Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol 1, No.1, 2011

Signaling For Multimedia Conferencing in Stand-Alone Mobile Ad Hoc Networks

Sudhakar Krishnan
Department of Computer Science & Engineering, Sengunthar College of Engineering, Namakkal, Tamilnadu, India.
ksudhakar.cs@gmail.com

Rubini Pandu
Department of Information Technology, Gnanamani College of Technology, Namakkal, Tamilnadu, India.
ruby_nila@yahoo.co.in

Abstract

Mobile ad hoc networks (MANETs) are infrastructure-less and can be set up anywhere, anytime. They can host a wide range of applications in rescue operations, military, private, and commercial settings. Multimedia conferencing is the basis of a wealth of “killer” applications that can be deployed in MANETs. Some examples are audio/video conferencing, multiplayer games, and online public debating. Signaling is the nerve center of multimedia conferences—it establishes, modifies, and tears down conferences. This paper focuses on signaling for multimedia conferences in MANETs. We review the state of the art and propose a novel architecture based on application-level clusters. Our validation employed SIP as the implementation technology and OPNET as our simulation tool. Our clusters are constructed dynamically and the nodes that act as cluster heads are elected based on their capabilities. The capabilities are published and discovered using a simple application-level protocol. The architectural principles and the clustering operations are discussed. Our SIP-based implementation is also presented along with the performance evaluation.

Keywords: MANET, SIP-technology, OPNET-simulation tool, cluster

1. INTRODUCTION

MOBILE ad hoc networks (MANETs) can be defined as transient networks formed dynamically by a collection of arbitrarily located wireless mobile nodes, without the use of existing network infrastructure or centralized administration [J. Liu and I. Chlamtac, (July 2004)]. They rely on wireless technologies, such as IEEE 802.11 and Bluetooth, and are either stand-alone or connected to other networks. This paper focuses on standalone MANETs. MANETs can host a wide range of applications in rescue operation, military, private, and commercial settings.

Many of these applications involve multimedia conferencing. One example is an audio/video conference in a rescue operation setting. After a natural disaster, there may be no communications infrastructure. A conference in such an environment will allow first aid squads, ambulance services, police, and other involved parties to communicate in order to facilitate coordination. Audio/video conferences in infra structure less environments are also commonplace in military settings.

In private settings, such as airports and university campuses, multiparty games can be contemplated. Furthermore, in commercial settings such as multihop cellular networks, network operators may, for performance reasons, consider decentralizing the execution of conferencing services in the MANET portion of the multichip cellular network whenever all of the conference participants are in that portion

A multimedia conference (multiparty sessions) can be defined as a conversational exchange of multimedia content between several parties. Some examples are audio/video conferencing, massively multiparty gaming, and debating. Two main components make up multimedia conferencing: signaling and media handling. Media handling deals with media transportation, mixing, and transcoding. Signaling is the nerve center of multimedia conferencing. It enables the initiation, modification, and teardown of conferences by establishing, controlling, and ending the signaling connections between participants. Signaling architectures for conferencing in infrastructure-based networks have been extensively discussed in the literature. Classical examples are H.323 [2] and SIP XCON [3]. Unfortunately, these architectures are highly inadequate for MANETs.

This paper is devoted to signaling for multimedia conferencing in stand-alone MANETs. It identifies the challenges, discusses the drawbacks of the existing architectures, and proposes and validates a new architecture. This architecture relies on application-level clusters and is an extension of the cluster-based signaling system [C. Fu, R.H. Glitho, and R. Dssouli, (Mar. 2005)] that we have proposed in previous work.

The cluster heads are elected based on their capabilities. The cluster size is based on both the capabilities of the cluster head and the cluster split value. Our clusters are significantly different from routing layer clusters, such as those discussed in [J.Y. Yu and P.H.J. Chong,(Mar. 2005)]. Routing level clusters are formed using the physical or geographical location of nodes, and they have no relation to whether or not there is a conference at the application layer. Our signaling clusters are formed only when there is a conference, and thus, they are completely independent of the routing layer clusters.

