A New Algorithm for Resolving Segregation in Hierarchical Conference

Sungmin Lee, Keonbae Lee, Dongsu Seong, and Minseok Oh
Department of Electronic Engineering, Kyonggi University, Suwon, Korea
forsungmin, kblee, dssung, msoh@kyonggi.ac.kr

Abstract

In ubiquitous environment, people can communicate with others and have access to various computing services any time, any place. The multipoint conference systems in ubiquitous environment are expected to be one of the leading applications to connect people. Even though a hierarchical multipoint conference system using the endpoint mixing scheme works effectively in ubiquitous environment, a participant leaving a conference session may divide the session into pieces depending on where the leaving participant is located in the hierarchy. A solution proposed previously for this may result in reducing conference session duration since parent nodes, which usually consume more power than leaf nodes, become again parent nodes while the connections are reconfigured to maintain the session. The article proposes a new scheme which prolongs the session duration when a user leave a conference group, compared with the earlier scheme.

Keywords: Endpoint mixing, multipoint conference, ubiquitous environment, VoIP.

1. Introduction

The recently attention-getting ubiquitous computing environment will provide communication means and computing service to people any time, any place. This revolutionary computing environment is a new paradigm, so that social, cultural, and economic impacts will never be light. The video conference system in a ubiquitous environment is expected to be one of the leading applications for connecting people [1].

One of the distinct characteristics in a conference system in ubiquitous environment is that users having a wireless mobile terminal are usually collocated in a confined area. In those cases, to communicate with each other in the confined area, a mobile terminal may have to use a fixed conference server located far away from those users. When the centralized conference scheme is used to set up calls between users close in distance, the calls have to be processed in a fixed server, which results in poor throughput performance due to the long-distance travel of user and signaling traffic. When the distributed conference scheme is used for conference call setup, mobile terminals exchange users' audio/video data directly peer-to-peer. It may result in rapid dissipation of battery power since each mobile terminal processes the data compression, decompress and mixing itself. In the distributed conference scheme, the call control signaling is performed still in a fixed server [2].

Schemes which do not require the conference server include the pseudo centralized conference scheme, the pseudo distributed conference scheme, and the endpoint mixing conference scheme [3]. The pseudo centralized conference scheme uses a mobile terminal as a centralized conference server for processing the call control messages and conveying the user data. The selected mobile terminal may experience a shortage of resource, such as CPU usage, memory availability, or battery power, since all the traffic is delivered through the selected mobile terminal. Sometimes given tasks, such as audio/video compression, decompression, and mixing, may not be completed within the time requirement for delivering audio/video data.

The pseudo distributed conference scheme uses a certain mobile terminal as a distributed conference server only for call control messages. The user data is exchanged directly between users without help of the mobile conference server as in the distributed conference scheme. In this case since each mobile terminal again processes and communicates the user data, the battery power consumption may be significant.

In ubiquitous environment, an endpoint mixing scheme can become an alternative for the pseudo centralized conference scheme and the pseudo distributed scheme. The endpoint mixing scheme is similar to the pseudo centralized conference scheme in that call control messages and user data exchanges for child mobile nodes are performed via a parent mobile node, but they differ in that the mobile node which functioned as a centralized server in the centralized conference scheme is not responsible for all the mobile nodes in the session any more. Each parent mobile node takes care of its child mobile nodes for exchanging call control messages and user data. In other words, a parent node becomes a conference server for its own child nodes. Then the endpoint mixing scheme can alleviate the battery power consumption since the processing for user data and their delivery is distributed to several parent mobile nodes [3]. The endpoint mixing scheme[3] effectively form the conference hierarchy.

However, the endpoint mixing scheme has also drawbacks. One of them is that in a hierarchical multipoint conference system using the endpoint mixing scheme, when a certain mobile node leaves a conference call group in session, it may cause some other mobile nodes to be kicked off from the conference call group. A solution for this problem proposed in [3] has a defect that leaving a conference session may reduce the lifetime of the session when the terminals are powered by batteries. Since terminal devices will be most likely connected wirelessly in ubiquitous environment, the duration of a conference session will heavily depend on the battery lifetime of the terminal devices [4]. This paper proposes a new scheme which improves the session duration when a user leaves a conference group, comparatively with the earlier scheme.

2. Segregation Problem and its Earlier Solution

The endpoint mixing conference scheme can alleviate the battery power consumption since processing for user data and their delivery is distributed hierarchically to all the parent mobile nodes as shown in Figure 1. When mobile node *I* requests a connection to node *D*, node *I* become a child node of node *D*, and node *D*, which accepts the connection request, becomes the parent node of node *I*. Once a hierarchical structure is built using the endpoint mixing scheme as in Figure 1, node *A*, which has child nodes but no parent node, is denoted a root node. On the contrary, nodes *E*, *F*, *G*, *H*, *I*, and *J*, which have a parent node but no child nodes, are denoted leaf nodes. All the nodes except the root and leaf nodes will have a parent and child nodes at the same time.

