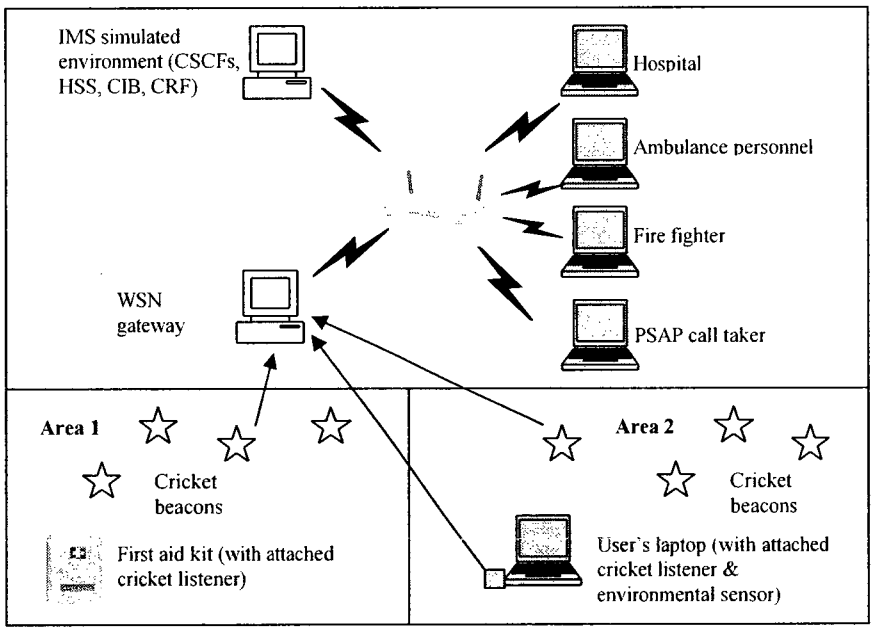
laptops, while one of the PCs represented the IMS simulated environment and the other PC represented a PSAP. For simplicity, all clients were stationary and their location information was pre-configured in the dummy context source. Three pairs of clients were used to establish three regular calls with different categories (i.e. silver, gold, and platinum), and the context source was configured to simulate different loading conditions. Then, the remaining client and PC (representing the PSAP) were used to test the different scenarios related to a 911 session establishment. Three scenarios were successfully tested, namely: session initiation without attempt to control ongoing sessions, session initiation after downgrade on an ongoing session, and session initiation after termination of an ongoing session. The same tests were successfully repeated with the client and PC representing mission critical users, therefore demonstrating the applicability of the solution proposed to different categories of emergency communications.



a)



b)



c)

Figure 7.17: Prototype settings for IMS enhanced emergency architectures: a) Test bed for QoS-enhanced emergency architecture; b) Test bed for conferencing-enhanced emergency architecture; c) Test bed for context-aware emergency architecture

A simple setup was used to test the conferencing-enhanced emergency service architecture, as shown in figure 7.17b. This setup consisted of three PCs (forming a LAN), one representing the IMS simulated environment (including the conference focus and the mixer) and the two other machines each hosting three conference clients (running on virtual

machines). The setup was used for the execution of the mission critical conferencing/sub-conferencing scenario, which was tested as follows: The six clients were registered as IMS users, then one of the clients acting as chief of operations interacted with the dial-out servlet (deployed in SDS) to initiate a dial-out conference between itself and the five other clients. The conference was then divided in two sub-conferences by the chief, and a GUI at the level of the CIB was used to simulate the chief's change of location. The information about this event was then accessed by the conference focus' dial-out servlet and used to switch the chief between the two sub-conferences, as if he moved between the two areas in which the sub-conferences are taking place. It should be noted that sensors and laptops were not used in this setup due to the lack of equipment at that time.

As shown in figure 7.17c, the context-aware emergency service architecture's test bed consisted of five laptops and two PCs, forming a WLAN, in addition to a set of MIT Cricket location sensors and an MTS300/Mica2 environmental sensor. The clients were running on laptops, while one of the PCs represented the IMS simulated environment and the other PC represented the sensor GW. Due to the lack of biometric sensors in our lab, we focused only on spatial and environmental information.

The interactions related to the enhanced emergency service scenario were tested as follows: first, the clients were used to register as IMS users, and then cricket beacons mounted to the ceiling were used to detect the location of the caller's laptop and of a first aid toolkit (each with an attached cricket listener). Furthermore, the MTS300/Mica2 sensor attached to the user's laptop was used to detect environmental conditions. All this information was conveyed to the WSN gateway, which publishes it (after proper formatting) in the CIB hosted by the IMS simulated environment. The published information was then accessed

locally by the CRF (also hosted by the IMS environment), and used for context-aware routing to the PSAP and the conveying of the needed information to the interested clients. All the interactions related to the enhanced emergency service scenario were carried successfully, therefore demonstrating the feasibility of the solution proposed.

7.4 Conclusions

In this chapter, we addressed the validation of our proposed solutions using proof-of-concept prototypes. Three prototypes were implemented, namely: a WSN/IMS integrated architecture prototype; an IMS call differentiation architecture prototype; and an enhanced emergency service architecture prototype. For each prototype, we presented the prototype architecture, discussed the design of the different components, along with the prototype's setup and the test scenarios. Some performance measurements were collected using the first prototype, while limitations in the two others (i.e. our inability to vary/control some of the circumstances under which the prototypes operated and the absence of sufficient test points within the IMS simulated environment) obliged us to resort to simulations to evaluate the performance of these solutions – as will be presented in the coming chapter. We also demonstrated the use of context awareness at the IMS service level as means to support the development of innovative value-added IMS services

Based on the experiments conducted using the developed prototypes, we found that the chosen implementation technologies work well for the different scenarios and that the main concepts proposed (i.e. WSN/IMS integration, signaling level call differentiation, and enhanced emergency operations) are feasible in an IMS environment. Moreover, we learned that the use of context as service enabler facilitates the development of context-aware value added IMS services.

Chapter 8

Simulation Results

In this chapter, we present the OPNET-based simulations used to evaluate the performance of our call differentiation architecture and QoS-enhanced IMS emergency service architecture. We start by describing the simulation environment and the network setup used, then present the design of the simulation models developed, along with the simulation scenarios and the performance measurements and analysis.

8.1 Simulation Environment and Network Setup

The simulations of our solutions were carried using the OPNET modeler v.11.5.A simulation tool, which provides a comprehensive environment for the modeling and performance evaluation of communication networks and distributed systems [111]. In order to simulate our targeted architectures, we reused and extended the SIP-IMS contributed model [112] in addition two OPNET standard libraries, namely: the applications and the Ethernet libraries. Figure 8.1 shows the network setup used to simulate the base scenario (i.e. the scenario involving basic two-party call setup without call differentiation), which was used as baseline for the calculation of the overhead introduced by call differentiation.

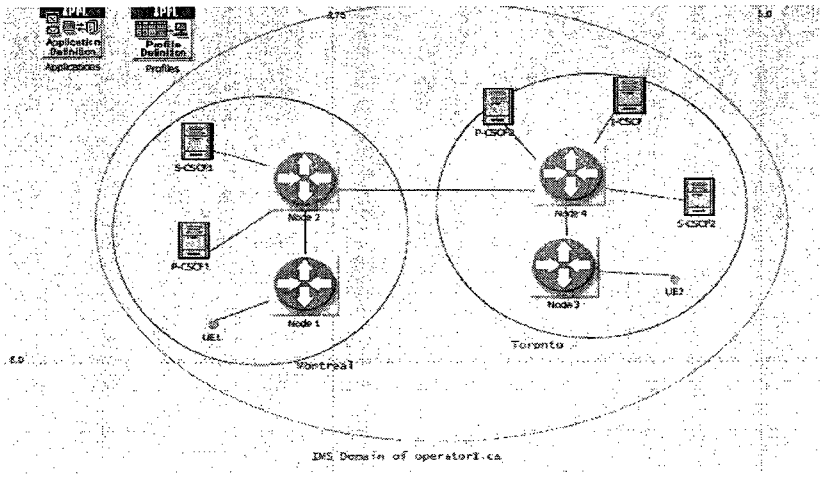


Figure 8.1: OPNET network setup for basic 2-party call scenario without call differentiation

As shown in the figure, three types of components distributed over two geographical areas (Montreal and Toronto) are used in the base scenario, namely: IMS UEs, IMS CSCFs (including P-CSCFs, S-CSCFs, and I-CSCFs), and transport level routers. The UEs are realized as Ethernet workstations, while CSCFs are modeled as Ethernet servers. Furthermore, the multimedia user profile is selected as the application profile supported by the different UEs, to enable the establishment of VoIP sessions between them.

To accommodate call differentiation related scenarios, additional types of nodes were developed and enhancements were made to some of the existing components. In the coming section, we detail the nodes and process models related to these new and enhanced components and present the different protocol stacks employed as well as the application profiles used for traffic generation.

8.2 OPNET Models Design

8.2.1 Node and Process Models

To build the simulation models of our targeted architectures, five types of nodes were used, namely: CSCFs (including P-CSCFs, S-CSCFs, I-CSCFs, and E-CSCFs), a SPF, a CIB, UEs, and transport level routers. Among these components, the SPF and the CIB are new components developed from scratch, while the E-CSCF and the UE are enhancements of existing standard OPNET components.

UEs are realized using the *Ethernet-Wkstn-adv* node model implementing the *gna-clsvr-mgr* application process on top of the TPAL (Transport Adaptation Layer) module. This application process then spawns a *SIP_UAC_mgr* process that spawns on its turn one or several *SIP_UAC_CallDiff* processes handling call differentiated SIP sessions. Furthermore, depending if the UE is the calling or the called party, a *gna-voice-calling-mgr*

process or a *gna-voice-called-mgr* process are respectively spawned to handle the RTP traffic associated with the session. Similarly, CSCFs are modeled using the *Ethernet-Server-adv* node model implementing the *gna-clsvr-mgr* application process, which in this case spawns a *SIP_UAS_mgr* process initiating one or several *SIP_UAS_CallDiff* processes. In the case of S-CSCFs and E-CSCFs, each *SIP_UAS_CallDiff* process also starts a *COPS_PEP* process that is responsible of communicating with the SPF for the admission of calls. Moreover, in the case of the E-CSCF, a *P-UAC* process is also used to obtain the needed contextual information from the CIB. This last is realized using the *Ethernet-Server-adv* node model, also implementing the *gna-clsvr-mgr* process model that starts a *P-UAS* process via each *SIP-UAS-callDiff* process. As for the SPF, it is modeled using the *Ethernet-Server-adv* node model, implementing the *PDP_mgr* process, which spawns several *PDP* processes (re)allocating resources to the different calls. A summary of the different nodes used and their associated process models is shown in table 8.1. These process models will be described with more details in the coming sub-sections.

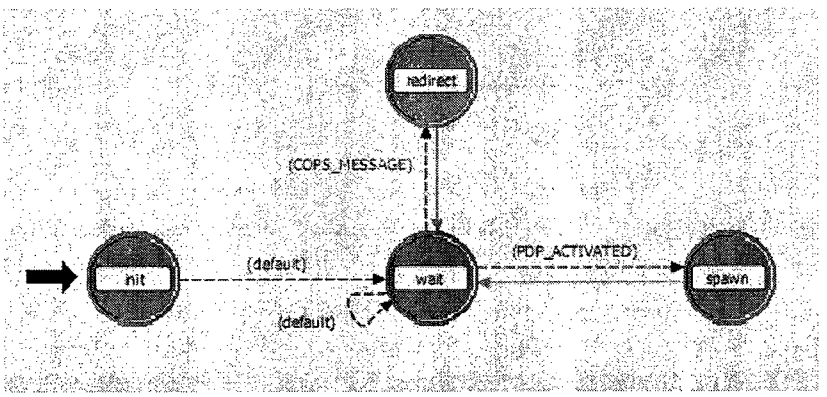
Node name	**Node model**	**Application layer process model**	**Child processes spawned**
UE	*Ethernet-Wkstn-adv*	*gna-clsvr-mgr*	For calling party: *(1)gna-profile-mgr →(1)gna-voice-calling-mgr* *(1)SIP-UAC-mgr → (1..n) SIP-UAC-callDiff* For called party: *(1)gna-voice-called-mgr* *(1)SIP-UAC-mgr →(1..n)SIP-UAC-callDiff*
P-CSCF & I-CSCF	*Ethernet-Server-adv*	*gna-clsvr-mgr*	*(1)SIP-UAS-mgr →(1..n)SIP-UAS-callDiff*
S-CSCF	*Ethernet-Server-adv*	*gna-clsvr-mgr*	*(1)SIP-UAS-mgr →(1..n)SIP-UAS-callDiff- → each (1)COPS-PEP*
E-CSCF	*Ethernet-Server-adv*	*gna-clsvr-mgr*	*(1)SIP-UAS-mgr →(1..n)SIP-UAS-callDiff- → each (1)P-UAC & (1)COPS-PEP*
SPF	*Ethernet-Server-adv*	*PDP-mgr*	*(1..n)PDP*
CIB	*Ethernet-Server-adv*	*gna-clsvr-mgr*	*(1)SIP-UAS-mgr →(1..n)SIP-UAS-callDiff- → each (1)P-UAS*

Table 8.1: OPNET simulation nodes and their associated process models

A. COPS-PDP-mgr and COPS-PDP process models

Figure 8.2 depicts the state diagram of the COPS-PDP-mgr process model and its associated attributes. Among these attributes, we mention the 'Policy Service' attribute that must be set to 'enabled' for the node to act as a policy server, and the 'Processing Delay' as the time it takes the server to process a COPS message. As shown in the state diagram, the COPS-PDP-mgr process waits for the reception of a PDP-activated interrupt upon which it starts a new PDP process and redirects to it any pending messages. Furthermore, upon the receipt of a COPS-message related interrupt, the COPS-PDP-mgr redirects the request to the appropriate PDP process or puts it in a queue if the PDP is not available.

a)



b)

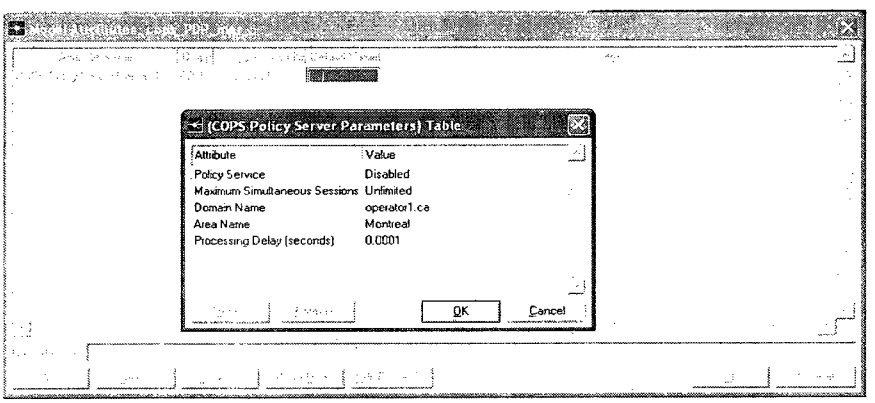


Figure 8.2: The COPS-PDP-mgr process model: a) The COPS-PDP-mgr state diagram; b) The COPS-PDP-mgr model attributes

Figure 8.3 shows the COPS-PDP process state diagram, which operates as follows: After doing some initialization, getting the parent-to-child memory, and opening a passive

connection, the process blocks until it receives a valid interrupt. Once it receives it, it determines its type/code and takes the appropriate handling action. For instance, if an open-IND interrupt is received, the PDP informs the PDP-mgr to spawn a new passive PDP in listening mode, and in the case of a COPS-MSG (message from a PEP) or a COPS-TRIGGER (trigger from another PDP to modify a previous decision), the interrupts are enqueued for further processing.

