iPTT: Peer-to-Peer Push-to-Talk for VoIP

Jiun-Ren Lin∗, Ai-Chun Pang++ and Yung-Chi Wang+
∗Department of Computer Science and Information Engineering
+Graduate Institute of Networking and Multimedia
National Taiwan University, Taipei, Taiwan, R.O.C.
Email: jrlin@newslab.csie.ntu.edu.tw, acpang@csie.ntu.edu.tw, dav@newslab.csie.ntu.edu.tw

Abstract

This paper proposes iPTT, a peer-to-peer Push-to-Talk (PTT) service for Voice over IP (VoIP). In iPTT, a distributed and mobile-operator independent network architecture is presented to accelerate the deployment of the PTT service. Based on the serverless architecture, we develop two mechanisms, i.e., flooding-based floor control mechanism (FFC) and tree-based floor control mechanism (TFC), for real-time talk-burst determination. The determination algorithms and the corresponding message flows for these two mechanisms are designed to show the feasibility of FFC and TFC. The performance of FFC and TFC is investigated through our analytical and simulation models in terms of the determination latency and the number of floor-control message exchanges.

1 Introduction

With the explosive growth of the Internet subscriber population, supporting Internet telephony service, also known as Voice-over-IP (VoIP), is considered a promising trend in the telecommunication business. In addition to globally-deployed wired Internet telephony services, integrating VoIP into wireless systems (e.g., 3G/GPRS, IEEE 802.11 and IEEE 802.16) is extensively studied [2]. Particularly, 3GPP introduced the IP Multimedia core network Subsystem (IMS) for Universal Mobile Telecommunications Systems (UMTS) to provide real-time services such as VoIP over an all-IP network architecture [1, 9].

With various wireless-VoIP applications, a walkie-talkie like service, also called Push-to-Talk (PTT), is gaining significant interest in the mobile telecommunications industry [6]. Most instant communication software provides one-to-one and many-to-many communicational style. The PTT provides another communicational style: one-to-many. This is suitable for mobile devices because of limitations of computing power and bandwidth. The PTT service emphasizes a simple interface for users.
A user only needs to push a button and can talk to a group of people. PTT is a half-duplex voice service that allows user-to-user and group communications. Unlike conventional walkie-talkie systems, the PTT service is supported by ubiquitous wireless access and thus not geographically restricted. A PTT session among a group of users is easily initiated by pressing a button, and the group members take turns talking when they obtain the floor. With PTT, the radio resources consumed by a multi-user call session are greatly reduced by the half-duplex voice transmission.

The existing PTT-over-cellular (PoC) solutions are provided by several mobile operators/vendors [12, 14]. However, the lack of specifications and standards for PoC systems leads to difficulty in supporting inter-operator roaming and compatibility between different equipment-vendors. To promote interoperability between different PTT equipment and networks, the Open Mobile Alliance (OMA) is working on developing the specifications of PoC [15]. The OMA PoC specifications utilize Session Initiation Protocol (SIP) for call signaling, and adopt a centralized architecture to support voice data broadcasting and multi-user coordination. A lot of work has been done to design and evaluate an OMA-based or proprietary PoC system [8, 20, 3, 16]. In an OMA-based centralized PTT architecture, a core node, i.e., PoC server, is responsible for call/floor control and voice relay. Such an architecture is intuitive and easy to implement, but the following issues should be addressed.

**Scalability:** The capacity of a centralized PTT system is limited to the capability of a server.

**Cost:** Maintenance of a standalone server incurs extra costs.

**Reliability:** The crash of the server results in the failure of the entire system. Also, the occurrence of congestion at the server significantly degrades system performance.

Different from the OMA-based design, a hierarchical Peer-to-Peer (P2P) service model is proposed to provide a scalable, cost-effective and robust PTT service, called iPTT [5, 18]. Our iPTT is implemented based on standard SIP/RTP (Real-time Transport Protocol)/RTCP (Real-time Transport Control Protocol). In iPTT, whether real-time voice communications can be achieved depends on the efficiency of the talk-burst determination (i.e., floor control). We develop two floor control mechanisms: flooding-based floor control mechanism (FFC) and tree-based floor control mechanism (TFC), for real-time talk-burst determination over a distributed PTT system. The performance of the proposed mechanisms is investigated through our analytical and simulation models. A series of experiments are conducted to show the capabilities of our FFC and TFC.

The rest of this paper is organized as follows: Section 2 describes the iPTT
In iPTT, a two-level hierarchical structure is adopted to avoid excessive message exchanges among the peers. In this structure, high-level super nodes perform signaling/voice relaying, and handle group-member joining/leaving. Moreover, the floor during an iPTT session is determined by the super nodes that include the session members. On the other hand, there are a large number of ordinary nodes scattered over the Internet. Each ordinary node serves as a group member, and is supervised by a specific super node\(^1\). In order to avoid triangular routing of voice/signaling packets, an iPTT ordinary node is recommended to connect to a super node with a small propagation delay. Some work has been done to study voice relaying for P2P VoIP, and their results could be incorporated into our iPTT super-node selection [17].

