Design and Implementation of an Audio/Video Group Chat Application for Wireless Mesh Networks

Misun Yu, Woosuk Cha, Jun-Keun Song, Ji-Young Kwak, Seon-Tae Kim
Electronics and Telecommunications Research Institute
Email: {msyu, wscha, jun361, jiyoung, stkim10}@etri.re.kr

***Abstract*—Widespread powerful mobile devices and the advanced technologies in wireless mesh networks (WMNs) open a new scenario where users require audio/video group chatting with their neighbors anywhere without the need of fixed communication infrastructure. However, development of audio/video group chat applications for WMNs are still challenging because the applications cannot rely on a central server, and the topology of the underlying WMNs is frequently changed and unreliable. In spite of those difficulties, the applications should perform audio/video data processing for multiple members and chat group management concurrently. In this paper, we define important requirements of audio/video group chat applications for WMNs, and then, explain our design and implementation for realizing them. To show the feasibility and scalability of our application, we implemented prototype group talk system composed of an 802.11n WMN and smartphones connected to wireless mesh nodes, and measured group management and video processing performance increasing the number of group members.**

I. Introduction

The widespread use of high performance smartphones recently enables real-time audio/video chat on the move. The video chat applications such as Skype, Google Talk, ooVoo, FaceTime [Colon, 2011] and Fring [Alcorn, 2011] are the well-known video chat applications for smartphones. Especially, Fring provides group video chat functionality, and four people are able to chat concurrently with video with this, which is beyond the typical 1:1 video calls. These video chat applications are very useful when we're collaborating at work, or talking to people staying away on their smartphones.

The applications mentioned above require priori-planned infrastructures like the Internet, 3G or LTE, and have fixed servers managing user identity and connectivity. When we cannot rely on a fixed infrastructure and servers in the disaster area or construction sites, dynamically configured network and the applications considering dynamism of underlying networks are required.

Wireless mesh networks (WMNs) are dynamically self-orga-nized and self-configured, with the nodes in the networks automatically establishing an ad hoc network and maintaining the mesh connectivity [Akyildiz and Wang, 2005]. The recent technical advances in WMNs enable mobile users to collaborate together anywhere using audio/video data transmission. And it opened a new scenario where users require audio/video group chatting with their neighbors anywhere without the need of fixed communication infrastructure. However, there is no suggestions for developing audio/video group chat applications for WMNs. Therefore, in this paper, we define important requirements for developing those applications and explain our design and implementation to realize them. To show the feasibility and scalability of our application, we implemented 802.11n WMNs and developed an audio/video group chat application working with them.

In this paper, we describe the requirements, design and implementation of the audio/video group chat application in detail. The rest of this paper is organized as follows: section II presents important requirements of audio/video group chat applications for WMNs without servers. In section III, the design and implementation of our group chat application is described. An experiment of audio/video group chat with our application is provided in section IV. We conclude in section V.

II. Requirements

Generally, audio/video group chat applications process calling, invitation of new members and audio/video transmission. There are additional requirements to reliable audio/video group chat functionality for WMNs without servers; those may be not important considerations for fixed infrastructure based applications.

In WMN, split and merge of networks caused by configuration change occur frequently, and data transmission is performed among mobile clients via the mesh nodes. Therefore, applications should provide reliable group management considering frequent packet loss.

And also, because of mobility of mobile clients, connection with WMNs is occasionally disconnected. This change of connectivity can causes change of allocated IP addresses of user devices during group chat.

And, if we do not assume fixed servers (dedicated nodes with high capability), each mobile client should discover and monitor status of online nodes for managing groups and members as well as convenience of mobile users.

Most important of all is that increase of number of group members should not lead to significant performance degradation of the applications.

Followings are the main requirements of audio/video group chat applications for wireless mesh networks that we considered.

- Automatic discovery and monitoring of on-line nodes
- Fast and reliable chat-group management
- IP change during group chatting
- Client-side group voice mixing
- Group video processing
- Scalability

III. Design and Implementation

A. Application Framework

To provide audio/video group chat functionality without fixed servers in WMNs, our application includes four components: user interface package, group management package, audio/video handling library and audio/video streaming library. Figure 1 shows the structure of our application. Mobile clients such as smartphones are connected with WMNs through access points (APs) attached to mesh nodes.

