Integrating Wireless Sensor Networks and Mobile Ad-hoc NETworks for enhanced value-added services

Saba Hamedi

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

Montréal, Québec, Canada

August 2010

Your file Votre référence
ISBN: 978-0-494-71013-5
Our file Notre référence
ISBN: 978-0-494-71013-5

ABSTRACT

Integrating Wireless Sensor Networks and Mobile Ad-hoc NETworks for enhanced value-added services

Saba Hamedi

In some situations where the standard telecommunication infrastructure is not available, Mobile Ad hoc NETworks (MANETs) can be deployed to provide the required communication. These networks are established "on the fly" without a need for prior communication organization and are composed of autonomous mobile devices, such as cell phones, PDAs or laptops. In similar conditions, such as in emergency response operations, integrating MANETs and Wireless Sensor Networks (WSNs) can notably enhance the MANET participant's end-user experience. WSNs sense and aggregate ambient information, such as physiological, environmental or physical data related to a nearby phenomenon. The integration, which provides end-user availability to WSN required information, is feasible via gateways. However, when the ambient information collected by WSNs is intended for applications residing in MANETs, centralized and fixed gateways are not practicably feasible. This is mainly due to ad-hoc nature, lack of centralized control and constraints on the end-user devices that are used in MANETs. These devices are usually limited in power and capacity and cannot host centralized gateways.

In this thesis we exploit the integration of WSN and MANET in order to provide novel value-added services which enhance the end-user experience of MANET participants. Motivating scenarios are introduced, background information is presented, requirements are derived and the state of the art regarding the integration of WSN with existing networks, including MANETs, is evaluated. Based on the evaluation, none of the existing solutions satisfies all of our derived requirements. Therefore, we propose an overall two-level overlay architecture to integrate WSNs (with mobile sinks) and MANETs. This architecture is based on the distributed gateway and applications which form the P2P overlays. Overlays are application-layer networks which are created on top of the exiting MANET. To interconnect gateway and application overlays we derive corresponding requirements and evaluate the existing approaches. Since none of these approaches fulfills all of our requirements, we propose protocols, mechanisms and design corresponding modules for the interconnection of overlays. Finally we refine our overall architecture based on the interconnection aspects. As a proof of concept, we implement a prototype for the inter-overlay information exchange. This implementation is based on SIP extensions and uses two existing P2P middlewares. We also simulate our prototype using Oversim simulation tool and collect experimental results. Based on these results, we can see that our architecture is a valid and promising approach for interconnecting different P2P overlays and can be deployed to provide the overall solution for WSN and MANET integrated system.

ACKNOWLEDGMENTS

This thesis is a part of research work done in the Telecommunication Service Engineering (TSE) research lab. It is a pleasure to thank the people who have helped me and made completion of this thesis possible. First, I would like to express my sincere gratitude to my supervisor, Dr. Ferhat Khendek, for his encouragement, valuable advice and continuous support. His critical comments, considerate attitude and immense knowledge helped me a lot during both research and thesis-writing periods. My special thanks to Dr. Roch Glitho whom I was delighted to interact with as my co-supervisor and professor. His enthusiasm and expertise in both teaching and research and his patience and willing to help his students, made the research life interesting and rewarding for me. His insightful guidance and constructive ideas were vital for the accomplishment of this thesis.

I am also grateful to my co-worker and friend, Majid Hormati, for his valuable contribution in this research work. His determination was always a source of motivation during these two years and his generosity to share his knowledge and skills with me means a lot to me. My thanks and appreciations go to my supervisory committee members Dr. Qiu and Dr. Desai for taking the time to assess my thesis and their helpful suggestions. I am grateful to Dr. Mustafa K. Mehmetali, my professor at Concordia University, for encouraging and enlightening me the first glance of research. I would like to acknowledge the financial support from Ericsson Canada Inc., Natural Sciences and Engineering Research Council of Canada (NSERC) and Concordia University.

Many thanks to my dear husband, Nicolas, for his love, constant cheering and understanding which provide me the best condition to concentrate on my research. Merci pour tout.

And finally, I wish to thank my parents for providing me with the opportunity to be where I am.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Acronyms and Abbreviations

3G	Third Generation wireless system
3GPP	Third Generation Partnership Project
APPIC	APPlication InterConnector node
AppC	Application Component
AODV	Ad-hoc On-demand Distance Vector routing protocol
API	Application Programming Interface
BSN	Body Sensor Network
CAN	Content Addressable Network
DARPA	Defense Advanced Research Projects Agency
DHT	Distributed Hash Table
DSL	Digital Subscriber Line
DSR	Dynamic Source Routing
GSM	Global System for Mobile communications
GWIC	GateWay InterConnector node
GWC	GateWay Component
H-P2PSIP	Hierarchical Peer to Peer Session Initiation Protocol
HTTP	Hyper Text Transport Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IPV4	Internet Protocol Version 4
JAIN	Java API for Integrated Networks
MANET	Mobile Ad-hoc NETwork
MSWSN	Mobile Sink Wireless Sensor Network
OLSR	Optimized Link State Routing protocol

PAN	Personal Area Network
PDA	Personal Digital Assistant
PRNet	Packet Radio Network
PSAP	Public Safety Answering Point
PSE	Presence Service Entry point
P2P	Peer to Peer
P2PSIP	Peer to Peer Session Initiation Protocol
QoS	Quality of Service
RF	Radio Frequency
RPC	Remote Procedure Call
SAPPIC	Super APPlication InterConnector node
SDM	Super Data Management
SDP	Session Description Protocol
SEP	Sink Entry Point
SGWIC	Super GateWay InterConnector node
SIP	Session Initiation Protocol
SSEP	Super Sink Entry Point
SPSE	Super Presence Service Entry point
TBRPF	Topology dissemination Based on Reserve-Path Forwarding
TCP	Transport Control Protocol
TTL	Time To Live
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VPN	Virtual Private Network

WAN	Wide area Ad-hoc Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
WWBAN	Wearable Wireless Body Area Network
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Research Domain

Mobile Ad-hoc NETworks (MANETs) are transient networks formed dynamically by a collection of heterogeneous and arbitrarily located wireless mobile nodes without relying on any existing infrastructure or centralized administration. They are particularly useful in emergency situations, such as wars and natural disasters, where the existing telecommunication infrastructure is destroyed, saturated or impossible to settle [1]. In such situations, MANETs' main advantages are their ease of deployment, fault tolerance, scalability in terms of the number of participants as well as low cost and spontaneous establishment of communication services. Recent enhancements in mobile technology, including wireless communication and mobile handheld devices, allow a vast deployment of MANET applications. Some of the actual applications of MANETs are in battlefield operations, emergency medical health care and disaster recovery, business group and inter-vehicle communication. However, these deployments introduce challenges at different levels. At the lower layers, such as physical and network layers, stability and reliability of wireless links and efficient dynamic routing protocols which consider frequent changes in MANET topology are some of the actual research subjects. However, there are limited works which focus on the application layer related issues and provision of general substrates for development of applications in MANET.

Wireless Sensor Networks (WSNs) are sets of distributed nodes that collaborate to monitor physical, environmental or physiological conditions. They are made up of sensors that do the actual sensing by being densely scattered over an area of interest, aggregators that aggregate information, and sinks and gateways that collect the sensors' data and enable communication with the end-user applications. WSNs are deployed for a wide range of application domains, including health and environmental monitoring, military supervision, and commercial applications, such as home automation. Information they collect is known as ambient information and can significantly enhance the end-user experience [2]. In health monitoring, for instance, the physiological status of patients can be tracked using the ambient information collected by WSNs. In many situations, the information is relayed to a fixed and centralized gateway which will then transmit it to end-user applications residing in infrastructure-based networks, such as 3G. Many value-added services can be developed based on the provided data. WSN research domain interests both academia and industry by introducing many challenging issues in hardware, networking and application levels. Design of efficient routing protocols for power consumption optimization of sensor nodes, reliability, robustness and security of the network, distributed data processing, mobility and dynamicity of different WSN entities are some of the topics which are currently issues of extensive research.

Peer to peer (P2P) overlays are logical application-layer networks built on top of the real networks, such as MANETs. In these networks, nodes, known as peers, communicate and collaborate with each other in a distributed and ad hoc manner without using a centralized control point [3]. Many services and applications, such as file sharing or multimedia conferencing, can be developed using the overlays' provided features. Distributed computation, storage and retrieval of information and reliable, scalable and robust routing architectures are some of the features provided by the overlays.

1.2 Problem Statement and Contributions of this Thesis

As mentioned before, MANETs are deployed in situations where the existing communication infrastructure is not available. Integrating WSN ambient information with MANETs, can provide a wide variety of value-added services to the MANET application end-users. A very common approach to integrate WSN and other networks is to deploy centralized and fixed gateways. However, when ambient information collected by WSNs destined to end-user applications which reside in MANETs, using centralized and fixed gateways is not practicably a feasible approach. This is due to the fact that MANETs are infrastructure-less networks; participants can leave and join the network anytime, therefore none of the involved entities can operate in a centralized manner. Moreover, since the end-user devices used in MANETs are limited in processing and power resources, they cannot host centralized and fixed gateways. We believe that integrating WSN ambient information in MANET based on the standardized protocols leads to the provision of many novel value-added services. To illustrate the potential benefits that can be obtained from this integration, we introduce two motivating scenarios in medical and military domains [4].

Let us imagine a large-scale disaster, such as an earthquake occurring in a city. In this situation, the existing infrastructure may be damaged or destroyed. Victims need to be evacuated and require intervention of paramedics. Because a hospital is either too far from the area or it is already overwhelmed from the disaster, the paramedics decide to provide on-site examinations of the patients.

WSNs are formed by installing physiological sensors which continuously monitor the patients' health situation, such as their body temperature or heart rate. In order to quickly and efficiently establish emergency medical services and due to the lack of infrastructure, a MANET is formed between the paramedics through IEEE 802.11 wireless interfaces. This MANET is then connected to the networks of hospitals with specialized doctors. In order to integrate WSN and MANET domain, the end-user devices, such as PDAs or other handheld devices carried by paramedics, collectively act as distributed and mobile gateways towards the applications residing in the MANET and that are distributed over the very same end-user devices. This is due to the non-feasibility of fixed and centralized gateways. Gateway devices can gather, aggregate, and transmit ambient information collected by the WSNs when the paramedics are walking in the disaster area. From an end-user application point of view, each paramedic may subscribe to specific emergency events related to each patient in order to receive alerts, such as instant messages, on their health status. In addition, they may need some interactions between each other, as well as instructions from the specialized doctors in the hospital. Therefore, a multimedia conferencing session may be established as the result of a trigger being hit, such as an urgent situation, based on WSN sensed data or on demand. Patients' information is captured, displayed and shared in real-time for remote interactive consultations. Remote specialized doctors can give audio or video guidance to the paramedics, based on the WSN ambient information, to assist them with the situation.

Integration of WSN and MANET can also be beneficial in military operations. During a war in an area with no infrastructure available, such as mountains or rural areas, existing state of military troops needs to be monitored and intercommunicate. A MANET is established between the interested troop members to provide this communication. Environmental and location sensors are installed on the troop members and equipments and form corresponding WSNs. A subset of soldiers and tanks equipped by sink and gateway devices gather, aggregate and translate the sensed data and integrate the WSNs with the MANET. Based on the current state of affairs, some notification may be sent to the interested troop leaders to inform of a triggered situation. Moreover, forces moving through the area require connectivity to each other in order to analyze the situation and decide on further actions in more efficient manner. In this case a multimedia conferencing session may be established between soldiers (on the ground, tank or helicopter) and troop leaders to provide voice commands to the soldiers in order to complete an assigned mission based on the video or audio information generated from the installed sensors' information [4].

As it can be seen from the scenarios, integration of WSN and MANET introduces some challenges. The first one is related to the heterogeneity and resource constraints of MANET end-user mobile devices which are acting as both application and gateway entities. Scalability is also another challenge to take into consideration. Solution should be applicable for a large number of MANET participants. The integration architecture should also support distributed, transient nature and frequent changes in MANET topology.

To the best of our knowledge, integration of WSNs and MANET is still an open research topic and this thesis constitutes a new contribution in this domain. The focus of this thesis is to propose an architecture for integration of WSNs (with mobile sink) and MANETs to provide novel value-added services which enhance the end-user experience of MANET participants. For this integration, we propose a P2P gateway and application overlay-based approach. We also tackle the challenging problem of interconnecting P2P application and gateway overlays of the overall architecture. Moreover, deployment and extending standard protocols, such as Session Initiation Protocol (SIP) [5], for the interconnection purpose is addressed in this thesis.

The main contributions of this thesis are:

- Requirements for the integration of WSN and MANET: We derive general and gateway and application specific requirements which identify our main challenges for the integration of WSN and MANET.

- A detail survey and evaluation of the state of the art on the integration of WSNs with existing networks, including MANETs.

- A two-level overlay-based architecture for the integration of WSN and MANET: We propose an overall architecture based on P2P gateway and application overlays which are built on top of the MANET end-user devices. This architecture meets all of our requirements.

- Requirements for the interconnection of application and gateway overlays: Based on our proposed architecture, we need to interconnect the application and gateway overlays of the overall architecture.

- A detail review and evaluation of the existing work on the interconnection of overlays.

- Refinement on the overall architecture by introducing the interconnection of application and gateway overlays aspects: Principles, protocols and related mechanisms for the interconnection of overlays of the overall architecture are discussed in detail.

- Design of the interconnection modules and implementation of a prototype as a proof of concept for the overlay interconnection based on the extensions of Session Initiation Protocol (SIP).

- A preliminary performance evaluation of the overlay interconnection architecture by using Oversim [6] simulation tool.

1.3 Organization of the Thesis

The structure of the thesis is as follows. Chapter 2 presents the background information which is necessary for the understanding of this thesis. We provide an overview on WSN, MANET, P2P overlays and SIP.

Chapter 3 discusses the derived requirements for the integration of WSN and MANET. It also provides a detailed review of the state of the art on the integration of WSN with other networks, including MANETs, and evaluates them with respect to the derived requirements.

Chapter 4 proposes a two-level overlay architecture for the integration of WSN and MANET. This overall architecture is based on P2P gateway and application overlays built on top of the MANET end-user devices. In this stage we face the challenging issue of interconnecting overlays. We therefore derive the requirements for this interconnection, review the existing approaches and evaluate them. We then refine our overall architecture by introducing the interconnection of gateway and application overlays related aspects. The principles, protocols, mechanisms and operational procedures are presented and discussed.

Chapter 5 is devoted to the design of interconnection modules, implementation of our prototype as a proof of concept for the interconnection of overlays based on SIP extensions. The preliminary performance evaluations and related measurements are also presented through the Oversim simulation tool. The simulation configuration and performance metrics are discussed and obtained results are analyzed.

Conclusions on the accomplished work in this thesis and potential open issues that can be addressed and investigated in the future are discussed in Chapter 6.

Chapter 2

Background on Wireless Sensor Networks, Mobile Ad-hoc NETworks, Peer to Peer overlays and Session Initiation Protocol

In this chapter, we discuss three major subjects including the Wireless Sensor Networks, Mobile Ad-hoc NETworks and P2P overlay networks. In each section, definitions, main characteristics, applications, architecture, protocols, standards and research challenges regarding each domain are discussed. We also discuss the Session Initiation Protocol which we deploy later as the implementation technology. This chapter will help to a better comprehension of the thesis contribution.

2.1 Wireless Sensor Networks

2.1.1 Introduction and applications

A Wireless Sensor Network (WSN) composed of a large set of wirelessly interconnected and randomly deployed tiny electronic devices called sensors. Each sensor contains sensing, processing, communication and an eventual storage units. The main task of sensor nodes is to sense the data related to a nearby phenomenon such as temperature of an environment or physiological data of human body. This raw sensor information is then converted to electrical signals and is processed by sensor's processing unit to enable an efficient transmission of the required information. Processed data may be then stored in the sensors' storage unit or forwarded directly to one or several nodes responsible for collecting the information via the communication unit. Ultimately, the whole procedure will provide a more comprehensive and better understanding of the monitored environment to WSN application end-user.

WSNs can be categorized as a type of wireless ad hoc network however some specific features and technical requirements differentiate these two networks and imply new protocols and algorithms for WSN domain. First, WSN contains several thousands of autonomous nodes which are densely distributed over an area of interest. This order of magnitude is much more than the typical ad hoc networks. Second, being often deployed in open areas, sensor nodes may frequently fail due to environmental conditions and physical attacks. Therefore, WSN topology is affected considerably and more frequently than other wireless ad hoc networks. Third, the tiny size of sensor nodes implies their constraint in power, processing and storage capabilities. In many applications, sensor nodes require to be installed in areas for a long time without being recharged. Although resource constraints may be an issue in some wireless devices it is less critical compared to sensors' requirements [2].

Depending on the type of sensors that are deployed in a WSN, these networks offer opportunities to a wide range of applications. Sensors may be thermal, acoustic, infrared, visual, seismic and etc. Therefore, they can be utilized to monitor the temperature, noise level, movements, light, speed, pressure, presence and other characteristics of predefined objects. A variety of potential applications which may require this information can be categorized into health, medical and emergency, military, environmental and commercial applications. Moreover, some specific characteristics, such as self-organizing and ad-hoc nature of WSN, result their more rapid, easier and wider deployment in many envisioned applications since no prior infrastructure is required to be established [2]. In the following section two of the most important applications of WSNs in health and military domains, which we have used to motivate our technical problem, are discussed in more details.

2.1.1.1 Health applications

Medical and health care is an emerging domain in which WSNs can be involved to simplify different aspects. WSNs enables the real time, remote and continuous monitoring of patients' health status based on their detailed biomedical and physiological data such as heart rate, blood pressure, body temperature, respiratory rate, mental status and oxygen saturation as well as environmental and location information coming from the small sensors installed on the patient's body or in its environment. The integration of these lightweight and tiny sensors together forms a Wearable Wireless Body Area Network (WWBAN) [7] .This domain which is sometimes known as telemonitoring or telemedicine aims at providing an enhanced, automated, accelerated and more efficient medical and health care assistance. Currently, this technology allows doctors, nurses and other medical staff to remotely access and analyze patient's physiological, environmental and location information. This information is provided by the installed sensors and may be transmitted periodically, on demand or as immediate notifications of patient's status life-threatening changes. This enables the medical staff to take appropriate decisions more effectively. In some cases WSN provided information may also leads to appropriate actions being taken automatically to ameliorate the situation such as reducing the temperature of the room or augmenting the level of the oxygen.

In a hospital or clinic environment, sensors can also be attached to the doctors to track their actual location and status and inform other personnel to ensure a more efficient collaboration among them. Moreover, patients and particularly elderly population can be supervised and assisted continuously in a long-term diagnostic procedure directly from their home. This may be required during the surgical recovery or physical and mental rehabilitation period [8].

Wireless medical Body Sensor Networks (BSNs) employed in pervasive health care field can extremely facilitate medical procedures during the mass causality incidents, such as earthquakes, fire or terroristic attacks, by assisting emergency responders. In these situations a large amount of data, such as victim's vital signs, its location and the location of rescue and medical staff, needs to be collected and communicated to avoid overwhelming the emergency medical services. Emergency staff installs related sensors on the victims and receives sensors' information on its mobile devices, PDAs, smart phones or sensor equipments. Specialized doctors in the hospital may also consult this data to give instructions to the staff in place. Many emergency applications, such as triage and treatment of the injured people in a coordinated manner, are already realized based on the WSN technology to overcome critical bottlenecks of emergency activities [9].

Several projects and research prototypes have been developed to take the benefits of the aforementioned issues in the medical care domains. Mobi-Health European project [10] provides a continuous monitoring of patients' health status while they are outside the hospital environment. CodeBlue [11] provides a wireless infrastructure and architecture which can be deployed in emergency medical care. This project combines low-power wireless biomedical sensors, PDAs and PCs.

Obviously, WSN is a promising and revolutionary subject in medical domain although this area necessitates some challenging requirements. Reliable, secure and timely communication of sensitive medical data as well as mobility of sensor and monitoring devices are some of the issues that needs to be addressed.

2.1.1.2 Military applications

WSN development was originally motivated by defense applications since it raises broad range of opportunities in the military systems including command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT) domains. The essential aspects of WSNs which make them attractive for the field of military are their self-organizing, ad-hoc and fault tolerant nature [2].

Installing environmental and physiological sensors on troop members and ammunition in a battlefield enables commanders, equipped by mobile assistant devices, to continuously supervise the actual status of their troop. Another application of WSNs in military domain aims at monitoring the adversary forces by rapidly deploying WSNs in regions of interest and gathering the critical information. This information ends up with an enriched awareness of the situation and higher quality of tactical decisions. After a battle operation taken place, the affected area can be covered by sensors to provide information regarding the level of damages and possible existing chemical agents and nuclear radiation in the battlespace [2].

Major issues of deployment of WSN in tactical military applications are the strong security and reliability since in these scenarios any attack on WSN information may result extremely dramatic results [12]. Many new research venues are opened up in order to ensure the confidentiality, integrity and correctness of WSN relayed data. One of the most recent research projects in this domain is Sensor Information Technology (SensIT) supported by DARPA [13]. This project defines new networking protocols and techniques to employ WSN in battleforce since these environments are highly dynamic. Notably, it covers issues related to sensor data processing and communication in a reliable, secure, efficient and timely manner.

2.1.2 Architecture, protocols and standards

2.1.2.1 Communication Architecture

The communication architecture of WSN typically consists of sensors, one or more sinks and gateway. The objective of these entities is to provide appropriate information regarding a phenomenon for an observer who is also known as WSN application end-user. Sink and gateway functional entities may be integrated in the same physical device. A typical architecture of a WSN is illustrated in Figure 2.1.

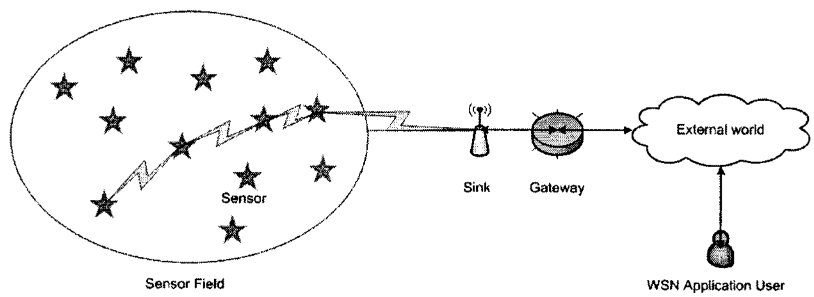

Figure 2.1: WSN communication architecture

Sensor nodes are randomly deployed in an area of interest, known as sensor field. Their task is to constantly sense, collect, measure and disseminate data related to a physical phenomenon.

