Presence-Based Integration of Wireless Sensor Network and IP Multimedia Subsystem: Architecture Implementation and Case Studies

Arif Kadiwal

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

November 2008

Your file Votre référence
ISBN: 978-0-494-63319-9
Our file Notre référence
ISBN: 978-0-494-63319-9

ABSTRACT

Presence-Based Integration of Wireless Sensor Network and IP Multimedia Subsystem: Architecture Implementation and Case Studies

Arif Kadiwal

Services are the main driving forces of the Telecommunication industry. In old days the traditional networks e.g. PSTN and mobile cellular networks provide basic services to customers such as voice call, SMS. Unlike mobile cellular networks, Internet provides wide variety of services that allow users to communicate in different ways for example E-mail, WWW, VoIP, and Instant Messaging. To benefit from these Internet (IP) based services and to support new value-added services in mobile cellular networks, the IP Multimedia Subsystem (IMS) is defined. It aims at convergence of Internet and the cellular world. IMS introduced an architectural framework envisioned by telecom experts for providing wide variety of multimedia services for example Presence, Context-based applications, Conferencing, Video-on-Demand, Instant Messaging to name just a few. The main ingredients of these services are data which can be either provided by users or derived from external sources (other networks). In recent years, Wireless Sensor Networks have emerged as networks of tiny devices, sensor nodes. WSNs are characteristically different from existing networks. In general, a WSN is a data oriented network in which sensors sense physical environment and produce data to deliver it to interested applications. These applications are usually external to WSN and reside in an external network (e.g. IMS). The applications or services use this physical data to deliver enhanced services such as context-based services to mobile users. The challenge is to

integrate WSN with IMS so that the WSN data can be accessible to IMS services or applications. This thesis exploits a presence-based approach for the integration of WSN and IMS network. A standard interface for data exchange between WSN and IMS has been devised in the Telecommunication Service Engineering group. The data exchange between WSN and IMS services/applications is realized as a publish-subscribe mechanism. In publish-subscribe mechanism the applications subscribe to WSN services (data) and get notified when sensors report any data while the WSN is acting as publisher to publish sensor data to IMS. The presence-based WSN – IMS architecture provides an abstraction to services and applications for accessing variety of sensed data from different WSNs. The presence based WSN-IMS architecture realizes two main architectural entities, the integrated WSN-IMS gateway and an IMS extended presence server. These two architectural entities play a key role in interworking of WSN and IMS network. The overall architecture has been implemented and tested with prototype applications as well as a performance evaluation has been done to see the efficiency and applicability of the integrated architecture.

Acknowledgements

This thesis evolved as a work done in Telecommunication Service Engineering (TSE) research group. The TSE is a joint research lab between Concordia University and Ericsson Canada Inc. As a member of TSE group, I have worked with a number of people whose guidance, experiences and knowledge making this work possible.

First I would like to express my sincere gratitude to my supervisors Dr. Ferhat Khendek and Dr. Roch Glitho for guiding me at every stage of my research work. Their sound advice, encouragement, criticism and support were vital in putting this work within right scope. I am also thankful to other group members for their ideas, comments and sharing of knowledge, and experiences with me.

I also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and Ericsson Canada Inc. for their financial support.

My personal thanks go to my parents and my family in Pakistan for their strong encouragement and support for pursuing a Master's Degree in Canada.

Arif Kadiwal, November 2008

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

3G	Third Generation
3GPP	3RD Generation Partnership Project
API	Application Programming Interface
AS	Application Server
CS	Circuit Switch domain
CSCF	Call Session Control Function
DSL	Digital Subscriber Line
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HSS	Home Subscriber Server
HTTP	Hyper Text Transport Protocol
I-CSCF	Interrogating CSCF
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFC	Initial Filter Criteria
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISDN	Integrated Service Digital Network
JAIN	Java API for Integrated Networks

MANET	Mobile Ad Hoc Network
P-CSCF	Proxy CSCF
PEA	Presence External Agent
PIDF	Presence Information Data Format
PS	Presence Server
PSTN	Public Switch Telephone Network
PUA	Presence User Agent
QoS	Quality of Service
RFC	Request for Comments
S-CSCF	Serving CSCF
SIP	Session Initiation Protocol
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
TCP	Transport Control Protocol
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VoIP	Voice over Internet Protocol
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Chapter 1 Introduction

1.1 Research Domain

The area of Wireless Sensor Networks (WSN) has emerged as a new research domain in Telecommunications and networking. There is still much extensive research work going on WSN from advancement in hardware technology: small size, low power radio transceiver, multifunctional sensors, to networking and to high-level application development. The advancement in networking includes efficient multi-hop routing, reliable data transfer, and self-organization. While development at the application level include sensor based operating system (e.g. TinyOS), efficient middleware and high-level programming APIs. WSNs are mostly application driven, i.e. they are deployed for specific purpose. Initially, focused on specialized applications but recent advances in micro-electro-mechanical, electronic systems and wireless communication made WSN widespread for commercial applications. The significant application areas include environmental monitoring, medical/health-care monitoring, military surveillance, home automation, and intelligent buildings.

A WSN consists of individual nodes scattered in a field for interacting with the physical environment through the sensing of physical phenomena. The size of wireless sensor network comprises from a few sensor nodes to thousands of nodes. The nodes in a WSN collaborate with each other to achieve their desired tasks. In a conventional Wireless Sensor Networks, sink/gateway collects data from sensors and interface with an external network or applications. Yet another possible scenario in which the sensor network is sink-less i.e. the sensors directly interact with external network or

applications. The potential web service based sink-less WSN approach is realized in [55]. Usually in WSN the gateway is a concentrator responsible for management of sensor network and provides remote access to sensor data from external network (e.g. Internet or IMS). A sensor is an excellent source of contextual data, broadly categorized as spatial (location or space), environmental (temperature, humidity, light etc) and physiological (blood pressure, heart beat, body temperature, glucose level etc).

The mobile telecommunication networks have evolved rapidly in past few years. Currently the 3G mobile system offer greater network access capabilities and services to end-user. A part of 3G mobile network, an IP Multimedia Subsystem (IMS) is an architectural framework specified by 3GPP that enables IP based multimedia services accessible to mobile end-user using standard Internet based protocols e.g. SIP [18]. Examples of multimedia services include presence, instant messaging, enhanced voice and video, pervasive gaming and emergency services.

1.2 Problem Statement and Contribution of this Thesis

In IMS the services use contextual information to deliver a rich set of multimedia services to end-users. WSNs are seen as a source of contextual information that provides different types of information e.g. location, environmental, and physiological. The integration of WSNs with IMS will provide a wide variety of services to end-users. Since the sensor networks are application specific that provides proprietary interfaces to access data. Therefore the motivation is to integrate WSN with IMS so that there should be a standardized mechanism (e.g. protocols and interfaces) to access and report WSN data to IMS. However today there is no such standardized mechanism or solution for such an integration. There has been some research work done and even some working prototypes

have been made towards the integration of WSN and Internet. Hence the focus of this thesis is on the integration of WSN with IMS and discusses the integrated WSN and IMS architecture.

The problem addressed by this thesis is the integration of WSN sensing capabilities with IMS. It focuses mainly on a Presence-based approach for making WSN data accessible to IMS. Presence is a principal IMS service that provides users or entities presence information to other IMS users and services. The thesis also addresses the use of standard protocols (e.g. Session Initial Protocol) and service related extensions for implementation.

The contributions of this thesis:

1. A detail survey and evaluation of existing work for the integration of WSNs with respect to different application areas
2. Implementation of a WSN-IMS gateway, a part of an integrated architecture designed by a PhD student in a research project [20], [28].
3. Experimenting with the architecture through the implementation of two applications, FruitQuest and SenseCall,
4. Analysis of performance measurements based on two prototype applications.

1.3 Organization of the Thesis

Chapter 2 "provides the necessary background information on WSNs, IMS, Presence and SIP.

Chapter 3 provides a detailed review on WSN integration and interworking with other networks. It also evaluates related work.

Chapter 4 presents the detail design and implementation of integrated WSN and IMS architecture. First we will look at the integrated WSN and IMS architecture. Secondly we will describe the architectural design and implementation which is divided into two phases: First phase involve the implementation of integrated WSN and IMS gateway architecture. Second phase involves the IMS presence service specific extensions required for interworking of WSN and IMS. At last we finish the chapter by discussing the overall system implementation.

Chapter 5 presents a proof of concept prototype and its performance evaluation. We selected two prototype applications: pervasive gaming ("Fruit Quest") and the third party call setup application ("SenseCall") for our implementation. A section on performance measurements and analysis is presented at the end of this chapter.

Chapter 6 draws conclusions and discusses potential future work.

Chapter 2 Background Information on WSN, IMS and SIP

This chapter provides the necessary background information on WSN, IMS and SIP.

2.1 Wireless Sensor Networks

2.1.1 Introduction

Sensors are tiny electronic devices that sense physical environment or surrounding to detect and gather data and convert it into electrical signals for processing and transmission to other network entities. The sensor nodes are equipped with onboard multifunctional sensors, processor, and communication components which are low-cost, and low power intelligent devices. Due to the recent advancement in wireless technologies most of the sensor nodes are equipped with wireless interface for inter-node communication. The collaboration among these nodes make up a network of sensors defined as Sensor Network or Wireless Sensor Network (WSN).

A WSN is a network which tasked for sensing, processing, and dissemination of information. WSNs have no infrastructure and are easily deployed without any human intervention or configuration. They can self-organize in a deployed environment. The size of WSNs varies from tens (small) to thousands (large) nodes.

Sensor network is similar to Mobile Ad hoc Network (MANET) however the applications and technical requirements for the two systems are different in several aspects. The following list characterized the WSN from MANET [1], [2] are:

- Sensors are densely deployed in a physical environment

- Data-centric i.e. they do not have any global identification because they sense and report data in large numbers.
- Topology changes occur when failure or malfunctioning of sensor nodes due to lack of power or environmental interference.
- Sensor nodes are limited in resources e.g. power, computation capabilities, and memory.

2.1.2 Network Architecture

The architecture of WSN greatly influences on many factors such as sensing hardware, network topology, and power consumption etc. These factors are well explained in [1].

In wireless sensor network the nodes are scattered in an assigned physical environment, a WSN comprises sensor nodes, sink and gateway. Sensor nodes do sensing and aggregation of data. These nodes are required to send data to a sink node which is acting as a central point of WSN that performs information gathering, processing and mapping at very high-level received from different sensor nodes. The gateway is a central point of access to WSN. It manages the sensor network and acting as an intermediary between WSN and the outside world. The gateway and sink are sometime co-located as single node. These architectural entities are explained in [55]. Figure 2.1 shows the overall architecture of WSN.

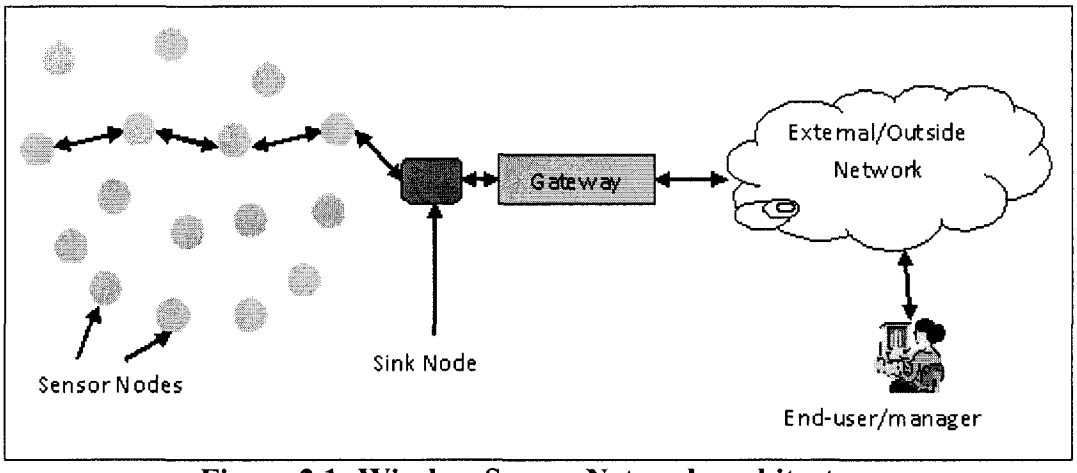


Figure 2.1: Wireless Sensor Network architecture

2.1.3 WSN Hardware

The WSN infrastructure consists of hardware and software components of sensor nodes.

The key hardware components of sensors are:

Sensing Unit: provides basic sensing capabilities e.g. temperature, light, velocity etc. The sensing unit comprises single or multifunctional sensors. The sensing unit also has Analog to Digital converter (ADC) that transforms the sensed signals into electrical signal which can be fed to processing unit for further processing.

Processing Unit: provides processing and transient storage (memory) of sensed data. This unit does onboard local processing and aggregation of data. It also manages the other internal units of sensor node for achieving the desired task.

Communication Unit: provides communication functionality with the neighboring nodes. This unit has a built-in wireless transceiver that transmits and receives at the same time. The transceiver is assigned a particular Radio Frequency (RF) for multi-hop communication within the network.

Power Unit: provides power supply to sensor nodes usually in the form of batteries but there are some energy scavenging schemes which derives power from ambient sources.

There are also some additional application-specific units such as location finding system and actuator unit. Figure 2.2 below shows the general architecture of sensor node.

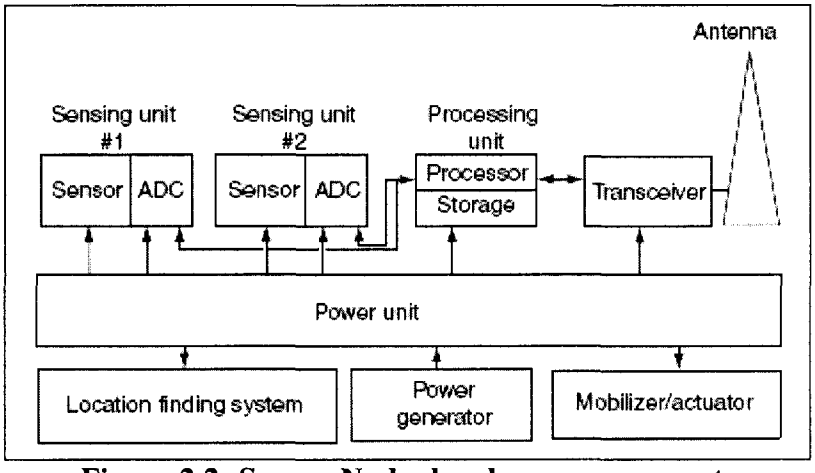


Figure 2.2: Sensor Node: hardware components

The commonly used sensor hardware platform in industry and commercial applications as well as for academic research purposes includes Motes platform (MICA, MICA2, TelosB) from Crossbow originally designed at University of California Berkeley, MIT Crickets for location detection from MIT, TMote-Sky from Moteiv (now Sentilla), Scatterweb is another platform from Freie Universität Berlin and BTnodes with Bluetooth wireless interface developed at ETH Zurich. Figure 2.3 shows some common sensor platforms

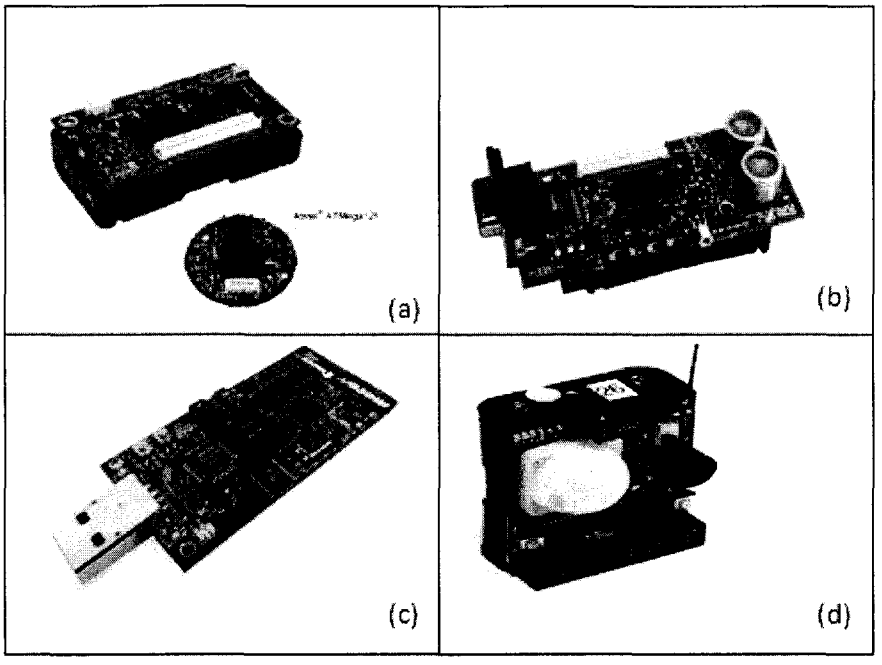


Figure 2.3: Various sensor platforms currently used in industry and academia. (a) Crossbow Motes – MICA2 [3], (b) MIT Cricket [21], (c) TMote-Sky [5], (d) ScatterWeb[4]

2.1.4 WSN Software

In addition to hardware components, sensors have two main software components: Operating System, middleware and applications. From WSN perspective the term operating system and middleware are used in the same context that provides an environment for application development. TinyOS is a standard open-source operating system for WSN that is built on component based architecture. The applications ported with OS provide in-network processing of data and query management for data access e.g. TinyDB, TinyREST. Figure 2.4 realizes software components (subsystem) of sensor node.

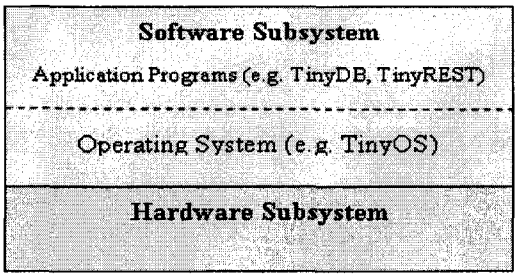


Figure 2.4: Software components of sensor nodes

2.1.5 WSN Applications

WSNs are actually application driven, in the old days the WSN are designed for large-scale applications (e.g. military). But during recent technological advancement and high demand of applications for the deployment of WSN brings in a commercial domain. The common application areas are:

Environmental Applications	Weather monitoring
	Highway Traffic monitoring
	Fire detection
Health Applications	Patient Health monitoring
	Hospital or medical clinic monitoring
Commercial Applications	Pervasive gaming
	Smart Buildings
	Warehouse Inventory tracking
Military Applications	Attack Detection
	Enemy movement

Table 2.1: WSN common application areas

2.2 IP Multimedia Subsystem

2.2.1 IMS Architecture

The Internet provides rich-set of IP based services to end-users in contrast mobile networks adhere to few basic services (e.g. voice) which are today not sufficient for growing customer needs and increased mobile user penetration. "The IP multimedia subsystem (IMS) is defined as a new global, access-independent and standard-based IP connectivity and service control architecture that enables various types of multimedia services to end-users using common Internet-based protocols" [10]. It is a unified architecture where existing and future networks come to one convergence point. It was specified by 3GPP, a main standard body for standardizing 3G mobile communication. The range of services includes enhanced voice services, video call, messaging, presence, gaming, conferencing and push to talk. To support these services many key service features e.g. session management, charging, quality of service (QoS), and roaming, have to be supported as well.

The IMS architecture is modeled into three layers: connectivity layer, control layer and service (application) layer. Figure 2.5 shows the IMS layered architecture.

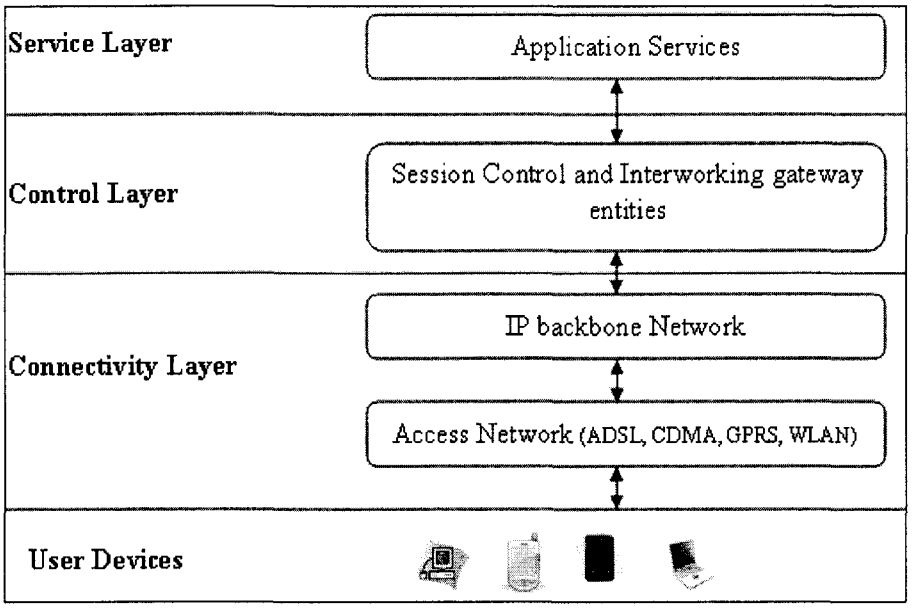


Figure 2.5: IMS Layered Architecture

Connectivity Layer: provides IP connectivity to IP backbone network that is responsible for carrying end user traffic. It connects end-users to the session control and application entities. The connectivity layer supports different network access technologies like GSM (GPRS), WCDMA, DSL, WLAN, and ISDN for accessing IP backbone network.

Control Layer: provides signaling functions in IMS. It consists of number of IMS core network components (e.g. CSCFs) to manage user sessions and perform mobility management, security, charging, and QoS for accessing the IMS services.

Application Layer: provides various IMS services to users. It is a home of deployed IMS services e.g. presence, instant messaging. The application layer contains different application servers to host these services and service capabilities required to access these services.

2.2.2 IMS Architectural entities

IMS is a collection of architectural entities. The functions of these entities are separated into three layers [9], [10]. Figure 2.6 shows these layered architectural entities as discussed below.