After that stage, the process checks if there are any pending COPS messages or inter-process triggers in the list and sets the COPS-REQ, COPS-RPT, COPS-DRQ, and the TRIGGER flags accordingly. Based on the value of these flags and the type of the packet that is being processed, the process passes to the appropriate state. For instance, for COPS REQ messages, the process passes to the 'New_dec' state in which it checks if the client handle exists (if yes rejecting the request and if not creating a new service record), and sets the admission status to pending, which takes the process to the 'adm-control' state. From that state, the process spawns a P-UAC to obtain the needed contextual information from the CIB, then determines if resources need to be freed (based on the call admission algorithm) before admitting the call. If resources need to be freed, the process passes to the 'freeing-res' state in which it sends an interrupt to another PDP process and waits until it receives a trigger response, at which point it sent the RES-FREED and the RES-UNAVAILABLE flags to their appropriate values before returning the to previous state. For COPS-TRIGGER messages, the process goes to the 'Modify-dec' state in which it gets the trigger type (i.e. downgrade or termination), checks if the session exists, sends a new DEC message to the involved PEP to carry the necessary actions, and goes to the 'wait' state to wait for the result. Based on this result the process sets the TRIGGER-OK flag to

an appropriate value and sends an inter-process interrupt with the operation result, to the process that had initiating the preemption trigger.

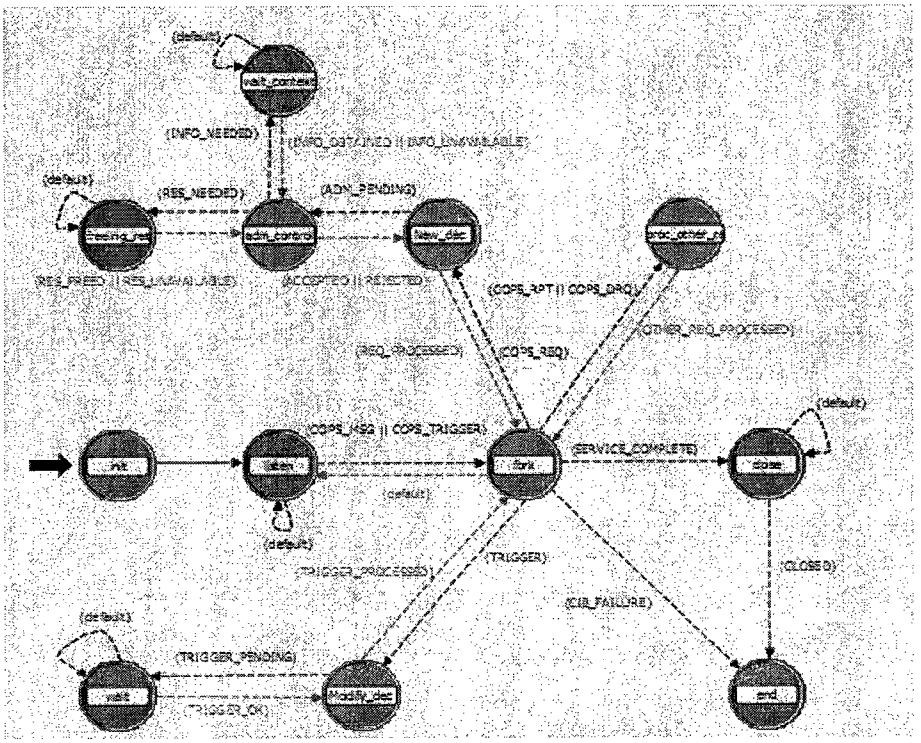


Figure 8.3: The COPS-PDP state diagram

B. COPS-PEP process model

Figure 8.4 shows the COPS-PEP process state diagram, which operates as follows: After doing some initialization, getting the call related information from the SIP-UAS that launched it, and initializing the address of the PDP to contact, the process passes the 'open' state in which it opens a connection with the PDP and sets the *estab* flag to the appropriate value. If the value of that flag is set to true, the process then goes to the 'listen' state in which it waits for the reception of valid interrupts, based on which it going to one of the following states: the 'send-REQ' state from which it creates a session record, builds a COPS REQ message based on the call info and sends it to the PDP; the 'proc-DEC' state during which COPS decisions received from the PDP are processed; the 'send-RPT' state responsible of the construction and sending of COPS RPT messages based on feedback

from the parent SIP-UAS process; and the 'send-DRQ' state used for the deletion of the session record and the sending of a COPS DRQ message to the PDP. From that last state, the process then passes to the 'close' state in which the connection with the PDP is closed, then the 'end' state in which the process is terminated and resources are freed.

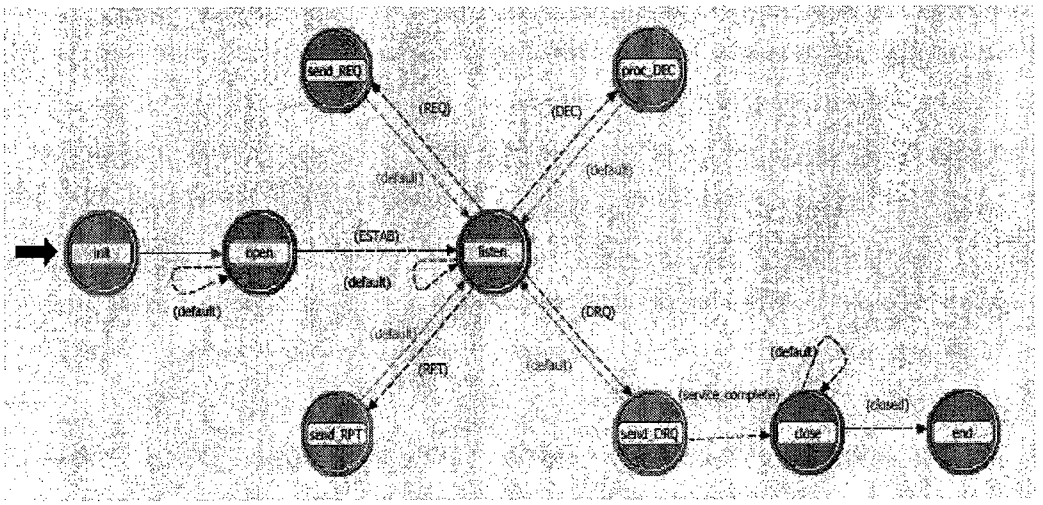


Figure 8.4: The COPS-PEP state diagram

C. P-UAS and P-UAC process models

To enable the exchange of contextual information, we defined two simple processes, namely: A presence user agent server (P-UAS) related to the operation of the CIB; and a presence user agent client (P-UAC) used by information consumers. Figure 8.5a details the P-UAC process, which operates as follows: After doing some initializations and opening a connection with the CIB, the process blocks until it receives a valid interrupt based on which it passes to one of two states – 'send-sub' for sending SIP subscription messages to get the needed information and 'proc-notify' for processing of the received notification messages. The 'close' and 'end' states are used for closing the connection to the CIB and the termination of the process.

As shown in figure 8.5b, the P-UAS process has a similar operation, except that the 'listen' state is also used for monitoring of information freshness. Then, there is a 'fork' state used

to fork the interrupt to the appropriate processing phase, including: 'proc-pub' and 'proc-sub' for the processing of information publication and subscription messages; and 'trigger-pub' for the triggering of information publication. Once again, the 'close' and 'end' states are used for process termination and resources freeing.

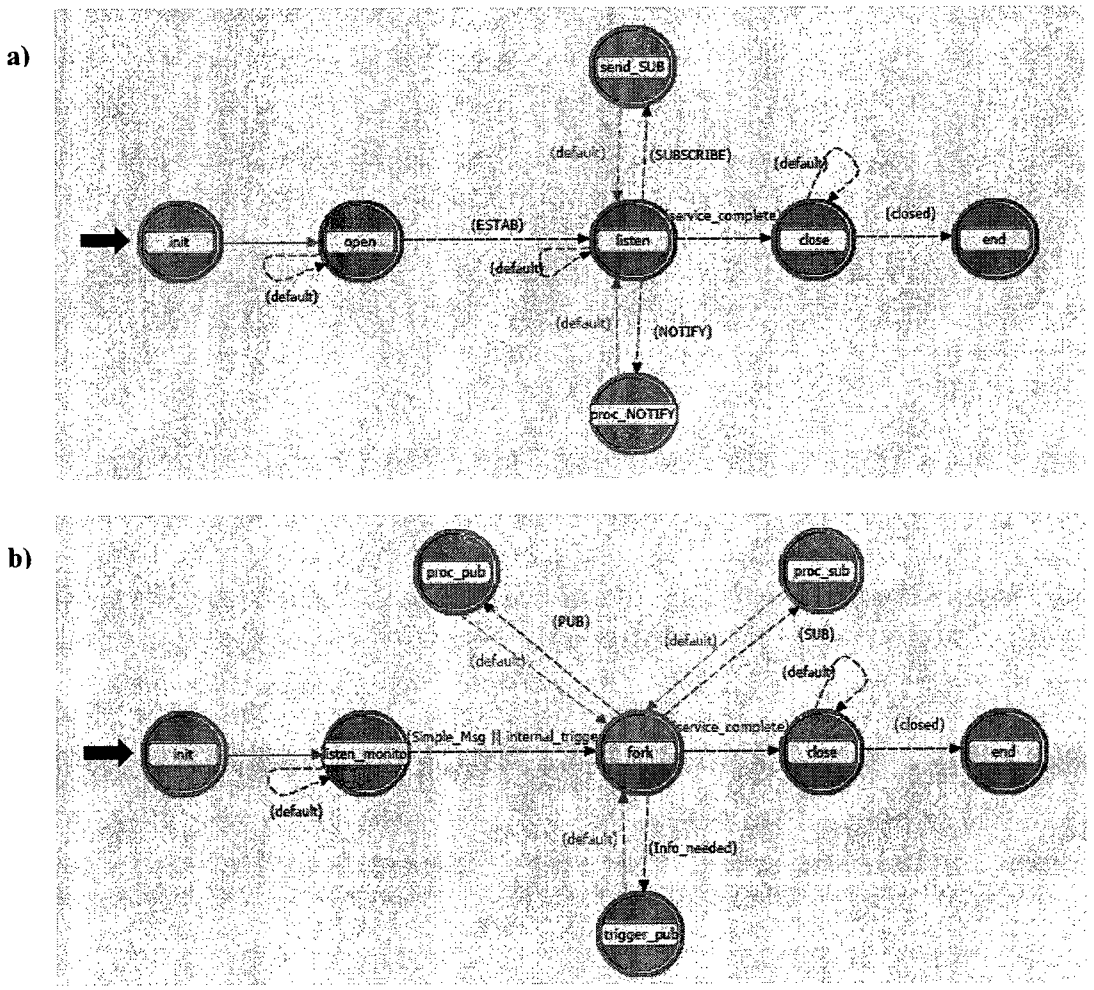


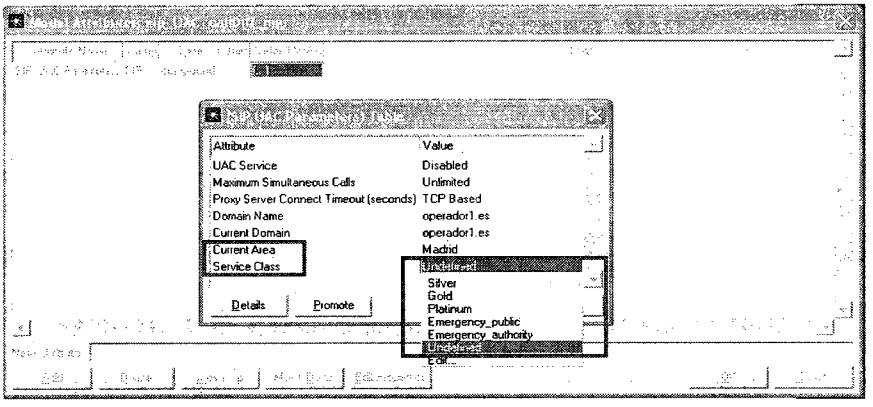
Figure 8.5: The information exchange related processes: a) The P-UAC state diagram; b) The P-UAS state diagram

D. Modified SIP-UAC and SIP-UAS process models

To achieve call differentiation, the original SIP-UAC-mgr and SIP-UAC processes designs were modified as follows: the service-class was added as a SIP-UAC-mgr attribute that can be specified by the user, and then parsed and passed as part of the parent-to-child memory between the SIP-UAC-mgr and its SIP-UACs; and the SIP-UAC logic was modified to get

the value of the service-class attribute and use it to set the appropriate field in outgoing SIP messages in addition to reacting to referral messages (related to the downgrade/termination of an ongoing session) and sending a notification with the result of the referral. Figure 8.6a depicts the service class attribute that was added to the model's design, while figure 8.6b highlights the modified states in the SIP-UAC state diagram. As shown, the modifications were made in the 'req-proc' and 'resp-proc' states as follows: The 'req-proc' state was augmented with the logic needed for the extraction of the service class parameter (from the parent to child memory that is created by the UAC-mgr) and the labeling of the SIP INVITE message to be sent to the UAS with it; while the 'resp-proc' state was enhanced with the logic related to handling of the SIP referral messages related to the downgrade/termination of the session.

a)



b)

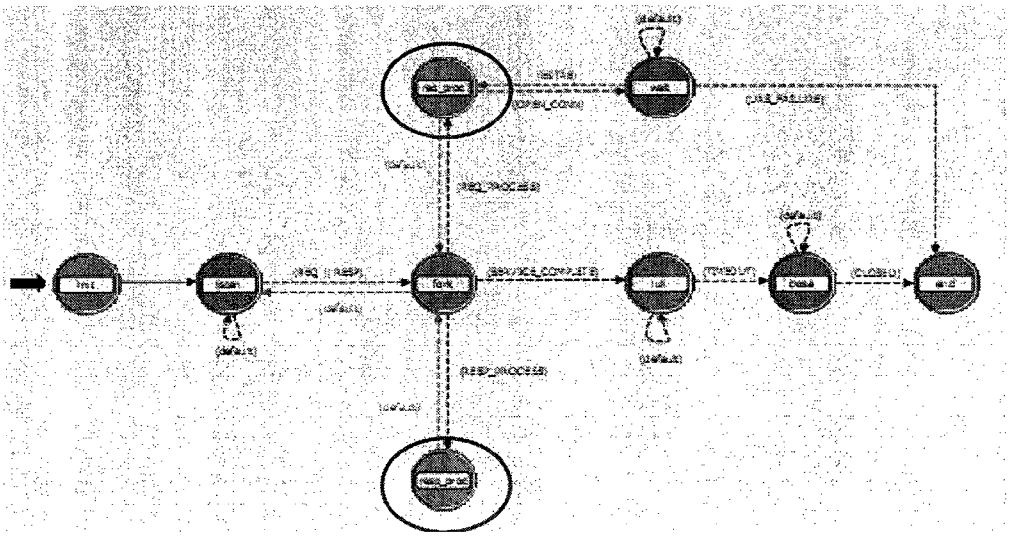
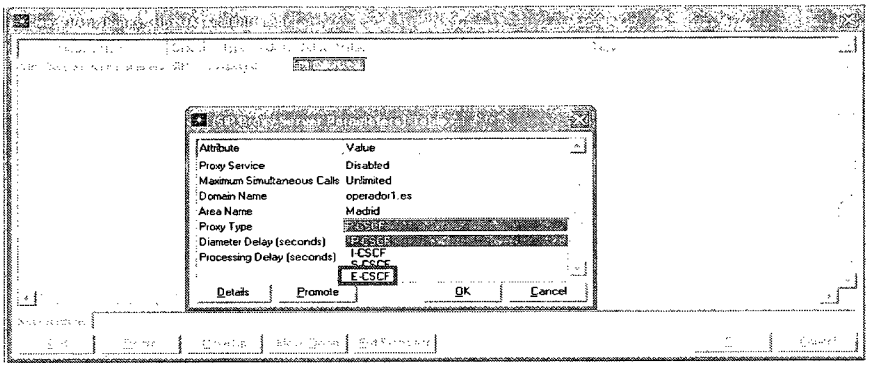


Figure 8.6: The modified SIP-UAC-mgr and SIP-UAC designs: a) The added service class attribute; b) The modified states in the SIP-UAC FSM

As for the SIP-UAS-mgr and SIP-UAS processes designs, they were modified as follows: The proxy type parameter (one of the SIP-UAS-mgr attributes) was enhanced with a new value (i.e. the E-CSCF) to accommodate emergency related scenarios. The SIP-UAS logic used by S/E-CSCFs was modified to extract the value of the service class attribute from SIP INVITE messages, then spawning a PEP process that is responsible of the communication with the SPF for call admission decisions. Furthermore, in the case of preemption decisions received from SPF, the PEP instance is responsible of enforcing those decisions by sending referral messages and processing their related notifications. Finally, additional logic is added to enable the proper handling of emergency calls by E-CSCFs. Figure 8.7a shows the new value that was defined for the proxy type attribute, while figure 8.7b shows the modified states in the SIP-UAS state diagram.