The main reason for the adoption of the hierarchical structure in iPTT is to reduce the bandwidth requirement of voice transmission for floor owners. In iPTT, the floor owners need to transmit their voice to all group members during an iPTT session. With a fully mesh structure (without hierarchy) among group members, each floor owner has to duplicate his/her voice packets and individually send the duplicated packets to the group members. This duplication may not be feasible for hand-held devices with limited link rates and computing capabilities. On the other hand, only one voice connection is established between the floor owner and its super node by using the hierarchical structure. Upon receipt of voice packets from the floor owner, the super node multicasts the packets to the group members through the participant super nodes covering the members.

Figure 1 shows an example of our iPTT network architecture. As shown in Figure 1, we assume that there are six nodes of the same group in our iPTT network: \(ON_1\), \(ON_2\), \(ON_3\), \(ON_4\), \(ON_5\), and \(ON_6\). They are supervised by three different super nodes: \(SN_A\), \(SN_B\), and \(SN_C\). \(ON_1\) and \(ON_2\) are supervised by \(SN_A\). \(ON_3\) is supervised by \(SN_B\). \(ON_4\), \(ON_5\) and \(ON_6\) are supervised by \(SN_C\). In this example, the ordinary node \(ON_1\) initiates an iPTT call session. Suppose that the call has been

\(^1\)Note that in iPTT, a super node could be a member of iPTT groups, and equipped with the functionalities of an iPTT caller/callee.
Figure 1: An Example of the iPTT Network Architecture

Successfully established. The arrows in this figure represent the voice transmission path from the call originator \(ON_1\) to all group members (i.e., \(ONs\) 2, 3, 4, 5 and 6). The voice packets generated from \(ON_1\) are transmitted to the destinations through the super nodes \(SN_A\), \(SN_B\) and \(SN_C\). Specifically, the voice packets between \(SN_A\) and \(SN_C\) are not duplicated even though there are three group members under the supervision area of \(SN_C\).

Before describing our iPTT signaling flow, we would like to clarify the relationship between our iPTT and the underlying P2P overlay functionalities. The underlying P2P overlay network provides an interconnection of super nodes, and handles the joining/leaving of super nodes and ordinary nodes. Also, Distributed Hash Tables (DHTs) are maintained in the P2P overlay layer of super nodes to record the locations of group members. Then our SIP-based iPTT layer utilizes those facilities supported by the P2P overlay layer to accomplish call establishment/teardown and floor control. Figure 2 uses a scenario of iPTT call establishment to illustrate the interactions between our SIP-based iPTT layer and the underlying P2P overlay layer. As \(ON_1\) initiates the iPTT call session, a \texttt{SIP INVITE} message is issued from \(ON_1\) to \(SN_A\) (see Step 1). Upon receipt of \texttt{SIP INVITE}, the iPTT layer of \(SN_A\) asks the underlying P2P overlay layer to query the locations of the super nodes that the group members are supervised by (see Step 2). The P2P overlay layer of \(SN_A\) sends \texttt{SIP REGISTER} to a specific super node (based on a specific hash function) to obtain the group information (see Step 3). Then the information is returned to the iPTT layer of \(SN_A\) (see Steps 4 and 5), and \(SN_A\) individually forwards \texttt{SIP INVITE} to each super node that covers the group members (see Step 6). The design of the P2P overlay layer can be found in [19], and is beyond the scope of this paper.

Based on the functionalities supported by the underlying P2P overlay layer, our
iPTT signaling flow can be divided into three stages:

**Stage I (Call Establishment).** $ON_1$ initiates an iPTT call, and issues a SIP INVITE message to the group members. At the end of this stage, voice transmission paths are established, and $ON_1$ broadcasts the voice message to the recipients through the paths.

**Stage II (Floor Determination).** Then in Stage II, $ON_1$ stops broadcasting, and releases the floor. The other group members that intend to talk press the button and contend for the floor via RTCP. The floor-control algorithm determines the next floor owner. One of the group members is granted the floor, and begins to speak. The floor determination process is repeatedly activated as the floor is released.

**Stage III (Call Teardown).** The call originator $ON_1$ leaves the iPTT session by sending the SIP BYE message. The call terminates, and the voice transmission path is disconnected.

The steps for call establishment/teardown are omitted, and readers can refer to [18] for the details. The real-time floor control algorithms and the corresponding procedure will be presented in Section 3.