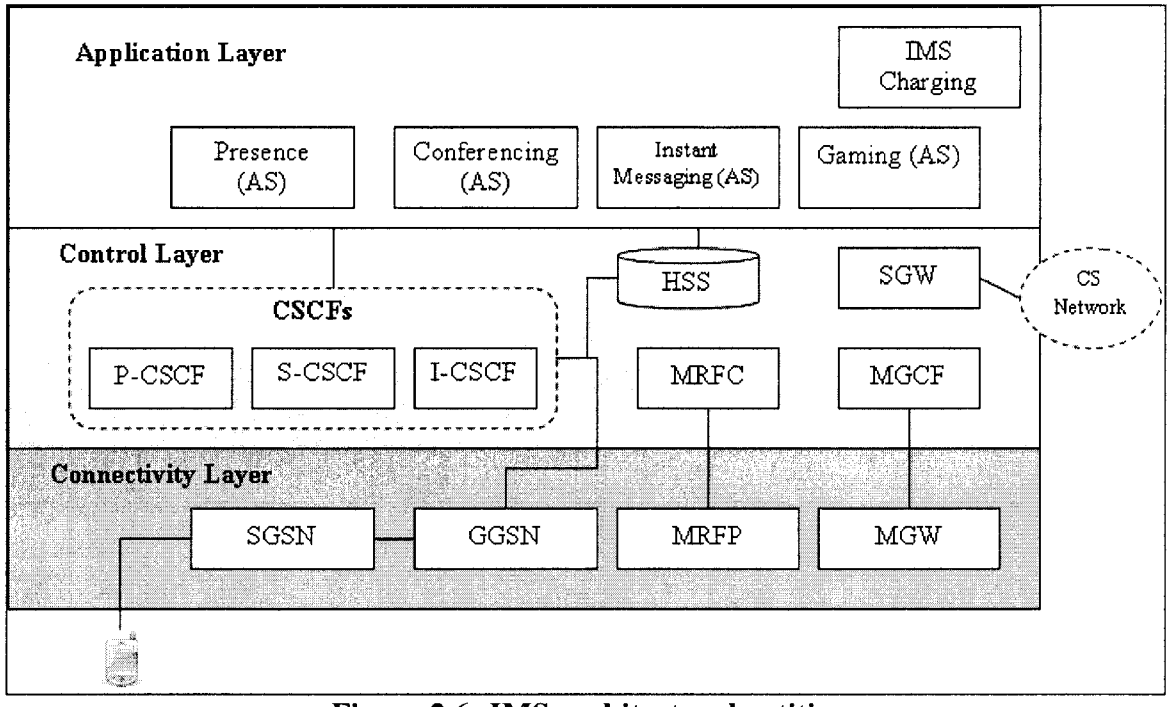


Figure 2.6: IMS architectural entities

GPRS Nodes (SGSN & GGSN): The Serving GPRS Support Node (SGSN) connects the RAN (Radio Access Network) to the packet network. It allows user terminal to access the IMS network. The SGSN is acting as a gateway that forwards user traffic between the user terminal and GGSN.

The Gateway GPRS Support Node (GGSN) provides connectivity to external packet data networks. The external data network could be the IMS or Internet. The GGSN forwards IP packets from user terminal to one of the CSCFs at session control layer that contains

SIP signaling information. It also assigns dynamic IP addresses to user terminals to interact with IMS network.

Media Resource Function Processor (MRFP): The MRFP provides media related resources for mixing media stream, playing announcement, media transcoding and other support functions. The MRFP is controlled by MRFC.

Media Resource Function Controller (MRFC): The MRFC is acting as a controller to manage media resources. It implements a SIP interface to interact with S-CSCF and H.248 interface to interact with MRFP.

Media Gateway (MGW): its basic function is to provide interface to media plane of existing CS network (e.g. PSTN). It does media translation from IMS (RTP) to CS (PCM) and from CS to IMS domain.

Media Gateway Control Functions: The MGCF is a core node for interworking between IMS and CS (PSTN) domain. It converts SIP signaling to ISDN User Part (ISUP), or Bearer Independent Call Control (BICC). It uses SGW (Signaling Gateway) to send the converted request to CS domain.

Signaling Gateway (SGW): The SGW provides interface to the signaling plane of CS network (PSTN). The SGW mainly performs signaling conversion at the transport level i.e. from IP based transport to SS7 based transport for CS domain and vice versa.

Call Session Control Functions (CSCF): The CSCFs which are actually SIP proxy servers and are core entities of IMS network that control all the signaling traffic of IMS. The CSCFs are categorized into three types based on their roles; Proxy-CSCF (P-CSCF), Serving CSCF (S-CSCF), and Interrogating CSCF (I-CSCF).

P-CSCF is the first point of access to the IMS that implies all traffic from and to users mobile devices have to pass through P-CSCF. It is nothing more than just an outbound proxy that relays every request from and to user's mobile device.

I-CSCF is a type of SIP proxy which resides on the boundary of operator's domain. Its main function is to find next designated hop (e.g. S-CSCF or AS) to fulfill the incoming SIP request by querying the HSS. It also performs a selection of assigned S-CSCF for user based on information provided by HSS. This process occurs when the user registering to IMS network.

S-CSCF is a vital node for session control in IMS it handles IMS registration, maintaining session states, routing SIP request, stores user's service profile downloaded from HSS. The S-CSCF acts as registrar to store user's location information during IMS registration. It interacts with different ASs to trigger services requested by users.

Home Subscriber Server (HSS): HSS is a database that store subscription data associated with IMS user. The subscribed data includes user identities, IMS authentication and authorization vectors, and service profile.

Application Server: The application server is an IMS SIP application server that hosts range of IMS services. The AS interface with S-CSCF via SIP protocol to render a particular service requested by user. From a deployment point of view the AS can host more than one service.

2.3 Session Initial Protocol

Session Initiation Protocol (SIP) is an IETF standard signaling protocol for establishing and terminating multimedia session. SIP was originally designed to support basic audio/video communication over internet but later it is extended to support different

multimedia services e.g. presence and instant messaging. SIP is a transport independent protocol i.e. it can use any underlying transport protocols e.g. TCP, UDP, and SCTP for message transport. The RFC 3261 [18] defines a SIP session is a dialog between two end-users which is usually created via INVITE request.

2.3.1 SIP Entities

Like other Internet protocols (e.g. HTTP), SIP is a client/server text based protocol used to establish and terminate session between two endpoints. SIP consists of following entities as defined in [18].

SIP User Agents: An entity that exchanges messages to establish and terminate sessions. The user agent can be of two types user agent client (UAC) and user agent server (UAS). The UAC sends request to the server (UAS) while UAS receives and serves a client request and generates response accordingly. There is also a service specific user agent like Presence Agent (Presence Server) which manages subscription and publication of presence information.

Proxy Servers: The proxy server is a SIP server that locates the destination user and routes the SIP requests toward the destination as shown in Fig 2.7. It also authenticates and authorizes access to network resources. There are two kinds of proxies stateful proxy and stateless proxy. The stateful proxy maintains the record of each transaction going through this proxy while stateless proxy does not have this feature at all.

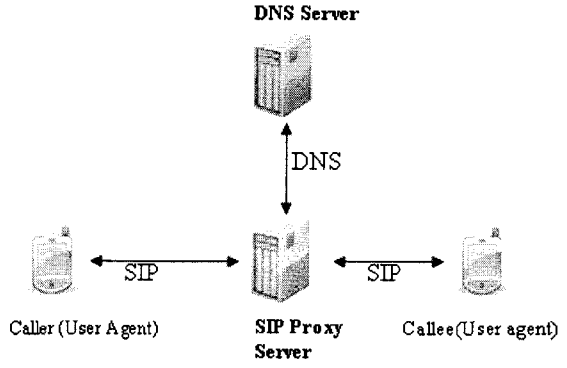


Figure 2.7: Routing request via SIP proxy server

Redirect Servers: the function of redirect server is to send redirect requests to clients by sending 3xx responses to client. In the response, it directs the client to send new request with new URI address as shown in Figure 2.8.

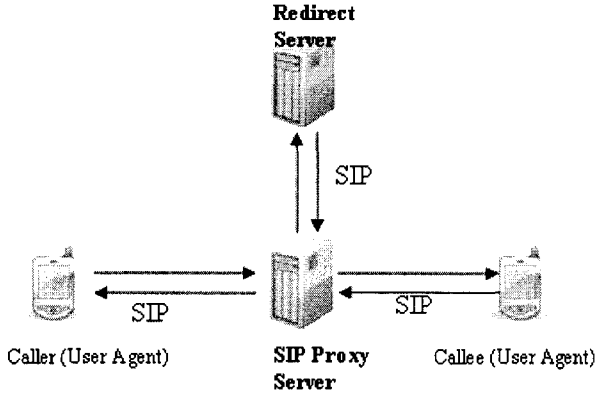


Figure 2.8: Redirecting SIP request via SIP redirect server

Registrar: accepts user's registration via SIP REGISTER request. During registration it stores users contact (location) information embedded in the REGISTER request. In IMS, the S-CSCF is acting as registrar.

2.3.2 SIP Request and Response Messages

The SIP request messages consists of set of methods to manage multimedia session and request desired SIP services, for example INVITE request(method) is used to establish a session between two end-users. Table 2.2 shows list of SIP request messages.

SIP Methods	Description
INVITE	Establishes a media session between end-users
ACK	Acknowledges the final response of media session
BYE	Terminate an existing media session
SUBSCRIBE	Subscribe to a resource (event)
NOTIFY	Notifies about subscribed resource (event)
PUBLISH	Publish info about resource (event)
MESSAGE	Send IM messages to other clients
REGISTER	To register a user to the network (Registrar)
OPTIONS	Query capabilities of SIP user agents

Table 2.2: SIP request messages

SIP response message is generated either by a SIP user agent or server in response to a client request. The type of response may vary depending on request; response may contain some additional headers and parameters for particular request during session negotiation. The response messages are classified into six types as shown in Table 2.3

Response Codes	Response Type
1xx	Provisional and Informational responses
2xx	Success response

3xx	Redirection responses
4xx	client responses (client errors)
5xx	Server responses (server errors)
6xx	Global failure (e.g. client busy or refused to take request)

Table 2.3: SIP response messages

2.4 IMS Operations

3GPP selected Session Initial Protocol (SIP) as a core signaling protocol for different IMS operations e.g. registration, and session establishment. This section describes some of these operations.

2.4.1 IMS Registration

Before registering to IMS the user terminal has to perform two basic operations i.e. to get IP connectivity and discover P-CSCF, an IMS access point. Once the P-CSCF discovery is completed the user register with IMS by initiating the SIP registration in the form of SIP REGISTER request. Every user in IMS has to register with IMS network before accessing IMS services. The 3GPP specified a basic IMS registration phase involves two rounds. The first round is the selection of S-CSCF for user and the second round is the authentication of user i.e. the S-CSCF authenticates the users, downloads the user's service profile and sends the final response to initial registration. The final response of registration contains an interval (expire time) during which a user is allowed to use an IMS network otherwise user renews its registration to extend this interval. This whole process involves a series of steps detailed in 3GPP specification TS 24.228 [32]. Figure 2.9 shows the simplified IMS registration process.

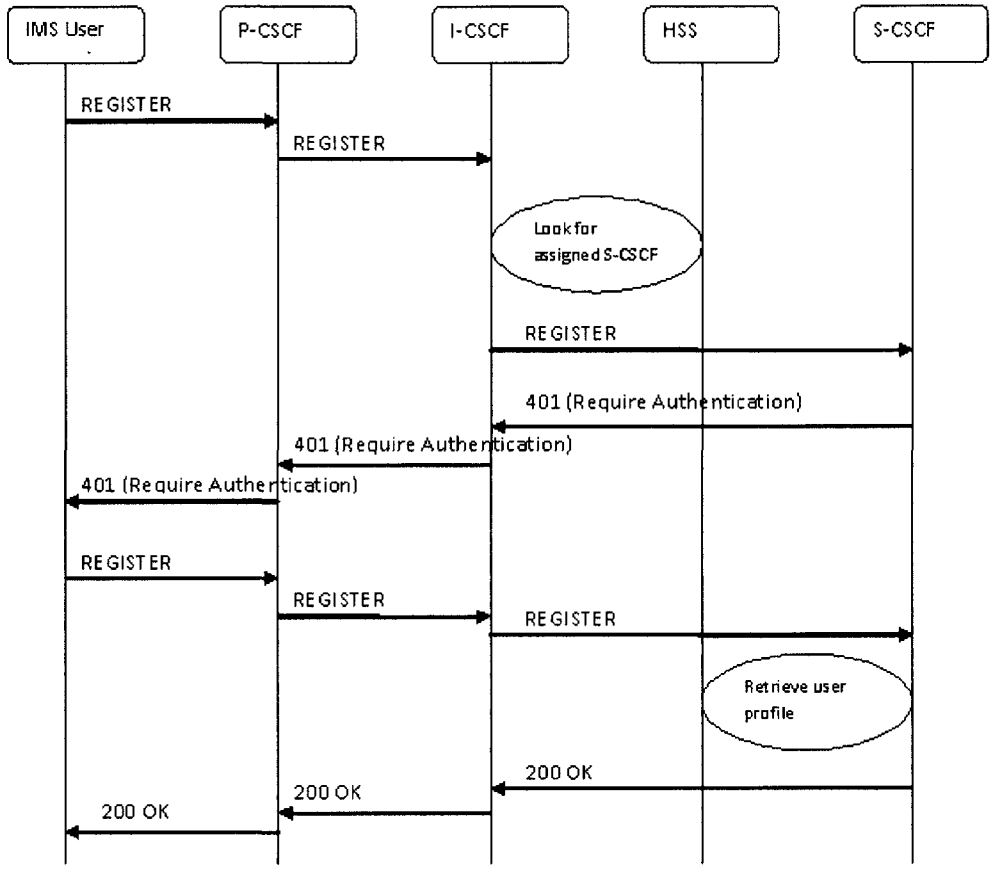


Figure 2.9: IMS Registration Flow

The main purpose of IMS registration is to register user's Public Identity that is in the form of Public URI and authorize user to use subscribed IMS services. The public URI is basically a SIP URI or Tel URI shown in figure 2.10. During the registration phase the user's public identity binds to its contact address which is a hostname or IP address associated with user's terminal. The user can register multiple contact addresses associated with one public identity.

SIP or SIPS URI	**sip: user@domain** **sips:user@domain** **e.g. alice@ericsson.com**
Tel URI	**Tel :< phone number >** **Phone number: local or global (+)**

Figure 2.10: Public URI format – SIP and Tel URI

2.4.2 Session Management in IMS

SIP is a signaling protocol for IMS which deals with session handling between end-users. The session contains information of point-to-point (two-party) or point-to-multipoint (multiparty) relation between users. This section discusses the session management scenarios; basic session setup (voice) and service specific session.

Basic Session Setup: the basic session setup here considered as a normal voice session between two users. Let's take an example scenario in which user X wants to establish a session with user Y. The User X creates a SIP INVITE request and sends it to the P-CSCF. The P-CSCF processes the request, authorizes user's identity and forwards it to S-CSCF. The S-CSCF further processes the request, trigger service control functions based on user's service profile that involves interacting with Application Server if needed, and determines the user Y network domain entry point i.e. the address of I-CSCF of user Y domain. S-CSCF(X) then forwards request to I-CSCF(Y). The I-CSCF(Y) processes the SIP INVITE request and forwards the request to S-CSCF(Y). The S-CSCF(Y) processes the request and handed it to P-CSCF(Y), the P-CSCF(Y) terminates the session by dropping request to user's Y terminal. After processing the request (SIP INVITE) by user Y, it generates a series of SIP responses including the final response (200 OK) which

sends back to user X taking the same route as the request was traversed. User X in return sends ACK response to confirm the session establishment. Once the ACK response from user X is received by user Y the session is established between the two users (X and Y). Figure 2.11 shows the detail flow of basic session setup between two users.

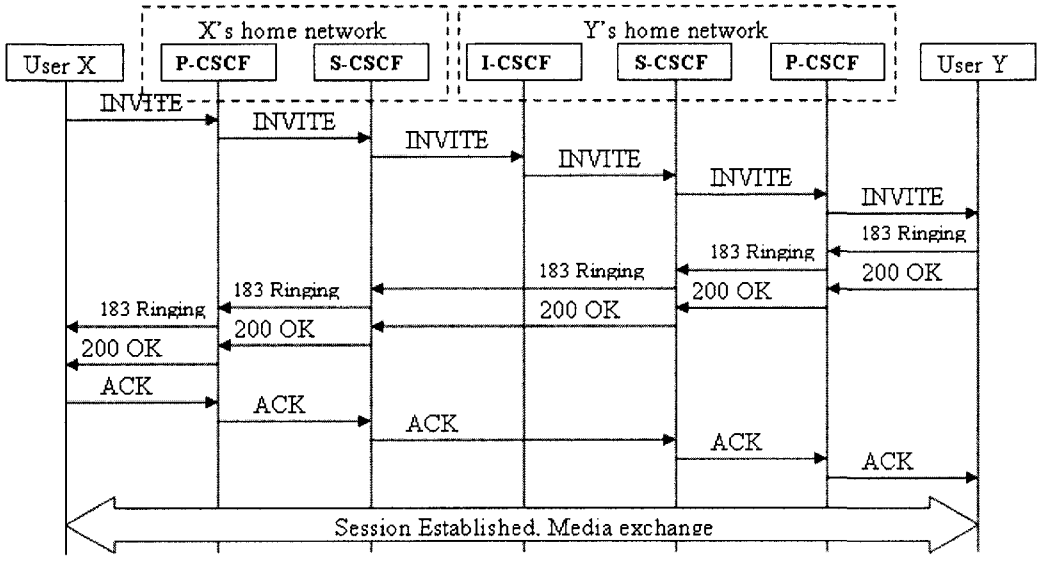


Figure 2.11: Basic session flow

Service Specific Session: the basic session setup discussed previously offered basic voice call session between two IMS users where there are no such services (e.g. presence, instant messaging) invoked other than voice call. This section provides example on establishing service specific session based on enhanced services subscribed to IMS users. These services are hosted in one of the Application Severs that provides a particular service to IMS users.

Figure 2.10 shows the scenario in which user X is accessing presence service. The user X first sends SIP SUBSCRIBE request to P-CSCF which forwards to S-CSCF. The S-CSCF verifies the request and sends it to a particular AS by authorizing the type of SIP request. The AS processes the request and sends back response to user X. The decision of sending

SIP request to a particular AS based on set of conditions defined in an Initial Filter Criteria (IFC) associated with each user's profile discussed in next section. Figure 2.12 shows the flow of accessing IMS service.

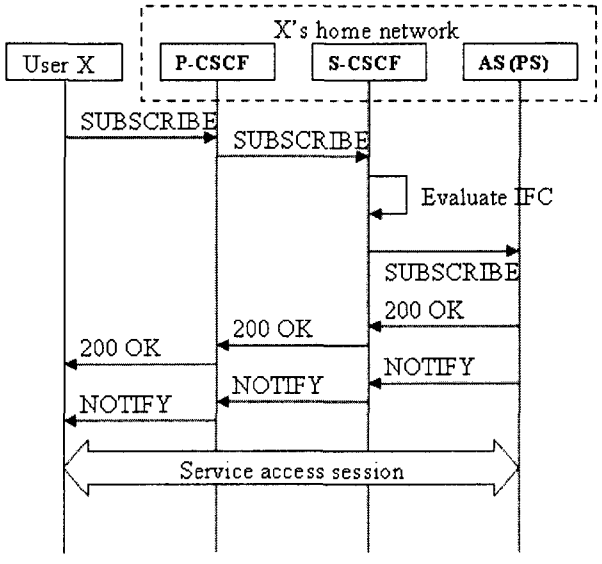


Figure 2.12: Service specific session flow

2.4.3 IMS User Profile

A user profile contains IMS subscription information as agreed by user and service provider. User profile consists of at least one private user identity, and one or more service profile. The private user identity is a secret key for user authentication which is not accessible to outside world. The figure 2.13 shows the general structure of user profile as specified in [30].

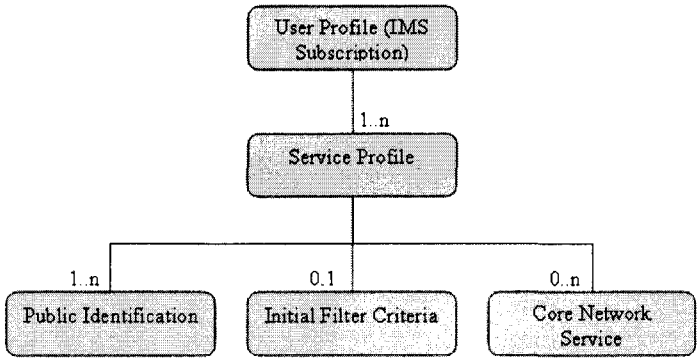


Figure 2.13: IMS user profile structure

A service profile contains information related to user's services and resides in HSS. The S-CSCF accesses this information related to each user during the registration phase. A service profile is represented as an Extensible Markup Language (XML) shown in Figure 2.15. The IMS service profile consists of following components:

- Public Identification
- Core Network Service Authorization
- Initial Filter Criteria

Public Identification: each service profile contains a set of public identities assigned to particular service profile. User can have multiple public identities to render different set of services based on different public identities. The public identity can be in the form of either SIP URI or TEL URI.

Core Network Service Authorization: provides information related to media policy. It contains at most one instance of subscribed media profile associated with a user and each media profile is uniquely identified with media profile Id of type Integer in S-CSCF.

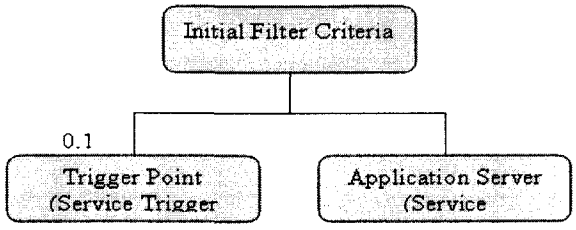


Figure 2.14: Initial Filter Criteria (IFC) structure

Initial Filter Criteria (IFC): is an important part of service profile that contains sets of conditions in the form of Trigger Points. IFC is checked when any incoming SIP request needs to access an IMS service that results in forwarding a service request to a designated AS. Figure 2.14 realizes the IFC structure. The IFC contains the following information

- Trigger Point
- Application Server

Trigger Point describes conditions that should be verified to render a particular service. It consists of set of service trigger points which determine whether the specific SIP request is sent to the designated AS. The service triggers contain multiple fields e.g. SIP URI, SIP Method, SIP header, Session Case etc. The service triggers are coupled by means of logical expressions (OR, AND, NOT).

Application Server provides information related to AS e.g. AS address (URI). If any of the Triggers are satisfied that AS will be selected to serve a request. Also the default handling field assumes the default action in case if the selected AS is unreachable.

```
<?xml version="1.0" encoding="UTF-8"?>
<IMSSubscription xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\ \CxDataType.xsd">
 <PrivateID>privateid.user1@ericsson.com</PrivateID>
 <ServiceProfile>
       <PublicIdentity>
             <BarringIndication>1</BarringIndication>
             <Identity> sip:user1@ericsson.com </Identity>
       </PublicIdentity>
       <InitialFilterCriteria>
             <Priority>0</Priority>
             <TriggerPoint>
                   <ConditionTypeCNF>1</ConditionTypeCNF>
                   <SPT>
                         <ConditionNegated>0</ConditionNegated>
                         <Group>0</Group>
                         <Method>PUBLISH</Method>
                   </SPT>
                   <SPT>
                         <ConditionNegated>1</ConditionNegated>
                         <Group>0</Group>
                         <Method>SUBSCRIBE</Method>
                   </SPT>
                   <SPT>
                         <ConditionNegated>1</ConditionNegated>
                         <Group>0</Group>
                         <Method>MESSAGE</Method>
                   </SPT>
             </TriggerPoint>
             <ApplicationServer>
                   <ServerName>sip:simpleAS@ericsson.com</ServerName>
                   <DefaultHandling>0</DefaultHandling>
             </ApplicationServer>
       </InitialFilterCriteria>
 </ServiceProfile>
</IMSSubscription>
```

Figure 2.15: Service Profile in XML format

2.5 3GPP Presence Framework

2.5.1 Introduction

Presence means the current status of users i.e. a user is available and willing to communicate for current instant of time. This status information is delivered as presence information which contains the following attributes: user's status (e.g. online or offline), communication preference (e.g. voice, SMS, email), contact addresses, and location information (e.g. geographical coordinates or civic location).

