

**Multiparty/Multimedia Conferencing in Mobile Ad-Hoc  
Networks for Improving Communications between  
Firefighters**

Moayad Aloqaily

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 (Computer Engineering)  
at  
Concordia University  
Montreal, Quebec, Canada

July 2012

© Moayad Aloqaily, 2012

**CONCORDIA UNIVERSITY**

**Department of Electrical and Computer Engineering**

This is to certify that the thesis prepared

By: Moayad Aloqaily

Entitled: **Multiparty/Multimedia Conferencing in Mobile Ad-Hoc Networks for Improving Communications between Firefighters**

and submitted in partial fulfillment of the requirement for the degree of

**Master of Applied Science (Computer Engineering)**

complied with the regulations of the University and meets with the accepted standards with respect to originality and quality.

Signed by the final examining committee:

\_\_\_\_\_ Chair  
Dr. B. Fung

\_\_\_\_\_ External Examiner  
Dr. J. Rilling, CSE

\_\_\_\_\_ Examiner  
Dr. F. Khendek

\_\_\_\_\_ Co-supervisor  
Dr. R. Glitho

\_\_\_\_\_ Co-supervisor  
Dr. A. Hammad

Approved by: \_\_\_\_\_  
Dr. W. E. Lynch, Chair  
Department of Electrical and Computer Engineering

## **ABSTRACT**

### **Multiparty/Multimedia Conferencing in Mobile Ad-Hoc Networks for Improving Communications between Firefighters**

Moayad Aloqaily, M.A.Sc.

Concordia University, 2012

In current practice, firefighters' communications systems are verbal, using a simplex Radio Frequency (RF) system (walkie-talkie). They use a push-to-talk mechanism in which only one person can talk at any time and all other firefighters will hear the messages. They use special codes (e.g. 1008, 1009, etc.) to express their current situation. Firefighters of the same team need to be in visual contact with each other at all times. This RF system does not support other functionalities (e.g. video communications, conference calls). In addition, because communication between firefighters is a flat structure, private communications is not possible.

Mobile Ad-Hoc Networks (MANETs) are infrastructure-less and self-organized wireless networks of mobile devices, which are not based on any centralized control. MANETs are suitable for the hosting of a wide range of applications in emergency situations, such as natural or human-induced disasters, and military and commercial settings. Multimedia conferencing is an important category of application that can be deployed in MANETs. This includes well-known sets of applications, such as audio/video conferencing, data communications, and multiplayer games.

Conferencing can be defined as the conversational exchange of data content between several parties. Conferencing requires, at the very least, the opening of two sessions: a call signaling session, and a media handling session. Call signaling is used to set up, modify, and terminate the conference. Media handling is used to cover the transportation of the media, and to control/manage the media mixers and media connections.

So far, very little attention has been devoted to the firefighters' communication system. In the present work, we focus on building a new communication system for firefighters using multimedia conferencing/sub-conferencing in MANETs. The background information for the firefighters' current communications system and MANETs, along with the multimedia conferencing, is provided. The limitations of this system are determined, and the requirements are derived to determine the functionalities of a better communication system that will overcome current limitations. We have proposed a cluster-based signaling architecture that meets our requirements. We have also identified a state-of-the-art media handling and mixing system that meets most of our requirements, and have adapted it to interwork with our signaling system. We have implemented the proposed architecture using SIP signaling protocol. Performance measurements have been performed on the prototype. Through experiments, we have found that the new multimedia communication system is a very promising approach to solve the current firefighters' communication problems.

## **ACKNOWLEDGEMENT**

My most sincere appreciation and gratitude go to my supervisors, Professor Roch Glitho and Professor Amin Hammad, for their kind help, intellectual and personal support, encouragement and patience in the success of my work as a researcher.

I would like to thank Doctor Fatna Belqasmi for spending her precious time in reading this thesis and for her valuable comments. I would also like to extend my thanks to all members of my examining committee for reviewing my thesis.

In addition, I would like to thank Mr. Darren Dumoulin from the emergency management department at Concordia University and all other firefighters from Sain-Mathieu Fire Station (Montreal, QC) for their contributions.

I have also known many wonderful friends in Montreal, and although I am not able to mention each of their names, I would like to extend my thanks to all of them.

## **DEDICATION**

To my parents, who made all of this possible, for their endless encouragement and support, my success is yours. My high regards are addressed to my parents-in-law as well.

I would like to express my gratitude to my wife Safa for her support and all the sacrifices she has made.

Finally, to my beloved son, Rawad, you are the joy of my life.

# Table of Contents

|   |            |
|---|------------|
| <b>LIST OF FIGURES .....</b>  | <b>XI</b>  |
| <b>LIST OF TABLES .....</b>   | <b>XIV</b> |
| <b>LIST OF ABBREVIATIONS .....</b>  | <b>XV</b>  |
| <b>CHAPTER 1        NTRODUCTION .....</b>                                 | <b>1</b>   |
| 1.1    MOTIVATION .....   | 1          |
| 1.2    PROBLEM STATEMENT .....  | 3          |
| 1.3    RESEARCH OBJECTIVES .....  | 4          |
| 1.4    THESIS ORGANIZATION.....   | 4          |
| <b>CHAPTER 2        BACKGROUND .....</b>                                  | <b>6</b>   |
| 2.1    INTRODUCTION .....   | 6          |
| 2.2    EMERGENCY RESPONSE SYSTEM .....                                    | 6          |
| 2.2.1 <i>Management functions of first responders.....</i>                | <i>7</i>   |
| 2.2.1.1.    Command Post (CP) or incident commander.....                  | 8          |
| 2.2.1.2.    Operations section.....                                       | 10         |
| 2.2.1.3.    Planning, logistics, and finance/administration sections..... | 15         |
| 2.3    MOBILE AD-HOC NETWORKS (MANETs).....                               | 17         |
| 2.3.1 <i>Evolution of MANETs.....</i>                                     | <i>17</i>  |
| 2.3.2 <i>Classification of MANETs .....</i>                               | <i>18</i>  |
| 2.3.3 <i>Clustering in MANETs.....</i>                                    | <i>18</i>  |
| 2.4    MULTIMEDIA CONFERENCING .....                                      | 19         |
| 2.4.1 <i>Multimedia conferencing basics .....</i>                         | <i>20</i>  |
| 2.4.2 <i>Classification of conferencing .....</i>                         | <i>20</i>  |

|                  |   |           |
|------------------|---|-----------|
| 2.4.3            | <i>Conference models</i> .....                                    | 21        |
| 2.5              | SUMMARY .....   | 23        |
| <b>CHAPTER 3</b> | <b>REQUIREMENTS AND EVALUATION OF RELATED</b>                     |           |
| <b>WORK</b>      | <b>.....</b>  | <b>24</b> |
| 3.1              | INTRODUCTION .....  | 24        |
| 3.2              | REQUIREMENTS FOR FF COMMUNICATION SYSTEM .....                    | 24        |
| 3.3              | RELATED WORK .....  | 27        |
| 3.3.1            | <i>Signaling protocols for multimedia conferencing</i> .....      | 28        |
| 3.3.1.1.         | Signaling protocol from ITU-T .....                               | 28        |
| 3.3.1.2.         | Signaling protocols from IETF .....                               | 29        |
| 3.3.1.3.         | Signaling solutions from outside the standardization bodies ..... | 31        |
| 3.3.2            | <i>Media handling protocols for multimedia conferencing</i> ..... | 33        |
| 3.3.2.1.         | Centralized mixing architecture .....                             | 33        |
| 3.3.2.2.         | Endpoint mixing architecture .....                                | 34        |
| 3.3.2.3.         | Hierarchical mixing architecture .....                            | 34        |
| 3.3.2.4.         | Partial mixing architecture .....                                 | 35        |
| 3.3.2.5.         | Distributed mixing architecture (DMA) .....                       | 36        |
| 3.3.3            | <i>Emergency response systems for first responders</i> .....      | 37        |
| 3.4              | SUMMARY .....   | 38        |
| <b>CHAPTER 4</b> | <b>PROPOSED APPROACH .....</b>                                    | <b>39</b> |
| 4.1              | INTRODUCTION .....  | 39        |
| 4.2              | OVERALL ARCHITECTURE .....  | 39        |
| 4.2.1            | <i>FFs layered architecture</i> .....                             | 39        |
| 4.2.2            | <i>Overall conferencing architecture</i> .....                    | 42        |
| 4.3              | SIGNALING ARCHITECTURE .....                                      | 44        |
| 4.3.1            | <i>A cluster-based signaling architecture</i> .....               | 44        |



|                  |  |           |
|------------------|--|-----------|
| 4.3.2            | <i>Operational signaling procedure</i> .....             | 45        |
| 4.4              | MEDIA HANDLING ARCHITECTURE .....                        | 48        |
| 4.4.1            | <i>Adapting DMA to our architecture</i> .....            | 48        |
| 4.5              | SUMMARY .....  | 51        |
| <b>CHAPTER 5</b> | <b>PROTOTYPE AND EVALUATION</b> .....                    | <b>52</b> |
| 5.1              | INTRODUCTION .....                                       | 52        |
| 5.2              | IMPLEMENTATION OF CONFERENCING/SUB-CONFERENCING .....    | 52        |
| 5.2.1            | <i>Implementation technology</i> .....                   | 52        |
| 5.2.2            | <i>Scenarios for conferencing/sub-conferencing</i> ..... | 53        |
| 5.3              | SIGNALING/MEDIA HANDLING PROTOTYPE .....                 | 64        |
| 5.3.1            | <i>Programming language and tools</i> .....              | 64        |
| 5.3.2            | <i>Software architecture</i> .....                       | 65        |
| 5.3.2.1.         | <i>Mixers' design</i> .....                              | 67        |
| 5.3.3            | <i>User interfaces' design</i> .....                     | 70        |
| 5.4              | PERFORMANCE MEASUREMENTS .....                           | 73        |
| 5.4.1            | <i>Environment settings</i> .....                        | 73        |
| 5.4.2            | <i>Metrics</i> .....                                     | 74        |
| 5.4.2.1.         | <i>The network load</i> .....                            | 74        |
| 5.4.2.2.         | <i>The signaling delays</i> .....                        | 79        |
| 5.5              | SUMMARY .....  | 84        |
| <b>CHAPTER 6</b> | <b>CONCLUSIONS AND FUTURE WORK</b> .....                 | <b>85</b> |
| 6.1              | SUMMARY OF CONTRIBUTIONS .....                           | 85        |
| 6.2              | FUTURE WORK .....  | 88        |
| <b>REFERNCES</b> | .....  | <b>90</b> |

|                    |   |           |
|--------------------|---|-----------|
| <b>APPENDIX A.</b> | <b>XML FILES FOR FF NODES .....</b>             | <b>97</b> |
| <b>APPENDIX B.</b> | <b>EXAMPLE OF A REFER MESSAGE WITH MULTIPLE</b> |           |
| <b>TARGETS</b>     | <b>.....</b>                                    | <b>99</b> |

## List of Figures

|  |    |
|--|----|
| Figure 1.1: Structure of communication between the FFs .....   | 2  |
| Figure 2.1: Major incident functions [2].....  | 8  |
| Figure 2.2: Composition of the first responders, including the FFs .....   | 11 |
| Figure 2.3: Strike teams .....   | 13 |
| Figure 2.4: A picture of Walkie-Talkie used by FFs .....   | 15 |
| Figure 2.5: Conceptual view of the communication hierarchy between the FFs.....  | 17 |
| Figure 2.6: Structure of clustering .....  | 23 |
| Figure 3.1: H.323 network structure and components [25][26].....   | 30 |
| Figure 3.2: Two-level media handling architecture [51][52].....  | 38 |
| Figure 4.1: Overall architecture .....   | 42 |
| Figure 4.2: A signaling architecture for conferences/sub-conferences in MANETs.....  | 43 |
| Figure 4.3: Steps of the signaling operations to create the FF communication system....  | 45 |
| Figure 4.4: Two-level media handling architecture showing the mixers for the leaders/CP<br>and the inactive peers for the members..... | 48 |
| Figure 5.1: Two leaders joining the main conference.....   | 53 |
| Figure 5.2: A third leader joins the main conference.....  | 55 |

|   |    |
|---|----|
| Figure 5.3: First choice of members joining the conference and the sub-conferences..... | 57 |
| Figure 5.4: Second choice of members joining the conference and the sub-conferences     | 57 |
| Figure 5.5: Modifying the second choice .....   | 60 |
| Figure 5.6: Overall SIP implementation architecture .....                               | 61 |
| Figure 5.7: Overall SIP implementation showing all messages .....                       | 62 |
| Figure 5.8: The CP software architecture .....  | 64 |
| Figure 5.9: The Leader software architecture .....                                      | 65 |
| Figure 5.10: The member software architecture .....                                     | 66 |
| Figure 5.11: The CP mixer core functional architecture.....                             | 67 |
| Figure 5.12: The Leader mixer core functional architecture .....                        | 68 |
| Figure 5.13: FFs User Interface: (a) CP, (b) Leaders, (c) Members .....                 | 70 |
| Figure 5.14: Prototype settings .....   | 73 |
| Figure 5.15: Total number of bytes sent/received by the CP.....                         | 74 |
| Figure 5.16: Total number of bytes sent/received by L1 .....                            | 75 |
| Figure 5.17: Total number of bytes sent/received by L2 .....                            | 75 |
| Figure 5.18: Total number of bytes sent/received by the members M1, M2, and M3 .....    | 76 |
| Figure 5.19: Total number of bytes sent/received by all FFs .....                       | 77 |

Figure 5.20: The delay in: (1) registration, (2) joining ..... 78

Figure 5.21: Comparisons the total delay between full mesh and clustering ..... 81

## List of Tables

|  |    |
|--|----|
| Table 3.1: Requirements of FF communications system.....         | 27 |
| Table 5.1: Comparison between the three choices.....             | 61 |
| Table 5.2: Registration and joining delays for nine leaders..... | 80 |
| Table 5.3: Nine-leaders with zero members - Full mesh mode ..... | 80 |
| Table 5.4: One-leader with eight-members .....                   | 80 |
| Table 5.5: Two-leaders with seven-members .....                  | 82 |
| Table 5.6: Three-leaders with six-members.....                   | 82 |

## List of Abbreviations

| <b>Abbreviation</b> | <b>Description</b>                             |
|---------------------|--|
| API                 | Application Programming Interface              |
| CCP                 | Conference Control Protocol                    |
| DMA                 | Distributed Mixing Architecture                |
| FF                  | Firefighter                                    |
| GUI                 | Graphical User Interface                       |
| IAP                 | Incident Action Plans                          |
| iBR                 | Intelligent Building Response                  |
| IETF                | Internet Engineering Task Force                |
| IP                  | Internet Protocol                              |
| ITU                 | International Telecommunication Union          |
| JAIN SIP            | Java APIs for Integrated Networks SIP          |
| JMF                 | Java Media Framework                           |
| MANET               | Mobile Ad hoc Network                          |
| MCU                 | Multipoint Control Unit                        |
| NIST                | National Institute of Standards and Technology |
| RTCP                | RTP Control Protocol                           |
| RTP                 | Real-time Transport Protocol                   |

|      |                                    |
|------|------------------------------------|
| SCCP | Simple Conference Control Protocol |
| SDP  | Session Description Protocol       |
| SIP  | Session Initiation Protocol        |
| SM   | Super Member                       |
| SSM  | Super-Super Member                 |
| TCP  | Transmission Control Protocol      |
| UA   | User Agent                         |
| UDP  | User Datagram Protocol             |
| UI   | User Interface                     |
| XML  | eXtensible Markup Language         |



# CHAPTER 1 INTRODUCTION

The main goal of this thesis is to provide firefighters (FFs) with a new communication system in order to replace the current radio communication system. The new system will be based on mobile ad-hoc networks (MANETs). In this chapter, we first present the motivation of our research. Then, we describe the problem addressed by this thesis, and the thesis objectives. The last section presents the organization of the dissertation.

## 1.1 Motivation

As of today's current communication system, all FFs use the simplex Radio Frequency (RF) system (walkie-talkie) to communicate with each other. In order to use the walkie-talkie, the FF presses the button of the device to talk, and releases the button and waits to hear the play back.

All FFs communications are verbal. The FF team members and leaders use the walkie-talkie to establish communication with each other during an incident. These walkie-talkies support audio only; it is not possible to extend this RF system to other functionalities (e.g. video communications, conference calls).

They use a push-to-talk mechanism in which, at any time, only one FF (Command Post (CP), leader, or member) can talk. It is not possible for two FFs to talk at the same time (in the normal situations). FFs use special codes (e.g. 1008, 1009, etc.) to express their current situation [1]. For high emergency situations, when there is a FF who needs to talk and make all other FFs stop talking and listen to him, he needs to repeat a certain code

three times (i.e. 1053) to guarantee that all FFs will hear him. This solution will produce problems and complexities, for instance, if two FFs start using the high emergency code at the same time, there is no mechanism in the walkie-talkie to let them know that they are talking at the same time. The FF can either listen or talk at a time.

Figure 1.1 shows other two issues with the communication between the FFs. First, the current FF communication system has a flat structure of communication, where we can see each FF is connected to all other FFs. Second, there is no private communication between the FFs; every FF at the incident hears every message from every team. Those teams are distributed at different areas of the incident. In general, FFs do not need to hear the messages from different teams. On the other hand, these messages could be disturbing and could make the other teams unfocused.

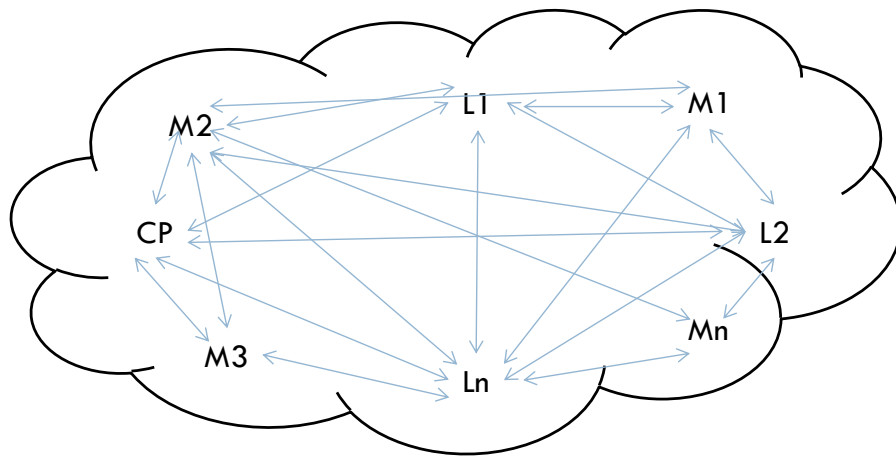


Figure 1.1: Structure of communication between the FFs

FFs of the same team need to be in visual contact with each other all the time, due to the limitations of their current communication system (support voice only, as explained previously).

## 1.2 Problem statement

In this thesis, we will tackle the critical issues discussed previously. We want to provide FFs with a new communication system in order to replace the existing one. The new communication system should overcome the current system limitations by:

- Providing a new structure for the FF communication system

The new communication system will have a new communication structure (i.e. hierarchical) instead of the current flat structure. The new structure will allow for more than two FFs to talk at the same time, along with private communication between team members and team leaders.

- Providing FFs with new functions on the fire ground

The new communication system should be able to support the FFs with new functionalities, along with audio communications, such as, video and floor control.

In this research, we consider Mobile Ad-Hoc Networks (MANETs) for the following reasons:

- MANETs are particularly useful in emergency situations such as natural disasters.
- These networks work without any pre-existing infrastructure. They can be set up at any place and time.
- The cost of building a MANET is low. The network only involves end-users' devices. Moreover, heterogeneous devices can be involved in ad-hoc networks. This will encourage a wide range of users to participate.

### **1.3 Research objectives**

The objectives of this research are:

- To derive the requirements through the study of the FFs communication system.
- To review related works.
- To propose an architecture for a FF communication system to satisfy the derived requirements.
- To validate the proposed architecture.

### **1.4 Thesis organization**

The rest of the thesis is organized as follows:

*Chapter 2: Background:* This chapter will present the background knowledge for both MANETs and the emergency response system focusing on FFs. We will provide an overview of MANETs including the technologies and the classifications. Then, we conclude this chapter by introducing multimedia conferencing.

*Chapter 3: Requirements and Evaluation of the Related Work:* This chapter presents the derived requirements for the FFs communication system. The review of related work is done in light of these requirements. The related work includes existing solutions for multimedia conferencing in MANETs.

*Chapter 4: Proposed Approach:* This chapter describes the details of the proposed approach, which includes the overall architecture, a cluster-based signaling architecture, the operational signaling procedure, the media handling architecture and inter-working

with signaling, and the operational scenario.

*Chapter 5: Prototype and Evaluation:* This chapter describes the development of a prototype system, software architectures, and shows the applicability of the proposed approach.

*Chapter 6: Conclusions and Future Work:* This chapter summarizes the present research work, highlights its contributions, and suggests recommendations for future research.

## **CHAPTER 2 BACKGROUND**

### **2.1 Introduction**

In this chapter, we present the background information for this research. The purpose of this review is to understand the basic concepts and technologies necessary for this research. Three groups of information are introduced. In the first group, we review the emergency response system, which will basically cover the composition of first responders including FFs, the management functions of first responders, and the operations of the FF communication system. In the second group, we will introduce the related information of MANETs, covering the definitions, classifications, standards, and clustering technologies. The last group gives an introduction to multimedia conferencing and its technical components.

### **2.2 Emergency response system**

**An emergency** is any unexpected event that can cause serious damage or significant injuries to people, that can shut down the system of a facility, disrupt operations, cause physical or environmental damage, or threaten the facility's financial standing or public image [2] [4]. Clearly, many events can be "emergencies" including fires, hurricanes, tornados, and earthquakes.

**Emergency management** is the process of preparing for mitigating, responding to and recovering from an incident. It is divided into four phases, namely mitigation, preparedness, response, and recovery. Mitigation tries to reduce loss of life's and properties by lessening the impact of disasters. The National Incident Management

System (NIMS) [3] defines preparedness as “a continuous cycle of planning, organizing, training, equipping, exercising, evaluating, and taking corrective action in an effort to ensure effective coordination during incident response”. Response is the process of responding to an incident by sending the first responders (e.g. FFs) to the incident location in order to provide important services, such as evacuating the victims, providing first aid services and collecting evidences. The aim of the recovery phase is to restore the affected area, reconstruct the buildings, and provide food and services to the people affected by the incident.