### 3 Real-time Floor Control

The efficiency of the floor control has a great influence on the performance of our iPTT system. A large determination latency makes the iPTT group communications “un-smooth,” and the conversations between group members are not real-time. Also, the fairness of the floor contention and the volume of signaling message exchanges of the floor-control procedure are taken into account for designing a proper talk-burst determination mechanism. In this section, we present two talk-burst determination
mechanisms: flooding-based floor control (FFC) and tree-based floor control (TFC). The determination algorithms and the corresponding message flows for these two mechanisms are developed. The performance of these mechanisms is investigated through our analytical and simulation models, which will be described in Section 4.

Unlike an OMA-based centralized PTT network, the floor determination for a P2P PTT system is much more challenging. A considerable number of peer nodes are involved in the determination process, and an appropriate node is selected in a short period to obtain the floor. However, the complexity of the determination algorithms and the number of message exchanges are slightly reduced with our two-level hierarchical iPTT network architecture. The floor information is mostly exchanged among the super nodes equipped with higher computing/processing power. In our iPTT system, message exchanging for floor information could be done by using RTCP, SIP or BFCP developed by IETF XCON [4, 7]. Without loss of generality, we assume that RTCP messages are adopted for floor control in this paper. RTCP was originally designed for quality feedback among voice/video session, and RTCP paths are basically the same as those for RTP. Thus it is natural to adopt RTCP for floor control to determine the next talk-burst and the voice transmission direction.

To support our iPTT floor control, RTCP messages are modified to accommodate some floor-control extensions [15]. Five RTCP messages are defined for the support of floor control: RTCP Floor Request, RTCP Floor Ack, RTCP Floor Granted, RTCP Floor Taken and RTCP Floor Release. When someone would like to contend for the floor, an RTCP Floor Request message is sent. Upon receipt of the RTCP Floor Request message, a iPTT member replies with RTCP Floor Ack if she/he has not requested or is not willing to request the floor. The RTCP Floor Granted and RTCP Floor Taken messages are respectively used to inform iPTT members that the floor is granted and taken by the other user. The floor owner announces the floor release to all group members through RTCP Floor Release.

In iPTT, the priorities of floor requests depend on their relative timestamp. A relative timestamp is the length of the period between the time when the RTCP Floor Release is received and the subsequent time when the floor request is made. A small relative timestamp implies that the member is more eager to get the floor and hence a higher priority is set for that request. Each time a RTCP Floor request is issued, the relative timestamp is computed and included. Besides, in order to prevent interference from messages of different floor contention iterations, a run number is included in each floor control message. Each node in an iPTT session maintains a run counter. Whenever the floor owner releases the floor, the run counter will be increased by one. If one node gets a floor-control message that has a smaller run number than that recorded in the node, the message will be ignored. Otherwise, the
message is queued and will be handled.

Before describing our flooding-based (FFC) and tree-based floor-control (TFC) mechanisms, we assume that both ON₂ and ON₅ request the next talk-burst when call originator ON₁ releases the floor. Based on our iPTT hierarchical network architecture, the floor contention is divided into two levels. The local floor-control is applied to the iPTT group members supervised by a single super node. The super node makes its best effort to filter unnecessary RTCP Floor Request messages, and to select a candidate for the upcoming upper-level global floor control. The global floor-control is executed among the super nodes, where floor information can be exchanged in a flooding-based or tree-based manner. The execution steps for global floor-control mechanisms are shown in Figure 3, and described below.
3.1 Flooding-based Floor Control

In FFC, each of the super nodes that cover floor-requesting members issues RTCP Floor Request to the remaining super nodes. Whether the receiving super nodes respond with RTCP Floor ACK depends on the intention of their group members to request the floor. If the super node that has issued the floor request receives floor requests from other super nodes, the super node compares the relative timestamps of the requests to that of the request it sent before. If the timestamp of the request it sent before is smaller, these requests are ignored. Otherwise, RTCP Floor ACK is sent back. The super node gains the floor only when the acknowledgements (i.e., RTCP Floor ACK) from all other super nodes are obtained. Figure 3 (a) shows the steps of FFC.

**Step 1:** $ON_5$ presses the talk button, and issues the RTCP Floor Request message (with relative timestamp $T_5$) to $SN_C$.

**Step 2:** $ON_2$ presses the talk button, and issues the RTCP Floor Request message (with relative timestamp $T_2$) to $SN_A$.

**Step 3:** $SN_C$ sends RTCP Floor Request to $SN_A$ and $SN_B$.

**Step 4:** $SN_A$ sends RTCP Floor Request to $SN_B$ and $SN_C$.

**Step 5:** Upon receipt of the request of $SN_A$, $SN_C$ drops the request since $T_5 < T_2$.

**Step 6:** Upon receipt of the request of $SN_C$, $SN_A$ sends RTCP Floor ACK back to $SN_C$.

**Step 7:** Since no group member of $SN_B$ requests the floor, $SN_B$ responds with RTCP Floor ACK after receiving the request from $SN_C$.