User interface and group management packages are implemented in Java to conveniently use android GUI and socket library. Group data handling and audio/video streaming libraries are implemented in C to improve performance, and to reuse our legacy code for handling audio codec (G.711, G.729), video codec (H.264), and streaming protocol (RTP: Real-time Transport Protocol). These libraries are implementing standard codecs and a protocol, and hardly changed. Thus, we implemented them as independent libraries.

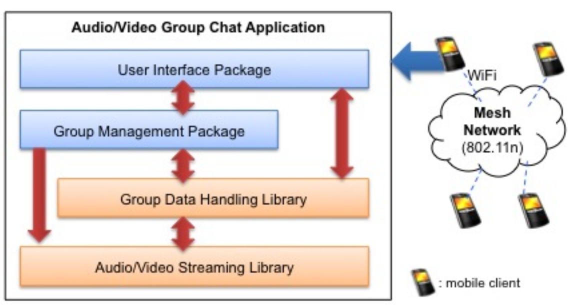

Fig. 1. Framework

The followings describe the detailed design of each component.

B. Group Management Package

Using our application, users can create chat groups, and invite new members during the chat. When a user creates a group, group management package allocates a unique group identifier (ID) to the group not to overlap with those of other groups in its WMN. To get unique group IDs, group management package always monitor currently activated groups using group-signaling messages described below.

There have been several group and member management methods in the routing level of ad-hoc networks [Liu et al., 2005][Vasudevan et al., 2004]. Because we manage the application level groups, we used a similar but simpler algorithm to reduce the processing time and number of transmitted packets.

1) Node Discovery: To discover online members and active groups, all nodes send periodic heartbeat messages just after connected to WMNs. If a node is not a member of a group, it broadcasts HELLO messages including its IDs and IP addresses. Node ID is a unique identifier of mobile clients. We are using media access control (MAC) addresses as node IDs.

After joining a group, the leader of the group broadcasts HELLO_GROUP messages, which containing the states, IDs and IP addresses of their group members. All nodes in WMNs use HELLO_GROUP messages for monitoring status of other nodes. The state is one of three states - audio chat, video chat or idle.

Other members of the group unicast HELLO_MEMBER messages containing the states, IDs and IP addresses of them for letting other members monitor the status of them without relying on the HELLO_GROUP messages from the leader. In WMNs, change of leaders occurs frequently due to unstable link state, but members should know the exact status of other members as soon as possible to inform to mobile users even if the leader cannot be reached.

Table I shows the fields of heartbeat messages.

TABLE I
Content of Group Messages

Message Type	Contents
HELLO	Node ID, node IP
HELLO_MEMBER	Group ID, Node IP, Node ID, State
HELLO_GROUP	Group ID, (Node IP, Node ID, and State of each member)

2) Group and Member Management: When inviting new members, group members send JOIN_R-EQUST message to them. A node received JOIN_REQUEST responses to the sender by sending JOIN_RESPONSE including user response (yes or no). The sender adds the node sending the positive response as a new member, and then send MEMBER_LIST message including new member list to all the other group members. When a member leaves a group, it send TERMINATE_IND messages to their group members.

Figure 2 shows the processes described above.

Initial leader is the creator of the group. When a leader leaves its group, it designates the next member in the member list as the next leader. When a group is split because of node or network problems, or a leader of a group is suddenly dead, the most top on-line member in the last MEMBER_LIST message becomes a leader of the group and starts sending HELLO_GROUP. When there is conflict caused by duplicated leaders, the member having smaller ID becomes a leader, gathers new member information and broadcasts it.

3) IP Change Support: Because of multi-hop nature of WMNs and mobility of mobile clients, mobile clients can be occasionally disconnected and connected with several mesh nodes of WMNs (actually with APs attached to mesh nodes), which can causes the change of IP address allocated to them.

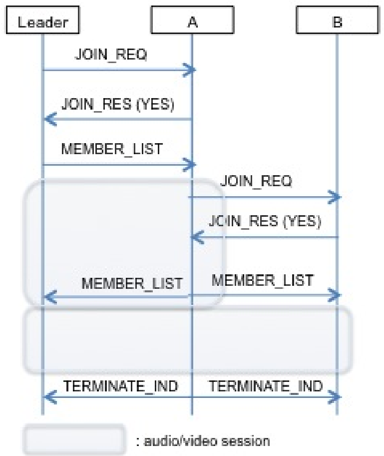

Fig. 2. Member Join

Therefore, the applications should monitor connectivity to APs and detect the change of IP addresses. If IP change of mobile clients is detected, new IP address is used to receive heartbeat messages and audio/video data from other members.