As shown, the modifications were made in the 'req-proc', the 'ack-proc', and the 'listen' states as follows: The 'req-proc' state was augmented with the logic needed for the handling of additional types of messages related to call differentiation (e.g. re-invite, notify-referral, reinvite-ack, and spf-dec). Furthermore, the handling of INVITE messages (by S-CSCFs and E-CSCFs) was modified so that the service-class information is extracted from the packet fields, a PEP is spawned and the call information is used for the construction of a COPS REQ message that is sent to the SPF for call admission decision, after which the call is established normally or rejected. As for the 'ack-proc' state, it was enhanced with the logic related to handling of refer-ok response messages. Finally, the 'listen' state was augmented with new logic related to E-CSCFs, such that for emergency calls, the user's location information is obtained from the CIB and used to determine the most appropriate PSAP as the next routing hop.

a)



b)

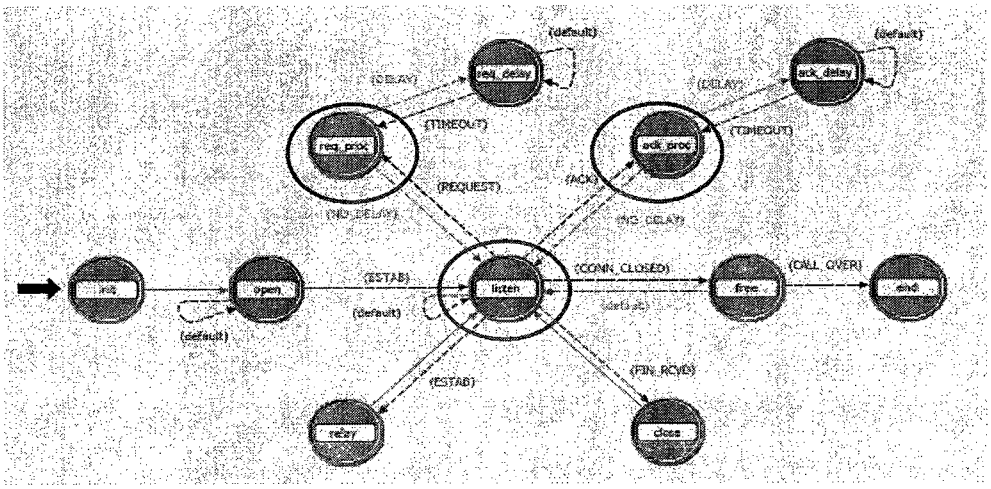


Figure 8.7: The modified SIP-UAS-mgr and SIP-UAS designs: a) The new value of the proxy type attribute; b) The modified states in the SIP-UAS FSM

8.2.2 Protocol Stacks

Two protocol stacks were used in the carried simulations: an extended version of the SIP stack provided by the SIP-IMS model; and a new COPS stack that was developed from scratch. Each of these stacks is realized using three files: an 'api' header file defining the codes of the interrupts (i.e. the events) used in the communication between an application and the stack as well as the basic functions related to the protocol operation; a 'support' header file defining additional constants, data structures, and utility functions needed for the handling of the protocol messages; and a C file implementing the different functions defined in the two header files. Figure 8.8 shows an excerpt from the 'Cops_api.h' header file with the interrupt codes and basic functions definitions, as an example.


```
/* cops_api.h: Definitions and declarations for the COPS model */

#ifndef _cops_api_h_
#define _cops_api_h_

/***** Include files *****/
#include <opnet.h>
#include <ots_qft.h>
#include <cops_support.h>

#if defined (__cplusplus)
extern "C" {
#endif

/***** Symbolic Constants and Macros ****/

/** COPS Operations Codes **/

/* COPS communicates with the Application Process that is using its services */
/* by using Process Interrupts (i.e. intrpt_type = OPC_INTRPT_PROCESS) and  */
/* one of the following interrupt codes:                                    */
#define  COPS_REQ                          500  /* Used by PEP to request a policy-based decisio
#define  COPS_DEC                          501  /* Used by PDP to return a decision to PEP.  */
#define  COPS_RPT                          502  /* Used by PEP to return a report to PDP, indica
#define  COPS_DRQ                           503  /* Used by PEP to indicate to PDP that the stat
#define  COPS_SSQ                          504   /* Used by PDP to ask the PEP to resend its st
#define  COPS_OPN                          505  /* Used by the PEP to specify to PDP the client t
#define  COPS_CAT                          506   /* Used to positively respond to OPEN messages
#define  COPS_CC                           507   /* Issued by either PEP or PDP to notify the c
#define  COPS_KA                           508  /* Keep alive messages sent from PEP to PDP and
#define  COPS_SSC                          509  /* Sent by PEP after the PDP sends a SSQ and the

/* PDP listening port - the default specified by RFC 2748 */
/* is 3288 - but since OPNET TCP model allows automatic   */
/* port number only up to 1024 at this time, this COPS model*/
/* version will use 328. Note that this number makes no   */
/* difference in the simulation performance/results       */

#define COPS_PDP_LISTENING_PORT            328
#define COPS_PEP_LISTENING_PORT            329

/* Type definition */
#define  COPS_Session_Handle                    COPS_Session_Info_Shell*

/*** Function Declarations ***/

/* This function starts the Policy Server service on a node. It can be called only once per node.  */
/* only one Policy Server process can be running on a given physical node at any given time.

Compcode              cops_policy_server_start (void);


/* This function starts the COPS PEP manager service on a node. It can be called only once per nod
/* only one COPS PEP manager process can be running on a given physical node at any given time.

Prohandle             cops_pep_mgr_start (void);


/* TRAL sends Remote Interupts to the Application Module to inform of TCP/UDP connection */
/* status. These interupts are first received by the root process on this module.        */
/* The following two functions make it possible for the root process to determine if an  */
/* interrupt was destined for COPS and if so, enable steering of the interrupt to the    */
/* appropriate COPS process.

Boolean               cops_intrpt_is_for_cops          (Ici* intrpt_ici_ptr);
void                  cops_redirect_intrpt_to_cops     (Ici* intrpt_ici_ptr);
Boolean               cops_session_valid               (COPS_Session_Info_Shell* session_info_shell_

/* The COPS PEP and Policy Server Parameters attributes have a 'Service' subattribute     */
/* which the user can set to enabled/disabled to choose if a he wants the COPS service   */
/* to be available or unavailable on a particular node. Following functions can be       */
/* used to read in the setting of this <Service> subattribute. Note that this attribute  */
/* does not automatically switch the SIP service on/off. To provide this feature the     */
/* application code has to be written such that it checks for the setting of this attr   */
/* before starting the COPS service                                                      */
Boolean               cops_policy_service_attr_enabled (Objid cops_module_objid);
Boolean               cops_pep_service_attr_enabled (Objid cops_module_objid);
```

Figure 8.8: Excerpt from the OPNET COPS stack implementation

It should be noted that the messages exchanged between different nodes are carried in OPNET packets. The COPS packet consists of three fields: a packet type 'type' indicating whether the message is solicited (1) or unsolicited (0); a message type 'msg' indicating the type of COPS message associated with the packet (e.g. REQ, DEC, RPT...etc); and a 'msg-info' field carrying the different objects related to the message in the form of a data structure. Similarly, the SIP packet consists of a 'type' field indicating whether the message is a regular or a network-initiated request or response, in addition to a 'msg' field specifying the type of SIP message associated with the packet (e.g. Invite, Bye, Refer ...etc) and a 'call_info' field containing the different headers related to the message. An

example of a packet carrying a COPS REQ message is shown in figure 8.9a, while figure 8.9b depicts an example of a packet carrying a SIP INVITE message.

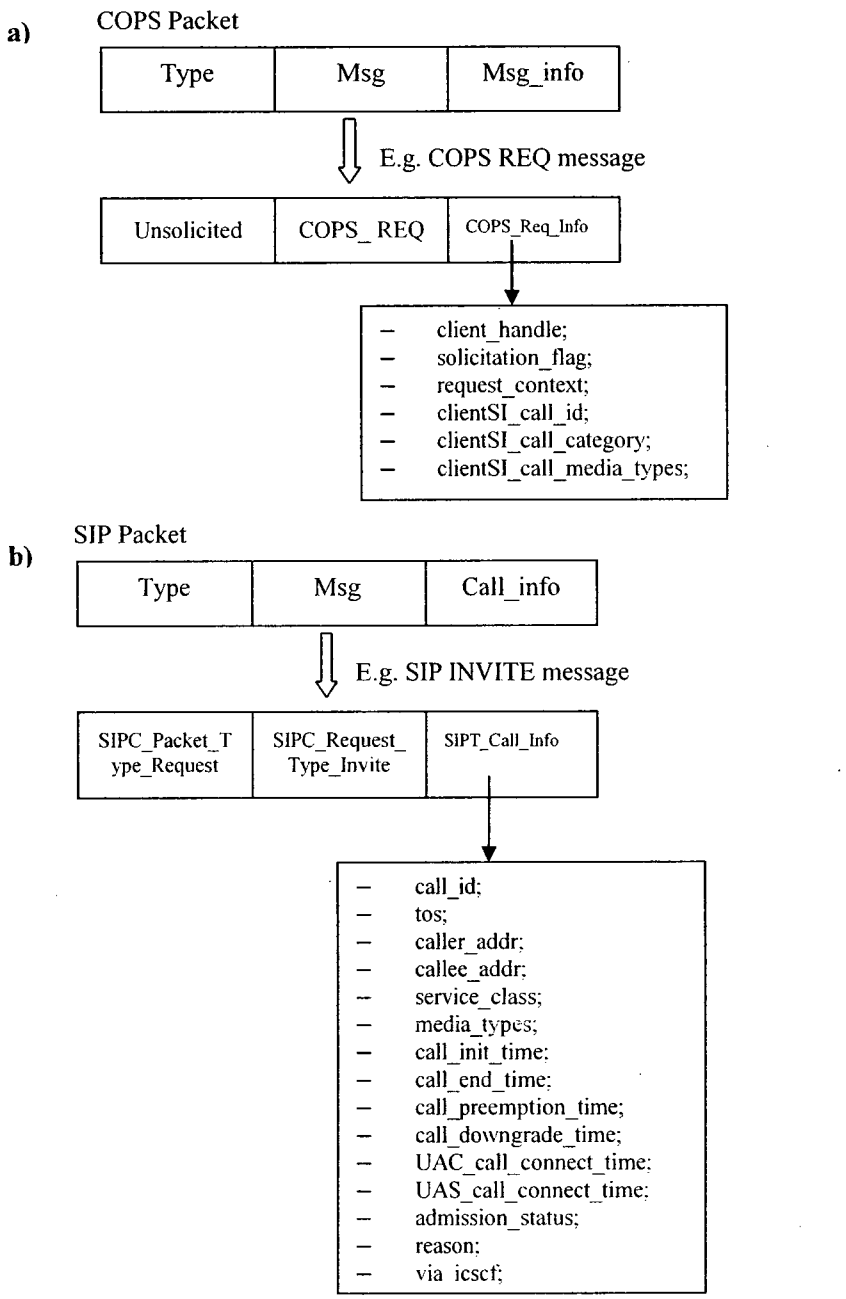


Figure 8.9: OPNET packets formats: a) COPS packet format; b) SIP packet format

8.2.3 Applications and Profiles Definition

In order to simulate different scenarios with different traffic generation patterns, we employed the application and profile definition nodes provided by OPNET. The application

definition node enables the definitions of various applications and voice encoding schemes. To accommodate IMS-based scenarios, we added the AMR codec definition to the list of available encoding schemes as shown in figure 8.10. Furthermore, we defined a new IMS voice call application, whose attributes are depicted in figure 8.11. One important attribute is the 'Encoder Scheme' that we set to the AMR_12.20 codec for both incoming and outgoing traffic. Another important parameter is 'Signaling' for which we specified SIP as the signaling protocol and chose a mix of signaling and data as traffic pattern- noting that the traffic could be restricted to signaling messages only if needed. We should mention that this new application was used in conjunction with the pre-defined video conferencing (heavy) application in some of the scenarios involving voice and video traffic.

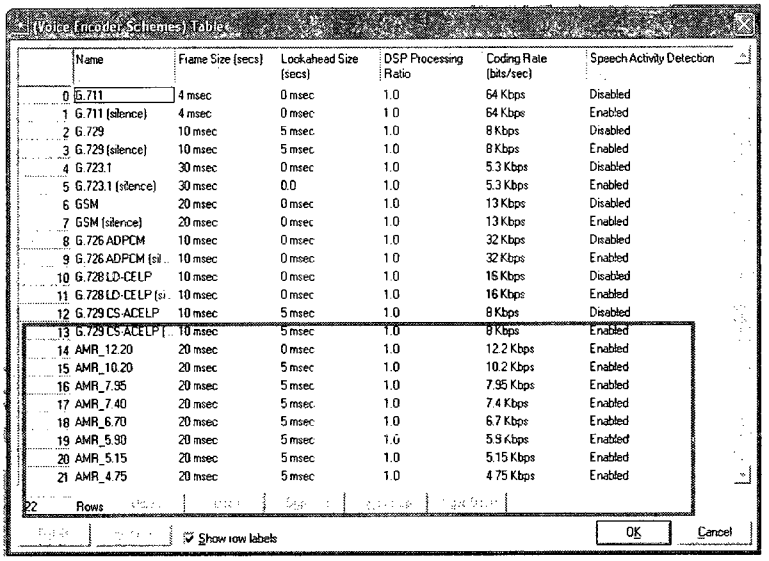


Figure 8.10: AMR voice codec definition

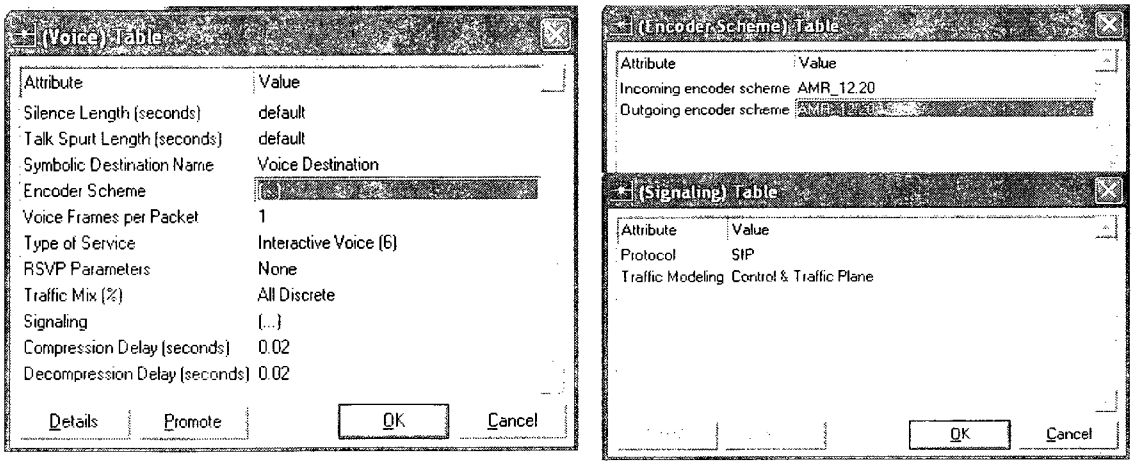


Figure 8.11: IMS voice call application definition

In addition to the applications definitions (which basically determine the different traffic patterns that could be used), application profiles were also defined to specify the set of applications that can be used by workstations, along with their start and end times and their repeatability. Such profiles help us simulate different scenarios with different application sessions, starting at interleaved or concurrent times.