**Step 8:** $SN_B$ also sends RTCP Floor ACK to $SN_A$ to respond to the request of $SN_A$.

**Step 9:** $SN_C$ collects the acknowledgements from all other super nodes, $SN_A$ and $SN_B$.

**Step 10:** $SN_C$ informs $ON_5$ that the floor is obtained through RTCP Floor Granted.

**Step 11:** $SN_C$ also notifies all other super nodes that the floor is taken through RTCP Floor Taken.
3.2 Tree-based Floor Control

In TFC, a tree structure among the super nodes is established during the call-establishment procedure. The floor-requesting information is delivered upward to higher-level super nodes (i.e., internal nodes) in the tree structure. Each super node compares the received requests, and discards the requests with lower priorities (i.e., larger relative timestamps). Finally, the root of this tree determines the floor owner, and propagates this information to all super nodes following the tree structure.

We note that the tree topology may affect the performance of our tree-based floor control. For a k-ary tree \( k \geq 2 \), a large \( k \) may lead to signaling congestion at the root. Conversely, when \( k \) is small, the signaling propagation delay to the root could be large. We set \( k = 2 \) since the number of super nodes in our experiments is small.

As shown in Figure 3 (b), the steps of our tree-based floor control are described as follows.

**Step 1:** This step is similar to Step 1 in Section 3.1.

**Step 2:** \( SN_C \) sends RTCP Floor Request to \( SN_A \).

**Step 3:** This step is similar to Step 2 in Section 3.1.

**Step 4:** Upon receipt of the first floor request, the root (i.e., \( SN_A \)) starts to countdown a timer. The timer is set for the root to have sufficient time to collect the floor requests from super nodes.

**Step 5:** When the timer of \( SN_A \) is over, \( SN_A \) determines the floor owner based on the relative timestamp of the collected requests.

**Step 6:** Then \( SN_A \) grants the floor to \( ON_5 \) by sending the RTCP Floor Granted message via \( SN_C \).

**Step 7:** \( SN_C \) also sends RTCP Floor Taken to \( ON_4 \) and \( ON_6 \).

**Step 8:** \( SN_A \) informs \( ON_1 \) and \( ON_2 \) that the floor is taken. \( ON_3 \) is notified by \( SN_A \) through \( SN_B \).

4 Performance Evaluation

This section investigates the performance of our iPTT flooding-based and tree-based floor-control mechanisms. An analytical model and a discrete simulation model are developed, and a series of experiments are conducted in this section. In terms of floor-determination latency and signaling-message quantity, some numerical examples are shown to indicate the capabilities of TFC and FFC.
4.1 Input Parameters and Output Measures

In the analytical and simulation models, a network architecture with \( N_s \) super nodes and several ordinary nodes is adopted. Each super node \( p \) (\( 1 \leq p \leq N_s \)) supervises \( M_p \) ordinary nodes (\( M_p \geq 1 \)). Without loss of generality, \( N_s = 3 \) is used in the experiments\(^2\). \( ON_{p,q} \) (\( 1 \leq q \leq M_p \)) denotes the \( q \)th ordinary node supervised by the super node \( p \). The propagation delay\(^3\) (\( s_{p,i} \)) of a connection between the super nodes \( p \) and \( i \) (\( 1 \leq p, i \leq N_s \) and \( p \neq i \)) follows an exponential distribution with an average \( S = 200 \) (ms). The assumption of Exponential propagation delay (\( s_{p,i} \)) will be relaxed to accommodate a more realistic distribution (i.e., Gamma) in our simulation experiments [10, 11, 13]. In addition, the propagation delay of a connection between a super node and its ordinary node can be ignored compared that for \( s_{p,i} \). The reason is that in iPTT, super nodes are widely spread out over the Internet while the ordinary nodes normally reside near their super nodes.

At each run of floor determination in iPTT, an exponentially distributed \( r_{p,q} \) denotes the relative time-stamp for each floor request issued by an ordinary node \( ON_{p,q} \) with an average value \( R \). The talk-burst time for the floor owner could be a general distribution, and does not have any influence on the performance of our floor control mechanisms. When TFC is adopted, \( w_p \) denotes the time that the root super-node \( p \) should wait for to collect the floor requests from the other \( N_s - 1 \) super nodes at each floor-determination run. We assume that \( w_p \) is exponential distributed with an average \( W = 200 \) (ms) or 400 (ms).

As to the output measures, the average floor-determination latency is an important metric for our iPTT floor control mechanisms (i.e., FFC and TFC). The latency \( T_l \) is defined as the average time that a floor owner goes through to obtain the floor. In other words, \( T_l \) is the average duration between the time that the owner pushes the button and the time that he/she actually receives an RTCP Floor Granted message. Another important metric is the average signaling message quantity \( H_M \) during the period of a floor contention run.