The IP change of other members can be detected from heartbeat messages, and new IP address is used to make new audio/video streaming sessions and send group signaling messages. All states of the members are preserved equally as before.

C. User Interface Package

User interface package provides graphic user interface to display on-line nodes, active groups, member states as well as video screens. This module delivers group-related user events such as group join/leave requests to group management package and receives group-related messages from it to display the information.

D. Group-Data Handling Library

Group-data handling library performs group-data management such as audio/video encoding/decoding, audio buffer management and video index management of members

This library generates encoded audio/video data using codec, and sends them to the audio/video streaming-library. When the group-data processing modules receive audio/video data from the streaming library, they process them like followings.

1) Client-side Group Voice Mixing: Because irregular data transmission delays, audio packets from group members cannot arrive in a fixed order. Therefore, we allocate a queue to each member saving its audio data temporarily. Received packets are saved in the queues if the state of the sender (member) is not 'idle'. After receiving at least one packet from all live group members whose state is not 'idle', the first packet from all queues of the members are de-queued, decoded, mixed and out to the speaker. When the state of a member is changed to 'idle', its queue is cleared.

2) Group Video Processing : We allocate display indices to group members whose states are 'video chat'. After receiving video packets from the streaming library, the packets are decoded and delivered with the index of the sender to the user interface package for screen display. Display indices are updated when the video members are changed. Each video index is matched with one part of the screen of each mobile device.

E. Audio/Video Streaming Library

Audio/video streaming library provides audio/video-streaming (sequential data transmission and reception) functionality using real-time transport protocol (RTP). Received data is delivered to audio/video data processing module. When transmitting data, this library uses the IP address of group members provided by the group management package.

To complement data loss, we transmit duplicated audio packets. Although duplication requires more bandwidth than transmission without duplication, it significantly improved quality in cases of frequent packet losses due to interference or topology change.

IV. Prototype of Audio/Video Group Chat System and Its Performance

To show the feasibility and measure performance of our application, We constructed prototype audio/video group chat system composed of mobile clients and mesh nodes just as we explained. We used hybrid wireless mesh protocol (HWMP) as a routing protocol of our WMN. Figure 3 shows the software structure of our mesh nodes.

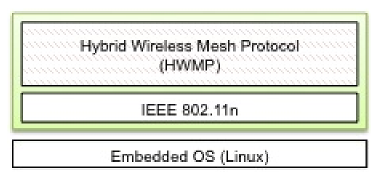

Fig. 3. Software Structure of a Mesh Node

We constructed a WMN having maximum 4-hops, and connected smartphones installed our applications. Figure 4 shows the capture screens of our prototype application. Left one is audio group chat and right one is video group chat. Size of video was QVGA and video data was transmitted at 300K bps.

We compared performance of the video group chat functionality with that of another application named Fring, which is a mobile application supporting 4-way DVQ (Dynamic Video Quality) video chat.

And also, we measured performance of our group management and audio group chat functionality of our application itself. In case of audio group chat, there are no audio group chat (or conferencing) applications. Although Skype supports conference calls, we just can participate in them not host them on smartphones using it. So, we measured additional CPU

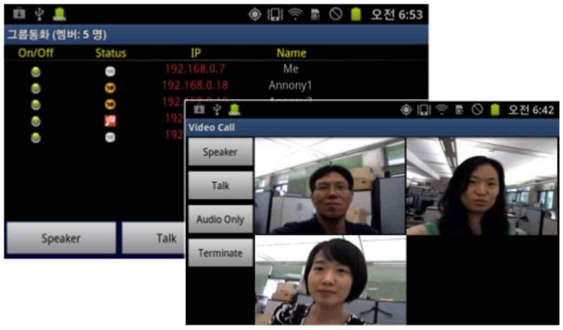

Fig. 4. Three-way Video Chat

load of our application increasing the number of concurrent speakers to test scalability of a chat group.

Table II the environment settings of this experiments. We measured CPU load of applications according to the different number of group members. In this experiments, all mobile devices are connected to one AP to stably measure the performance only increasing group members.