In a hierarchical multipoint conference system, when a certain mobile node leaves a conference call group in session, it may cause some other mobile nodes to be separated from the conference call group. It can occur when a mobile node which performs audio/video mixing leaves the conference, so that all the child mobile nodes under the mobile node will be separated from the conference call group. For example, when mobile node *A* in Figure 1 leave the conference call group, it segregates the connection between nodes *B*, *C*, *D* and then divide the conference call group into three as shown in Figure 2.

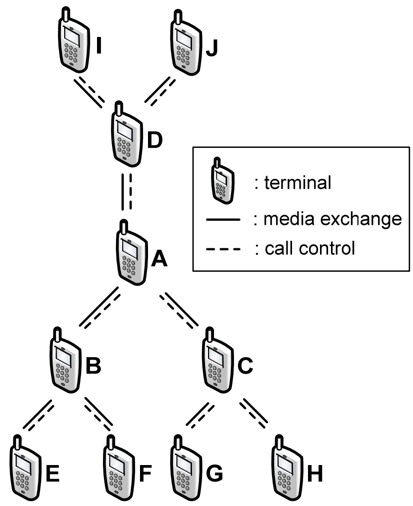


Figure 1. A hierarchical conference session having 10 participants.

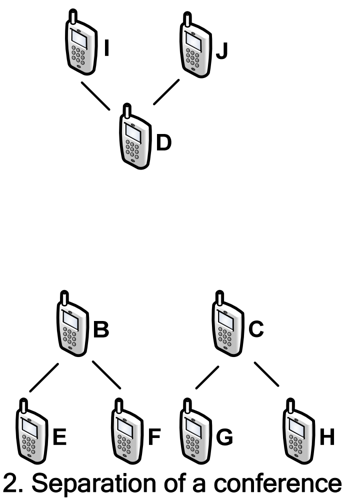


Figure 2. Separation of a conference system when a participant leaves.

We have proposed a scheme which uses the REFER command immediately before or after the node, who wants to leave the conference call group and sends the BYE command [3], [5]. This prevents segregation of a conference group when a mobile node leaves a hierarchical multipoint conference system. Figure 3 shows the signal flow when node *A* leaves the session. Node *A* instructs nodes *B* and *C* to make connection requests to node *D* using the REFER command before it initiate its own departing procedure (using the BYE command) and nodes *B* and *C* send connection requests to node *D* using the INVITE command. Then node *B*, *C*, and *D* will construct a new formation as shown in Figure 4 without dismissing and establishing a new session again.

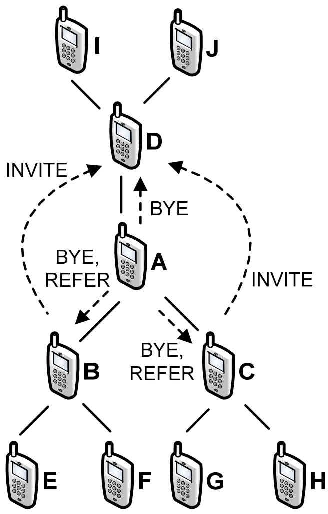


Figure 3. Connection request by node *A*, from node *B* and *C*, to node *D* before departure.

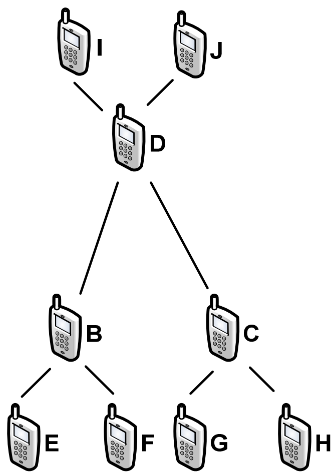


Figure 4. A new formation after departure of node *A*.

3. Reduction Rate of Session Duration

As the number of neighbor nodes (defined as a parent node plus child nodes of a node in question) increases, the node in question will have more user data to process, which will deplete quickly the node's resources, e.g., typically battery power. In Figure 1, nodes *A*, *B*, *C*, and *D* carries most child nodes and they will consume more battery power compared to other nodes. Battery power directly affects the wireless devices' lifetime. Currently widely used wireless devices are based mostly on the IEEE802.11 radio transmission technology. Since it does not perform automatic power control to adjust its transmission power accordingly and the processing power for endpoint mixing is considered to be greater than the one for radio transmission, we use a simple power consumption model for a node which is mainly proportional to the number of neighbor nodes attached to it.