4.2 Analytical Modeling

This subsection elaborates on our developed analytical model to investigate the performance of FFC and TFC. Specifically, the average waiting times (\( T_l \)) of the floor owner for our FFC and TFC are derived in the following subsections.

\(^2\)In our iPTT system, the number of super nodes is much smaller than that of the members in an iPTT group.

\(^3\)In practice, it means end-to-end delay, including transmission, store-and-forward, and possibly queuing delays.
4.2.1 Derivation of $T_l$ for FFC

Figure 4 shows an example of the FFC timing diagram of the floor owner for the $k$th determination run of an ongoing iPTT session. Assume that $ON_{p,q}$ releases his/her floor of the $(k-1)$st run at the time $\tau_0$, and $ON_{i,m}$ obtains the floor at the coming run. The RTCP Floor Release message issued by $ON_{p,q}$ is received by $ON_{i,m}$ at the time $\tau_1$, where $t_{p,q}$ represents the message propagation-delay from $ON_{p,q}$ to $ON_{i,m}$. $t_{p,q}$ will be 0 if $p = i$. If $p \neq i$, $t_{p,q} = s_{p,i}$. At the time $\tau_2$, $ON_{i,m}$ pushes the button, and sends an RTCP Floor Request message with the relative timestamp $r_{i,m}$ to the super node $i$. Then the super node $i$ forwards RTCP Floor Request to the other two super nodes, and receives the acknowledgements from these super nodes at the times $\tau_3$ and $\tau_4$. The random variables $d_{i,1}$ and $d_{i,2}$ respectively represent the delays for the RTCP Floor Request/ACK message exchange between the super node $i$ and the other two super nodes. Upon receipt of the acknowledgements from all super nodes involved in this iPTT session, the super node $i$ forwards RTCP Floor Granted to $ON_{i,m}$. Finally, $ON_{i,m}$ obtains the floor, and begins to talk to his/her group members. As shown in Figure 4, $ON_{i,m}$ waits for a time period to be the floor owner after pushing the talk button, and the average waiting time $T_l$ can be expressed as

$$T_l = E[\max\{d_{i,1}, d_{i,2}\}]$$

(1)

Based on (1), the derivations of $E[\max\{d_{i,1}, d_{i,2}\}]$ can be divided into two cases. In Case I (i.e., a general case), $d_{i,1}$ and $d_{i,2}$ are respectively the round-trip delays between the super node $i$ and the other two super nodes with the distribution functions $F_{d_{i,1}}(t)$ and $F_{d_{i,2}}(t)$, where $d_{i,1} = s_{p,i} + s_{i,p}$ and $d_{i,2} = s_{i,v} + s_{v,i}$. It is obvious that both $d_{i,1}$ and $d_{i,2}$ follow an Erlang distribution with the average value $2S$. We define a random variable $d_I$ as $\max\{d_{i,1}, d_{i,2}\}$ in this case, and $E[d_I]$ can be
expressed as

\[
E[d_I] = \int_0^\infty [1 - F_{d_{i,1}}(t)F_{d_{i,2}}(t)]dt
\]

\[
= \left(\frac{11}{4}\right)S.
\]  

On the other hand, Case II (i.e., a special case) occurs when any of the two super nodes has not received RTCP Floor Release of the previous determination run upon receipt of RTCP Floor Request of the super node \(i\) at the current floor-determination run. In this case, the super node has to wait for receiving RTCP Floor Release from the floor owner of the previous run. This wait operation avoids malicious iPTT users to contend the floor by advancing their request transmission before the end of the iPTT talk burst of the floor owner. To analyze \(E[\max\{d_{i,1}, d_{i,2}\}]\) in Case II, the following two situations are considered.

**Case II\(_a\):** \(p \neq i\). As shown in Figure 5 (a), the super node \(p\) issues RTCP Floor Release to the super nodes \(i\) and \(v\). Due to the varying propagation delays of the connections, the message first arrives at the super node \(i\). Then in Figure 5 (b), \(O_{i,m}\) requests the floor, and the super node \(i\) sends RTCP Floor Request to the super nodes \(p\) and \(v\). The super node \(p\) responds to the super node \(i\) once it receives RTCP Floor Request. However, the super node \(v\) does not acknowledge the request of the super node \(i\) until obtaining the RTCP Floor Release message. Figure 5 (c) indicates that the super node \(v\) obtains RTCP Floor Release, and sends RTCP Floor Ack back to the super node \(i\). In this case, \(d_{i,1} = s_{p,i} + s_{i,p}\), and \(d_{i,2}\) is expressed as \(s_{i,v} + s_{v,i} + t_v\). \(t_v\) represents the residual time for the super node \(v\) to get the RTCP Floor Release message from the super node \(p\) upon receipt of RTCP Floor Request of the super node \(i\). With an exponentially distributed \(s_{p,v}\), the residual time \(t_v\) will have the same distribution as that of \(s_{p,v}\). Let \(d_{IIa}\) be a random variable \(\max\{d_{i,1}, d_{i,2}\}\) in Case II\(_a\). Then the derivation \(E[d_{IIa}]\) is similar to that in (2), and
\[ E[d_{IIa}] = \left( \frac{55}{16} \right) S. \] (3)