While configuring the application profiles, we took into consideration our goals in terms of the performance measurements that we are interested in collecting. Those goals are summarized as follows: 1) Calculating the overhead introduced by the different call differentiation operations/scenarios in terms of call setup time and network load in order to assess the system's general performance; 2) Studying the variation of the different service differentiation metrics used with respect to different loading conditions in order to show the effect of call differentiation on the network resources' utilization.

To achieve the first goal, we defined scenarios in which one call is associated with each UE, and the start/end times of the different calls are set in an overlapped manner in order to test the different scenarios related to call differentiation (i.e. call establishment with no attempt to control ongoing call, after downgrade on an going call, and after termination of an ongoing call). Figure 8.12 shows the profiles defined for one of the scenarios involving six UEs and three voice calls (i.e. one call per pair of UEs).

As shown in the figure, the three application sessions are configured to respectively begin after 100, 150, and 200 seconds after the start of the simulation, and last until the end of the simulation run. Furthermore, the three UEs designated for initiating the three voice sessions (i.e. UE1, UE3, and UE5) are each configured with the appropriate application profile, by adding this profile to the workstation's supported profiles list, and the address of the UE to

which each session is destined is configured in the workstation's destination preferences. For instance, in the mentioned scenario, UE1's workstation is configured with the application profile 'Multimedia User_AudioCall_Silver' as part of its supported profiles, and with UE2's address as its voice destination preference, as shown in figure 8.13.

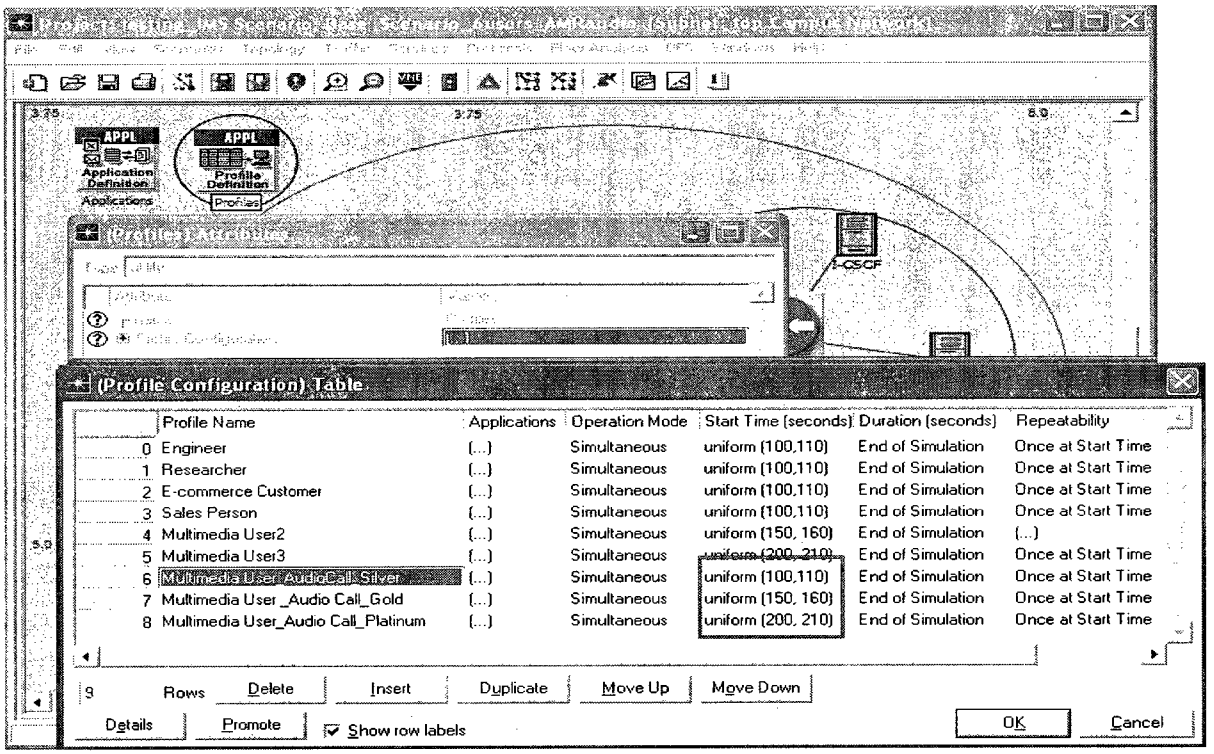


Figure 8.12: Application profiles definition for three overlapped voice sessions scenario

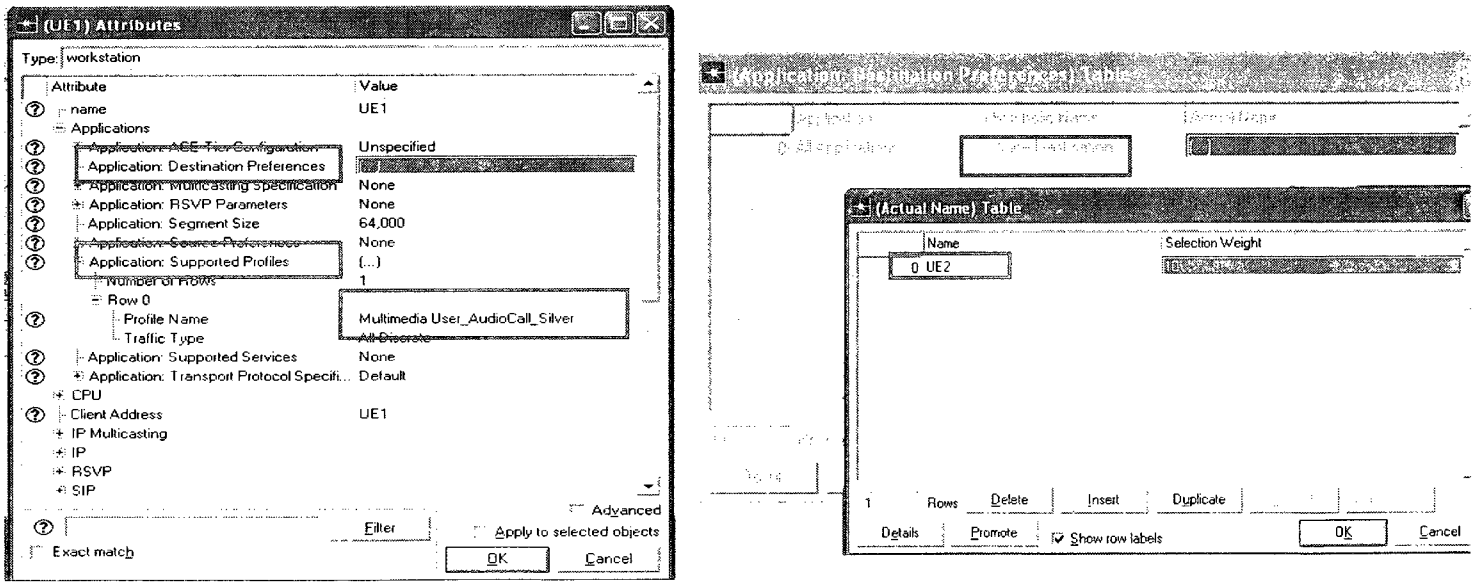


Figure 8.13: Workstation configuration with supported application profile and destination preferences

In order to measure the values of the different service differentiation metrics with respect to various network loads, we defined an additional application profile in which UEs act as call generators. To achieve this, the application profile was configured to establish one call at the beginning, and then add a call every two seconds, as shown in figure 8.14. We note that the profile start time is set to 60 seconds and the start time offset is set to 10 seconds, thus leading to traffic generation after 70 seconds from the start of the simulation. Furthermore, the repeatability attribute is set to unlimited with an inter-repetition time of 2 seconds, to achieve the incrementally increasing network load required.

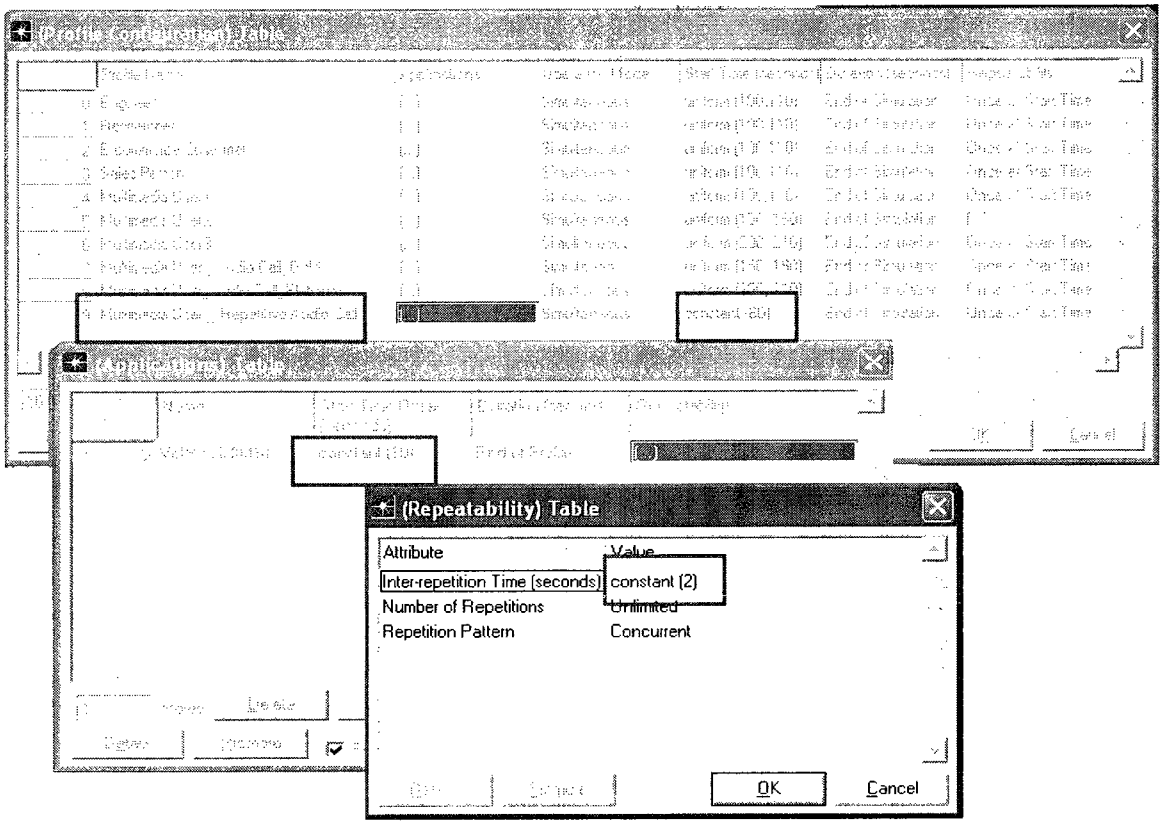


Figure 8.14: Repetitive audio call application profile definition

8.3 Simulation Scenarios

Figure 8.15 depicts the scenario we used to test the basic performance of the call differentiation solution, focusing on the case of regular calls. As shown, the scenario

involves three calls established between Six UEs (i.e. one call between each pair of UEs) as follows: First, a video session of type Silver is established between UE1 and UE2; then an audio session of type gold is established between UE3 and UE4, triggering the downgrade of the first session; the first session is then terminated; and finally, an audio session of type platinum is established between UE5 and UE6 triggering the termination of the second session. The scenario ends with the termination of the third session. It should be noted that the three sessions are defined in an overlapped manner, in order to enable the testing of various operations, such as call establishment after the downgrade/termination of an ongoing session. To force such an operation, we pre-configured the CIB to disseminate contextual information simulating high loading conditions.

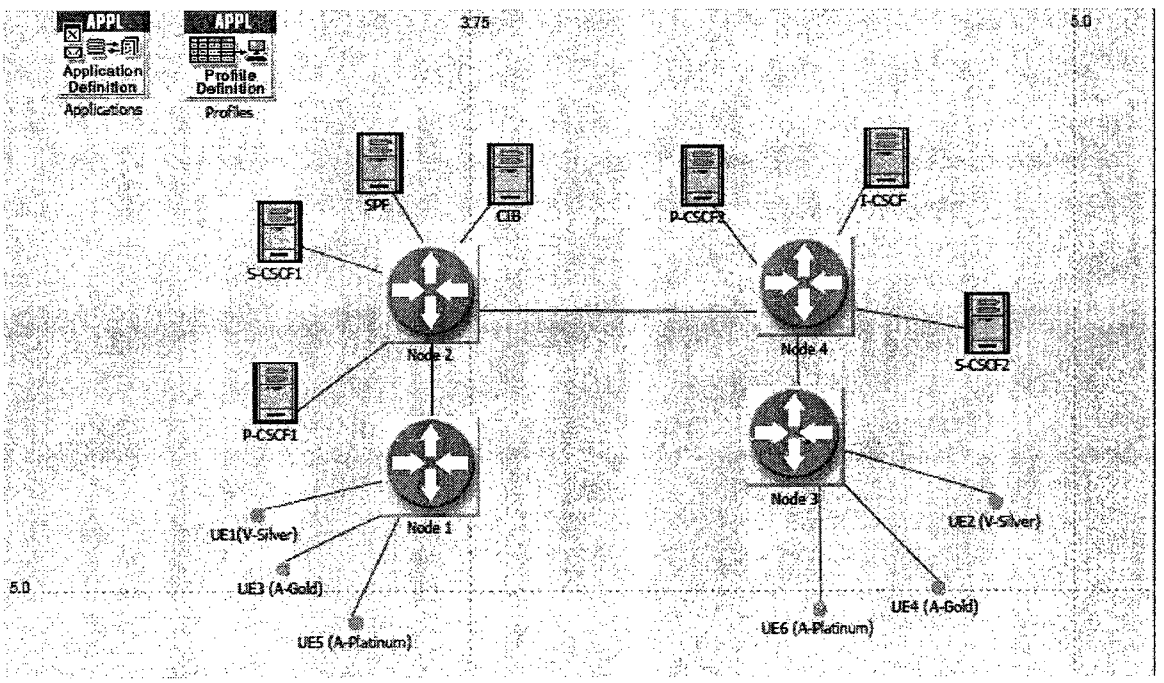


Figure 8.15: Call differentiation scenario – high load regular calls case

In order to test emergency related operations, we defined another scenario as shown in figure 8.16. This scenario involves the following steps: First, an audio session of type Silver is established between UE1 and UE2 in addition to a video Gold session between

UE3 and UE4; then an audio mission critical call is attempted between UE5 and UE6 leading to the downgrade of the second session. Afterwards, another audio mission critical call is established between UE7 and UE8, triggering the termination of the first session. This is followed by the establishment of a video Gold session between UE9 and UE10, triggering the termination of the second session. UE11 and UE12 then create a Silver video session, triggering the downgrade of the fifth session (i.e. the one between UE9 and UE10). This fifth session is then preempted due to the establishment of a 911 call by UE13. Finally, UE14 makes a 911 call triggering the downgrade of the sixth session (i.e. the one between UE11 and UE12). To end the scenario, all the remaining active sessions (i.e. the 3rd, 4th, 6th, 7th, and 8th sessions) are ended. It should be noted that all these interactions were necessary to create an environment with different types of regular and emergency calls, and to allow the testing of the different operations related to emergency sessions.