Sinks are the nodes with higher capability responsible for collecting sensor nodes measurements and eventually issue basic computations, filtering or aggregation on the received information. Each sensor in WSN broadcasts its sensed data until it reaches the sink node following a multi-hop communication paradigm. A problematic issue with stationary sinks is the high energy consumptions of sensor nodes in the vicinity of the sink. These nodes always act as relay for other sensor nodes' information delivery to the sink. Consequently, these sensor nodes are more concerned with the energy exhaustion and their withdrawal may cause a critical problem in the operation of the whole network. Recently, WSN with Mobile Sink nodes (MSWSN) [14] attracts a lot of interest since it overcomes the limitations of static sensor networks in terms of processing and communication resources. Sink nodes in such systems, which may be integrated in mobile devices carried by users, dynamically query the WSN and collect the information of sensors by moving around the sensor field and deploying single-hop forwarding paradigm. This strategy is beneficial for application scenarios which are more tolerant to latency constraints with the objective of having a long WSN lifetime and energy efficiency.

Gateway is an entity which provides connectivity between WSN and the external world meaning where the application end-user resides. Acting as a dual network interface, gateway receives high level requests from the application end-user, analyzes it to identify required information which should be provided by WSN, maps it to commands in relevant format and forwards the query to WSN. Inversely, it interprets the data received from the sink or directly from the sensors in a sink-less WSN [15], processes, aggregates and translates it to high level information understandable for application end-user and sends it to the interested user. Based on the application and deployment environment, gateway may be fixed or mobile, in addition, it may be a centralized module to which all the information is forwarded or employed in a distributed manner among several gateway devices.

The observer is WSN application end-user who requires information about an observed phenomenon and will be provided by this information in a proactive or reactive manner. In other words, an observer may receive information at regular intervals or on demand by sending requests to WSN. As an attempt to automization procedures, the observer may be replaced or collocated with an actuator network.

Actuator receives WSN data, takes decisions and performs relevant actions based on the received information.

2.1.2.2 WSN protocols and standards

Many standard and proprietary specifications have been already defined in hardware and software domain of WSN. Some of the mostly used WSN hardware platforms in both research and industrial areas are Crossbrow Motes [16], MIT crickets [17], Scatterweb [18]. IEEE 802.15.4 [19] defines radio standards for low rate and low complexity data transmission considering devices which require long battery life, such as sensors. ZigBee [20], WirelessHART [21] and ISA100 [22] are networking specifications which are standardized based on IEEE 802.15.4 protocol. The protocols designed for WSN communication should consider specific characteristics, such as self-organization, fault tolerance and low energy consumption.

One of the very commonly used operating systems which is specifically designed for WSN applications is TinyOS [23]. This operating system is mostly accompanied by applications, such as TinyDB [24] and TinyREST [25] programmed in a special language known as nesC [26] to provide sensor data processing and access.

2.1.3 Related challenges

WSN domain raises many exciting challenges and technical issues regarding the communication protocols, architectures and software. Most of these protocols target the optimization of power consumption, security, robustness, fault tolerance, network routing, self-organization, network discovery and dynamicity problems as well as sensor data processing techniques.

In the recent research works, a wide attention is devoted to the design of new networking, communication and data processing technologies to overcome the resource constraints mainly in terms of energy efficiency, memory and computing power. This issue is considerably challenging since it is in contradiction with the requirements on sensor devices to be small and light weight. Distributed data processing may be beneficial in this field since WSN, as an ad hoc network, requires collaboration of sensor nodes to gather, process, filter and aggregate information. Moreover, many research subjects focus on providing interactive application interfaces which can manage WSN and basically sensors such as developing flexible languages and data bases for management and organization of sensor nodes.

As discussed, mobility of WSN devices, such as sensors and sinks, is also a hot research topic since there are tremendous progress in development of multi task hand held mobile devices. Constraint in WSN resources and mobility of the nodes results to very dynamic network topology. Special mechanisms are needed so that each sensor node dynamically becomes aware of the current network topology. Many studies have been done to design routing protocols which are efficient in terms of energy consumption [27–29]. Following the same objective, some works address issues related to the design of low power processors for computation and minimum transmission rates and bandwidth for WSN communication. Another challenge is the security concerns by providing specific techniques in different levels, such as application or network layers [30, 31].

2.2 Mobile Ad-hoc NETworks

2.2.1 Introduction and applications

Infrastructure-based communication is based on the traditional cellular network concept where mobile devices communicate via pre-established access points connected to the fixed network infrastructure. In contrast to these networks, Mobile Ad-hoc NETworks (MANETs) are the ones which deployment does not require any existing fixed infrastructure or predefined organization. These networks can be dynamically established anytime with no prior agreement and in any environment. MANETs composed of autonomous mobile wireless devices such as cell phones, PDAs, laptops or sensors which can join, leave and participate in the network in a peer to peer manner. Since there is no centralized control or organizing entity, such as a server, this is the responsibility of each individual node in the network to discover other nodes with which it wants to communicate directly and exchange information. Each individual node should also consider issues, such as security or QoS, and provide a certain level of reliability in a distributed manner [32].

Communication of the mobile nodes in MANET is based on the multi-hop wireless paradigm in which intermediate nodes act as relays to transmit the data on behalf of other nodes between source and destination [33]. The original development and deployment of ad hoc networks goes back to Defense Advanced Research Projects Agency (DARPA) Packet Radio Network (PRNet) project in 1972 with the objective to provide mobile communication means in hostile areas where no infrastructure is in place. Followed by considerable improvements in wireless communication, mobile devices and ad-hoc networks, in 1997 IETF defined MANET working group. The objective of this group is to design appropriate standards and protocols and expand the MANET technology to non-military domains [1]. Currently, several international conferences, such as MobiHoc [34], held by IEEE and ACM are devoted to MANET technology.

To summarize the aforementioned features, major characteristics of MANETs are mobility, dynamic network topology, self-organization and self-recovery, distributed operation, multi-hop communication, light-weight devices, robustness against node malfunctions and spontaneous formation and deployment.

Basically, MANET is a promising technology for many application scenarios in which communication infrastructure is unavailable. In such scenarios, infrastructure may be destroyed, saturated, expensive, inconvenient or impossible to be set in place. MANET offers an on-the-fly and rapid configurable wireless network which can be deployed in several application domains, such as military and battlefield operation, civilian and business group communications, Personal Area Networks (PAN) like multimedia conferences and electronic classes, emergency, disaster recovery and inter-vehicle communication. Enhancements of mobile technology in both wireless communication and handheld devices make deployment of MANET applications more realistic. As in case of WSN applications, we focus on two specific domains of MANET deployment: military tactical and medical emergency operations.

2.2.1.1 Military applications

Stationary or moving land, air and marine forces can be equipped by mobile devices which form MANETs and intercommunicate in wide geographical environments with no fixed infrastructure [35]. Strategic and tactical military related information can be distributed, shared, analyzed and utilized among the troop members in different levels of hierarchy so that more efficient instructions and decisions can be made based on the better global situational awareness. As an example of these applications, we can refer to the distributed tactical map idea presented in [36] which relies on a dynamic and cooperative generation of a military tactical information plan.

2.2.1.2 Emergency applications

Emergency Applications also take notable benefits of MANET technology since emergency staff, nurses and doctors can interact with each other in special circumstances while communication infrastructure is unavailable. This mostly happens during the natural or human made disasters, such as earthquake or terroristic attacks, as well as in some rural and hardly accessible areas where provision of a fixed infrastructure is so costly or even impossible. In [37] authors propose an architecture for emergency response operation in MANET environment. Emergency responders can carry a wireless mobile device, such as a PDA, on which a decentralized multimedia application is developed to allow rescuers to communicate.

2.2.2 Architecture, protocols and standards

2.2.2.1 General architecture

As discussed in the previous section, MANETs composed of heterogeneous mobile devices randomly interconnected via wireless interfaces. As the wireless coverage range of different devices is limited, some nodes in MANET may need to act as relays to forward the messages coming from other nodes to the destination, hence, MANETs follow a multi-hop communication scheme. Figure 2.2 illustrates the architecture of a MANET deployed in a military operation.

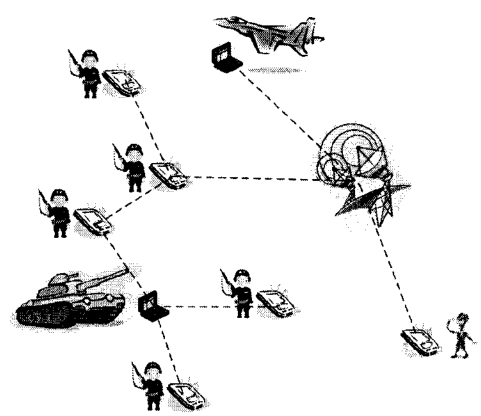

Figure 2.2: Example of MANET architecture in a military operation

2.2.2.2 Protocols and standards

Each participant in MANET can provide specific services to other participants. Discovery and advertisement of existing physical or virtual services are done using application layer protocols while lower layer protocols address establishment of MANET itself [38]. Resources can be distributed and stored in different nodes in

MANET and many protocols have been designed to address storage, search and retrieval of resources in an efficient manner.

Development of dynamic routing schemes is essential for MANET environment since topology of these networks may change frequently and existing routing protocols designed for fixed networks do not fulfill the requirements of these networks. Some of these requirements are decentralization, power conservation and security concerns. Many routing protocols destined to MANET but until now none of them offers all the mentioned properties. These protocols can be divided to proactive and reactive ones. In proactive or table-driven routing protocols, routing information is maintained and updated consistently and based on any changes in the network topology. These protocols result to considerable power consumption and overhead on the network devices, especially in highly dynamic mobile environments. Optimized Link State Routing protocol (OLSR) [39] and Topology dissemination Based on Reserve-Path Forwarding (TBRPF) [40] are two examples of proactive protocols. In case of reactive or on-demand routing protocols a route is discovered only when it is required for a communication which leads to more communication delays. Adhoc On-demand Distance Vector (AODV) routing [41] and Dynamic Source Routing (DSR) [42] are two reactive protocols destined to MANET environment. Comparing the reactive and proactive schemes, one needs to consider the trade-off between latency and overhead in order to choose the most suitable protocol which better fulfills specific application requirements.

In some MANET applications one-to-many information exchange is required. A basic example of this group of techniques is flooding wherein each node forward any received packet to its neighbors. This method is suitable for dynamic mobile environments with less constraint in resources, such as bandwidth or energy.

MANET can be established using any wireless technology, such as infrared and radio frequency (RF). Two sets of wireless technologies that are mainly employed for MANET formation are IEEE 802.11 [43] and Bluetooth [44]. IEEE 802.11 is a simple and high-rate communication standard also known as Wireless Local Area Network (WLAN) or WiFi technology. It provides a wireless coverage range of maximum of 500 meters. Bluetooth standard presented also as Wireless Personal Area Network (WPAN) is a low-cost and short-range communication standard which enables communication between portable devices around 10 meters. Wide area Ad-hoc Network (WAN) is under development for large scale applications such as vast military operations. Medium size ad-hoc networks which are known as Wireless Metropolitan Area Networks (WMANs) use mainly extensions of IEEE 802.16 or WiMax technology to establish the ad-hoc communication [45].

2.2.3 Related challenges

Although MANET technology offers great opportunities for plenty of applications, it introduces many complex issues and severe challenges which interest both academia and industry as open research topics. These issues need to be investigated thoroughly before worldwide deployment of MANET applications.

Designing energy efficient algorithms and protocols for wireless mobile devices recently became an active research topic since having no infrastructure leads to limitation of power in nodes. This issue can be addressed in different layers in the mobile device protocol stack such as physical, data link, network, transport or application layers. For instance, deployment of sleep modes in which a node that is not participating in information exchange does not utilize its energy until it is engaged in the communication [46, 47].

Another challenging issue is security menaces due to decentralized contribution of nodes which transmit each others' information, broadcast nature and shared wireless environment of MANETs. Surprisingly most of the routing and forwarding protocols designed for communication in MANET assume a trusted network of users which is not always true. Authentication and authorization of MANET participants, availability and integrity of relevant data are some of the existing research topics [48]. Mobile devices' constraints in energy and computation resources represent considerable technical problems for provision of secure communication in MANET.

Quality of Service (QoS) requirements in MANET environment is much more challenging to be satisfied. In this condition quality of the wireless links changes very frequently which leads to unpredictable degradation of the quality. Cross-layer design approach is introduced into MANET domain in order to improve the QoS. In this approach information of different layers is accessed whenever required without considering classical constraints on only-adjacent-layer communication [49]. QoS prioritization protocols can be deployed to assign better connections to applications with more quality requirements and still maintaining enough resource for less quality-sensitive applications.

Scalability is another challenging problem in MANET domain, specifically for military applications. This issue is usually addressed by hierarchization and clustering of the ad-hoc network routing algorithms. In many situations, internetworking between MANET and legacy networks like GSM or Internet is desirable to provide a universal communication therefore the designed routing protocols should enable the coexistence of MANET and other external domains especially the IP-based ones.

2.3 Peer to Peer overlay networks

2.3.1 Introduction and applications

Peer to Peer (P2P) communication can be defined as an alternative to the traditional client-server paradigm. Communication between nodes, known as peers, in a P2P network is done directly and does not require any centralized control entity. Nodes in pure P2P networks act as both client and server and offer their resources, including computation power, storage capacity, communication bandwidth, sharable information and other physical or virtual resources, to the whole network. These resources are used to provide different services in a distributed manner, such as file sharing or large-scale collaborative working environments, to the P2P network participants who are symmetrically both service provider and consumer. The term overlay is referred to an application-layer virtual network which is formed on top of a physical network. The main objective of overlay operation is to provide a substrate for easier development of different P2P services specially in MANET environment [3], thus, these logical networks are known as P2P overlay networks. Figure 2.3 illustrates the correspondence between virtual peers participating in an overlay network and real devices in the underlying physical network.

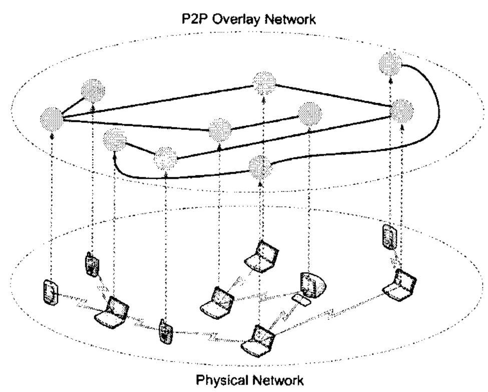

Figure 2.3: A P2P overlay network

Provision of large-scale and enhanced services, such as multimedia conferencing, in MANET is a complex task due to mobility of nodes and decentralized nature of these networks. P2P overlays offer useful peer-level features such as distributed computation, efficient storage and retrieval of data items, hierarchical naming and potentially self-organized, reliable, robust, fault tolerant and scalable routing architectures. To our days, no general-purpose P2P system has been designed.

Improvements in personal computers' resources enables many applications, including instant messaging, file sharing, IP telephony, multimedia conferencing and collaborative computation to take advantages of the P2P paradigm. P2P Instant messaging, as the first application of P2P, provides a more rapid, scalable and robust solution compared to the traditional client-server based message exchange. Many companies proposed P2P file sharing systems, such as Napster [50], Gnutella [51], Bittorrent [52] and Kazaa [53], some of them turned out to stop their applications due to legalization issues. In IP telephony domain, a very popular application, Skype [54], is based on P2P networks paradigm. Data is distributed over all the Skype users and each user shares its resources to the network and uses other users' resources. However, in this system some critical information related to the users account is maintained in special servers. Recently, multi-player online games become a more and more active filed. Deployment of P2P overlay networks instead of the commonly used client-server paradigm addresses problematic issue of scalability in such systems [55]. Moreover, recent enhancements in mobile communication open up new opportunities to develop P2P applications which integrate both mobility and decentralization. Many applications designed for MANET environment can deploy P2P paradigm to ensure scalability and performance [56]. However, this domain is still newborn and poses considerable technical challenges.

2.3.2 Architectures

A general P2P overlay architecture is proposed by [3] and illustrated in Figure 2.4. This architecture consists of five layers: network communications management, overlay nodes management, feature management, services-specific management and application level management.

Network communications, as the lowest layer, defines how nodes in a P2P overlay are connected together. Join, leave, discovery of peers in the overlay and

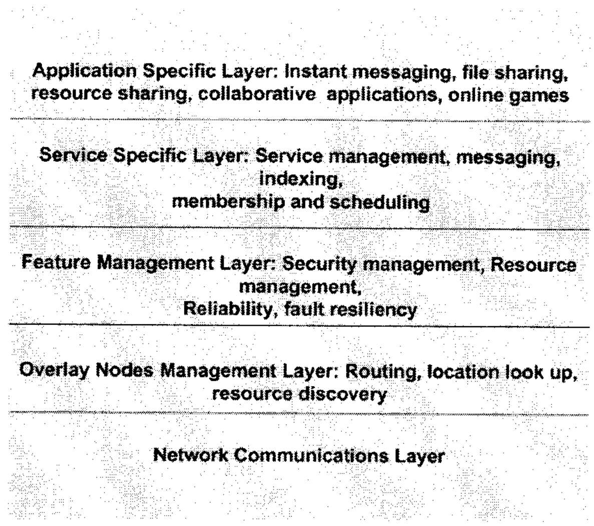

Figure 2.4: P2P overlay architecture

routing algorithms are managed by overlay nodes management layer. Feature management layer focuses on the reliability, fault tolerance, security and robustness of overlay networks. Service and application management layers are concerned with high level aspects of specific P2P services and applications, such as searching and data retrieval, indexing, scheduling, messaging and document sharing. There is an interest and possibility to reuse the lower layers in different P2P systems. For instance, many P2P applications are developed based on Chord [57], CAN [58] or JXTA [59]. Reusability cannot be applied to higher layers since applications and services may be different and specific to each P2P system [60]. Depending on the level of organization, there are two categories of P2P overlay networks: structured and unstructured overlay networks.

2.3.2.1 Structured P2P overlay networks

Structured overlay networks provide support for efficient data storage and retrieval by maintaining a tightly controlled overlay network topology. In these overlays data is stored in specified peers; its location can be deterministically calculated any time and accessed directly. Distributed Hash Tables (DHTs) are deployed in structured P2P overlays to provide the aforementioned functionality. In these tables, each peer is assigned by a random unique identifier. Similarly, each data object is given by an identifier (key) and is placed in the overlay based on a mapping function. This function maps the data item key to a peer identifier which is closed to it. Object key is then used to lookup the desired data items based on the mapping function which returns the actual location of the information.

Messages in structured overlays are transmitted by using the routing tables which store contact information of neighboring peers and are managed by each peer. This information is mainly the node unique identifier and its IP address. A node in an overlay can directly exchange information with other nodes which are in its routing table through single overlay hop communication. In theory, DHT-based networks ensure discovery of an object key in an order of O (log(N)) number of overlay hops where N defines the total number of peers in the overlay. The virtual path taken by a query is based on the peer identifier and may be totally different from the underlying physical network. This may incurs much more physical routing hops and results a higher discovery delays than the theoretical guarantee [3]. These networks are also based on strict rules for peer joining and leaving the overlay which makes them less resilient and may introduce high maintenance overhead. Some examples of implemented DHT-based overlay networks are Kademlia [61], CAN [58], Pastry [62], Chord [57] and Tapestry [63].

2.3.2.2 Unstructured P2P overlay networks

In contrast to structured overlays, unstructured P2P networks impose no constraint on the placement of data objects, neither on nodes entering and leaving the network. Data is randomly stored and retrieved by flooding techniques in which a peer simply broadcast the query [64]. A basic approach to limit the propagation of a query among the peers is to define a Time To Live (TTL) value associated with each query which will be decremented each time a peer receives the query. When TTL reaches zero the query message will be destroyed. Although these networks have low maintenance overhead, they may introduce high network traffic due to flooding especially in case of rare data item discovery. As a result, they do not scale well and are issue of overloading. To overcome this inefficiency, some methods have been proposed such as forwarding, caching and overlay topology optimization [65]. In forwarding technique, each peer transmits the messages only to a subset of its neighbors which are candidate based on some statistic information, such as latency. As in case of caching, each peer stores an index of resources which are maintained by peers with limited logical distance from it. The third approach takes the real network topology into consideration and build the overlay connections based on that. Freenet [66], Gnutella [51] and Overnet [67] are some popular examples of unstructured overlay.

2.3.3 Related challenges

Potential advantages of P2P overlay networks motivated many researchers to focus on rendering operation of these networks more efficient. Design of locality-aware overlays is an active research which reflects the physical network topology in formation of the overlay topology and reduces the length of physical routing path taken by the message queries [68, 69]. Moreover, development of hierarchical routing, and super peers architectures attracted many interest [70].

Security concerns is another challenge in un-trusted P2P systems which is addressed by encryption techniques, distributed trust and reputation methods [64]. Efficient and scalable routing, look up and caching algorithms, specially in case of unstructured P2P overlays, are another problematic issues currently addressed by many researchers [51, 71]. To our knowledge, there is very limited number of implemented P2P application development frameworks. The most popular general purpose P2P middleware, JXTA [59], developed by Sun provides many P2P functionalities, such as content-based searching and multi-casting.

2.4 Session Initiation Protocol

Session Initiation Protocol (SIP) is an infrastructure-independent application-layer protocol standardized by Internet Engineering Task Force (IETF) as the 3GPP signaling protocol. RFC 3261 [5] defines the last version of specifications for this protocol. The main task of SIP is to initiate, control, manipulate and terminate two-party or multi-party multimedia sessions over IP-based networks. Some of the applications which can use SIP as signaling protocol are collaborative multimedia conferencing, instant messaging and online games. This protocol brings a simple, scalable, extensible and efficient solution for a large variety of scenarios in which application-layer signaling is mandatory.

In applications which are developed based on SIP, each end-user is identified by a SIP Uniform Resource Identifier (URI). Translation of names to the current network addresses aims at determining the actual geographical location of an end-user. Availability and presence information of end-users can also be tracked by SIP functions. Moreover, SIP can manage the capability of each user regarding the communication media, such as audio, video or different codex formats. In multi-party communication for instance, involved parties can negotiate the supported features before establishing the session. Once this information is negotiated, communication parameters are defined and the session is then established between the interested parties. During the session, SIP functions can be used to modify session's corresponding parameters, transferring or terminating the session.

2.4.1 SIP components

SIP operation is based on the client-server request-response paradigm. SIP architecture consists of four types of functional entities which may exist during the communication depending on the application requirements. Each of the SIP components can participate in communication as a client, a server or both and may reside in the same or separated physical devices.