Furthermore, the probability \( P_{IIa} \) that Case IIa occurs is expressed as

\[
P_{IIa} = P_{CIIa} \left\{ \left( \frac{M_i}{\sum_{j=1}^{N_s} M_j} \right) \left( \sum_{p=1}^{N_p} \sum_{j=1}^{N_s} M_p - \sum_{j=1}^{N_s} M_j \right) \right\}. \tag{4}
\]

where \( P_{CIIa} \) denotes the conditional probability that a special event occurs given that \( p \neq i \). From Figure 5, we have

\[
P_{CIIa} = \Pr[s_{p,v} > (s_{p,i} + t_{x_i} + s_{i,v})], \tag{5}
\]

where \( t_{x_i} \) represents the period from the time when the super node \( i \) forwards RTCP Floor Release to its ordinary nodes to the time when the first RTCP Floor Request is received by the super node \( i \) from one of its ordinary nodes. Then we have \( t_{x_i} = \min(r_{i,m}), \forall m, 1 \leq m \leq M_i \), and the density function \( f_{t_{x_i}}(x) \) of \( t_{x_i} \) will be

\[
f_{t_{x_i}}(x) = \left( \frac{M_i}{R} \right) e^{-\frac{M_i x}{R}}. \tag{6}
\]

Based on (6), \( P_{CIIa} \) can be rewritten as

\[
P_{CIIa} = \Pr[s_{p,v} > (s_{p,i} + t_{x_i} + s_{i,v})] = \frac{1}{4} \left( \frac{M_i S}{M_i S + R} \right). \tag{7}
\]

**Case IIb:** \( p = i \). In Case IIb, the previous and current floor owners reside in the same super node. That is, the super node \( p \) issues the release messages to the super nodes \( u \) and \( v \), and then the request messages to these super nodes after \( ON_{i,m} \) pushes the talk button. If any of the super nodes \( v \) and \( u \) receives RTCP Floor Request earlier than RTCP Floor Release, a special event occurs. As shown in Figure 6 (b), this special event occurs at the super node \( v \). Then the super node \( v \) has to wait, and responds to the super node \( p \) immediately after receiving the RTCP Floor Release message (see Figure 6 (c)). Based on the scenario shown in Figure 6, \( d_{i,1} = s_{p,u} + s_{u,p} \), and \( d_{i,2} \) can be expressed as \( s_{p,v} + s_{v,p} + t_{r_v} \). Then we have

\[
E[d_{IIb}] = \left( \frac{55}{16} \right) S, \tag{8}
\]

where \( d_{IIb} \) is the random variable of \( \max\{d_{i,1}, d_{i,2}\} \) in the scenario of Figure 6. Furthermore, if the special situation occurs at both the super
nodes \( u \) and \( v \), \( d_{i,1} \) will be \( s_{p,u} + s_{u,p} + t_{ru} \). Then the average \( E[d_{IIb2}] \) of \( d_{IIb2} \) can be calculated, and will be

\[
E[d_{IIb2}] = \left( \frac{63}{16} \right) S.
\]  

(9)

The probabilities \( P_{IIb1} \) and \( P_{IIb2} \) that Case \( II_b \) occurs are expressed as

\[
P_{IIb1} = P_{CIb1} \left\{ \left( \frac{M_i}{\sum_{j=1}^{N_s} M_j} \right)^2 \right\},
\]

(10)

\[
P_{IIb2} = P_{CIb2} \left\{ \left( \frac{M_i}{\sum_{j=1}^{N_s} M_j} \right)^2 \right\},
\]

(11)

From Figure 6, we have

\[
P_{CIb1} = \{ \text{Pr}[s_{p,v} > (t_{xp} + s_{p,v})] \{1 - \text{Pr}[s_{p,v} > (t_{xp} + s_{p,v})]\} \}
+ \{ \text{Pr}[s_{p,u} > (t_{xp} + s_{p,u})] \{1 - \text{Pr}[s_{p,u} > (t_{xp} + s_{p,u})]\} \},
\]

(12)

\[
P_{CIb2} = \text{Pr}[s_{p,v} > (t_{xp} + s_{p,v})]\text{Pr}[s_{p,u} > (t_{xp} + s_{p,u})],
\]

(13)