In the response phase, there are different types of first responders (e.g. FFs, police, and medics) who respond to the incident. The FFs are vital and important during an incident. The process of how FFs communicate and organize their teams goes through many steps. In the following section, we will review how the FFs respond to an incident and how they communicate directly among each other and with their leaders during the incident.

### **2.2.1 Management functions of first responders**

Regardless of the magnitude of an incident, it always requires that certain management functions be created and applied to the incident [1] [2] [5] [6]. There are five major management functions that are the foundation upon which the response organization develops. These functions apply for a routine emergency (i.e. fire call), or a major disaster (i.e. hurricane or terrorist attack). Below is a brief description of the major incident functions [2]. See Figure 2.1.

(1) **Incident Command Post (CP)**: This function sets the incident objectives, goals,

strategies, and priorities and has the overall responsibility for the incident.

- (2) **Operations:** This function performs certain operations to reach the incident objectives, establishes and directs all operational resources.
- (3) **Planning:** This function supports the incident action planning processes by tracking resources, collecting/analyzing information, and maintaining documentation.
- (4) **Logistics:** This function provides resources and needed services to support the achievement of the incident objectives.
- (5) **Finance and Administration:** This function monitors costs related to the incident, provides accounting, procurement, time recording, and cost analyses.

#### 2.2.1.1. Command Post (CP) or incident commander

The CP (incident commander) has the overall responsibility of managing the incident by establishing objectives, planning strategies, and implementing tactics. The CP is responsible for creating the sections that are needed. If a section is not staffed, the CP will personally manage the functions of the section (see Figure 2.1).

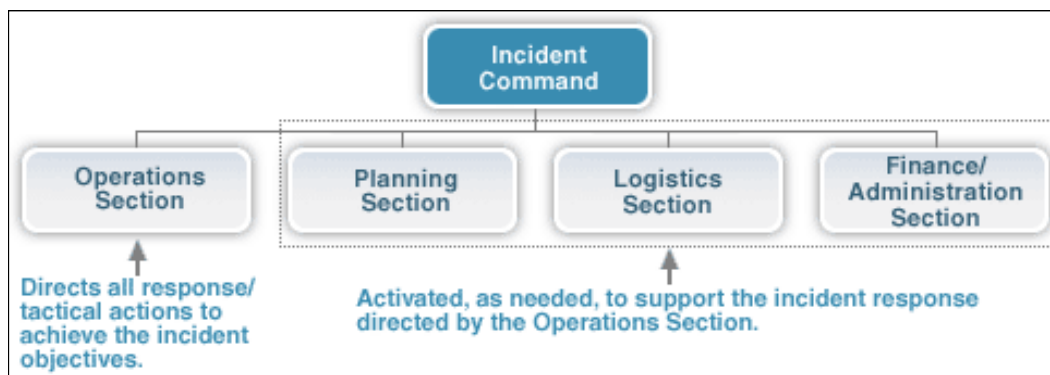


Figure 2.1: Major incident functions [2]



In addition to having overall responsibility for managing the entire incident, the CP is responsible for:

- Ensuring overall incident safety.
- Providing information services to internal and external stakeholders (owners).
- Establishing and maintaining liaisons with other agencies participating in the incident, e.g., media.

#### **(a) Position of the CP**

Typically, the CP will be positioned as close to the incident and as safely as possible in order to facilitate the communication process [7] [8]. Structures, such as high-rise buildings, tunnels, and sub-basements, may disturb the effectiveness of portable radios and communications. In these situations, some fire departments may position the CP inside the structure to compensate for the portable radios' weak RF output. For example, with fires located on the upper levels of a high-rise structure, the CP may be established several floors below the fire (as opposed at street level) to help improve communication.

#### **(b) Professional CP and temporary CP**

The process starts by assigning a CP to the whole operation. He is a professional person directly responsible for organizing communication and teams around and on incident location [7]. Once the CP arrives to the incident location, he starts collecting important information about the incident. The first task when he arrives to the incident is to find out if there is a previous CP on duty (i.e. temporary CP). The temporary CP can be any responsible person that arrived to the incident location before the professional CP, for

example, a police officer, building owner, or a private guard.

#### **2.2.1.2. Operations section**

The CP first establishes the operations section. The remaining sections are established as needed to support the operations section (see Figure 2.1).

After the professional CP arrives at the incident location and directly gathers the information from the people who arrived before him, the problem must be identified and assessed, a plan must then be developed and implemented in order to deal with the situation, and the necessary resources should be ready.

In the operations section, the CP is responsible for forming the FF teams, and the teams should be established based on the type of incident. Once the FFs arrive at the scene, they should directly meet the CP in order to receive instruction (e.g. forming the teams). The CP will decide, based on the magnitude of the incident, what kind of teams need to be formed.

##### **(a) Composition and roles of the FF teams**

Figure 2.2 shows the composition of the first responders. Assuming there are  $n$  FF (i.e.  $F_1, F_2, \dots, F_n$ ), they can be organized in several teams (e.g.  $T_1=[F_1, F_5]$ ,  $T_2=[F_6, F_{10}]$ , and  $T_3=[F_{11}, F_{15}]$ ). Notice that we are using  $F$  to represent the role of a FF as a member of a team [1] [2]. In each team, one of the members will also be the leader of the team (e.g.  $F_1$  is the leader of  $T_1$ ). The leader of  $T_i$  is referred to as  $L_i$  ( $L_1$  and  $F_1$  are the same person in this example). The FF teams are composed of 4 to 5 members. The same situation applies for the police teams, as shown in Figure 2.2.

In the case where we have several types of first responder groups (e.g. police, FFs, etc.), it is common to assign one of the leaders as the head of all teams belonging to that type. Therefore, there would be one head for all FF teams ( $H^F$ ) and another head for all police teams  $H^P$ , etc.

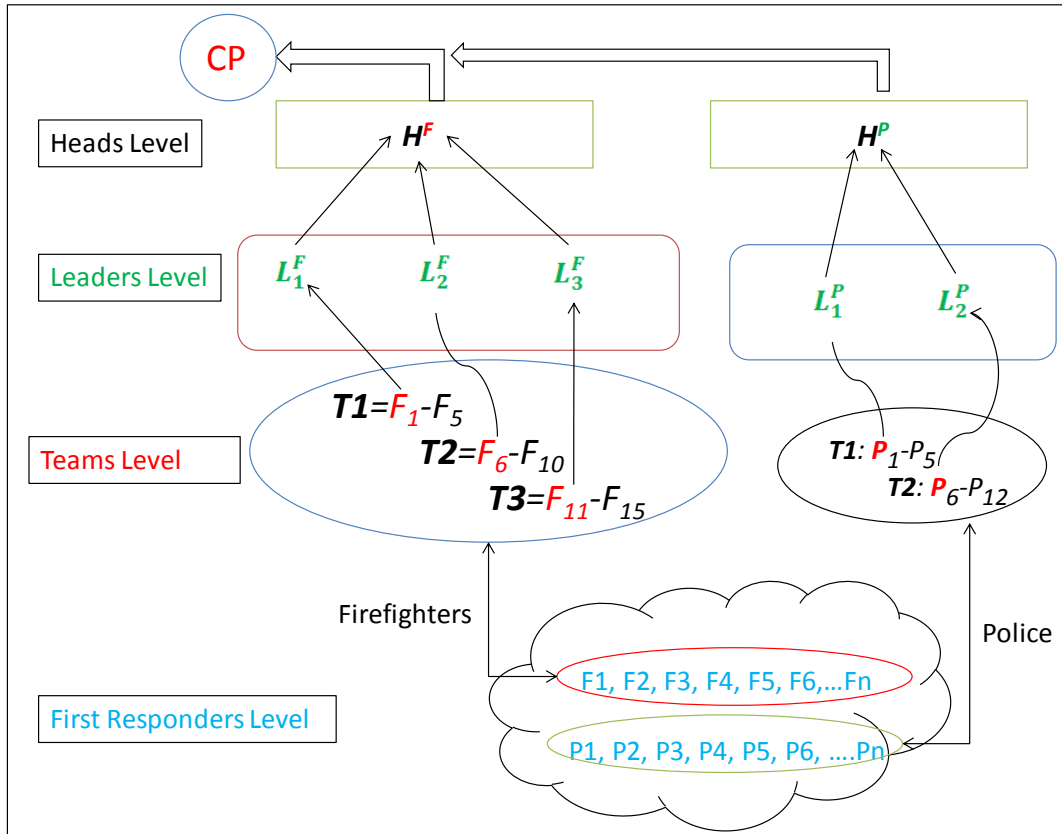


Figure 2.2: Composition of the first responders, including the FFs

At the top of the organizational hierarchy (Figure 2.2), one CP will be in charge of coordinating the communication of the incident site. The CP is always outside the operations and never becomes a FF and CP together at the same time.

The person who is directly responsible for organizing the entire internal and external

communications is the CP. He collects information from the heads of the responders, and supports the responders with the materials and the resources required. In addition, he is responsible for reporting serious problems to external authorities. For example, if the incident happens in a public place, the CP will report to the head of the fire department of the city.

The CP is the focal point of all communication between all three organizational levels: heads, leaders, and members (see Figure 2.2). The members of the same team communicate with each other and with their leaders directly. The leaders of each team should report any important information and decision making to the head. The role of the head is to report the state of the incident and all related information to the CP.

In the upcoming sections we will explain the different types of FF teams, and the key factors in deciding which teams should be formed, as well as the responsible person for forming these teams.

### **(b) Type of FF emergency teams**

We have two types of emergency teams [1] [2]:

**Strike Teams:** include all similar communication resources with common communications operating under the direct supervision of a leader, as is the case of those working in the same department, e.g., only one agency working on the incident, (e.g. FFs only).

**Task Forces:** are combinations of mixed communication resources with common communications operating under the direct supervision of a leader, when more than one

agency is working together (e.g. police, FF, etc.).

In the case of the strike team, the CP will assign the leaders and the members of each team only. The communications here are directed from the members to the leader, and from the leader to the CP (see Figure 2.3).

Figure 2.3 shows the case where only one first responder (type of resources are FFs) has responded to the incident, Therefore, only two of the three levels of communications are needed. The direction of the communication will be from members to leaders, then from leaders to CP. There is no need for the third level of communication (i.e. Head level); it is not used and it is not necessary in this type of emergencies.

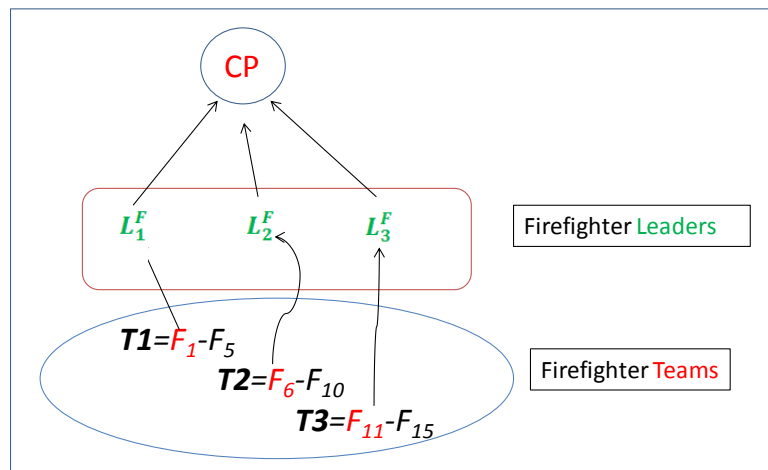


Figure 2.3: Strike teams

However, in the case of *Task Force* teams, the CP will require three levels of communication; the heads, the leaders, and the members. This hierarchy will help the CP organize the process of communication. The members will report to the leaders, the leaders will report to the heads, and the heads will report to the CP (see Figure 2.2).

The major difference between the previous two teams is that in the Strike Team, all FFs will use the same walkie-talkies (same resources of communication), which means that they will communicate under the same frequency. In the Task Forces, the FFs will work on a different frequency than those of the police, or other parties.

Figure 2.2 shows that the task force team reduces the number of persons that will communicate with the CP. The Task Forces team makes sure to save time and resources during the incident.

After the CP forms the teams, he needs to setup the first operation plan. The FFs will distribute at the incident locations, and will start doing their jobs using this plan. In the case of an unexpected situation, or if any member needs to report something to the leader, the communication session will start.

### **(c) The FFs current communications system**

Currently, walkie-talkies, which are based on radio system frequency, are the only way for the FFs to communicate with each other [7] [9]. Walkie-talkies are hand-held devices that allow two or more people to communicate using radio waves. Most walkie-talkies let users perform one function at a time (either listening or talking). They are a type of device known as a “two-way radio.” Two way radios transmit and receive radio communication signals.

There are six main parts to a walkie-talkie (see Figure 2.4). These are the transmitter, the receiver, the crystal, the speaker, the microphone, and the power source. The transmitter sends the audio, after it has been made into a radio signal, to the other walkie-talkies the

user is communicating with. The receiver picks up the radio signal that the transmitter from the other walkie-talkies sent. The crystal changes the frequency (also known as channel) that the walkie-talkies are on. Walkie-talkies must be on the same frequency to communicate. The speaker takes the signal that is picked up by the receiver and amplifies it so that it can be heard. The microphone is the part that the user speaks into and it converts the voice into electric signals. It is the exact opposite of the speaker, which converts electric signals into voices. The power source is a type of battery, which are sometimes rechargeable. The walkie-talkie device is attached to the FFs chest at all times, and connects to an external microphone/loudspeaker in order to eliminate the need to hold the device.



Figure 2.4: A picture of Walkie-Talkie used by FFs

### 2.2.1.3. Planning, logistics, and finance/administration sections

The CP will determine if there is a need for a planning section and if so, will designate a

planning section chief. If no such section is established, the CP will perform all planning functions, such as finding the best entrance to the incident location, and where the first responders could easily help injured people to exit the building. The CP also determines the time frame (operational period) if the building has started to collapse. The major activities of the planning section include [2]:

- Collecting, evaluating, and displaying incident intelligence and information.
- Preparing and documenting Incident Action Plans (IAP).
- Tracking resources assigned to the incident.
- Developing plans.

The CP will also determine if there is a need for a logistics section at the incident, and if so, will designate an individual to fill the position of the logistics section chief. The logistics section chief helps make sure that there are adequate resources (personnel, supplies, and equipment) for meeting the incident objectives. The logistics section chief is responsible for all of the services and support needs.

The CP will determine if there is a need for a finance/administration section at the incident, and if so, will designate an individual to fill the position of the finance/administration section chief. The finance/administration section is responsible for [2]:

- Cost analysis.
- Compensation for injury or damage to property.
- Documentation for reimbursement.



## **2.3 Mobile Ad-Hoc Networks (MANETs)**

MANETs are infrastructure-less and self-configuring networks of mobile routers connected by wireless links without the use of existing infrastructure or centralized control [10] [11]. Devices in MANETs can be heterogeneous and use wireless technologies, such as IEEE 802.11. MANETs are very suitable to host a wide range of applications in emergency situations, like natural or human-induced disasters, military and commercial settings. It is also being used in business sectors for conferences among staffs or other business parties.

### **2.3.1 Evolution of MANETs**

Historically, MANETs have mainly been used for strategic network related applications in order to improve military communications. Military operations cannot rely on access to a fixed infrastructure. MANETs create a suitable framework to address this issue by providing a multi-hop wireless network without a preconfigured infrastructure. The first ad-hoc network applications can be traced back to the DARPA Packet Radio Network (PRNet 1972) [12] [13], which were primarily inspired by the efficiency of the packet switching technology. PRNet features a distributed architecture consisting of networks having broadcast radios with minimal central control.

The growth of laptops and 802.11 wireless networking have made MANETs a popular research topic since the 1990s. The IEEE replaced the “PRNet” with “ad-hoc network”. After introducing the new name, IEEE hoped for new scenarios other than the battlefield, for example, emergency networks, and disaster-relief networks [12] [13]. The term

MANET was used by IETF as a working group for mobile ad-hoc networks.

### **2.3.2 Classification of MANETs**

MANETs can mainly be classified into two different types that focus on the relationships with the other networks [14]: standalone and integrated MANETs. Standalone MANETs are ad-hoc networks that are isolated and each node can only communicate with other nodes in the same networking area. It does not involve a connection with other networks, like the Internet. On the other hand, integrated MANETs are capable of interconnecting with other networks, such as the Internet or 3G networks. For example, Internet-Based Mobile ad-hoc Networks (iMANET) are ad-hoc networks that link MANET nodes to the Internet using fixed internet-gateway nodes. Several proposals are also introduced for interconnecting MANETs and 3G networks. The main objectives of this interconnection are to extend the 3G network coverage [15], or to balance the load between cells [16].

### **2.3.3 Clustering in MANETs**

Clustering in MANETs is about dividing the nodes into different virtual groups. Each group is called a cluster. The cluster nodes may be assigned a different status or function; these nodes can be classified into three different types as shown in Figure 2.6; cluster-head, cluster-gateway, or cluster-member. A cluster-head acts as a local coordinator for its cluster, performing transmission action arrangement, and data forwarding. A cluster-gateway is a non-clusterhead node; it has inter-cluster links that connect neighbor clusters and forward information between clusters. A cluster-member is a regular node, which is a non-clusterhead node, without any inter-cluster links.

Clustering in MANETs is important because it makes it possible to guarantee basic levels of system performance, such as throughput and delay, in the presence of both mobility and a large number of mobile terminals [17].

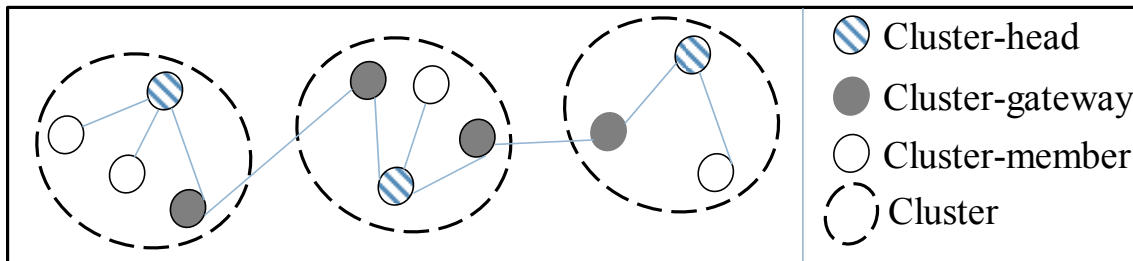


Figure 2.5: Structure of clustering

Reference [17] provides at least three benefits of the ad-hoc networks' requirement for clustering. First, a cluster-based structure facilitates the spatial reuse of resources in order to increase the system capacity. A cluster can also coordinate its transmission events with the help of the special mobile node, like a cluster-head. This can save many resources that are used for retransmission resulting from reduced transmission collision. The second benefit is in routing; because the set of cluster-heads and cluster-gateways can normally form a virtual backbone for inter-cluster routing, the generation and spreading of routing information can be restricted in this set of nodes. Last, the nature of a cluster structure makes an ad-hoc network appear smaller and more stable in the light of each mobile terminal.

## 2.4 Multimedia conferencing

Multimedia conferencing is an important application that can be deployed in MANETs. Conferencing [18] [19] is an essential application that includes different media such as

audio, video, and data communications.

#### **2.4.1 Multimedia conferencing basics**

Multimedia conferencing (also known as multimedia multiparty sessions) can be defined as the conversational exchange of multimedia contents between several parties [20]. It consists of three basic components: conference control, signaling, and media handling. Conference control refers to conference policies, admission control, and floor control. Signaling is used to setup, modify, and terminate the conference. Media handling is concerned with the transportation of the media stream and mixing.

#### **2.4.2 Classification of conferencing**

There are several classification schemes for conferencing; the most common of which are presented in this section. Conferences can be dial-in/dial-out, prearranged or ad-hoc, private (closed) or public (open), with or without sub-conference, and with or without floor control. In the dial-in/dial-out conference, users can attend the conference in two ways: dial-in mode, in which the user calls an authority entity to join the conference, and dial-out mode, in which the user is invited by a user who is already in the conference.

Another scheme discusses whether the conference is pre-arranged or ad-hoc. A pre-arranged conference starts at a pre-determined time and is sponsored by specific parties. The duration of the conference may also be predefined. An ad-hoc conference, on the other hand, starts when the first two parties decide to create a session.

A conference can also be private (closed) or public (open). A closed conference does not

allow the participants to join freely; only the participants who are invited by the conference participants can join. An open conference, on the other hand, publishes its information to all users in the network; any party can join the conference at any time.

Another scheme is whether the conference is with or without floor control. Floor control is a technology that coordinates the concurrent usage of shared resources and data among the participants of a conference. Floor control organizes the conference to ensure fairness and to avoid collision.

Finally, the conference may have sub-conference capabilities. These sub-conferences simulate a conference with different rooms, as in the real world. In each room, parties can hear/see each other, while the people in the other rooms cannot see/hear them. IETF RFC 4353 [22] refers to the sub-conference as a conversation among a subset of participants while the remaining participants are not privy. A sub-conferencing is also called sidebar.

### **2.4.3 Conference models**

A conference model describes the topology used for signaling and media handling in a conference. Media mixing [21] is the core component of media handling. Media mixing's main task is to reduce the number of media streams in the network when there are several users on the call. This is accomplished through entities called mixers that combine the input streams into a single output stream.

Media signaling [21] [23] is defined as the exchange of information, specifically dealing with the establishment of media connections and the management/control of mixers. There are several commonly used topologies for media handling in a conference.

References [21] [23] [24] discuss four main topologies: full-mesh, end-system mixing, multi-cast, and centralized mixing. In a full mesh, every end-system does its own mixing. In end-system mixing, one of the end-system (participants) does the mixing for all other participants. Multicast is enhanced from full-mesh; every end system still does its own mixing, but sends the packets to a multicast address instead of sending it point-to-point. In centralized conferencing, there will be a centralized bridge to do the mixing for all of the participants.