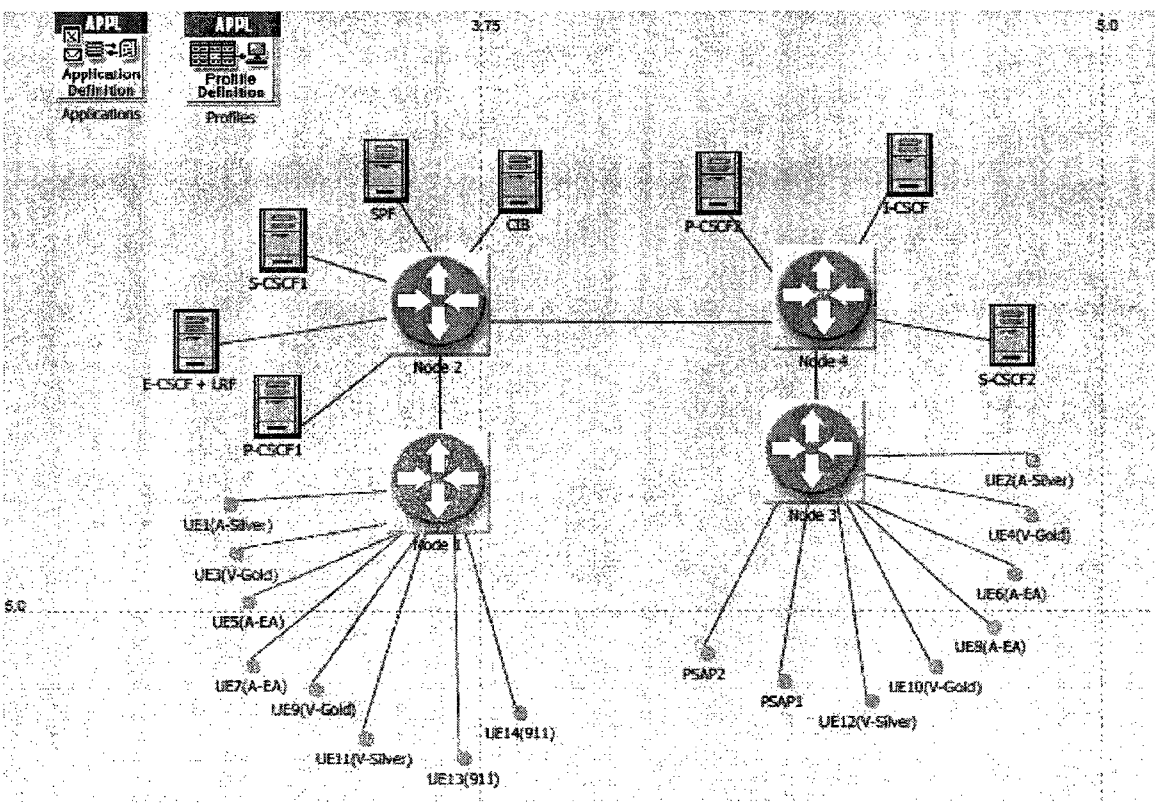


Figure 8.16: Call differentiation scenario – high load emergency calls case

In addition to the two previous scenarios that involved one call per UE under high loading conditions, we defined a third scenario to test the system dynamics with respect to changing loading conditions. Figure 8.17 depicts that scenario that involved twenty UEs, ten of which acting as call generators. These generators were used for the initiation of all the different categories of calls (i.e. audio-silver, video-silver, audio-gold, video-gold...etc), while producing a gradually increasing network traffic. In the coming section, we will present the performance measurements collected using these three scenarios.

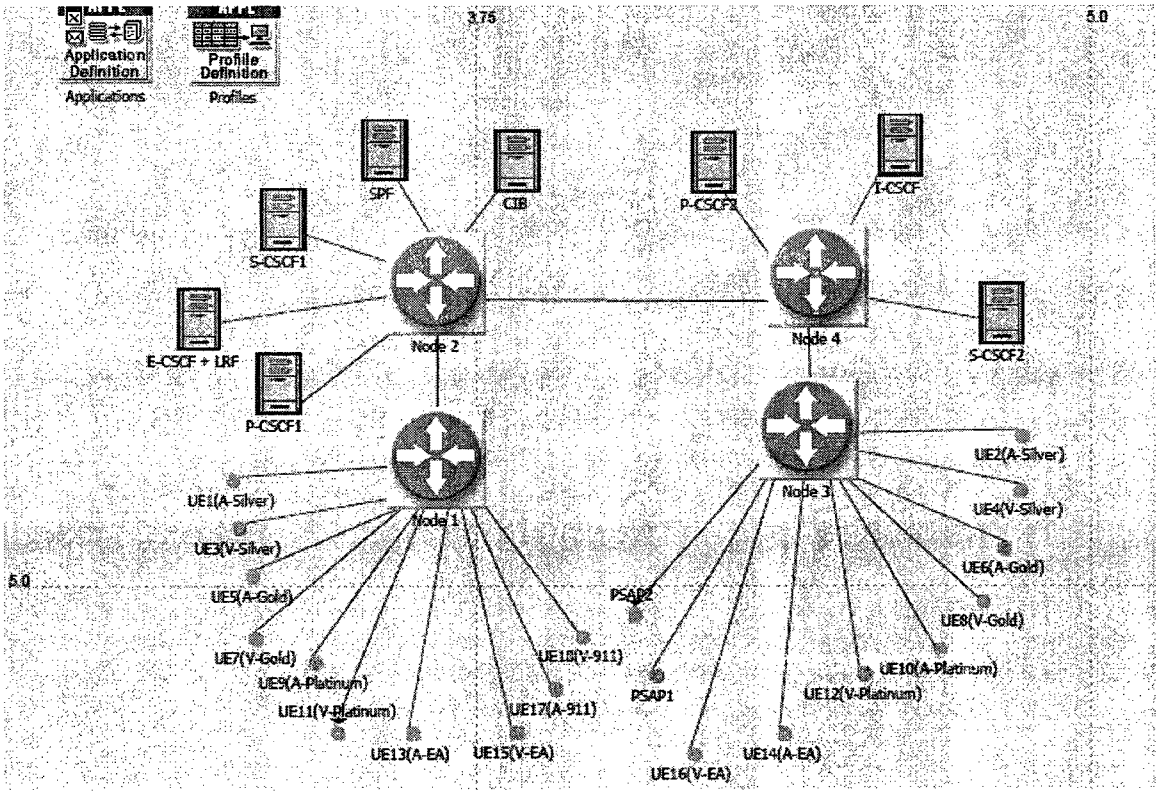


Figure 8.17: Call differentiation scenario – gradually increasing network load case

8.4 Measurements and Analysis

In this section, we present the performance metrics followed by the analysis of the performance measurements collected using the simulation.

8.4.1 Performance Metrics

We used the following metrics to evaluate the performance of our solution:

- **Call setup time and network load:** In order to evaluate whether our system achieves a satisfactory performance in terms of call setup time (i.e. requires short call setup times for the different scenarios) and calculate the overhead introduced by call differentiation (with respect to the basic case involving no call differentiation), we used two metrics: the call setup time (in ms) and the network load (in bytes). The call setup time is measured at the UE level as the time duration from when the initial SIP INVITE message is sent by the call originator until a final SIP OK response is received by it. As for the network load that is measured at the CSCFs level, it consists of the amount of signaling traffic exchanged within the network – the data traffic load not being considered in our measurements. We note that in the case of call differentiation scenarios, three main components contribute to the network load, namely: session management messages, resource allocation messages, and context information exchange messages.
- **Service differentiation metrics:** In order to examine the system's dynamics with respect to varying load conditions, we focused on three of the service differentiation metrics defined in our differentiation scheme, namely: the Call Blocking Probability, the Forced Call Termination Probability, and the Media Type Guarantee. The CBP is evaluated by measuring the number of the calls blocked (i.e. rejected at admission), while the FCTP probability is evaluated using the number of calls terminated based on a SPF decision. As for the MTG, it is evaluated using the number of downgraded sessions.

8.4.2 Call Setup Time and Network Load

Figure 8.18 shows the call setup time measurements for various regular and emergency call scenarios involving call differentiation with respect to the base scenarios involving no call differentiation. As shown, the smallest call setup time (i.e. 2.3167 sec) is achieved for the basic emergency call scenario with no call differentiation (Emg – No call diff.). In comparison, the basic regular call setup time is more than twice the emergency call setup time (i.e. 5.090 sec). This is due to the fact the number of intermediary nodes involved in regular call setup is more than for regular call setup (i.e. 2 nodes for the emergency case vs. 5 nodes for the regular case), thus increasing the overall messages processing and propagation time within the network.

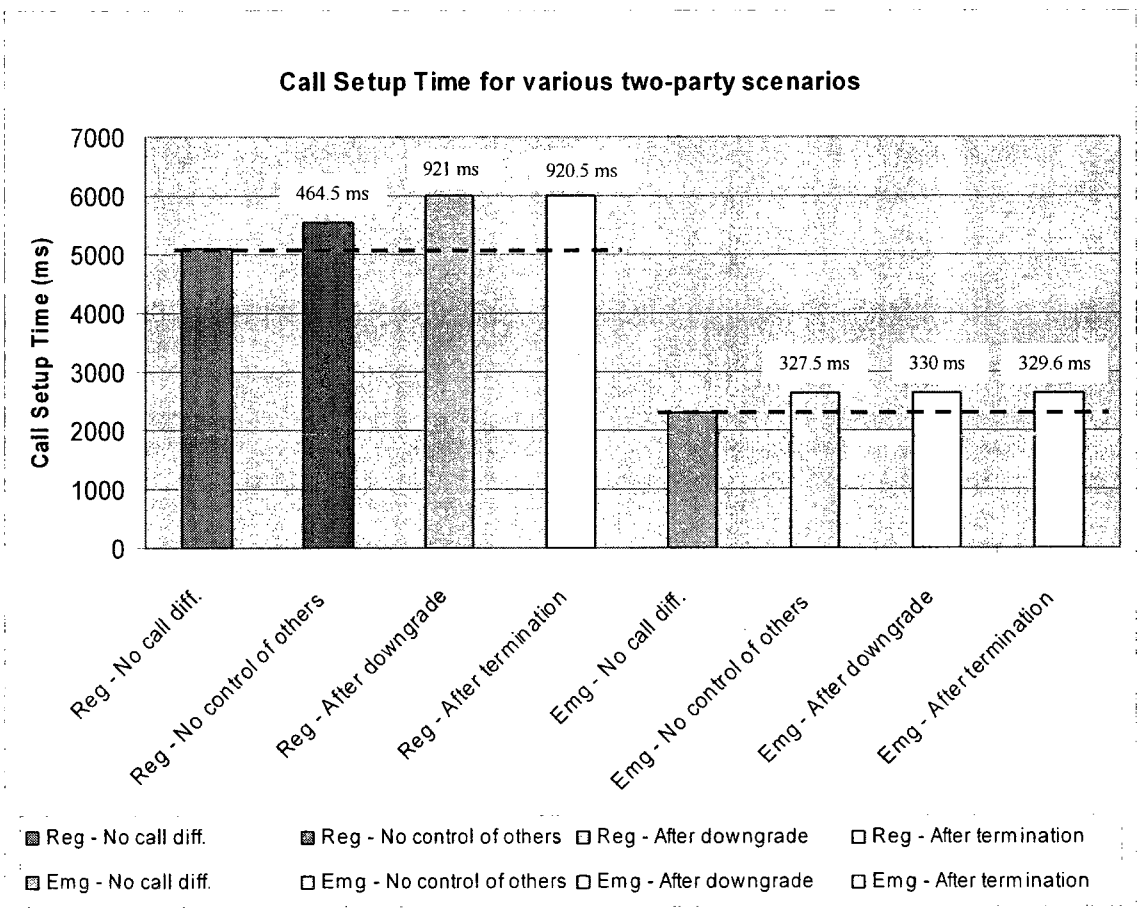


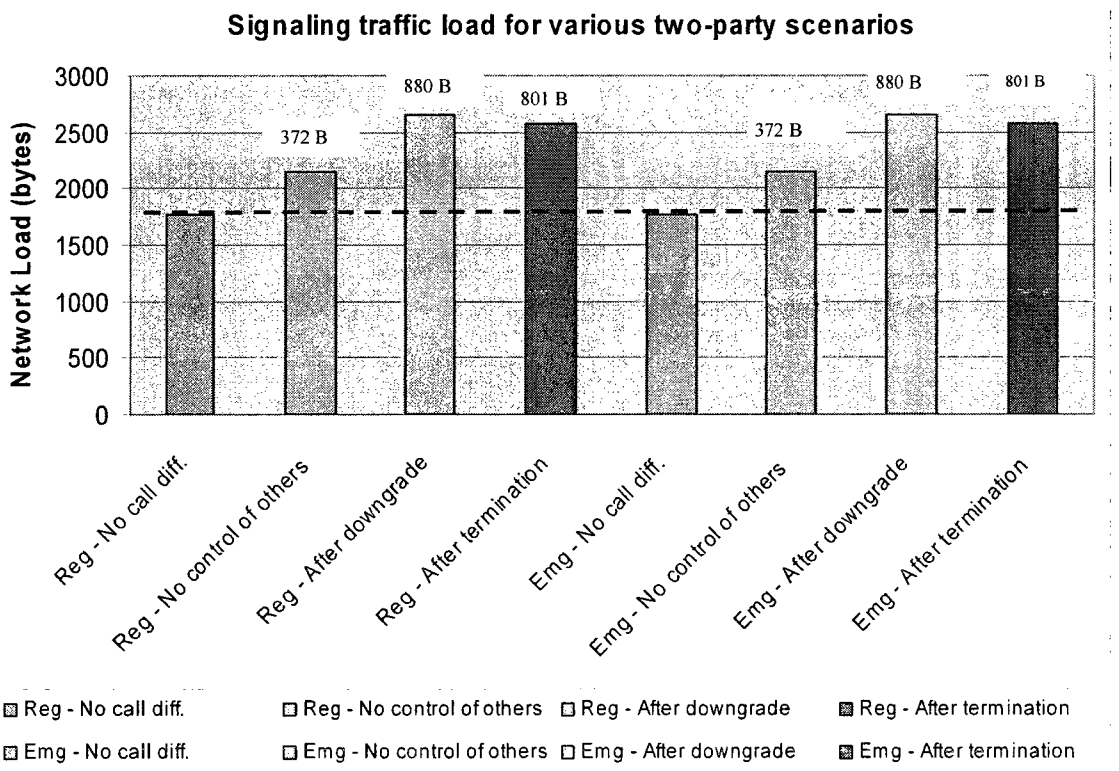
Figure 8.18: Call setup time measurements for regular and emergency call scenarios

Moreover, we notice that call differentiation introduces an overhead in terms of call setup time as expected. This overhead is due to the additional resource allocation and session management interactions needed in this case. As shown in the figure, the successful call establishment scenario with no attempt to control other calls (Reg – No control of others) introduces the lowest overhead with respect to the base scenario (i.e. an overhead of 464.5 ms). This is due to the fact that the number of additional messages introduced is limited to six messages (i.e. a pair of COPS messages for resource allocation and two pairs of SIMPLE messages for context information exchange) in this case, and the number of nodes involved is only increased by two (i.e. the SPF and the CIB). In contrast, the successful call establishment scenario after the downgrade of an ongoing call (Reg – after downgrade) introduces the highest overhead (i.e. an overhead of 921 ms), due to the 23 additional SIP messages it requires for the downgrade of the ongoing session as well as the additional pair of COPS messages for the modification of the previous resource allocation decision. The successful call establishment scenario after the termination of an ongoing call (Reg – after termination) achieves a similar performance with an overhead of 920.5 ms – the slight difference with respect to the downgrade scenario stemming from the fact that a smaller number of SIP messages (18 vs. 23 messages) is needed for the termination of an ongoing session in comparison to its re-negotiation. A similar analysis can be made for the emergency related scenarios in which the overhead ranges between 327.5 ms (for the case involving no control of other sessions) and 330 ms (for the case involving sessions downgrade).