2.5.2 Presence Service

A service that provides collection and dissemination of user's presence information, it controls access to user's (presentity) presence information among watchers (subscribers of user's presence). The applications and other services consume presence information supplied by presence service to provide wide range of services to end-user e.g. the presence service may use presence and contextual information to enable context-aware services. Figure 2.16 shows the presence service model as specified by 3GPP which deals with two major entities: presentity and watcher.

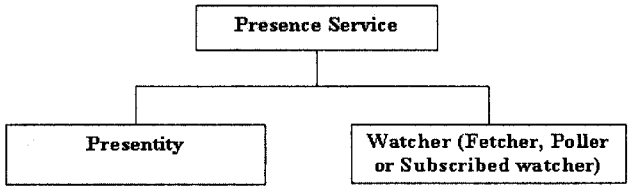


Figure 2.16: IMS Presence Service model

Presentity: an entity (user) that provides presence information to presence service. It controls access of its presence information by defining access rules in presence service.

Watcher: an entity (user or application) that requests presence information of a presentity via presence service. There are two types of watchers: subscribed-watchers and fetchers. Those watchers who request presence information only once when needed are described as fetcher also there is a special type of fetcher called poller who polls presence information on regular interval basis. Unlike fetcher, subscribed-watcher subscribes to presence information to get subsequent notification when the presentity's presence information changes.

2.5.3 3GPP Presence Service Architecture

3GPP opted for the IETF presence model as specified in RFC 3856 [17] and RFC 3903 [25]. Figure 2.17 shows the 3GPP presence service functional architecture for IMS environment. The architecture depicts the functional entities and their roles in 3GPP IMS Presence framework.

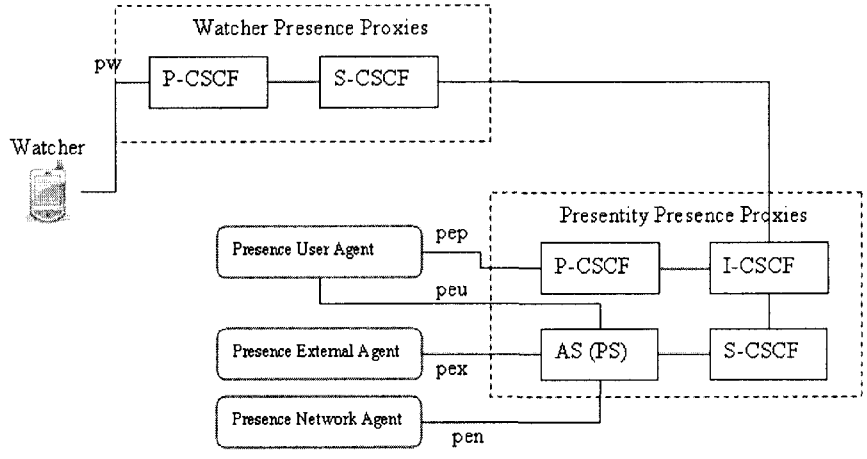


Figure 2.17: Presence Service functional architecture

2.5.3.1 Presence Server (PS)

The Presence Server (PS) hosts the presence service that manages the publication and notification of presence information published by Presence User Agent (PUA), Presence External Agent (PEA). PS gathers and consolidates the presence information from multiple presence user agents of a presentity. PS allows watchers to subscribe to full or partial presence information of presentity. It stores access policies related to subscription and control access to presence information of presentities. PS exchanges SIP messages related to publication (PUBLISH), subscription (SUBSCRIBE) and notification (NOTIFY) of presence information.

2.5.3.2 Presence User Agent (PUA)

The PUA manipulates and publishes presence information of a presentity. It merges information from different sources (e.g. user terminals) and publishes it into one unified document format (PIDF). The PUA uses 'Pep' interface that is a SIP interface for information publication.

2.5.3.3 Presence External Agent (PEA)

The PEA supplies presence information from external network (e.g. alarm services, other non-IMS presence services, and WSN) to PS. PEA acts as an interworking node between PS and external network. As shown in Figure 2.17 PEA uses a 'pex' interface to publish presence information to PS. The 'pex' interface supports information transfer without regard to size limitation. Currently there is no such standard interface (e.g protocol) defined for information exchange on 'pex' interface we choose a SIP protocol for the information exchange on this interface discussed in chapter 4.

2.5.3.4 Watcher

The watcher or watcher application could be a user or service that subscribes to presentity presence's information. PS has to authorize a watcher before accepting any subscription. A watcher subscription request is routed through watcher presence proxies (CSCFs) to intended presence server as shown in Figure 2.17.

2.5.3.5 Presence Proxies (Presentity & Watcher)

The presence proxies provide routing of presence service request to presence server and vice versa. The presence proxies support other features like security, selection of PS of presentity etc.

2.5.4 Presence Information Data Format

RFC 3863 [41] specified Presence Information Data Format (PIDF) that is an XML data format to represent user's presence information. The PIDF can be transported through any compliant protocol that carries XML data (e.g. SIP). In PIDF the user's presence information contains the following items:

- Presentity URI: the address of presentity whose presence information is being reported.
- PRESENCE TUPLE: tuple contains status and user communication information. It has some additional or extended information like location of user (e.g. office or home), mood of user (e.g. busy, away).
- SATUS: provides user`s current status, the possible values are open and close. Open implies user intention to communicate whereas close implies unavailability of user.
- COMMUNICATION ADDREESS: provides user contact address like how user can be reached.

Some existing PIDF extension have been defined by IETF based on different applications but our work is based on the extension of basic PIDF model to support WSN data mapping to PIDF discussed in chapter 4. Figure 2.18 shows the basic PIDF format.

```
<?xml version="1.0" encoding="UTF-8"?>
 <presence xmlns="urn:ietf:params:xml:ns:pidf"
   entity="pres:player1@ericsson.com">
   <tuple id="tinfo1">
     <status>
       <basic>open</basic>
     </status>
     <contact>sip:player1@ericsson.com</contact>
     <!— extension tags -->
     </tuple>
   </presence>
```

Figure 2.18: PIDF data format

2.5.5 Publish-Subscribe System

The Publish-Subscribe is a distributed system consists of different entities like subscriber who subscribes to an event of interest while publisher publishes information related to an event. There is also an entity called event manager who manages different events. The publisher produces new information and publishes it to event manager. When the event manager receives new information from publisher related to particular event that information will be reported to a subscriber in the form of notification. The figure 2.19 shows general view of publish-subscribe system.

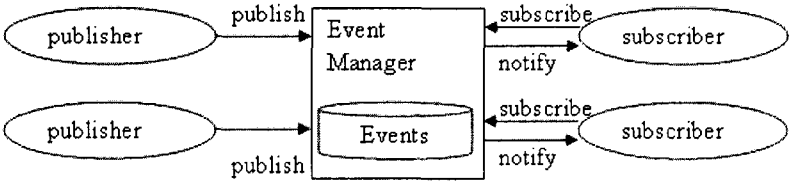


Figure 2.19: General view of Publish-Subscribe System

Similarly the presence service is based on publish subscribe system. The presence service uses SIP protocol for implementing publish-subscribe system. SIP is extended to support presence and instant messaging applications in the form of SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE). SIMPLE introduces some SIP extension

methods (SUBSCRIBE, NOTIFY, MESSAGE and PUBLISH) and presence event notification framework [17], [25] for publication and subscription of presence information as shown in figure 2.20.

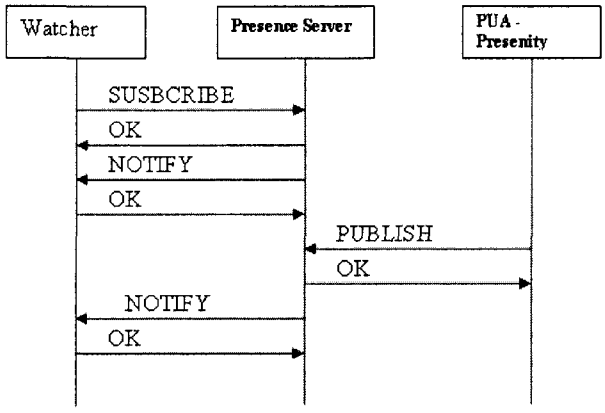


Figure 2.20: Presence Service (publish – subscribe) information flow.

2.6 Chapter Summary

In this chapter we discussed about WSN and its architecture as well as its software and hardware components. Later we looked at the IP multimedia subsystem (IMS) architecture and more importantly the 3GPP presence framework as well as the Session Initiation Protocol (SIP). SIP is a core signaling protocol in IMS to support different IMS operations (e.g. registration, session establishment). In the next chapter we will look into the state of the art work for the integration of WSN with other (existing) networks.

Chapter 3 WSN Integration with existing networks: State-of-the-Art

This chapter describes the integration of Wireless Sensor Network with other (existing) networks and focuses on the evaluation of state-of-the-art by first setting the evaluation criteria. We end this chapter with a summary of evaluation of related work.

3.1 Wireless Sensor Network integration with existing networks

WSN integration can be described as the seamless and coherent integration of WSN with other (existing) networks e.g. Internet or 3G mobile networks, etc. A WSN is an excellent source of ambient information. The ambient information considered as physical context contains bulk of contextual data (e.g. temperature, location, humidity etc). But the challenge is how such information is to be exchange consistently with existing networks (e.g. IMS or Internet). Moreover, emerging applications and services require ambient information to be available to external network in order to support wide range of services by utilizing ambient information. As illustrated in [38], the integration of WSN with existing network may be based on two approaches: one is gateway-based and another one is overlay-based. In a gateway-based approach, every sensor node in WSN has a proprietary interface to relay data to a gateway. The gateway implements different sets of protocol and information mappings in the form of standardized interface (e.g. HTTP, and SIP) accessible to any user or application in an external network. For overlay-based approach, every sensor or some selected sensors in WSN implement a compatible protocol stack (e.g. IP, HTTP, and SIP) of an external network; So that the host in the

external network can communicate directly with sensors. The sensor nodes that implement an external network protocol stack are called overlay nodes and form an overlay-network.

The later sections detail the evaluation of state-of-the-art with respect to the following criteria.

3.2 Evaluation Criteria

We set the criteria for the integration of WSNs with IMS. We have introduced the following criteria and these criteria are discussed in [28]:

Criterion 1: The approach should support all possible WSN sensing capabilities in IMS. The sensing capabilities refer to different types of sensed data (spatial, environmental, and physiological) accessible to IMS services.

Criterion 2: The approach should allow WSN data to be exchanged in a standard IMS format (PIDF) [41] and also support the aggregation of data based on individual or group of sensors.

Criterion 3: Support of information management of different types of physical world entities (e.g. persons, places, objects etc.) in IMS. Currently IMS only support user (person) as a subscribed entity. The WSN can provide information related to different user entities e.g. what's the current: location of a 'user A' (person), temperature of a 'corridor' (place), and spot of a 'car' (object) in parking lot.

Criterion 4: Support the standard publish-subscribe mechanism (e.g. SIMPLE) to exchange WSN data in IMS. This criterion suggests the preferred way to exchange WSN

information is via publication to IMS so that services or application users can have easy access to WSN data.

Criterion 5: Support of standard IMS communication protocols (e.g. SIP) when interacting with IMS network.

3.3 Evaluation of Related Work

This section highlights the evaluation of existing work related to integration of WSNs with other networks

3.3.1 IP-enabled Wireless Sensor Networks

The work discussed in [24] realized the integration of WSN with Internet by enabling an IPv6 stack in sensor nodes. IP-enabled WSN requires implementing an IP stack in different sensor nodes to allow communication with any host in the Internet. It adapts IP networking to WSN by supporting different functionalities like sensor link layer technologies (e.g. IEEE 802.15.4 - Zigbee), auto-configuration of sensors nodes i.e. dynamic IP address assignment, service discovery, and security for encryption, authentication, and data integrity.

This architecture includes a gateway acting as a router (e.g. default gateway) and provides multiple link access technologies (e.g. Zigbee, WLAN, GPRS etc) that enable global connectivity of sensor nodes with Internet hosts. The gateway also assigns IP addresses to different sensor nodes. Figure 3.1 shows the network architecture of IP-enabled WSN.

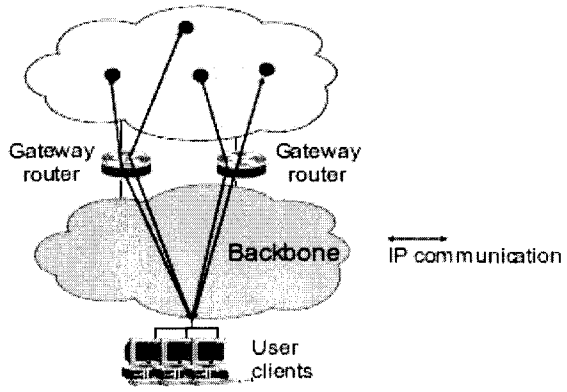


Figure 3.1: IP-enabled WSN architecture

Although the approach focused on IP based integration of WSN with Internet, neither the gateway nor the sensor node provide the standard application interface i.e. which application protocol (e.g. HTTP or SIP) will carry the WSN information. It only supports the sensor based application software running on top of IPv6 layer i.e. the application directly interfaces with an IP layer. Sensor application software would be a process that collects data from sensors, stores in database while server process reports data to Internet clients. Unlike, Internet (TCP/IP) protocols stack the application layer protocols (e.g. SIP or HTTP) do not interface directly with IP layer so in this case the Internet standard application layer protocol may not be supported. In addition to that there are no such support of different user entities (e.g. persons, places, and objects) and the standard format (PIDF) of information exchange with IMS.

3.3.2 E-Sense Project

The e-Sense [39] provides an architecture that allows WSN ambient information of users and service related objects accessible to B3G mobile networks. It developed an e-Sense protocol stack for integration and interworking of WSN and B3G mobile networks as shown in Figure 3.2. Every node in the WSN either sensor or gateway implements an e-SENSE protocol stack [39]. The protocol stack consists of different components to

perform different functions. The sensor gateway in WSN provides connectivity to B3G mobile network [40]. The components (subsystems) of the protocol stack are as follows:

- Connectivity Subsystem: provides functions performed at physical, MAC, network and transport layer. The Physical layer functions operate the e-SENSE radio transceiver. The MAC layer functions control access to radio channel by using multiple access protocol scheme and reliable transmission of frames. The network layer functions include mechanisms to join and leave network, message routing, security and reliability of message transfer etc.
- Middleware Subsystem: provides functionality for exchanging application related data (WSN data) messages. This subsystem implements publish-subscribe mechanism for data exchange.
- Management Subsystem: performs tasks related to configuration and initialization of connectivity and middleware subsystem. It manages different sensor nodes by defining their roles (e.g. sensor, actuator or sink). With respect to application profiles the management subsystem also provides and implements functions like service discovery, node discovery, security etc.
- Application Subsystem: supports various sensor based applications that access services provided by middleware subsystem to exchange WSN data between application entities(servers) in B3G mobile network

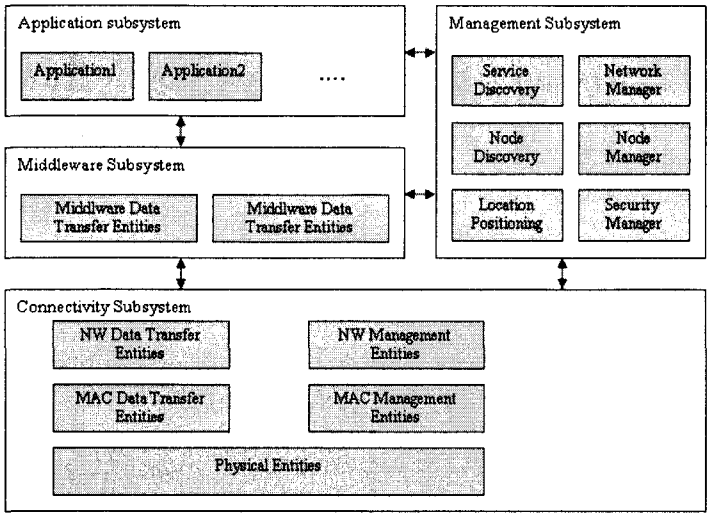


Figure 3.2: E-Sense protocol stack

The e-Sense approach provides a service enabling environment that makes WSN ambient information accessible to B3G mobile services. It proposes a possible way to integrate WSN with IMS via Generic User Profile (GUP) [57]. The GUP is a collection of user related data (e.g. contextual data) stored and manages by UE and the network provider. However the e-Sense did not address issues like support of different user entities (e.g. persons, places, and objects) and the standard format (PIDF) of information exchange.

3.3.3 Alarm-Net

Alarm-net [19] a WSN based health-care monitoring system for assisted living and residential monitoring. It is a heterogeneous architecture that integrates different sensing devices with IP networks. The Alarm-Net supports a query based protocol for data collection and analysis. It provides different types of sensing capabilities e.g. physiological sensors measure patient's vital sign (e.g. heart rate, blood pressure etc) while physical environment is monitored by environmental sensors. The physiological

sensors are attached with patient's body while the environmental sensors are deployed in patient surroundings to monitor patient living environment. The network architecture of Alarm-Net is a hierarchal sensor network consisting of body sensor network (physiological sensors), emplaced sensors are devices deployed in living space to sense environmental quality or conditions (e.g. temperature, dust, motion, light) and AlarmGate (Wireless Sensor Network gateway) as realized in figure 3.3. The emplaced sensors communicate with body networks to gather and report data based on user queries. Additionally emplaced sensors communicate with AlarmGate acting as a sensor gateway to provide sensor data accessible to users (e.g. doctors or nurses). The AlarmGate is similar to an application level gateway between the sensor network and IP network. The gateway host applications to provide aggregation and analysis of sensor data. Each sensor in Alarm-Net implements a query processor stack to facilitate in-network processing of users queries and report the results back to clients. Figure 3.3 below shows the Alarm-Net network architecture.

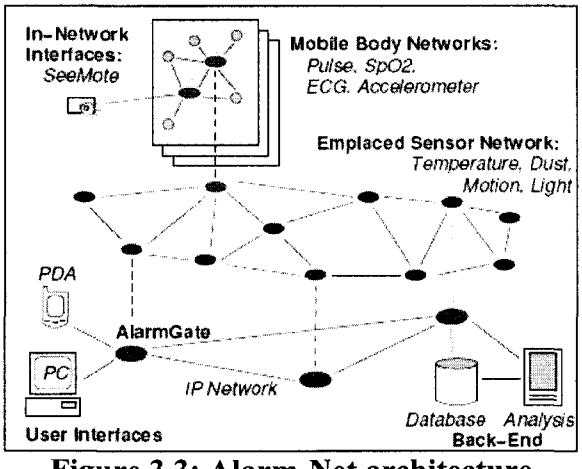


Figure 3.3: Alarm-Net architecture

This approach supports different WSN sensing capabilities (physiological and environmental). It also provides the aggregation of data from different sensor nodes. It uses a query based protocol for exchanging WSN data. However the format (PIDF) of information exchange is not standardized and the gateway does not support a standardized application interface (e.g. SIP or HTTP) for querying sensor network.

3.3.4 TinySIP

TinySIP as discussed in [26], is an another approach that provides a communication abstraction for accessing sensor-based services from existing wired or wireless networks. TinySIP is actually based on SIP. Since the sensor nodes are very resource constrained, TinySIP is a light-weight and a subset version of SIP implemented on each sensor node to allow users to access services provided by WSN. In TinySIP, the gateway is installed to map actual SIP messages to TinySIP messages that are relayed to sensor nodes for processing. TinySIP aims at providing different messaging abstraction to access different WSN services. The messaging abstractions includes: Short Instant Messaging used to configure or perform any task on sensor network. Long session-based messaging is used to establish a session based communication by sending a SIP INVITE message to one or multiple sensor nodes to collect a large data stream. Publish-subscribe messaging abstraction is based on events i.e. users subscribe to events and get notified of events occurring in WSN. A sensor publishes event data via TINYSIP-PUBLISH request while the gateway sends notifications via NOTIFY request to users subscribed to events. Figure 3.4 shows TinySIP message flow.

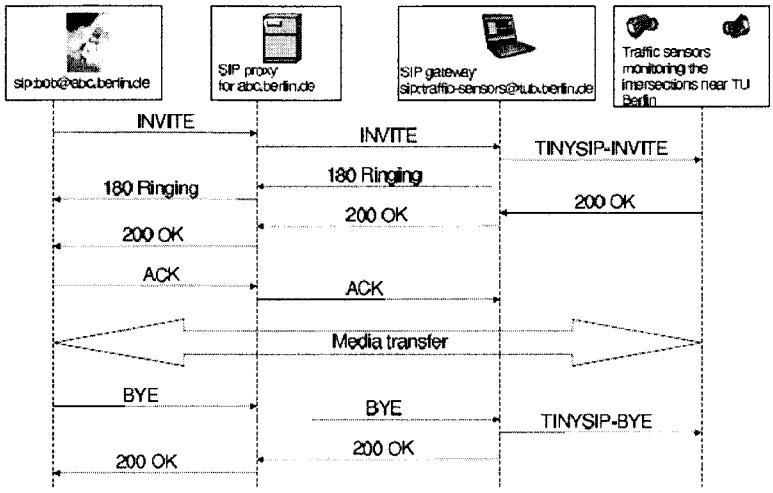


Figure 3.4: TinySIP message flow.

TinySIP approach provides a communication abstraction based on SIP that supports a remote messaging mechanism to remotely access a sensor network. It supports subset features of SIP and does one to one mapping of TinySIP messages with actual SIP messages. However, it lacks support for different types of user entities (persons, places, and objects). In addition to that it does not support the standard format (PIDF) of information exchange as well as the aggregation of sensor data from individual or group of sensors. This approach requires implementing TinySIP protocol on individual sensor nodes considering the fact that sensor nodes in WSN are resource constrained in terms of computation and communication capabilities.

3.3.5 TinyREST

TinyREST [23] is an HTTP based application level approach to integrate WSN with Internet. The aim of TinyREST is to support the development of internet based WSN applications. HTTP is chosen as means for information exchange between sensor network and Internet. The client on Internet issues an HTTP request to access a resource that could be a sensor or actuator. An HTTP request is mapped to TinyOS messages and

delivered to sensors. The TinyREST gateway acts as a middleware and performs mapping of HTTP messages to TinyOS messages and vice versa. HTTP requests supported are: POST used to send commands to control sensors and GET used to collect sensor data. The SUBSCRIBE request is used by the clients to register their interest in specific events and get notifications (NOTIFY) depending on events occurrence. In TinyREST the addressing of sensors is taken care by Home service framework hosted as Home Server which maps sensor or group Ids to locations on location manager. The TinyREST gateway registers with the Home Services Framework (HSF) to receive a HTTP request from clients sent to a particular sensor network.