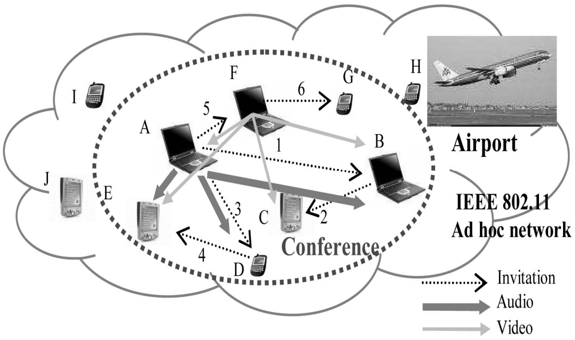

Fig.1

We have selected Session Initiation Protocol (SIP) [J. Rosenberg et al., (June 2002)] as the implementation technology. The SIP extensions are described and the deployment techniques are discussed. We also evaluated the performance of our architecture through simulation using OPNET. The rest of the paper is organized as follows: In Section 2, we present an application scenario and derive the requirements. In Section 3, we evaluate the related work. Section 4 is devoted to the architecture. In Section 5, we discuss the implementation. Performance evaluation is covered in Section 6, and we conclude in the last section.

2. APPLICATION SCENARIO AND REQUIREMENTS

Conferencing enables a broad range of applications in MANETs. These applications may comprise different media types and involve different numbers of participants. They may be public or private—for preselected members only. They may be prearranged or established in an ad hoc manner. Fig. 1 illustrates an ad hoc conference scenario in an airport. Passengers waiting for their flights wish to play a multiparty multimedia game using their handheld devices (laptop, PDA, or cell phone). We assume that the game is preloaded on their devices, which are connected through IEEE 802.11 wireless cards. Passenger A starts the game by inviting Passenger B and they establish an audio session. Then, Passenger B invites Passenger C to join the session, and then informs Passenger A that C has joined. Passengers D, E, F, and G are invited to join the game and a seven-party game is established. Various media are exchanged among the parties. In this scenario, the signaling system is responsible for establishing sessions, negotiating media types, and propagating participant information. These functions, well-understood and established for infrastructure-based networks, become very challenging for MANETs. We therefore derive six requirements for signaling in MANETs.

Because MANETs are infrastructureless, the first requirement is that none of the involved functional entities can be permanently centralized. The second requirement stipulates that the system should be able to dynamically propagate conference-related information (e.g., who joins, who leaves) to all of the involved parties. This is no easy task, as conferences are normally very dynamic in MANETs. Parties can join and leave at any time.

A party may leave the conference when it decides to do so or when it is forced to because it has moved out of the coverage area or its battery power is used up. In this paper, we term the first case (which may occur in any network) “voluntary departure” and the second (which is specific to MANETs) “unintentional departure.” A signaling system for multimedia conferencing in MANETs should handle both situations gracefully. The third requirement is scalability. A varying level of scalability is expected from MANETs, depending on the different application scenarios. The signaling system should scale automatically, and in the same manner as the network in which it is implemented. MANETs are made up of heterogeneous nodes. Some nodes may have a high level of resource (e.g., processing power, memory), while others may have very limited capabilities, a situation that leads to two additional requirements. The system should be lightweight, a prerequisite that accommodates nodes with limited resources, and the use of the resources available to the system should be optimal.

The last requirement is independence from the lower layer protocols. There is a plethora of lower layer routing protocols in MANETs. The signaling system is at the application level and it cannot afford to rely on the features of specific lower layer protocols because these lower layer protocols may not be available in some environments.

3. RELATED WORKS

Seminal work has been done on signaling for multimedia conferencing in infrastructure-based networks by standard bodies (i.e., ITU-T and IETF). This work has triggered further research on the same topic independently of these organizations. In this section, we first review approaches proposed for infrastructure-based networks, and then review approaches that take into account some of the specifics of MANETs.

3.1 Signaling for Conferencing in Infrastructure-Based Networks

3.2 Signaling Protocols from Standards Bodies

Multimedia conferencing over packet-switched networks is specified by the ITU-T, as a subset of the H.323 series of recommendations [H.323 Series,(2003)]. H.323 defines four entities: terminal, gateway, gatekeeper, and Multipoint Control Unit (MCU). Multimedia conference control in H.323 is done via the MCU. The MCU can be divided into two entities: the multipoint controller (MC) and the multipoint processor (MP). The MC is devoted to signaling while the MP handles media. H.323 defines three conference models: centralized, distributed, and mixed. Each of these models requires a centralized MC. As an infrastructure-based protocol, H.323 does not meet most of our requirements. It is complex and heavy.