SIP User Agent (UA) is an end-user device, such as PC, call agent or IP-phone, which communicates via SIP and is in charge of session establishment, modification and termination. A SIP UA may be a UA Client (UAC) or a UA Server (UAS). SIP UAC is the entity which initiates SIP requests while the SIP UAS is the one which receives the requests, informs the end-user and issues the appropriate responses. Register servers, which are associated to each domain, act as repository for SIP UAs' contact information, such as their location. Corresponding information of an end-user can be retrieved from the register server any time it is required. Proxy servers act as both client and server. They are in charge of receiving requests from the SIP UAs, retrieving the destination party required information from the register servers and transmitting the request to the corresponding destination. This may be done directly if the UA resides in the same domain as the proxy server or translated and passed to another proxy server which belongs to the same domain as the destination UA. Redirect servers are used by the proxy servers to manage the transmission of requests to an external domain. This is done by sending redirect responses to the requester UA indicating the destination UA new URI address.

2.4.2 SIP messages

Having reused the HTTP model, all messages in SIP are text-based. They are embedded in SDP (Session Description Protocol) [72] format which is used to transfer the session related information. There are two types of messages in SIP: Request messages and response messages. The request messages, known also as SIP methods, are sent from the SIP client to the server and the corresponding results are sent back from the server as the numerical code responses. Table 2.1 represents some of the existing SIP request messages and their description.

SIP Method	Description
REGISTER	Registeration of a user to location service of the register server
INVITE	Initiating a session between users
ACK	Confirming the final response of the session initiation
BYE	Terminating the session
SUBSCRIBE	Subscription of a user to an event
NOTIFY	Notifying a user about the subscribed event
INFO	Sending the session information without changing the session state
OPTIONS	Request for SIP user capabilities

Table 2.1: SIP methods

SIP response messages are in numerical format and may include additional information. Since SIP is a session-oriented protocol, the numerical codes define the next operation of the corresponding SIP UA. Response messages may be provisional, to indicates the progress of SIP transactions, or final responses. Table 2.2 shows the existing SIP response codes and their corresponding response class.

Each message in SIP contains a start line, headers and a body. The start line indicates the type of the message and supported protocol version. For a SIP request, message type is the SIP method and the request URI. In case of a SIP response the start line, which is known as status line, indicates the numerical response code and its associated textual definition. SIP headers include the message signaling attributes and are used to convey the message from one entity to another. The message body defines the session-related parameters or any other required information and

Response Code	Response Class
1xx	Provisional, searching, ringing and other informational responses
2xx	Success response
3xx	Redirection and forwarding response
4xx	Resuest failure (client errors)
5xx	Request failure (server errors)
6xx	Global failure (busy, refused, unavailable)

Table 2.2: SIP response codes

is usually in a SIP-independent format such as SDP.

2.5 Summary

This chapter explained WSN, MANET and P2P overlay networks main characteristics, applications, architectures, protocols, standards and related challenges. Moreover, the SIP protocol, which is later on deployed as the implementation technology, is discussed in detail. In the next chapter we explore the requirements and state of the art for the integration of WSN and MANET.

Chapter 3

WSN and MANET Integration: Requirements and State of the art Evaluation

Existing networks, such as 3G, Internet or MANET, can be integrated with WSNs and make use of their contextual information. Interworking of WSNs with these networks enables the development of a large variety of value-added services. Our goal is to integrate WSN and MANET in order to enhance the end-user experience of MANET participants. This integration imposes specific challenges and requirements which are mostly due to the ad-hoc nature, lack of centralized control and frequent topology changes in these networks.

This chapter is divided into four sections. In the first section, we derive the requirements for the integration of WSN and MANET. In the second section, we organize the related works for the integration of WSN with other networks into two categories. We then review some of the major works in each category. Evaluation of these works with regard to our derived requirements is discussed in the third section. A summary of the discussed issues is presented at the end of the chapter.

3.1 Requirements for the Integration of WSN and MANET

In this section, we derive three sets of requirements for the integration of WSN and MANET. The first set includes the general requirements for the overall system. The second set is the specific requirements for gateway entity which interconnects WSN and application end-users. The third set comprises the specific requirements for application entity which resides in MANET and wishes to use WSN information.

3.1.1 General requirements

As discussed in Chapter 2, MANETs are infrastructure-less networks in which participants may leave and join dynamically. Our overall architecture should be built on top of MANET end-user devices. Due to the frequent topology changes and resource constraints of mobile devices, none of the MANET participants can operate as a centralized node. Therefore, our first requirement is that the integrated system should not be based on any centralized functional entity.

Anytime MANET participants join or leave the network, they should be managed dynamically and in a distributed manner. These devices may leave the network voluntarily or unintentionally, due to the link failure or lack of resources. Therefore, the second requirement is that the integrated system should provide self-organization and self-recovery mechanisms.

The system should operate properly even in large scale deployments, such as natural disasters and military operations. Therefore, as the third requirement, system should scale both in terms of number of WSN nodes and MANET participants.

Quick communication establishment is one of the essential motivations for deployment of ad hoc networks. As the fourth requirement, our approach should be rapid to set up.

The fifth requirement stipulates that the system should use resources, such as bandwidth, memory, energy and processing power, efficiently. This is due to the fact that heterogeneous mobile devices deployed in MANETs are limited in resources. This implies design of light-weight communication protocols. Moreover, standard protocols bring interoperability and easier deployment of the integrated system. Therefore, use of light-weight and standard protocols is our sixth requirement.

As the last requirement, solution should be completely based on application-layer protocols and independent of lower layers, such as physical, data link or network protocols which may not be compatible or available in different environments.

3.1.2 Gateway and application specific requirements

As a consequence of the first general requirement, only distributed gateways and applications can be used in our system. Furthermore, based on the scenarios we have envisioned in Chapter 1, both gateway and application entities must be mobile. Therefore, decentralization and mobility are the first two common requirements for the gateway and application entities.

3.1.2.1 Gateway-specific requirements

First of all, gateway should aggregate, process and store ambient data it receives from WSN. As the next requirement, it should translate the information model employed by sink to an interpretable format required by end-user applications and vice a versa. It should also allow both synchronous and asynchronous communication modes. The last requirement is that it should support information management of different types of physical world entities, such as persons and objects as well as their related sensed data, like physiological or environmental information. This provides a general-purpose solution which can be deployed for different applications.

General requirements	1. No centralized entity
	2. Self organization and self recovery
	3. Scalability in terms of WSN nodes and MANET participants
	4. Rapid set up
	5. Efficient use of resources
	6. Support of light-weight and standard protocols
	7. Independency from lower layers
Gateway-specific requirements	1. Distributed
	2. Mobility
	3. Processing and storage
	4. Information model translation
	5. Synchronous and asynchronous information exchange
	6. Support of different type of physical world data
Application-specific requirements	1. Distributed
	2. Mobility
	3. Support of session-based and non session-based applications

Table 3.1: Derived requirements for the integration of WSN and MANET

3.1.2.2 Application-specific requirements

In order to cater various needs of MANET participants, application entity should support different types of applications, including session-based and non session-based. In session-based applications, application entities need to establish a session and store information about it. An example of this sort of applications is multi-party applications, such as multimedia conferencing or multi-party decision systems. In non session-based applications, users neither need to establish a session nor have knowledge of previous messages. Event notification applications and stateless request-response systems are examples of these applications.

Table 3.1 represents a summary of derived general and specific requirements for the integration of WSN and MANET.

3.2 Related Work on the Integration of WSN and MANET

To the best of our knowledge, despite its considerable benefits, there are a very few works on the integration of WSN and MANET. This is due to many challenging issues, such as resource limitations, in both WSN and MANET domains. The most common approach to integrate WSN and other networks is to deploy an entity known as application-level gateway. The gateway's main task is to interconnect sensors in WSN to end-users on the other networks via defined interfaces. This is thereby the responsibility of gateway to implement relevant protocols and translate received information from WSN to an appropriate format for end-user applications. We divide this group of approaches into two categories: One deploys centralized gateways and the other one is based on distributed P2P gateway overlays. In the first category, gateway functionality is centralized in one sole node to which all the information is relayed. To ensure fault tolerance, there may be more than one gateway in these systems. However, these gateways do not inter-communicate and each of them provides a complete set of functionalities. On the contrary, in the second approach, gateway capabilities are distributed over several overlay nodes. These nodes are known as gateway peers which collaborate in a decentralized manner to provide required gateway functionalities.

In this section we introduce six related works on the integration of WSN and other networks, including MANETs. We organize these works into two subsections, one for each category.

3.2.1 Centralized gateway approaches

In [73] authors propose a single-gateway architecture for the integration of WSN sensing functionalities into the IP Multimedia Subsystem (IMS). This architecture provides standardized mechanisms to access required WSN data for different applications via IMS framework. IMS is a global and standardized architecture which enables connectivity and service control functionalities for the 3G networks based on the Internet protocols. Through IMS, enhanced services, such as multimedia communication, can be provided to the 3G end-users in a standard manner [74]. An example of possible value-added services which can be developed based on the proposed WSN/IMS integrated architecture is in emergency applications. If patient's physical condition, as measured by WSN, so indicates, these applications can automatically establish a 911 call between the patient and Public Safety Answering Point (PSAP). This greatly enhances the patient's end-user experience.

The proposed architecture, as illustrated in Figure 3.1, is based on a fixed and centralized application-layer WSN/IMS gateway. The gateway receives queries from the applications through IMS and translates them into WSN-level messages and vice versa. This architecture uses the presence service framework as the entry point to the IMS. Presence service provides status, such as availability, and contact information of users to an interested entity known as watcher.

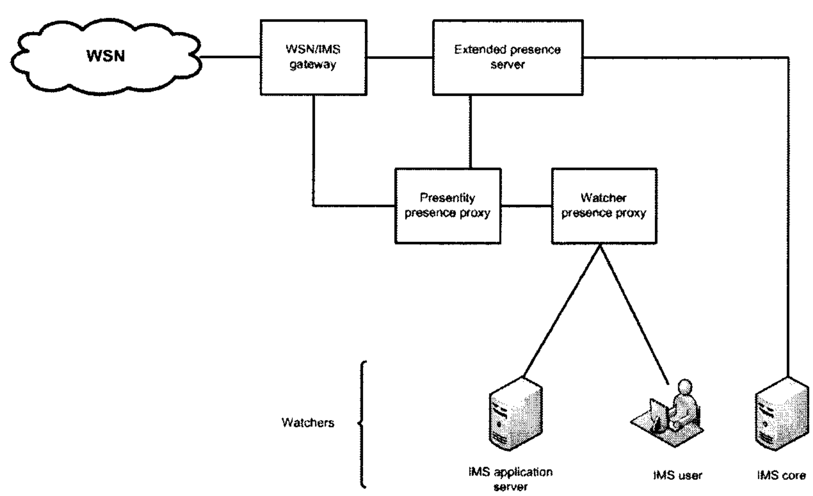

Figure 3.1: WSN/IMS integration architecture

This architecture is based on application-layer, lightweight and standard protocols, such as SIP. Deployment of light-weight protocols results efficient use of resources of 3G end-user devices. Gateway specific requirements, except decentralization and mobility, are also satisfied. Applications residing in the IMS are centralized fix or mobile applications thereby our first application specific requirement which is the decentralization is not satisfied. Consequently, this architecture does not satisfy our first two general requirements. Moreover, due to centralization, the overall system does not scale and the designed gateway may easily become a bottleneck in large-scale deployments.

The architecture defined in [75] is another example of centralized gateway approaches. As illustrated in Figure 3.2, this architecture consists of several small-scale P2P mobile sensor networks and their representative gateways. Mobile sensors are carried by operators and gateway functionality is developed on mobile phones or PDA devices. Gateway is employed as the only access point and control entity between its proprietary sensor network and the 3G network. It is assumed that users have no knowledge of the sensor networks capabilities and related services. Each time user wants to access a WSN, corresponding gateway present these capabilities to the interested user. This approach also proposes collaboration of several WSN domains which results an enriched level of services.

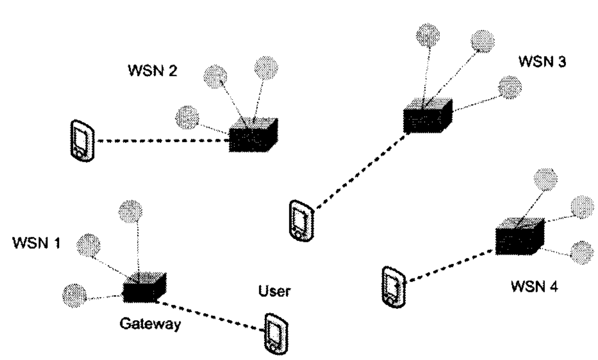

Figure 3.2: P2P mobile sensor network integration with 3G network

Gateway architecture consists of several layers. Wireless interface layer, as the lowest layer, provides support for lower layer communication technologies, such as Bluetooth and 3G communication interfaces. Its task is to discover the existing sensors, set up communication links and execute relevant routing protocols to relay information from sensors to users and vice-versa. Sensor network abstraction layer provides a high-level view of the existing sensor network capabilities to application end-user. Sensor network API layer provides functions and messages to interact with sensor nodes. The proposed gateway architecture destined to arbitrary types of physical world and sensor types and can be deployed for different kinds of applications with very little modifications.

The proposed gateway supports almost all of our gateway-specific requirements, including mobility, processing, synchronous and asynchronous communication and support of a variety of sensor and user entity types. It also deploys lightweight application-layer protocols and takes efficient use of resources in to consideration. However, our first two general requirements are still not satisfied since gateway and application in this system are centralized. This approach does not scale due to centralized gateway and creates a single point of failure. Furthermore, it does not address the session-based applications and collaborative communication of users in MANET which is one of our major motivations.

As the third related work, gateway proposed in [76] provides long-distance access to WSN information for environmental monitoring applications. A web service based application is designed for long-distance access of end-users via TCP/IP protocols. The main focus of this work is design and implementation of hardware and software modules of gateway and web service application. This architecture, as illustrated in Figure 3.3, consists of two stationary centralized modules: one is the gateway and the other one is the manager system.

The proposed gateway architecture meets our gateway specific requirements

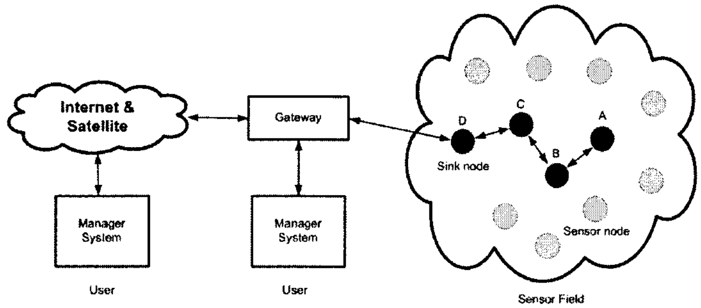

Figure 3.3: Environmental WSN integration with other networks

except mobility and decentralization. However, the whole system does not meet many of our requirements. Having gateway and manager system as two centralized entities, this system does not satisfy our two first general requirements. It is not scalable in terms of WSN nodes and centralized gateway entity causes a single point of failure. Gateway architecture also relies on lower layer protocols, such as TCP/IP. Furthermore, efficient use of resources and deployment of lightweight protocols are not addressed. Finally, the application entity is centralized and does not support the multi-party session-based applications.

3.2.2 Gateway overlay approaches

In [77] authors propose an architecture for the integration of WSN (with mobile sinks and gateway) and IMS. Similar to [73], the presence service is used to interconnect proposed gateway to the IMS world. Gateway entity forms a P2P overlay which is built on top the mobile phones. These devices also act as mobile sinks, presence watcher and publisher and application end-user devices.

Gateway architecture, as illustrated in Figure 3.4, is based on five types of roles that each peer may take when it joins the overlay. Some nodes act as entry point for IMS presence service and are known as Presence Service Entry point (PSE). The second role is to interconnect gateway and mobile sink entity as Sink Entry Point

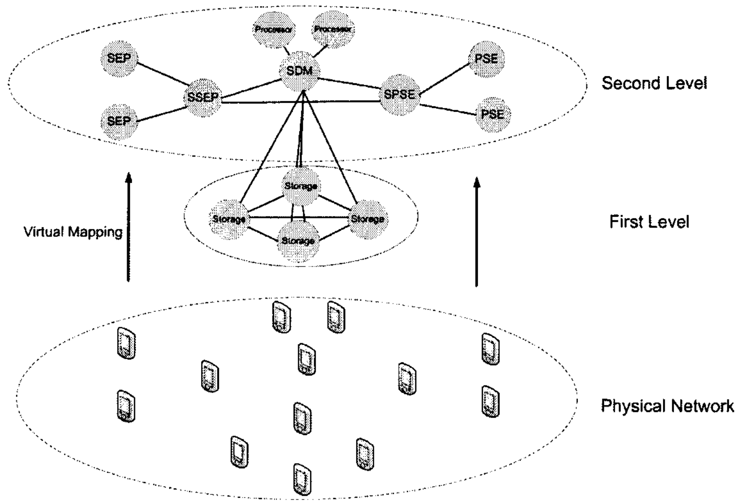

Figure 3.4: WSN/IMS gateway overlay architecture

(SEP). The third and fourth roles are storage and processing of information. The last role is assigned to special peers which manage all the aforementioned peers. These peers are known as super peers and are divided to Super Sink Entry Point (SSEP), Super Presence Service Entry point (SPSE) and Super Data Management (SDM). Each super peer is in charge of managing its respective set of peers. For instance, after receiving the sink information from SEP, SSEP decides to which entity (SDM or SPSE) this information should be relayed. In the same manner the SPSE interconnects the gateway and the IMS presence service via its PSE peers and relay the received information to SDM or SSEP peers. Moreover, SPSE publishes the required information to the IMS presence service via the PSE peers. The proposed gateway overlay is designed in two levels in order to separate the storage module from the other gateway functionalities. SDM is responsible to connect the storage level with the rest of the gateway overlay. Depending on the requirements, SDM forwards the information to one of the processor peers to process it or to one of its storage peers to store it.

This gateway overlay meets all of our gateway-specific and many of our general requirements. However, application end-users access WSN-related services via IMS which is based on a centralized architecture. Although it is mentioned that the proposed gateway architecture can interact with P2P IMS [78], related mechanisms for this interaction are not addressed. Consequently, the proposed architecture is partially centralized and cannot completely satisfy our requirements on self-organization and decentralization.

DUMBONET [37] is a hybrid combination of MANETs, satellite IP network, and the Internet. This system aims at providing a collaborative emergency response application for mass causality events. Application motivation in DUMBONET is very similar to one of our motivating scenarios. Medical and rescue staffs are equipped with ad-hoc mobile devices in order to interact among themselves, along with remote communication with specialists in hospital. This system can be combined with physiological and environmental sensor networks that provide ambient information required by emergency staff. Proposed architecture is illustrated in Figure 3.5 and mainly focuses on multimedia communication of nodes in local and remote sites via MANETs, Internet and over the satellite links.

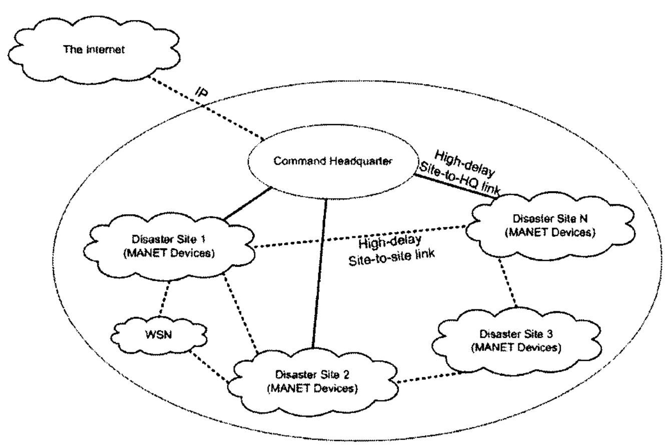

Figure 3.5: DUMBONET architecture

In this architecture, command headquarter is a special peer which can communicate with the other sites in a privileged fashion. This node has the possibility of broadcasting messages to all the other normal peers in different sites. All involved mobile devices, including headquarter and ordinary nodes, belong to an IPV4 Virtual Private Network (VPN). Application nodes cooperate to perform victims' face image recognition methods by comparing the captured images with an existing data base. Headquarter, as a more powerful node, run a comprehensive version of P2P multimedia application with more access and global view of the situation. It is interesting to mention that the described emergency response application is experimentally deployed during the 2004 tsunami in one of the provinces in Thailand. This experiment arises several challenges regarding the QoS in MANET environment as result of frequent topology changes and high packet loss ratio.

Application nodes in this architecture operate in a P2P fashion and can be mobile. Therefore, our first two application specific requirements are satisfied. The system also supports both session-less and session-oriented applications. Although the system can support a variety of heterogeneous mobile devices and communication technologies, such as satellite or WiFi, solution still relies on lower layer protocols. Authors state that DUMBONET can be integrated with WSN based on a gateway entity, implemented on the very same MANET devices. However, they do not provide any mechanisms or protocols for this integration. The first two general requirements thereby are partially satisfied. Moreover, deployment of the proposed system is time consuming, as mentioned by the authors, and the resource consumption efficiency is not addressed. System is scalable in terms of number of mobile devices but scalability in terms of WSN nodes is not clarified.

As the last related work in this category, we review the architecture proposed in [79] for farm animals monitoring. This system indicates general health status of the cattle and detects contaminating diseases. The proposed architecture, as illustrated in Figure 3.6, includes several MANETs. Each MANET consists of mobile sensors mounted on animals, mobile gateways and application end-user mobile devices. These entities form a structured DHT-based overlay. Stockmen, farmers or veterinaries in place can directly request information about the animals by use of their mobile devices. Moreover, remote users can access the required information through the Internet while they are at home or office. This is done by deploying specific WSN/Internet gateways which connects to the DSL modems. Gateway functionality can be distributed over the MANET. In addition to this, to extend the information access range, different MANETs can intercommunicate. This is feasible by deploying vehicles or people which act as data couriers.