Most of the previous models can be used to establish the signaling session. In the centralized model, the participant can dial-in the bridge to join the conference, or be called (dial-out) to be invited by the bridge. In full-mesh, every participant has a signaling relationship with all other participants in the network. Multi-cast topology can only be used to do the media handling. However, conferencing in MANETs cannot be centralized because it is not possible to assume there are existing bridges in dynamic network environments.

IETF RFC 4353 [22] presents a modern classification of the previous models. Three models are defined and different names are used. A fully distributed conference (full mesh model), a loosely coupled conference (multicast model), and a tightly coupled conference (centralized model).

## **2.5 Summary**

In this chapter, we presented background of emergency response systems and MANETs, followed by the background on multimedia conferencing including description of conferencing basics, classification of conferencing, and conference models.

# **CHAPTER 3 REQUIREMENTS AND EVALUATION OF RELATED WORK**

## **3.1 Introduction**

The FF communication system has various limitations. One of the most significant problems facing FFs on the fire ground is the ability to communicate using multiparty/multimedia with the FF members themselves as well as with the FF leaders and the CP. In an ideal world, FFs would be able to communicate with one another and with the team leader at all times, regardless of where they are or what they are doing.

In this chapter, we derived the requirements for the new FF communication system using MANETs. We also review the existing solutions for building conferencing/sub-conferencing in MANETs in the light of these requirements. The related work is organized into two categories: (1) the work from the standard bodies and the work from outside of these bodies for conferencing component, signaling and media handling, and (2) the work from the emergency response systems for first responders.

## **3.2 Requirements for FF communication system**

In this section, we are going to present our derived requirements into two sub-sections: (1) communication requirements, and (2) user requirements.

### **3.2.1 Communication requirements**

The peculiarities of ad-hoc networks make multimedia conferencing a very challenging task. The first requirement is that none of the FF entities can have centralized control, due



to the fact that MANET is a self-organized wireless network of mobile devices which is not based on any infrastructure or centralized control [10].

The second requirement is that the system should accommodate the FF nodes with limited resources. This requirement is due to the heterogeneity of MANET nodes. In our FF system, we will have different types of nodes (i.e. members, leaders, and CP). Some of these nodes require more communication than the other nodes. This fact implies that some of our nodes will perform more functions than the other nodes.

The third requirement is that the system should be a hierarchical structure instead of the existing flat structure of communication. A hierarchical structure will resolve the problem with flat communication between the FFs. The FFs communication between the members and the leaders/CP will be more organized with hierarchical structuring preventing them from hearing each other's conversations. In addition, the hierarchical structure is important because it makes it possible to guarantee basic levels of system performance.

The fourth requirement is that the system should provide audio/video conferences/sub-conferencing for FFs at all times on the fire ground. Sub-conferencing will allow private communications between FFs, and team leaders will be more focused on their team members and the other leaders/CP. This requirement will provide the FFs with the new communications. For example, The FF member will be able to talk with his team leader and his team members without other teams being able to hear them. Similarly, the team leaders will be able to communicate among each other in private room.

### **3.2.2 User requirements**

In this sub-section, we will present the FFs user requirements in the light of their composition system that have been discussed in Section 2.2.1.2.

The FF composition system has three layers. In each layer, we have different types of FFs. In the first layer, we have the FF members grouped in teams of 4 to 5 members. The first requirement is that we have the FF members grouped in the first layer, the FF leaders in the second layer, and the CP in the last layer.

The second requirement is that one of the FF members from the first layer plays the role of the FF leaders in the second layer. The FFs roles on each layer are fixed and known all the times.

The third requirement is that the FF teams in the first layer need to communicate only within their own teams. However, the team leaders in the second layer need to communicate with each other.

The fourth requirement is that the system should be user friendly to avoid too many or unnecessary options. FFs on the fire ground need a simple system to interact with as their work is very time sensitive.

Table 3.1 summarizes all the derived requirements for the new FF communication system in MANETs. In the table, R1 to R4 are the communication requirements for conferencing/sub-conferencing in MANETs, and R5-R8 are the user requirements for the FF system

Table 3.1: Requirements of FF communications system

| <b>Communications requirements</b> |   |
|------------------------------------|---|
| R1                                 | No centralized entity                                 |
| R2                                 | Accommodate the nodes with limited resources          |
| R3                                 | Hierarchical structure                                |
| R4                                 | Audio/video conferencing/sub-conferencing             |
| <b>User requirements</b>           |   |
| R5                                 | The composition of the FF layers                      |
| R6                                 | The different roles of the FFs                        |
| R7                                 | Communications within each team and among all leaders |
| R8                                 | User friendly GUI                                     |

### 3.3 Related work

In general, multimedia conferencing consists of two main components: a signaling session and a media handling session. The first session is to set up, maintain, and terminate the conference signaling connections. The second session deals with media transportation, and controls the media mixers and media connections.

In this section, we are going to present the existing signaling and media handling schemes for multimedia conferencing. The previous research related to emergency response systems for first responders is also presented in this section.

We first present the signaling protocols for multimedia conferencing. After that, we

present media handling protocols for multimedia conferencing, followed by emergency response systems for first responders.

### 3.3.1 Signaling protocols for multimedia conferencing

Different signaling schemes for multimedia conferencing are introduced and reviewed in this section; some of them are of special interest for our work, but none of them satisfy all of our requirements. We will present them in three subsections: signaling protocol from International Telecommunication Union (ITU-T), signaling protocols from Internet Engineering Task Force (IETF), and signaling solutions from outside the standardization bodies.

#### 3.3.1.1. Signaling protocol from ITU-T

The most widely applied signaling protocol from ITU is H.323. H.323 is a set of specifications that allow multimedia communication services' exchange between several H.323 terminals [25] [26]. H.323 provides multimedia communication services over packet networks. It includes four components, as shown in Figure 3.1:

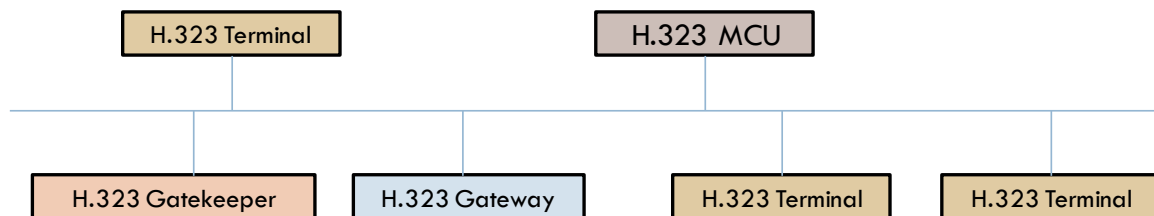


Figure 3.1: H.323 network structure and components [25][26]

- (1) **Terminals** are the client endpoints that provide real-time, bidirectional, multimedia

communications. H.323 terminal can be a personal computer (PC) or a stand-alone device. It supports audio and can optionally support video.

- (2) **Gatekeeper** is the most important of the H.323 components. The gatekeeper's primary job is to act as the central point for all calls within its zone and to provide call control services for registered H.323 endpoints.
- (3) **Gateway** provides connectivity between an H.323 and non-H.323 network.
- (4) **MCU (Multipoint Control Unit)** provides support for conferences between three or more H.323 terminals. All terminals participating in the conference establish a connection with the MCU.

H.323 uses MCU to manage the H.323 terminal calls. Three possible conference models are defined in H.323. One is a centralized model, another is decentralized model, and the third model is a hybrid of the previous two models.

H.323 does not meet our first requirement (R1) because it needs the centralized MCU. H.323 is complex and heavy, and devices in ad-hoc networks may not have enough capability to use it. Optimal use of resources is not considered.

### **3.3.1.2. Signaling protocols from IETF**

The most widely applied signaling protocol from IETF is Session Initiation Protocol (SIP). It is used to create, manage and terminate sessions in an IP based network with one or more participants [27]. A session could be a simple two-way telephone call or it could be a collaborative multi-media conference session.

IETF has defined one core SIP specification [28] and a set of SIP extensions (e.g. [29])

[30]). The core SIP defines a set of functionalities. It also provides the user location, user availability, and user capability [59]. Two kinds of components are defined in SIP: User Agent (UA) and servers.

**User Agent (UA):** is mandatory entity in SIP. A UA contains a user agent client (UAC) and a user agent server (UAS). UAC generates requests and sends those to UAS; UAS receives the requests and responds based on the user input.

**Proxy Servers:** The client (end point) sends the registration, invitation, and other requests to a proxy server. The server, on behalf of the client, forwards the requests to another proxy server, or to the recipient itself.

**Redirect Server:** “A redirect server is a user agent server that generates 3xx responses to requests it receives, directing the client to contact an alternate set of URIs” [33].

**Registrar:** “A registrar is a server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles” [33].

**Location Server:** “A location service is used by a SIP redirect or proxy server to obtain information about a callee's possible location(s)” [33].

SIP can be flexibly applied to different conference models – loosely coupled, and fully distributed. IETF Internet draft [33] describes a loosely coupled conference that uses SIP as the signaling protocol. It is using multicast technique (signaling messages are exchanged using multicast). The signaling architecture in this model is centralized. If we evaluate this model in the light of our requirements, we will find that it does not meet

most of our requirements. For example, this architecture is centralized and does not consider accommodate the nodes limited resources.

IETF Internet draft [33] also introduces dial-in and dial-out conference modes that use SIP as the signaling protocol. In dial-in mode, a centralized conference server is needed. Nodes invite the conference server to join the conference. Dial-out mode also needs a conference server; however, the server invites the nodes to join the conference. If we also evaluate both modes, we will find that it does not meet most of our requirements. A centralized entity is required and the model does not discuss how to use the resources in an optimal manner.

SIP has also been used in [34]. This reference describes a fully distributed approach. In this approach, each node maintains a SIP session with each other node. This approach includes only the end user system and no server is required. Thus, our first requirement is met. However, this approach has several drawbacks. First, the coincident join problem; when two or more participants are invited to join an on-going conference at the same time. In this case, for the participants, there is no general way to ensure that each of the invited parties is aware of the other invited parties. Second, in light of our requirements, this approach still has the same problem of the flat structure; hierarchical structure is not considered in this approach. Moreover, the number of the signaling connections increases exponentially with the number of nodes.

### **3.3.1.3. Signaling solutions from outside the standardization bodies**

Reference [35] continues the work in [34]. It is a SIP solution for conferencing in

MANET. Basically, this work solves the coincident join problem defined in [34]. It proposes a new framework managed by a new entity called “the conference leader”. The conference leader propagates session related information to all participants in the conference. However, the framework is still fully distributed in terms of the signaling architecture. Thus, the drawbacks mentioned in [34] (e.g. the number of the signaling connections increases exponentially with the number of nodes) are still an issue in this approach.

Reference [36] is another framework that uses SIP for conferencing. This work extends the work that has been done in [33]. This framework introduces multiple conference focuses (i.e. servers) and each focus manages a set of participants. The conference focuses are connected in a tree structure. This work solves the scalability issue in [33], however, the framework is still centralized and is not applicable to our system.

Reference [37] is the only work that we found for multimedia conferencing in MANETS. It is a signaling architecture for multiparty/multimedia sessions in peer-to-peer ad-hoc networks based on application-level clustering. This architecture used the cluster mechanism to construct its nodes. There is only one functional entity in this architecture. It is the Signaling Agent (SA). Two types of nodes represent the SA, either a member or a super member. Only one super member exists in each cluster and all the members of the cluster are connected to this super member. Super members are connected to each other. The clusters are dynamically created and deleted by splitting and merging, if the cluster meets the conditions of splitting to form another cluster, or merging with another cluster.

We believe that this architecture meets most of our requirements. For example, the



architecture is decentralized, and it considers the nodes with limited capabilities. However, this work only discusses the conferencing in MANETs, and sub-conferencing is not considered.

### **3.3.2 Media handling protocols for multimedia conferencing**

Media handling systems consist of different components. The media mixers are the cores of the multimedia systems. Media mixing consists of receiving the media streams from multiple sources, combining them into a single media stream, and sending out the new mixed stream without returning the streams to their sources.

In multimedia conferencing [46], mixing sometimes cause delays in the transmission. However, mixing is necessary to reduce the number of streams in the network, and to synchronize between the media streams being played to deliver streams with the correct ordering and timing. Mixing also contribute by controlling which streams are passed and which are muted.

Different mixing architectures for media handling are found in the literature. Centralized mixing, endpoint mixing, hierarchical mixing, partial mixing, and Distributed Mixing Architecture (DMA) are the main categories that will be reviewed in this section.

#### **3.3.2.1. Centralized mixing architecture**

Reference [47] describes this architecture in details. It consists of a centralized mixer that is used by all participants. All participants are connected to this mixer and the mixing process happens in this mixer only. The mixer first receives a set of input streams from

different participants, mixes them, and sends out the mixed stream. The mixed stream is organized so that participants do not receive their own streams.

The centralized mixing architecture does not meet our first requirement for decentralization (R1), making it unsuitable for conferencing in MANETs. This architecture does not use the resources in an optimal manner since some of the nodes may have mixing capabilities that are not used. Furthermore, this architecture is not a hierarchical structure.

#### **3.3.2.2. Endpoint mixing architecture**

This architecture does not use a centralized mixer; each participant does its own mixing [48]. All participants have a direct connection with each other. Each mixer receives the media streams from different participants (i.e. mixers), mixes it and plays the result.

This architecture meets our first requirement for decentralization (R1). However, our second and third requirements (R2 and R3) are not satisfied, since the most powerful nodes and the nodes with limited capabilities perform the same work load. Furthermore, this architecture is not a hierarchical structure. Another drawback is that, this architecture duplicates the mixing streams in the network. Furthermore, to attend the conference in this architecture, each participant should support the media format used by the current session.

#### **3.3.2.3. Hierarchical mixing architecture**

This architecture uses clustering to support the applications with the large number of audio streams [47]. Mixers and participants are connected like the hierarchical structure

of children and parents. Mixers mix the media streams coming from their children and send the mixed results to their parents, then to the root mixer. The root mixer forwards the mixed result to all of the nodes by multicasting mechanism. Once each participant receives the common mixed stream, each participant removes its own stream before playing it.

This architecture meets our third requirement (i.e. hierarchical structured). However, it is not suitable for our system because it does not accommodate the nodes with limited capabilities (R2). Furthermore, our first requirement (R1) is also not met, since the root mixer plays the role of the centralize node.

#### **3.3.2.4. Partial mixing architecture**

The work done on the partial mixing [49] is extended to the work in [47]. This architecture is designed for large scale audio applications on the Internet. The mixers are connected in a tree structure, they receive the media streams from different sources and they choose to mix only a subset of the available media streams based on the network status (e.g. link speed capacity, bandwidth availability). The rest of the unmixed streams will be forward directly to the participants.

This architecture meets our first and third requirements. However, it presents the same drawbacks as the hierarchical mixing architecture. Moreover, media streams in this architecture are possible to deliver at different orders and at different times. This is because of the variation in the mixing procedures between the mixers. Another shortcoming is that, this architecture is designed to support audio application on a large

scale, but does not consider the ability to deliver video streams or sub-conferences (R4).

### 3.3.2.5. Distributed mixing architecture (DMA)

This architecture [51] [52] consists of two main components: a distributed mixing system and a self-organizing system. The distributed mixing system is a two-level structure. The first level is a full mesh network of active nodes that act as media mixers. Nodes, in this level, are assumed to have mixing capability. The second level is made up of inactive nodes without mixing capability that rely on the mixers in the first level. Each second-level node is connected to a first-level node. This architecture can also be described using clustering concept. Mixers and their related nodes are clusters with mixers, acting as a clusterhead. This architecture is self-organized and mixers are selected and removed during the conference. The overview of this architecture is shown in Figure 3.2.

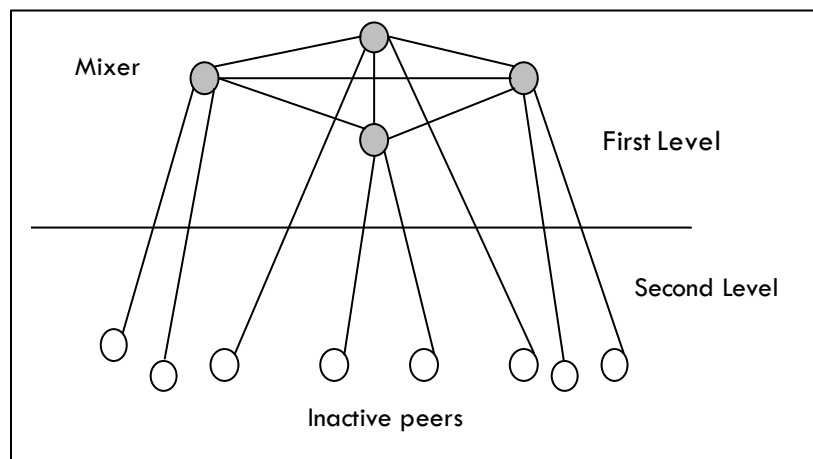


Figure 3.2: Two-level media handling architecture [51][52]

The mixing process in this architecture is done in two steps: first, the media streams arriving from related nodes (i.e. inactive peers) are mixed and immediately sent to other

mixers. Then, buffered streams from this step are mixed with the streams coming from the other mixers and the results are sent to related nodes. In the second step, the mixer should not return media streams to their sources.

The architecture is of special interest to us because it meets most of our requirements. As we can see, the architecture is decentralized (R1), the mixers in the first level are designed to accommodate the nodes without mixing capability in the second level (R2), it is a hierarchical structure (R3), and the architecture targets small, medium, and large MANETs for an audio conferencing system. However, this architecture does not consider video media stream, or sub-conferencing. In the next chapter, we will explain in details how we can adapt this architecture for our system.

### **3.3.3 Emergency response systems for first responders**

The framework defined in [45] and [53] is the only work that uses MANETs for FF that we could find in the literature. It presents a framework that enables a collaboration space between first responders using MANETs. The framework is basically designed to share important information between the fourth first responders (FFs, police officers, medics, and structural engineers) with the civil engineers. This framework redefines the role of civil engineers as part of the first responders' teams. Thus, the use of MANETs is for sharing data, distributing operations, and storing information. This framework does not consider any kind of multimedia/multiparty conferencing/sub-conferencing between the first responders or between the FFs.

The work from the National Institute of Standards and Technology (NIST) [54] presents

an architecture called iBR (Intelligent Building Response) to be used by the first responders.

This architecture seeks to get the real time building information and present it to the first responders while they are en route to the incident, for example, where a fire is, where occupants are, which devices are operating, which lights are on, or which doors are open. There is no multimedia communications in this work either.

The work from [50] is using RFID in combination with an ad-hoc wireless communication network to provide reliable tracking of first responders in stressed indoor environments. The use of ad-hoc networks is only to inform the first responder of their positions via an ad-hoc network of radio terminals that combine RFID reading and radio communication. This work is outside the scope of our work and no multimedia communications are presented. It is mostly on indoor tracking of the first responders' locations.

### **3.4 Summary**

In this chapter, we gave an overview of the work related to signaling for multimedia conferencing, media mixing architectures, and different systems presented for first responders. A list of new FF system requirements was derived. The pros and cons for the previous work were discussed in the light of our requirements. The conclusion is that none of the existing solutions meets all of our requirements.

## **CHAPTER 4 PROPOSED APPROACH**

### **4.1 Introduction**

Providing multiparty/multimedia conferencing/sub-conferencing between the FFs in MANETs is not an easy task due to the peculiarities of ad-hoc networks. Multimedia conferencing is important for FFs. To build a conference, we require two main components: a signaling part, and a media handling part. In this chapter, we present the overall architecture first, and then, we describe the signaling and media handling architectures used to create our conference/sub-conferences. In the last section, we will present the operational scenario of the whole architecture.

### **4.2 Overall architecture**

In this section, we will present two types of architectures: FFs layered architecture and overall conferencing architecture

#### **4.2.1 FFs layered architecture**

In this section, we are going to discuss the complete set of rules governing a particular conference between the FF members, leaders, and CP [1] [2] [7]. Figure 4.1 shows that we have three layers of communication in the FF composition system. Layer one: member's layer, layer two: leader's layer, and layer three: CP layer. It shows also that we have two types of communication: (1) Intra-layer communication, and (2) Inter-layer communication.

Communication in the intra-layer between layer one (the members) and layer two (the

leaders). However, communication in inter-layer between: (1) layer one and two, (2) layer two and three, and (3) layer one and two and three.

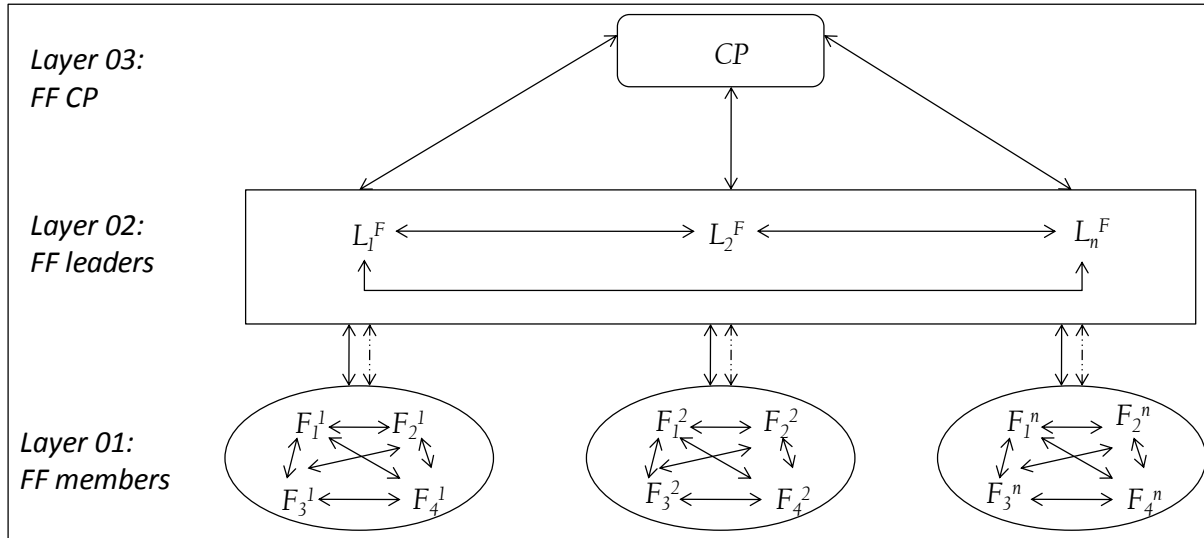


Figure 4.1: Conceptual view of the communication hierarchy between the FFs

*Policies of intra-layer communications between the FFs are organized as follows (the communication here is bidirectional):*

**Layer one:** between the team members

A team member can communicate with other team members within the same team only, for example,  $F_1^1$  from the first layer can only communicate with his team members  $F_2^1$ ,  $F_3^1$ ,  $F_4^1$ .