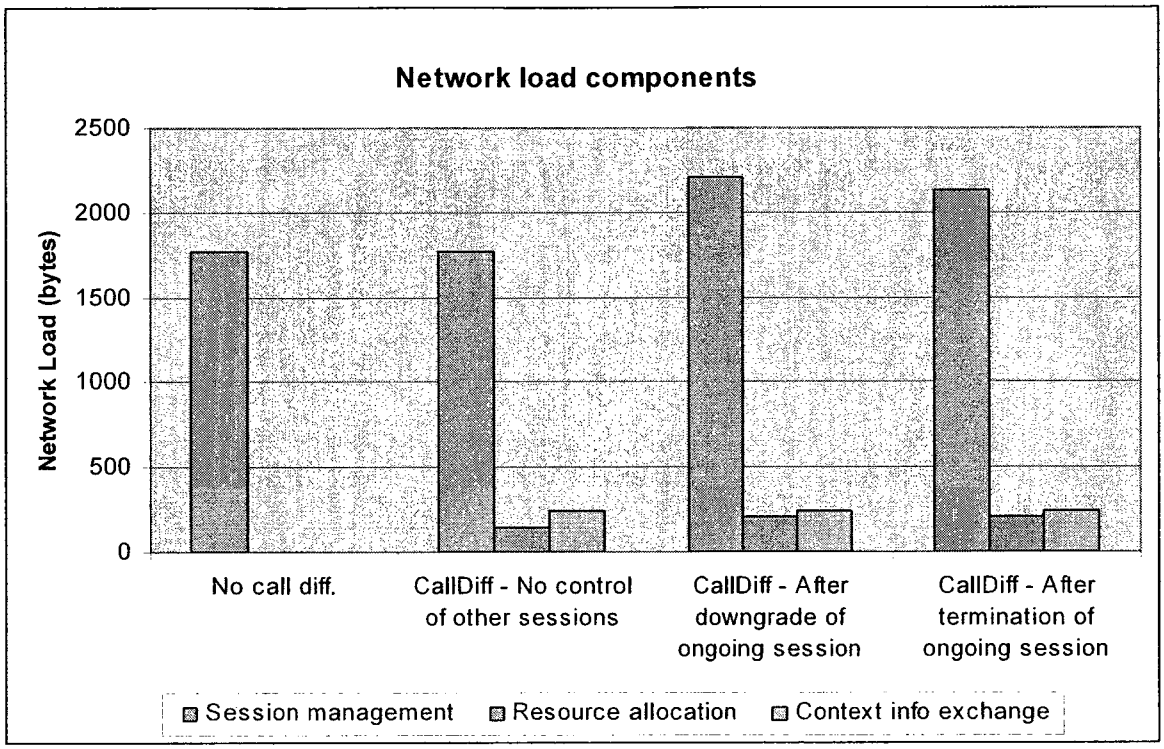
By examining these results, we can conclude that the overhead introduced by call differentiation (ranging between 327.5 ms and 921 ms) is acceptable, since it will be barely

felt by the application's user. Furthermore, we note that this delay is only introduced (to allow the establishment of high priority sessions) during high loading conditions.

Figure 8.19a shows the signaling traffic load measurements. Unlike the call setup delay measurements that depend on the messages processing time by the different intermediary nodes and the messages propagation time between those nodes, the network load is only affected by the size of the messages sent/received by each of these nodes. Since regular and emergency scenarios involve the same number of messages (the difference being in the number of intermediary nodes involved in sessions' establishment), the same network load results were obtained for both cases. As shown in the figure, basic scenarios involving no call differentiation (i.e. Reg-No call diff. and Emg-No call diff.) generate a signaling traffic load of 1.73 Kbytes for successful session establishment.



a)



b)

Figure 8.19: Signaling traffic load measurements for regular and emergency call scenarios: a) Network load measurements; b) Network load components

In terms of the overhead introduced by call differentiation, the following results were obtained: an overhead of 372 bytes for the case of successful call establishment with no control of other sessions; an overhead of 880 bytes for the case of successful call establishment after the downgrade of an ongoing session; and an overhead of 801 bytes for the case of successful call establishment after the termination of an ongoing session. Figure 8.19b clarifies the different components contributing to this overhead. As depicted, the basic scenario involving no call differentiation only relies on one type of messages (i.e. session management messages) in its operation, while the call differentiation scenario with no control of other sessions introduces two additional types of messages (i.e. resource allocation and context information exchange messages) but keeps the number of session management messages unchanged. In contrast, the call differentiation scenarios involving

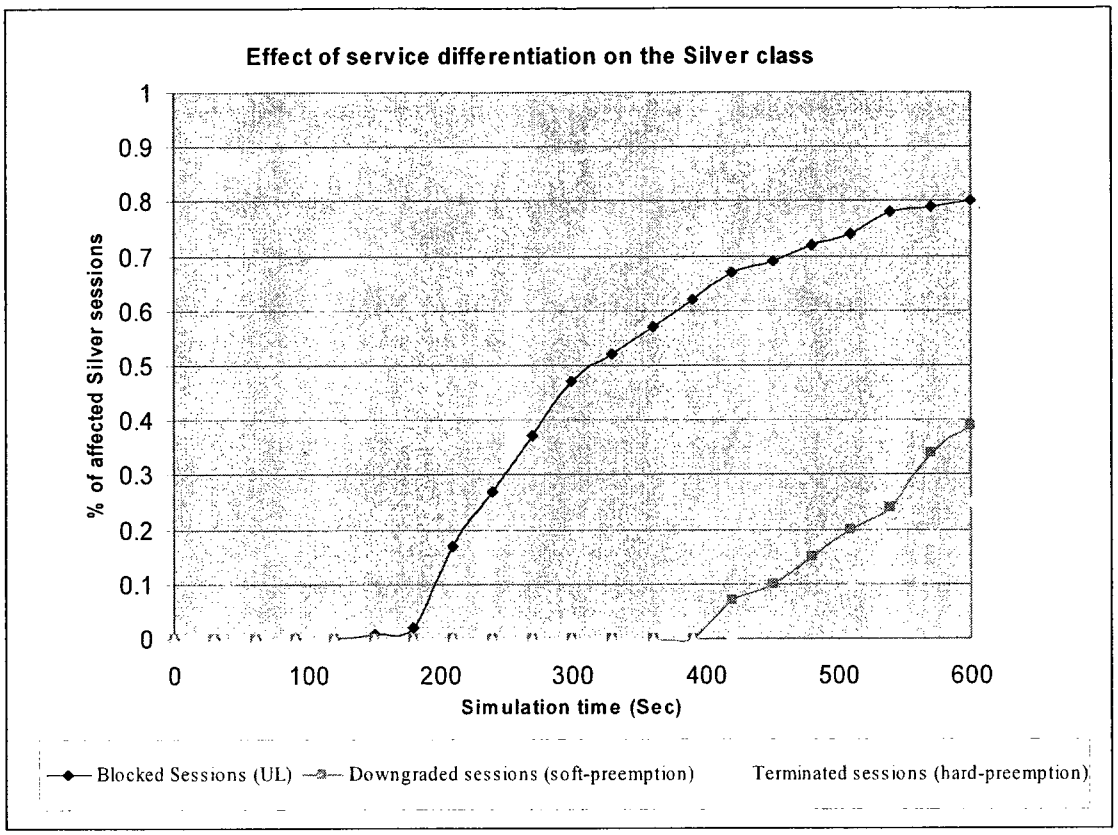
the downgrade/termination of ongoing sessions introduce some additional resource allocation messages and a significant amount of session management messages to enable the preemption of ongoing sessions.

Based on these results, we can conclude that the network load overhead introduced is not significant, even in the cases involving sessions' preemption, considering the savings that will be achieved by downgrading/terminating the media streams of low priority sessions. Furthermore, the impact of this overhead on the air interface will be further minimized using the signaling compression techniques [113] already used in 3G networks.

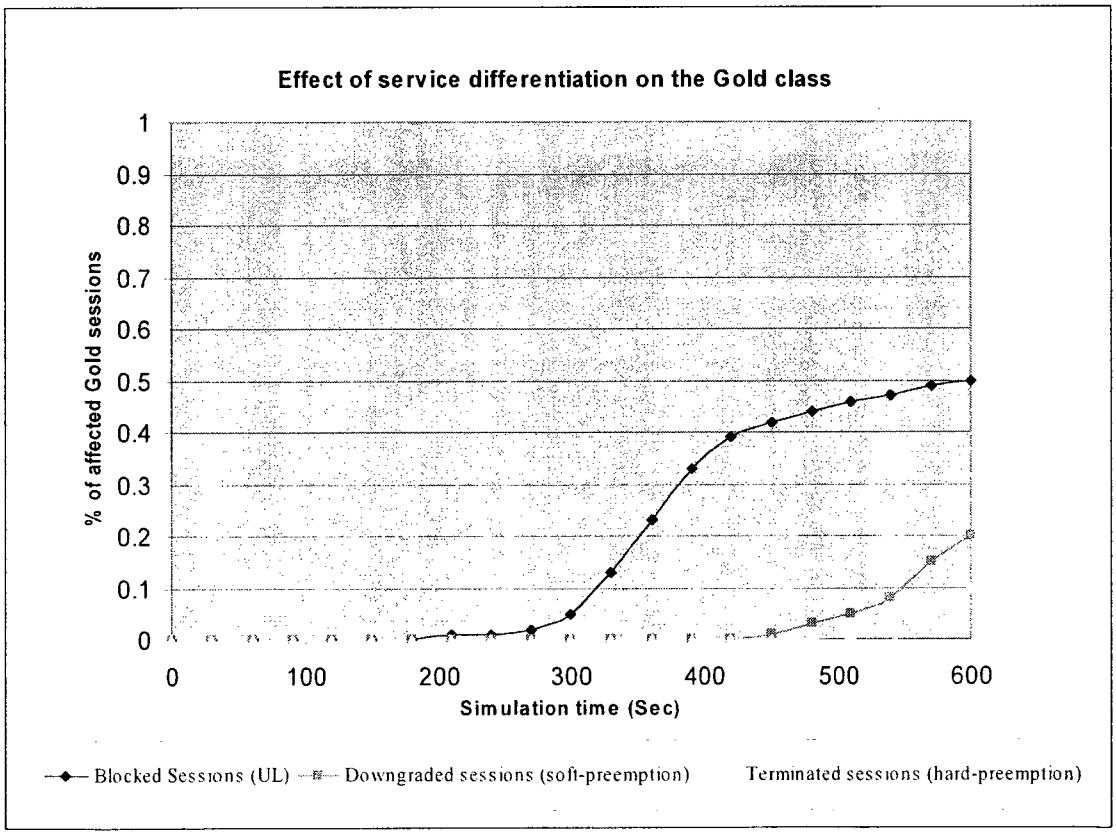
8.4.3 Service Differentiation Parameters

To study the effects of service differentiation on the different classes of calls, we used the scenario presented in figure 8.17, and configured each call generator to generate calls at a rate of 10 calls/minute following a Poisson distribution – each class contributing to 20% of the total network load. We should note that the sessions initiated are allowed to run until the end of the simulation, thus leading to an incrementally increasing network load with no resource release within the simulation run (except via network-initiated sessions' preemption). Furthermore, the simulation time was set to 10 minutes and three runs with different seeds were used for the calculation of the results presented in figure 8.20. The measurements used for the computation of these results were collected at the SPF level, by logging the admission, rejection, termination, and downgrade events of sessions of different categories and the timestamps of these events.

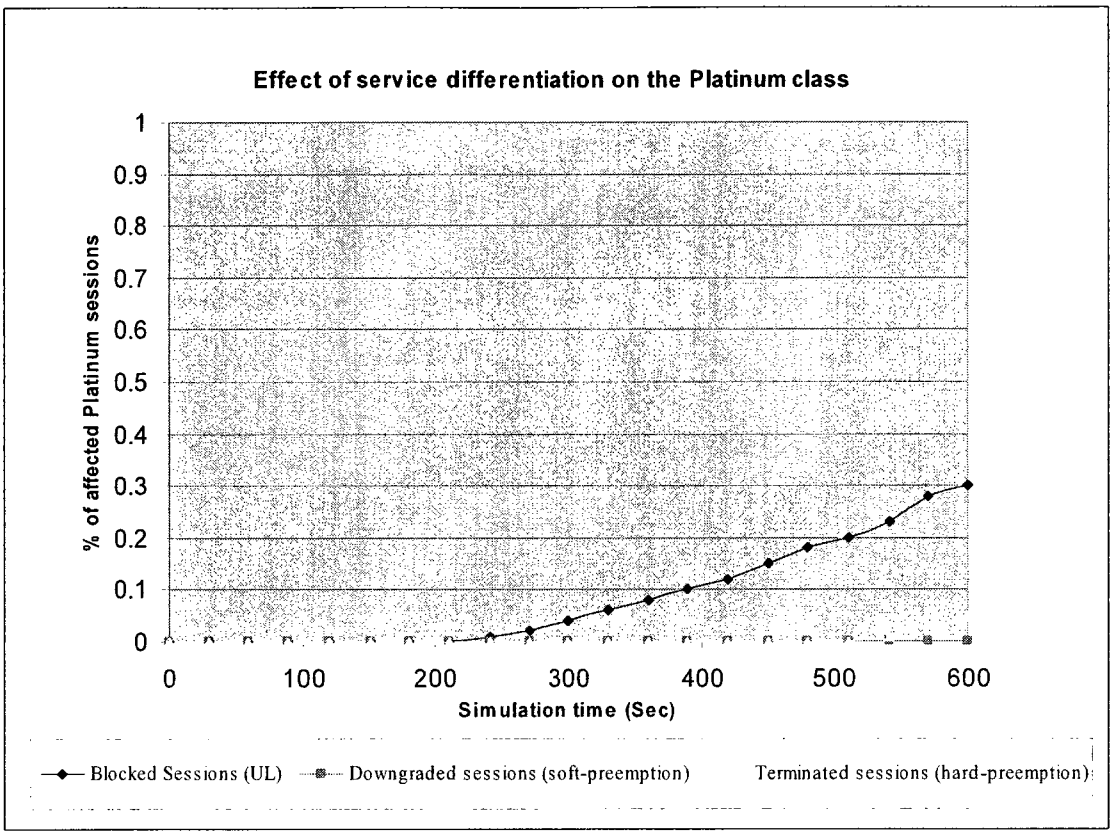
a)


Effect of service differentiation on the Silver class
% of affected Silver sessions
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Simulation time (Sec)
Blocked Sessions (UL)
Downgraded sessions (soft-preemption)
Terminated sessions (hard-preemption)

b)


Effect of service differentiation on the Gold class
% of affected Gold sessions
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Simulation time (Sec)
Blocked Sessions (UL)
Downgraded sessions (soft-preemption)
Terminated sessions (hard-preemption)

c)



d)