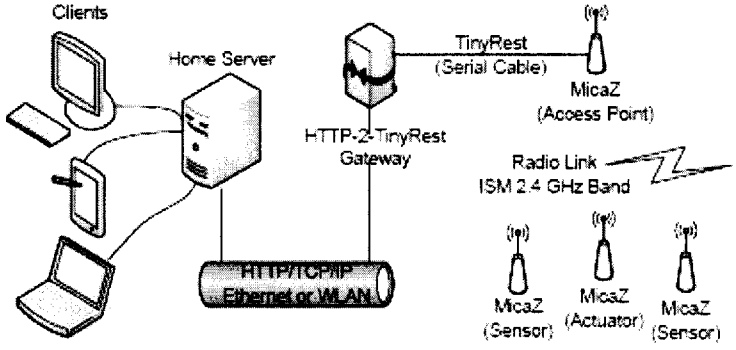


Figure 3.5: TinyREST network architecture

TinyREST is an application level approach focuses on the integration of sensor networks with Internet that uses an HTTP based mechanism for information exchange. While in IMS communication among entities (e.g. users and services/applications) is based on a signaling protocol i.e. SIP and HTTP is not a signaling protocol that could be supported in IMS. Besides that TinyREST neither supports the standard format (PIDF) of information exchange nor does it supports different user entities (e.g. persons, places and objects).

3.3.6 3GPP Presence Framework

The 3GPP presence [16], as discussed in Chapter 2, specified a presence service that provides the user's presence information and possibly the user's context information. The context information can be provided through external sources (e.g. WSN) to entities e.g. watchers or watcher applications in IMS. In presence service an entity that produces a presence information is called presentity, whereas an entity that requests or accesses presence information is called a watcher. A sensor in WSN is a source of contextual information acting as presentity to provide contextual information to presence service. The services and applications in IMS is acting as watcher access contextual information (WSN data) from presence service. The presence service supports a publish-subscribe mechanism for information exchange between presentities and watchers.

The presence based approach seems promising for integration of WSN and IMS. The presence is an integral IMS service that enables the creation of new multimedia services by exploiting the presence (contextual) information. It supports the information exchange in a standard IMS format (PIDF) via SIP messages that is a core signaling protocol of IMS. It possibly support the information related to different entities (e.g. places, objects) proposed in [28], and different WSN capabilities (spatial, physiological, environmental).

3.4 Evaluation Summary

The chart shown in Table 3.1 provides the evaluation summary of previously discussed approaches. These existing works support some of these criteria for the integration of WSN with IMS. The notations used in evaluation table are:

Y – Supported, **N** –Not Supported, **P** -Partially supported.

Related Work	Criteria 1, WSN Capabilities (Spatial, Environmental, Physiological)	Criteria 2, Information Aggregation & Publication in standard IMS format(PIDF)	Criteria 3, Support of different Entities (persons, places, Objects)	Criteria 4, Support of Publish-Subscribe mechanism	Criteria 5, Support of standard IMS protocol - SIP
IP-Enabled WSN	P	N	N	N	N
e-Sense	Y	N	N	Y	Y
Alarm-Net	Y	P	N	N	N
TinySIP	P	N	N	Y	Y
TinyREST	P	N	N	Y	N
3GPP Presence	P	Y	P	Y	Y

Table 3.1: Summary of the evaluation of the related work

As it can be seen from the summary above that 3GPP presence could be viable approach towards the integration of WSN and IMS. Although some features are not supported well but through the proper extensions of presence framework different WSN sensing capabilities can be supported in IMS and the support of different user entities (e.g. persons, places, objects).

3.5 Chapter Summary

This chapter focused on the integration of WSN with other networks in which we discussed two integration approaches: a gateway based approach and an overlay based approach. We also set the evaluation criteria for the integration of WSN with other networks and evaluated the related work with respect to a number of criteria. From the

evaluation we found out that 3GPP Presence could be a potential approach for the integration of WSN and IMS. In the following chapter we will explore the presence-based integrated WSN-IMS architecture.

Chapter 4 An Architecture for Integrating Wireless Sensor Network with IP Multimedia Subsystem

4.1. Introduction

In the previous chapter, we evaluated the existing work for the integration of Wireless Sensor Network with other networks. We chose 3GPP presence approach to integrate WSN with IMS. This chapter discusses the presence service based architecture for the integration of WSNs with IMS and focuses mainly on its design and implementation. The presence-based integrated architecture has been designed by a PhD student in our Telecommunication Service Engineering research laboratory [28], [20]. This section describes the presence-based integrated WSN-IMS architecture. The architecture aims at how information is exchanged between WSN and IMS presence service. Figure 4.1 shows the presence-based integrated WSN and IMS architecture.

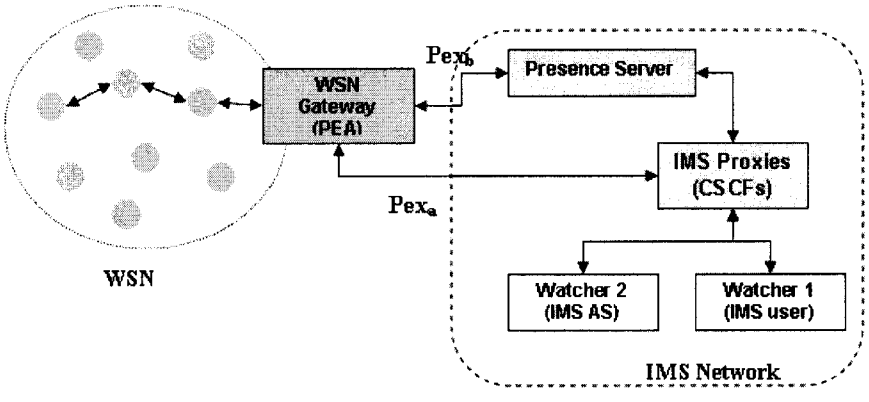


Figure 4.1: WSN and IMS integrated architecture

In the integrated architecture, the WSN-IMS gateway is acting as Presence External Agent (PEA). As defined in 3GPP specification TS 23.141 [14] the role of PEA is to

provide presence information from external networks (e.g., WSN). The WSN-IMS gateway (PEA) task is to publish information about different user entities (person, place, objects) to presence server. The presence information will be exchanged on 'pex' interface defined between presence server and PEA as specified in TS 23.141 [14] and shown in presence functional architecture in Figure 2.15. In the proposed architecture the 'pex' interface is divided into two sub interface: 'Pex_a' and 'Pex_b'. The 'Pex_a' sub interface is used for the exchange of contextual (WSN) information between WSN-IMS gateway and presence server via IMS proxies (CSCFs). 'Pex_a' sub-interface has an indirect interaction with presence server because of the several IMS support functions (e.g., identification, charging, authentication/authorization, service discovery etc.) is already supported by existing IMS infrastructure (e.g., IMS proxies). While the 'Pex_b' sub interface is a direct link between the WSN-IMS gateway and the presence server for managing subscription policies to control access to WSN information. Currently, there is no such standard mechanism (i.e. protocol) defined for information exchange on 'pex' interface but in IMS context SIP is the core signaling protocol for communication. Therefore, SIP is chosen as a standard communication protocol for information exchange on 'pex' interface.

In the next section, we discuss the WSN-IMS gateway architecture and the rest of the chapter discusses the detail design and implementation. The architectural design and implementation involves two phases; the first phase involves the WSN side implementation of integrated WSN-IMS gateway. The second phase involves the IMS side implementation of presence server and extensions required for interworking of WSN and IMS.

4.2. The WSN-IMS gateway architecture

The WSN-IMS gateway is an interworking node between WSN and IMS. It is an application level gateway that provides standard interfaces (Pex_a, Pex_b) for information exchange. The gateway implements protocol and information mappings at different layers. The gateway architecture is structured into two layers: connectivity layer and abstract layer.

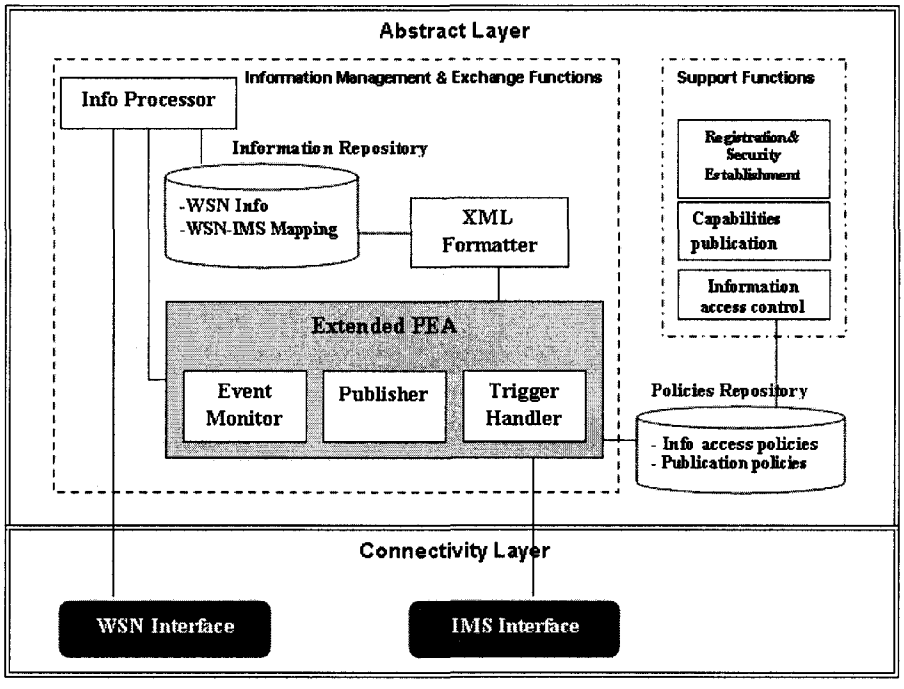


Figure 4.2: WSN-IMS gateway architecture

4.2.1. Connectivity Layer

The connectivity layer consists of dual network interfaces for WSN and IMS. This layer mainly performs a protocol mapping between WSN and IMS networks.

WSN Interface: this interface implements a communication stack of WSN. The communication interfaces between sensor nodes are proprietary. It supports different proprietary interfaces to communicate with diverse sensing platform such as Crossbow

Motes, MIT Crickets. Besides, it collects raw data from sensors and hands it over to abstract layer for further processing and mapping to IMS.

IMS Interface: this interface implements IMS communication stack (SIP) and establishes connectivity to IMS network. The IMS interface interacts with IMS network entities, CSCFs (Proxies) and presence server (PS).

4.2.2. Abstract Layer

The abstract layer is the core functional layer of WSN-IMS gateway. It performs information processing and mapping between WSN and IMS. This layer splits into two main functions; the information management and exchange functions and support functions.

4.2.2.1. Information Management and Exchange Functions

The information management and exchange functions consist of different functional modules:

Information Processor: This module gets the raw WSN data from WSN interface, processes and maps it to an abstract form (e.g., average temperature 24°C or warm, cold). The abstract data is then stored in the information repository and at the same time forwarded it to an event monitor for event detection.

Repositories: The repositories are the storage of WSN and IMS related information. There are two types of repositories, information repository and, policies repository. The information repository stores sensed WSN information as well as WSN and IMS mappings. Policy repository stores the publication and access policies of WSN information accessible to IMS.

Extended Presence External Agent: The presence external agent (PEA) is a core module of WSN gateway. Its main functions include information publication, event monitoring and triggers handling. The extended PEA contains the following sub-modules:

- **Publisher:** Publishes information on interval, event and trigger basis. The interval is a time (e.g., 20 seconds) set for recurrent publication. The information is retrieved either from event monitor or from the repository. The information will always be published in a PIDF format that is a standard XML document.
- **Event Monitor:** Monitors events on WSN data (e.g. temperature > 30). If the event is detected it sends a WSN data to a publisher for immediate publication to IMS. The event monitor maintains a list of events in an XML format. The events are predefined in a gateway and can be defined at runtime. We will discuss the WSN events creation in later sections.
- **Trigger Handler:** Activates information publication upon receiving a publication request from presence server. The publication request is in the form of triggers that shows the requested information. The trigger handler processes the request (triggers), accesses the required information from the repository and hands it to publisher module for publication.

XMLFormatter: This module transforms mapped WSN data to PIDF format. It maps different types of WSN data (e.g., spatial, environmental, and physiological). The XMLFormatter implements an extended PIDF schema [20]. The WSN information would be associated with a WSN entity.

4.2.2.2. Support Functions

These are support functions for interworking between WSN and IMS:

Registration and Security Establishment: This module initiates registration and security association to register a gateway as SIP user agent in the IMS network. It sends a SIP REGISTER request that is the first step in accessing the IMS network.

Capabilities Publication: It publishes WSN service capabilities to Presence Server. The service capabilities imply type of WSN services provided (location, environmental, physiological) and other supported information.

Information Access Control: Enforces the policies defined in policy repository to control access to WSN information. For instance, the publication policies allow publication of information in two modes: proactive and reactive. The access policies allow authorized access to WSN information by IMS users and services.

4.2.3. WSN Events

As discussed previously, the event monitor watch events pre-configured or dynamically configured in a gateway. These events are described in an XML script. The possible way to create, modify and delete events dynamically in the WSN-IMS gateway is through a SIP MESSAGE request. The SIP MESSAGE body contains events description in an XML form as demonstrated in Figure 4.3.

```
MESSAGE sip:wsngw1@ericsson.com SIP/2.0
Via: SIP/2.0/TCP user1pc.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:user2@ericsson.com;tag=49583
To: sip:wsngw1@ericsson.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/xml
Content-Length: 50

<?xml version="1.0" encoding="UTF-8"?>
<WSNMessage>
<WSN-EVENTS>
      <EVENT ID="user1@ericsson.com">
        <COMMAND>ADD</COMMAND>
        <DESCRIPTION>TRACKING USER LOCATION</DESCRIPTION>
      <MASK>
      <LOCATION>
        <CONTION>EQ</CONDITION>
        <VALUE>EV8.235</VALUE>
      </LOCATION></MASK>
</EVENT>
<EVENT ID=" room50@ericsson.com">
<COMMAND>EDIT</COMMAND>
<DESCRIPTIONE>CHECK ROOM CONDITION</DESCRIPTION>
      <MASK>
      <TEMPERATURE>
        <CONTION>GT</CONDITION>
        <VALUE>30</VALUE>
      </TEMPERATURE></MASK>
      <LIGHT-INTENSITY>
        <CONTION>GT</CONDITION>
        <VALUE>NN</VALUE>
      </LIGHT-INTENSITY></MASK>
</EVENT>
</WSN-EVENTS>
</WSNMessage>
```

Figure 4.3: Events description in XML form.

The <COMMAND> tag tells the operation to perform e.g., add, edit or delete event. Events are identified by an event ID i.e., a WSN entity ID that this event belongs. The <EVENT> tag defines an event with its event mask <MASK> tag contains an event attributes (e.g., location, temperature) and its conditions. If the conditions of these attributes are matched, then the event will fire and will be handled by an event monitor.

4.3. WSN - IMS Mappings

This section discusses the detailed design and implementation of WSN-IMS integration. We will discuss here how to map WSN raw data to a unified IMS format and more importantly how we map sensors to IMS entities.

4.3.1. Mapping of WSN Data to IMS

Sensors provide a lot of contextual data (e.g. temperature, light, humidity, and location etc.) sensed from physical phenomena. The WSN data are mostly in proprietary format that would not be meaningful for application users or services consuming it. Therefore a mechanism is needed to map this raw data into a standard format. The gateway does the processing and the mapping of raw sensors data into a standard form (PIDF). Figure 4.4 illustrates an entire information flow for the mapping and publication of WSN data.

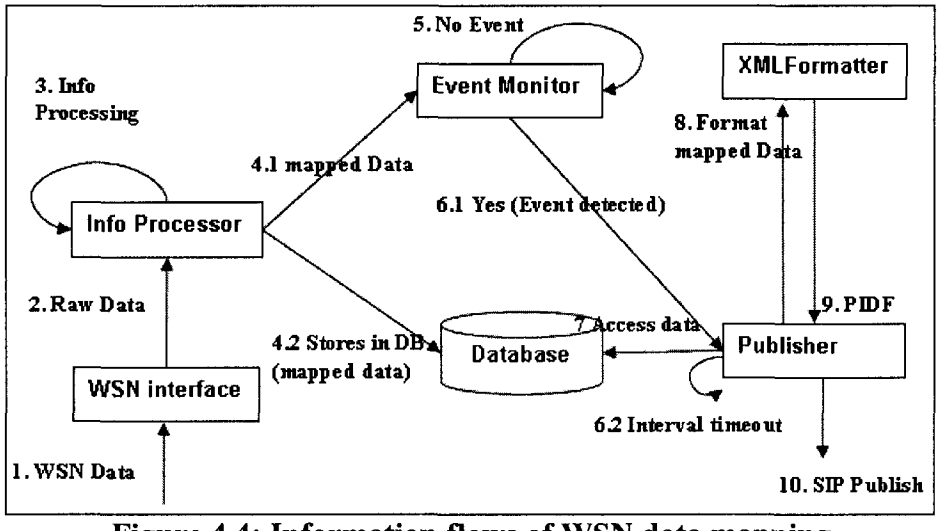


Figure 4.4: Information flows of WSN data mapping

First the sensor data is forwarded by the WSN interface to Information Processor. The Information Processor processes and performs mapping of this raw data to an abstract form. The data is stored in a repository and forwarded to an event monitor for event detection against WSN data. If the event is detected then the data will be sent to the publisher for publication. During information publication the WSN data is transformed into an IMS format (PIDF).

Figure 4.5 shows the raw data stream taken from Crossbow MTS300 and MIT cricket sensors.

```
MTS300 Sensor Output

[2007/04/23 17:59:17] MTS310 [sensor data converted to
engineering units]:
   health:      node id=1 parent=1
   battery:  = 2795 mv
   temperature=25.270422 degC
   light: = 2623 ADC mv
   mic: = 498 ADC counts
   AccelX: = -8880.00000 milliG, AccelY: = -8020.00000 milliG
   MagX: = 107.766724 mgauss, MagY: =107.631676 mgauss
```

```
CRICKET Sensors Output

VR=2.0,ID=1:5e:3c:3c:a:0:0:ba,SP=C-32-0,DB=14,DR=423,TM=1045,TS=207040
VR=2.0,ID=01:dd:be:be:09:00:00:95,SP=A-32,DB=224,DR=6479,TM=6789,TS=455424

space=C-32 pos=( 110.68068181818181 51.996363636363654 74.20246458136243 )
space=C-32 pos=( 105.02765151515152 49.79030303030304 70.9467680654887 )
space=A-32 pos=( 105.02765151515152 49.79030303030304 70.9467680654887 )
space=A-32 pos=( 99.99583333333334 47.826666666666675 67.46006180099782 )
```

Figure 4.5: MTS300 and Cricket sensors output

The whole scenario of WSN data mapping to IMS can be realized at different levels of data abstraction as shown in Figure 4.6. At lowest level we have (raw) data sensed from individual sensors that is processed and transformed into a mapped data object. The mapped data object is then formatted to an XML document at the highest level that can be published to IMS.

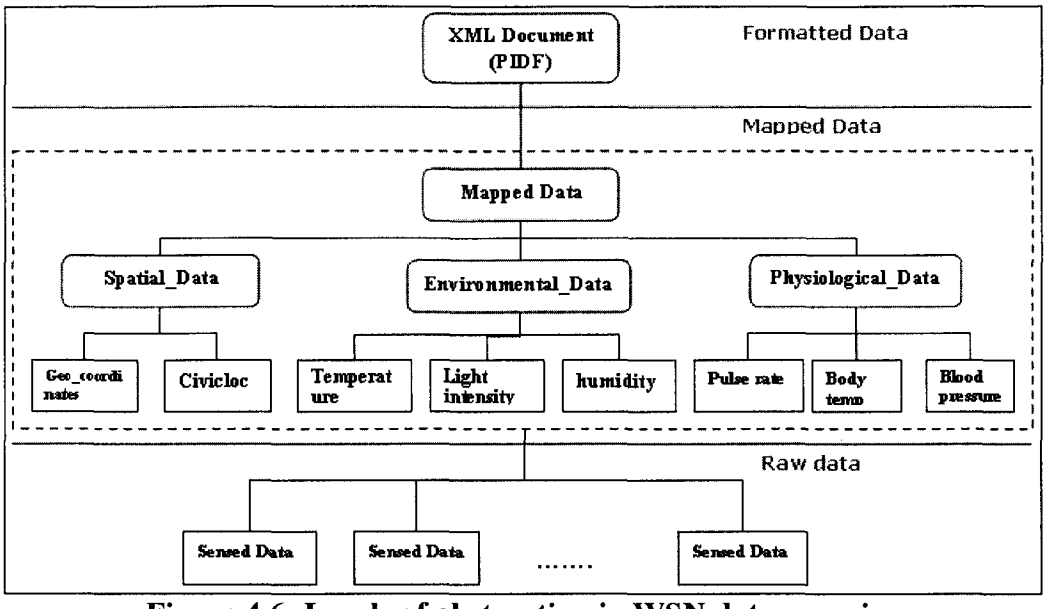


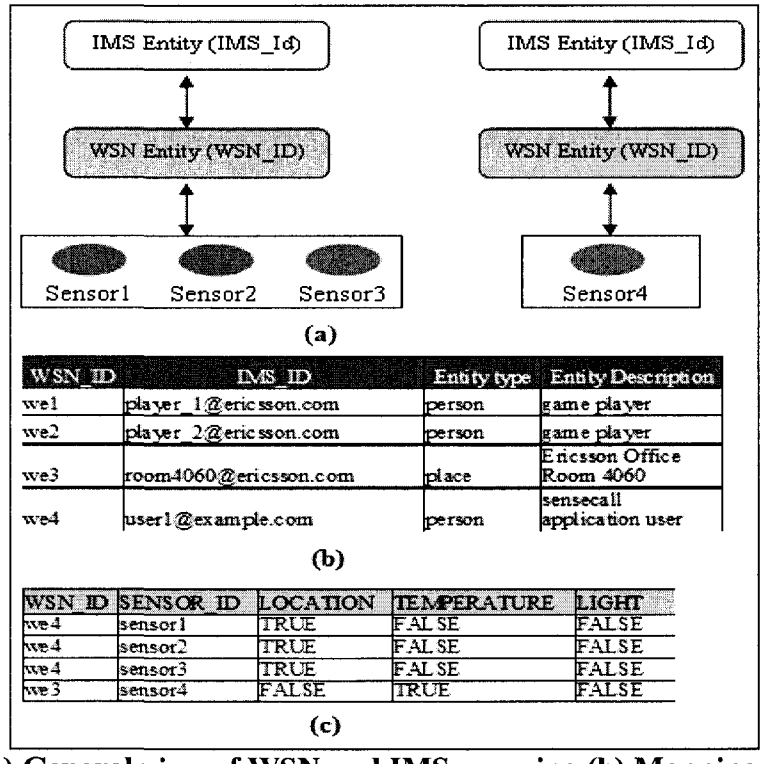
Figure 4.6: Levels of abstraction in WSN data mapping

The WSN mapped data object can be classified as spatial, environmental and physiological data. The spatial data refers to user's location in a bounded space. The location can be represented either as geographical coordinates (e.g. GPS) or space (e.g. postal code or civic address). The environmental data consists of sensed data related to environmental conditions (e.g. temperature, light intensity, noise, and humidity). The physiological data represent vital signs of any living person or patient (e.g. heart rate, blood pressure, glucose level, body temperature).