**Layer two:** between the team leaders

A team leader can communicate with other team leaders when necessary:

- From one leader to all other leaders.



- From one leader to another specific leader without the other leaders hearing their conversation.

For example, the leader  $L_1^f$  can communicate with all other leaders  $L_2^f, L_3^f, \dots, L_n^f$  all together, or he can communicate with a specific leader, e.g.  $L_2^f$  only.

*Policies of inter-layer communications between the FFs are organized as follows (the communication here is also bidirectional):*

**Between layers one and two (members and leaders)**

- A team member can communicate with his leader directly while his team members can hear them. For example, the FF member  $F_1^l$  can communicate with his leader  $L_1^f$ .
- A team member from the first layer cannot communicate with other leaders from the second layer.
- A team member can communicate with his leader from the second layer in private, when it is necessary, without others hearing their conversation.
- A team leader can communicate with his team members all together, or with a specific member, without the other members hearing them.
- A team leader cannot communicate with other teams' members.

**Between layers two and three (leaders and CP)**

- A team leader can communicate with the CP while the other leaders will hear their conversation. For example, the leader  $L_1^f$  from the second layer can communicate with the CP from the third layer.

- A team leader can communicate with the CP in private when it is necessary while the other leaders will not hear their conversation.
- The CP can communicate with the leader(s) when it is necessary:
  - From the CP to all leaders.
  - From the CP to a specific leader without other leaders hearing their conversation.

*Policy of communications between all three layers (the communication here is unidirectional):*

### **Between layers one, two and three**

In emergency cases, any FF (member, leader, or CP) in the network can communicate with all other FFs to alert them. In this case, every FF in the same team, and those in different teams, along with the leaders and the CP can hear this message. This policy is designed for high danger situations, e.g. bomb explosion, building collapsing.

By examining the FFs layered architecture and the communications in the intra-layer and inter-layers, we notice that this solution satisfies most of our system user requirements (R5-R7).

### **4.2.2 Overall conferencing architecture**

Figure 4.2 gives an overview of the proposed architecture. The architecture has three types of the functional entities: User Agent (UA), Super Member (SM), and Super-Super Member (SSM). The UA is mapped to the FF members, SM is mapped to the FF leaders,

and the SSM is mapped to the CP.

This architecture is about the structure of communication between the FFs. As we can see, the architecture has a main conference for all the FFs and a sub-conference for each team; each sub-conference includes a typical number of (i.e. 4 to 5) of UAs and a SMs. The SM (leader) and the UAs (the members of its team) are grouped together in a cluster. The SMs and the SSM are also grouped together. Each cluster along with its members represents a sub-conference.

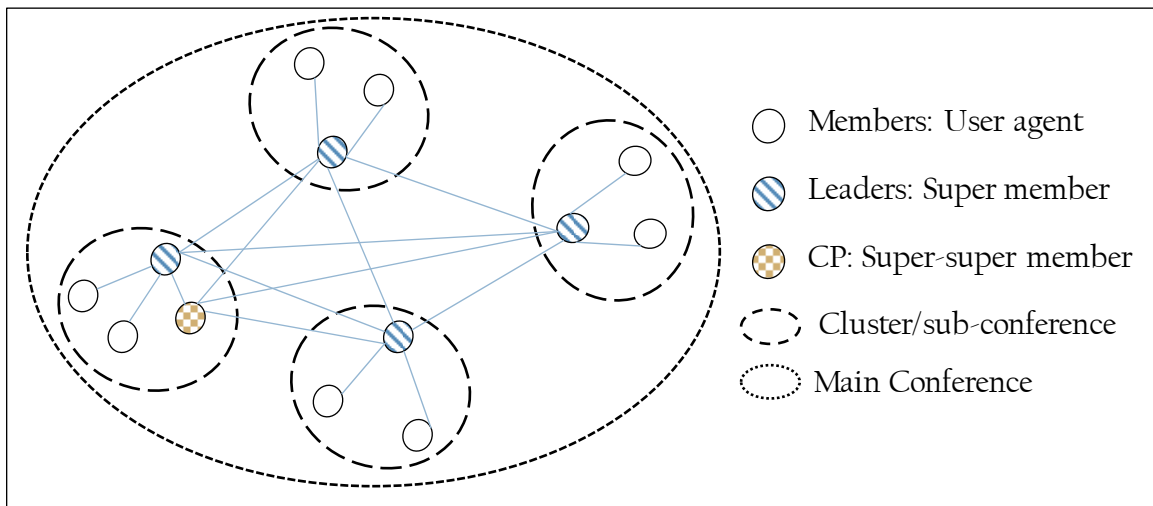


Figure 4.2: Overall architecture

In each cluster, at any given time, there is one and only one SM, and all the other UAs in the cluster are connected to it. A SM has direct connection to the SM of the neighbouring clusters. A cluster with its members (UAs and SM) represents a FF team in the real world.

By examining the FFs conferencing architecture, we notice that this solution satisfies all of our communication requirements (R1-R4).

### 4.3 Signaling architecture

Clustering in MANETs is important because it makes it possible to guarantee basic levels of system performance, such as throughput and delay [17]. Clustering also does not require centralization and it is a main factor in providing scalability. Thus, we believe that it can help in meeting our requirements. This section explains the signaling architecture used to establish our communication system. Then, we will present the operational signaling procedure used to realise our signaling system.

#### 4.3.1 A cluster-based signaling architecture

Figure 4.3 gives an overall view of the proposed cluster-based signaling architecture.

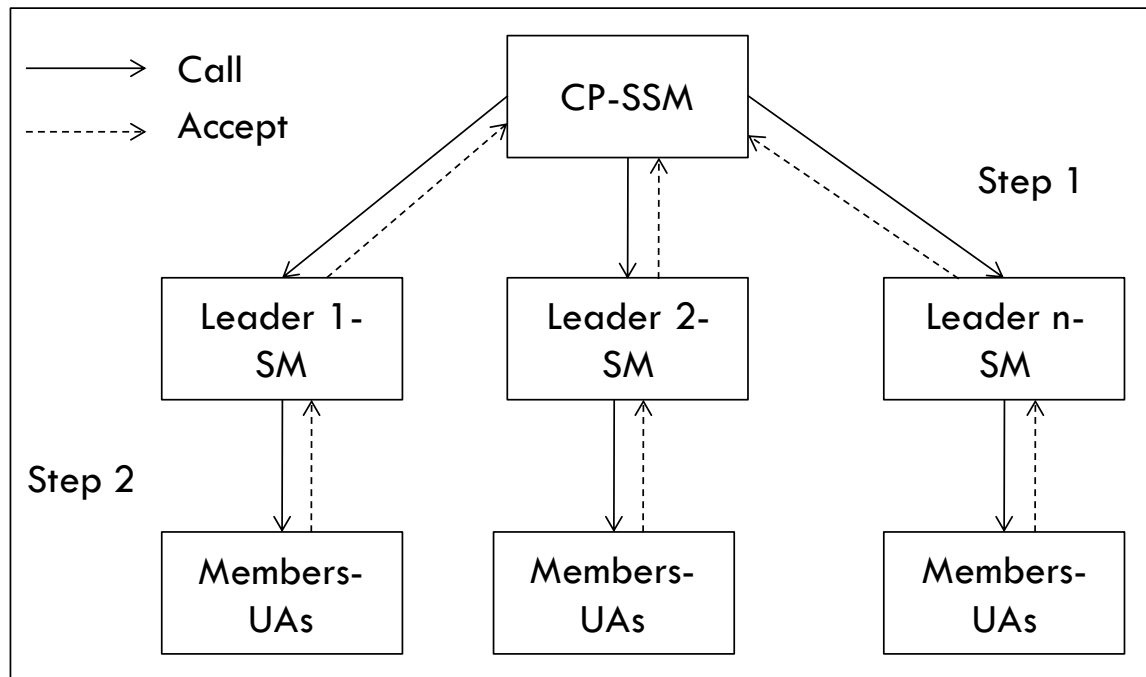


Figure 4.3: A signaling architecture for conferences/sub-conferences in MANETs

The CP (SSM) is a functional entity that is responsible for creating the main conference and for inviting the team leaders to the main conference. Each team leader (SM) will receive an invitation message to the main conference from the CP; the leader will accept the CP invitation message, and join the main conference.

The team leader will start inviting his team members (UAs) to the main conference after he completes joining the main conference. The members will respond to their leaders and accept the invitation to the main conference, which allows the leaders to invite them to the sub-conferences; the members will accept the second invitation and join the sub-conferences with their leaders.

#### **4.3.2 Operational signaling procedure**

This section presents the operational signaling procedure to create the communication system between the FFs. The procedure is based on our cluster-based signaling architecture presented in the previous section and it follows a top-down approach.

Before proceeding with the procedure, we first present the underlying assumptions.

- (1) Each leader knows the CP information (i.e. multicast IP address). The CP knows the leaders information (e.g. names). The leaders will use this information to register with the CP; the CP will make sure that the registered users are the ones who are supposed to be invited to the main conference.
- (2) The leaders know all the related information about their members and the members knows the leader information (i.e. multicast IP address). The members will use this information to register with the leaders. The leaders will perform search on the

registered members to get their team members and to make sure that the registered users are the ones who are supposed to be invited to the main conference then to the sub-conference.

Using these assumptions, all leaders will be able to register with the CP using his multicast IP address. Then, The CP will be able to invite all the leaders to the main conference. Similarly, the leaders will be able to receive the members registration, pick their members to invite them to the main conference and then to the sub-conferences. Figure 4.4 shows an illustrative sequence scenario, where we have one CP, two leaders, and any members per team. The procedure will be executed in six steps.

**Step 1:** The leaders register with the CP. Then, The CP invites all the registered team leaders to the main conference.

**Step 2:** L1 accepts the CP invitation, the first cluster including L1 and the CP is created, and L1 and CP are now in the main conference.

**Step 3:** Once the cluster is created, the cluster super member (i.e. L1 in this case) can invite his team members to the main conference and then to the sub-conference. In Figure 4.3, we can see that L1 invites his team members to the main conference, while the CP is still waiting for L2 to respond.

**Step 4:** The team members respond and accept the invitation of their leader. Now, L1 can invite them all to the sub-conference. In this step, L2 has also responded and accepted the CP invitation.

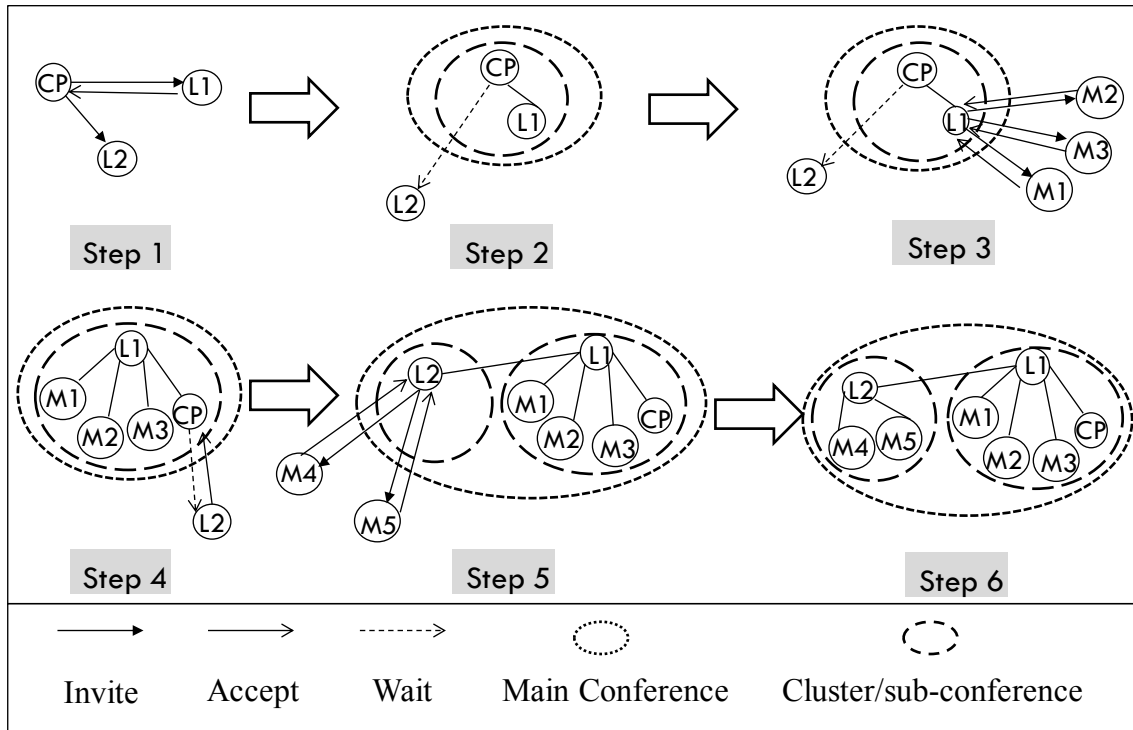


Figure 4.4: Steps of the signaling operations to create the FF communication system

**Step 5:** L2 creates a new cluster and starts inviting its members. In our system, the team leader is allowed to form his own cluster where he is the only member in the beginning, before having his team members in the conference. This situation is acceptable as long as we know that the leader will not be alone in the cluster and will invite his team members immediately after forming and joining the cluster. L2 repeats what happened with L1 in steps 3 and 4. The same thing will be executed if there is another leader. The CP will circulate L2 information (e.g. L2 IP address and port number) to all the leaders who are already ahead of him in the conference (i.e. L1), so they will be able to connect to him. In Figure 4.4, L1 will invite L2 and the two leaders will connect with each other. This process can happen while L2 is inviting his members, or before and at the time he joined the conference.

**Step 6:** The signaling procedure is completed: one main conference is created among all of the participants, leaders, and CP; and a different sub-conference is created for each team. Each sub-conference is represented by a cluster. The members of the same cluster are connected with their super member (i.e. leader), and all the super members and the CP are connected together.

#### **4.4 Media handling architecture**

The work done in [52] proposed a distributed media mixing architecture (DMA) for handling media in MANETs. In this architecture, the authors divided the nodes into two levels. The first level contains the nodes with mixing capabilities (or mixers); these mixers are connected to each other in a full mesh mode. The second level of the architecture contains the nodes without mixing capabilities (called inactive peers). The inactive peers in the second level connect only to one mixer in the first level.

We choose to use DMA architecture as basis of the media handling architecture proposed in this work, because it meets most of our requirements. Therefore,

##### **4.4.1 Adapting DMA to our architecture**

In this thesis, we assume that the team leaders will always have sufficient mixing capabilities. This assumption is due to the fact that our leaders will perform media mixing for their members. The overall architecture presents the fact that the leaders are required to participate in the main conference and in one the sub-conferences, which implies that the team leader is responsible for more communications than his members.



To adapt DMA and use it as a media handling architecture for our solution, we will describe what we have kept from DMA, what we have changed, and which requirements are met or not.

Regarding the decentralization requirement, DMA is decentralized and none of the architecture nodes are permanently centralized, which meets perfectly our needs.

Our second requirement is about accommodating the FF nodes (i.e. FF members) with limited resources (e.g. mixing capabilities). DMA receives the nodes capabilities during the joining time and decides which nodes will be mixers and which will be inactive. In our scheme, the FF nodes' capabilities are already known based on our assumption beforehand. Thus, the team leaders (super members) can be designated as mixers and the user agents (members) can be designated as inactive peers, as shown in Figure 4.5.

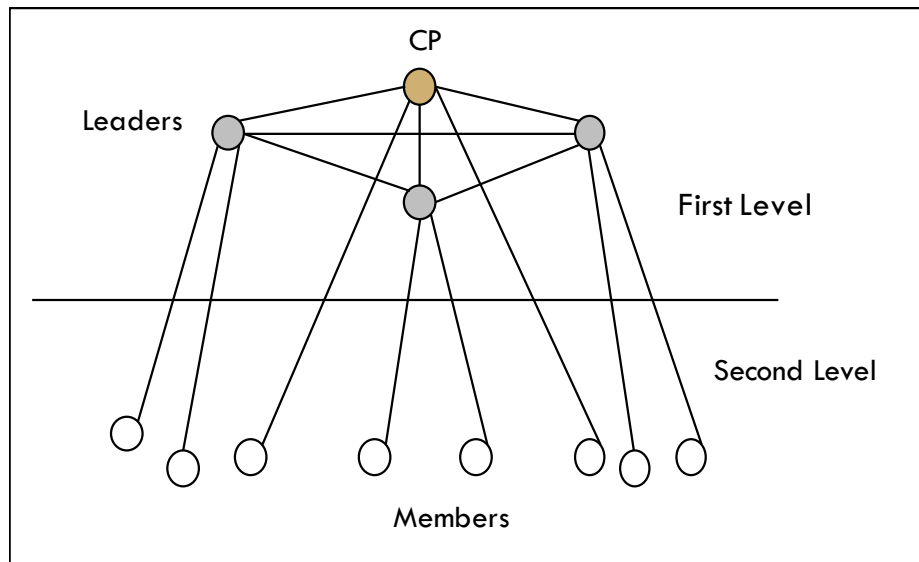


Figure 4.5: Two-level media handling architecture showing the mixers as leaders/CP and the inactive peers as the members

The third requirement states that the system should be hierarchically structured. In DMA scheme, the nodes are divided into two levels and the nodes are dynamic. In other words, the system should automatically allocate or de-allocate the mixers when the network grows or shrinks. Furthermore, it should assign mixers to joiners (inactive peers). It should also preserve efficient resource usage as each event occurs and each decision is made. The situation is different in our FF scheme. We re-use the two levels structure, as shown in Figure 4.5, but our nodes' organizations and the connections among these nodes are fixed, due to the FF system organization (Figure 4.1). The CP assigns a specific number of the FF members (i.e. 4 to 5) to each leader, so each leader will be connected to the same FF members all the time during the incident. Moreover, the team leaders should always be connected with the CP as well.

When it comes to the fourth requirement, which is related to multimedia conferencing/sub-conferencing, the DMA supports conferencing only. It does not address the video or the sub-conferencing support. In our scheme, we added support for sub-conferencing. This is not an easy task because, by adding sub-conferencing, we have to adapt the mixing capability for the nodes (members) in the sub-conferences. The mixing for both the main conference and the sub-conferences is done by the leaders.

The last requirement is for the system to be user friendly. This requirement is about the user interface of our system and is not related directly to DMA.

## 4.5 Summary

We have proposed a new architecture to support multimedia/multiparty communication between FFs on the fire ground using MANETs. The signaling architecture is based on clustering. The operational procedure and the definition of each entity were presented. The media handling part is based on the work done in [52]. Certain modifications have been done to the media handling architecture in order to meet all of our requirements.

From our work, we can see that the proposed architecture can meet all of our requirements. First, there is no permanently centralized entity. Second, the leaders' entities are able to accommodate the FF members by the process of signaling and media handling. Third, the system is hierarchically structured to enable the FFs to communicate on different levels and to enable a basic level of performance. Furthermore, the FFs will benefit from using private communication between the team members and the team leaders using sub-conferencing.

## **CHAPTER 5    PROTOTYPE AND EVALUATION**

### **5.1 Introduction**

In the previous chapter, we proposed an architecture that supports multiparty/multimedia conferencing/sub-conferencing in standalone MANETs for the purpose of use by FFs. Consequently, this chapter will present the implementation of this architecture.

In the section 5.2, we present the implementation of conferencing/sub-conferencing in standalone MANETs. In the section 5.3, the prototype for signaling/media handling is presented. In the section 5.4, we validate our architecture, a scenario is presented and some performance measurements are collected and analyzed.

### **5.2 Implementation of conferencing/sub-conferencing**

This section will present and review the technologies and tools used to implement the proposed architecture. In the first, we present the selected implementation technologies. In the second, we describe scenarios for conferencing/sub-conferencing in standalone MANETs. In the third, we present the overall software architectures and mixers design. Lastly, the user interfaces' design is present.

#### **5.2.1 Implementation technology**

SIP is not the only signaling protocol available, but it is an application layer control protocol that can establish, modify, and terminate multimedia sessions. We choose to use SIP (Session Initiation Protocol) as the signaling protocol of our architecture for the following reasons: (1) SIP is a simple, light-weight and flexible, (2) it is easily extensible

and widely deployed, and (3) it can be used with a variety of mobile devices.

SIP can start new sessions, or modify the existing sessions, by inviting new participants, for example, in multimedia conferencing. Media can be added to or removed from an existing session. SIP supports user location, user availability, user capabilities. SIP is presented in the IETF draft [28].

We re-use the same SIP messages from core SIP [28], without the need to extend them. The SIP messages we use are: REGISTER, INVITE, OK, ACK, and REFER. The REGISTER message is used to handle the leaders' and the members' registrations for both the main conference and the sub-conference. The INVITE, OK, and ACK are used to deal with basic session set up. The CP uses the REFER message to refer all the existing leaders in the main conference to INVITE the newly arrived one.

### **5.2.2 Scenarios for conferencing/sub-conferencing**

This section presents the following scenarios: (1) start the conference and leaders' registration; (2) two leaders invited to the main conference; (3) three leaders or more invited to the main conference; and (4) members invited to the conference and to the sub-conference.

For the purpose of these scenarios, we are going to use a CP with three leaders and three members distributed between the main conference and the sub-conferences.