Suppose that we want to establish a conference session which will last at least *T*. Node *i* measures its remaining battery power P_i just before a conference session is established. Then node *i* can calculate its maximum number of neighbor nodes, i.e., n_i^{max} using Eq. (1) according to the model we adopted above. When node *i* receives a call connection request, it accepts the request if the number of the current neighboring nodes is less than n_i^{max}, reject it otherwise [4].

$$P_i = kTn_i^{\max}, \tag{1}$$

where *k* is a heuristically determined constant.

Upon assuming each node starts a conference session with the same amount of the remaining power, when node *i* in the session have a number of the neighbor nodes less than n_i^{max}, the conference duration node *i* can sustain can be described as

$$T_i = \frac{P_i}{kn_i}. \tag{2}$$

From Eqs. (1) and (2), we can derive relationship between T_i and *T* as follows.

$$T_i = T\frac{n_i^{\max}}{n_i} \tag{3}$$

Assuming T_i is smaller than *T* (this occurs when n_i is greater than n_i^{max}), let the reduction rate R_i of conference duration in node *i* be defined as

$$R_i = \frac{T - T_i}{T} \times 100. \quad (4)$$

Plugging Eq. (3) into Eq. (4), the duration reduction rate R_i can be rewritten as

$$R_i = \frac{n_i - n_i^{\max}}{n_i} \times 100 \quad \text{for } n_i > n_i^{\max}. \quad (5)$$

n_i may grow beyond $n_i^{\max}$ and the conference may end abruptly due to lack of power at a certain parent node in the middle of a session [4].

4. Leaving a Session Keeping the Lifetime Longer

The earlier scheme [3] may result in greater neighbor nodes n_i than $n_i^{\max}$ after a participants leaves a conference group and a new group is formed. For example, in a newly formed conference group as shown in Figure 4, the number of neighbor nodes in node *D* has increased to 4 from 3. It may reduce the lifetime of the whole conference session.

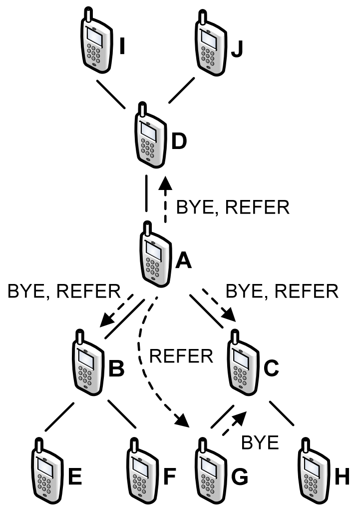
The newly proposed scheme for maintaining the lifetime when a participant leaves a group is as follows: Suppose all nodes in a conference have the current conference states with information exchanges such as topology, remaining power and IP address periodically. Also suppose all nodes start a conference session with the same amount of battery power. Leaf nodes will usually use less power than a parent node since they carry less neighbor nodes and do not perform user data mixing. When a node is about to leave a conference group, a node maintaining the most battery power informs its existence to the leaving node's neighbor nodes. Then, the leaf node takes the position of the leaving node after the node leaves the group. Figure 5 shows the procedure when node *A* leaves a conference group. Assuming node *G* has the maximum remaining battery power, nodes *B*, *C*, and *D* will be informed of the existence of node *G* as a node having the maximum battery resource. When node *A* leaves, nodes *B*, *C*, and *D* will request connection setups to node *G* and node *G* will become a parent node of nodes *B*, *C*, and *D* as shown in Figure 6.

Algorithm: choose node *N* which has the maximum remaining power

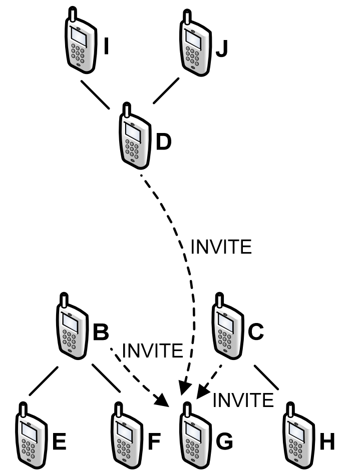
$P_{\max} = 0$, $N = 0$;

for terminal $T_i \in$ all terminals in the system
{
Calculate remaining battery power P_i.

if $P_i > P_{\max}$
$P_{\max} = P_i$, $N = T_i$;
}
Select Node *N*.
Node *N* takes the position of the leaving node.



(a)



(b)

Figure 5. Connecting separated groups (a) G informs using REFER, (b) B, C, and D are connecting to *G*.