A rather early work in IETF, the Connection Control Protocol (CCP) is based on multicast. It considers only one conference scenario, i.e., the scenario in which the conference initiator creates a conference. Schooler and Casner [E.M. Schooler and S.L. Casner, (1992)] give an introduction. If we review the CCP in terms of our requirements, we find that there is no permanent centralized entity, but the CCP does not meet the other requirements. For example, it relies on a network layer that supports multicast, while our requirements stipulate independence from the lower layers.

The IETF has been working on new approaches since the early 2000s, as part of the work on SIP. SIP is a set of specifications, which includes a baseline specification [J. Rosenberg et al., (June 2002)] and many extensions. It defines four entities: User Agent (UA), proxy server, location server, and registrar. A session control function is located in the user agent. SIP servers are non mandatory entities that help to route SIP messages and locate SIP user agents. SIP is lightweight and extendible and it has been used for two different conference models, tightly and loosely coupled, in infrastructure-based networks. A loosely coupled conference is based on multicast. The IETF draft [C. Bormann, J. Ott, and C. Reichert,(Dec. 1996)] describes Simple Conference Control Protocol (SCCP), a loosely coupled conference control protocol that uses SIP as the signaling protocol. The signaling architecture is centralized. Signaling messages are exchanged between a controller and a participant through multicast. If we evaluate SIP in SCCP, it cannot meet most of the requirements, e.g., there is a permanent central point and multicast is required in the network layer. A tightly coupled conference is central-server-based. Rosenberg [J. Rosenberg, (Feb. 2006)] defines the SIP usage in this sort of conference model. SIP creates sessions between each participant and a conference focus (i.e., the central server).
This conference model cannot meet our first requirement because it has a permanently central entity.

3.3 Proposals outside Standard Bodies

The framework proposed in [J. Rosenberg, (Feb. 2006)] is still another work deploying SIP for conferencing. It extends the tightly coupled conference of SIP. Multiple conference focuses are proposed and each focus manages a set of local participants. The conference focuses are interconnected and form a tree structure. This work extends the scale of the work defined in [J. Rosenberg, (Feb. 2006)], but does not solve other issues. For example, there are still permanent central entities in this framework. Global MMCS [H. Bulut, A. Uyar, and G.C. Fox, (July 2005)] is designed to bridge H.323, SIP, Access Grid clients, and 2.5G/3G cellular phones in audiovisual collaborative applications. The system makes use of a publication-/subscription-based message delivery middleware, the Narada Broking overlay network. As far as multimedia conferencing is concerned, the system borrows the ideas of MCU from H.323. However, unlike H.323, the MCs can be distributed. Although this distribution brings more scalability, our first requirement is still not met because each MC is a permanently centralized entity. ICEBERG signaling [H.J. Wang et al., (Aug. 2000)] proposes a signaling system for management of dynamic and multi device multiparty sessions. Unlike the other signaling protocols such as SIP, it is a signaling system that is specifically designed for multimedia conferencing. It does not have a permanently central point but defines “call sessions” that are responsible for signaling control. A “call session” is only created when there is a conference. The call sessions dynamically propagate information to participants. Thus, the first and second of our requirements are met. However, the system does not take into account device heterogeneity, does not handle unintentional departures of participants, and relies on the routing layer multicast.

3.4 Signaling for Conferencing in Infrastructure less Networks

Mark and Kelley [H.323 Series,(2003)] describe an SIP application for a fully distributed conference model. In the model, each participant maintains an SIP session with other participants. The approach is of special interest to us because it only involves an SIP end system (UA) and no server is required. Thus, the first requirement—no permanent central point—is met. In addition, the approach is lightweight because there is no extra control message added to the baseline SIP but the session related information is carried by the basic SIP messages.