**Start the conference and leaders registration** – Figure 5.1 represents the CP as the entity who is responsible for start the conference. After the CP starts the conference, he

listens to a specific multicast address, waiting for any leader to register at this multicast address. The team leaders have the CP multicast address. This multicast address stored in an XML file (Appendix A) that will be uploaded on the leaders' machines. An XML file contains the leaders information will uploaded on the CP machine as well.

The CP stores the leader's information sequentially after registration. Then, he starts inviting them to the main conference one by one as shown in Figure 5.1. The next scenario will describe in details how the two leaders invited to the main conference.

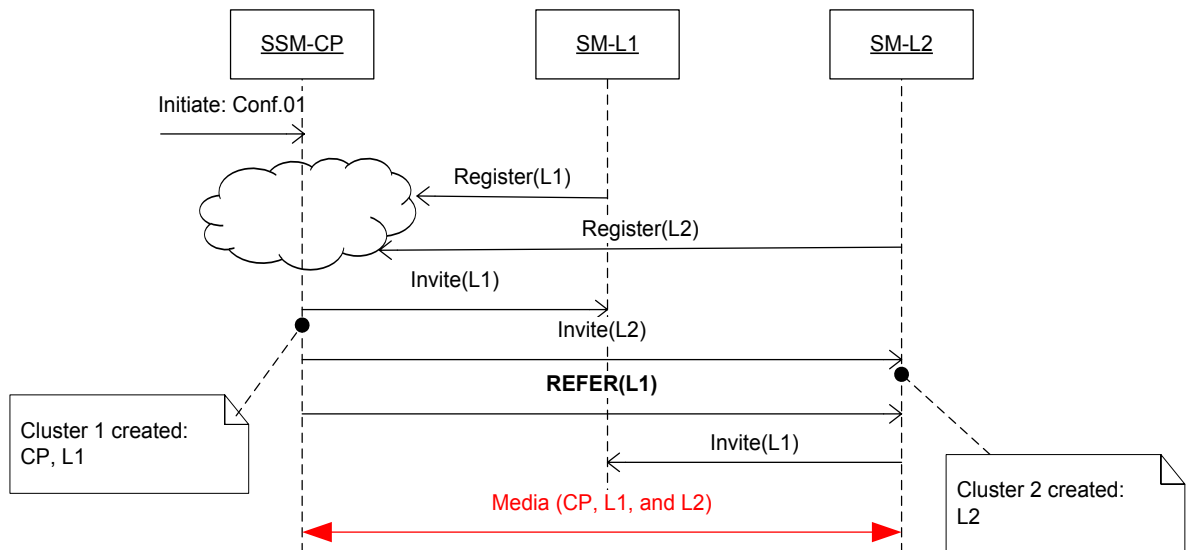


Figure 5.1: Two leaders invited to the main conference

**Two leaders invited to the main conference** – After the leaders have been registered, the CP uses the leaders stored information to INVITE them to join the main conference (Figure 5.1). The CP INVITEs the registered leaders sequentially, following the order on

which their registrations have been received. We assumed L1 has registered first.

The signaling mode in the main conference between the leaders and the CP is full mesh. However, the leaders (i.e. L1 and L2) do not have each other's information in order to INVITE each other. To resolve this problem, we reuse the same stored information that the CP has acquired from the registration process, which we make available to the leaders. After the CP INVITEs the leader L2, it sends a REFER message to that leader with the registration information of L1. When L2 receives the REFER message, it INVITEs the leader whose information is given. At the end of the sequence diagram of Figure 5.1, we have a main conference in a full mesh mode with three entities CP, L1, and L2.

**Three leaders or more invited to the main conference** – Figure 5.2 extends the process of joining the leaders to the main conference as given in the previous scenario. Figure 5.2 represents a situation in which a third leader joins the main conference. He has to go through the same process as L1 and L2; however, in the normal case, the CP must send two REFER messages to L3 so that L3 will be able to INVITE the other two leaders that existed before him in the main conference. For the sake of simplicity and to use fewer SIP messages, we used another technique, namely, that of using multiple targets in the same REFER message. Reference [55] explained the procedure of using multiple targets in the same SIP REFER message. An example of a REFER message with a list of targets is given in Appendix B.

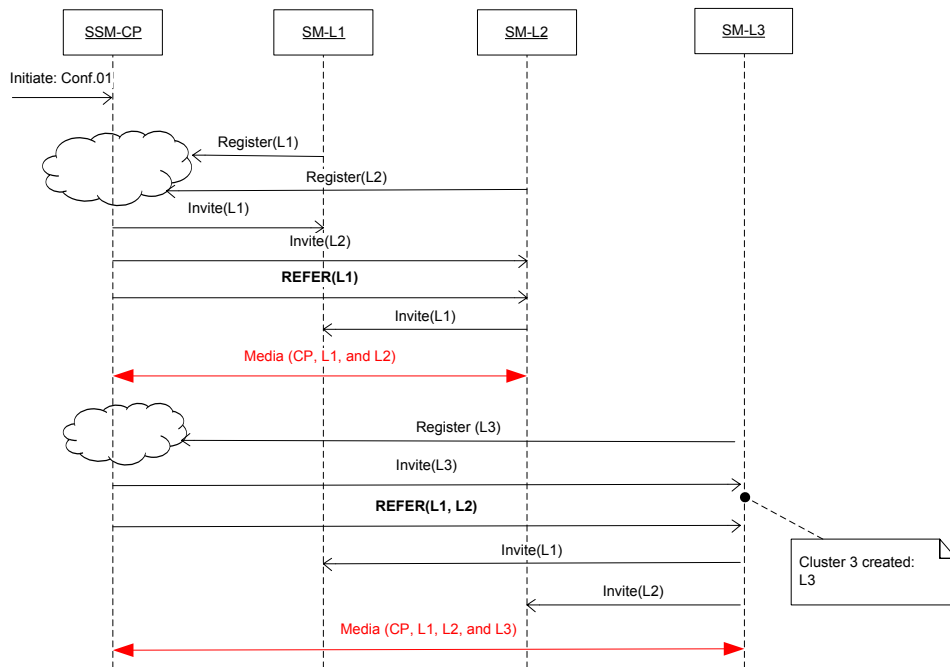


Figure 5.2: A third leader invited to the main conference

The same procedure is repeated when any number of leaders wants to join the main conference. They register first. The CP then INVITEs them to the main conference and, using the technique of multiple REFER-Targets, REFERs them to all the leaders who are already in the main conference.

Now we have a main conference with a CP and the number of leaders that is required, each leader having his own cluster (no members added yet), except the first cluster, which contains the CP and L1.

**Members invited to the conference and to the sub-conferences** – The procedure used for members to join the conference and the sub-conferences is not the same as that used for leaders. For this, there are three choices, and each choice has advantages and



disadvantages. We first present and discuss the choices and then identify the one we choose for our solution.

**First choice:** the members REGISTER with the CP. The CP forwards this information to his leader to which to members belong. The leader then uses this information to INVITE his members to the main conference and then to the sub-conference.

Figure 5.3 represents a member, M1, who wants to join the main conference and sub-conference with his leader L1. In this situation, this member should register with the CP multicast address, and the CP forwards this message to the entitled leader, who is waiting for this member to join.

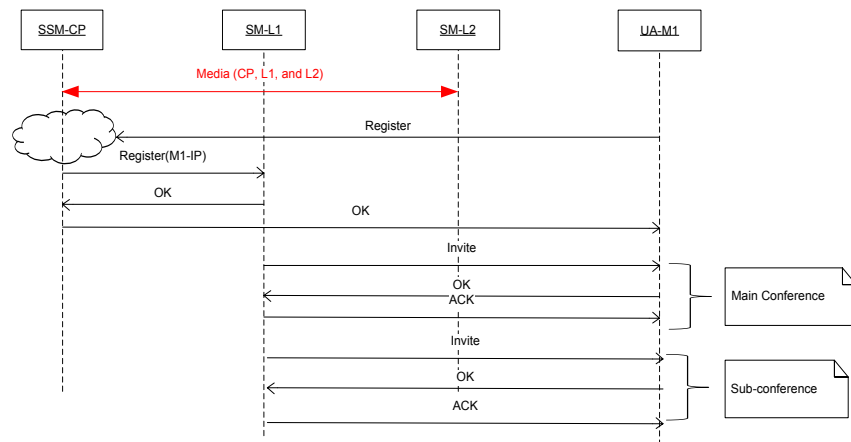


Figure 5.3: First choice of members invited to the conference and to the sub-conferences

**Second choice:** the members REGISTER to a shared multicast address, in which all leaders and the CP listen. Each leader searches and gets his members information from this shared multicast address.

Member M1 in Figure 5.4 does not need to register to a specific multicast address; he will

directly register to a shared multicast address between the CP and the leaders. Each leader has access to this shared multicast address. He is thus able to get his members information from there and to INVITE them to the main conference and to the sub-conference.

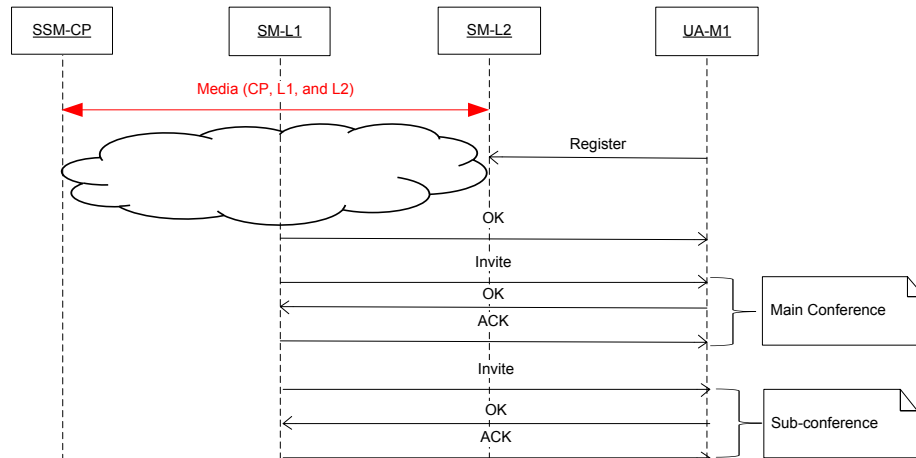


Figure 5.4: Second choice of members invited to the conference and to the sub-conferences

**Third choice:** the members REGISTER to a shared multicast address, in which only the leaders listen. Each leader searches and gets his members information from this shared multicast address.

Figure 5.5 shows that the CP is not a part of the leaders' multicast address. Member M1 will register to the leaders' multicast address. Each leader has access to this shared multicast address. He is thus able to get his members information from there and to INVITE them to the main conference and to the sub-conference.

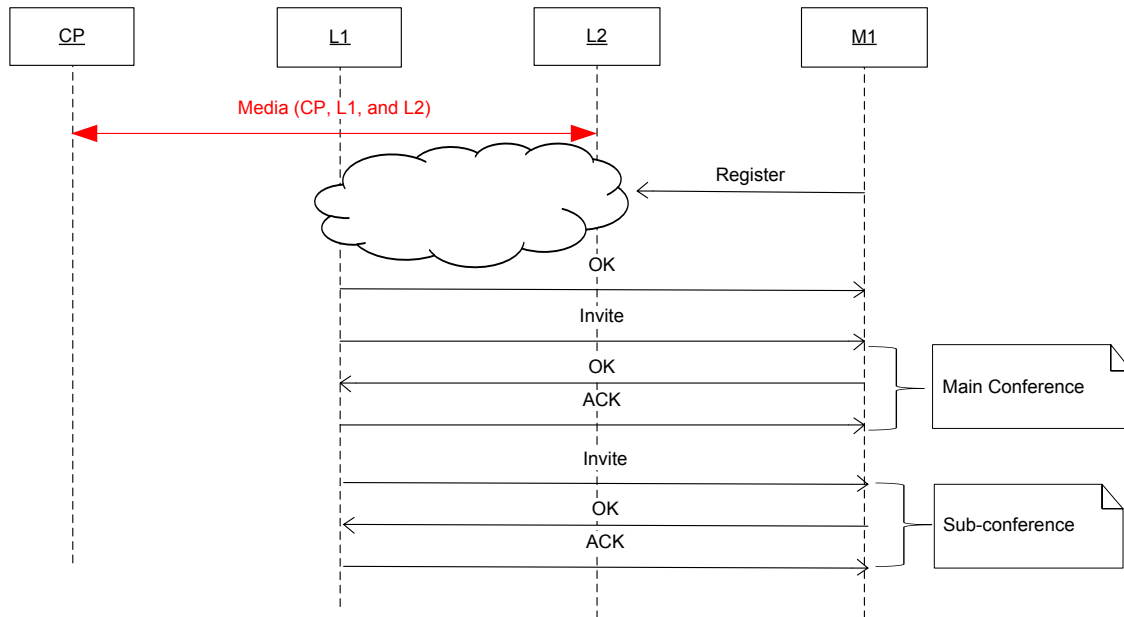


Figure 5.5: Third choice of members invited to the conference and to the sub-conferences

In this choice, we have two types of multicast addresses. Type 1: CP multicast address, used by the leaders to register with the CP. Type 2: Leaders multicast address, used by the members to register with their leaders.

By examining the previous choices, we see that they are very similar. It is therefore difficult to determine the better choice. The following reasons clarify why one alternative was chosen over the others:

**(1) Number of SIP messages**

There are fewer SIP messages in choices three and two than in choice one.

## **(2) Scalability**

Choice two and three are capable of handling a growing number of members because each leader handles his team members only (4-5 members). Choice one is not scalable because the CP handles all the signaling from the leaders and the members together.

## **(3) Resources consumption**

Choice one requires little work from the leaders. The CP handles all the registration processing. However, choice two requires both the CP and the leaders to process member registrations. In choice three, the CP is not required to perform any thing.

## **(4) Node Failure**

The first choice puts all signaling registrations on the CP, in that case, the CP has to handle all members registration and forward it by himself, thereby it may lead to failure in case if we have a big number of FFs (unsuccessful connection). On the other hand, the second and the third choices let the leaders to handle only small groups of members at a time.

## **(5) Information required for each FF in each choice**

In choice one, the CP needs to have its leaders' and members' information (this choice requires an additional assumption in our proposed architecture). In choices two and three, the leaders need to know only their members' information (work fine with our assumption).

The conclusion of the previous comparison is shown in Table 5.1.

Table 5.1: Comparison between the three choices

|                               | <b>Choice 1</b>            | <b>Choice 2</b>         | <b>Choice 3</b>        |
|-------------------------------|----------------------------|-------------------------|------------------------|
| <b>Number of SIP messages</b> | More                       | Less                    | Less                   |
| <b>Scalability</b>            | Less                       | More                    | More                   |
| <b>Resource consumption</b>   | Leaders: YES<br>CP: YES    | Leaders: YES<br>CP: YES | Leaders: YES<br>CP: NO |
| <b>Node failure</b>           | More                       | Less                    | Less                   |
| <b>Information required</b>   | CP→ (Leaders ,<br>Members) | Leaders→<br>Members     | Leaders→<br>Members    |

An examination of these results shows that choice three is much better than choices one and two; however, the only different between choice two and three is that we have resource consumption during the time spent waiting to get the members information. We also notice that the CP does not need to be included in this process because the CP will not use or benefit from members information. Using this information introduces the third choice in a way that will save some resource consumption.

The third alternative shows an advantage, namely, that the CP does not need to waste his processing power with the second group of multicast address.

If we combine the first part of our SIP implementation (Figure 5.2) with the third choice, we have full implementation allowing the CP, leaders, and members to join the main conference and the sub-conference. Figure 5.6 summarize the whole procedure. The OK and ACK messages are not presented to keep the figure clear.

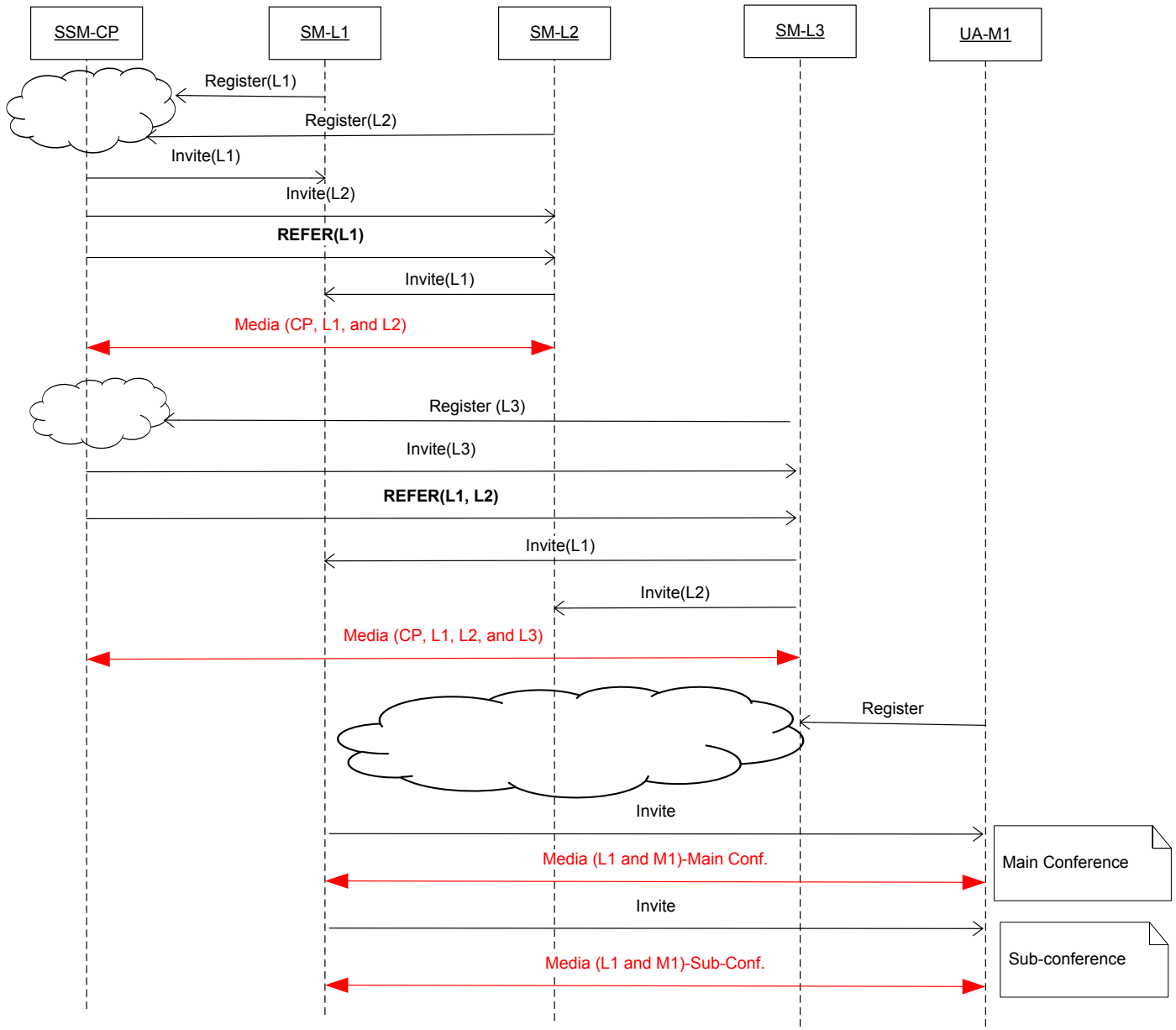


Figure 5.6: Overall SIP implementation architecture

Figure 5.7 summarize the whole procedure. The OK and ACK messages are presented here.

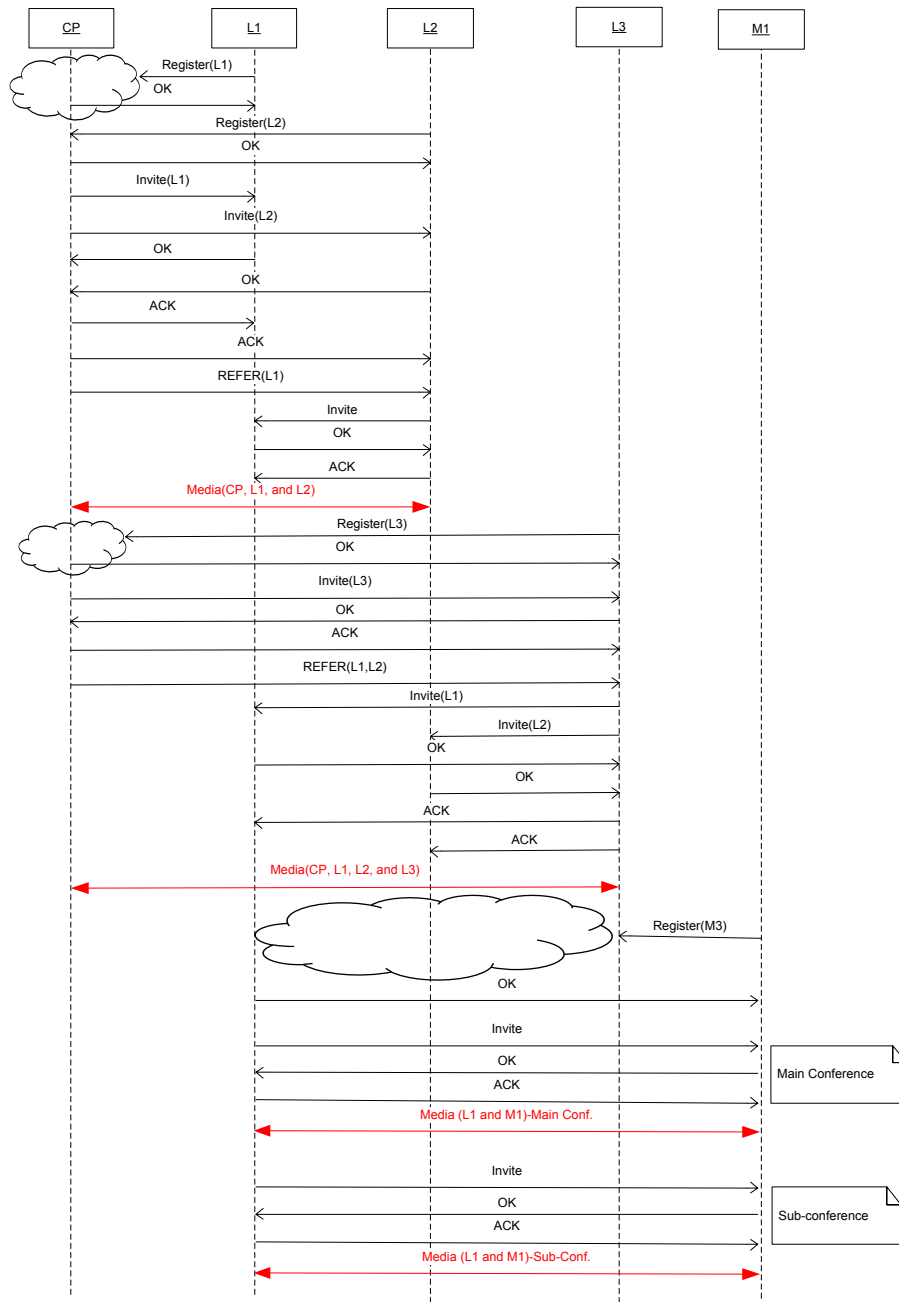


Figure 5.7: Overall SIP implementation showing all messages

### **5.3 Signaling/media handling prototype**

We have developed an audio conferencing/sub-conferencing system in MANETs that can be used by FFs. We implement all FFs nodes (CP, Leader, and member), and signaling, media handling for each node. The present section introduces the tools that were used in this implementation, the system's software architecture, and the User Interface (UI) design.