4.3.2. Mapping of Sensors to IMS Entities

Sensor networks are data centric, i.e. sensors do not possess any global identification. Unlike WSN, in IMS every entity (user) has an assigned public identification (URI) to communicate with other entities (users). The sensors can provide information related to different entities. The entities can be a person (user), a place, or an object. As examples, what is the current location of user A (person) in bounded space?, temperature of corridor (place)?, or location of first-aid kit (object)? Currently IMS only

supports user as a subscribed entity and no such support for non-user entities such as places and objects. Therefore, a solution has been proposed to support non-user entities and the identification scheme for non-user entities in [28]. The main issue is how to map sensors associated with these entities to IMS? We devised a logical address mapping from WSN to IMS. Each sensor or group of sensors in WSN represents a WSN entity that is assigned a unique logical identity which could be an arbitrary string (e.g. "WE1") as shown in Figure 4.7a. For each WSN identity there is an IMS identity defined in a mapping table as shown in Figure 4.7b. The same process applies the other way around for mapping from IMS to WSN. First the IMS entity is mapped to a WSN entity and then the mapped WSN entity to a particular sensor. The WSN mapped data is assigned to a designated WSN entity with the help of the mapping table.



(a)

WSN_ID	IMS_ID	Entity type	Entity Description
we1	player_1@ericsson.com	person	game player
we2	player_2@ericsson.com	person	game player
we3	room4060@ericsson.com	place	Ericsson Office Room 4060
we4	user1@example.com	person	sensecall application user

(b)

WSN_ID	SENSOR_ID	LOCATION	TEMPERATURE	LIGHT
we4	sensor1	TRUE	FALSE	FALSE
we4	sensor2	TRUE	FALSE	FALSE
we4	sensor3	TRUE	FALSE	FALSE
we3	sensor4	FALSE	TRUE	FALSE

(c)

Figure 4.7: (a) General view of WSN and IMS mapping (b) Mapping table of WSN entity to IMS entity (c) mapping of sensors to WSN entity

4.3.3. WSN Service Discovery

WSNs provide different types of information at different levels of granularity usually the information is accessed via WSN gateway. To access these services provided by WSN, the mechanism is needed for discovery of WSN gateways by presence server. This process enables the PS to choose the best information sources (WSNs) for subscribers needs. The possible approach to discover WSN gateway is by publishing WSN service capabilities to IMS. The service capabilities refer to types of WSN data (e.g. spatial, environmental, and physiological) provided by WSN. The scenario is illustrated in Figure 4.8.

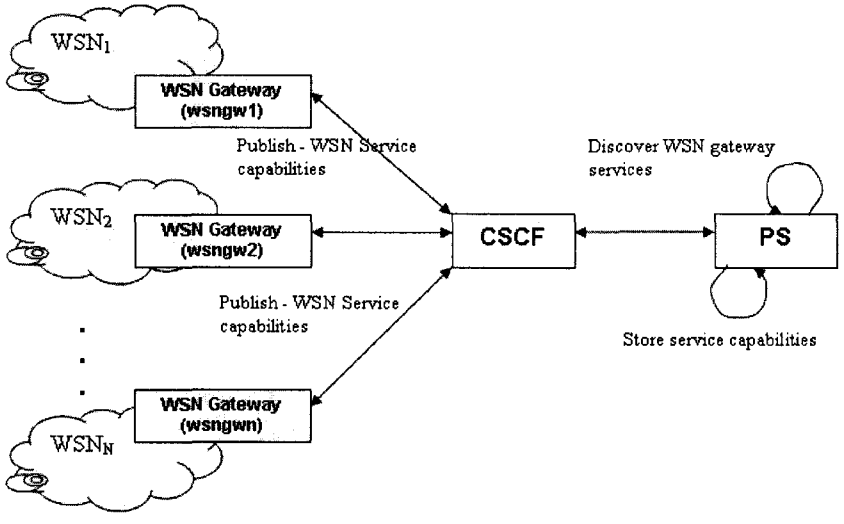


Figure 4.8: WSN capabilities publication.

The WSN gateways publish their service capabilities to presence server. The presence server accesses the service capabilities of different WSNs and makes a service profile for different WSNs. Service capabilities can be published during IMS registration of WSN gateway via SIP REGISTER request. The register request contains WSN service capabilities information. The CSCF registers the gateway and forwards the request to

presence server. The presence server extracts and stores the capabilities information contained in a register request. Figure 4.9 shows an example of service capabilities publication.

```
REGISTER sip:registrar.ericsson.com SIP/2.0
Via: SIP/2.0/UDP
wsngw1.ericsson.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Bob <sip:wsngw1@ericsson.com>
From: Bob <sip:wsngw1@ericsson.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:wsngw1@192.0.2.4>
Expires: 7200
Content-Length: 40

<?xml version="1.0" encoding="UTF-8"?>
<WSNMessage>
<WSNGW id="wsngw1@ericsson.com">
     <SERVICE-CAP>
       <SPATIAL>
             <SPACE>TRUE</SPACE>
             <COORDINATES>FALSE</COORDINATES>
       </SPATIAL>
       <ENVIRONMENTAL>
             <TEMPERATURE>TRUE</TEMPERATURE>
             <LIGHT>FALSE</LIGHT>
             <HUMIDITY> FALSE </HUMITIDTY>
       </ENVIRONMENTAL>
     </SERVICE-CAP>
     <SENSOR-NODES>10</SENSOR-NODES>
</WSNGW>
</WSNMessage>
```

Figure 4.9: WSN capabilities publication via SIP REGISTER message.

4.4. Extended IMS Presence Server

This section covers the extensions related to presence framework for accessing WSN information in IMS.

4.4.1. Presence Server operations

Presence server is a functional entity (presence agent) and deployed as an IMS application server that collects and distributes users' (presentity) presence (status) information. The presence information can be collected from different sources e.g. user devices – phones, PCs, PDAs and external sources like other network services (MSN, Yahoo) or WSNs. The information collected from different sources are unified into an

XML document called PIDF. A presence server performs the following basic operations, also shown in Figure 4.10.

- **Subscription**: handles subscription requests to a presentity's status. The subscription request is in the form of SIP SUBSCRIBE message that contains event header field set to 'presence'.
- **Notification**: notify watchers about changes in presentity's state. The notification is a SIP NOTIFY request that contains a user's updated presence information.
- **Publication**: handles publication of different user's status. The publication is a SIP PUBLISH message that contains presence information in its body.
- **Authorization**: authorizes access of presentity's presence information by applying access policies set by presentity or network operator.

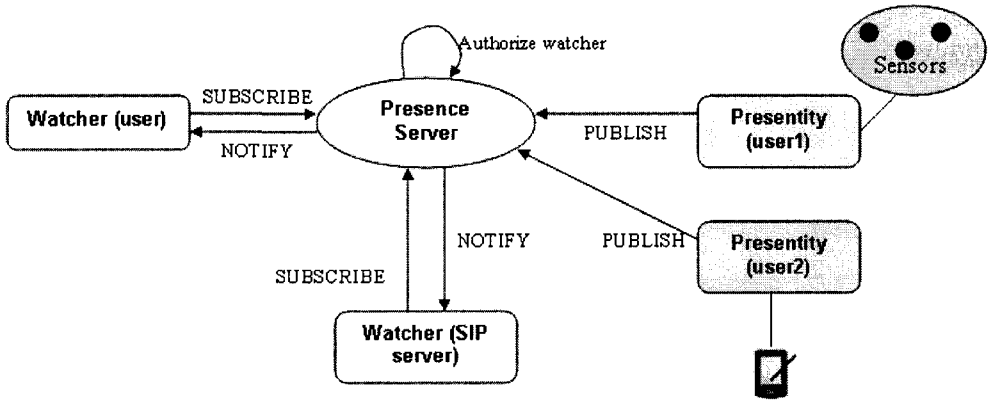


Figure 4.10: Presence service operations

Figure 4.11 below show the mappings of presence operations to SIP messages.

```
SUBSCRIBE sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP
host.example.com;branch=z9hG4bKnashds7
To: <sip:presentity@example.com>
From: <sip:watcher@example.com>;tag=12341234
Call-ID: 12345678@host.example.com
CSeq: 1 SUBSCRIBE
Max-Forwards: 70
Expires: 3600
Event: presence
Contact: sip:user1@host.example.com
Content-Length: 0
```

(a)

```
NOTIFY sip:user1@host.example.com SIP/2.0
Via: SIP/2.0/UDP
pa.example.com;branch=z9hG4bK8sdf2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 1 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active; expires=3599
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: N

[ PIDF Document ]
```

(b)

```
PUBLISH sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP
pua.example.com;branch=z9hG4bK652hsge
To: <sip:presentity@example.com>
From: <sip:presentity@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
Expires: 3600
Event: presence
Content-Type: application/pidf+xml
Content-Length: N

[ PIDF Document ]
```

(c)

Figure 4.11: SIMPLE presence protocol messages (a) SUBSCRIBE (b) NOTIFY (c) PUBLISH

4.4.2. Extended Presence Information Model

IETF defined the basic presence information model in RFC 3863 [41] as Presence Information Data Format (PIDF). PIDF is compatible with various presence protocols (e.g. SIMPLE and XMPP) that carry presence information. The standard PIDF format enables different presence applications to exchange user's presence information in a unified way.

PIDF is an XML document that defines a basic structure of user's presence status. The presence status contains number of XML elements to represent user's status (e.g. open or close), contact address (e.g. sip:alice@ericsson.com) and additional extension elements. Figure 4.12 shows the basic structure of PIDF.

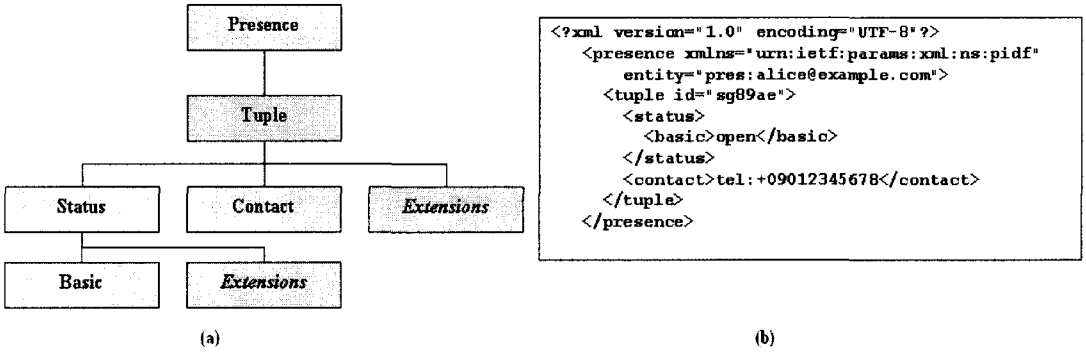
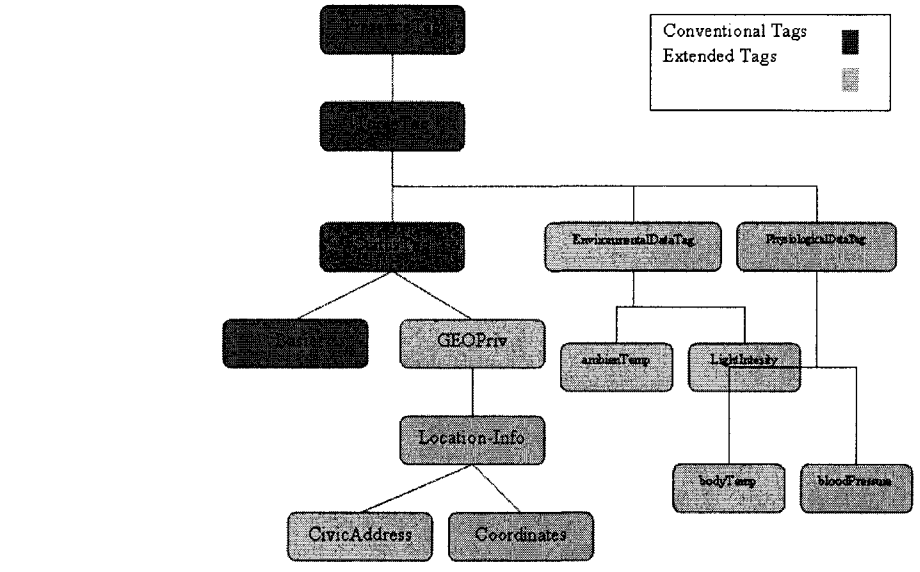


Figure 4.12: (a) Basic PIDF structure (b) Example PIDF document

Value-added service e.g. context aware emergency services and pervasive gaming, are not only interested in user's availability but also in user's contextual information. Therefore, the existing PIDF needs to be extended to support different types of contextual information (e.g. spatial, environment, and physiology etc). The extension for spatial (location) type information is already defined in [42] called GEOPRIV. The GEOPRIV specified the format for representation of geographic location. However, for the environmental and physiological information the basic PIDF schema has been extended

[27]. The extension tags for environment and physiological data are defined under Tuple tag of the PIDF document. Figure 4.13a shows the extended PIDF model and an example is given in Figure 4.13b.



(a)

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:pidf"
xmlns:tns="urn:ietf:params:xml:ns:epidf"
entity="pres:room4060@ericsson.com" entityType="place"
entityDescription="Ericsson Office Room 4060">
<tuple id="wsninfo" >
<tns:environmentalData>
<tns:ambientTemperature unit="celcius">26.68
</tns:ambientTemperature>
<tns:lightIntensity unit="candelas">1552
</tns:lightIntensity>
</tns:environmentalData></tuple>
</presence>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv"
xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicLoc"
entity="pres:sip:user1@ericsson.com" entityType="person"
entityDescription="SenseCall application user">
<tuple id="wsninfo" >
<status>
<gp:geopriv><gp:location-info>
<cl:civicAddress>
<cl:country>Canada</cl:country>
<cl:A1>Quebec</cl:A1>
<cl:A3>Montreal</cl:A3>
<cl:A4>Ericsson-8400</cl:A4>
<cl:LOC>room50</cl:LOC>
</cl:civicAddress>
</gp:location-info></gp:geopriv>
</status>
</tuple>
</presence>
```

(b)

Figure 4.13: (a) Extended PIDF information model, (b) Example PIDF extensions

4.4.3. WSN information publication

The task of WSN gateway is to publish WSN data. The sensors produce data (e.g. location, temperature, velocity, sound level etc.) in a continuous stream. Thus, it could be

burdensome to publish sensed data in a continuous fashion and this may result in a network overload. To control the rate of information publication, two publication modes have been introduced: proactive and reactive.

The proactive publication mode is further categorized into two types. The interval based publication in which WSN information would be published on regular intervals (e.g. every 30 seconds). While the event based publication publishes information on event basis (e.g. temperature > 30). The events are examined by an event monitor and discussed in Section 4.2.3. Figure 4.14 illustrates the proactive mode of publication.

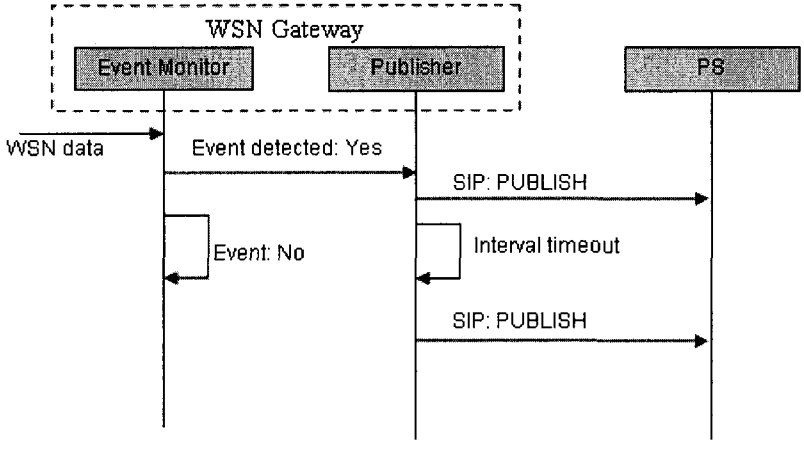
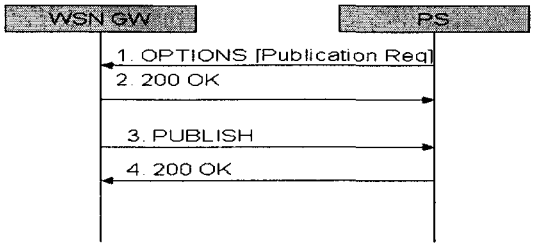


Figure 4.14: Proactive mode of publication

The reactive mode is a trigger based publication. It is a special type of publication that is rendered by presence server requesting certain type of information to be published by the WSN gateway. The trigger request is sent as SIP OPTIONS message and the body of message contains publication triggers in an XML form. The trigger describes which presentity's and what type of information e.g. location (LOC), temperature (TEMP) has to be published. The publication policies stored in the WSN gateway select either mode

for information publication of WSN data. Figure 4.15 shows an illustration of reactive publication.




```
OPTIONS sip:wsngw1@ericsson.com SIP/2.0
Via: SIP/2.0/UDP gwhost.ericsson.com;branch=z9hG4bKh8as588
Max-Forwards: 70
To: <sip:wsngw1@ericsson.com>
From: <sip:presence@ericsson.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 6 OPTIONS
Contact: <sip:presence@ericsson.com>
Accept: application/xml
Content-Length: 50

<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<WSNMessage>
<PUBLISH>
        <PRESENTITY>
        <EntityId>user1@ericsson.com</EntityId>
            <InfoType>LOC</InfoType>
        </PRESENTITY>

        <PRESENTITY>
        <EntityId>ev-build@sgwcampus.concordia.ca</EntityId>
            <InfoType>temp</InfoType>
        </PRESENTITY>
</PUBLISH>
</WSNMessage>
```

Figure 4.15: Reactive publication flow and publication request

4.4.4. Presence Server functional architecture

The presence server is an essential component of presence-based integrated WSN and IMS architecture. The extended presence server is complaint with IETF SIMPLE standards and design based on the extended features in our proposed solution of WSN and IMS integration. Extended presence server is an information base of contextual data (WSN data) that is accessed by different IMS services and applications. The design of a presence server considers the following extended features discussed in pervious sections (4.3.3, 4.4.2, 4.4.3):

- Extended Presence Information Model
- WSN information publication (publication modes)

- WSN service discovery.

The commercially available presence servers in the industry today, like IBM Websphere Presence Server [48], Ericsson PGM [46] and Jabber XCP [49], support none of these extensions discussed here. Figure 4.16 shows the functional architecture of the extended presence server.

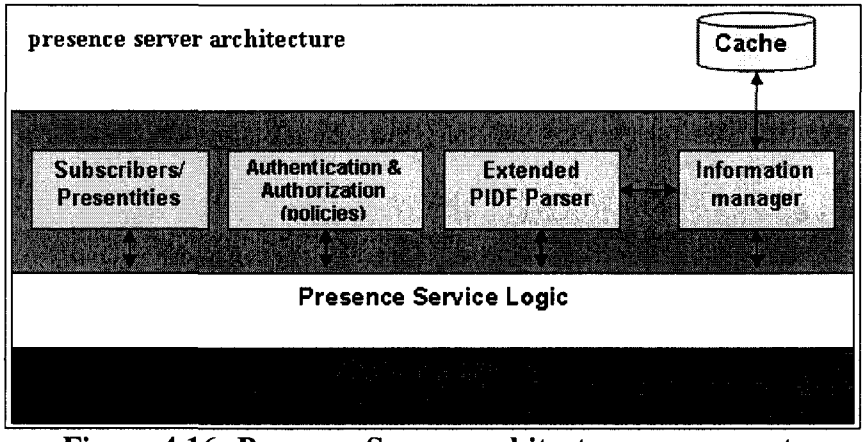


Figure 4.16: Presence Server architecture components

Presence Server architecture consists of protocol and service related components. The SIMPLE stack implements SIP based presence and instant messaging features. The presence service implements the basic presence service operations (subscription, notification and publication of information). The two main components that implement the extended functionalities listed above are: Extended PIDF parser, and Information Manager. PIDF parser implements the extended PIDF information model to support different types of WSN data (spatial, environmental, and physiological). Information manager is an information base for processing, accessing all published WSN information such as WSN service capabilities, presence (context) information and reactive publication request (querying WSN data). The Information manager stores all published WSN data in a cache. The cache is a storage space of published WSN information. The

subscriber/presentity component maintains a list of subscribers and presentities that subscribe and publish presence information. Finally, the authentication/authorization module store policies that control access to presentity's presence information. The policies are defined by either presentity or WSN-IMS gateway. The presence server also supports SIP PUBLISH message that is currently not supported by SIP Serlvet API [45]. The SIP PUBLISH message specification is defined in RFC 3903 [25].

4.5. System Implementation

In this section we discuss the implementation of WSN-IMS gateway, presence server and implementation environment.

4.5.1. Implementation of the WSN-IMS gateway

For the implementation, we chose Java based SIP APIs software realizing the WSN-IMS gateway. All the components of WSN gateway have been implemented except capability publication and policy control modules. The components of WSN-IMS gateway architecture are structured into Java classes. Figure 4.17 shows the implemented Java classes.

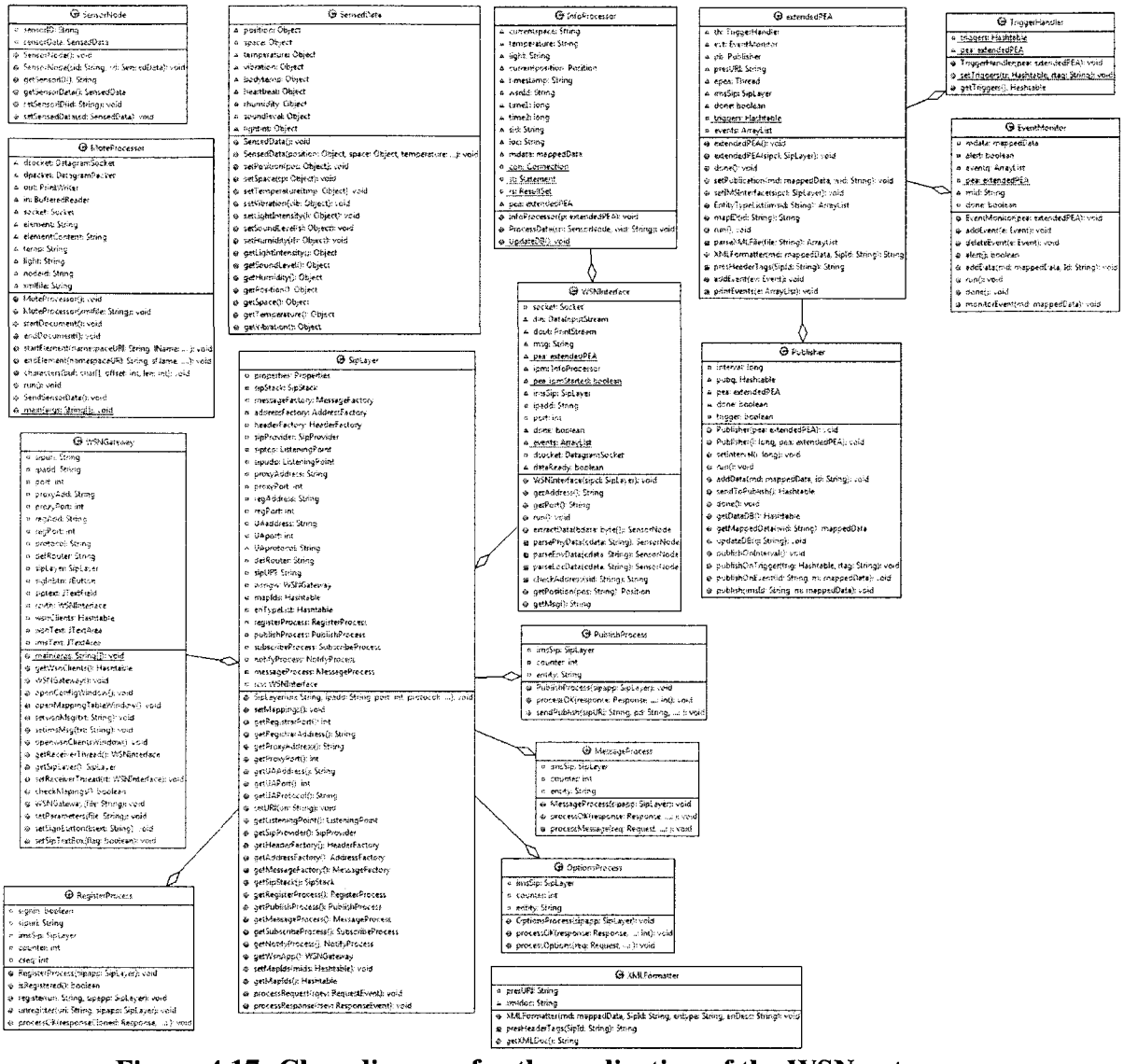


Figure 4.17: Class diagram for the realization of the WSN gateway

The 'WSNGateway' class is a main application of WSN-IMS gateway. It provides a user interface to allow authorized users or network administrators to configure different parameters related to WSN and IMS network. The two network interfaces for IMS and WSN are implemented into separate classes as 'SipLayer' (IMS Interface) and 'WSNInterface' (WSN Interface). The 'WSNInterface' class implements WSN communication stacks for different WSN vendors platform devices. Similarly the 'SipLayer' class implements SIP stacks for exchanging SIP messages with IMS network.

The JAIN SIP API [43] had been used for the implementation of SIP protocol stack. 'SensorNode' class is an object representation of a sensor node including the sensed data reported by each sensor node. The raw sensor data is encapsulated in a 'SensedData' class. The main abstract layer functionalities are implemented as a set of classes: InfoProcessor, ExtendedPEA, Publisher, EventMonitor, TriggerHandler and XMLFormatter.

4.5.2. Implementation of the presence server

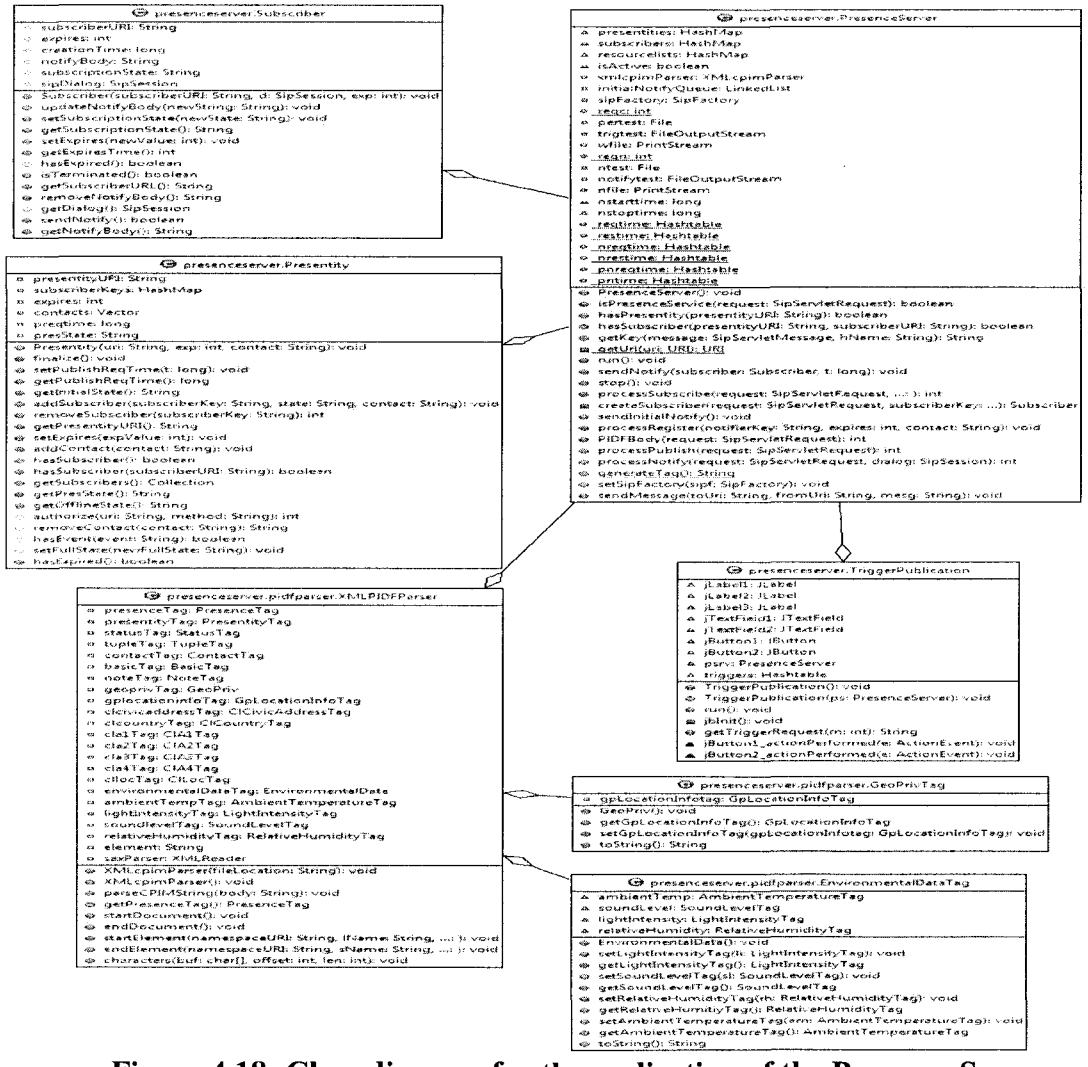


Figure 4.18: Class diagram for the realization of the Presence Server

Similarly for the implementation of presence server, we used Java based SIP APIs. In presence server functional architecture different components are implemented separately according to their roles except the authentication/authorization and the information manager modules. We used SIP Servlet API [45] to implement SIP (SIMPLE) stack in presence server. The 'PresenceServer' class is the main class that implements presence service logic. 'Subscriber' class is an object representation of a subscribed watcher similarly a 'Presentity' class is an object representation of a presentity. 'XMLPIDFParser' class implements an extended PIDF parser. It decomposes PIDF document into separate XML elements represented as small java classes that store tag values or its child elements (tags). The 'TriggerPublication' class implements reactive mode of publication. Figure 4.18 shows the simplified class diagram of presence server.

4.5.3. Implementation environment

Different software and hardware tools have been used for the implementation.

4.5.3.1. Hardware environment

Sensor devices used in our implementation include Crossbow sensors [50] for environmental data and MIT cricket sensors [21] for spatial (location) data.

The Crossbow devices for WSN platform includes

- MICA2 (MPR400) wireless node,
- Gateway node (MIB510), and
- Sensor board (MTS300).

MTS300: provides various sensing functions: temperature, light, and microphone. The light sensor detects light intensity and the output of sensor is a numeric value that

represents resistance e.g.. 2kΩ. The temperature sensor contains a thermostat that outputs the temperature value. Figure 4.19 shows the MT300 sensor board.

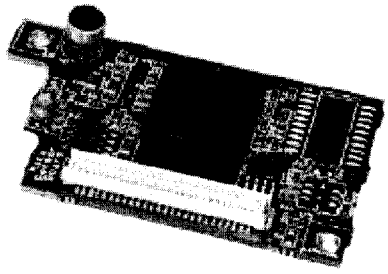


Figure 4.19: Crossbow MTS300 sensor board

MICA2: is a wireless node that has a microprocessor and a wireless transceiver. The MTS300 is attached with MICA2 to make up an operational sensor node that can sense and communicate with other sensor nodes in a WSN. The block diagram in Figure 4.20 illustrates onboard components of MICA2.

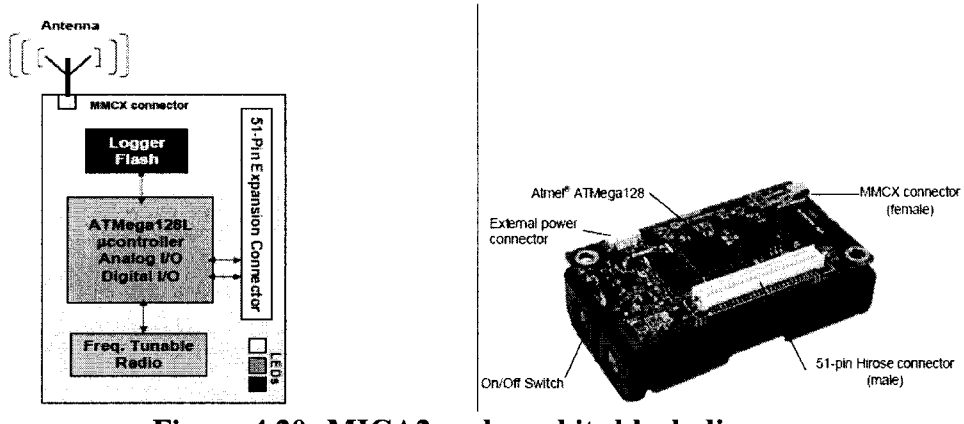


Figure 4.20: MICA2 node and its block diagram

MIB510: is a gateway (sink) for WSNs. It has a RS232 interface to connect with PCs for programming and monitoring of sensor nodes. It collects data from different sensor nodes and reports it to applications e.g. MoteView [51]. MoteView is an application provided by Crossbow to monitor sensor data. Figure 4.21 shows MIB510 node.

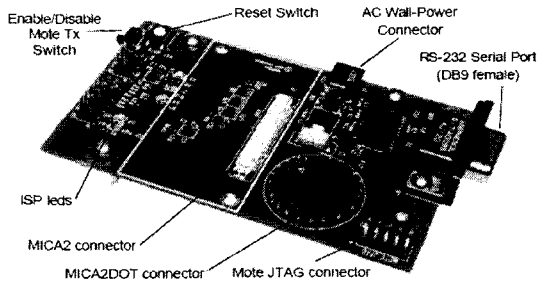


Figure 4.21: MIB510 gateway node

MIT Cricket Sensors: location sensors enable the development of location-based applications. Cricket sensors report location in the form of space identifier (Id), and coordinates assigned to cricket sensors. The space Id is a location identifier which is actually a user defined string (e.g. "room50", "EV5.251") in each sensor node. The coordinates are in the form of (x,y,z) in Cartesian coordinate system. It operates in either of two modes, beacon or listener. The beacons are configured with space identifiers and coordinates that are actively transmitting data to listeners, While the listener passively listening to different beacons at the same time. Usually, the listener is attached with hosts PC via an RS 232 serial interface. Figure 4.22 shows a cricket sensor.

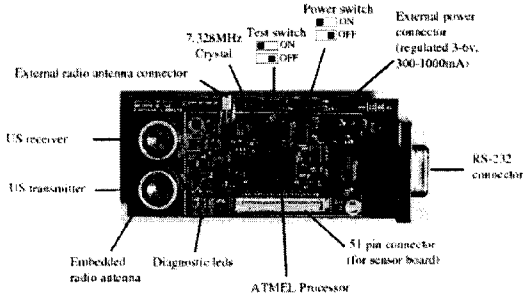


Figure 4.22: MIT cricket sensor

4.5.3.2. Software environment

The software tools include the development APIs for the implementation of WSN-IMS gateway and IMS presence server. We tested our system in Ericsson Service Development Studio that is an IMS emulated environment.

JAIN SIP API: The JAIN SIP is a Java based SIP specification defined in JSR 32 by Java Community Process (JCP) [43]. The API provides interfaces for application development and portability across different vendor specific SIP implementation. It can be used to develop different SIP entities like standalone SIP user agent, proxy and registrar server. The JAIN SIP API supports the following standard SIP specifications (RFCs) shown in table 4.1.

SIP Specification	SIP Feature
RFC 3261	SIP protocol features
RFC 2976	INFO method
RFC 3262	support reliability of provisional (1xx) responses
RFC 3265	SIP Event Notification Framework
RFC 3311	UPDATE method
RFC 3326	support Reason header
RFC 3428	MESSAGE method
RFC 3515	REFER method
RFC 3581	Distributing Authoritative Name Servers via Share Unicast Addresses
RFC 3903	PUBLISH method

Table 4.1: Supported SIP specifications in JAIN SIP

The WSN-IMS gateway is implemented as a standalone SIP user agent using the JAIN SIP API.

SIP Servlet API: Another Java based SIP API defined in JSR 116 [45]. Like HTTP servlets, SIP servlets are deployed in an application server. The application server includes SIP container that provides an execution environment for SIP servlets. The SIP Servlet API supports the following SIP specification (RFCs) as shown in Table 4.2

SIP Specification	SIP Feature
RFC 3261	SIP protocol features
RFC 2976	INFO method
RFC 3262	Support reliability of provisional (1xx) responses
RFC 3265	SIP Event Notification Framework
RFC 3428	MESSAGE method

Table 4.2: Supported SIP specifications in SIP Servlet API

Ericsson Service Development Studio (SDS): The Ericsson service development studio (SDS) provides an environment for design, implementation and testing of IMS value-added services. The client side of SDS consists of two components, the IMS Client Platform (ICP) and the developed IMS Device Client. The ICP supports an IMS client application development for different platforms e.g. mobile client with Symbian OS and PC clients with Windows OS. The server side of SDS contains IMS network entities (CSCFs, HSS, and DNS) and hosted application services. The SDS emulator comes with

pre-configured IMS proxies and application servers (e.g. PGM, IMS-M) as illustrated in Figure 4.23.

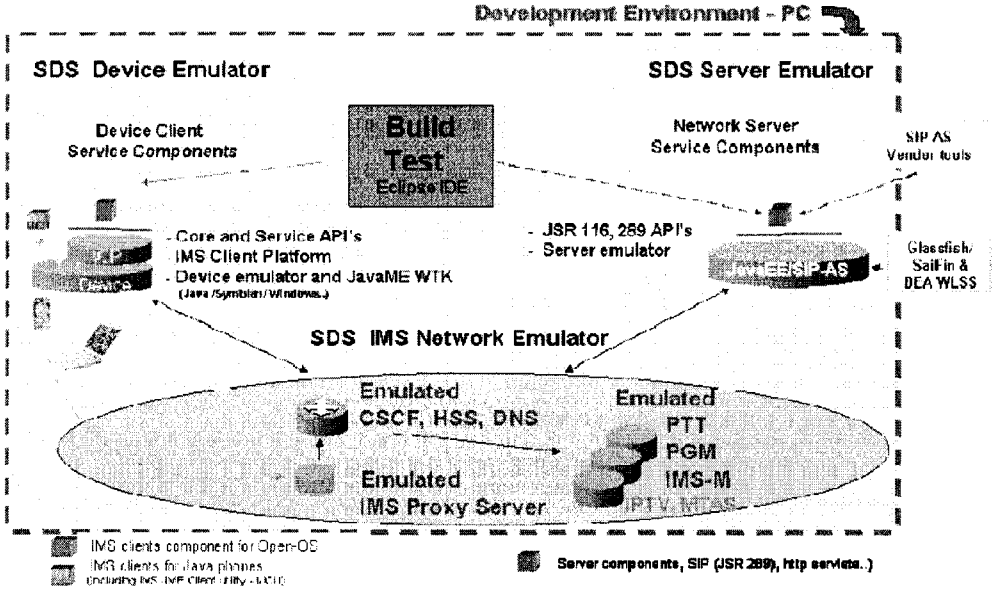


Figure 4.23: SDS emulation environment

Cricket Java API: MIT Cricket sensors provide Cricket java client API called Clientlib [47] to develop location-based applications. The cricket provides software called cricketd and cricketdaemon that runs on host device with attached listener to retrieve sensor data. Figure 4.24 shows the cricket software architecture.

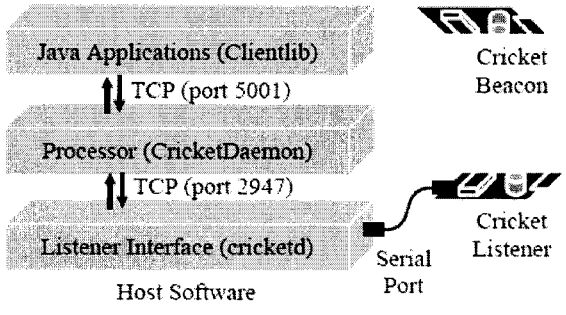


Figure 4.24: Cricket software architecture

4.6. Chapter summary

This chapter explored the design and implementation of integrated WSN and IMS architecture. We discussed mappings between WSN and IMS worlds, and the publication mechanism of WSN data. We explored IMS related extensions required for information exchange between WSN and IMS. The extended features include extension of basic presence information data format (PIDF) to shape different types of WSN data and WSN service discovery in IMS. We also discussed the system implementation and the implementation environment. Above all the proposed work should be tested to see its applicability to real world applications with performance evaluation of the overall system. The next chapter will focus on prototyping part of our integrated WSN and IMS solution in which we discuss prototype applications and performance evaluation.

Chapter 5. Prototype Applications and Performance Evaluation

This chapter discusses the proof of concept prototype through prototype applications and preliminary performance evaluation.

5.1. Prototype Applications

The presence-based integrated WSN and IMS architecture provides an abstraction to IMS service developers for rapid creation of new services and applications. This can be realized as an integrated WSN-IMS gateway that hides all the complexity and details from developer for communicating with WSN. With the help of presence operations such as subscription, publication, and notification, services and applications can access WSN services (e.g., spatial, environmental data) via presence server. Therefore, service developers do not need to implement any proprietary WSN logic in creating new IMS services and applications to access WSN data. The service developers only have to care about what services (e.g. mobile gaming, instant messaging) they are implementing, and what additional services (e.g. presence) required to support this new service. Figure 5.1 shows the generic WSN - IMS architecture.

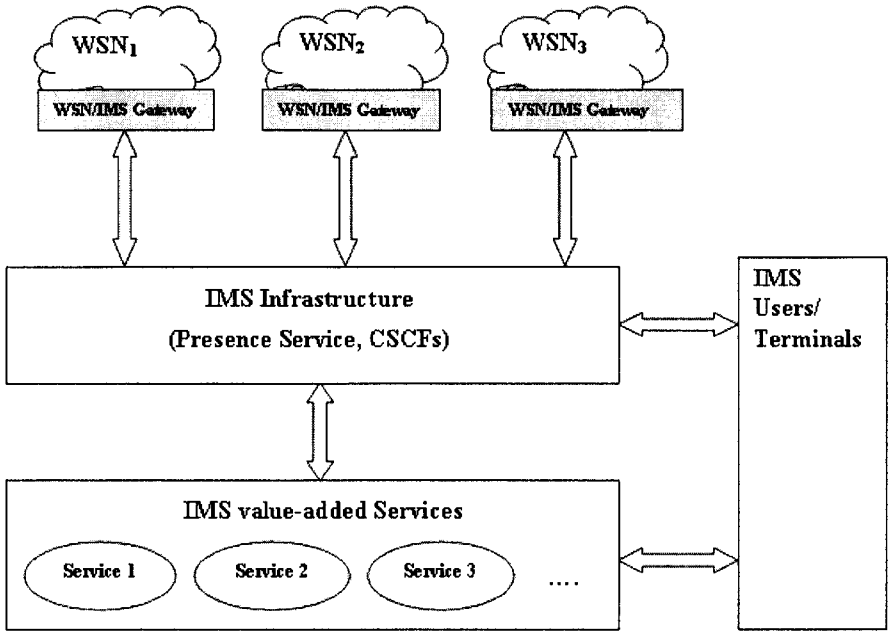


Figure 5.1: Prototype architecture of WSN and IMS

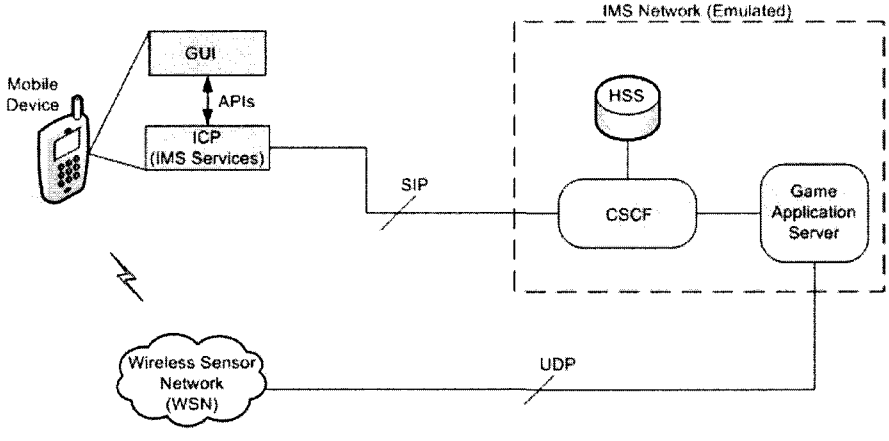
Today service providers provide basic services like telephony and some additional services e.g. SMS, MMS but these services are not sufficient with growing customers needs. The customers are now looking for new multimedia and entertainment services like mobile gaming, interactive chat etc. Therefore in the following section, we will demonstrate the applicability of integrated WSN-IMS architecture with the help of two prototype applications: Fruit Quest is a mobile pervasive gaming application and the SenseCall, which is a third party call setup application.

5.1.1. Fruit Quest Application

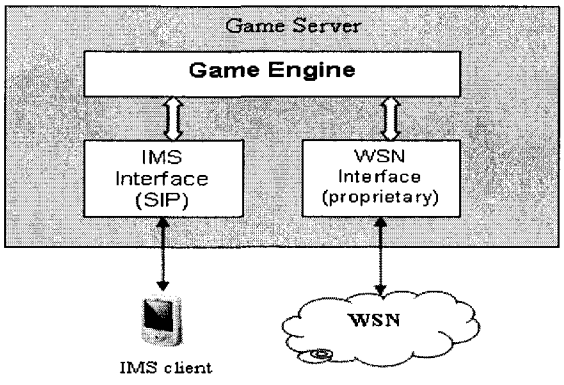
MOTIVATION: Mobile gaming is emerging as a new entertainment and multimedia service domain in mobile networks. For instance, mobile users can play game anytime, anywhere with other mobile users without regard to their current location. However in the

gaming world the games are categorized as: computer-based games and pervasive games [58]. In computer based games users interact with virtual (simulated) environment of computer screen with 2D/3D graphics. While pervasive games bring experience of both physical environment (player's interaction with physical world) and virtual environment into the gaming world. The physical environment can be realized through WSN. Therefore we opted for mobile pervasive game called Fruit Quest for illustrating of WSN - IMS integration.

ARCHITECTURE: Fruit Quest is a mobile pervasive game developed as a part of capstone project [53]. The goal of the project was to design and implement a pervasive game as a prototype for their WSN - IMS integration. The initial Fruit Quest architecture is shown in Figure 5.2.



(a)

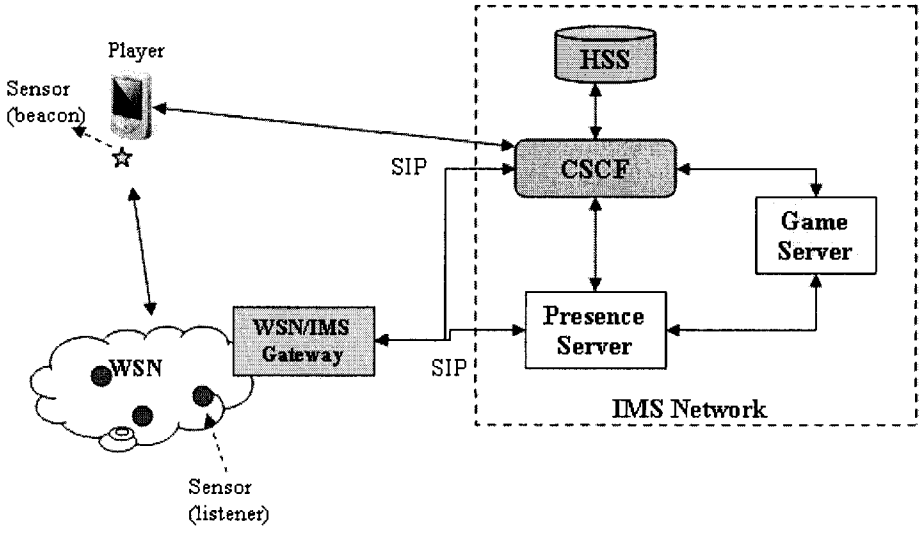


(b)

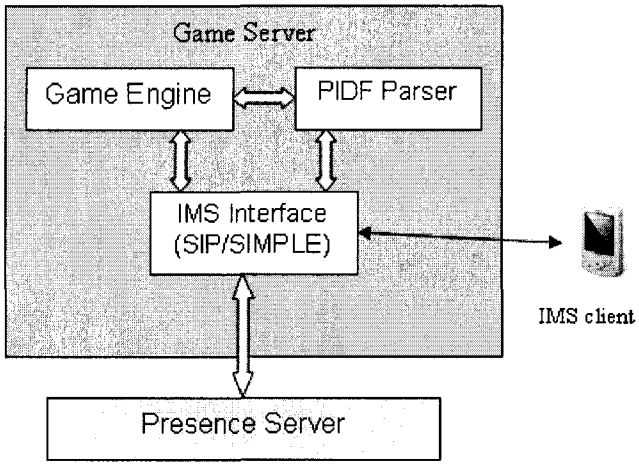
Figure 5.2: (a) Original Fruit Quest architecture, (b) original game server internal architecture

In the initial Fruit Quest design [53] the WSN is directly communicating with the game server which is deployed in IMS as an application server as shown in Figure 5.2b. There is no gateway in this architecture. The interface between the game server and the WSN is proprietary. We redesigned the game architecture in the context of our integrated WSN and IMS architecture as shown in Figure 5.3. The proprietary WSN interface is removed from the game server and implemented an enhanced SIP interface with SIMPLE support to access WSN information from Presence server. In addition to that a PIDF parser is implemented in the game server to process PIDF document. This also requires modification in the game engine software to interact with newly implemented modules. The SIP interface initially implemented in the Games Server provides basic SIP operations (INVITE, MESSAGE) and later extended to support SIMPLE interface for presence service operations. The presence server is implemented from scratch and deployed as an IMS application server. PS is acting as a service enabler to provide presence (context) information to game server and generally to other IMS services or applications as well. The integration of WSN and IMS in this application scenario is

realized as a deployment of WSN-IMS gateway i.e., a standard access point between WSN and IMS network. Figure 5.3 shows the customized architecture of Fruit Quest game.



(a)



(b)

Figure 5.3: (a) Fruit Quest customized architecture, (b) modified game server architecture

WORKING: In the game we actually track the player's location in distinct zones. The game physical environment is a WSN in which each listener sensor represents a zone

while the beacon sensor represents a game player. The listener listens to beacons to locate players in a particular zone. In a game every player must try to occupy as many zones (plantations) and grow a fruit in these zones. The zone is a plantation area for a specific fruit. At the start of the game each player has to choose a specific fruit and a zone (e.g. 1, 2, 3, and 4) for plantation. The game server is deployed as an IMS SIP application server in Ericsson SDS platform. The IMS network entities (CSCF, HSS, and Presence Server) are also deployed in Ericsson SDS platform. The players have an IMS enabled mobile device with attached sensor roaming in WSN. The sensor reports location information of a player to WSN-IMS gateway. The gateway publishes player's location information to presence server and the game server which is acting as a watcher will access player's location information by subscribing to player's presence (location) information. Presence server will notify game server about the location updates of a player, also the game server sends game related updates (statistics) to player's mobile devices. Figure 5.4 shows the information flow of the Fruit Quest application.

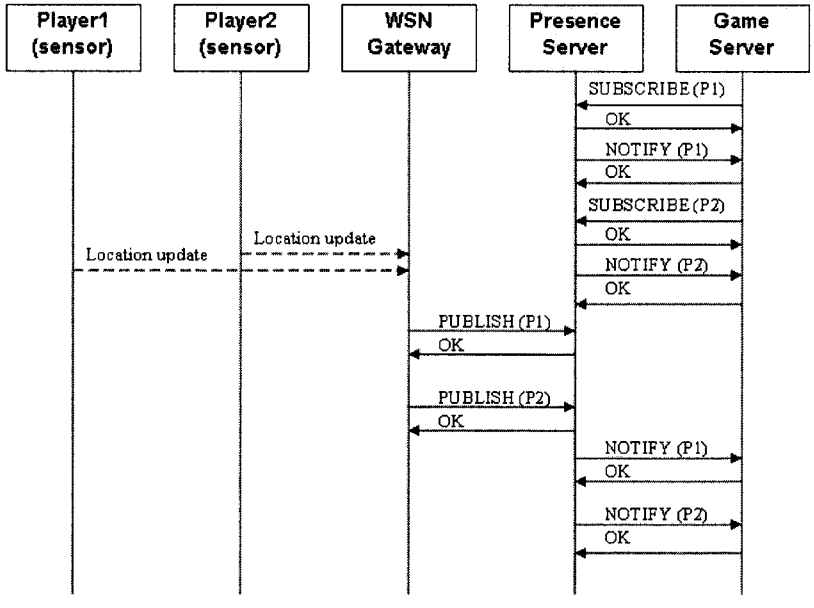
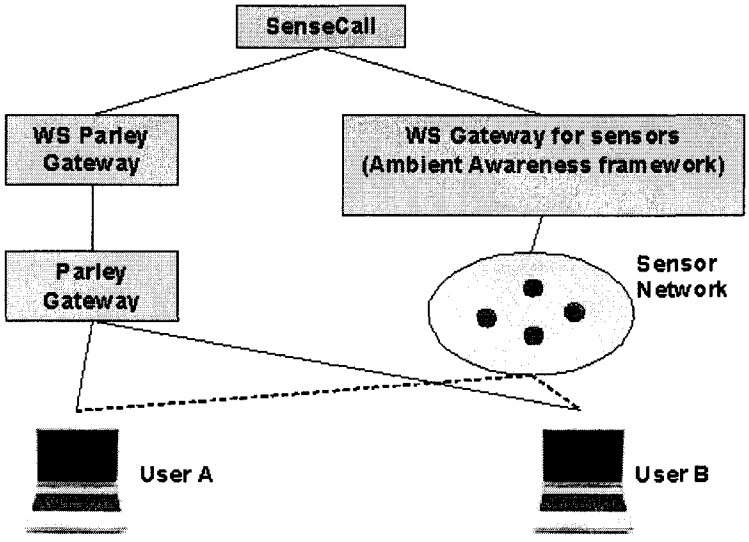


Figure 5.4: Information flow of Fruit Quest

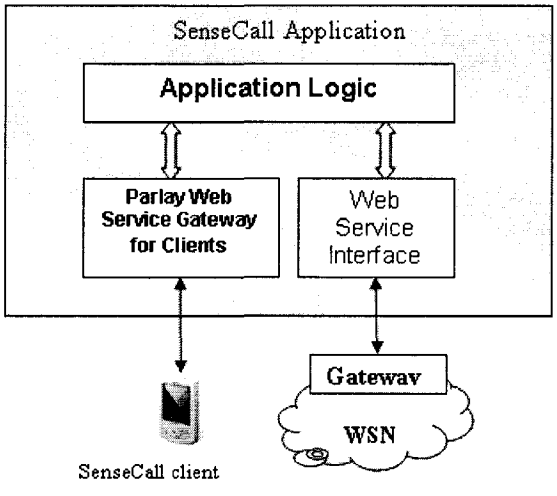
5.1.2. SenseCall Application

MOTIVATION: The call services for example two party calls or multiparty calls are one of the traditional services provided by telecommunication industry. Traditional services have focused only on audio/video conversation between users. However the users require enhanced services like for example to get a call as an announcement about restaurants, museums nearby in his present area or to establish a call automatically when his friend or colleague is nearby or in a designated area (e.g., meeting room). These services depend on user's context (i.e. a physical data that can provided through sensors) and can be enhanced and supported through WSN and existing IMS services infrastructure (e.g. presence) to provide new value-added services to end-users. Therefore, in this section we demonstrate a third party call setup application known as SenseCall.

ARCHITECTURE: SenseCall is a third party call setup application. It establishes a call between two users when their presence (location) is detected in some designated areas. SenseCall was previously designed with a Web Service framework [54]. Figure 5.5 shows the original design of SenseCall application.



(a)

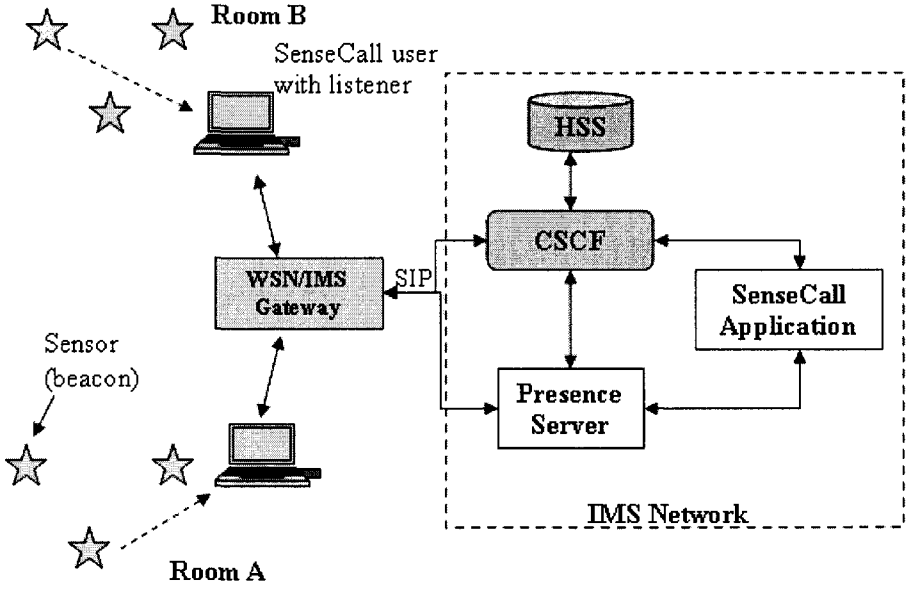


(b)

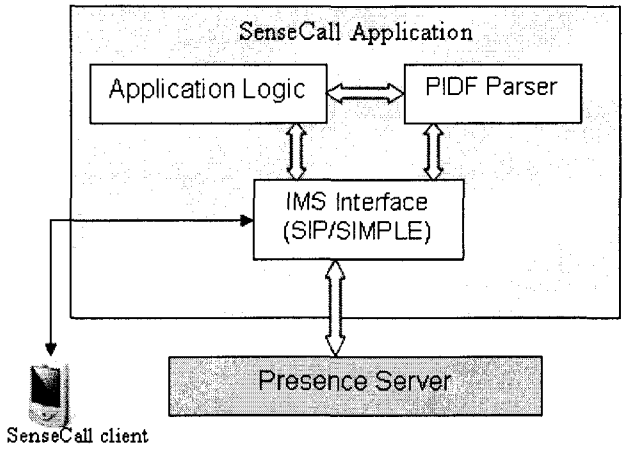
Figure 5.5: (a) Original SenseCall system architecture, (b) original SenseCall application architecture

The original design was based on web service infrastructure in which the sensor gateway implements a web service interface. It enables applications on web to access services (WSN information) provided by WSN gateway. While IMS provides SIP based multimedia services to mobile user. Therefore, we customized the whole architectural

design by replacing the web service gateway for WSN with WSN-IMS gateway developed as a part of integrated WSN and IMS architecture. The complete set of web service interfaces including Parlay web service interface is replaced with IMS SIP/SIMPLE interface. The presence server is implemented and deployed as an IMS application server that is acting as a service enabler to provide presence (context) information. In design the WSN-IMS gateway implements a SIP/SIMPLE interface and the SenseCall application logic is modified to compatible with IMS (SIP/SIMPLE) interface. Additionally, the PIDF XML parser is implemented in SenseCall application to process sensor data in PIDF form. The application (SenseCall) interacts with presence server to access WSN services provided by WSN-IMS gateway. The SensCall application is deployed as an IMS server application. Figure 5.6 shows the SenseCall application architecture.



(a)



(b)

Figure 5.6: (a) SenseCall customized system architecture, (b) modified SenseCall application architecture

WORKING: In this application scenario two rooms are chosen (A and B) and are equipped with cricket location sensors (beacons). The SenseCall application is deployed as an IMS SIP application server in Ericsson SDS platform. Each user's PC is equipped with a cricket listener sensor that is roaming in WSN environment. The beacons transmit location information while listener sensor receives data from beacons. The data received from beacons shows the current location of listener (user). The listener reports user's location information to gateway. The gateway will then publish each user's location information to Presence server. On the other hand the SenseCall application as a watcher monitors the user's location by subscribing to users location information from presence server. So, whenever both users presence (location) is detected in their respective areas it will send a SIP REFER message to one of the user so as to setup a call between them. Figure 5.7 shows the information flow of SenseCall.

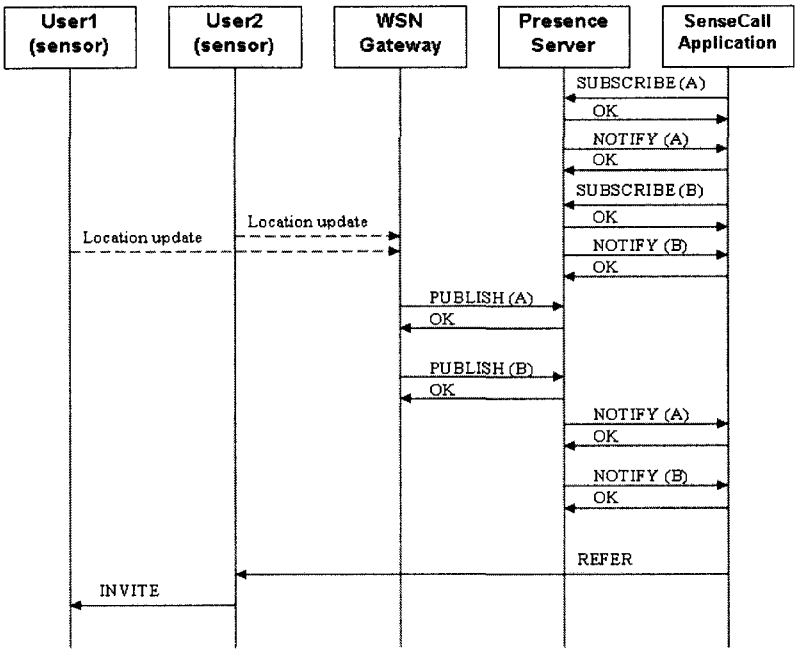


Figure 5.7: SenseCall information flow

5.2. Performance Evaluation

In this section we discuss a preliminary performance evaluation of integrated WSN and IMS architecture.

5.2.1. Performance Metrics

The experiments were conducted using the two prototype applications. The intent of performance evaluation is to examine the efficiency and feasibility of presence-based integrated WSN and IMS architecture. In IMS, the services/applications have different performance requirements such as delay, bandwidth requirements etc to provide agreed quality of service to end-user. One of the performance metrics considered is response time i.e. the elapsed time between the sending of a service request and the reception of a response. From the end-user perspective, a user expects timely responses to their service

requests within a given time constraint for delay sensitive services/applications such as voice, instant messaging. On the other hand, WSNs exchange substantial amount of data with applications, therefore it is important to measure the network load in this case. The response time and network load are the two metrics we considered and are measured using SIP messages exchanged for accessing WSN services (data) between WSN-IMS gateway and IMS services/applications.

5.2.2. Test Bed

To evaluate the performance of the overall system, we take measurements of IMS (SIP) messages exchanged in presence operations i.e., publication and notification between WSN and IMS. The response time for each request-response was calculated with a pair of timestamps recorded when the request was sent and a desired response received. We measured response times (RT_P, RT_R) for two different information exchange scenarios: proactive and reactive. The information exchange based on publication and notification transactions in which the sensed data is first published to PS and then notified to services/applications. The 'proactive' mode publishes data on regular interval (e.g., 15 sec) or on events. While 'reactive' mode publishes data upon request from presence server. In case of 'proactive' publication the response time (RT_P) is measured when the SIP PUBSLIH request is sent and the corresponding response is received. In reactive publication the response time (RT_R) is calculated when OPTIONS request is sent and the corresponding PUBLISH message received. Similarly the response time (RT_N) for information notification is measured in the same way as for publication. For the notification the watcher is first subscribed once to a number of publishers (WSN entities).

Figure 5.8 shows the test flow for two information exchange scenarios (proactive and reactive).

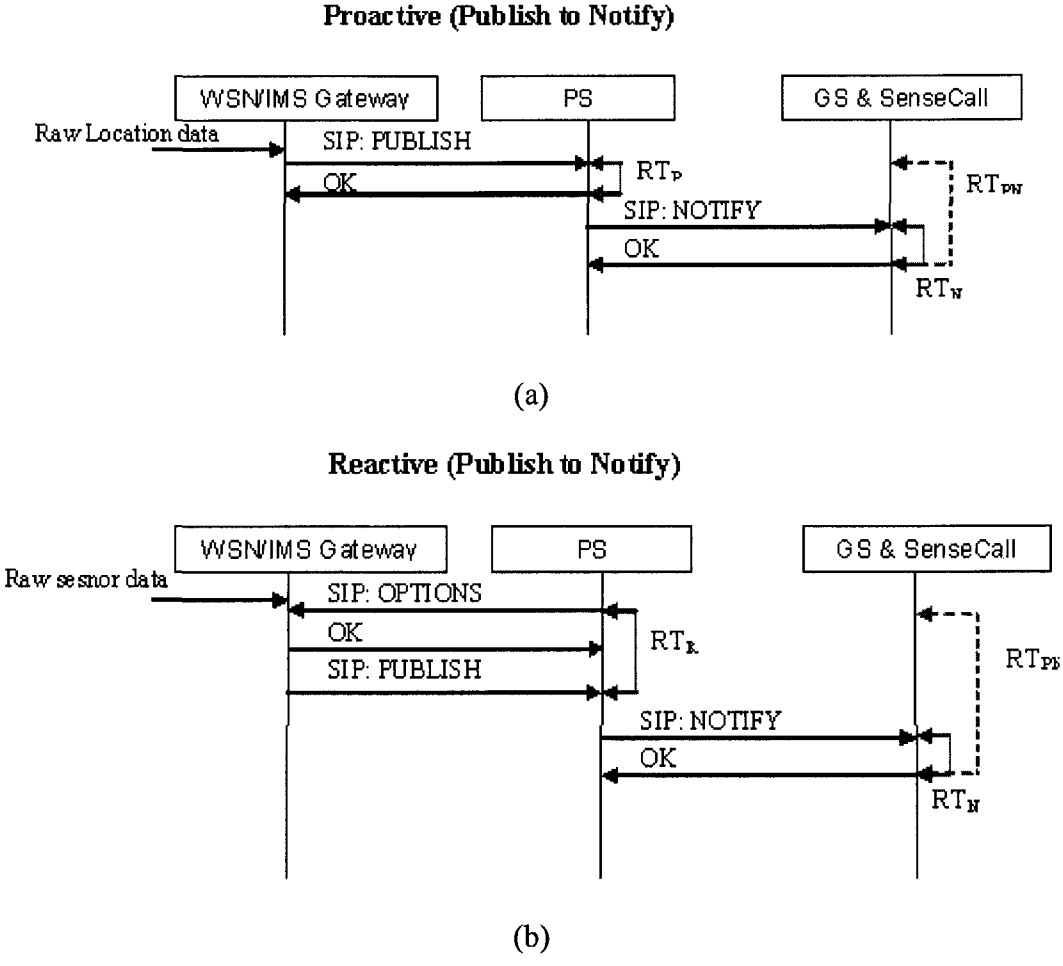


Figure 5.8: Test flow for information exchange scenarios (a) proactive (b) reactive modes.

The experimental setup consists of MIT cricket location sensors for location data and Crossbow MTS300 for environmental data. The CSCFs, Presence Server, Game Server and SenseCall application are deployed in Ericsson SDS 3.1 IMS emulated environment. The WSN-IMS gateway is deployed as a standalone IMS SIP user agent on a separate PC. The Table 5.1 shows the testbed specifications.

Due to the lack of hardware resources in our lab we have done mix of both real and simulated performance evaluation. In real environment we deploy few sensor nodes in a physical environment and take performance measurements accordingly while in simulated approach we wrote a small program that generates the simulated sensor data and report it to WSN gateway. The WSN gateway then publishes simulated data to PS in the same fashion as for real sensor data. Each running instance of a program represents a sensed data from a single sensor and we have run number of instances of this program in order to have multiple simulated sensors data for evaluation.

Laptop 1	WSN-IMS Gateway Intel Pentium 4 with windows XP, MoteView 2.4 Ghz/ 512MB/802.11a/g Wi Fi along with MIB510 sink node.
Laptop 2	CSCFs and Presence Server Intel Pentium Duo with windows XP, Ericsson SDS 1.6Ghz/1GB/802.11a/g Wi Fi.
Laptop 3	SenseCall and Game Server application Intel Pentium 4 with windows XP, Ericsson SDS 2.4 Ghz/ 512MB/802.11a/g Wi Fi.
Laptop 4,5,6	WSN Clients with attached cricket listeners (for FruitQuest and SenseCall) Intel Pentium 4 with windows XP, Cricketlib, Cygwin 2.4 Ghz/ 512MB/802.11a/g Wi Fi.
Cellphones	Sony Ericsson P990i Cellphones – (IMS mobile clients)
Sensors Kit	Cricket location Sensors (beacon and listener modes)
	Crossbow MTS300/MIB510 sensor nodes

Table 5.1:Testbed hardware specification

5.2.3. Measurements and Analysis

We have taken measurements to compare two information exchange scenarios (proactive and reactive) and for each scenario the readings are taken for publication and notification transactions. The 20 test runs were executed for each scenario. Table 5.2 shows an average response time for both scenarios. The first four readings in Table 5.2 are taken from real sensors while readings from 5 to 10 sensors are virtual nodes i.e. based on simulated sensor data.

Response Time (ms)					
No. of WSN entities publishing/subscribed	**Publication**		**Notification (RT_N)**	**Total Response Time (RT_{PN})**	
	Proactive (RT_P)	**Reactive (RT_R)**		**Proactive**	**Reactive**
1	74	81	69	143	150
2	95	132	87	182	219
3	120	153	115	235	268
4	162	228	170	332	398
5	187	271	207	394	478
6	225	327	234	459	561
7	288	395	259	547	654
8	314	459	243	557	702
9	339	483	301	640	784
10	353	549	321	674	870

Table 5.2: Average response time measurement for proactive and reactive mode of information exchange.

The publication column shows the response time for proactive and reactive publication scenarios with respect to number of WSN entities (sensors) publishing. The notification column shows the response time for notification transaction. The total response time (RT_{PN}) includes publication time plus notification time as follows:

$$RT_{PN} = RT_P + RT_N \quad \text{(Proactive)}$$

$$RT_{PN} = RT_R + RT_N \quad \text{(Reactive)}$$

The total response time for reactive mode is higher than proactive mode because of the processing of trigger requests i.e. SIP OPTIONS message plus data access time for publication. The response time increases with the increase in number of WSN entities information exchanged and the maximum increase in case of proactive mode is 68% while for reactive mode is 87%. The Figure 5.9 shows the same results in the form of a graph.

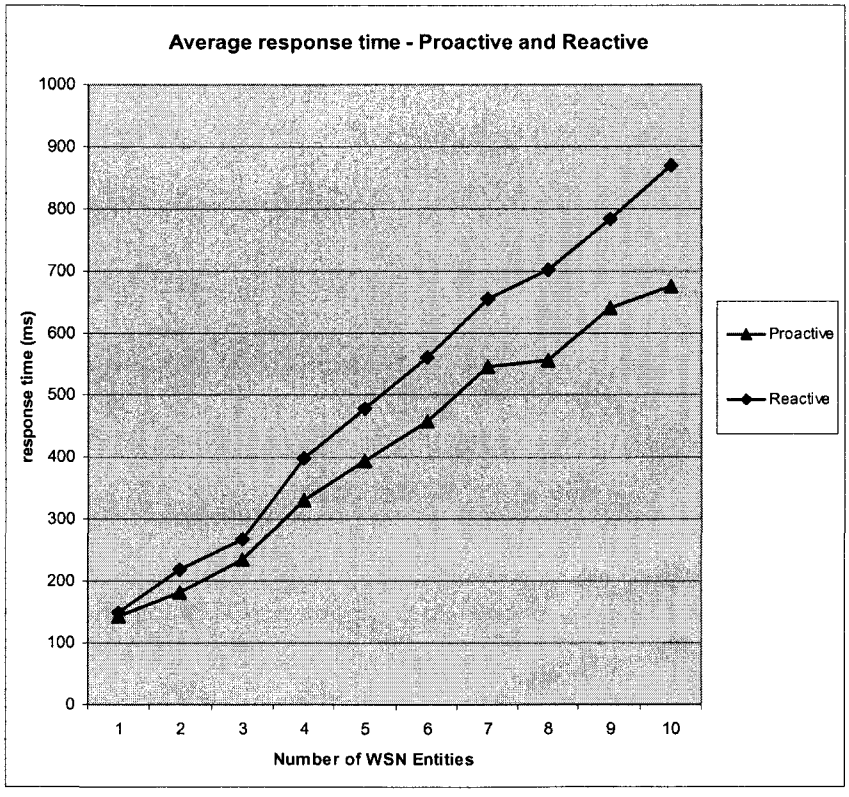


Figure 5.9: Response time for proactive and reactive modes of information exchange

Table 5.3 shows an average network load for proactive and reactive mode of publication and notification. Ethereal protocol analyzer is used for the measurement of network load.

Network Load (bytes)					
No. of WSN entities publishing/subscribed	**Publication**		**Notification**	**Total Network Load**	
	Proactive	**Reactive**		**Proactive**	**Reactive**
1	1263	2685	1454	2717	4139
2	2596	5433	2996	5592	8429
3	3934	8205	4536	8470	12741
4	5263	10971	6088	11351	17059
5	6668	13747	7656	14324	21404
6	7997	16493	9207	17204	25699
7	9326	19238	10715	20041	29953
8	10655	21984	12258	22913	34242
9	11984	24730	13747	25731	38477
10	13313	27478	15315	28629	42791

Table 5.3: Average network load measurement for proactive and reactive mode of information exchange.

The results show that proactive publication generates less network load than reactive publication. The high network load in case of reactive mode is due to two types of message exchange (OPTIONS and PUBLISH) as shown in Figure 5.8 including payload. The payload size of PUBLISH and NOTIFY messages are range from 428 to 500 bytes carrying PIDF data (e.g. environmental or location). The network load difference

between proactive and reactive for single WSN entity publishing is 1422 bytes and this difference increases with the multiples of WSN entities publishing e.g. for two WSN entities publishing the difference becomes 2837 bytes, and for three and four entities publishing are 4271 and 5708 bytes respectively. This effect can be seen from Figure 5.10 that this difference between the proactive and reactive line becomes wide with varying number of WSN entities. Thus, in terms of network load the reactive mode of information exchange will be inefficient compared to proactive if there are numerous WSN entities exchanging information. The network load varies depending on the size of SIP message that contains varying number of headers (overhead) and message body (payload) attached with SIP. In case of SIP PUBLISH message the data portion (payload) contributes 58% while header portion contributes 42% of total message size whereas NOTIFY message contributes 48% of payload data and 52% of headers portion. It is clear from the above two analysis that proactive mode has better performance over reactive mode and would be an efficient option for information exchange.