TABLE II
EXPERIMENT SETTINGS

# of Groups	1
# of Group Members	1 ∼ 10
# of Audio Members	1 ∼ 10
# of Video Members	1 ∼ 3
User Devices, OS	Galaxy S II (S5PC110 processor), Android 2.3

Figure 5 shows the performance of our group management. We measured CPU load when our application displays states of group members without audio/video transmission. The blue line represents load of group leader according to the member increase, and the red line represents that of non-leader. Because a group leader broadcasts HELLO_GROUP message once every interval, and non-leaders unicast HELLO_-MEMBER to all the other members, non-leaders require more processing resources than the leader. We set the interval of heartbeat messages as 500ms.

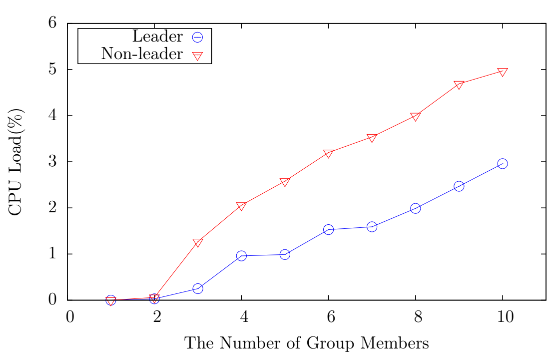

Fig. 5. Performance of Group Management

Figure 6 shows the performance comparison of group video chat between Fring and our application. Red bars represent AVGroup, and green ones represent Fring. Fring remains stable CPU load because of its dynamic video quality but always require more CPU resource than ours.

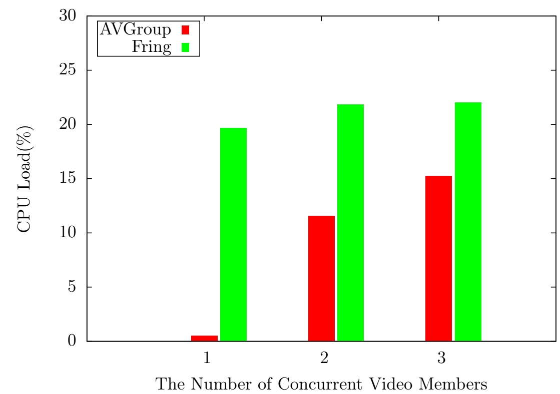

Fig. 6. Performance of Group Video Chat

Figure 7 shows the CPU load of a member (non-leader) according to the increase of concurrent speakers in a group. This test shows the possibility of mobile group talk without servers in mesh networks because member increment and multiple audio mixing did not increase CPU load dramatically.

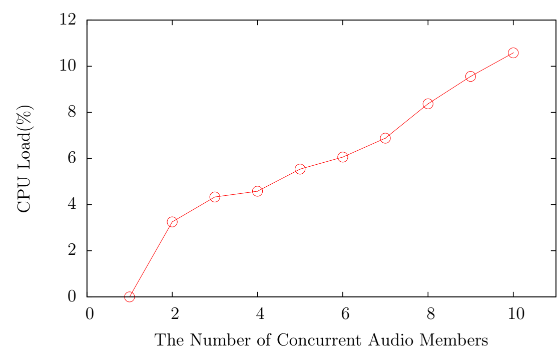

Fig. 7. Performance of Group Audio Chat

V. CONCLUSION

We finished the design and implementation of audio/video group chat application for wireless mesh networks. And also, we tested prototype application with real mesh network and showed better video group chat performance than the other group chat application named Fring. Now, we are adding security-related functions in order to successfully use our application in commercial and military applications and markets.

REFERENCES

[Akyildiz and Wang, 2005] Akyildiz, I. and Wang, X. (2005). A survey on wireless mesh networks. *Communications Magazine, IEEE*, 43(9):S23 – S30.

[Alcorn, 2011] Alcorn, A. (2011). Get group video chat for your smartphone using fring [android & iphone].

[Colon, 2011] Colon, A. (2011). The best video chat apps for your smartphone.

[Liu et al., 2005] Liu, J., Sacchetti, D., Sailhan, F., and Issarny, V. (2005). Group management for mobile ad hoc networks: design, implementation and experiment. In *Proceedings of the 6th international conference on Mobile data management*, MDM '05, pages 192–199, New York, NY, USA. ACM.

[Vasudevan et al., 2004] Vasudevan, S., Kurose, J., and Towsley, D. (2004). Design and analysis of a leader election algorithm for mobile ad hoc networks. In *Proceedings of the 12th IEEE International Conference on Network Protocols*, ICNP '04, pages 350–360, Washington, DC, USA. IEEE Computer Society.