#### **5.3.1 Programming language and tools**

We have used Java as our programming language. There are two types of Java freeware used in the prototype. The first is JAIN SIP (Java API for Integrated Networks), which is considered the basis for implementing our SIP-based nodes, and the other is JMF (Java Media Framework), which is used for media transformation and mixing.

We selected JAIN SIP for our signaling system for the following reasons. JAIN SIP implements a SIP stack using Java. Its source code can be found at [56] [57]. JAIN SIP provides a number of Java APIs that are typically used for client-side application development, but they can also be used for implementing server side applications. The latest version of JAIN SIP supports (1) the entire core of SIP (RFC 3261) messages, (2) provisional responses (RFC 3262), and (3) the REFER method (RFC 3515).

For the media transportation and mixing, we used the real time protocol (RTP) [46], which is provided by the JMF API (JMF) [58]. The prototype implements a two-level structure, in which the mixers in the first level represent the FF leaders. In the second level of the structure we have the list of FF members connected with only one leader



(mixer) from the first level.

We used JMF API to facilitate the implementation of the mixers; this API includes functions such as: enable mixer, disable mixer, start mixer, and stop mixer. The media streams of the names of the members coming from the sub-conferences are received and merged. The result is sent back to the related nodes.

### 5.3.2 Software architecture

Three different architectures are presented in this section. The three architectures represent the three FF nodes (CP, leader, and member). Figure 5.8 shows the CP architecture, which includes two main components: the *Conference Core* and the *Mixing Core*. The *Conference Core* handles the signaling part between the CP and the leaders. On the other hand, the *Mixing Core* handles media transportation and mixing.

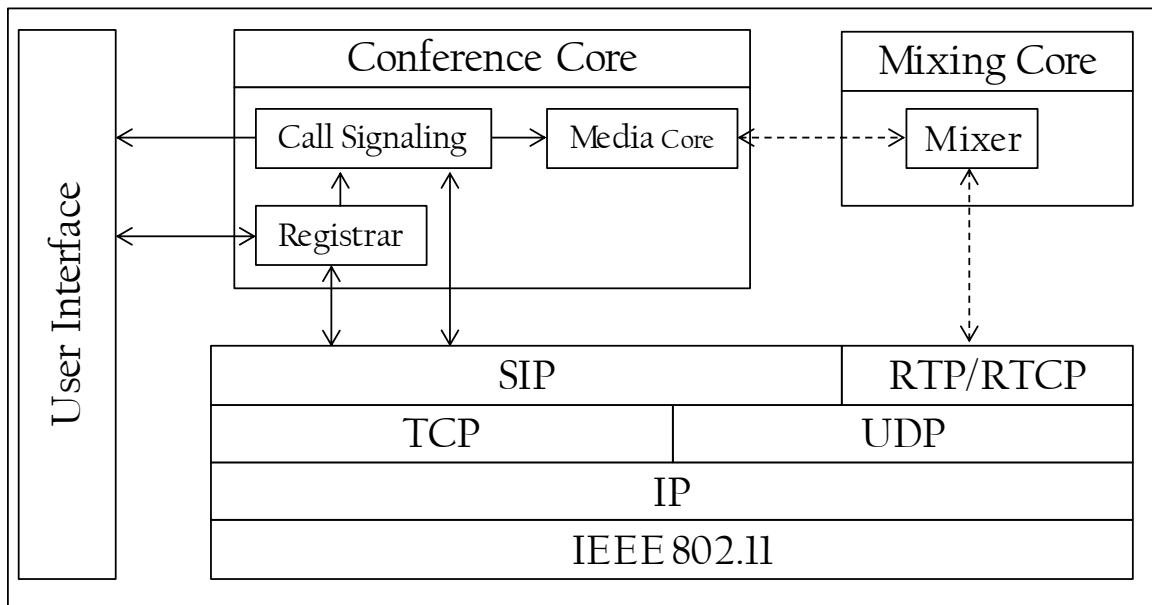


Figure 5.8: The CP software architecture

The User Interface (UI) is used by the CP to initiate the main conference and waiting the leaders' registrations to invite them to the main conference. The *Conference Core* has three main components: *Call Signaling*, *Media core*, and *Registrar*. The *Call Signaling* handles the leaders' registration messages, inviting and referring them to each other, and enables the mixer from the *Mixing Core* to know when the mixer should start or stop mixing.

The leader software architecture is similar to that of the CP, as shown in Figure 5.9.

The only two differences are the following:

- (1) the *Conference Core* at the leader side, includes an extra component called *Handle Register (HR)*. The HR handles the leader registration with the CP. The *Registrar* component here is to handle the members' registrations;
- (2) the mixer in the *Mixing Core* is completely different from the CP mixer. The mixer's design is explained in the next section.

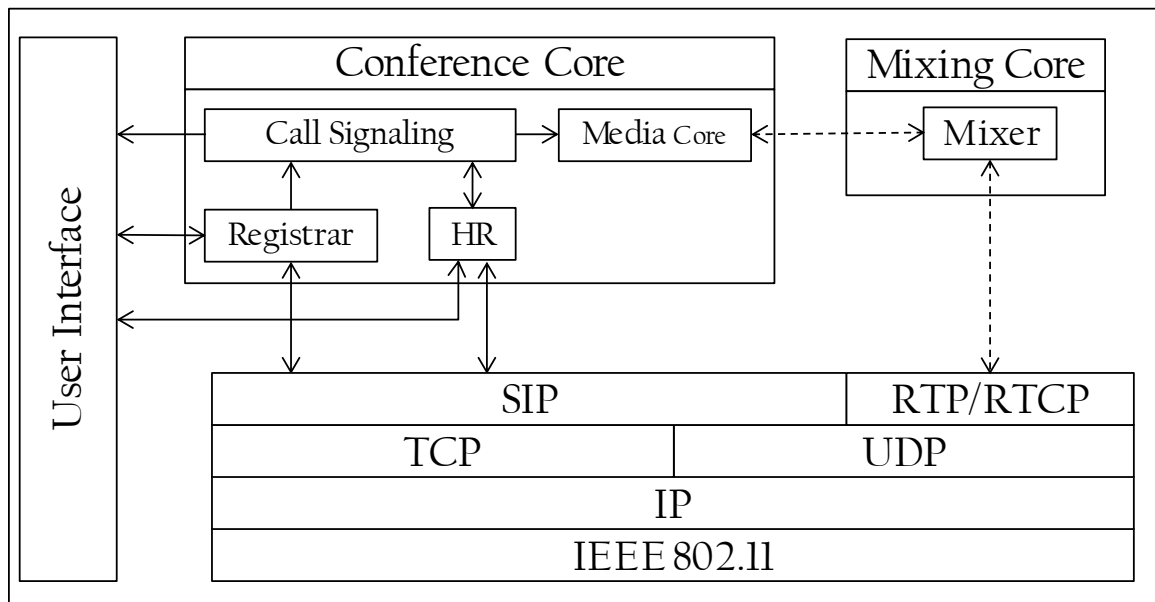


Figure 5.9: The leader software architecture

The last software architecture is the member software architecture as shown in Figure 5.10. The member software architecture also has two main components: the *Conference Core* and the *Media Core*. However, there is no component for mixing. The member simply sends or receives media, but it doesn't mix. The *Handle Register (HR)* is to handle the member registration with the leaders.

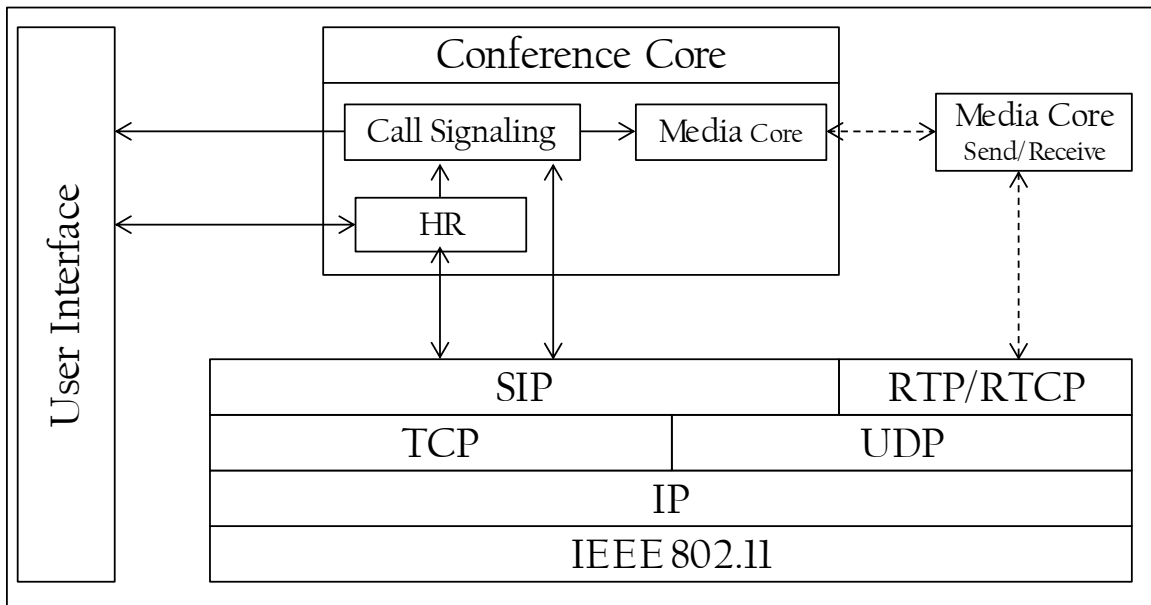


Figure 5.10: The member software architecture

### 5.3.2.1. Mixers' design

Two mixers are presented in the previous section without the explanation of how each one of them works. This section briefly explains mixers and their components.

The first mixer architecture is the CP architecture shown in Figure 5.11. The mixer acts both as a source and as a receiver. It consists of six components: *Receiver*, *Media Buffer*, *Synchronizer*, *Source Merger*, *Codec Transformer*, and *Transmitter*.

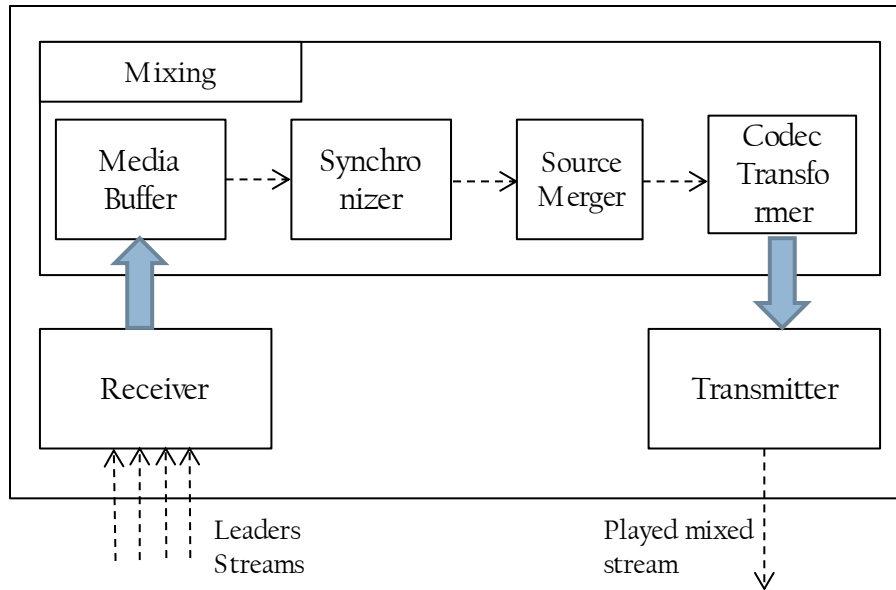


Figure 5.11: The CP mixer core functional architecture

The CP basically mixes for himself. It receives the leaders' media streams, mixes them, and then plays the mixed result. The mixing process starts by passing the media streams to the mixer; the media streams are received by the *Receiver* component. The *Receiver* receives the leaders' streams from different sources and forwards them to the *Media Buffer*. The *Media Buffer* removes jitter and passes the new set of streams to the *Synchronizer* component. The *Synchronizer* component prepares the media streams that have the same timestamp to be mixed in the next step. The mixing operation is done in the *Source Merger* after the streams are received from the *Synchronizer*. The *Source Merger* makes sure that the mixed result passes through the *Codec Transformer* before the *Transmitter* plays the mixed result.

The mixing procedure is completely different in the case of the leader mixer as shown in Figure 5.12. In this case, the leader requires mixing for himself and for his team

members. The team leader could be in two different places, main conference or sub-conference. In case when he is in the main conference, he mixes for himself and still performs mixing for his team members while they are in the sub-conference. The same situation occurs if he is in the sub-conference. To facilitate the control of media streams received from different sources, we have designed two *Receivers* and two *Media Buffers*. The *Main Receiver* is to receive the media from the CP and the leaders (main Conference) only, while the *Sub Receiver* is to receive the media from the team members (sub-conference) only.

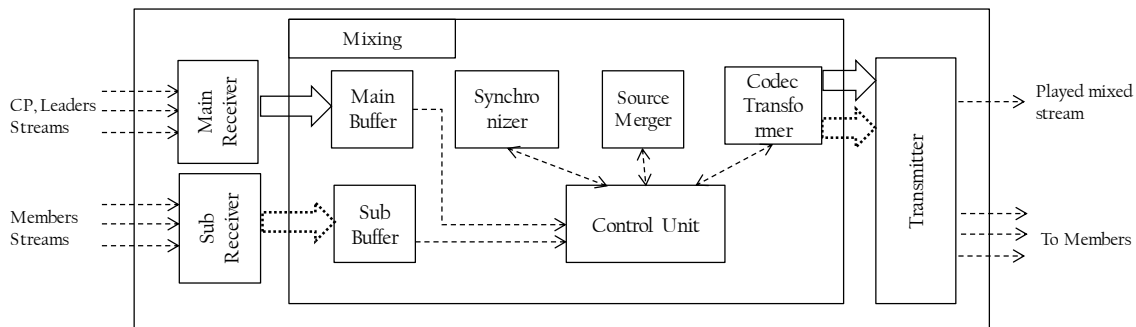


Figure 5.12: The leader mixer core functional architecture

Each *Receiver* forwards the media streams to different *Media Buffers*. The *Main receiver* forward the media received from the CP/leaders to the *Main Buffer*. The *Sub receiver* forward the media streams received from the members to the *Sub Buffer*. The *Main Buffer/Sub Buffer* responsible for activate the *Control Unit*. The *Synchronizer*, *Source Merger*, and *Codec Transformer* are enabled and disabled from the *Control Unit*. Based on which streams are received at the moment, the *Control Unit* enables/disables the right component to perform the mixing and to forward the mixed result to the *Transmitter*. The *Control Unit* makes sure not to return the media streams to their sources.

### 5.3.3 User interfaces' design

Based on our requirements and the structure of our FFs communication system, we have created three different user interfaces: the first one for the CP, the second one for the leaders, and the last one for the members. Each of these interfaces includes different functionalities to meet the FFs needs and requirements. In this section, we present the three different user interfaces that were created.

We used an XML file (Appendix A) to upload the configuration information for each FF (e.g. CP multicast IP address, port number, and FFs names). To save the FF time, we decided to use a file and upload it on the FF application instead of entering this information by hand. The FF does not have extra time to enter information that he does not have knowledge about (e.g. Multicast addresses). This file can be prepared before the incident. This file contains the following information: multicasts IP address, FF related information (e.g. Names, team leaders, etc.). These files are shown in Appendix A for the CP, leaders, and members.

Figure 5.13 shows the three different user interfaces. The first user interface (Figure 5.13-a) is the CP user interface. The CP XML file has the CP multicast IP address associated with the multicast port number (e.g. IP: 228.5.6.7, port: 5000). The XML file also has the list of leaders who will participate in the conference.

The CP first presses *Start Conference* button. Then, he gets the list of his leaders from the appropriate XML file and adds them to the GUI with an offline status. Then, he will start receiving the leaders REGISTER messages. The system will start inviting the registered

leaders automatically and referring them to each other when needed. Once the first leader accepts the CP invitation, the conference is started. The leaders' names will move from the offline status to the online status, upon the acceptance of the CP invitation to the main conference.

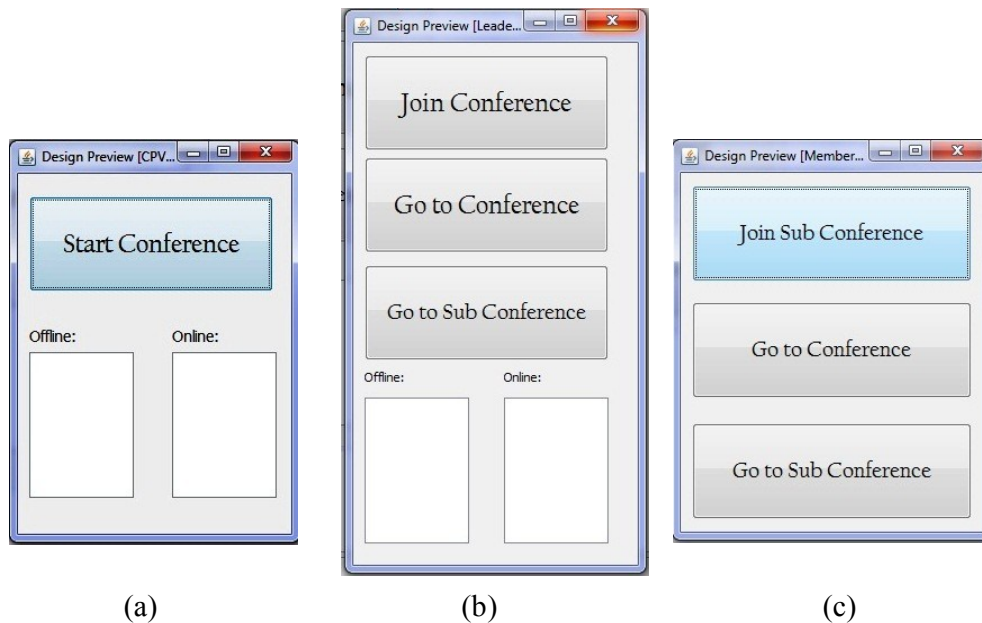


Figure 5.13: FFs User Interface: (a) CP, (b) Leaders, (c) Members.

Figure 5.13-b shows the leader user interface. The Leader XML file (Appendix A) has the CP multicast IP address that is associated with the multicast port number (e.g. IP: 228.5.6.7, port: 5000), the leader multicast IP address that is associated with the multicast port number (e.g. IP: 228.5.6.8, port: 6000), and the list of members who will participate in the sub-conference. When the leader presses the *Join the Conference* button, two functions are executed:

- (1) The leader registers with the CP by sending a REGISTER message to the CP multicast address; and

(2) The leader gets the list of his members from the appropriate XML file and adds them to the GUI with an offline status. It then waits to receive their registration information.

Figure 5.13-c shows the member user interface. The member XML file (Appendix A) includes the leader's multicast IP address and port number (e.g. IP: 228.5.6.8, port: 6000). The interface requires from the member at the beginning to press the *Join Sub Conference* button.

When the member presses the button *Join Sub Conference*, he registers with the leaders' multicast address. All leaders are able to access this member registration information. The leader who has this member in his team will basically invite him to the main conference first and then to the sub-conference.

After repeating the previous steps for all the team leaders and members, we have one main conference between the CP and any number of leaders, and the same number of sub-conferences, a sub-conference for each team leader.

Each team leader is in the main conference at the beginning as a default. While in the main conference, he hears all of his members in the sub-conference. When the leader wants to talk in his sub-conference, he presses the *Go to Sub Conference* button as shown in Figure 5.13-b. When he wants to go back to the main conference, he presses *Go to Conference*. The same applies for the members. The member is in the sub-conference as his default place; if he wants to talk in the main conference, he needs to push the *Go to Main Conference* button as shown in Figure 5.13-c. We provided the member with the ability to talk in the main conference for the following reasons:



(1) if he received no response from his leader, he may need to go to the main conference to inform the CP and the other leaders;

(2) to satisfy our emergency situation policy. This policy is explained in Section 4.2.1. It allows any FFs in the network to talk to the all other FFs even if he is not allowed to talk with them in the normal cases.

The UI is deliberately design to be simple in order to satisfy our last user requirement (R8).

## **5.4 Performance measurements**

In this section, we present the performance results collected from the prototype. The environment settings are presented first. Second, the performance metrics are presented, followed by the results.

### **5.4.1 Environment settings**

We have developed a prototype for an audio conferencing/sub-conferencing system, using Java as our programming language in an MS Windows environment. We have created an ad-hoc environment using 10 PCs. Five of these machines are Pentium 4s with 4 GB RAM running Windows 7; the other 4 machines are Pentium 4s with 1 GB RAM running Windows XP. The machines are located in two different rooms on the 9th floor of the EV Building at Concordia University. The prototype setting is shown in Figure 5.14.