Then \( P_{CIb1} \) and \( P_{CIb2} \) can be rewritten as

\[
P_{CIb1} = \left( \frac{M_i S}{M_i S + R} \right) \left[ 1 - \frac{1}{2} \left( \frac{M_i S}{M_i S + R} \right) \right],
\]

(14)

\[
P_{CIb2} = \left[ \frac{1}{2} \left( \frac{M_i S}{M_i S + R} \right) \right]^2.
\]

(15)

Based on (2),(3),(4),(8),(9),(10),and (11) listed above, the average waiting time \( T_l \) can be derived as

\[
T_l = \left\{ \left[ 1 - \sum_{i=1}^{N_s} (P_{IIa} + P_{IIb1} + P_{IIb2}) \right] E[d_I] \right\}
+ \left\{ \sum_{i=1}^{N_s} \{ P_{IIa} E[d_{IIa}] + P_{IIb1} E[d_{IIb1}] + P_{IIb2} E[d_{IIb2}] \} \right\}
\]

(16)
Figure 7: An Example of the Timing Diagram of TFC

4.2.2 Derivation of $T_i$ for TFC

Figure 7 shows an example of the TFC timing diagram of the $k$th floor owner for an ongoing iPTT session. Assume that $ON_{p,q}$ releases his/her floor of the $k-1$st run at the time $\tau_0$, and $ON_{i,m}$ obtains the floor at the coming run. Without loss of generality, it is assumed that the super node $p$ plays the role of the root super node in this example. The RTCP Floor Release message issued by $ON_{p,q}$ is received by $ON_{i,m}$ at the time $\tau_1$. Then $ON_{i,m}$ pushes the talk button, and sends RTCP Floor Request out at the time $\tau_2$. Upon receipt of RTCP Floor Request, the root super node $p$ starts its timer $w_p$ to count down. When the timeout event occurs at $\tau_4$, the super node $p$ has collected the floor requests from the ordinary nodes, and determines that $ON_{i,m}$ obtains the floor. At the time $\tau_5$, $ON_{i,m}$ is informed of that, and begins to talk. To analyze $T_i$ for TFC, the following two cases are considered. If the super node of the floor owner is root, then $T_i$ can be expressed as $W$. On the other hand, when $p \neq i$ (i.e., the floor of the $k$th run is granted to the ordinary node that is not supervised by the root super node $p$), then we have $T_i = 2S + W$. The probability $P_a$ for the case of $p = i$ is derived as follows.

First, $V_j$ is defined as the probability that the relative timestamp of the request from the root super node is smaller than the one of the request from the super node $j$ ($2 \leq j \leq N_S$). Then we have

$$V_j = \frac{M_p}{M_p + M_j}. \quad (17)$$

It is obvious that the root super node will get the floor when the above event occurs. However, if the relative timestamp of the root is larger than that of the super node $j$, the root super node $p$ can still get the floor when the request of $j$ cannot arrive the root super node before timeout. We define $Q_j$ as the probability that the $j$’s request with a smaller relative timestamp can not arrive the root super node $p$ before timeout. $Q_j$ can be derived as

15
\[ Q_j = \Pr[w_p + t_{x_p} < s_{j,p} + t_{x_j} + s_{p,j} \text{, and } t_{x_p} > t_{x_j}] \]

\[ = \frac{M_j M_p}{M_j + M_p} \left\{ \left[ \frac{R}{M_p} + \frac{S^2 R}{(S + W)(SM_p + R)} \right] - \frac{S^3 R^2}{W(S + W)(SM_p + R)^2} - \frac{S^3 R}{(S + W)^2(SM_p + R)} \right\}. \quad (18) \]

Then \( P_a \) can be

\[ P_a = \prod_{j=2}^{N_s} (V_j + Q_j), \quad (19) \]

Finally, \( T_l \) for our TFC can be expressed as

\[ T_l = WP_a + (2S + W)(1 - P_a) \quad (20) \]

### 4.3 Simulation and Numerical Results

This subsection develops a simulation model to investigate the performance of our iPTT floor-control mechanisms. Our simulation program follows the discrete-event model with the input parameters and output measures presented in Section 4.1.

Figure 8 plots \( T_l \) obtained from our developed mathematical analysis and simulation experiments for FFC and TFC. From this figure, the analytical and experimental results are consistent, which indicates that our simulation has been validated against the mathematical analysis. Figure 8 shows the effect of the request arrival rate \((1/R)\) on the floor determination latency \( T_l \) for our flooding-based and tree-based floor control mechanisms. In this figure, we observe that as the request-rate increases, \( T_l \) for TFC decreases and \( T_l \) for FFC increases. For TFC, a large request-rate for an ordinary node results in the increase of the probability that the ordinary nodes supervised directly by the root obtain the floor. On the other hand, more frequent floor-requesting in FFC leads to higher determination overhead, and thus \( T_l \) increases. When \( W = 200 ms \), for all arrival rates under investigation, FFC has a larger \( T_l \) than TFC. When \( W = 400 ms \), \( T_l \) of TFC is larger than that of FFC for \( \frac{1}{R} \leq 0.8 \). However, an opposite result is observed for \( \frac{1}{R} \geq 0.8 \). Figure 8 also indicates that the decreasing rate of \( T_l \) for TFC is larger for a small \( W \) than that for a large \( W \), which implies that the ordinary nodes directly supervised by the root have a short waiting timer and an unfair situation would raise.