However, this approach has several drawbacks, one of which is how the session-related information is propagated. There is a problem when two (or more) parties are invited to join an ongoing conference at the same time. There is no general solution to ensure that each of the invited parties is made aware of the other invited parties. The problem is identified in [H.323 Series,(2003)] as the “coincident join,” and no solution is provided. Another drawback is that unintentional departures are not considered and are not handled gracefully. Also, the approach does not scale because the number of signaling connections increases exponentially with the number of participants in the conference. The framework defined in [H.J. Wang et al., (Aug. 2000)] is the first work that we found for conferencing in MANETs. It is also an SIP-based solution and it has solved the “coincident join” problem identified in [H.J. Wang et al., (Aug. 2000)]. It proposes a conference leader that maintains conference states.

The leader is responsible for propagating session-related information to every other participant in the conference. However, the approach is fully distributed in terms of signaling architecture, i.e., every participant maintains a session with every other party. Thus, related drawbacks such as limited scalability are issues with this approach. In addition, this framework does not consider the unintentional departure of participants and the optimal use of resources. In [C. Fu, R.H. Glitho, and R. Dssouli, (Mar. 2005)], we proposed an early version of cluster-based signaling architecture for multiparty sessions in MANETs. This architecture potentially meets most of the derived signaling requirements. However, the requirements for handling the unintentional departure of nodes and the independence of lower layer routing protocols were not addressed.

In that work, the general principles of the cluster schemes were presented, but no detailed description was provided and some issues were not addressed. For example, we specified that a cluster head is elected based on the resource level of conference participants, but there is no description of how a participant knows the resource levels of other parties. As another example, a cluster may split if its size reaches the split value, but what happens if a cluster head does not have enough capability to handle participants? In addition to these issues, how the cluster scheme solves the “coincident join” and other problems such as “coincident cluster head leaving” have not been discussed. We selected SIP as the implementation technology and built a prototype in [C. Fu, R.H. Glitho, and R. Dssouli, (Mar. 2005)], but the issue of SIP deployment in MANETs was not fully addressed. The deployment of SIP in MANETs includes two sub issues, i.e., how to discover other SIP end points in the network and how to route an SIP message to its correct destination. In this paper, we address all of the aforementioned disuses and provide a comprehensive signaling solution for conferencing in MANETs.

4. CLUSTER-BASED SIGNALING ARCHITECTURE

Clusters enable scalability without centralization and we believe that they can aid in meeting all of the requirements. The overall architectural principles are presented first,
followed by a description of the clusters’ operational procedures. We then discuss the critical issues related to the operational procedures.

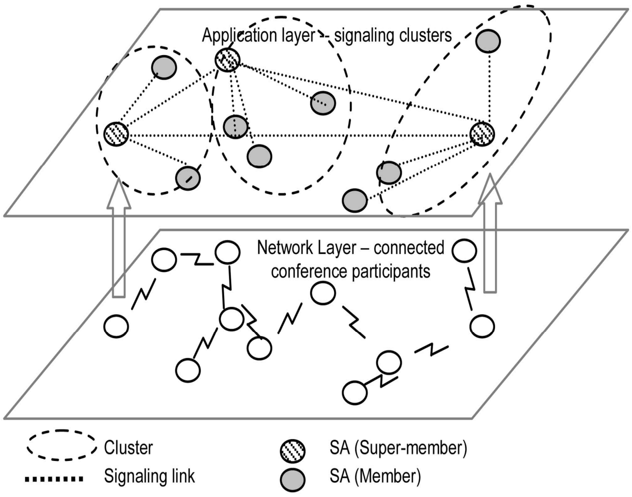

Fig.2 Overall Architecture and Principles

Fig2. gives an overall view of the proposed cluster-based architecture. The only functional entity is the signaling agent (SA). There is one per party, or more generally, one per node in a stand-alone MANET. They are grouped in clusters that we call signaling clusters. These signaling clusters are application-level clusters that are independent of lower layer clusters such as routing clusters. In each cluster, at any given time, there is one and only one cluster head (we call it the super member), and all the other members of the cluster are connected to it. All super members have direct links to the super members of the neighboring clusters. There are two general parameters of a cluster: Split value (Sv) and Merge value (Mv). Every node in a conference maintains the same Sv and Mv. If the sizeof a cluster reaches Sv, the cluster will split into two clusters. If it reaches Mv, the cluster will find another cluster to merge with.