Figure 5.14 shows that we have assigned the computers with the highest memory RAMs

to the CP and the leaders, as the leaders should have powerful machines because they perform mixing for their members as well as the leaders are required to move between the main conference and the sub-conference. The members are assigned to other sets (i.e., 1 GB RAM) of computers because they need to perform only basic conferencing operations.

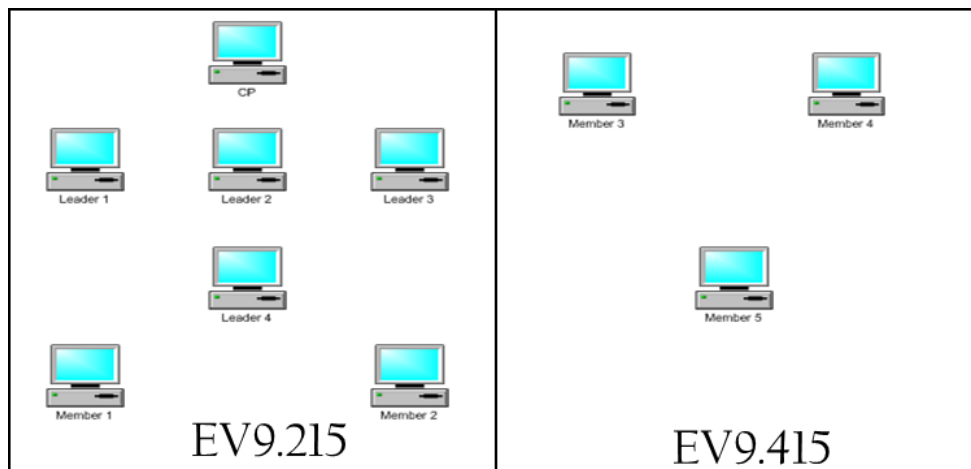


Figure 5.14: Prototype settings

## 5.4.2 Metrics

Two metrics are used to measure our signaling system: network load and delay. We first discuss the network load, followed by the delay.

### 5.4.2.1. The network load

In our test, we calculated the total number of bytes sent and received by each FF in the signaling messages. The signaling overhead measurement includes all signaling activities from the registration phase until joining the main/sub-conference.

In this set of experiments we evaluate the total number of bytes sent and received in the

following scenario, we have used six FFs organized as follow: three FFs in the main conference (CP, L1, and L2), and three FFs in the sub-conferences (M1, M2, and M3). The main conference includes the CP and the two leaders, a sub-conference for each team leader with two FF members in the first sub-conference and one FF member in the second one. To simplify the measurements and get clear result, we present the total number of bytes in each FF node in our scenario.

**In the main conference-CP, L1, and L3 node:** in the main conference, after the two leaders registered, the CP needs to invite them to the main conference. As the first to be called, L1 exchanges 2000 bytes only as shown in Figure 5.15. L2 uses more bytes (i.e. 5466 bytes) because he receives more signaling than L1 (i.e., a REFER message from the CP to call L1). No signaling from the CP occurs after referring L2.

Figures 5.16 and 5.17 show that after the two leaders join the main conference; they will exchange other sets of messages. These messages are coming from inviting their members to the main conference and then to the sub-conference. Figure 5.16 and Figure 5.17 show that L1 exchanges in total more than what L2 exchange (10505 bytes vs. 8825 bytes). L2 uses more bytes in total because he has two FF members that should invite them to the main conference and then to the sub-conference, while L1 has only one FF member.

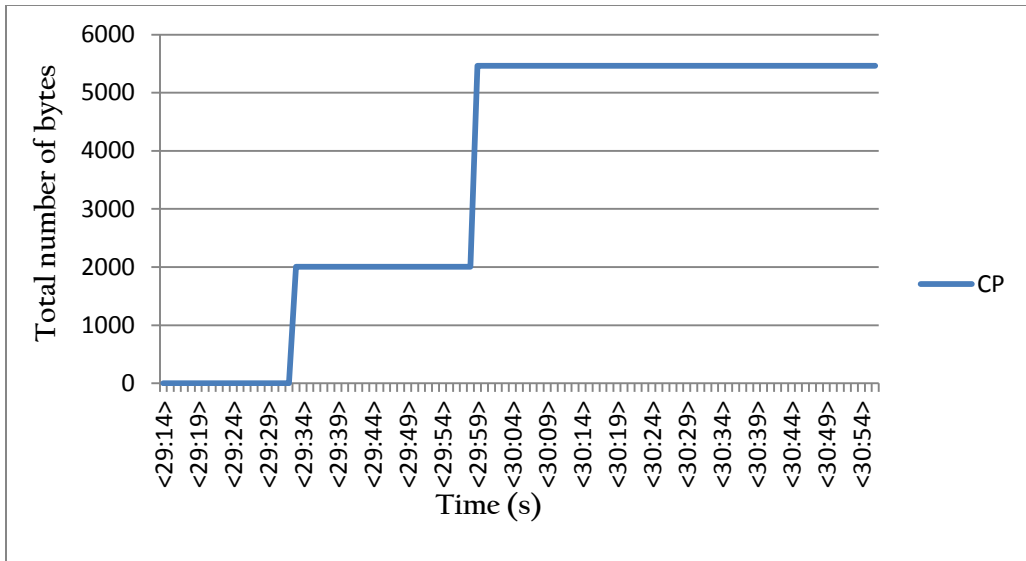


Figure 5.15: Total number of bytes sent/received by the CP

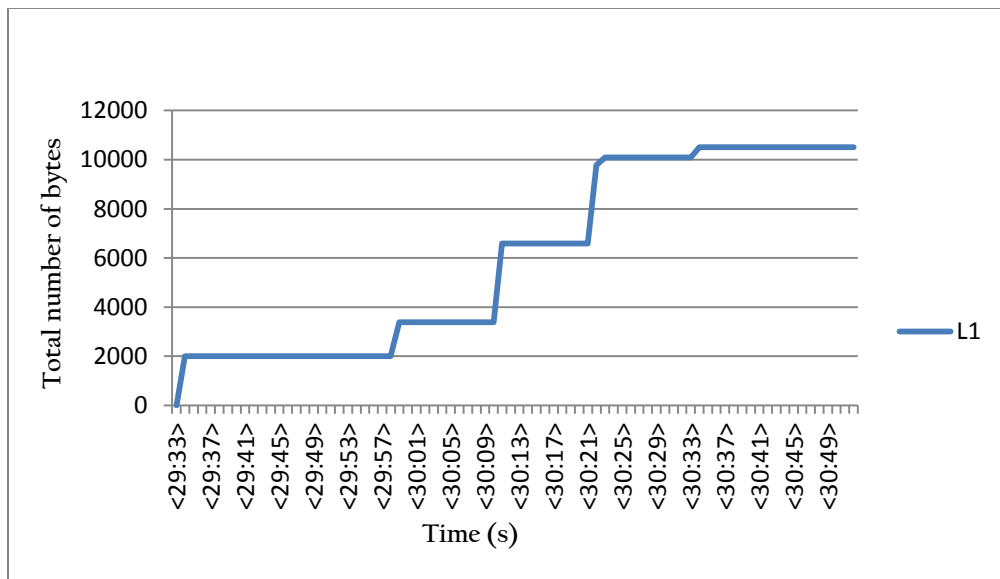


Figure 5.16: Total number of bytes sent/received by L1

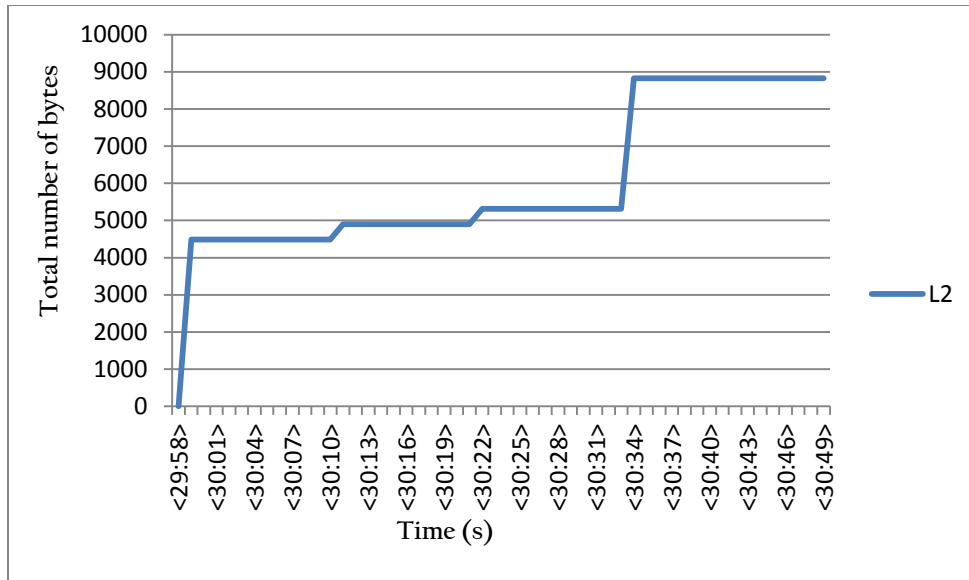


Figure 5.17: Total number of bytes sent/received by L2

**In the sub-conference-M1, M2, and M3 nodes:** Figures 5.18 to 5.20 show that all the members require 2500 bytes to register and to be invited to the main conference and to the sub-conference. They have the same number of bytes because they do the same type of signaling all the time.

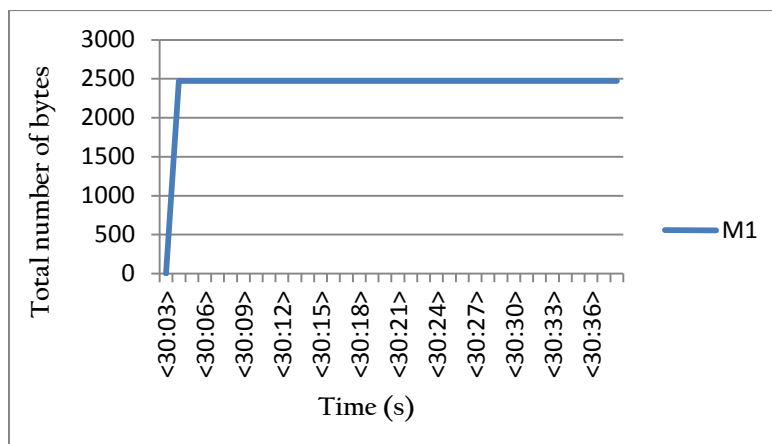


Figure 5.18: Total number of bytes sent/received by M1

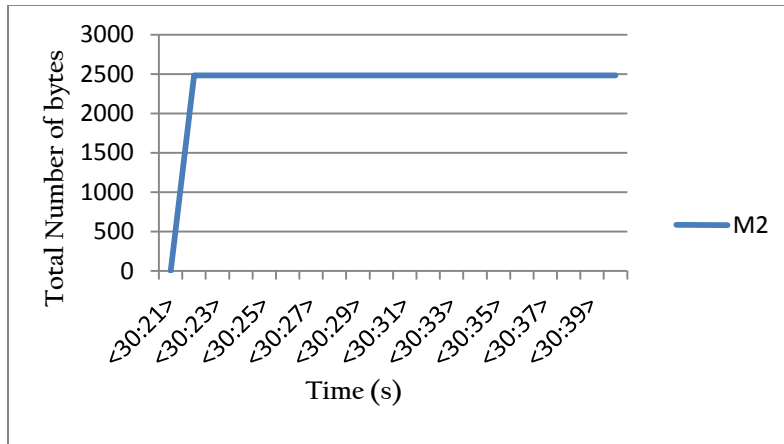


Figure 5.19: Total number of bytes sent/received M2

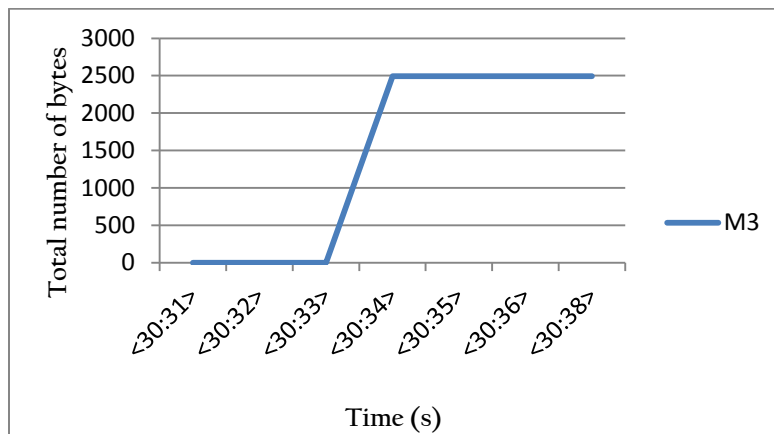


Figure 5.20: Total number of bytes sent/received by M3

Figure 5.19 presents all the FF network load results together. The leader with more members requires more signaling, and thus a larger number of bytes.

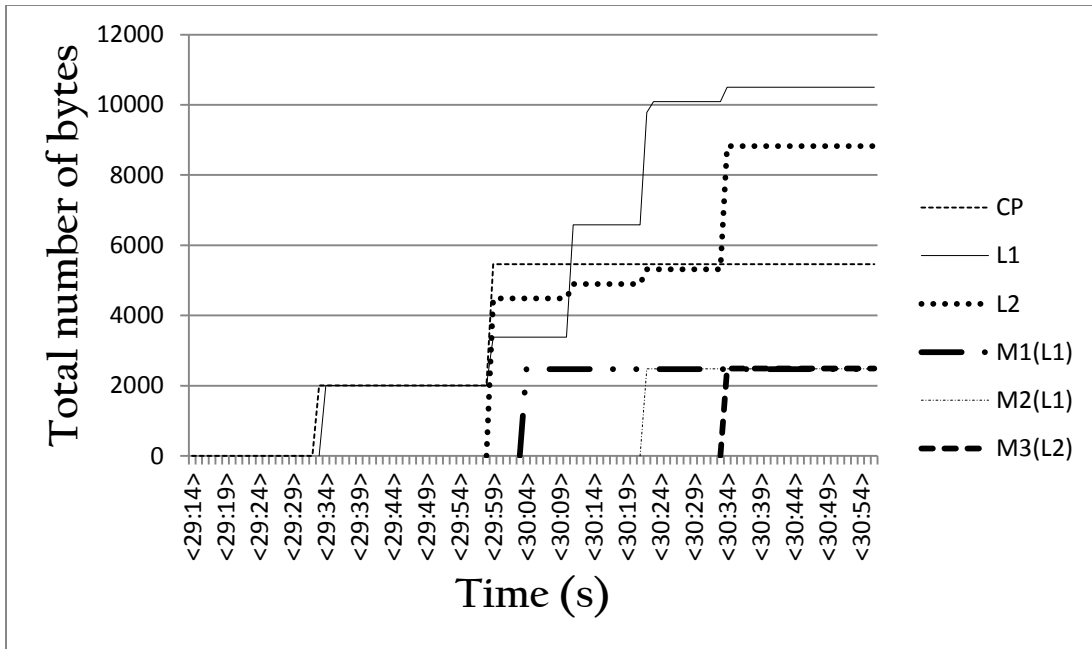


Figure 5.21: Total number of bytes sent/received by all FFs

#### 5.4.2.2. The signaling delays

We measured the total joining delay and the total registration delay in milliseconds. These delays are from the time that the FFs start registering until they are fully joined the main conference/sub-conference. In this section we present the signaling delay for:

- (1) registration to the conference/sub-conferences; and
- (2) joining the conference/sub-conferences.

To measure these delays we used different scenarios. Nine FFs nodes are associated with these scenarios.

**Registration delays**, Figure 5.20 and Table 5.2 show that all FFs nodes (i.e. leaders)

require a range of 145 to 216 ms to register with the CP. The registration delays for the nine nodes are similar, which is a good indication that all the nodes require similar time to register. There was no irregularity shown in the registration process.

Table 5.2: Registration and joining delays for nine leaders

| FF nodes | Registration delay (ms) | Join delay (ms) |
|----------|-------------------------|-----------------|
| L1       | 216                     | 216             |
| L2       | 157                     | 312             |
| L3       | 145                     | 297             |
| L4       | 188                     | 344             |
| L5       | 190                     | 450             |
| L6       | 165                     | 497             |
| L7       | 180                     | 482             |
| L8       | 163                     | 512             |
| L9       | 188                     | 614             |

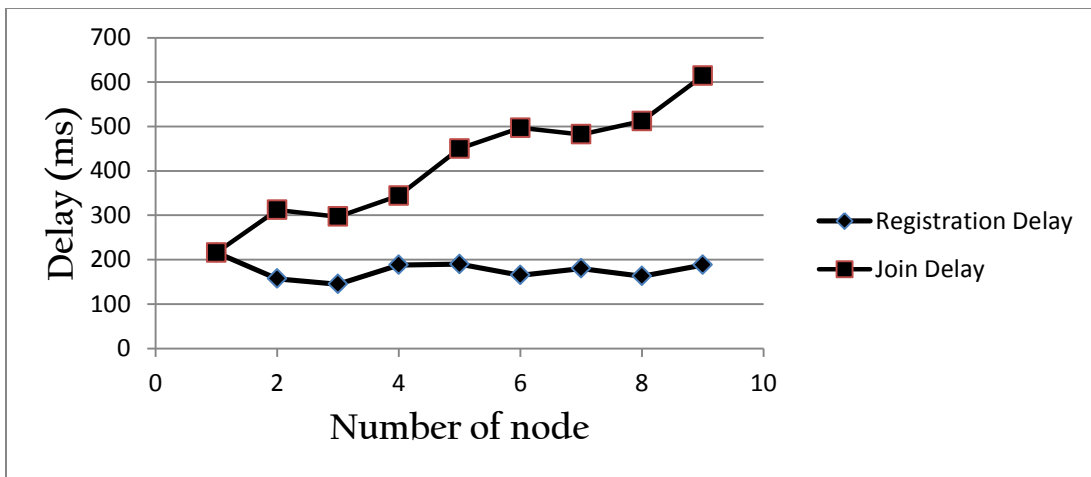


Figure 5.22: The delay in: (1) registration, (2) joining



**Joining delays**, from Figure 5.22, we notice that the delay increases in a linear manner when we have more than three leaders in the main conference. This is due to the fact that we need to REFER the new arrival leaders to the ones who are already in the main conference. The delay here depends on the number of leaders.

Now, we measure the delay to join a sub-conference and compare it to the delay to join the main conference. To do these measurements, four sets of nodes are distributed as follows: (1) Nine-leaders are in the main conference with zero members in the sub-conference (i.e. fully mesh mode); (2) One-leader is in the main conference with eight-members in the sub-conference; (3) Two-leaders are in the main conference and Seven-members are in two sub-conferences; and (4) Three-leaders are in the main conference with six-members are in the sub-conferences.

The measurement data for these distributions are shown in the following tables:

Table 5.3: Nine-leaders with zero members - Full mesh mode

| Node | Delay (ms) | Total delay (ms) |
|------|------------|------------------|
| L1   | 216        | 216              |
| L2   | 312        | 528              |
| L3   | 297        | 825              |
| L4   | 344        | 1169             |
| L5   | 450        | 1619             |
| L6   | 497        | 2116             |
| L7   | 482        | 2598             |
| L8   | 512        | 3110             |
| L9   | 614        | 3724             |

Table 5.4: One-leader with eight-members

| Node | Delay (ms) | Total delay (ms) |
|------|------------|------------------|
| L1   | 167        | 167              |
| M1   | 163        | 330              |
| M2   | 155        | 485              |
| M3   | 191        | 676              |
| M4   | 211        | 887              |
| M5   | 91         | 978              |
| M6   | 179        | 1157             |
| M7   | 261        | 1418             |
| M8   | 136        | 1554             |

Table 5.5: Two-leaders with seven-members

| Node | Delay (ms) | Total delay (ms) |
|------|------------|------------------|
| L1   | 284        | 284              |
| L2   | 353        | 637              |
| M1   | 148        | 785              |
| M2   | 153        | 938              |
| M3   | 151        | 1089             |
| M4   | 172        | 1261             |
| M5   | 147        | 1408             |
| M6   | 162        | 1570             |
| M7   | 144        | 1714             |

Table 5.6: Three-leaders with six-members

| Node | Delay (ms) | Total delay (ms) |
|------|------------|------------------|
| L1   | 277        | 277              |
| L2   | 195        | 472              |
| L3   | 450        | 922              |
| M2   | 170        | 1092             |
| M3   | 162        | 1254             |
| M4   | 172        | 1426             |
| M5   | 141        | 1567             |
| M6   | 165        | 1732             |
| M7   | 154        | 1886             |

When we compare the four distributions, we actually compare the full mesh with clustering. The first set represents a full mesh between the leaders, and the other three sets represent the clustering between the leaders and their members.

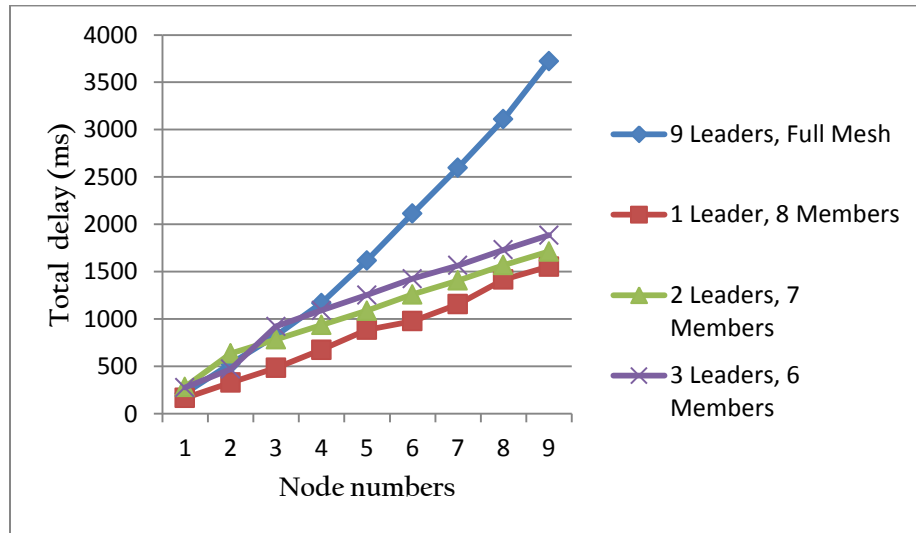


Figure 5.23: Comparisons the total delay between full mesh and clustering

When we compare the full mesh and the other three types of clustering in Figure 5.21, we notice that we have less joining delay in the cluster cases than in the full mesh case. We also have interesting results in set 2 (1 leader, 8 members). In this case, it performs better than in the other cases (less joining delay). There are two explanations for this difference: (1) the total joining delays are measured in milliseconds; Thus, all the measured results are close to each other, and (2) this case has less signaling messages (less delay); in other words, we only have one leader in the main conference. This leader is not required to receive or to invite any other leaders; however, the other two cases (2-leaders and 3-leaders) require sending REFER messages and INVITE messages to the other leaders.