Figure 9 shows the effect of the request rate \((1/R)\) on the number \( H_M \) of signaling message exchanges for FFC and TFC. This figure indicates that for all request rates under investigation, TFC has a smaller \( H_M \) than FFC. As the request rate increases, the total number of request messages increases, and hence \( H_M \) of TFC and \( H_M \) of FFC both increase. To further investigate the effect of variances of propagation
delays between the super nodes on average waiting time $T_l$, a Gamma distributed random variable $s_{p,i}$ is adopted. Figure 10 indicates that the waiting time $T_l$ of FFC increases as the variance $v_s$ of $s_{p,i}$ increases. Specifically, the increasing rate is larger for a larger $v_s$ than that for a small $v_s$. For FFC, each floor-requesting super node has to wait for acknowledgements from all other super nodes before obtaining the floor. As $v_s$ increases, there are probably more extremely long propagation delays between the super nodes, which results in the increase of $T_l$ in FFC. On the other hand, we observe that the waiting time $T_l$ of TFC drops slightly when the variance is large. In TFC, the timer of the root super node starts to count down at the time that the first request arrives. A large $v_s$ implies that the first request arrives the root in a very short period, and the timer can be quickly triggered. From this figure, TFC outperforms FFC when the network is in an unstable situation with much varying propagation delays.

5 Conclusion

In this paper, we proposed iPTT, a peer-to-peer Push-to-Talk (PTT) service for Voice over IP. In iPTT, a distributed and mobile-operator independent network architecture was presented to accelerate the deployment of the PTT service. Based on the proposed two-level hierarchical architecture, we presented two floor control mechanisms, FFC and TFC, for real-time talk-burst determination. The performance of the proposed floor control mechanisms was investigated through our analytical and simulation models in terms of the determination latency and the number
Figure 9: Effect of Request Rate on Average Number of Signaling-Message Exchanges

Figure 10: Effect of Variance of Propagation Delays Between Super Nodes on Average Waiting Time
A series of experiments are conducted to show the capabilities of our FFC and TFC.

**Acknowledgements**

Our work was sponsored in part by Excellent Research Projects of National Taiwan University under contract 96R0062-AE00-07, National Science Council under contracts NSC96-2219-E-002-004 and NSC95-2221-E-002-096-MY3, Intel and Information and Communications Research Lab/Industrial Technology Research Institute.

**Biography**

**Jiun-Ren Lin** received his B.S. degree in CSIE from National Chiao Tung University (NCTU), HsinChu, Taiwan, in 2004, and M.S. degree in CSIE from National Taiwan University (NTU), Taipei, Taiwan, in 2006. His current research interests include design and analysis of personal communications services networks, mobile computing, and performance modeling.

**Ai-Chun Pang** received the B.S., M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 1996, 1998 and 2002, respectively. She joined the Department of Computer Science and Information Engineering (CSIE), National Taiwan University (NTU), Taipei, Taiwan, as an Assistant Professor in 2002. From August 2004 to July 2005, Dr. Pang served as an Assistant Professor in Graduate Institute of Networking and Multimedia (INM), and an adjunct Assistant Professor in CSIE/NTU, Taipei, Taiwan. Currently, she is an Associate Professor in INM and CSIE of NTU, Taipei, Taiwan. Her research interests include design and analysis of personal communications services network, mobile computing, voice over IP and performance modeling.

Dr. Pang has served as the program co-chairs and committees of many international conferences/workshops. She was the guest editor of IEEE Wireless Communications, and is currently an associate editor of International Journal of Sensor Networks. Also, Dr. Pang was a recipient of the Teaching Award at NTU in 2005, 2006 and 2007, received the Investigative Research Award from the Pan Wen Yuan Foundation in 2006, and received Wu Ta You Memorial Award from National Science Council (NSC) in 2007.

**Yung-Chi Wang** received the B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chi Nan University, Taiwan, in 2002 and 2004, respectively. He is working toward the Ph.D. degree Graduate Institute of Networking and Multimedia, National Taiwan University, Taiwan. His research interests lie in overlay networks, with an emphasis on building Push-to-Talk system and floor control mechanism that can be deployed on the Internet.
References