A super member is responsible for keeping track of the information of its members and its neighboring super members. It also propagates the information when there is a change in membership. In addition, it detects the eventual unintentional departures of the nodes connected to it by sending periodic heartbeat messages to them. In our architecture, it is the node with the most capabilities that is elected as the super member. In this paper, the capability of an MANET node can be any attribute usually considered as node capability in the literature (e.g., battery power, memory capacity, and processing power). However, we do not specify either the factors used to compute the capability or how to do the computation. We assume, for the sake of simplicity, that a node always “knows” its capability. We also assume that this capability is represented by a numerical value. A participant that initiates a conference is responsible for collecting the capability of the called party before the conference is initiated. Super members keep track of the capability changes of their members and neighboring super members during the conference.

4.1 Operational Procedures of Clusters

In our architecture, clusters are dynamically created and deleted when conferencing. The signaling system is responsible for maintaining the state of the conference and the clusters. Each signaling cluster has a life cycle. The first phase is its creation when a supermember is elected. After its creation, the cluster moves to the active phase. The cluster membership evolves (parties join and leave). These changes may lead the cluster to split in two, or to merge with another cluster. Ongoing activity may also lead to the election of a new supermember, necessitated by the departure of the supermember, for example. The life cycle ends with the deletion of the cluster. In this section, we describe the signaling procedures related to each of the phases of the cluster life cycle.

4.2 Creation and Deletion

The first cluster is created when the conference starts. The creation procedure is as follows: first, the party (called the initiator) that wishes to establish a session collects the capability of the called party. It compares the capabilities of itself and the called party and designates the one with more capability as the super member. Second, it requests the super member (itself or the called party) to create a session. The initiator needs to set the Sv and Mv and passes the parameters to the called party. After the first session is set up, the super member starts to periodically collect the capabilities of its members. The last cluster is deleted when the last two parties leave the session. All the states and parameters of the cluster are then cleared.

4.3 Changes in Cluster Membership

Both members and super members can invite parties to join a conference. If it is a supermember inviting and the supermember is capable of handling more members, the supermember directly establishes a session with the party. If the supermember cannot handle more members, it may ask a neighboring supermember to do so. If a member invites a party, that member asks its supermember to establish the session. A new member is then added to the cluster. The supermember of the cluster propagates the membership change to the neighboring clusters. Any participants, including members and supermembers, may leave a conference whenever they want to. In the case of a member departure, the member terminates its connection with its supermember and the supermember propagates the membership change to the neighboring clusters. With the departure of a supermember, that supermember designates a new supermember (choosing the member with most capability among the member list) before leaving. It passes its member list and neighboring supermember list to the new supermember. The new supermember sets up a session with each member and each neighboring supermember and forms a new cluster. After this procedure, the old supermember terminates all of its connected sessions. In the situation where there is nomember in a cluster, the supermember that wishes to leave simply terminates all of its connected sessions. Frequent changes of clusters' supermembers can lead to instability. The clustering algorithm described in [E.M. Schooler and S.L. Casner, (1992)] has proposed rules to make sure that the cluster head changes as infrequently as possible. Similarly, we set a rule that when a party joins a cluster, it does not replace the supermember, even if it has more capability. This helps to maintain the stability of the clusters, but does not prevent the supermember from leaving if it decides to do so.

4.4 Splitting and Merging

When a new member is added to a cluster, the supermember initiates the split procedure if the size of the cluster has reached Sv or if the supermember does not have enough capability to handle more members. This happens, for instance, when the battery power of the supermember decreases. First, the supermember selects a new supermember, based on capabilities. It also selects half of its members that are to become members of the new cluster. The selection may be random or according to certain rules, such as the members with higher addresses. It then asks the new supermember to form a new cluster that contains the selected members, passing the selected member list and neighboring supermember list to the new supermember. The new supermember creates a new cluster by establishing sessions. The supermember then terminates sessions with the selected members. Fig. 3 shows a signaling architecture before and after splitting.