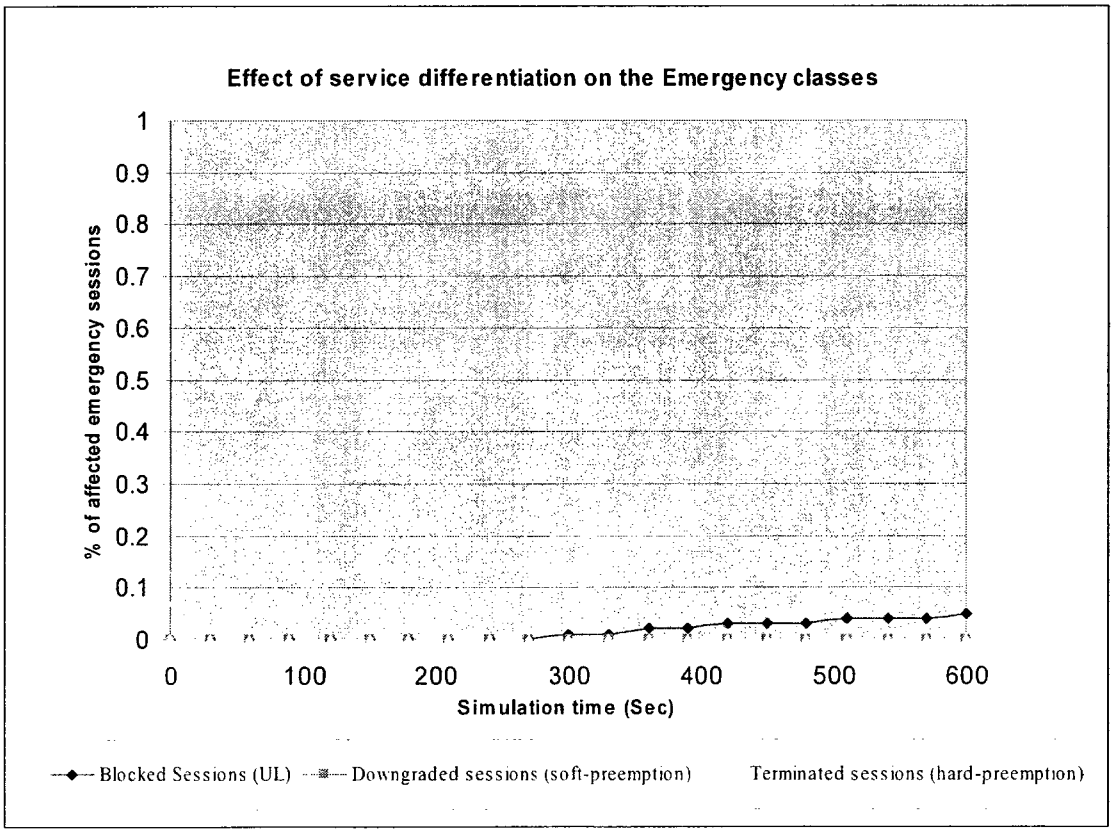


Figure 8.20: Service differentiation measurements for the different session classes: a) Effect of service differentiation on the silver class; b) Effect of service differentiation on the gold class; c) Effect of service differentiation on the platinum class; d) Effect of service differentiation on the emergency classes

Figure 8.20a illustrates the effect of service differentiation on the handling of silver class sessions, by showing the % of affected sessions (i.e. blocked, downgraded, and terminated sessions) within the simulation run. The blue curve shows the blocking of sessions as a result of the upper limit call admission policy. We notice that sessions start getting blocked after 120 seconds of the start of the simulation. This is the moment at which the number of silver sessions starts exceeding the threshold set for this class. The number of blocked sessions then increases gradually (since all the resources allowed for this class have been used), until reaching 80% of the attempted sessions. This percentage is affected by the CBP set for the silver class as part of the call admission algorithm. As for the purple and yellow curves, they respectively show the % of sessions affected by soft and hard preemption. We notice that these two curves start having positive values after 420 seconds of the simulation – this moment representing the point at which the high load threshold is reached. Furthermore, although both curves show an increasing trend, the soft preemption curve increases at a faster rate then the hard preemption curve - thus implying that more sessions are downgraded then terminated. In fact, more sessions are affected by soft preemption than hard preemption since fewer resources are freed by sessions' downgrade (thus requiring the downgrade of more sessions for freeing enough resources), and also because soft-preemption is attempted before hard preemption in our algorithm. Figures 8.20b, 8.20c, and 8.20d show similar results obtained from the measurements made for the remaining service classes. We note that for the gold class, the maximum percentage of blocked sessions is much lower than for the silver case (50% vs. 80%). The same applies to the maximum percentages of gold sessions affected by soft/hard preemption (i.e. 20% and 10% respectively). This is due to the values of the blocking and preemption probabilities

that are set differently for these two classes. Similarly, since the soft-preemption probability is set to zero for the platinum class, the graph shows no results for sessions' downgrade. As for the emergency classes, we notice that they experience no sessions' downgrade/termination and show a very small % of session blocking (ranging between 1% and 5%). This case occurs when no resources from other classes could be freed using preemption during a high load situation and the emergency session had to be rejected.

Based on these results, we can conclude that service differentiation is effectively achieved between the defined classes, since higher priority classes experience smaller blocking and preemption rates than lower priority classes. Furthermore, we see that service differentiation helps achieves a better and more controlled repartition of resources among different session classes based on their priorities and the various loading conditions, as opposed to resources being allocated arbitrarily to calls without any consideration to their priorities.

8.5 Conclusions

In this chapter we presented the OPNET-based simulations used to evaluate the performance of our call differentiation architecture. The designed OPNET node and process models were presented, and the protocol stacks implementation and applications/profiles definitions were detailed. Moreover, the simulation scenarios used were presented and the performance measurements and their analysis were discussed. These measurements showed that the overhead introduced by call differentiation, in terms of call setup time and network load, remains acceptable for the different call differentiation scenarios. Furthermore, the results demonstrated that service differentiation is effectively achieved within the system and that this capability helps accomplish a more efficient and controlled repartition of resources among different sessions based on their priorities.

Chapter 9

Conclusions and Future Work

In this chapter, we highlight the contributions of this thesis and give hints about future research directions.

9.1 Summary of Contributions

The 3GPP-defined IP Multimedia Subsystem is rapidly becoming the de-facto standard for IP-based multimedia services. Nevertheless, the IMS still faces several challenges as it continues to evolve. Examples of these challenges include: the enabling of innovative and personalized services; its interaction with other types of networks as means to enhance its capabilities; and the support of advanced QoS schemes that would manage network resources in an efficient/adaptive manner.

The Context-awareness technology is considered to enhance users' experience and is seen as an enabler to adaptability and service personalization - two capabilities that could play important roles in telecommunication environments. In this thesis, we addressed the introduction of the context-awareness technology in the IMS, as means to enhance its control operations and service provisioning capabilities. The combination of the IMS and the context-awareness technologies entails two main categories of issues: issues related to the acquisition and management of contextual information in the IMS; and issues related to the integration of contextual information in IMS operations. The main contributions of this thesis tackling these issues are summarized as follows:

- **Derived requirements related to the management and usage of contextual information in IP-based networks and reviewed the related work:** We derived requirements related to the issues of context acquisition/management, service

differentiation, and enhanced emergency support in IP-based networks, and reviewed the related work in light of these requirements. Two categories of information acquisition/management solutions were reviewed: WSN integration solutions and information management solutions. The evaluation showed that the 3GPP presence framework is the most promising of these solutions, although has it misses some important requirements related to context information management. Related to service differentiation in IP-based networks, two groups of solutions were evaluated, namely: solutions operating at the access network level; and solutions operating at the core network level. These solutions show several limitations with respect to our requirements such as: the lack of flexibility in terms of QoS negotiation, offering limited control over the different communication aspects used for differentiation, in addition to mostly offering coarse-grained service differentiation schemes and not employing dynamic/adaptive resource management strategies nor specialized charging models. Finally, in relation to enhanced emergency communication support in IP-based networks, both legacy emergency solutions and emerging IP-based emergency solutions were reviewed. The evaluation showed that all these solutions present limitations with respect to three main aspects: QoS and resource management; context-awareness and service personalization; and the emergency communication models used.

- **Proposed a presence-based architecture for context information acquisition and management in the IMS:** To ensure the availability of contextual information within the network, we have proposed a solution for context information acquisition and management in the IMS. This solution, which leverages and extends the 3GPP presence

framework, consists of an IMS context management architecture and several components related to its operation. The proposed architecture extends the existing 3GPP presence architecture to enable the IMS interaction with physical/logical sensors for the collection of a variety of contextual information in addition to enabling the effective management and dissemination of this information in the network. Related to the operation of this architecture, an extension of the presence information model was proposed to permit the representation of additional types of information provided by sensors, and three information exchange models were devised to achieve flexible and resource efficient information exchange within the network. Furthermore, an example business model along with suitable identification/charging schemes were proposed to enable the practical deployment of the proposed architecture, and other support functions needed for its operation (i.e. security, info access control, and service discovery) were discussed.

- **Proposed an IMS call differentiation framework:** We demonstrated the use of context-awareness at the IMS control level as means to offer enhanced QoS support in IMS-based networks, by proposing a context-aware call differentiation solution. This solution, which enables the differentiation between different categories of calls at the IMS control level via dynamic and adaptive resource allocation, consists of the following components: A novel call differentiation scheme; along with an architectural framework; two dynamic/adaptive resource management techniques (i.e. call admission control and media parameter control); and a charging model to support this scheme in the IMS. Compared to existing service differentiation solutions, this solution offers several benefits, such as: flexible QoS negotiation mechanisms; control over many

communication aspects as means for differentiation; a dynamic and adaptive resource management strategy; and a specialized charging model.

- **Proposed a framework for enhanced IMS emergency communication services:** We demonstrated how context-awareness can be integrated at the IMS control level as means to support enhanced IMS emergency communication services. The enhanced IMS emergency solution proposed addressed the main limitations of existing IP-based emergency solutions, by offering three improvements, namely: a QoS-enhanced emergency service; a context-aware personalized emergency service; and a conferencing-enhanced emergency service. The enhancement of the QoS/resource management aspect of emergency sessions was achieved by generalizing our previously proposed call differentiation solution (originally tackling the case of regular calls), and applying it to the emergency case. The second enhancement focused on the exploitation of a wide range of contextual information as means to improve the efficiency of emergency operations and offer personalized emergency services to users. As for the third enhancement, it tackled the improvement of the IMS emergency service architecture with conferencing capabilities, as means to offer richer forms of emergency communications to users.
- **Designed and implemented proof-of-concept prototypes and context-aware applications using Ericsson's service development studio as implementation platform:** We used Ericsson's Service Development Studio (an IMS simulated environment) and three implementation technologies (SIP, COPS, and Diameter) to build proof-of-concept prototypes of our solutions. We also demonstrated the use of context awareness at the IMS service level using two new context-aware applications.

The used COPS stack was extended with a new policy client type and client related objects to enable the exchange of policy-based resource allocation/re-allocation decisions, while the SIP stack was enhanced with two existing extension headers to support QoS negotiation related interactions. As for the Diameter stack, it was extended to carry the additional information needed for price differentiation and compensation credits calculation. All the functional entities proposed in our solutions were built and introduced either as extension components to SDS or as enhancements in the logic of one of its existing components, and different test scenarios were carried to prove the feasibility of the solutions proposed and collect some performance measurements about a sub-set of the solutions. From the experiments, we found that the chosen implementation technologies work well for the different scenarios and that the main concepts proposed (i.e. WSN/IMS integration, signaling level call differentiation, and enhanced emergency operations) are feasible in an IMS environment. We also showed that the use of context as an IMS application building block facilitates the development of novel context-aware value-added services, by abstracting developers from the complexities of context acquisition and management operations.

- **Evaluated the performance of the call differentiation architecture using the OPNET simulation tool:** We used OPNET simulations to evaluate the performance of the call differentiation architecture and its generalization (i.e. the QoS-enhanced emergency service architecture). To achieve these simulations, several node/process models were designed and two protocol stacks were implemented. Furthermore, some application profiles were defined to enable the generation of different traffic patterns and the execution of different scenarios involving various application sessions with

specified start/end times. Three types of scenarios were defined using these application profiles and various performance measurements were collected. The analysis of these measurements showed that the overhead introduced by call differentiation, in terms of call setup time and network load, remains acceptable for the different call differentiation scenarios. Furthermore, the results demonstrated that service differentiation is effectively achieved within the system and that this capability helps accomplish a more efficient and controlled repartition of resources among different sessions based on their priorities.

9.2 Future Work

This thesis focused on the introduction of the context-awareness technology in the IMS and touched several areas related to this topic, including: context acquisition/management; service differentiation; enhanced emergency services; and the support of innovative value-added services in IMS-based networks. Several other venues related to the use of context-awareness in the telecommunications field could be investigated.

One of the interesting venues related to the topic of improved QoS support in IMS-based networks is the integration of the signaling-level call differentiation scheme we proposed with other schemes operating at the access and routing levels. In fact, enabling the interaction between the different prioritization mechanisms acting at the different levels would guarantee the consistency of the mapping between those mechanisms and lead to an optimization of the resource management solution. This idea could be achieved using cross-layering [114] interactions between the different schemes and multi-objective optimization. Other interesting venues include: the application of the proposed service differentiation scheme to other types of services (e.g. streaming and interactive services)

beside the conversational services targeted; studying the effect of differentiation on value-added services operation; and the incorporation of user preferences in the call differentiation scheme - thus combining users' preferences and prioritized call handling in IMS-based networks.

As another research direction, it would be interesting to investigate the introduction of the context-awareness technology in other types of networks, such as the internet, mobile ad hoc networks (MANETs) [115], and multi-hop cellular networks (MCNs) [116], as means to improve their operations. For instance, in the internet domain, context-awareness can be used to support the combination of web services in order to offer composite services that suit the user's current situation and provide him/her with a good QoS in adaptation to variations in the execution environment. In the case of MCNs, information about the network resources status could be leveraged to switch to the multi-hop routing mode, in order to improve the network's throughput. As for challenged networks such as MANETs, context-awareness could be used to enable interesting features addressing the particularities of these networks. An example of such features is automatic protection of sessions against security threats via the monitoring of the network events and the isolation of malicious nodes from sessions. Another interesting feature could be the consideration of users profiles (instead of identities as it is done today) as basis to spontaneously initiate sessions between users sharing similar interests.

REFERENCES

[1] G. Camarillo and M. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*, Second edition. England: John Wiley & Sons Ltd., May 2006.

[2] The Third Generation Partnership Project, [online] available at: http://www.3gpp.org/ [accessed on Dec. 6th 2008].

[3] The European Telecommunications Standards Institute, [online] available at: http://www.etsi.org/ [accessed on Dec. 6th 2008].

[4] PacketCable, [online] available at: http://www.packetcable.com/ [accessed on Dec. 6th 2008].

[5] List of global IMS deployments, [online] available at: www.incodewireless.com/media/whitepapers/2006/Global_IMS_Deployments.xls [accessed on Dec. 6th 2008].

[6] NEC's Service Convergence Integrated Platform, [online] available at: http://www.telephonyworld.com/products/nec-introduces-ip-multimedia-subsystem-ims-migration-path-for-north-america/ [accessed on Dec. 10th 2008].

[7] Advanced IMS Inc. software products, [online] available at: http://www.advancedims.com/index.html [accessed on Dec. 10th 2008].

[8] G. Abowd et al., "Towards a Better Understanding of Context and Context-Awareness," in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing (HUC'99)*, September 1999, pp. 304-307.

[9] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications Magazine,* vol. 8, no.4, pp. 10-17, August 2001.

[10] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek and R. Dssouli, "A Presence-Based Architecture for the Integration of the Sensing Capabilities of Wireless Sensor Networks in the IP Multimedia Subsystem," in *Proceedings of IEEE Wireless Communications and Networking Conference 2008 (WCNC' 08),* March 2008, pp. 3116-3121.

[11] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek and R. Dssouli, "An Architecture for the Provision of Context-Aware Emergency Services in the IP Multimedia Subsystem," in *Proceedings of the 67th IEEE Vehicular Technology Conference 2008 (VTC 2008-Spring)*, May 2008, pp. 2784-2788.

[12] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek and R. Dssouli, "The Design and Implementation of a Gateway for IP Multimedia Subsystem/Wireless Sensor Networks Interworking," in *Proceedings of the 69th IEEE Vehicular Technology Conference 2009 (VTC 2009-Spring)*, April 2009, pp. 1-5.

[13] D. Balakrishnan, M. El Barachi, A. Karmouch and R. Glitho, "Challenges in Modeling and Disseminating Context Information in Ambient Networks," in *Proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications (MATA 05)*, October 2005, pp. 32-42.

[14] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek and R. Dssouli, "The Design and Implementation of Architectural Components for the Integration of the IP Multimedia Subsystem and Wireless Sensor Networks," Submitted to IEEE Communications Magazine's Design and Implementation Series.