## 5.5 Summary

We implemented a multipart/multimedia conferencing/sub-conferencing for FFs in a standalone MANETs using clustering-based signaling scheme. We used JMF as our media transportation and mixing and JAIN SIP, which is considered the basis for implementing our SIP-based nodes. Performance measurements were made and results were obtained from the prototype. We performed different scenarios to compare our architecture with the full mesh approach.

Through the experiments, we have found that all the scenarios worked well and showed better performance than that found in full mesh approach. In conclusion, we can see that our clustering-based signaling scheme is feasible to accomplish multipart/multimedia conferencing/sub-conferencing for FFs in standalone MANETs.

## CHAPTER 6 CONCLUSIONS AND FUTURE WORK

In this chapter, we first summarize the thesis contributions and then discuss items for future work.

### 6.1 Summary of contributions

The present thesis focuses on the design and evaluation of a new communication system for FFs based on MANETs. The new communication system overcomes current system limitations. It provides a hierarchical communication structure, allowing for more than two FFs to talk at the same time, and for private communication between team members and team leaders. It supports the FFs with new functionalities, along with audio communications, such as, video and floor control. The major contributions of the thesis are as follows:

- **Studied the related work of the FF communication system** - The related work of the FF communication system are reviewed through several interviews with specialized FFs, and also available publications have been studied. The current FF communication system has certain limitations (e.g. flat structure of communications, no private communications, the FF teams need to be in visual contact with each other all the times) and there has not been enough research to help overcome these problems.
- **Investigated a suitable communication environment** - The evaluation and the related work of ad-hoc networks are studied and we found that MANETs are suitable to host the new communication system because these networks work without any pre-

existing infrastructure.

- **Derived the requirements of the new communication system** - The derived requirements are related with current FF communication system and the particularity of MANETs to provide the FFs with a new set of functionalities to facilitate their job such as, audio/video conferencing/sub-conferencing.
- **Reviewed the state-of-the-art of signaling protocols for multimedia conferencing in MANETs** - The state of the art on signaling protocols are reviewed (e.g., SIP, H.323) in the light of our requirements. None of the existing signaling protocols meets all of our requirements. Most of the solutions are designed for infrastructure-based networks. Some of the protocols handle the situations of signaling in infrastructureless; however, it still does not fit our system requirements.
- **Reviewed the state-of-the-art of media handling protocols for multimedia conferencing in MANETs** - The state-of-the-art on media handling protocols are reviewed (e.g., full mesh, centralized) in the light of our requirements. Most of the existing media handling protocols do not meet all of our requirements. One architecture (i.e. DMA) handles most of our requirements.
- **Proposed a cluster-based signaling architecture** - A new signaling architecture to support multimedia/multiparty communication between FFs on the fire ground using MANETs has been presented. The signaling architecture is based on clustering. This architecture meets all of our derived requirements.
- **Adapted DMA to meet our media handling requirements** - We adapted the DMA architecture to fit all of our system requirements. We have presented a new mixer design for both CP and leaders nodes. This mixer can accommodate the FF nodes

with limited capability. The presence of these mixers will help significantly in decreasing the number of media stream in the network.

- **Implemented a proof-of-concept prototype of the proposed architecture** - We have selected SIP as our signaling protocol because SIP is a simple, light-weight, and flexible protocol. For the media transportation and mixing, we have used the real time protocol (RTP), which is provided by the JMF APIs. We have developed a prototype for an audio conferencing/sub-conferencing system. This system includes a CP, any number of FF team leaders, and any number of FF team members. The CP invites the leaders to join the main conference with him, and then each team leader invites his team members to the main conference and then to the sub-conference.
- **Designed a Graphical User Interface (GUI) for the proposed architecture** - A new UI for all the FF nodes are designed. This UI is user friendly and easy to interact with.
- **Evaluated the performance of the implemented prototype** - Different scenarios have been tested and different aspects of the signaling system were measured for the whole architecture. Performance measurements are made and results are obtained from the prototype. We performed different scenarios to compare our architecture with the full mesh approach. Through the experiments, we have found that all the scenarios worked well and showed better performance than that found in full mesh approach.

## **6.2 Limitations and constrains**

In this section, we will present the limitations and constrains of the proposed communication system. The limitations and constrains of the system can be summarized as follows:

- The mobility of the FF team members and the FF leaders has not been addressed in the scope of this research.
- For the realistic validation of the proposed system, a wireless connection needs to be established between FF nodes. However, because of the limited resources at our disposal, the system was validated using wired connections.
- The roles of the different nodes within the communication system (i.e. SSM, SMs, and UAs) are statically fixed to the roles of FFs (i.e. CP, leaders, and members, respectively). This constrain is due to the fact that the roles of each FFs are fixed in the team composition.
- Last but not least, owing to the complexity of dealing with video communication, as compared to audio, video communication has not been considered in the proposed system.

## **6.3 Future work**

This thesis provides a solution for the FF current communication system. The new solution aims at providing multiparty/multimedia communications for the FFs to facilitate their job. This work has been validated using a prototype. However, there are still some issues that have not been addressed, which can be summarized as follows:



- The issue of mobility has not been considered in our architecture. We believe that FFs can do their job better without worrying about signal disconnecting or being out of range. The FF should be able to move to some areas while he is away from his team members. This issue affects the signaling links and the media quality.
- We have shown the important need of having video connection between the FFs in high emergency situations. In our implementation, we introduced audio communication as a solution without the presence of video. However, we believe that FFs should be able to use the video from time to time on the fire ground. Floor control is another related topic that can be linked to the video. Applying floor control to the video will help that the video stream will not consume the entire network bandwidth. The members can for instance use video only when they are allowed to by their leaders.
- Expand our architecture to involve the four first main responders (FF, Police, Medics, and structural engineers) in the same communication network. We believe that having all the first responders collaborating and communicating with one another greatly improves first response efforts.
- In our performance measurements, we have evaluated our architecture with ten nodes. It would be interesting to check the behavior of our signaling and media behavior with a larger number of nodes. However, we could not go for more than 10 nodes because of lack of resources.
- We would like also to try to run our application on hand-held devices running Windows mobile and re-test it with real FFs to assess the shortcomings and to improve our prototype.

## REFERNCES

- [1] Dumoulin, D., Senior Advisor on Emergency Management, Concordia University. Personal Communication with Mr. Dumoulin, 6 October, 2010, darren.dumoulin@concordia.ca.
- [2] Federal Emergency Management Agency (FEMA), “Introduction to Incident Command System (ICS 100)”, available online course, FEMA, <<http://emilms.fema.gov/IS100HE/indexMenu.htm>> November 2012.
- [3] Federal Emergency Management Agency (FEMA), “The National Incident Management System (NIMS) Resource Center”, FEMA, <<http://www.fema.gov/emergency/nims/>> March 2012.
- [4] Bahora, A.S., Collins, T.C., Davis, S.C., Goknur, S.C., Kearns, J.C., Lieu, T.N., Nguyen, T.P., Zeng, J.S., Horowitz, B.M., Patek, S.D., "Integrated Peer-to-Peer Applications for Advanced Emergency Response Systems. Part I. Concept of Operations", Systems and Information Engineering Design Symposium, IEEE, pp. 255-260, 24-25 April 2003.
- [5] James F. Br, “Incident Command System”, Business Recovery Managers Association newsletter, December 2001.
- [6] Ponderosa Fire Department, Harris County Emergency Service District #28, “Incident command operations”, <<http://www.ponderosavfd.org/standard-operating-guidelines/2007/Operational%20PDF/INCIDENT%20COMMAND%20OPERATIONS.pdf>> October 2011.
- [7] Personal Communications with Sain-Mathieu Fire Station, Montreal, QC, Canada,

2011-2012.

- [8] NIOSH Firefighter Radio Communications, <<http://www.cdc.gov/niosh/fire/>>. June 2012.
- [9] Rappaport, T., "AI Gross Remembered", IEEE Communications Society, <<http://www.comsoc.org/socstr/org/operation/awards/remport.html>>, November 2004.
- [10] Basagni, S., Conti, M., Giordano, S., "Mobile Ad hoc Networking with a View of 4G Wireless: Imperatives and Challenges", Mobile Ad Hoc Networking, Wiley-IEEE Press, pp. 1 -45, July 2004.
- [11] Ramanathan, R. and Redi, J., "A Brief Overview of Ad Hoc Networks: Challenges and Directions", IEEE Communication Magazine, 50<sup>th</sup> Anniversary Commemorative Issue, May 2002.
- [12] Aggelou, G., "Mobile Ad Hoc Networks – From Wireless LANs to 4G Networks", McGraw-Hill Companies, Inc., 2005.
- [13] Frodigh, M., Johansson, P., Larsson, P., "Wireless Ad Hoc Networking – The Art of Networking without a Network", Ericsson Review No.4, 2000.
- [14] Xu, B., Hischks, S., and Walke, B., "The Role of Ad Hoc Networking in Future Wireless Communications", Proceedings of International Conference on Communication Technology, Vol. 2, pp. 1353-1358, 2003.
- [15] Luo, H., Meng, X., Ramjee, R., Sinha, P., Li, L., "The Design and Evaluation of Unified Cellular and Ad-Hoc Networks", IEEE Transactions on Mobile Computing, Vol. 6, No. 9, pp. 1060-1074, September 2007.
- [16] Wu, H., Qiao, C., De, S., Tonguz, O., "Integrated Cellular and Ad Hoc Relaying Systems: iCAR", IEEE Journal on Selected Areas in Communications, Vol. 19, No.

- 10, pp. 2105-2115, October 2001.
- [17] Yu, J. Y., and Chong P., “A Survey of Clustering Schemes for Mobile Ad Hoc Networks”, IEEE Communication Surveys and Tutorials, First Quarter 2005V.
- [18] “Conferencing Using the IP Multimedia (IM) Core Network (CN) Subsystem”, 3GPPTS 24.147, June 2005.
- [19] University College London, Computer Science Department, “Introduction to Multimedia Conferencing”, version 1, September 1998.
- [20] Belqasmi, F., Fu, C., “Advance Conferencing-Part 1”, Available online: <[http://users.encs.concordia.ca/~glitho/F10\\_inse7110.htm](http://users.encs.concordia.ca/~glitho/F10_inse7110.htm)>, Fall 2010.
- [21] Khedher, D.B., “Media Handling for Conferencing in MANETs”, Ph.D. Thesis, Concordia University, November 2007.
- [22] J. Rosenberg, “A Framework for Conferencing with the Session Initiation Protocol (SIP)”, IETF RFC 4353 Internet Draft, June 2002.
- [23] Fu, C., “Signaling for Conferencing in Mobile Ad-Hoc Networks”, Ph.D. Thesis, Concordia University, March 2008.
- [24] Schulzrinne, H., Rosenberg, J., “Signaling for Internet Telephony”, Proceeding 6th International Conferencing Network Protocol, pp. 298-307, 1998.
- [25] H.323 Series Recommendations, Implementers Guide for ITU-T H.323 Systems, <[https://www.itu.int/itudoc/itu-t/com16/implgd/h323ig\\_ww9.doc](https://www.itu.int/itudoc/itu-t/com16/implgd/h323ig_ww9.doc)>, May 2005.
- [26] H.323 Series, ITU-T Recommendations, Geneva, 2003.
- [27] Internet Engineering Task Force (IETF), <<https://datatracker.ietf.org/doc/>>, March 2012.
- [28] Rosenberg, J., et al., “SIP: Session Initiation Protocol”, RFC 3261,

- < <http://www.ietf.org/rfc/rfc3261.txt>>, June 2002.
- [29] Sparks, R., “The Session Initiation Protocol (SIP) Refer Method”, RFC 3515. April 2003.
- [30] Donovan, S., “The SIP INFO Method”, RFC 2976, October 2000.
- [31] Simple Conference Control Protocol (SCCP), IETF Internet Draft, < <http://tools.ietf.org/html/draft-ietf-mmusic-sccp-00>>, June, 2012.
- [32] A Framework for Conferencing with the Session Initiation Protocol (SIP), IETF RFC 4353, < <http://tools.ietf.org/html/rfc4353> >, April 2012.
- [33] Rosenberg, J., Schulzrinne, H., “Models for Multi-Party Conferencing in SIP”, IETF Internet Draft, November 2000.
- [34] Kelly, K., “Distributed Multipoint Conferences using SIP”, IETF Internet Draft, < <http://tools.ietf.org/html/draft-mark-sip-dmcs-00> >, April 2012.
- [35] Khlifi, H., Agarwal, A., Gregoire, J-C., “A Framework to Use SIP in Ad-Hoc Networks”, Canadian Conference on Electrical and Computer Engineering, IEEE CCECE 2003, pp. 985-988, vol. 2, 4-7 May 2003.
- [36] Koskelainen, P., Schulzrinne, H., Wu, X., “A SIP-based Conference Control Framework”, ACM Proceedings of the 12<sup>th</sup> International Workshop on Network and Operating System Support for Digital Audio and Video, pp. 53-61, May 2002.
- [37] Fu, C., Glitho, R., Khendek, F., “Signaling for Multimedia Conferencing in Stand-Alone Mobile Ad-Hoc Networks”, IEEE Transactions on Mobile Computing, Vol. 8, No. 7, pp. 991-1005, July 2009.
- [38] Chen, A., Tsai, M., Lantz Jr., T.S., T., Plans, A., Mathur, S., Lakhera, S., Kaushik, N., and Pena-Mora, F., “A Collaborative Framework for Supporting Civil

- Engineering Emergency Response with Mobile Ad-Hoc Networks”, *Journal of Computing in Civil Engineering*, ASCE, pp. 575-582, 2007.
- [39] Manoj, B.S., Baker, A.H., “Communication Challenges in Emergency Response”, *Communications of the ACM*, Vol. 50, No. 3, pp. 51-53, March 2007.
- [40] Smith, P.C., Simpson, D.M, “The Role of Mobile Emergency Tactical Communication Systems for Disaster Response”, Working Paper of the Center for Hazards Research and Policy Development (US House of Representatives), Vol. 6, No. 2, pp. 94-106, 5-6 June 2005.
- [41] Hooper, R., Orgen, B., Hankin, N., Williams, J., “Current Status, Knowledge Gaps, and Research Needs Pertaining to Firefighter Radio Communication Systems”, National Institute for Occupational Safety and Health, Final Report, September 2003.
- [42] Al Rubaye, M., Belqasmi, F., Fu, C., Glitho, R., “A Novel Architecture for Floor Control in the IP Multimedia Subsystem of 3G Networks”, *IEEE 69th Vehicular Technology Conference, VTC*, pp. 1-5, Spring 2009.
- [43] Ouzzif, M., Erradi, M., Mountassir, H., “Description of A Teleconferencing Floor Control Protocol and Its Implementation”, *Engineering Applications of Artificial Intelligence*, Vol. 21, No. 3, pp. 430-441, April 2008.
- [44] Basu, P., Khan, N., Little, T.D.C., "A Mobility Based Metric for Clustering in Mobile Ad-Hoc Networks", *International Conference on Distributed Computing Systems Workshop*, pp. 413-418, April 2001.
- [45] Peña-Mora, F., Chen, A.Y., Aziz, Z., Soibelman, S., Liu, L.Y., El-Rayes, K., Arboleda, A.R., Lantz Jr., T.S., Plans, A.P., Lakhera, S., and Mathur, S., “Mobile Ad Hoc Network-Enabled Collaboration Framework Supporting Civil Engineering

- Emergency Response Operations”, *Journal of Computing in Civil Engineering*, ASCE, Vol. 24, No. 3, pp. 203-212, May-June 2010.
- [46] Schulzrinne, H., et al., “RTP A Transport Protocol for Real-Time Applications”, IETF RFC 3550, July 2003.
- [47] Rangan, P.V., Vin, H.M., Ramanathan, S., “Communication Architectures and Algorithms for Media Mixing in Multimedia Conferencing”, *IEEE/ACM Transactions on Networking*, Vol. 1, pp. 20-30, February 1993.
- [48] Yang, S., Yu, S., Zhou, J., Han, Q., “Multipoint Communication with Speech Mixing Over IP Network”, *Computer Communications*, Vol. 25, pp. 46-55, January 2002.
- [49] Radenkovic, M., “A Framework for Building and Deploying the Multiparty Audio Service for Collaborative Environments”, *Presence: Teleoperators and Virtual Environments*, Special Issue, VRST ACM 2002, vol. 13, No. 6, pp. 708–25, December 2004.
- [50] Guerrieri, J.R., Francis, M.H., Wilson, P.F., Kos, T., Miller, L.E., Bryner, N.P., Stroup, D.W., Klein-Berndt, L., "RFID-Assisted Indoor Localization and Communication for First Responders," *First European Conference on Antennas and Propagation (EuCAP)*, pp. 1-6, 6-10, November 2006.
- [51] Khedher, D.B., Glitho, R., Dssouli, R., “Media Handling for Multiparty Sessions in Ad-Hoc Peer-to-Peer Networks: A Novel Distributed Approach”, *IEEE International Symposium on Computer Communications (ISCC)*, pp. 131-136, 2005.
- [52] Khedher, D.B., Glitho, R., Dssouli, R., “Media Handling Aspects of Multimedia Conferencing in Broadband Wireless Ad-Hoc Networks”, *IEEE Network*, Vol. 20,

No. 2, pp. 42–49, March-April 2006.

- [53] Aldunate, R., Ochoa, F., S., Peña-Mora, F., Nussbaum, M., “Robust Mobile Ad Hoc Space for Collaboration to Support Disaster Relief Efforts Involving Critical Physical Infrastructure”, *Journal of Computing in Civil Engineering*, ASCE, January-February 2006.
- [54] Holmberg, D., Davis, W., Treado, S., Reed K., "Building Tactical Information System for Public Safety Officials, Intelligent Building Response (iBR)", NISTIR 7314, <<http://www.fire.nist.gov/bfrlpubs/fire06/PDF/f06044.pdf>>, July 2012.
- [55] Camarillo, G., Niemi, A., Isomaki, M., Garcia-Martin, M., Khartabil, H., “Referring to Multiple Resources in the Session Initiation Protocol (SIP)”, IETF Internet Draft, December 2007.
- [56] JAIN SIP Web Page Online, <<http://jsip.java.net/>>, November 2011.
- [57] An Introduction to the JAIN SIP API: <<http://www.oracle.com/technetwork/articles/entarch/introduction-jain-sip-090386.html>>, November 2011.
- [58] JMF Home Page, <<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>>, March 2012.
- [59] Rosenberg, J., “A Hitchhiker's Guide to the Session Initiation Protocol (SIP)”, IETF RFC 5411 Internet Draft, November 2008.



## APPENDIX A. XML files for FF nodes

### (a) Example of CP XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<CPConfig>
  <Name>CP</Name>
  <IP>.</IP>
  <Port>3000</Port>
  <LogPath>CP.log</LogPath>
  <CPMulticastIP>228.5.6.7</CPMulticastIP>
  <CPMulticastPort>5000</CPMulticastPort>
  <LeaderList>
    <Leader>Leader1</Leader>
    <Leader>Leader2</Leader>
    <Leader>Leader3</Leader>
  </LeaderList>
</CPConfig>
```

### (b) Example of Leader XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<LeaderConfig>
  <Name>Leader1</Name>
  <IP>.</IP>
  <Port>4000</Port>
  <LogPath>Leader1.log</LogPath>
  <LeadersMulticastIP>228.5.6.7</LeadersMulticastIP>
  <LeadersMulticastPort>5001</LeadersMulticastPort>
  <CPMulticastIP>228.5.6.7</CPMulticastIP>
  <CPMulticastPort>5000</CPMulticastPort>
  <MemberList>
    <Member>Member11</Member>
    <Member>Member12</Member>
  </MemberList>
</LeaderConfig>
```

(c) Example of Member XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<MemberConfig>
  <Name>Member1</Name>
  <IP>.</IP>
  <Port>5000</Port>
  <LogPath>Member1.log</LogPath>
  <LeadersMulticastIP>228.5.6.7</LeadersMulticastIP>
  <LeadersMulticastPort>5001</LeadersMulticastPort>
</MemberConfig>
```

## APPENDIX B. Example of a REFER message with multiple targets

```
REFER sip:conf-123@example.com;gruu;opaque=hha9s8d-999a SIP/2.0
Via: SIP/2.0/TCP client.chicago.example.com
    ;branch=z9hG4bKhjhs8ass83
Max-Forwards: 70
To: "Conference 123" <sip:conf-123@example.com>
From: CP <sip:CP@Concordia.example.com>;tag=32331
Call-ID: d432fa84b4c76e66710
CSeq: 2 REFER
Contact: <sip:CP@client.Concordia.example.com>
Refer-To: <cid:cn35t8jf02@example.com>
Refer-Sub: false
Require: multiple-refer, norefersub
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE,
NOTIFY
Allow-Events: dialog
Accept: application/sdp, message/sipfrag
Content-Type: application/resource-lists+xml
Content-Disposition: recipient-list
Content-Length: 385.0
Content-ID: <cn123@example.com>
<?xml version="1.0" encoding="UTF-8"?>
  <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <list>
      <entry uri="sip:Leader1@example.com?method=INVITE" />
      <entry uri="sip:Leader2@example.org?method=INVITE" />
    </list>
  </resource-lists>
```