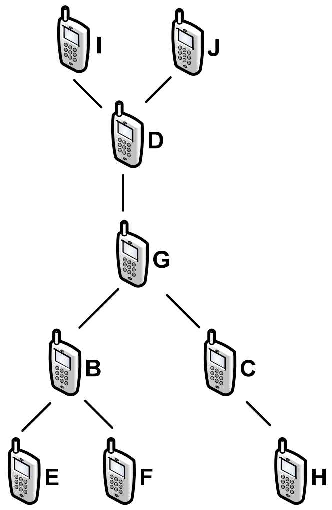


Figure 6. Newly formed conference group.

5. Experimental Results

A series of simulations were performed varying the number of participants from 1 to 15. For each simulation, we have established randomly generated 50,000 conference sessions. When a new node joins a conference in session, the call connection request is assigned to an arbitrary node and the node accepts it as long as the number of its neighbor nodes does not exceed n_i^{max}. Once a conference group is formed with the desired number of nodes, a parent node leaves the group and conference group is reformed with the old scheme [3] and the newly proposed scheme. We compare the old scheme and the new scheme in terms of the duration reduction rate of a conference.

Figure 7 shows the average duration reduction rate of a conference. The newly proposed scheme results in 0 reduction rate for all cases, but the old scheme shows the duration reduction rate starts to increase when the number of participants becomes 4 and the rate increases more as the participants increases. This shows that the newly proposed scheme always provides the longest conference duration the system can provides regardless of the number of participants. If the remaining power of the terminal node selected from the algorithm in section 4 is greater than one of the leaving node, the reduction rate of the conference is always 0. Generally, terminal nodes will usually use less power than a parent node since they carry less neighbor nodes and do not perform user data mixing. Therefore, the conference duration of the proposed scheme is longer than old scheme to select one among neighbor nodes of leaving node.

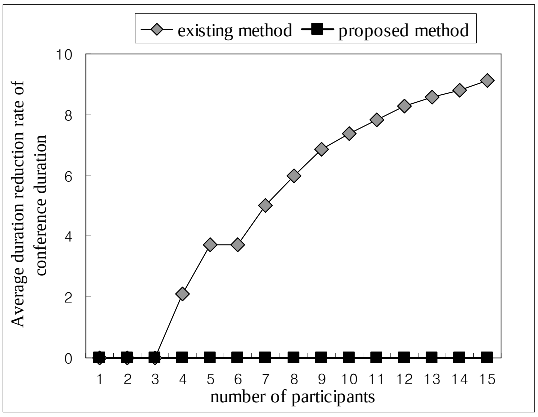


Figure 7. Average duration reduction rate of conference.

A Ubiquitous Fashionable Computer [1] as shown in Figure 8, a wearable computer, is being developed for easy access to various services in a ubiquitous environment. We have been testing the endpoint mixing conference scheme on the UFC terminal in a mesh network environment. In the experiment using actual working nodes, we have found that when a node leaves a group and a conference group is reconfigured, the average duration reduction rate of the newly proposed scheme is lower than the rate of the old scheme.

Figure 8. A Ubiquitous Fashionable Computer

6. Conclusions

The endpoint mixing scheme imposes advantages as well as disadvantages when applied in ubiquitous environment. One of them is separation of a conference group into pieces when a participant, specially a parent node, leaves the session. After the leaving, the session will continue and the new root node will undergo severe battery consumption regardless of its remaining battery resource assuming the conference duration strongly depends on the battery power of every wireless node. In this article, we have proposed an improved scheme for prolonging the conference duration. The experimental results have shown that the proposed scheme did not shorten the conference duration when a parent node left at all. We are sure that the multipoint conference scheme will help enhancing the various application services in ubiquitous environment. The multipoint conference system is considered to be an important area for further research for the proliferation of the ubiquitous application services.

7. Acknowledgement

This work was supported by the “GRRC” Project of Gyeonggi Provincial Government, Korea.

References

[1] Kyu Ho Park, Seung Ho Lim, and Dae Yeon Park, "UFC: A Ubiquitous Fashionable Computer," Next Generation PC 2005 International Conference, pp.142-147, 2005.

[2] W. J. Rosenberg, A Framework for Conferencing with the Session Initiation Protocol, Draft-ietf-sipping-conferencing-framework-05, 2005.

[3] Minseok Oh, Keonbae Lee, and Dongsu Seong, "Multipoint conference systems in a ubiquitous environment," Proc. of SPIE Vol.6749, pp.67945I-1-6, ICMIT 2007.

[4] Pyungsu Kim, Sungmin Lee, Kiyong Kim, Keonbae Lee, Dongsu Seong, and Minseok Oh “A Resource Management Scheme in the Endpoint Mixing Conference in a Ubiquitous Environment”, Proceeding of 7th APIS, pp.157-160, 2008.

[5] J. Rosenberg, H. Schulzrinne, et al, "SIP : Session Initiation Protocol," IETF, RFC 3261, 2002.