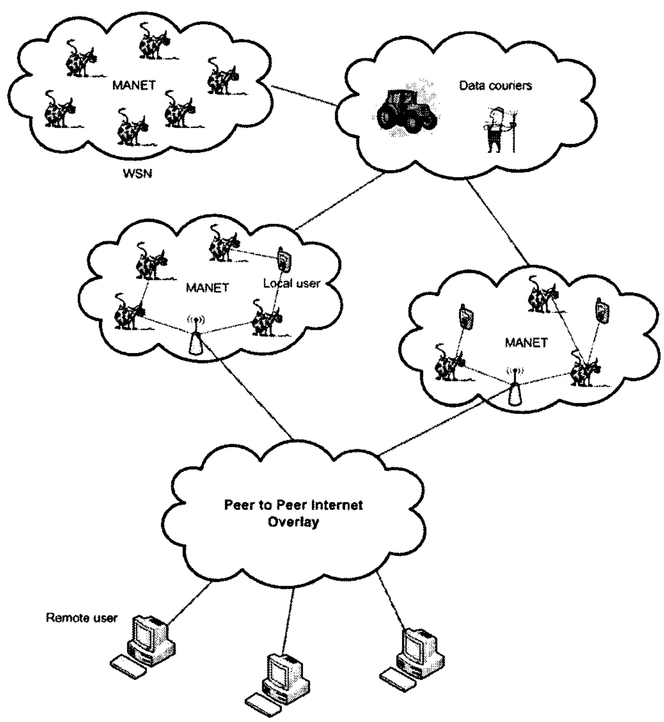

Figure 3.6: P2P cattle monitoring architecture

This architecture meets many of our requirements. It is infrastructure-less, self-organized and has no centralized entity. Gateway and application entities are mobile and distributed over the MANETS. This approach is scalable and can be deployed in large-scale scenarios when millions of devices, including sensors and end-user devices, may be present. It is also efficient in mobile devices' resource consumption. However, it does not address some fundamental issues such as application-layer deployed protocols and relies on the routing layer protocols. It also does not consider the communication of application end-users which is one of our problem motivations. It is based on a specific gateway information model and cannot be generalized to different WSN and physical world types.

3.2.3 Evaluation summary

Table 3.2 summarizes our evaluation of the discussed approaches with respect to the requirements we have previously shown in Table 3.1. We can observe that none of the discussed integration approaches can meet all of our requirements.

From this review we can conclude that the centralized gateway approaches do not satisfy most of our requirements. When ambient information collected by WSNs is intended for end-user applications residing in MANETs, centralized gateways are usually not practicably feasible. This is mainly due to infrastructure-less nature of MANETs and constraints on the end-user devices that are most often used in MANETs. These devices may be the PDAs and smart phones with limited processing power and memory capacity that cannot host centralized gateways.

On the other hand, from the evaluation, it is evident that the gateway overlay architecture presented in [77] could be used as a starting point for the design of the gateway entity of our system. The same observation applies to overlay-based architecture proposed in [37] to provide enhanced end-user multimedia services. Although this approach does not meet some of our general requirements, it can be a viable approach toward the design of the application entity of our overall system.

	Requirements \ Related works	Gateway architecture for integration of WSN and IMS	P2P mobile sensor networks	WSN Gateway based on environmental monitoring	An overlay gateway for the integration of IMS and mobile sink based WSN	DUMBONET	MANET approach for cattle monitoring
General	1. No centralized entity	Not satisfied	Not satisfied	Not satisfied	Partially satisfied	Partially satisfied	Satisfied
	2. Self organization	Not satisfied	Not satisfied	Not satisfied	Partially satisfied	Partially satisfied	Satisfied
	3. Scalability	Not satisfied	Not satisfied	Not satisfied	Satisfied	Partially satisfied	Satisfied
	4. Rapid set up	Not discussed	Satisfied	Not discussed	Satisfied	Not satisfied	Satisfied
	5. Efficient use of resources	Satisfied	Satisfied	Not discussed	Satisfied	Not discussed	Satisfied
	6. Light-weight and standard protocols	Satisfied	Satisfied	Not discussed	Satisfied	Not discussed	Not satisfied
	7. Independency from lower layers	Satisfied	Satisfied	Not satisfied	Satisfied	Not satisfied	Not satisfied
Gateway-specific	1. Distributed	Not satisfied	Not satisfied	Not satisfied	Satisfied	Satisfied	Satisfied
	2. Mobility	Not satisfied	Satisfied	Not satisfied	Satisfied	Satisfied	Satisfied
	3. Processing and storage	Satisfied	Satisfied	Satisfied	Satisfied	Not discussed	Satisfied
	4. Information mode translation	Satisfied	Satisfied	Satisfied	Satisfied	Not discussed	Satisfied
	5. Synchronous and asynchronous information exchange	Satisfied	Satisfied	Satisfied	Satisfied	Not discussed	Satisfied
	6. Support of different type of physical world and sensed data	Satisfied	Satisfied	Satisfied	Satisfied	Not discussed	Not satisfied
Application-specific	1. Distributed	Not satisfied	Not satisfied	Not satisfied	Not discussed in detail	Satisfied	Satisfied
	2. Mobility	Satisfied	Satisfied	Satisfied	Satisfied	Satisfied	Satisfied
	3. Support of session-based and session-less applications	Satisfied	Partially satisfied	Partially satisfied	Satisfied	Satisfied	Partially satisfied

Table 3.2: Summary of the evaluation of the related work for integration of WSN and other networks

3.3 Summary

In this chapter, we derived the general, gateway and application specific requirements for integration of WSN and MANET. We then introduced the related work on gateway-based integration of WSN with other networks. The related work was organized into two categories, one for centralized gateways and the other one for distributed gateway overlays. We evaluated these works with respect to our requirements and we found that gateway overlay based approaches satisfy many of our requirements in contrast to the centralized gateways.

With these ideas in mind, we propose an overlay-based architecture for integration of WSN and MANET. This architecture is presented in the next chapter.

Chapter 4

A Two-level Overlay Architecture for the Integration of WSN and MANET

In the previous chapter, we derived the requirements for integration of WSN and MANET. With respect to these requirements, overlays are the most appropriate approach to be deployed for gateway and application components of our entire system. There are two main reasons for this claim: the first one is decentralized nature of overlays which is essential for deployment of MANETs and the second one is that this approach enables scalability in terms of both number of WSN nodes and MANET participants.

This chapter is organized into five sections. In the first section, we introduce our overall architecture for integration of WSN and MANET. This architecture is based on gateway and application overlays. The key issue to be targeted is how these overlays should be interconnected in order to exchange information. In the second section, we thereby derive the requirements for interconnection of these overlays, review and evaluate the existing approaches for overlays' interconnection. In the third section, with respect to the requirements, we refine our architecture by introducing gateway and application overlays interconnection mechanisms into the overall WSN/MANET integrated system. In the fourth section, an operational scenario is presented to illustrate how the refined architecture works. We end this chapter with an evaluation of our refined architecture with respect to the requirements.

4.1 An Overall Architecture for the Integration of WSN and MANET: First Look

There are two main functional entities in our overall WSN (with mobile sinks) and MANET integrated architecture. The first entity is the decentralized gateway which resides in MANET and connects end-users to WSN. The second entity consists of one or several decentralized end-user applications which also reside in MANET. We assume that all mobile devices involved in the overall architecture are equipped with MANET specific interfaces, such as WiFi or WiMax. These protocols and standards were discussed in Chapter 2 in detail.

As it is depicted in Figure 4.1, the overall architecture comprises two P2P overlays built on top of the heterogeneous MANET end-user devices. The first is gateway and the second is application which are mapped from the real physical network to the virtual overlays. We define two components to represent each peer in the overlays: GateWay Component (GWC) in the gateway overlay and Application Component (AppC) in the application overlay.

GWCs are the peers in the gateway overlay which collectively act as a gateway entity. Sensors' raw data is gathered by mobile sink devices and is transmitted to the gateway overlay through GWC peers which are acting as gateway entry points. As assumed in [77], there is a business agreement between the providers of WSN with mobile sink modules and operators which manage the gateway. This

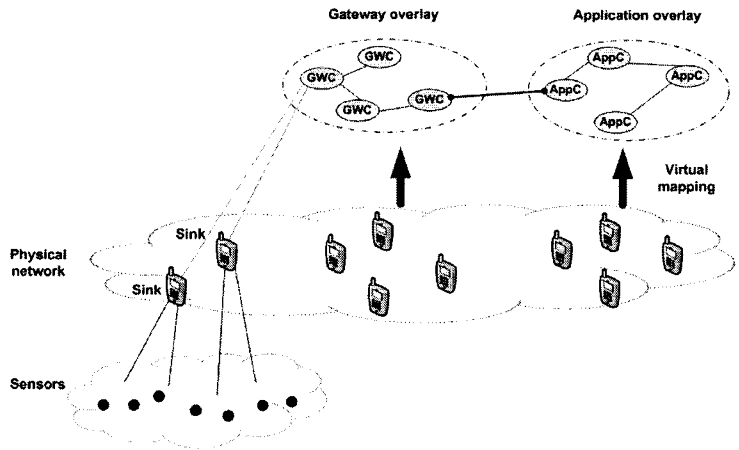

Figure 4.1: The overall architecture for integration of WSN and MANET

means that WSN, mobile sink and gateway infrastructures utilize common protocols and know each other. Once the data is received in the gateway overlay, GWC peers collaborate to process, store and translate it to an appropriate format. As introduced in the previous chapter, authors in [77] propose an overlay-based gateway architecture for the integration of WSN (with mobile sinks) and IMS. This gateway satisfies all our gateway specific requirements. Therefore, the GWC peers can be mapped to a subset of the defined roles in the proposed gateway overlay. These roles consist of the sink entry points, super sink entry point, processor, storage and super data management. Details of these roles and their interaction have been discussed in the previous chapter. Collaboration of these peers via the defined protocols and mechanisms provide complete gateway functionalities. However, in this work application is assumed to be centralized and is accessed via the IMS modules. In the contrary, in our overall architecture we deploy distributed applications residing in MANET end-user devices.

In the application overlay, AppC nodes are application components that collaborate to fulfill application-related tasks. Logic of these nodes is defined by the provider of each existing application overlay. Although Figure 4.1 shows only one application overlay, there may be several applications, such as instant messaging or multimedia conferencing, using the same gateway entity. Required information for each application should be transmitted from the WSN and through the mobile sinks to the gateway overlay. Once the information is processed and translated by the gateway, it should be sent to the appropriate application overlay. In this stage we should refine our overall architecture and propose mechanisms which provide interconnection between gateway and application overlays. The overlay interconnection is discussed in the next two sections.

4.2 Interconnection of Gateway and Application Overlays: Requirements and State of the art Evaluation

This section is divided into three subsections. In the first subsection, we derive the requirements for interconnection of gateway and application overlays. We then review the related work for interconnection of overlays and categorize these works into two groups. The last subsection is an evaluation of the discussed works with respect to our requirements.

4.2.1 Requirements for interconnection of gateway and application overlays

As the first and most important requirement, refined architecture and related protocols should be independent of architecture and middleware of the underlying gateway and application overlays. This is due to the fact that a wide range of P2P overlay architectures, such as structured and unstructured, and a variety of P2P middleware systems like JXTA and Chord exist. Therefore, the proposed mechanisms should be deployable for interconnection of any gateway and application overlays.

The second requirement is that the system should provide both synchronous and asynchronous intercommunication of overlays in order to be coherent with our gateway-specific requirements.

As the third requirement, interconnection architecture and mechanisms should not rely on lower layer protocols used in the overlays that are being interconnected since these protocols may change in different environments.

The fourth requirement is scalability. The system should scale in terms of number of overlays to be interconnected as well as number of nodes in each of the underlying overlays. This is necessary since there will be several end-user applications running as separated overlays while using the same gateway overlay. In addition, in some applications, such as military operations or large-scale accidents, there may be an important number of participants in MANET.

The next two requirements stipulate that overlay interconnection mechanisms should be self-organized and have no centralized entity. Since the overall architecture discussed in the previous section is built on top of the MANET, none of its entities, including the interconnection-related ones, can be centralized.

The seventh requirement is that our interconnection procedures should be efficient in use of resources of mobile devices involved in the interconnection procedure. The proposed solution should be based on lightweight protocols to fulfill the optimal usage of resource-constrained mobile devices. Moreover, the protocols used in our architecture should be standardized and commonly used protocols in order to be easily deployable. Table 4.1 summarizes the derived requirements for the interconnection of gateway and application overlays.

1. Independency of P2P overlays' architecture and middleware
2. Synchronous and asynchronous intercommunication of overlays
3. Independency from lower layers
4. Scalability in terms of the number of overlays to be interconnected and the number of nodes in each overlay
5. Self-organization
6. No centralized entity
7. Efficient in use of resources
8. Support of light-weight and standard protocols

Table 4.1: Derived requirements for the interconnection of gateway and application overlays

4.2.2 Related work on the interconnection of overlays

In most of the existing P2P applications, all the functionalities are included in a single overlay. Although this overlay may be organized in multiple levels, peers in different levels still belong to the same overlay and communicate by use of the overlay protocol. In the literature there are two motivations for services which are based on cooperation of multiple separated overlays. The first motivation is to interconnect several end-user application overlays in order to extend the existing services. The second motivation is to increase the overall performance by dividing one multi-service overlay to several uni-service overlays and interconnecting these overlays by use of more efficient mechanisms.

There are at least two categories of approaches for interconnection of different overlays. In the first category a node is selected in each of the underlying overlays and these nodes together form an interconnection overlay which provides inter-overlay connectivity. Protocols and mechanisms which provide inter-overlay information exchange via the interconnection overlay can differ from one approach to another. The second category is based on an interconnection node which belongs to all the underlying overlays to be interconnected. In this subsection, two related works for each of the discussed categories are presented.

4.2.2.1 Interconnection overlay approaches

In [80], authors propose a multiple P2P overlay architecture for large-scale emergency services. Each individual emergency service, such as medical, rescue, fire, and police emergency responders, is presented in an overlay. In addition to ordinary team members, which can communicate only with members in their own overlay, each overlay has a team leader. These nodes form an interconnection overlay and enable the inter-overlay information exchange. Proposed architecture is shown in Figure 4.2. Inter-overlay and intra-overlay communication mechanisms are based on structured DHT overlays where the DHT domain is divided into several sections, each of them dedicated to one of the involved services. The integral part of [80] focuses on cross-layer design approach in order to provide an overlay topology which is based on the underlying MANET. This will result to considerable overlay performance improvements. Cross-layer design violates the layered protocol stack paradigm by enabling communication of non-adjacent layers. In this work the DHT-based architecture utilizes the routing layer information to build an overlay topology similar to the real network.

This architecture meets our requirement on scalability in terms of number of overlays to be interconnected. Any team leader which joins the interconnection overlay offers its resources to the overlay based on P2P paradigm. However, when the number of nodes in one overlay increases, corresponding team leader may become overloaded. This is due to the fact that only a single team leader is presented per overlay. Therefore, this architecture partially satisfies the requirement on scalability. Moreover, formation and maintenance of the interconnection overlay

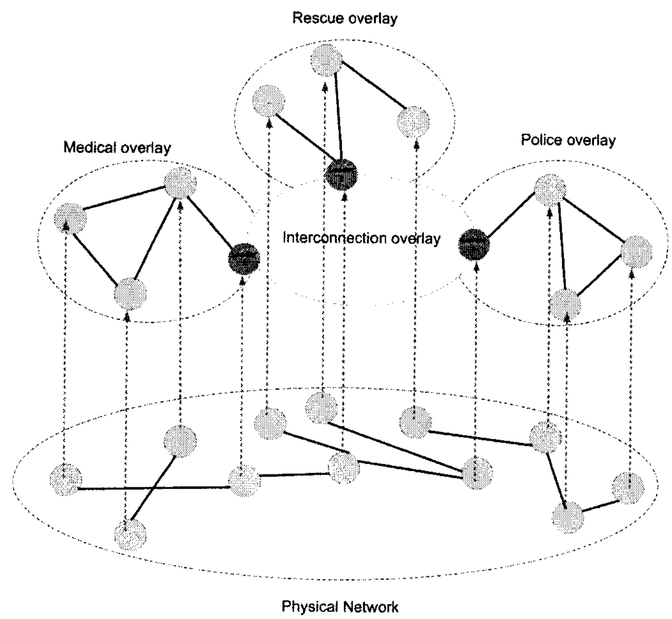

Figure 4.2: Multiple P2P overlays architecture for emergency services

introduce an overhead to the network. Therefore, this approach is not efficient in terms of usage of resources. However, by deploying the cross-layer design, authors aim at reducing the overall overhead. The proposed architecture is self-organized and does not need a centralized entity since it deploys the interconnection overlay self-organization mechanisms. However, this architecture does not fulfill two of our most essential requirements. It considers only a specific P2P overlay architecture and overall mechanisms are based on network layer protocols such as AODV [41] and OLSR [39]. The deployed application layer protocols are also not discussed.

As the second related work, in [81] authors propose a hierarchical two-level P2P overlay architecture named as H-P2PSIP. This architecture interconnects several P2PSIP domains and provides global distributed multimedia services based on standardized protocols. P2PSIP protocol [82] combines the SIP protocol [5] and DHT-based P2P resource distribution and discovery mechanisms. This protocol can be deployed for decentralized communication inside a single SIP domain. As it is illustrated in Figure 4.3, H-P2PSIP interconnects different P2PSIP domains presented as P2P overlays, such as Chord, Kademlia and CAN. The inter-domain architecture is based on an interconnection overlay which is formed by nodes from each overlay which have the most capabilities (super peers). When a peer searches a multimedia resource, it first tries the overlay to which it belongs. If the resource cannot be found in the original overlay, the peer sends a discovery query to the super peer of its overlay. It is assumed that all peers are aware of the contact information of at least one super peer in their domain. Discovery mechanism will then be performed in second layer of hierarchy between the super peers of different domains. Based on the result, query will be forwarded to appropriate super peer which will transmit it to the peer in its overlay which may store the requested resource. Query response will be sent back to the original peer along the same path.

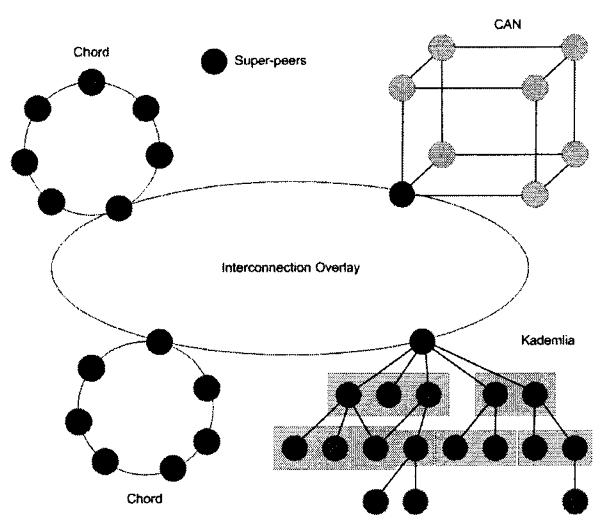

Figure 4.3: H-P2PSIP architecture

Similar to the previous related work, this architecture is self-organized and does not need a centralized entity. Due to hierarchization, it is scalable in terms of number of overlays to be interconnected. However, it is not clarified if several nodes can become the super peer in each overlay. Therefore, this architecture is partially scalable. It is also independent of lower layer protocols since it is completely based on P2PSIP. Although hierarchization results to an overhead for the discovery mechanism, deploying light-weight protocols, such as P2PSIP, which lead to more efficient use of resources, may compensate this drawback. However, this architecture only considers the interconnection of DHT-based structured overlays and keyword-based search mechanisms and does not provide a generalized interconnection architecture. Therefore, our first requirement is still not satisfied.

4.2.2.2 Interconnection node approaches

As the first approach in this category, [83] presents mechanisms which enable a P2PSIP User Agent (P2PSIP UA) to participate in multiple DHT-based overlays, such as OpenChord [84] and Bamboo [85]. These mechanisms are based on a new conceptual module known as DHT plug-in. Figure 4.4 illustrates architecture of the plug-in module. The main task of the plug-in is to provide a flexible and scalable interface which translates the underlying DHT API into a generic API interpretable for an application end-user.

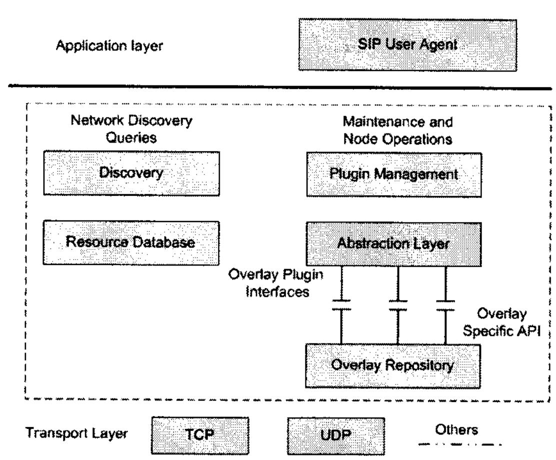

Figure 4.4: Plug-in module architecture

Plug-in management, resource database and overlay repository entities enable the following tasks: end-users can join the overlay via a bootstrap node, map its SIP URI to the actual contact address, insert, remove or search the data objects and finally leave the overlay. A node uses the discovery module to multicast the bootstrap node's discovery requests whenever it wants to join the overlay. Once a bootstrap node received the query, it responds the requester with its current location information. To ensure scalability and reduce the overhead of multicasting, a node may temporarily store the bootstrap node address whenever it joins the overlay. Therefore, in the next joining procedure it will first try to contact the stored address before going through the multicasting mechanism. Intercommunication of heterogeneous DHT-based overlays is illustrated in Figure 4.5. A node first discovers the bootstrap node, gets the detailed information of the overlay to which it wants to join and joins the corresponding overlay. Now, it can communicate with nodes in the same overlay as well as the ones in other overlays. For inter-overlay communication, a node should first discover the destination overlay and send an INVITE request to it by use of the DHT plug-in module. Once the request is received, the internal operation of the overlay will be performed to find the actual contact information of the destination node. The INVITE message will then be forwarded to the destination node. Related response will be sent back to the source node in the originating overlay and communication will be set up directly between two nodes.

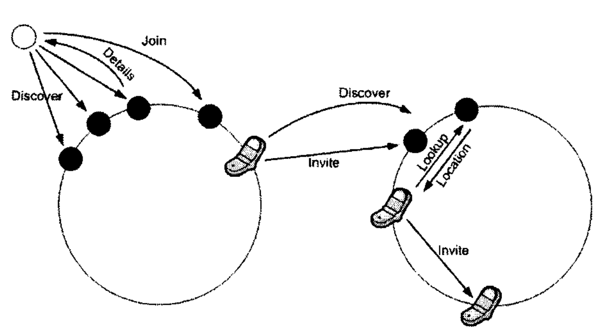

Figure 4.5: P2PSIP inter-overlay communication

This architecture meets our requirements on self-organization and decentralization. It also deploys P2PSIP which is a light-weight and standard protocol and leads to efficient use of resource-constraint mobile devices. Moreover, the architecture is based on application layer protocols. It can also cover a wide range of applications based on synchronous and asynchronous communication of multiple overlays. However, this architecture lacks of scalability and it is not clear which nodes are destined to inter-overlay communication purpose. In addition, introduced mechanisms and plug-in module only considers structured DHT-based overlays.

Another approach is presented in [86]. In this work, the maintenance overhead improvements by coexistence of structured and unstructured P2P overlays in the same node, is analytically illustrated. Pastry [62] and C-gossip [87] are deployed as structured and unstructured overlays, respectively. Information related to the construction of each overlay is divided into primary and secondary components. A primary component has more overlays-based constraints than a secondary one. The structure of Pastry overlay is constructed by use of two sets of information: a set of closest node identifiers as the primary component and a routing table as the secondary component. C-gossip unstructured network maintains the interest-based links as primary component while a set of pointers to random nodes forms the secondary component. As it is illustrated in Figure 4.6, authors prove that in case of cohabitation of a node in two overlays, the primary component of one overlay can be deployed as the secondary component of another overlay. Therefore, only the primary components of each of the overlay need to be maintained.

The proposed approach does not fulfill many of our requirements. First of all, it depends on specific overlay middleware and P2P architectures and relies on lower layer protocols. In addition, it does neither address communication in the joint overlay, nor self-organization operations. This system does not scale in terms of number of overlays that can cohabit and the overall solution is only limited to the

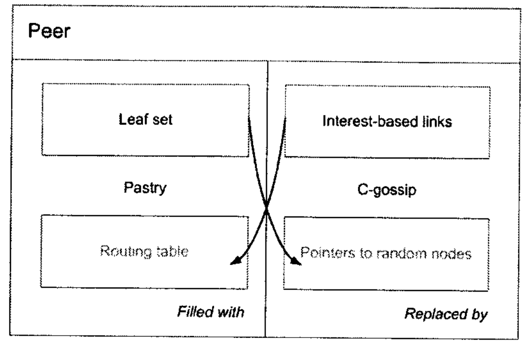

Figure 4.6: Pastry and C-gossip overlays coexistence

coexistence of two overlays. Moreover, assuming that all the nodes in two overlays should support joint overlay coexistence and the consecutive amount of exchanged messages in the network, the overall system cannot be efficient in terms of the resource consumption.

4.2.3 Evaluation summary

Table 4.2 summarizes our evaluation of the related work presented for interconnection of overlays. As it can be seen from the table, all of the discussed approaches are based on specific overlay architectures. None of these works provides P2P overlay middleware and architecture-independent protocols and procedures for the interconnection of overlays. Moreover, both of the defined categories do not satisfy some of our requirements. However, this evaluation leads to two interesting observations. First of all, comparing the two categories of approaches, deployment of an interconnection node can considerably decrease the overall overhead of interconnection overlay formation and maintenance. This issue is essential for our resource-constraint mobile devices where interconnection procedure is a subset of the whole operation and should be as light as possible. Second, the common problem of interconnection node category is scalability. Appropriate modifications and extensions should be provided to meet this requirement.

Requirements \ Related works	Architecture for Multiple P2P Overlays in MANET in Emergency Situations	H-P2PSIP	P2P SIP User Agent With Support for Multiple Overlays	Leveraging the Coexistence of Multiple P2P Overlay Networks
1. Independency of P2P overlays' architecture and middleware	Not satisfied	Not satisfied	Not satisfied	Not satisfied
2. Synchronous and asynchronous intercommunication of overlays	Satisfied	Not satisfied	Satisfied	Not discussed
3. Independency from lower layers	Not satisfied	Satisfied	Satisfied	Not satisfied
4. Scalability in terms of the number of overlays to be interconnected and the number of nodes in each overlay	Partially satisfied	Partially satisfied	Not satisfied	Not satisfied
5. Self-organization	Satisfied	Satisfied	Satisfied	Not discussed
6. No centralized entity	Satisfied	Satisfied	Satisfied	Satisfied
7. Efficient in use of resources	Partially satisfied	Partially satisfied	Satisfied	Not satisfied
8. Support of light-weight and standard protocols	Not discussed	Satisfied	Satisfied	Not discussed

Table 4.2: Summary of the evaluation of the related work for interconnection of overlays

In the next section, with the idea of interconnection node in mind, we refine our overall architecture by introducing the interconnection of gateway and application overlays related mechanisms and protocols.

4.3 Refining the Overall Architecture: Interconnection of Gateway and Application Overlays

This section is organized in four subsections. In the first two subsections, we introduce the assumptions and role of the entities for the interconnection of gateway and application overlays. The inter-overlay interaction and interconnection protocol are discussed in the third subsection. The last subsection describes the operational procedures related to the refined architecture.

4.3.1 Interconnection assumptions

Our fundamental assumption is that the gateway and application overlays are designed with interconnection in mind. This means, for instance, that the nodes in the overlays to be interconnected are aware of the roles they play in their respective overlays. In addition to their basic roles, some nodes are able to eventually play additional roles. These roles are newly defined for the interconnection purpose. It also means that some nodes belong to both gateway and application overlays. These nodes may already play roles in both overlays or may be introduced, for the purpose of interconnection, into the real network on which the overlays are built.

4.3.2 The roles of the entities

For the purpose of interconnection we introduce new roles in each overlay of our overall architecture. These roles manage the interconnection mechanisms, interaction between the peers and provide inter-overlay communication between gateway and application entities. Refined architecture for the interconnection of gateway and application overlays is illustrated in Figure 4.7. We define two new roles: interconnector and super interconnector. To meet our requirements on scalability, decentralization and efficient use of mobile devices resources, several interconnectors and super interconnectors may exist at the same time. We also assume that each interconnector can eventually take the role as a super interconnector.

Interconnector

An interconnector is a node which belongs to both overlays. In other words, it can be seen as a node that plays an interconnection role in each overlay. The corresponding role in each of the application and gateway overlays is called APPlication InterConnector (APPIC) and GateWay InterConnector (GWIC), respectively. There is no direct communication between application or gateway components and an interconnector and all the communication between gateway and application overlays should be done through the interconnectors.

Super interconnector

A super interconnector is a more capable node than other nodes and in charge of a set of interconnectors. Each super interconnector belongs to a single overlay. The super interconnector in the application overlay is named as Super APPlication InterConnector (SAPPIC) and the one in the gateway overlay as Super GateWay InterConnector (SGWIC). All the super interconnectors in each overlay have a common super interconnector group address which is known by all the other nodes in the overlay. Super interconnectors keep track of the information related to their corresponding interconnectors. In addition to this, they maintain the information related to the components in their overlay (i.e. GWC or AppC) which require intercommunication with the other overlay. A super interconnector is also responsible for selecting an appropriate interconnector for the purpose of inter-overlay communication. Selection is based on a defined criterion, such as the number of messages relayed through an interconnector or its current load level. This information is tracked by the super interconnector. Moreover, each super interconnector dynamically maintains the number of interconnectors and the overlay components connected to it. Finally, it is in charge of handling all departures, including the voluntarily and unintentional departures of the interconnectors.

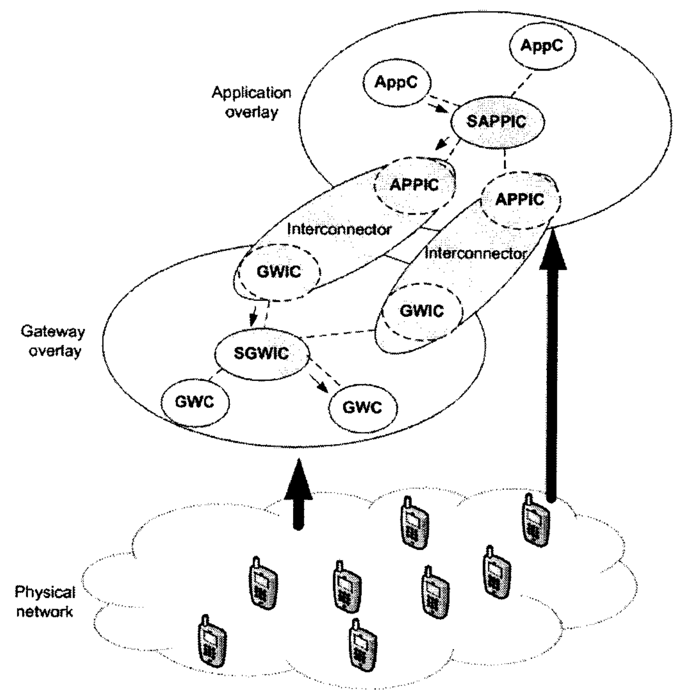

Figure 4.7: Refined architecture for the interconnection of gateway and application overlays

4.3.3 Overlays interconnection protocol

Whenever a source node, such as an Application Component (AppC), in an overlay wants to communicate with a destination node, such as a GateWay Component (GWC), in another overlay, it sends its request to the super interconnector of its overlay. Super interconnector selects an appropriate interconnector based on the selection algorithm and forwards the message to it. The interconnector connects the two overlays and relay the message to the destination overlay through the corresponding super interconnector. In a symmetric manner, the super interconnector of the destination overlay will send the information to the corresponding node in its overlay. Black arrows in Figure 4.7 illustrates an example of the interaction of peers in gateway and application overlays.

Information exchange relies on a protocol; we call the interconnection protocol, which must be supported by all the involved nodes in the overlays to be interconnected. This ensures that the proposed architecture is independent of specific protocols used in the overlays, and consequently of the architectures of overlays to be interconnected. We have selected the Session Initiation Protocol (SIP) [5] to realize the inter-overlay interactions. SIP meets all the requirements for overlays interconnection which we derived in the previous section. First of all, it is inherently a P2P protocol. It is also a light-weight standard protocol which is easily extensible, widely deployed and interoperable with a variety of mobile devices. Moreover, it is independent of the P2P overlay middleware and architecture and can also carry the required information with an acceptable overhead. Furthermore, SIP has been used successfully in overlay self-organization procedures and provides a toolkit which offers more facility and rapidity for development and validation of our overlay interconnection procedures.

The only entity of SIP that we use is SIP User Agent (SIP UA). Registrar, proxy and redirect servers are not required in our approach. To realize our defined procedures, two extensions of SIP methods are deployed. The first one is the INFO method [88] which is used to implement the discovery purpose, and the second one is the SIP event notification framework (SUBSCRIBE/NOTIFY) methods [89] for self-organization, self-recovery and event-based inter-overlay information exchange operations. In order to adapt these methods to our procedures, we add a new header which is the "role" of the transmitter of a message. This role can be the interconnector, super interconnector or gateway and application component which requires the inter-overlay communication.

4.3.4 Operational procedures for the interconnection of application and gateway overlays

In this subsection, we describe operational procedures which are induced by the newly introduced interconnection roles and also the gateway and application components which require inter-overlay communication. These procedures can be divided into two sets of operations. The first one is related to self-organization and self-recovery of peers in each overlay, also known as overlay churn. The second operation is information exchange flow between gateway and application overlays.

4.3.4.1 Self-organization and self-recovery procedures

In MANET environment nodes may join and leave dynamically and frequently. Self-organization procedures should be accomplished whenever one of the defined nodes, including the interconnector, super interconnector and application or gateway components, joins an overlay. Self-recovery procedures should be put in place in case of a voluntarily or unintentional departure of these nodes from the overlay.

A. Interconnector (APPIC and GWIC) joining the overlay

An interconnector joins the network as a physical node which takes roles in both gateway and application overlays. Once each of APPIC and GWIC nodes joined application and gateway overlays respectively, they try to discover a super interconnector in their corresponding overlay. Discovery mechanism is done by sending discovery messages to a SIP multicast address which is assigned to all the super interconnectors in an overlay and that all nodes in overlay have prior knowledge of it. This message is realized by using the SIP INFO method. The interconnector indicates its SIP contact information and corresponding role in the SIP INFO message. This approach for discovery meets all the requirements presented in the previous section since it is based on the SIP protocol. Each super interconnector which receives the discovery message will respond with a 200 OK message to the requester interconnector. This message contains the super interconnector contact information and the number of interconnectors currently connected to it. Interconnector will choose the super interconnector with the least number of interconnectors connected to it and stores its related contact information which will be used to relay information later on. If this number is equal for all the super interconnectors, the interconnector chooses one of the super interconnectors randomly. This basically will lead to a load balanced grouping of super interconnectors and interconnectors. Load balancing is advantageous for two reasons: first, it increases the scalability in terms of number of nodes in each overlay. Second, it improves efficient use of resource-constraint mobile devices.

Once discovery is done, the interconnector sends a SIP SUBSCRIBE message to the super interconnector's presence event. This event is defined to enable the super interconnectors to dynamically track the set of relevant interconnectors and be aware of the actual interconnectors connected to it. Figure 4.8 shows an example of the discussed subscription message. Super interconnector stores information related to the subscribed interconnectors and responds with a 200 OK message.

```
Call-ID: d10f8b7433a585946b897d6f7c260599@127.0.0.1
CSeq: 1 SUBSCRIBE
From: "APPIC" <sip:APPIC@127.0.0.3:5003>;tag=APPIC
To: "SAPPIC" <sip:SAPPIC@127.0.0.1:5001>
Via: SIP/2.0/UDP
127.0.0.3:5003;branch=z9hG4bK9902581f2f7b1c6901420a6ea04ef9d0323635
Max-Forwards: 70
Contact: "APPIC" <sip:APPIC@127.0.0.3:5003;transport=udp>
Role: "APPIC"
Expires: 3600
Event: Presence
Content-Length: 0
```

Figure 4.8: Example of subscription of APPIC to SAPPIC presence event message

It may occur that an interconnector does not succeed to discover a super interconnector. In this case it will take the role of super interconnector after a predefined amount of discovery period, SIP multicast address will be assigned to it and it can receive discovery and subscription messages from other interconnectors or overlay components. This node participates in the overlay as a node having both roles simultaneously. Figure 4.9 illustrates the interconnector joining operation.

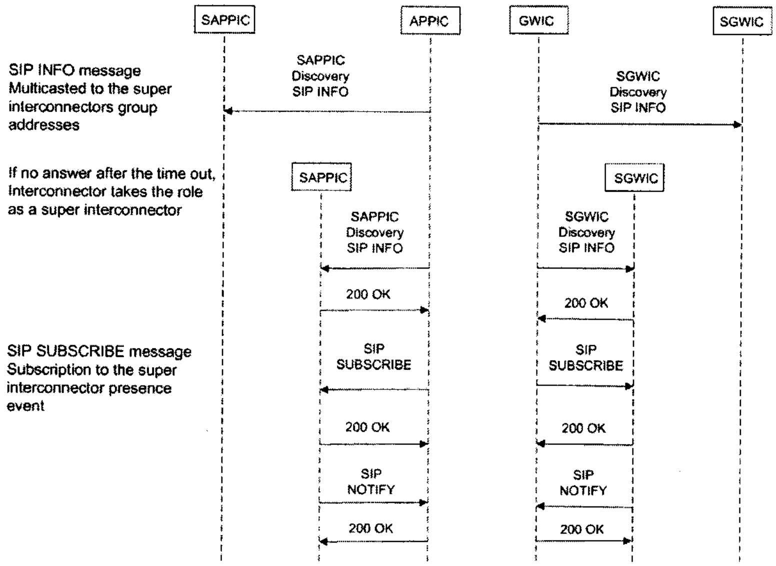

Figure 4.9: Interconnector (APPIC and GWIC) joining the overlay

B. Overlay component (GWC or AppC) requires inter-overlay communication

Gateway or application component which needs inter-overlay communication should introduce itself to the interconnection operation. For instance an AppC may require information which should be provided by the gateway overlay. In this case it tries to discover a super interconnector in its overlays (SAPPIC) by multicasting a SIP INFO message including its role as AppC and its SIP contact information. If no response arrives from the super interconnector, AppC continues the discovery mechanism by sending the SIP INFO messages until it receives 200 OK responses from one or several SAPPICs. This response contains the contact information of the corresponding SAPPIC and the number of current AppC nodes connected to it.

Requester AppC chooses the SAPPIC with the least number of the nodes connected to it and sends a SIP SUBSCRIBE message to subscribe to the presence event of the selected super interconnector. This procedure is presented in Figure 4.10. The same principle applies to the gateway component which requires information from the application overlay.

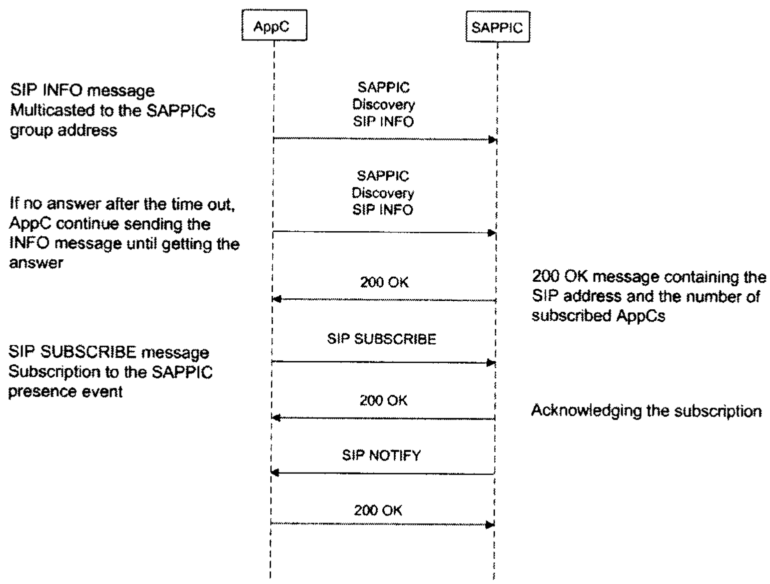

Figure 4.10: Application component (AppC) requires inter-overlay communication

C. Interconnector (APPIC or GWIC) voluntarily leaving the overlay

When an interconnector is leaving the MANET voluntarily, each of its corresponding peers (APPIC and GWIC) informs their super interconnector (SAPPIC and SGWIC) by sending an unsubscription message. This message is a SIP SUBSCRIBE message with the "Expires" field set to zero. An example of the unsubscription message is shown in Figure 4.11.

Super interconnector will respond to the unsubscription message with a final SIP NOTIFY message setting the "expires" field to zero. It will then update the

```
Call-ID: 1cc021ef00c80f4a949039f39dba5f42@127.0.0.7
CSeq: 3 SUBSCRIBE
From: "GWIC" <sip:GWIC@127.0.0.6:5011>;tag=GWIC
To: "SGWIC" <sip:SGWIC@127.0.0.7:5012>;tag=67015f18
Via: SIP/2.0/UDP
127.0.0.6:5011;branch=z9hG4bK2e028e2bee838a33030be5ac1af9558f373036
Contact: "GWIC" <sip:127.0.0.6:5012>
Role: "GWIC"
Event: Presence
Expires: 0
Content-Length: 0
```

Figure 4.11: Example of unsubscription of GWIC from SGWIC presence event message

list and number of interconnectors connected to it. This operation is depicted in Figure 4.12.

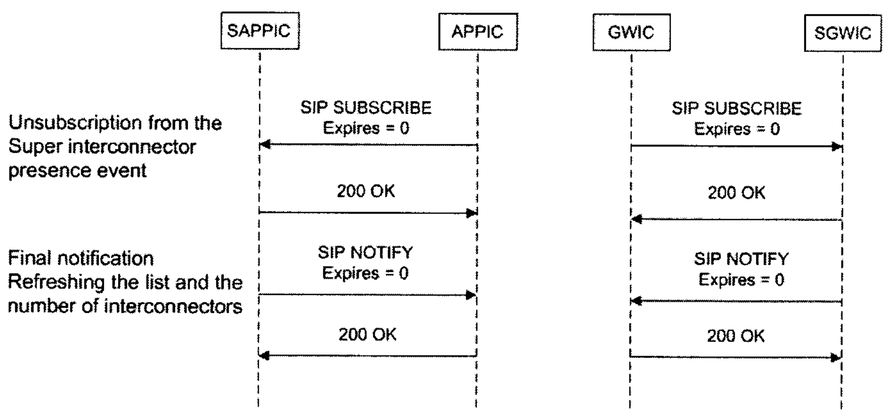

Figure 4.12: Interconnector (APPIC and GWIC) voluntarily departure

D. A subscribed GWC or AppC voluntarily leaving their overlay

This procedure is similar to the case of an interconnector node leaving the overlay. Each of AppC or GWC nodes which wants to leave the overlay, should first unsubscribe from its previous subscription to the super interconnector's presence event. Once the super interconnector received the unsubscription message, it will refresh the list and number of its subscribed GWC or AppC nodes. Figure 4.13

illustrates a voluntarily departure of an application component (AppC) from the application overlay.

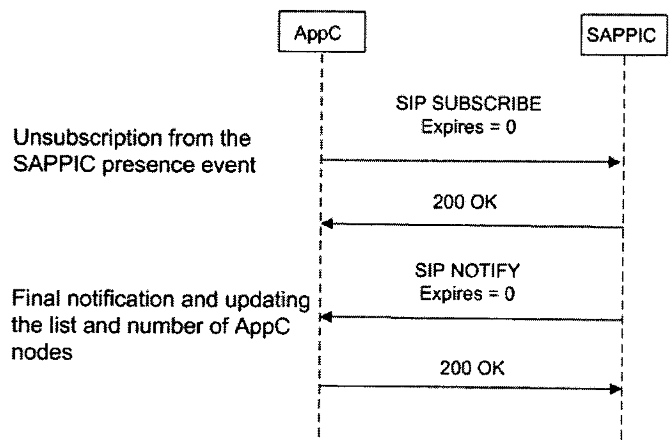

Figure 4.13: Application component (AppC) voluntarily departure

E. Super interconnector (SAPPIC or SGWIC) voluntarily leaving the overlay

If the leaving node is a super interconnector, it should elect one of its subscribed interconnectors as the new super interconnector in charge of a group of interconnectors and overlay components connected to it. First, the super interconnector sends SIP multicast addressing information to the new super interconnector in a SIP NOTIFY message. Then, it informs other subscribed interconnectors and GWC or AppC nodes of the contact information of the new super interconnector. The SIP NOTIFY message is deployed as the notification for the super interconnector's presence event anytime this node leaves the overlay. Interconnectors and GWC or AppC nodes will then subscribe to the new super interconnector which contact information is provided in the SIP NOTIFY message sent by the leaving super interconnector. In this way they do not require to perform any new super interconnector's discovery mechanism. The previous subscription of these nodes will be expired after "expires" duration. Once the subscription is done successfully, the subscribed nodes can continue their inter-overlay communication operation. Figure 4.14 illustrated the voluntarily departure operation of an application super inter-connector (SAPPIC) from the application overlay.

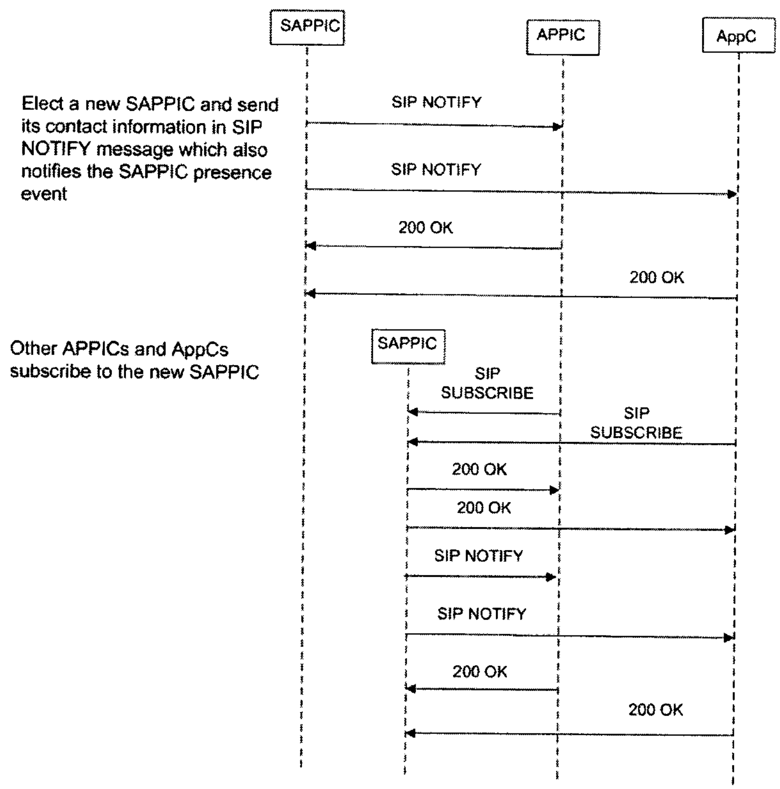

Figure 4.14: Application super interconnector (SAPPIC) voluntarily departure

F. Unintentional departures of the involved nodes

Nodes in MANET may issue unintentional departures which mainly happen due to the lack of devices' power and their consequent failure. In order to detect an unintentional departure of MANET nodes, some failure detection and recovery mechanisms have been proposed. A fast, light-weight and application-level procedure based on request/reply heartbeat protocol is discussed in [90]. This protocol defines a failure monitor, which periodically attempts sending heartbeat requests to the other nodes. The failure monitor in our system is the super interconnector and by deploying the SIP protocol heartbeat request may be SIP NOTIFY messages. Whenever the failure monitor does not receive any response after a predetermined amount of time a failure is detected and the super interconnector refreshes the information related to its connected nodes.

4.3.4.2 Inter-overlay information exchange procedures

Any information which needs to be exchanged between application and gateway overlays should pass through the interconnectors. However, this procedure is transparent to the application end-user. An AppC which wants to get some application-level information based on the WSN provided information needs only to send an application-level query and get the corresponding response. Overlay interconnection protocol performs the entire task of transferring an understandable query to the other overlay, getting the required response and sending it back in an appropriate format to the originating overlay. Our procedures for inter-overlay information exchange rely on SIP subscription and notification methods. SIP SUBSCRIBE message that contains the request for the required information is sent by the application component (AppC) to the application super interconnector (SAPPIC). This node relays the message to a selected application interconnector (APPIC). The request will then be forwarded to the gateway interconnector (GWIC). In the gateway overlay, GWIC sends the relevant SIP SUBSCRIBE message to the gateway super interconnector (SGWIC) to which it is subscribed. The SGWIC will forward the request to one of its subscribed gateway components nodes (GWC). Remaining tasks of retrieving the required information is done in the gateway overlay by deploying the overlay specific protocols. Once the user's subscribed event is detected, required information is retrieved and sent back to the gateway super interconnector as a SIP

NOTIFY message. SGWIC will choose one of the appropriate GWIC nodes and forward the notification message to it. Message will then be relayed to the APPIC which will forward it to its related SAPPIC. This node transmits the SIP NOTIFY message to one of its subscribed AppCs. Application related tasks will then be applied to the retrieved information and the required response will be provided to the application end-user in an appropriate format. In Figure 4.15, the described inter-overlay information exchange flow is illustrated.

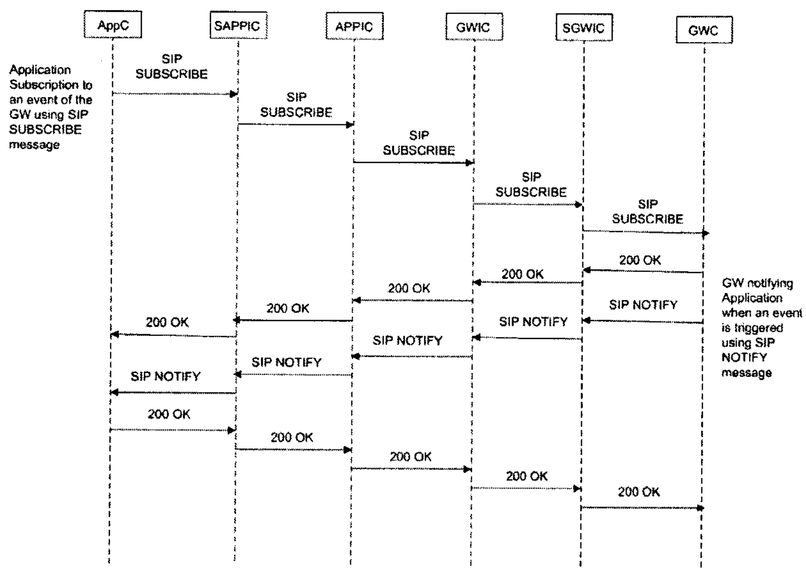

Figure 4.15: Inter-overlay information exchange

4.4 Operational Scenario for the Overall WSN and MANET Integrated System: An Illustration

In this section, we illustrate how the proposed two-level overlay architecture is used along with the overlay interconnection mechanisms. Our scenario is illustrated in

Figure 4.16. This procedure is triggered when an application end-user, who resides in MANET and mapped to an AppC, requests some information that should be provided by WSN. For emergency application, previously described as the motivating scenario in Chapter 1, the required information can be a patient's health status. Therefore, the corresponding request may be a subscription message for an urgent situation related to the patient. In this case, the related response is notification message sent whenever the subscribed event is detected. To illustrate the scenario, we assume that MANET is already established and at least one interconnector and a super interconnector in each overlay are present.

Once all involved nodes joined their corresponding overlays, they start interworking with the interconnection mechanisms. Application and gateway interconnectors (APPIC and GWIC) declare their roles to their corresponding super interconnectors (SAPPIC and SGWIC) and subscribe to its presence event. Application and gateway components (AppC and GWC) also discover the corresponding super interconnector in their overlays and subscribe to its presence event. They are now ready to perform the inter-overlay information exchange operations. Application end-user request is mapped to a SIP SUBSCRIBE message by the AppC peers. This message arrives to the gateway overlay via the interconnection path already discussed in the previous section. Once the SIP SUBSCRIBE message arrived at the gateway, gateway component (GWC) first sends the corresponding 200 OK response. In this stage, GWC peers in gateway overlay of our WSN/MANET integrated architecture perform their roles. They collaboratively process the subscription message and translate this message to an understandable message for the sink. The translated subscription message is then transmitted to the sink via the GWC peers which act as the sink entry points. Sink processes the message and sends the appropriate query to the sensors. When the requested data is received from the sensors, sink sends it back to the entry point peers in gateway overlay. GWC peers store and translate the received data and map the final response to a SIP NOTIFY message. SGWIC and GWIC relay this message to the application overlay of our integrated architecture along the same interconnection path. APPIC and SAPPIC peers forward the SIP NOTIFY message to the AppC. When the AppC receives the SIP NOTIFY message, it sends back the related 200 OK response. This response arrives at the gateway via the same interconnection path. In the application overlay, AppC peers process the SIP NOTIFY message and retrieve the required information from it. This information is processed by AppC using the application overlay specific protocol and is presented in the appropriate format. Consequently, based on the application overlay logic, the final notification message may be sent to the application end-user as an instant message or may trigger a conference between the interested application end-users.

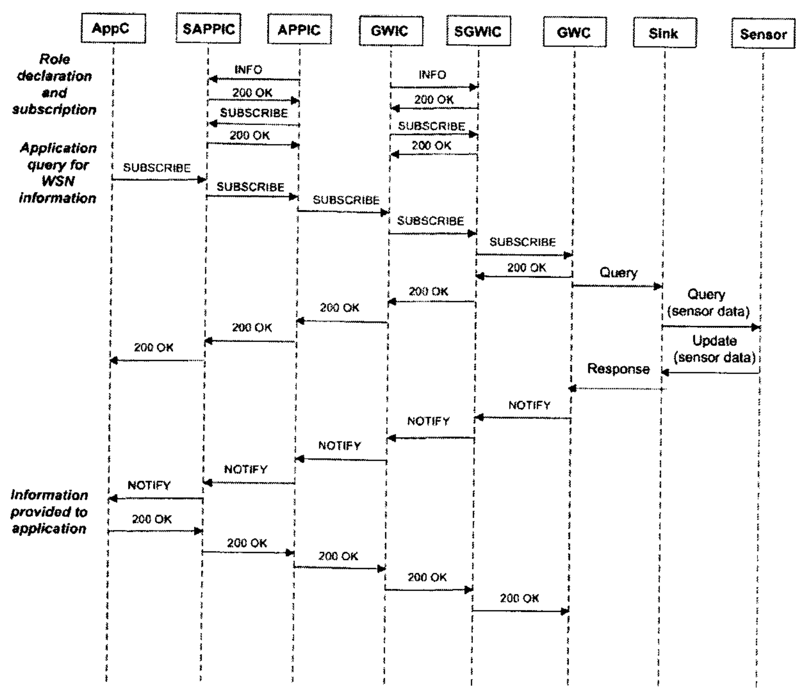

Figure 4.16: Operational scenario for overall WSN and MANET integration

Figure 4.17 shows a specific interconnection information model for the described event-based subscription/notification application. This model consists of an event, a subscriber and an event resource. Event contains a name and some parameters such as the state of the event and presents a set of application end-user required information. For instance, in an emergency application event may be the body temperature of the patients while the state of the event may be active or pending based on the application requirements. Subscriber of an event is presented by a name, contact information and additional parameters, such as the frequency that he wishes to be notified about a subscribed event. Event subscriber in an emergency application may be the doctors or nurses who are interested in a medical event. Finally, event resource is the entity to which the event belongs and consists of a name and some additional parameters. In our example each patient represents an event resource.

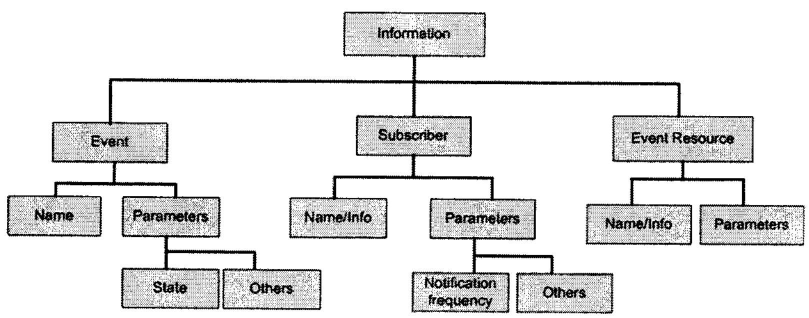

Figure 4.17: Interconnection information model for event-based subscription/notification application

An example of the exchanged subscription and notification messages is illustrated in Figure 4.18 and 4.19. In this example required information to be retrieved from WSN is the temperature of a patient. This is done by defining a subscription event named as "temperature". This information will be notified to the subscriber any time the temperature is higher than a predefined amount.

```
Call-ID: dc91a1c819b5749a9f01ffdbd86f2355@127.0.0.1
CSeq:4 SUBSCRIBE
From: "AppC" <sip:OAPP@127.0.0.1:5000>;tag=AppC
To: "SAPPIC" <sip:SAPPIC@127.0.0.1:5001>
Via: SIP/2.0/UDP
127.0.0.1:5000;branch=z9hG4bKeb15bd50cc26f5753a351dded37828ac333439
Max-Forwards: 70
Contact: "AppC" <sip:AppC@127.0.0.1:5000;transport=udp>
Role:"AppC"
Expires: 1000
Event: Temprature;id=Temprature;Subscriber=Alain; Subsciption=Bob
Content-Length: 0
```

Figure 4.18: Example of a subscription message to the temperature event

```
sip:AppC@127.0.0.1:5000;transport=udp SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5001;branch=z9hG4bKf694d8c54fbdb253adf3bcf2fa621b69323635
CSeq: 6 NOTIFY
Call-ID: 32a979ee1650d594f27f9cbfcb49ee1c@127.0.0.1
From: "SAPPIC" <sip:SAPPIC@127.0.0.2:5001>;tag=10c45b08
To: "AppC" <sip:AppC@127.0.0.1:5000>;tag=AppC
Max-Forwards: 70
Contact: "Notifier" <sip:127.0.0.2:5001;id=not>
Role:"SAPPIC"
Subscription-State: Active
Event: Temprature;id=Temprature;Subscriber=Alain;Subsciption=Bob;Value=38
Content-Length: 0
```

Figure 4.19: Example of a notification message for the temperature event

4.5 Summary

In this chapter, we proposed a two-level overlay architecture for integration of WSN and MANET. The main functional entities of this architecture are gateway and application overlays. To interconnect these overlays, we refined our architecture by introducing relevant protocols and mechanisms. Self-organization and self-recovery mechanisms along with inter-overlay information exchange procedures have been designed. Moreover, an operational scenario has been provided to illustrate the overall integration of WSN and MANET. A specific interconnection information model for event-based subscription/notification applications has been discussed at the end of this chapter.

The refined architecture satisfies all the requirements we derived for interconnection of overlays in this chapter as well as the general requirements we derived for integration of WSN and MANET in the previous chapter. Our first general requirement is decentralization of WSN/MANET integrated system. Based on our overall architecture, formation and management of gateway and application overlays do not require any centralized entity. For purpose of interconnection, we have defined new roles in each overlay which are dynamically assigned to the nodes operating as part of the existing gateway and application overlays. Second requirement is support for self-organization and self-recovery of the entities of the integrated system. Our architecture enables self-organization and self-recovery by deploying application and gateway overlay specific protocols. Moreover, we defined a SIP-based interconnection protocol for self-organization and self-recovery procedures of the newly introduced nodes in the overlays. As the third requirement, due to decentralized nature of overlays and deployment of several interconnectors and super interconnectors in the interconnection mechanisms, our architecture scales well in terms of number of overlays and number of nodes in each overlay which is similar to number of MANET participants. Rapid establishment of the system is fulfilled since fast setup is one of the P2P overlays characteristics. Moreover, by deploying the interconnection node approach for interconnection of gateway and application overlays, the inter-overlay operations can be performed rapidly without a need to form an interconnection overlay. Both interconnection protocol and the internal gateway overlay protocol which we extended in this work are based on SIP. Therefore our requirements on light-weight and standard protocols are also satisfied. Decentralization and deployment of light-weight protocols render our solution to be efficient in use of mobile devices' limited resources. Overlays as virtual application-layer networks decouple application and gateway components from the lower layer protocols. Our interconnection protocol and mechanisms are also independent of the lower layer protocols. Our last general requirement is thereby fulfilled.

In the next chapter, we will discuss the implementation of the proposed inter-overlay communication mechanisms and preliminary performance evaluations.

Chapter 5

Design of Modules, Proof of Concept and Preliminary Performance Evaluations

In the previous chapter we have refined our architecture to interconnect gateway and application overlays of the overall WSN/MANET integrated system. Refinement is done by introducing new roles for the purpose of interconnection in each overlay. We now focus on the design of these modules and validation of our proposed architecture through the prototype implementation and simulation. In this chapter, first the design of each of the induced interconnection modules is addressed. In the second section, prototype we implemented as a proof of concept for the inter-overlay communication is introduced. In order to validate our architecture for a large scale scenario, in the third section, we discuss some performance measurements collected from simulation using Oversim [6] simulation tool.

5.1 Design of the Interconnection Modules

The software architecture of an interconnector is depicted in Figure 5.1. It consists of three sets of modules and protocol stacks. Self-organization module manages discovery and role declaration of the interconnector to the super interconnectors by deploying SIP multicasting mechanism. Based on the information received from the super interconnectors, corresponding interconnector chooses the super interconnector with the least number of interconnectors. Subscription to the presence event of the super interconnector and interconnector departure procedure are also included in this module. APPIC and GWIC message relay modules deal with inter-overlay transfer of messages based on event-based subscription/notification procedures. Self-organization and message relay modules provide all the interconnection functionalities by using the SIP protocol stack. Application and gateway overlay-specific modules use the corresponding overlay protocol stacks to provide overlay-specific functionalities which are internal to each overlay and may completely differ based on the overlay logics. Interconnector uses these modules to participate in overlays' internal tasks.

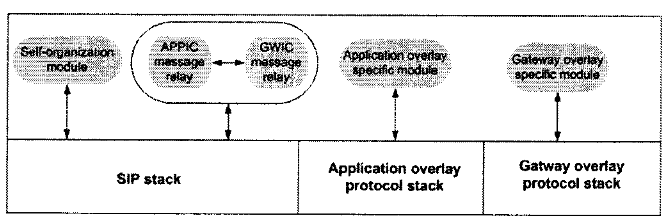

Figure 5.1: Interconnector software architecture

Figure 5.2 illustrates software architecture of an application super interconnector, which is very similar to the architecture of gateway super interconnector. Self-organization module is responsible for keeping track of APPICs and AppCs connected to SAPPIC. This information will be sent to the interested APPIC or

AppC nodes during the discovery procedure. It also manages departure of SAPPIC by sending notifications to subscribed APPIC and AppC nodes. APPIC selection module, which communicates with self-organization and message relay modules, is in charge of selecting the appropriate APPIC in two situations. First is to replace the leaving SAPPIC by choosing a new SAPPIC from the connected APPICs. Second is to select the appropriate APPIC based on a defined criterion which is specific to each application. This criterion may be the number of messages already relayed to an APPIC as an attempt to balance the load on each APPIC and improve the resource consumptions. APPIC message relay deals with the transmission of messages from SAPPIC to the selected APPIC. These three modules are based on the SIP protocol stack. Application overlay-specific module is as it was defined earlier.

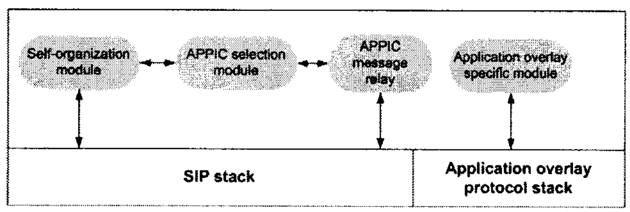

Figure 5.2: Application super interconnector software architecture

As it is depicted in Figure 5.3, an application or gateway component which requires inter-overlay communication is provided with a plug-in module. All procedures required for communication of AppC or GWC with their corresponding super interconnector are implemented in this module. Self-organization module is responsible for super interconnector discovery, role declaration, subscription to super interconnector presence event and overlay component departure. Message relay module provides the inter-overlay information exchange. All of these procedures are based on the SIP protocol stack. Overlay specific module is as described earlier and based on the overlay protocol stack.

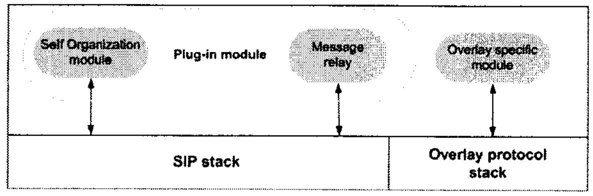

Figure 5.3: Overlay component (AppC or GWC) software architecture

5.2 Overall Prototype Design and Implementation

As a proof of concept, we have implemented a prototype for information exchange between application and gateway overlays and ran it in a small-scale 802.11 ad-hoc network. In this section, the environmental setting of our prototype, the software tools we have used for implementation and the overall prototype main components along with a simple experimentation result are discussed.

5.2.1 Environmental settings

In order to test and experiment with our solution, a MANET environment was created. As it can be seen in Figure 5.4, this network consists of five nodes (2 laptops and 3 PCs) which are all equipped by 802.11 (WLAN) adaptive cards. PCs are dual core Intel E5400 machines and laptops are dual core Intel T5500, all with 2 GB RAM and running Windows XP. Each of these machines is assigned by one of the roles defined for the interconnection purpose.

5.2.2 Software tools

We choose Java based development APIs for the implementation of our prototype. Three types of Java middlewares are used for this purpose. Different applications can be rapidly developed by using the middleware provided services and APIs. However,

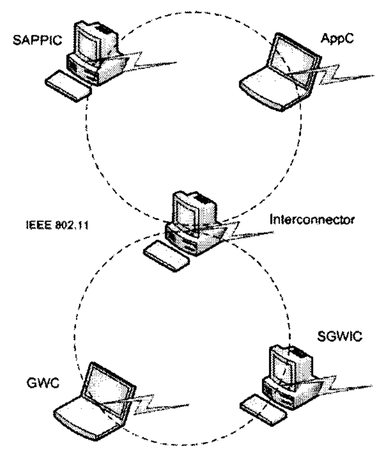

Figure 5.4: Prototype environmental settings

most of the existing middleware systems do not support P2P overlay issues such as peer naming, service discovery, message routing and overlay security concerns. The first two middlewares in our prototype are used for realizing the P2P application and gateway overlays and the third one is to implement the inter-overlay communication mechanism. We use Netbeans IDE 6.8 as the development environment.

5.2.2.1 Application and gateway overlays

Currently, a very limited number of implemented general-purpose P2P middleware exists. JXTA [59] and Open Chord [84] are two existing P2P middleware that provide generic frameworks and set of building blocks to develop P2P applications.

JXTA, as the most advanced general-purpose P2P middleware, includes a set of open P2P protocols [91]. These protocols have been designed to provide multi-hop communication of peers in ad hoc networks. However, JXTA does not target high dynamicity in MANET environments. For our prototype, since we focus on the interconnection of overlays rather than their internal processes, JXTA provided functionalities are well-placed for the creation of our application overlay. These protocols include service discovery, peer request resolver, peer status information retrieval, rendezvous, end-point routing and pipe binding protocols. Messages in JXTA overlay are in XML format and underlying lower layer protocol is mostly TCP/IP [92]. The architecture of JXTA overlay is illustrated in Figure 5.5. It consists of JXTA core, service and application layers. JXTA core is responsible for the creation of peers and peer groups. Peer groups are a set of peers which are assigned with the same role. It also establishes the pipes which provide virtual communication of peers in the overlay. Service layer provides generic services that are required by many P2P applications, such as searching and indexing. Specific tasks of each P2P application, such as instant messaging and document sharing, are included in the application layer [91].

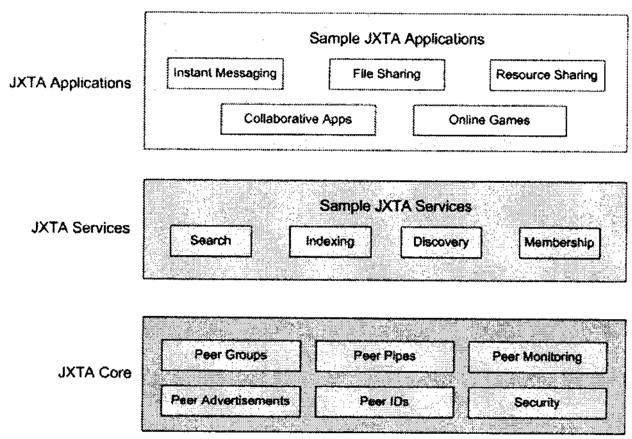

Figure 5.5: JXTA overlay architecture

To form a JXTA overlay, we create a startJXTA method in which a bootstrapping peer is defined as a new instance of JXTA NetworkManeger. Attributes of this peer, such as name and contact information, can be set during the peer creation. Other nodes can join the overlay through the bootstrapping peer and participate in the overlay corresponding operations. As discussed in [4], for simplicity, in our prototype the bootstrapping node is the application super interconnector and other joining nodes can directly contact it without performing the discovery operation.

For the creation of our gateway overlay, we choose Open Chord, an open java-based implementation of Chord DHT overlay network [93]. This middleware provides an interface to develop P2P applications which may require DHT for data storage and retrieval. Open chord architecture also contains three layers as it is shown in Figure 5.6. Communication layer embeds lower layer communication protocols. Communication abstraction layer deals with the virtual communication of peers. The main functionalities of DHT Chord overlay network are implemented in the Chord logic and java applications can be developed on top of this layer.

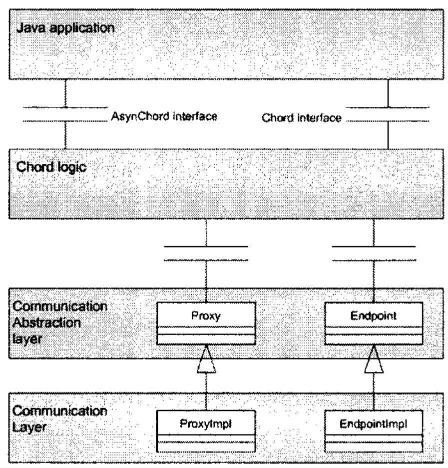

Figure 5.6: Open Chord overlay architecture

We create the gateway overlay of our prototype by defining a startOpenChord method. In this method, we define protocols and a URL which is used to create the Open Chord overlay. Similar to the application overlay, gateway super interconnector acts as the bootstrapping node in the gateway overlay through which other peers can participate in the overlay using the Join method. They can then insert, retrieve or remove data in DHT by using the corresponding methods.

5.2.2.2 Interconnection of two overlays

To implement new concept of overlays' interconnection and its corresponding SIP-based protocol we used JAIN SIP API [94], a standardized Java-based interface to implement SIP protocol stack. This interface provides SIP UA required functionalities that should be supported by all the nodes involved in the interconnection. JAIN SIP supports all our required SIP specifications, including SIP INFO (RFC 2976) method and SIP Event notification framework (RFC 3265). We use the provided SIP messages, headers and dialog concepts in JAIN SIP. Based on our interconnection protocol, we need to extend and adapt the existing INFO and SUBSCRIBE/NOTIFY methods. In addition to the existing headers in these messages, we added a "Role" header which indicates the roles that the transmitter of the message is playing in its overlay. INFO method is used for discovery purpose between an interconnector or an overlay component (GWC or AppC) and the corresponding super interconnector. SUBSCRIBE/NOTIFY method is used for self-organization, self-recovery and event-based information exchange procedures. We also define a new event type, known as "presence", which will be indicated in the event header field of SUBSCRIBE/NOTIFY messages for the purpose of self-organization and self-recovery. The dynamic event types will be defined for each specific application. For instance, in our emergency subscription/notification prototype, a "temperature" event is defined to keep track of the patient's body temperature information and transfer it to the interested medical staff.

To provide a proof of concept, a subset of the inter-overlay information exchange scenario is implemented. SAPPIC and SGWIC join the application (JXTA) and gateway (Open Chord) overlays respectively as the bootstrapping nodes. AppC and GWC nodes join their corresponding overlays through the bootstrapping nodes. The fifth node is the interconnector (APPIC and GWIC) between application and gateway overlays which also joins the two overlays through the corresponding bootstrapping nodes. Interconnection is then established between the application and gateway overlays to provide the end-user service. Whenever a predefined event, such as the temperature higher than a threshold to which a user has been subscribed is triggered, a text notification is sent to the application end-user via the established inter-overlay path.

5.2.3 Main components and information exchange

Main components of our inter-overlay communication prototype as illustrated in Figure 5.7 depicts a simplified view of the corresponding java class diagram.

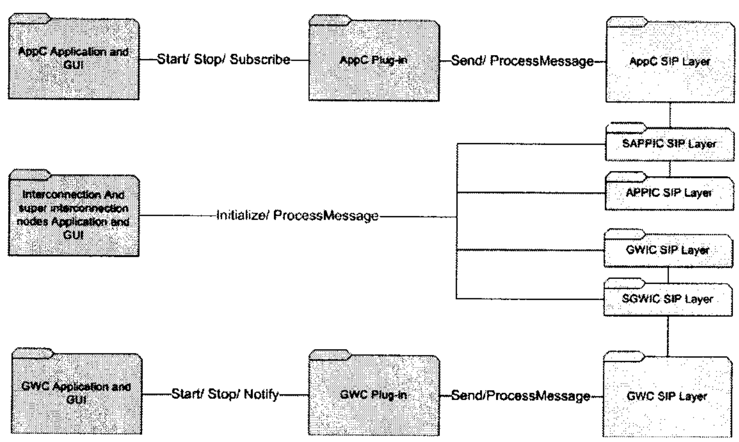

Figure 5.7: Prototype main packages

Application and GUI classes implement an interface for gateway and application components to subscribe and trigger a predefined event and consult the corresponding information. For instance in AppC, health notification application includes a graphical user interface through which an end-user can consult the medical information related to the patients. Plug-in module encapsulates the defined interconnection protocols and provides APIs for the procedures defined in Figure

5.8. Information exchange is realized via the SIPLayer class of each of the implemented nodes. The SipLayer class implements the SIP UA in order to exchange the event-based interconnection SIP messages.

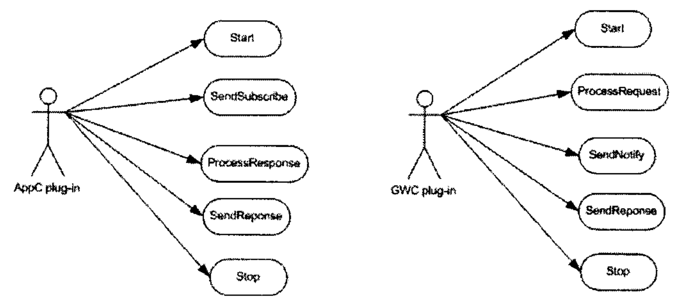

Figure 5.8: Prototype main actors

Figure 5.9 depicts software architecture of our prototype and communication flow between the peers in gateway and application overlays. All the communications for the interconnection purpose use the SIP protocol stack and our implemented SIPLayer. JXTA and Open chord protocol stacks are in charge of application and gateway overlays internal tasks, respectively.

5.2.4 Simple experimentation

The discussed prototype validates our proposed architecture in a small-scale MANET environment consisting of five nodes. We see that the proposed SIP-based inter-overlay communication protocol operates properly in the implemented prototype. However, to understand the real behavior of our introduced mechanism and provide the corresponding detailed measurements we need to run the scenario in a large-scale network. This can be done through the simulation by reusing the same concepts deployed in our prototype implementation.

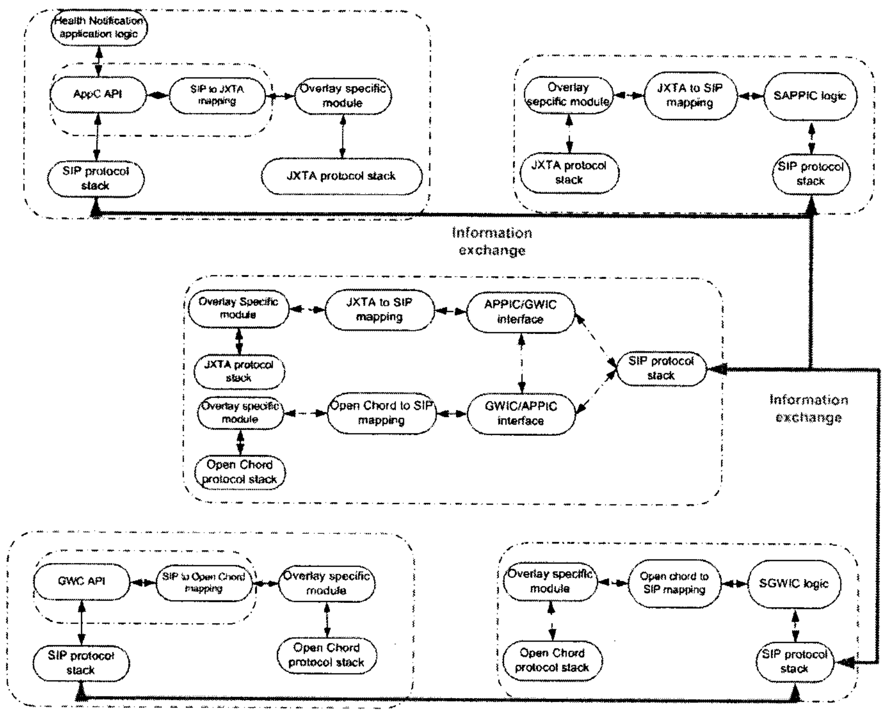

Figure 5.9: Prototype communication flow

5.3 Preliminary Validation and Performance Evaluation

We need to evaluate the performance of our proposed inter-overlay communication mechanisms in a large-scale scenario using the Oversim [6] simulator. Scenarios that we test consist of information exchange between application and gateway overlays and a simplified version of self-organization procedures which we have discussed in our published paper [4]. Some examples of exchanged messages have been discussed in the previous chapter. In this section first the simulation settings and scenarios are discussed. We then introduce measurements and analysis of the collected data.

5.3.1 Introduction to Oversim

Oversim [6] is an open source P2P overlay network simulation framework developed in the Institute of Telematics of Karlsruhe University. This simulator supports several structured and unstructured P2P overlay networks, such as pastry [62], Chord [57] and GIA [95]. It uses the OMNeT++ graphical user interface to visualize the overlay simulation topology and exchanged messages. Architecture of Oversim is based on three flexible modules. The lowest layer is the underlay module which defines the access network parameters, such as bandwidth, packet loss ratio, access delay and node mobility. In our project, we chose the "Simple" underlay network which is a fast model for high simulation performances. The area of simulation is formed by 1500*1500 nodes which are placed in aleatory locations. The second layer is the overlay module which implements the overlay protocols and interacts with the underlay module via a UDP/IP interface. This layer provides supports, such as overlay message handling, service discovery and bootstrapping. The highest layer is the application module which is used to implement different overlay-based P2P applications. These applications communicate with the overlay module via XML-RPC interfaces. To generate the overlay churn, a predefined lifetime based churn model can be used which supports several distribution functions, such as exponential or Weinbull. The user can also create a churn scenario model and integrate it to the system. We setup our simulation configuration parameters to have a realistic overlay network scenario. The configuration parameters are presented in table 5.1.

Simulation time limit which is set to 30 minutes is the amount of time that we run the simulation scenario. We choose Pastry as the application overlay network and define two random churn generators each of them based on different configurations. The first churn generator is related to SAPPIC which we assume to be the only bootstrapping node in our overlay network and always present and the second churn generator is defined for AppC and APPIC nodes. Target mobility delay is the

Overlay network simulation configuration parameters
Simulation time limit = 1800 s
Overlay type = Pastry
Churn generator type = Randm churn
Target overlay terminal number 1 (SAPPIC) = 1 Target overlay terminal number 2 = 50
Target mobility delay = 300 s
Creation probability = 0.5 Removal probablity = 0.5
Init interval = 10 s
Send request interval = 5 s
Appc probablity = 0.5
Statistics interval = 30 s

Table 5.1: Configuration parameters for overlay network simulation

period of time each node stays in a location before moving to a new location and is set to 300 seconds. Terminal creation and removal of random generator are set to equal probabilities. Target overlay terminal numbers is the approximate number of AppC and APPIC nodes in the network with respect to the equal creation and removal probabilities. These nodes participate in the overlay with equal probability. Performance metrics are evaluated by scaling the target number of terminals from small-scale overlays with 5 to 10 nodes to large scale overlays with 50 terminals which is realistic with respect to our motivating scenarios. Since communication is symmetric in application and gateway, the evaluated number of terminals represents only the number of nodes in one of the overlay. Init interval is the amount of time that AppC and APPIC nodes should wait before transmitting their first request. Send request interval which is set to 5 seconds defines the frequency that each node sends a request. In other words, in our simulation each node randomly sends requests with a mean of 1 request per 5 seconds. Statistics interval represents the frequency of statistic data collection.

5.3.2 Simulation setup and scenarios

A view of our simulation setup, including the overlay network terminals and exchanged messages, is depicted in Figure 5.10.

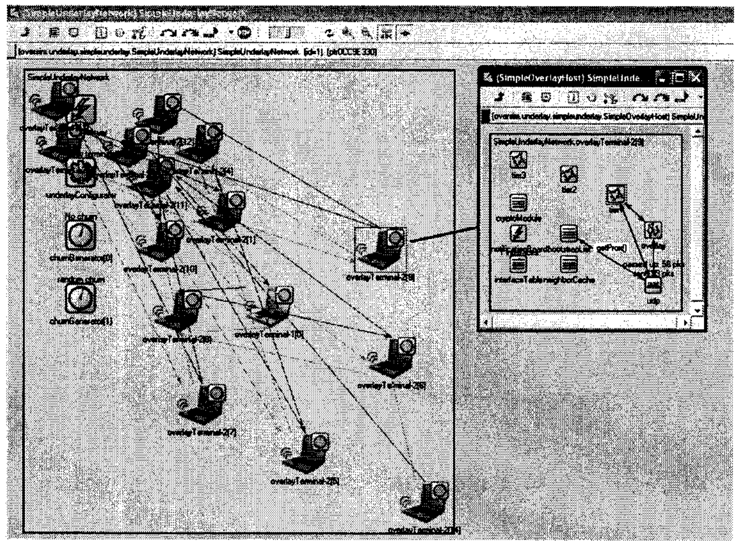

Figure 5.10: Simulation setup

Our application module is implemented on top of the overlay framework by using comprehensive C++ development environment. The simulated overlay composed of a set of nodes which run the overlay component (e.g. AppC), interconnector (e.g. APPIC) implemented logic and one node which act as the super interconnector (e.g. SAPPIC). Since communications in application and gateway overlays are completely symmetric, in our simulation we only focus on the application overlay mechanisms. Simulated state machines of the application overlay nodes are presented in Figure 5.11, 5.12 and 5.13.

SIP messages can not directly be integrated from the existing SIP protocol stacks to the simulation environment. Therefore, we simulate SIP messages and encapsulate them in the application packets which are transmitted through Oversim. A SIP simulated message consists of five fields: Type field defines the SIP method

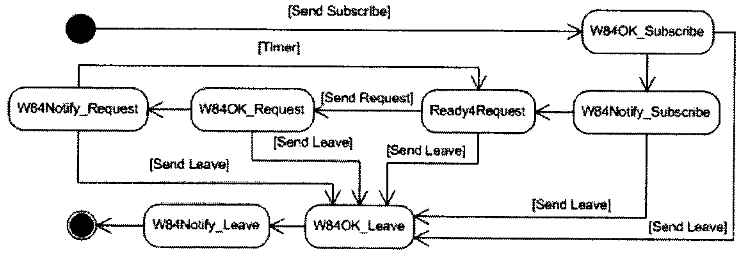

Figure 5.11: Application Component state machine

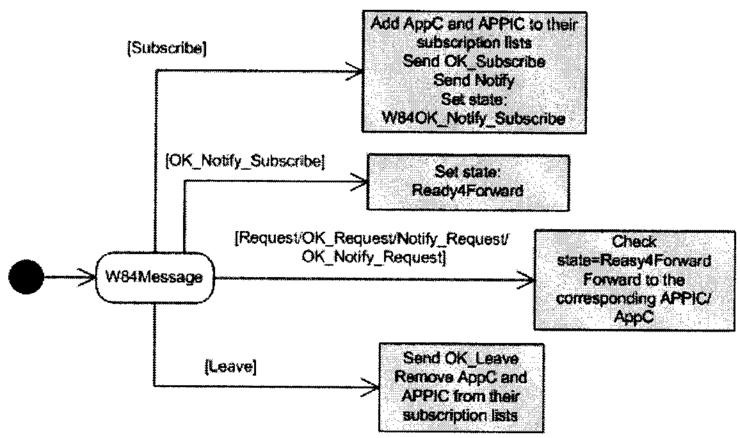

Figure 5.12: Super APPlication InterConnector state machine

that is sent from each of the AppC, APPIC and SAPPIC nodes. Value field is equal to the SIP message body and related information. Source and destination transport addresses are used to convey the message in the overlay. Time stamp is added to the packet to enable a precise measurement of message life time. A SIP SUBSCRIBE packet is illustrated in Figure 5.14 as an example.

Two scenarios are implemented to evaluate inter-overlay mechanisms of our proposed architecture. The first scenario is related to self-organization aspects of the interconnection protocol and consists of subscription of overlay components and interconnectors to super interconnector's presence event. The second scenario is based on the transmission of requests between the defined overlay nodes through a

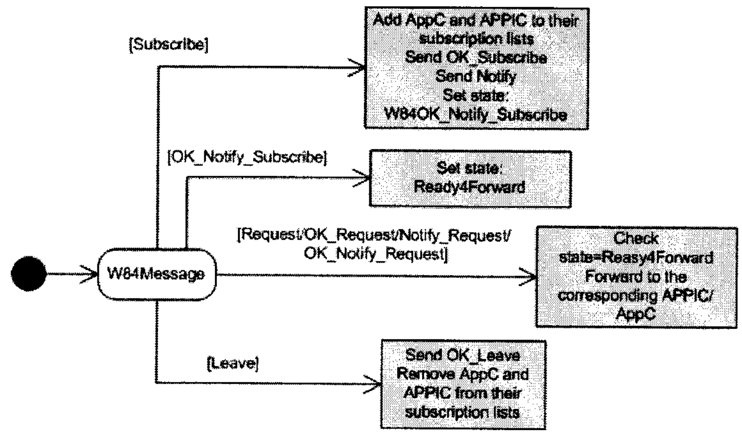

Figure 5.13: APPlication InterConnector state machine

Message type: APPICMSG_SUBSCRIBE	Value: temp	Source @: 1.0.0.3:2000	Destination @: 1.0.0.1:2000	Time stamp: 11.269177478

Figure 5.14: SIP simulated message format

subscription/notification application. In the first scenario the super interconnector acts as a bootstrapping node for other involved nodes and is assumed to be always present in the overlay. Once the super interconnector initiation phase is finished, AppC and APPIC nodes can join the overlay through this node and start sending their subscriptions. In the second scenario the subscribed nodes communicate via the super interconnector, which selects the interconnector with the lowest load to transmit the message. We also implemented the departure of AppC and APPIC nodes and their unsubscription from the super interconnector in any of the two aforementioned scenarios.

5.3.3 Measurements and analysis

In this subsection, we introduce the performance metrics, present and analyse the simulation results.

5.3.3.1 Performance metrics

Four metrics are used to evaluate the performance of our inter-overlay communication architecture.

- Network load: The total number of inter-overlay messages transmitted in the network during the simulation. These messages include the subscription, inter-overlay request, corresponding responses and departure messages.

- Node load: The number of packets sent and received by an overlay component node. It effects the node life time and power consumption.

- Success ratio: The ratio of successful request cycles over the total request cycles. A request sent by AppC is considered to be successful if the final response in received by the corresponding AppC.

- Delay: This metric consists of the subscription delay and request/response latency. Subscription delay is the amount of time from the transmission of a subscription message until receiving a notification response. The request/response delay is the elapsed time between the request creation and transmission from an overlay component to super interconnector, from there to interconnector, and reception of the final response.

5.3.3.2 Performance measurements and analysis

The simulation time is set to 30 minutes and each simulation scenario is executed three times by changing the seed. Presented results are the average of these three measurements. Figure 5.15 illustrates the network load evolution over the simulation time for different number of overlay nodes (total of AppC, APPIC and SAPPIC). As it can be seen from the plot and considering all the exchanged SIP messages, network load increases linearly and the growth of overlay terminals affects it in a reasonable manner.

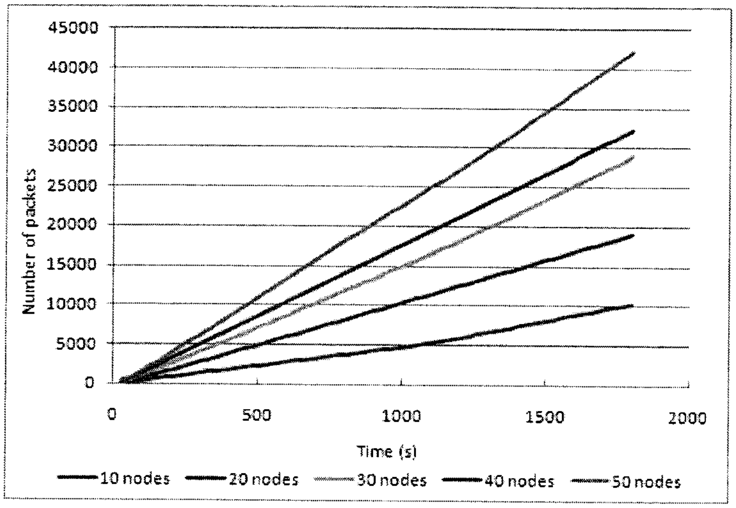

Figure 5.15: Network load evolution over simulation time

Figure 5.16 illustrates the average of maximum node load ratio per type of node as function of total number of nodes. From this figure, it can easily be seen that SAPPIC average load is much higher compared to Appc and APPICs. This expected result is due to presence of only one super interconnector in our simulated system. Although super interconnectors are assumed to be nodes with higher capabilities, this result shows the performance improvements which can be gained by having several super interconnectors in the network as proposed in our architecture. By comparing the collected load for AppC and APPIC nodes and their evolution based on the number of nodes we learn that our load balancing algorithm works perfectly.

The ratio of successful requests over the total requests exchanged in the network as function of time for different number of nodes is illustrated in Figure 5.17. Since a few number of sent messages are lost before the nodes subscription to the super interconnector being completed, this ratio is low in the very first beginning but immediately increases to one. We also observe that since the super interconnector acts as bootstrapping node the total number of nodes in overlays does not really affect the success ratio.

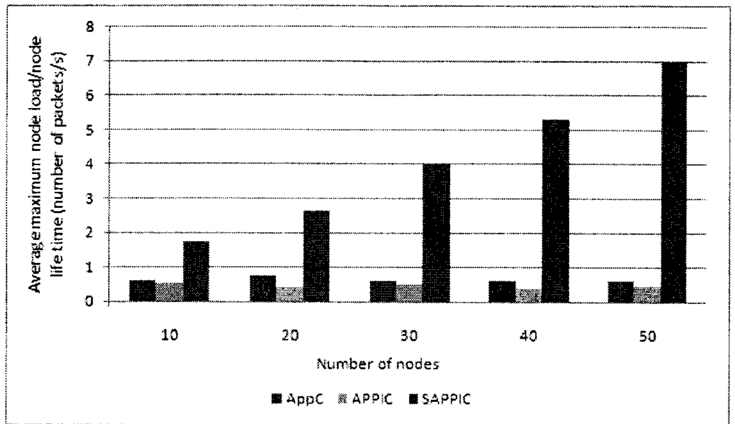

Figure 5.16: Node load for each type of node

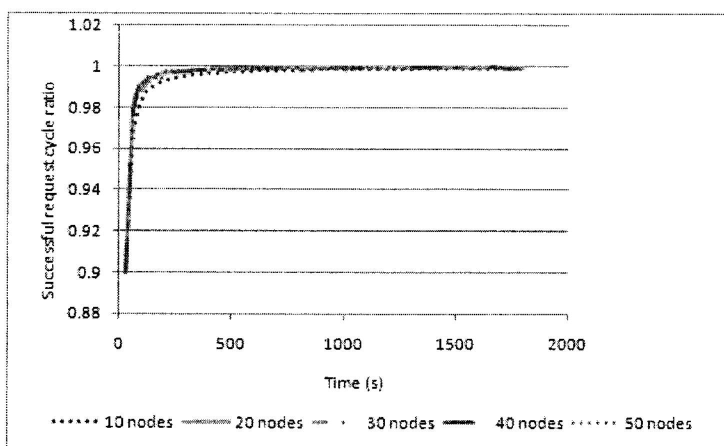

Figure 5.17: Ratio of successful requests over simulation time

The last observation which is related to the subscription and request/response delays is presented in Figure 5.18. These delay are reasonable and increase in an acceptable manner with network growth.

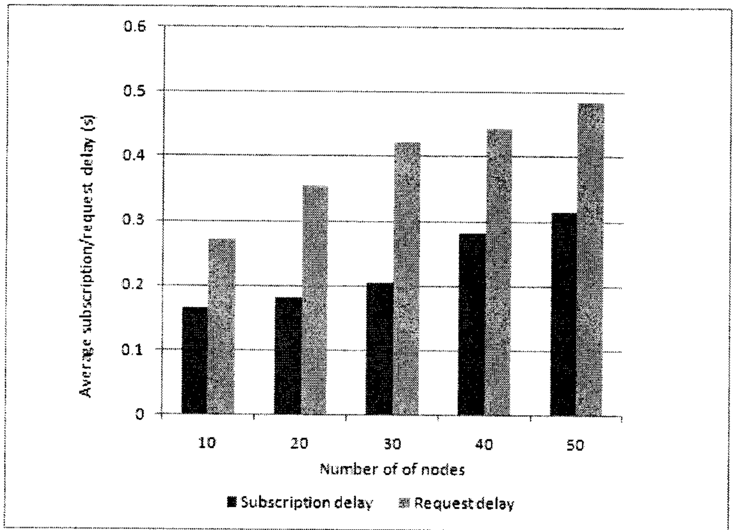

Figure 5.18: Subscription and request/response delays

In view of these results, we can conclude that our proposed architecture is a valid and promising approach for interconnection of gateway and application overlays of our overall architecture.

5.4 Summary

In this chapter, design of interconnection modules and implementation of our prototype as a proof of concept for inter-overlay communication, which is a part of our overall WSN/MANET integrated architecture, is addressed. Implemented protocol and mechanisms are based on SIP and its extensions. JXTA and Open Chord were used as the P2P middleware for implementation of gateway and application overlays and JAIN SIP provided SIP stack is deployed to realize the inter-overlay communication protocol. We ran the prototype in a small-scale 802.11 ad-hoc environment consisting of five nodes and we concluded that it works properly. To validate our proposed architecture in a larger scale scenario, we evaluated the performance of our inter-overlay mechanisms and protocols by simulating a simplified version of the corresponding architecture. The simulation is implemented in Oversim simulator environment reusing the same concepts defined for our prototype. We have used different performance metrics to show the validity and efficiency of our architecture and collected the corresponding results from the simulation. Through the experiments we learn that our proposed architecture is a valid and promising approach for the interconnection of gateway and application overlays of our overall WSN/MANET integration architecture. We also found that the multiplicity of super interconnector nodes is necessary in large scale networks to ensure a proper level of node load and a balanced network communication.

Chapter 6

Conclusions and Future Work

In this chapter, we summarize the contributions of this thesis and discuss the remaining issues which can be considered as the future work.

6.1 Summary of the Contributions

WSNs provide contextual information, such as environmental and physiological data related to a phenomenon, which may be interesting for many end-user applications residing in different networks. Integration of WSNs with these networks, such as Internet, 3G or MANET, enables provision of a wide range of value-added services. This integration is usually realized by deploying gateways which provide interfaces between two domains. In some situations, such as military and emergency response operations, traditional communication infrastructure is not practicably operational. In these situations MANETs are deployed to provide the required communications. Moreover, WSN ambient information allows an enhanced level of application end-user experience in similar scenarios. The integration of WSN and MANET is a challenging task due to mobility and ad-hoc nature of MANETs which render utilization of centralized and static gateways inappropriate and even infeasible in some situations. In this thesis, we have addressed integration of WSNs and MANET by taking in to account specific characteristics of each of these networks.

As a part of the contributions of this thesis, we identified main issues related to MANETs which makes the integration of WSNs more challenging than for infrastructure-based networks. These issues are mainly related to ad-hoc nature, high network dynamicity, devices resource constraint and absence of centralized and pre-established infrastructure in MANETs. We then derived general, application and gateway specific requirements for integration of WSNs and MANET. We also performed a detailed survey on the existing gateway-based solutions for integration of WSN with other networks including MANETs. We divided these solutions into centralized gateways and distributed gateway overlay approaches and evaluated them with respect to the derived requirements. We observed that none of the solutions meets all of our requirements. However, from this evaluation we concluded that the gateway overlay approaches are potentially more appropriate for integration of WSN and MANET. P2P overlays are application layer networks built on top of the physical networks which operations do not require any centralized entity.

As the core contribution of this thesis, we proposed a two-level overlay-based architecture for integration of WSNs and MANET, which satisfies all of our requirements. The main components of the overall architecture are gateway and application overlays which are built on top of the MANET end-user devices. In order to interconnect these two overlays, we needed to refine the overall architecture. In this stage, we derived the requirements for interconnection of gateway and application overlays, examined the state of the art, divided them in to two categories of interconnection node and interconnection overlay approaches and evaluated them. None of the existing works satisfied all of our requirements mostly because there were all based on specific overlay architectures and middlewares. Therefore, we refined our overall architecture by proposing protocols (based on SIP) and mechanisms for the interconnection of gateway and application overlays with respect to our derived requirements. For this purpose in addition to the overlays components, we defined two roles in each of the gateway and application overlays: interconnector and super interconnector. The mechanisms composed of self-organization and information exchange procedures for each type of the defined nodes. We also designed the software architecture of each of the interconnection modules.

As a proof of concept, we implemented a prototype for inter-overlay information exchange of gateway and application overlays based on the designed interconnection protocol. SIP was used as the implementation technology since it meets many of our requirements, such as being light-weight and standard. Therefore, we extended SIP messages to adapt them to our interconnection mechanisms. To test our prototype, we established a real small-scale ad-hoc network on top of which two different overlays were built. We observed that our prototype operates properly and our proposed SIP-based mechanisms are promising for the inter-overlay information exchange. However, to validate our architecture in larger scale, we simulated a simplified version of the interconnection mechanisms using Oversim simulator and collected the experimental results. Based on these results, we conclude that our architecture is a valid and promising approach for interconnecting overlays in large-scale overlay-based environments.

6.2 Future Work

The present work has mainly concentrated on the interconnection of the gateway and application overlays of our proposed overall WSN-MANET integrated system. Based on our evaluations this architecture provides a good potential for the emergency situations where MANETs are deployed, although there are some interesting works remained to be done. Future work includes the implementation of a real comprehensive emergency system as discussed in the beginning to motivate integration of WSN and MANET. For this mean, two aspects should be taken into consideration. The first one is the implementation of gateway specific functionalities in the gateway overlay and integrating it to the sensor network. The second one is the implementation of a real value-added service application, such as conferencing, in the application overlay. The information model for WSN-MANET integration also needs to be standardized and extended to all sort of applications instead of a special case of event-based subscription-notification applications. We faced many constraints for the realization of our prototype and simulation scenarios due to the lack of appropriate development environment for MANET and mobile P2P middleware. To provide a complete performance evaluation, implementation of application and gateway entities of our architecture on the end-user mobile devices, such as PDAs and smart phones can be an interesting project to be tackled later on. Simulation also needs to be extended to the overall architecture including the sensor networks.

Bibliography

[1] S. Sesay, Z. Yang, and J. He, "A survey on mobile ad hoc wireless network," *Information Technology Journal*, no. 3, pp. 168–175, 2004.

[2] L. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40(8), pp. 102–114, August 2002.

[3] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, pp. 72–93, 2005.

[4] S. Hamedi, M. Hormati, R. Glitho, and F. Khendek, "Integrating wireless sensor networks and mobile ad hoc networks for an enhanced end-user experience," *ITU Kaleidoscope Beyond the Internet? - Innovations for future networks and services, Pune, India*, December 2010.

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

[6] I. Baumgart, B. Heep, and S. Krause, "Oversim: A flexible overlay network simulation framework," *IEEE Global Internet Symposium, 2007 In IEEE Global Internet Symposium*, pp. 79–84, 2007.

[7] A. Milenkovic, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Computer Communications Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond*, vol. 29, pp. 2521–2533, 2006.

[8] P. Binkley, "Predicting the potential of wearable technology," *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, pp. 23–27, May 2003.

[9] T. Gao, T. Massey, M. Sarrafzadeh, L. Selavo, and M. Welsh, "Participatory user centered design techniques for a large scale ad-hoc health information system," *HealthNet '07: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, pp. 43–48, 2007.

[10] D. Konstantas, R. Bults, V. Jones, and R. Herzog, "Mobihealth - wireless mobile services and applications for healthcare," *International Conference on Telemedicine - Integration of Health Telematics into Medical Practice*, September 2002.

[11] K. Lorincz, D. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: Challenges and opportunities," *IEEE Pervasive Computing*, vol. 3, pp. 16–23, 2004.

[12] E. Sabbah, A. Majeed, K.-D. Kang, K. Liu, and N. Abu-Ghazaleh, "An application-driven perspective on wireless sensor network security," *Q2SWinet '06: Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks*, pp. 1–8, 2006.

[13] S. Kumar and D. Shepherd, "Sensit: Sensor information technology for the warfighter," *4th Int. Conf. on Information Fusion*, 2001.

[14] S.-H. Chang, M. Merabti, and H. M. Mokhtar, "Coordinate magnetic routing for mobile sinks wireless sensor networks," *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pp. 846–851, 2007.

[15] N. Y. Othman, S. Chebbine, F. Khendek, and R. Glitho, "A web services-based architecture for the interactions between end-user applications and sink-less wireless sensor networks," *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, 2007.

[16] Crossbrow, available at Internet: http://www.xbow.com/.

[17] MIT Cricket Sensors, available at Internet: http://cricket.csail.mit.edu/.

[18] ScatterWeb sensor platform, available at Internet: http://cst.mi.fu.berlin.de/projects/scatterweb/.

[19] IEEE 802.15 Task Group, available at Internet: http://www.ieee802.org/15/pub/TG4.html/.

[20] Zigbee alliance, available at Internet: http://www.zigbee.org/.

[21] Hart communication protocol, available at Internet: http://www.hartcomm.org/.

[22] Isa100 wireless compliant, available at Internet: http://www.isa100wci.org/.

[23] TinyOS - sensor network operating system, available at Internet: http://www.tinyos.net/.

[24] TinyDB - sensor network database system, available at Internet: http://telegraph.cs.berkeley.edu/tinydb/.

[25] T. Luckenbach, P. Gober, S. Arbanowsk, A. Kotsopoulos, and K. Kim, "TinyREST - a protocol for integrating sensor network into the internet," *in Proc. of REALWSN*, June 2005.

[26] NesC, available at Internet: http://nescc.sourceforge.net/.

[27] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," *Parallel and Distributed Processing Symposium, International*, vol. 3, p. 30189a, 2001.

[28] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[29] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Elsevier Ad Hoc Network Journal*, no. 3, pp. 325–349, 2005.

[30] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[31] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal*, no. 1, pp. 2–3, 2003.

[32] R. Ramanathan and J. Redi, "A brief overview of ad hoc networks: Challenges and directions," *IEEE Communications Magazine*, vol. 40, pp. 20–22, 2002.

[33] S. Corson and J. Macker, "Mobile ad hoc networking - IETF RFC 2501," 1999.

[34] MobiHoc, available at http://www.sigmobile.org/mobihoc/.

[35] A. Tate, J. Levine, P. Jarvis, and J. Dalton, "Using AI planning technology for army small unit operations," *Poster Paper in the Proceedings of the Artificial Intelligence Planning and Scheduling Systems Conference*, pp. 379–386, 2000.

[36] L. Barrere, S. Chaumette, and J. Turbert, "A tactical active information sharing system for military MANETs," *In Proceedings of MILCOM 2006*, October 2006.

[37] K. Kanchanasut, A. Tunpan, M. A. Awal, D. K. Das, T. Wongsaardsakul, and Y. Tsuchimoto, "DUMBONET: A multimedia communication system for collaborative emergency response operations in disaster-affected areas," *International Journal of Emergency Management*, no. 4, pp. 670–681, 2007.

[38] A. N. Mian, R. Baldoni, and R. Beraldi, "A survey of service discovery protocols in multihop mobile ad hoc networks," *IEEE Pervasive Computing*, vol. 8, pp. 66–74, 2009.

[39] T. Clausen and P. Jacquet, "Optimized Link State Protocol (OLSR) - IETF RFC 3626," 2003.

[40] R. Ofier, F. Templin, and M. Lewis, "Topology Dissemination Based on Reserve-Path Forwarding (TBRPF) - IETF RFC 3684," 2004.

[41] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV) routing - IETF RFC 3561," 2003.

[42] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing protocol (DSR) for mobile ad hoc networks for IPV4 - IETF RFC 4728," 2007.

[43] "IEEE std 802.11 - wireless LAN medium access control and physical layer specifications," *IEEE Communications Society*, 1999.

[44] B. Chatschik, "An overview of bluetooth wireless technology," *IEEE Communications Magazine*, vol. 39, pp. 86–94, December 2001.

[45] M. Sherman, K. McNeill, K. Conner, P. Khuu, and T. McNevin, "A PMP-friendly MANET networking approach for WiMAX/IEEE 802.16," *Military Communications (MILCOM)*, pp. 1–7, 2006.

[46] Z. Wu, H. Song, S. Jiang, and X. Xu, "Energy-aware grid multipath routing protocol in MANET," *AMS '07: Proceedings of the First Asia International Conference on Modelling and Simulation*, pp. 36–41, 2007.

[47] C.-Y. Wang, C.-J. Wu, G. N. Chen, and R.-H. Hwang, "p-MANET: Efficient power saving protocol for multi-hop mobile ad hoc networks," *International Conference on Information Technology and Applications (ICITA'05)*, vol. 2, pp. 271–276, 2005.

[48] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," *IEEE Wireless Communications*, 2004.

[49] P. Mohapatra, J. Li, and C. Gui, "QoS in mobile ad hoc networks," *IEEE Wireless Communications*, June 2003.

[50] Napster, available at Internet: http://www.napster.com/.

[51] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," *Proceedings of the First International Conference on Peer-to-Peer Computing*, p. 99, 2001.

[52] Bittorrent, available at Internet: http://www.bittorrent.com/.

[53] Kazaa, available at Internet: http://www.kazaa.com/.

[54] Skype, available at Internet: http://www.skype.com/intl/.

[55] A. p. Yu and S. T. Vuong, "MOPAR: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games," *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video*, pp. 99–104, 2005.

[56] H. Le, N. Schiff, J. D. Plessis, and D. B. Hoang, "A pervasive telehealth system for continual and low intrusive monitoring using peer to peer networks,"

International Journal of Computer Applications in Technology, vol. 34, no. 4, pp. 330–334, 2009.

[57] I. Stoica, R. Morris, D. Karger, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Technical Conference*, 2001.

[58] S. Ratnasamy, P. Francis, S. Shenker, R. Karp, and M. Handley, "A scalable content addressable network," *in Proceedings of the ACM SIGCOMM*, pp. 161–172, 2001.

[59] Jxta, available at Internet: https://jxta.dev.java.net/.

[60] R. S. Aygun, Y. Ma, K. Akkaya, G. Cox, and A. Bicak, "A conceptual model for data management and distribution in peer-to-peer systems," *Springer New York*, 2009.

[61] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," *in proceeding of the IPTPS*, pp. 53–65, 2002.

[62] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *IFIP/ACM International Conference on Distributed Systems Platforms*, pp. 329–350, November 2001.

[63] B. Y. Zhao, J. Kubiatowicz, A. D. Joseph, B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," 2001.

[64] X. Shen, H. Yu, J. Buford, and M. Akon, "Handbook of peer-to-peer networking," *Springer*, 2009.

[65] K. Ponmozhi, N. Bharath, and R. Rajesh, "Applying P2P in MANETs for resource sharing," *International conference on control, automation, communication and energy conservation*, 2009.

[66] I. Clarke, O. Sandberg, B. Wiley, O. S, O. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *International workshop on designing privacy enhancing technologies: Design issues in anonymity and unobservability*, pp. 46–66, 2000.

[67] Overnet, available at Internet: http://www.overnet.org/.

[68] R. Winter, T. Zahn, and J. Schiller, "DynaMO: Applying topological locality to the construction of dynamic, mobility-aware overlays," *Freie Universitt Berlin*, 2004.

[69] T. Zahn, R. Winter, and J. Schiller, "Simple, efficient peer-to-peer overlay clustering in mobile, ad-hoc neworks," *In: Proc. of the 12th IEEE Intl Conf. on Networks*, pp. 520–524, 2004.

[70] K. il Jeong, U. H. Yoon, J.-Y. Han, J.-M. Ahn, J.-H. Song, and S.-D. Kim, "RNet: A hierarchical p2p overlay network for improving locality in a mobile environment," *NCM '08: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, vol. 1, pp. 623–630, 2008.

[71] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," *SIGCOMM*, vol. 32, no. 4, pp. 177–190, 2002.

[72] M. Handley and V. Jacobson, "Session Description Protocol (SDP) - IETF RFC 2327," 1998.

[73] M. E. Barachi, A. Kadiwal, R. Glitho, F. Khendek, and R. Dssouli, "The design and implementation of a gateway for IP multimedia subsystem / wireless sensors networks interworking," *IEEE Vehicular Technology Conference Barcelona, Spain*, April 2009.

[74] G. Camarillo and M. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS):Merging the Internet and the Cellular Worlds.* 3 ed., 2004.

[75] S. Krco, D. Cleary, and D. Parker, "P2P mobile sensor networks," *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, vol. 09, 2005.

[76] D. Ye, L. Min, and W. Wang, "Design and implementation of wireless sensor network gateway based on environmental monitoring," *Environmental Science and Information Application Technology, International Conference on*, vol. 2, pp. 289–292, 2009.

[77] M. V. Pulgarin, R. Glitho, and A. Quintero, "An overlay gateway for the integration of IP multimedia subsystem and mobile sink based wireless sensor networks," *IEEE Vehicular Technology Conference (VTC)*, 2010.

[78] A. Liotta and L. Lin, "The operator's response to P2P service demand," *IEEE Communications Magazine*, pp. 76–83, July 2007.

[79] B. Wietrzyk and M. Radenkovic, "Energy efficiency in the mobile ad hoc networking approach to monitoring farm animals," *International Conference on Networking*, 2007.

[80] G. P. Millar, T. A. Ramrekha, and C. Politis, "A peer-to-peer overlay approach for emergency mobile ad hoc network based multimedia communications," *Mobimedia '09: Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, pp. 1–5, 2009.

[81] I. Martinez-Yelmo, A. Bikfalvi, R. Cuevas, C. Guerrero, and J. Garcia, "H-P2PSIP: Interconnection of P2PSIP domains for global multimedia services based on a hierarchical DHT overlay network," *Computer Networks*, pp. 556–568, 2009.

[82] D. Bryan, P. Matthews, E. Shim, and D. Willis, "Concepts and terminology for peer to peer SIP," July 2008.

[83] M. Tsietsi, A. Terzoli, and G. Wells, "Prototyping a P2P SIP user agent with support for multiple overlays," *Pervasive Computing and Communications, IEEE International Conference on*, vol. 0, pp. 474–479, 2008.

[84] OpenChord DHT, available at Internet: http://open-chord.sourceforge.net/.

[85] Bamboo, A Robust Open Source DHT, available at Internet: http://www.bamboo-dht.org/.

[86] B. Maniymaran, M. Bertier, and A.-M. Kermarrec, "Build one, get one free: Leveraging the coexistence of multiple P2P overlay networks," *Distributed Computing Systems, International Conference*, vol. 0, p. 33, 2007.

[87] S. Voulgaris and M. van Steen, "Epidemic-style management of semantic overlays for content-based searching," *Euro-Par 2005 Parallel Processing*, pp. 1143–1152, August 2005.

[88] "SIP INFO method - IETF RFC 2976," *http://www.ietf.org/rfc/rfc2976.txt*, October 2000.

[89] "Session Initiation Protocol (SIP) specific event notification - IETF RFC 3265," *http://www.ietf.org/rfc/rfc3265.txt*, June 2002.

[90] C. Fu, R. H. Glitho, and F. Khendek, "Signaling for multimedia conferencing in stand-alone mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 7, pp. 991–1005, 2009.

[91] "Jxta java standard edition v2.5: Programmers guide," *Sun Microsystems, Inc*, September 2007.

[92] A. Singh and M. Haahr, "A survey of P2P middlewares," *Distributed Systems Group, Department of Computer Science. Trinity College, Dublin.*

[93] S. Kaffille and K. Loesing, *Open Chord version 1.0.4 Users Manual.* Lehrstuhl fur Praktische Informatik Otto-Friedrich-Universitat Bamberg, October 2007.

[94] JAIN SIP, available at Internet: https://jain-sip.dev.java.net/.

[95] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P systems scalable," *In Proc. ACM SIGCOMM*, 2003.