If the size of the cluster diminishes to the Mv, the supermember initiates a merger procedure. This procedure consists of searching for an existing cluster with which to merge, with the constraint that the size after the merger will be less than Sv. A new supermember is elected as soon as the merger begins. The new supermember will be one of the two supermembers (the one with more capability) of the two clusters. The procedure continues as follows: the elected supermember establishes sessions with the members of the cluster to merge with. The unelected supermember then terminates sessions with its members and sets the elected supermember as its supermember, and it becomes a regular member. The merger information will then be propagated to the neighboring supermembers. Setting the Sv is critical for the signaling system. For example, if the Sv is too small, the system may exhibit poor performance due to frequent splitting. If the Sv is too large, the signaling will have a centralized structure. On the one hand, we expect a minimum number of clusters so that the overall signaling overhead is low. On the other hand, a centralized architecture is unacceptable for a large-scale conference

4.5 Super member Election

An election algorithm is used whenever there is a need to select a new super member among several candidates. The basic rule is that the candidate with the most capability is selected as a super member. We assume that a party that wishes to elect a new super member knows a list of candidate's fcandidate1 . . . candidateng, where n is the number of candidates. The respective capabilities of these candidates are fCap1 . . . Capng. The algorithm is described below:

```
Capmax Cap1
loop i (from 2 to n)
    if (Capmax is less than Capi)
        Capmax Capi
        max i
    end
end
super member candidatemax
```

4.6 Information Propagation

In order to maintain a signaling cluster system, efficient information propagation is very important, i.e., rapid propagation with as little introduced overhead as possible. In our architecture, super members are responsible for propagating membership and capability information whenever there is a change. The information can be propagated to all the signaling agents in no more than two hops. Fig. 2 shows an example. An option that can further reduce propagation overhead is that only part of the information is propagated, and only to super members. This is realistic because it is not necessary for every member to have knowledge of every system change, e.g., a splitting of a cluster.

4.7 Specific Issue Discussion-Coincident Behaviors

One issue of a distributed signaling architecture is the state synchronization when there are coincident behaviors of participants in the conference. Such behaviors may cause inconsistent states among participants, e.g., with a coincident joining (defined in [P. Koskelainen, H. Schulzrinne, and X. Wu, (May 2002)]), two newly joined parties have no means to know each other, so no session will be established between them, and thus, the fully distributed signaling architecture cannot be maintained. Due to the information propagation procedure, the cluster scheme defined in this paper can handle most coincident behaviors, but in some cases, protection mechanisms should be used to prevent inconsistencies. We discuss this issue case by case.

4.7.1 Coincident join:

Two or more parties join a conference at the same time. They may join the same cluster or different clusters. Our scheme can handle this case because the coincidently joined parties do not have a direct session with Fig. 1. Cluster splitting. (a) Before splitting. (b) After splitting. Fig. 2. Information propagation as a new party joins. each other. Instead, they establish sessions with the supermembers that are already in the cluster and later they can "know" each other from their supermember(s).

4.7.2 Coincident departure.

Two or more participants leave a conference at the same time. They may leave the same cluster or different clusters. Similar to the first case, our scheme can handle the coincident departure of members and less than two super members. Our scheme cannot handle the coincident departure of supermembers because the newly selected super members cannot already have knowledge of each other. In order to avoid such a case, we define a protection phase when a super member leaves. A supermember should reject any session establishment or termination request when it starts to leave. The protection phase ends when the super member has completed the leaving procedure. Within this protection phase, a super member leaving procedure will be failed if another super member is leaving at the same time, because the newly selected super member cannot establish a session with a leaving, neighboring super member. If a super member fails to leave, it will retry after a random period of time.

4.7.3 Coincident splitting.

Two or more clusters split at the same time. Due to the mesh structure of super members, the super members in older clusters maintain a session with every newly split super member. After a run of the information propagation procedure, a newly split super member will have knowledge of the other new super members. The logic to be added in order to handle this case is that if a super member finds that it has not established a session with a neighboring super member and if it has a higher address, it will establish a session with that super member. Coincident merging. Two or more pairs of clusters merge at the same time. Our scheme can handle this case because there is no new super member elected. The cluster state can be propagated to all the neighboring super members.