[15] M. El Barachi, R. Glitho and R. Dssouli, "Context-Aware Signaling for Call Differentiation in IMS-Based 3G Networks," in *Proceedings of the 12th IEEE Symposium on Computers and Communications 2007 (ISCC'07)*, July 2007, pp. 789-796.

[16] M. El Barachi, R. Glitho and R. Dssouli, "Charging for Multi-Grade Services in the IP Multimedia Subsystem," in *Proceedings of the 2nd IEEE International Conference and Exhibition on Next Generation Mobile Applications, Services, and Technologies (NGMAST'08)*, September 2008.

[17] M. El Barachi, R. Glitho and R. Dssouli, "Service Differentiation in the IP Multimedia Subsystem Utilizing Context-Aware Signaling," United States Patent, Serial No.: 891378, Series Code: 60, February 23rd, 2007.

[18] M. El Barachi, R. Glitho and R. Dssouli, "Enhancing the QoS and Resource Management Aspects of the 3GPP IMS Emergency Service Architecture", in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference 2008 (CCNC 2008)*, January 2008, pp. 112-116.

[19] 3GPP TS 23.228, "IP multimedia subsystem (Release 5 – v.5.5.0),"June 2002.

[20] 3GPP TS 23.228, "IP multimedia subsystem (Release 6 – v.6.7.0),"September 2004.

[21] 3GPP TS 23.228, "IP multimedia subsystem (Release 7 – v.7.11.0),"March 2008.

[22] The Third Generation Partnership Project 2, [online] available at: http://www.3gpp2.org/ [accessed on Dec. 9th 2008].

[23] The Telecoms and Internet Converged Services and Protocols for Advanced Networks, [online] available at: http://www.etsi.org/tispan/ [accessed on Dec. 9th 2008].

[24] 3GPP TS 22.228, "Service Requirements for the IP multimedia core network subsystem (Release 5 – v.5.5.0),"March 2002.

[25] R. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, Internet Engineering Task Force, June 2002.

[26] A. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, Internet Engineering Task Force, June 2002.

[27] G. Camarillo, W. Marshall and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)," RFC 3312, Internet Engineering Task Force, October 2002.

[28] H. Schulzrinne and J. Rosenberg, "The Session Initiation Protocol: Internet Centric Signaling," *IEEE Communications Magazine*, vol. 14, no.4, pp. 134-141, October 2000.

[29] P. Calhoun, J. Loughney, E. Guttman, G. Zorn and J. Arkko, "Diameter Base Protocol," RFC 3588, Internet Engineering Task Force, September 2003.

[30] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan and A. Sastry, "The COPS (Common Open Policy Service) Protocol," RFC 2748, Internet Engineering Task Force, January 2000.

[31] ITU-T, "Gateway Control Protocol: Version 2," Recommendation H.248, International Telecommunication Union, May 2002.

[32] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, Internet Engineering Task Force, July 2003.

[33] 3GPP TS 23.002, "Network Architecture (Release 7 – v.7.1.0),"March 2006.

[34] 3GPP TS 23.207, "End-to-End Quality of Service (QoS) Concept and Architecture (Release 6 – v.6.4.0),"September 2004.

[35] 3GPP TS 23.207, "End-to-End Quality of Service (QoS) Concept and Architecture (Release 7 – v.7.0.0),"June 2007.

[36] 3GPP TS 23.203, "Policy and Charging Control Architecture (Release 7 – v.7.8.0),"September 2008.

[37] 3GPP TS 32.240, "Charging Architecture and Principles (Release 7 – v.7.2.0)," March 2007.

[38] 3GPP TS 32.260, "IMS Charging (Release 7 – v.7.4.0)," September 2007.

[39] 3GPP TS 32.299, "Diameter Charging Applications (Release 7 – v.7.7.0)," September 2007.

[40] H. Hakala, L. Mattila, J. Koskinen, M. Stura and J. Loughney, "Diameter Credit Control Application," RFC 4006, August 2005.

[41] M. day, J. Rosenberg and H. Sugano, "A Model for Presence and Instant Messaging," RFC 2778, Internet Engineering Task Force, February 2000.

[42] H. Sugano, S. Fujimoto, G. klyne, A. Bateman, W. Carr and J. Perterson., "Presence Information Data Format (PIDF)," RFC 3863, August 2004.

[43] H. Schulzrinne, V. Gurbani, P. Kyzivat and J. Rosenberg., "RPID: Rich Presence Extensions to the PIDF," RFC 4480, July 2006.

[44] H. Schulzrinne, "CIPID: Contact Information in PIDF," RFC 4482, July 2006.

[45] J. Peterson, "A presence-Based GEOPRIV Location Object Format," RFC 4119, December 2005.

[46] SIMPLE working group, "SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)", [online] Available at: http://www.ietf.org/html.charters/simple-charter.html [accessed on Dec. 6th 2008].

[47] P. Saint-André, "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 3920, October 2004.

[48] 3GPP TS 23.141, "Presence Service: Architecture and Functional Description (Release 7 – v.7.3.0)," September 2007.

[49] 3GPP TS 23.167, "IMS Emergency Sessions (Release 7 – v.7.3.0)," December 2006.

[50] A. Dey, "Providing Architectural Support for Building Context-Aware Applications", Ph.D. thesis, *Georgia Institute of Technology*, November 2000.

[51] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, pp. 393-422, August 2002.

[52] B. Rosen, H. Schulzrinne, J. Polk, and A. Newton, "Framework for Emergency Calling in Internet Multimedia", Internet draft – v.1, March 2007.

[53] A. Wood et al. "Alarm-Net: Wireless Sensor Networks for Assisted-Living and Residential Monitoring", Technical Report CS-2006-13, Department of Computer Science, University of Virginia, 2006.

[54] S. Krishnamurthy, "TinySIP: Providing Seamless Access to Sensor-based Services", in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems (MOBIQUITOUS'06),* July 2006, pp. 1-9.

[55] T. Luckenbach, P. Gober, S. Arbanowski, A. kotsopoulos, and K. Kim, "TinyREST - a Protocol for Integrating Sensor Networks into the Internet", in *Proceedings of the Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*, June 2005.

[56] K. Mayer and W. Fritsche, "IP-enabled Wireless Sensor Networks and their Integration into the Internet", in *Proceedings of the 1st International Conference on Integrated Internet Ad Hoc and Sensor Networks,* May 2006.

[57] A. Gluhak et al. "e-SENSE Reference Model for Sensor Networks in B3G Mobile Communication Systems", in *Proceedings of the 15th Information Society Technologies (IST) Summit*, June 2006.

[58] A. Haber, M. Gerdes, F. Reichert, and R. Kumar, "Remote Service Usage through SIP with Multimedia Access as a Use Case", in *Proceedings of the 18th International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2007)*, September 2007, pp. 1-5.

[59] G. Montenegro and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", Internet Draft – v1, October 2005.

[60] IEEE Standard 802.15.4-2003, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", May 2003.

[61] K. Kim, S. Park, G. Montenegro, S. Yoo, and N. Kushalnagar, "6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)", Internet Draft - v3, June 2007.

[62] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[63] S. Thomson, and T. Narten, "IPv6 Stateless Address Auto configuration", RFC 2462, December 1998.

[64] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

[65] OMA's presence and availability group, "Presence SIMPLE architecture (v. 2.0)", August 2008.

[66] OMA's presence and availability group, "Presence SIMPLE data specification (v.2.0)", July 2008.

[67] 3GPP TS 23.240, "3GPP Generic User Profile (GUP): Architecture (Release 6 – v.6.7.0)," March 2005.

[68] N. Nasser and H. Hassanein, "Adaptive Bandwidth Framework for Provisioning Connection-Level QoS for Next-Generation Wireless Cellular Networks", *The Canadian Journal of Electrical and Computer Engineering*, vol. 29, no. 1, pp. 101-108, January 2004.

[69] Y. Guo and H. Chaskar, "Class-Based Quality of Service over Air Interfaces in 4G Mobile Networks", *IEEE Communications Magazine*, vol. 40, no. 3, pp. 132-137, March 2002.

[70] C. Beard and V. Frost, "Prioritized Resource Allocation for Stressed Networks", *IEEE/ACM Transactions on Networking*, vol. 9, no. 5, pp. 618-633, October 2001.

[71] M. Luoma and J. Huttunen, "Differentiation of Traffic on Access Networks," in *Proceedings of the 1st International Conference on Multimedia Services Access Networks (MSAN 2005)*, June 2005, pp. 59-63.

[72] T. Raimondi and M. Davis, "Design Rules for a Class-Based Differentiated Service QoS Scheme in IEEE 802.11e Wireless LANs," in *Proceedings of the 7th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2004)*, October 2004.

[73] ETSI ES 282.003, "Resource and Admission Control Subsystem: Functional Architecture (v.1.1.1)", June 2006.

[74] H. Batteram, E. Meeuwissen and J. Bemmel, "SIP Message Prioritization and its Applications," *Bell Labs Technical Journal*, vol. 11, no. 1, pp. 21-36, May 2006.

[75] ITU Recommendation Q.735.3, "Stage 3 Description for Community of Interest Supplementary Services using SS7: Multi-Level Precedence and Preemption", March 1993.

[76] TINA-C, "TINA Business Model and Reference Points (v.4.0)", May 1997, [online] http://www.tinac.com/specifications/documents/bm_rp.pdf [accessed on June 7th 2008].

[77] 3GPP TS 33.203, "3G Security – Access Security for IP Services (Release 7 – v.7.5.0),"March 2007.

[78] 3GPP TS 33.210, "3G Security – Network Domain Security (Release 5 – v.5.0.0),"March 2002.

[79] J. Rosenberg, "The XML Configuration Access Protocol (XCAP)", RFC 4825, Internet Engineering Task Force, May 2007.

[80] M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, Internet Engineering Task Force, December 1997.

[81] OASIS Standards Consortium, [online] available at: http://uddi.xml.org/ [accessed on Mar. 9th 2009].

[82] L. Gong, "JXTA: A Network Programming Environment", *IEEE Internet Computing Magazine*, vol. 5, no. 3, pp. 88-95, May 2001.

[83] UPnP Forum, "UPnP Device Architecture, v. 1.1", [online] available at: http://www.upnp-ic.org/resources/UPnP_device_architecture_docs/ [accessed on Mar. 9th 2009].

[84] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol (v2)", RFC 2608, Internet Engineering Task Force, June 1999.

[85] OpenGIS OGC 02-023r4, "OpenGIS Geography Markup Language (GML) Encoding Specification (v. 3.0)", December 2002.

[86] M. Lonnfors, J. Costa-Requena, E. Leppanen, and H. Khartabil., "SIP Extension for Partial Notification of Presence Information", RFC 5263, Internet Engineering Task Force. September 2008.

[87] H. Khartabil, E. Leppanen, M. Lonnfors, and J. Costa-Requena, "Functional Description of Event Notification Filtering", RFC 4660, Internet Engineering Task Force, September 2006.

[88] A. Niemi and K. Kiss, "SIP Event Notification Extension for Notification Throttling", Internet draft – v.6, February 2008.

[89] M. Ahmed, "Call Admission Control in Wireless Networks: A Comprehensive Survey", *IEEE Communications Surveys and Tutorials Magazine*, vol. 7, no. 1, pp. 49-68, 2005.

[90] K. Ross, *Multiservice Loss Models for Broadband Telecommunications Networks* - Chapter 4 "Admission control", First edition: Springer Verlag, February 1997.

[91] C. Beard, "Preemptive and Delay-Based Mechanisms to Provide Preference to Emergency Traffic", *Elsevier's International Journal of Computer Networks*, vol. 47, no. 6, pp. 801-824, 2005.

[92] 3GPP TS 32.299, "Diameter Charging Applications (Release 7 – v.7.7.0)", September 2007.

[93] H. Hakala, L. Mattila, J-P. Koskinen, M. Stura, and J. Loughney, "Diameter Credit-Control Application", RFC 4006, Internet Engineering Task Force, August 2005.

[94] 3GPP TS 32.298, "CDR Parameter Description (Release 7 - v.7.4.0)", September 2007.

[95] H. Schulzrinne and J. Polk, "Communications Resource Priority for SIP", RFC 4412, Internet Engineering Task Force, February 2006.

[96] J. Polk, "Extending the SIP Reason Header for Preemption Events", RFC 4411, Internet Engineering Task Force, February 2006.

[97] 3GPP TS 24.147, "Conferencing using the IMS (Release 6 - v.6.3.0)", June 2005.

[98] ETSI Emergency Telecommunications (EMTEL) official website, [online] available at: http://www.emtel.etsi.org [accessed on Dec. 27th 2007].

[99] H. Dommel and J. Aceves, "Floor Control for Multimedia Conferencing and Collaboration," *ACM Multimedia Systems Magazine*, vol. 5, no. 1, pp. 23-38, 1997.

[100] H. Schulzrinne and J. Rosenberg, "Signaling for Internet Telephony", in *Proceedings of the Sixth International Conference on Network Protocols (ICNP'98)*, October 1998, pp. 298-307.

[101] A. Kadiwal, "Presence-based integration of Wireless Sensor Network and IP Multimedia Subsystem: architecture implementation and case studies", Master thesis, Concordia University, January 2009.

[102] Ericsson documentation, "Ericsson Service Development Studio 3.1 – Technical Product Description", February 2006, [online] available at: http://www.ericsson.com/mobilityworld/developerszonedown/downloads/docs/ims_poc/SDS_technical_description.pdf. [accessed on July 22nd 2009].

[103] JAIN-SIP-PRESENCE-PROXY, [online] available at: http://www-x.antd.nist.gov/proj/iptel/nist-sip-downloads.html [accessed on November 20th 2007].

[104] T. Ta, N. Othman, R. Glitho, and F. Khendek., "Using Web Services for Bridging End User Applications and Wireless Sensor Networks", in *Proceedings of the 11th IEEE Symposium on Computers and Communications 2007 (ISCC'06)*, June 2006.

[105] Adam Smith et al., "Tracking moving devices with the cricket location system", ", in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys 2004)*, June 2004.

[106] MTS300 sensor, [online] available at: http://www.xbow.com/Products/productsdetails.aspx?sid=75 [accessed on November 25th 2007].

[107] SIP Servlet API Specification (JSR 116), [online] available at: www.jcp.org/en/jsr/detail?id=116 [accessed on August 20th 2008].

[108] University of New South Wales, Open Source Bandwidth Broker Implementation-June 2003, [online] available at: http://nrl.cse.unsw.edu.au/ downloads.html [accessed on September 17th 2007].

[109] M. El Barachi, R. Glitho and R. Dssouli, "Developing Applications for Internet Telephony: A Case Study on the Use of Web Services for Conferencing in SIP Networks," *The International Journal of Web Information Systems,* vol. 1, no.3, pp. 147-159, September 2005.

[110] S. Hawwa, "Audio Mixing for Centralized Conferences in a SIP Environment," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, August 2002, pp. 269-272.

[111] OPNET Modeler, [online] available at: http://www.opnet.com/solutions/network_rd/modeler.html [accessed on July 22nd, 2009].

[112] The SIP-IMS OPNET model, [online] available at: https://enterprise1.opnet.com/tsts/4dcgi/MODELS_FullDescription?ModelID=724 [accessed on July 22nd, 2009].

[113] G. Camarillo, "Compressing the Session Initiation Protocol (SIP)", RFC 3486, Internet Engineering Task Force, February 2003.

[114] V. Srivastava and M. Motani, "Cross-Layer Design: A Survey and the Road Ahead", *IEEE Communications Magazine*, vol. 43, no.12, pp. 112-119, December 2005.

[115] G. Aggelou, *Mobile Ad Hoc Networks – from Wireless LANs to 4G Networks*, First edition. USA: McGraw-Hill Companies, Inc., 2005.

[116] Y. Lin and Y. Hsu, "Multihop Cellular: A New Architecture for Wireless Communication", in *Proceedings of the IEEE INFOCOM 2002*, June 2002, pp. 1273-1282.