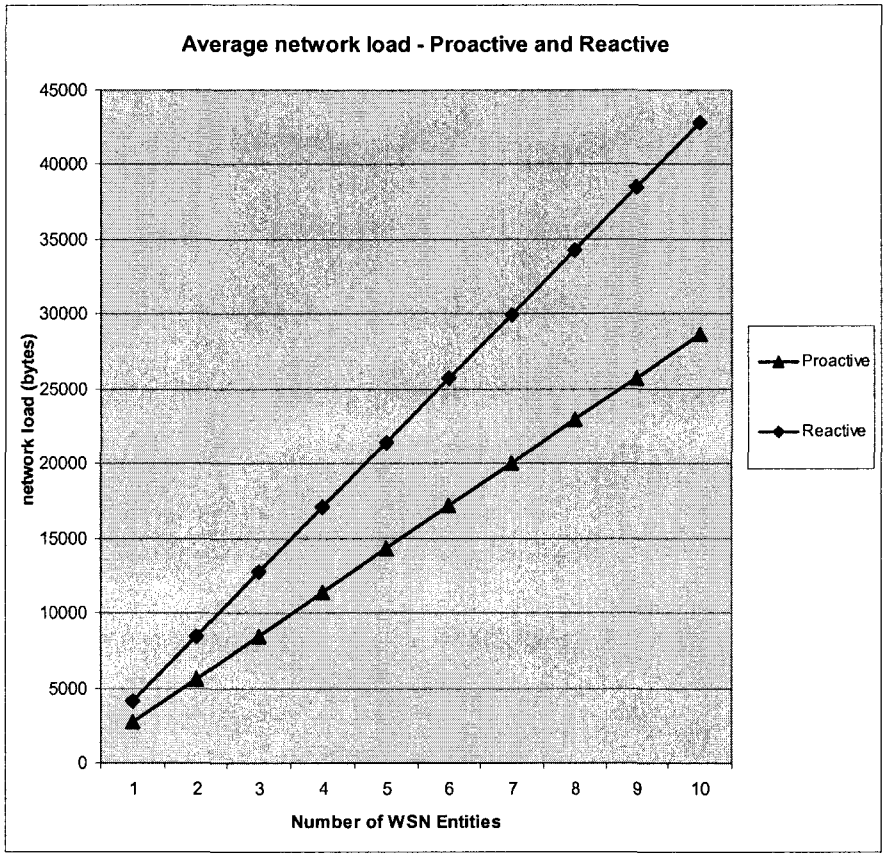


Figure 5.10: Response time for proactive and reactive modes of information exchange

Also the footprint of the above applications was reduced in the context of our integrated WSN-IMS architecture. In SenseCall application the web service based implementation consists of 639 lines of code (loc) while IMS based application implementation has only 396 loc.

5.3. Chapter Summary

In this chapter we demonstrated the proof of concept of an integrated WSN-IMS architecture by implementing two prototype applications: FruitQuest and SenseCall. The presence based integrated WSN – IMS architecture provides an advantage to service

developers which do not need to have a prior expertise in WSN domain to implement any new IMS service/application based on WSN. IMS services/applications use presence operations (e.g. SUBSCRIBE, NOTIFY) that access WSN data without implementing any proprietary WSN logic. The applications implemented in the context of our integrated WSN-IMS architecture showed better footprint over previous implementations. The performance study was conducted to show the efficiency of this architecture. The two main performance metrics 'response time' and 'network load' were considered for performance measurements. From the measurements it was analyzed that the proactive mode of information exchange has good performance characteristic over reactive mode. The response time and network load could be improved by decreasing the extra overhead (e.g. headers) carried by SIP to minimize the processing time and size of messages.

Chapter 6. Conclusion

This chapter summarizes contributions of this thesis and future work.

6.1. Thesis Contributions

The IP multimedia subsystem is a service control architecture for enabling the provisioning of a wide range of multimedia services to end-users. One important ingredient of these services is data or more specifically contextual data accessible or derived from external sources. One of the sources of these contextual data can be WSNs. As of today WSN provide proprietary interfaces and protocols to access sensor data. Therefore, entities like proxies and gateways are put in place to enable the standard interface for data exchange with other networks (e.g. IMS).

As a part of this thesis contribution, we examined the existing solution related to integration of WSN with other networks. A number of criteria had been set for the integration of WSN with IMS. These criteria were considered for the evaluation of existing solutions. As a result of evaluation we considered 3GPP presence service as a potential approach for the integration of WSN with IMS. Presence is an important IMS service that enables the development of new communication services (e.g. gaming, Instant Messaging). The main feature of a presence service is to provide user presence and contextual information to applications and other users.

The core contribution of this thesis is the implementation of the integrated WSN/ IMS architecture. We first discussed the integrated WSN-IMS architecture and then its

implementation. The integrated WSN-IMS architecture provides a communication abstraction for accessing sensor data from IMS. The main component of the integrated architecture is WSN-IMS gateway that provides a standard interface (e.g. SIP) for exchanging sensor data. The work includes mapping schemes between WSN and IMS. One of the mappings includes how to map raw sensor data to an abstract IMS format, i.e. PIDF. Another scheme is the mapping of sensors to IMS Ids and the other way around from IMS Ids to sensors. In addition to that some extensions related to IMS presence service (in Presence Server) for accessibility of WSN data have been implemented. The extensions include mechanism to access sensor data i.e. publish upon request (reactive publication) and the presence information modeling i.e. extended PIDF schema for supporting different types of sensor data (spatial, environmental, and physiological).

As a proof of concept prototype we demonstrated two application case studies: Fruit Quest and SenseCall. These two case studies depict that the presence service based approach can act as an enabler for the integration of WSN with IMS to provide wide range of IMS services. Performance evaluation was done to see the efficiency of presence based WSN-IMS architecture. In the performance evaluation we compare the two information exchange modes i.e. proactive versus reactive to analyze the performance characteristic of presence based integrated WSN – IMS architecture. The analysis showed that in terms of both response time and network load the proactive mode of information exchange is more efficient than reactive mode.

6.2. Future work

The future work would be to implement the remaining features of the integrated WSN and IMS architecture and perform a thorough performance analysis. The features have not implemented yet are WSN service capabilities publication and policies based information exchange between WSN (gateway) and IMS entities (services or applications). WSN service capabilities publication is done by WSN-IMS gateway to presence server. For this presence server needs to be further extended to support the discovery of WSNs service capabilities published by the WSN-IMS gateway. WSN service discovery enables presence server to know different sources (WSNs) of contextual information and what type of information (location, environment, physiological) a WSN is providing. The policies based information publication and access (subscription/notification) has to be implemented in order to have a greater control of information exchange between WSN and IMS. In addition to that the performance evaluation should be done in a more realistic environment with a larger number of sensors and the gateways acting as thin user devices like PDA's and mobile phones assuming that these devices have limited resource e.g. storage, processing, and communication capabilities.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E Cayirci. "A Survey on Sensor Networks", *IEEE Communications Magazine*, vol.40(8), pp 102-114, August 2002.

[2] K. Sohraby, D. Minoli, T. Znati., "*Wireless Sensors Networks Technology, Protocols and Applications*," New Jersey, Wiley-InterScience 2007 pp 1-31, 75-80

[3] "University of California-Berkeley Motes." Internet: http://www.xbow.com [July 11th 2008]

[4] "ScatterWeb sensor platform", Internet: http://cst.mi.fu.berlin.de/projects/scatterweb/ [July 11th 2008]

[5] "Tmoke-Sky sensors", Internet: http://www.sentilla.com/moteiv-endoflife.html [July 11th 2008]

[6] "BTnodes sensors", Internet: http://www.btnode.ethz.ch [July 11th 2008]

[7] "TinyOS - sensor network Operating System", Internet: http://www.tinyos.net [July 13th 2008]

[8] 3GPP TS 22.228, "Service requirements for the Internet Protocol (IP) multimedia core network subsystem; Stage 1", September 2007.

[9] T. Drempetic, T. Ribtarec, D. Bakic. "Next Generation Networks Architecture for Multimedia Applications," in *Proc. 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, Vol. 2, pp 563-568, July 2003.

[10] M. Poikselka, G. Mayer, H. Khartabil, Aki Niemi.,*The IMS IP Multimedia Concepts and Services*, England, John Wiley & Sons 2006

[11] K. Kinghtson, N. Morita, T. Towle. "NGN Architecture: Generic Principles, Functional Architecture, and Implementation," *IEEE Communication Magazine* Vol 43(10), pp 49-56, October 2005

[12] G. Camarillo, M. A. Garcia-Martin, "*The 3G IP Multimedia Subsystem (IMS) Merging the Internet and the Cellular Worlds*," England, John Wiley & Sons 2006

[13] N. Kinder. "IMS- IP Multimedia Subsystem IMS overview and the Unified Carrier Network," *Sonus Networks*, [Online]. available at http://www.iec.org/newsletter/sep06_2/analyst_corner.pdf [July 14th 2008]

[14] 3GPP TS 23.141, "Presence Service; Architecture and functional description", September 2006.

[15] 3GPP TS 24.229, "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", September 2007. June 2008.

[16] 3GPP TS 22.141, "Presence Service;Stage 1", December 2005.

[17] J. Rosenberg. "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, IETF August 2004

[18] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. "SIP: Session Initiation Protocol", RFC 3261, IETF June 2002

[19] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, "ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring," Technical Report CS-2006-11, Department of Computer Science, University of Virginia, 2006 [Online] Available: http://www.cs.virginia.edu/wsn/medical/pubs.html [June 2008].

[20] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek, R. Dssouli, "Integrating the Sensing Capabilities of Wireless Sensor Networks in the IP Multimedia Subsystem: A Presence-based Approach", [Submitted Journal] in *IEEE Communication Magazine series on Adhoc & Sensor Networks*, [under review].

[21] "MIT Cricket Sensors", Internet: http://cricket.csail.mit.edu [July 11th 2008]

[22] "TinyDB", Internet: http://telegraph.cs.berkeley.edu/tinydb/index.htm [July 12th 2008]

[23] T Luckenbach, P Gober, S. Arbanowski, "TinyREST - a Protocol for Integrating Sensor Network into the Internet", in *Proceeding of REALWSN 2005*, June 2005.

[24] K. Mayer, W. Fritsche, "IP-enabled Wireless Sensor Networks and their integration into the Internet", in *Proc. of the First International Conference on Integrated Internet Ad-Hoc and Sensor Networks (INTERSENSE06)*, Article No. 5, Nice, France, May 2006

[25] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, IETF August 2004

[26] S. Krishnamurthy, "TinySIP: Providing Seamless Access to Sensor-based Services", in *Proc. of IEEE Mobile and Ubiquitous Systems – Workshop*, July 2008 pp 1-9

[27] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek, R. Dssouli, "An Architecture for the Provision of Context-Aware Emergency Services in the IP Multimedia Subsystem", in *Proc. of the IEEE Vehicular Technology Conference*, May 2008, pp 2784-2788.

[28] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek, R. Dssouli, "A Presence-based Architecture for the Integration of the Sensing Capabilities of Wireless Sensor Networks in the IP Multimedia Subsystem", in *Proc. of the IEEE Wireless Communications and Networking Conference*, April 2008, pp 3116-3121.

[29] H. Schulzrinne. "The tel URI for Telephone Numbers", RFC 3966, IETF December 2004

[30] 3GPP TS 29.228, "IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signalling flows and message contents", June 2008.

[31] 3GPP TS 23.218, "IP Multimedia (IM) session handling; IM call model; Stage 2", June 2008.

[32] 3GPP TS 24.228, "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", September 2006.

[33] H. Schulzrinne, V. Gurbani, P. Kyzivat, J. Rosenberg, "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", IETF RFC 4480, July 2006

[34] H. Schulzrinne, "CIPID: Contact Information for the Presence Extensions to the Presence Information Data Format (PIDF)", IETF RFC 4482, July 2006

[35] H. Schulzrinne, "Timed Presence Extensions to the Presence Information Data Format (PIDF) to Indicate Status Information for Past and Future Time Intervals", IETF RFC 4481, July 2006

[36] A.B. Johnston, "*SIP Understanding the Session Initiation Protocol*", 2nd Edition, Boston, Artech House, 2004

[37] "SIP: Protocol Overview," Internet: http://www.radvision.com [Oct 2nd 2008].

[38] M. Zhang, S. Pack, K. Cho, D. Chang, Y. Choi, and T. Kwon, "An Extensible Interworking Architecture (EIA) for Wireless Sensor Networks and Internet," in *Proc. Asia-Pacific Network Operations and Management Symposium (APNOMS)*, September 2006.

[39] A. Gluhak, M. Presser, Z. Shelby, P.Scotton, W. Schott, P. Chevillat, "e-SENSE Reference Model for Sensor Network in B3G Mobile Communications Systems", 15th IST Summit 2006, Myconos Greece Workshop [Online] Available: http://www.ist-esense.org/index.php?id=132 [July 24th 2008].

[40] "Initial e-SENSE System Architecture", Deliverable# D2.2.1 page 84-92 [Online] Available: http://www.ist-esense.org/index.php?id=33 [July 24th 2008].

[41] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, J. Peterson, "Presence Information Data Format (PIDF)", IETF RFC 3863, August 2004.

[42] J. Peterson, "A Presence-based GEOPRIV Location Object Format", IETF RFC 4119, December 2005.

[43] "JAIN SIP API Specification", JSR 32, Internet: http://www.jcp.org/en/jsr/detail?id=32 [August 18th 2008]

[44] M. Ranganathan, "JAIN-SIP: Architecture, Implementation, Testing," Invited talk at SIP Summit, Las Vegas, NV, May 7-9, 2002 [Online] http://w3.antd.nist.gov/ [August 18th 2008].

[45] "SIP Servlet API Specification", JSR 116, Internet: www.jcp.org/en/jsr/detail?id=116 [20th August 2008]

[46] "Ericsson Service Development Studio (SDS) 4.0 Technical Product Description" Internet: http://www.ericsson.com/mobilityworld/sub/open/technologies/ims_poc/docs.htm [August 20th 2008]

[47] "Cricket User Manual", Internet: http://cricket.csail.mit.edu [August 22nd 2008]

[48] "IBM Webshpere Presence Server" Internet: http://www-01.ibm.com/software/pervasive/presenceserver/ [August 20th 2008]

[49] "Jabber XCP" Internet: http://www.jabber.com/CE/JabberXCP [August 20th 2008]

[50] "Crossbow Sensor Products Overiew", Internet: http://www.xbow.com/Products/wproductsoverview.aspx [August 20th 2008]

[51] "MoteView Client Application", Internet: http://www.xbow.com/Technology/UserInterface.aspx [August 21st 2008]

[52] P. O'Doherty, "SIP Specification and the Java Platforms", [Online]: Available http://java.sun.com/products/jain/SIP-and-Java.html. [August 22nd 2008]

[53] F. Akber, J. Carrier, M. Watfa, A. Qureshi, "Integration of IMS and WSN to design a Pervasive game", B.Eng. Final Year Project Report, Concordia University, Montreal, March 2007.

[54] Truong Ta, "Web Services for the Dissemination of Ambient Information to I-centric Applications", Masters Thesis, Electrical and Computer Engineering Dept. Concordia University, Montreal, December 2005.

[55] Nuru Yakub Othman, "Web Services as application enabler for Sink-less Wireless Sensor Networks", Masters Thesis, Electrical and Computer Engineering Dept. Concordia University, Montreal February 2007

[56] 3rd Generation Partnership Project, Internet: http://www.3gpp.org [July 13th 2008]

[57] 3GPP TS 29.240, "3GPP Generic User Profile (GUP),Stage 3, Network", June 2007

[58] C. Magerkurth, A. D. Cheok, R. L. Mandryk, T. Nilsen, "Pervasive games: bringing computer entertainment back to the real world", *ACM Transactions on Computers in Entertainment (CIE), Section: Pervasive Gaming*, Vol. 3, Issue 3, July 2005, pp 4-4.