5. CONCLUSIONS AND FUTURE WORK

MANETs have been an active research area for several years. A key motivation is the possibility of novel application scenarios. Conferencing enables a set of attractive applications in MANETs. We find it an interesting research topic because it is very challenging to deploy a conference in MANETs. For example, the centralized conference model defined for traditional infrastructure-based networks is not adequate for MANETs.

In this paper, we addressed an important conferencing issue: signaling. We have presented the challenges, reviewed the state-of-the-art signaling protocols, and proposed a cluster-based signaling scheme. The clusters are dynamically created at the application level, taking the node capabilities into account. This scheme overcomes the problems of the related works and meets the requirements that we derived. In order to implement the scheme in a real MANET environment, we have also discussed the implementation technologies. We found that SIP extensions and SIP distribution are promising solutions.

The performance of the signaling scheme has been evaluated through simulation. For the simulation experiments, we have developed all of the signaling entities that we defined. The results show that our signaling scheme outperforms the existing conferencing framework for ad hoc networks. It can support different scales of conferences and it can run on different lower layer routing protocols. In this work, we have evaluated conference scales up to 100 nodes. It would be very interesting to know the behaviors of the signaling scheme with hundreds and even thousands of nodes. However, the current experimental environment has a limited scope. Another interesting direction for future work is the evaluation of our conference scheme under different wireless settings, such as various coverage ranges, fading conditions, and background noises. Still another interesting future work is optimization of the signaling scheme. This can be done by improving the signaling scheme at the application layer or by using a cross-layer design.

6. REFERENCES

[1] C. Fu, R.H. Glitho, and R. Dssouli, (Mar. 2005) "A Novel Signaling System for Multiparty Sessions in Peer-to-Peer Ad Hoc Networks," Proc. IEEE Wireless Comm. And Networking Conf. (WCNC '05),.1004 IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 8, NO. 7, JULY 2009 TABLE 3 Average Recovery Delay with Different Types of Session Failures

[2] O. Novo, G. Gamarillo, D. Morgan, and R. Even,(Apr. 2006) "A Common Conference Information Data Model for Centralized Conferencing (XCON)," IETF Internet draft, work in progress,.

[3] J. Rosenberg, (Feb. 2006) "A Frame Feb. 2006work for Conferencing with the Session Initiation Protocol"(SIP), IETF RFC 4353.

[4] H. Bulut, A. Uyar, and G.C. Fox, (July 2005)"Adapting H.323 Terminals in a Service-Oriented Collaboration System," IEEE Internet Computing, vol. 9, no. 4, pp. 43-50.

[5] J.Y. Yu and P.H.J. Chong,(Mar. 2005) "A Survey of Clustering Schemes for Mobile Ad Hoc Networks," IEEE Comm. Surveys, vol. 7, no. 1, pp. 32-48.

[6] J. Liu and I. Chlamtac, (July 2004) "Mobile Ad-Hoc Networking with a View of 4G Wireless: Imperatives and Challenges," Mobile Ad Hoc Networking, chapter 1, Wiley-IEEE Press.

[7] H.323 Series,(2003) ITU-T Recommendations.

[8] J. Rosenberg et al., (June 2002)SIP: "Session Initiation Protocol", RFC 3261.

[9] P. Koskelainen, H. Schulzrinne, and X. Wu, (May 2002) "A SIP-Based Conference Control Framework," Proc. ACM Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '02).

[10] H.J. Wang et al., (Aug. 2000)"Iceberg: An Internet Core Network Architecture for Integrated Communications," IEEE Personal Comm., vol. 7, no. 4, pp. 10-19 .

[11] C. Bormann, J. Ott, and C. Reichert,(Dec. 1996) "Simple Conference Control Protocol," IETF Internet draft, work in progress.

[12] E.M. Schooler and S.L. Casner, (1992)"An Architecture for Multimedia Connection Management," Proc. IEEE Fourth Comm. Soc. (ComSoc) Int'l Workshop Multimedia Comm. (MM '92), pp. 271-274

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:
http://www.iiste.org

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. **Prospective authors of IISTE journals can find the submission instruction on the following page:** http://www.iiste.org/Journals/

The IISTE editorial team promises to the review and publish all the qualified submissions in a fast manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar