Ad-hoc Voice-based Group Communication

Lu Han, Stephen Smaldone, Pravin Shankar, James Boyce, Liviu Iftode
Department of Computer Science, Rutgers University
{*luhan, smaldone, spravin, jaboyce, iftode*}*@cs.rutgers.edu*

***Abstract*—People waste many hours driving each day. Although unavoidable, this time can be very boring to motorists. Similar to people taking mass transit who often pass the time socializing with those around them, motorists could benefit from social interactions if they were given broader social opportunities. Unfortunately, existing Multiparty Voice Communication (MVC) systems do not scale to large numbers of users and do not provide adequate access controls. We present RoadSpeak, a scalable MVC system that allows motorists to automatically join *Voice Chat Groups* (VCGs) along popular roadways. RoadSpeak achieves interruption-free communication through the use of voice chat message buffering, flow control and in-order delivery of voice messages to participants.**

We have implemented a RoadSpeak prototype on Nokia N95 smart phones using 3G cellular networking for voice message transfer. We have also built an MVC simulator to perform large-scale simulations that compare RoadSpeak with existing MVC systems. The results of our evaluation prove the feasibility of RoadSpeak and demonstrate that it performs similarly to a traditional MVC systems while supporting substantially larger groups of users.

I. Introduction

Motorists, over the years, have tried to socialize in many ways while traveling on roadways. Today, it is quite common to see motorists engaging in cell phone conversations or calling in to their favorite radio talk shows. Alternately, CB Radio provides an avenue for multiple people to socialize, while on the road. For a time, it was very popular, but became a victim of its own success. CB channels were often intolerably noisy making meaningful communication difficult, due to the large numbers of users trying to occupy CB Radio frequencies during the 1970s and early 1980s [1].

Multiparty Voice Communication (MVC) systems, such as CB Radio and modern telephone conferencing systems, enable small groups of people to participate in real-time, simultaneous conversations. The main problem with current MVC systems is that they do not allow meaningful voice-based communication to scale to large numbers of users. This is of particular concern for U.S. highway scenarios, where the typical traffic densities average $40-50$ vehicles per mile per lane under normal conditions and grow to as much as $185-250$ vehicles per mile per lane under peak congestion conditions [2]. To handle social communication on today's congested roadways, MVC systems must be able to scale to large user groups.

Multiparty group formation and discovery is another problem in current MVC systems. For example, CB Radio provides no support for motorists to discover others with whom to socialize. Since CB Radio is broadcast-based and utilizes well-known channels, anyone who is within broadcast range can join a channel and a conversation, without restriction. Unfortunately, open channels are a poor substitute for social group discovery and formation.

Online Social Networks (OSNs) such as MySpace [3], Facebook [4], and LinkedIn [5] allow people with common interests to discover each other, come together, and socialize in groups. Recently proposed by us, Vehicular Social Networks (VSNs) [6], a form of OSN on roadways, take advantage of motorist mobility patterns (e.g., daily commuting of motorists) to discover and form social networks of motorists. In the VSN model, roadways provide a sufficient, local, and regular concentration of people, while a VSN provides the conduit for motorists to form groups and socialize with each other.

In this paper, we present RoadSpeak, a VSN-based system that allows large groups of motorists to socialize and communicate with each other by automatically joining *Voice Chat Groups* (VCGs) formed along popular roadways. Aside from interests, on which traditional OSNs rely, a RoadSpeak group enforces time and location in its profile definition, which may limit membership to specified roadways and time slots. A RoadSpeak VCG profile is defined by a group owner when the VCG is created and users are admitted to the groups based upon matches between user and VCG profiles.

We have implemented a prototype RoadSpeak system. A client runs on a user's smart phone and geo-matches her location with the VCGs in which she is interested. The user is then automatically logged on to one of her groups of interest, and can participate in that group's discussion via voice chat. In a traditional MVC system, *audio collisions* effectively limit the number of active participants in a group, occurring when participants speak simultaneously or when one participant is interrupted by another. Such collisions increase the frustration level of participants, because they slow the rate of progress in the conversation and force participants to repeat their messages at the next opportunity. To enable scaling to large voice chat groups, RoadSpeak allows individual participants to submit voice chat messages in an interruption-free manner. In this way, multiple participants may submit messages concurrently, thereby eliminating au-

978-1-4244-5328-3/09/$25.00 ©2009 IEEE

dio collisions. These voice chat messages are interleaved by the RoadSpeak server and transmitted to all clients. The server implements a flow control mechanism, by stopping clients from sending more messages when the queue is full. The server ensures that messages arrive at clients reliably and in the same order.

To evaluate the performance of RoadSpeak, we have performed a small-scale field trial consisting of four drivers commuting on a common route. We have also built an MVC simulator to compare the performance of RoadSpeak with that of a traditional conference call system. The results of our evaluation demonstrate that RoadSpeak scales to large groups of users beyond what can reasonably be accommodated in more traditional MVC systems, while providing similar system performance.

In summary, we make the following three contributions in this paper:

- We present the design and prototype implementation of RoadSpeak, a VSN-based multiparty voice communication system that enables the scaling of roadway-based voice chat groups to large numbers of users.
- We present the results of an experimental evaluation using our RoadSpeak prototype. These results demonstrate that RoadSpeak completely prevents audio collisions through voice chat message buffering and collision-free communication, thereby enabling substantial improvements to MVC scalability.
- Finally, we describe a novel MVC simulator, developed by us. Using this simulator, we evaluate the scalability and performance of RoadSpeak against more traditional MVC models, and present the results of this evaluation.

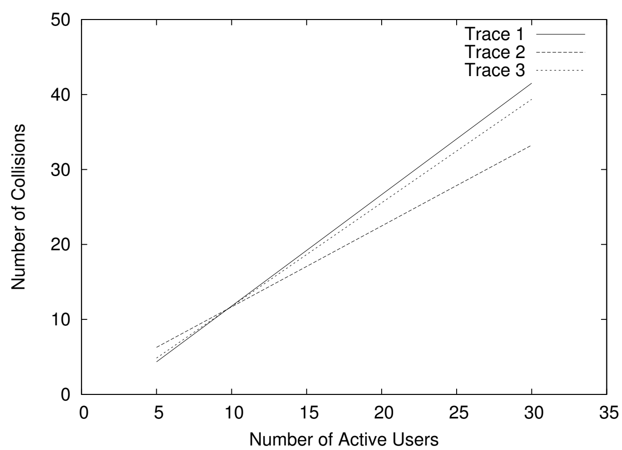
II. Multiparty Voice Communication

Multiparty Voice Communication (MVC) has been extensively investigated over the past 40 years, most recently under Massively Multiplayer Online Game (MMOG) scenarios. In a typical MVC system, speech samples are collected from an audio device, compressed, broken into packets, and transmitted to the receiver. At the receiver, the speech samples are retrieved, decompressed, reassembled, and sent to the audio device for playback.

Today, a number of MVC systems exist as commercially and freely available products. Push to Talk over Cellular (PTT PoC) [7] is a service option for cellular phone networks that permits subscribers to use their phones as walkie-talkies. DT-Talkie [8] enables PTT PoC communications over infrastructure-less and latency-challenged environments by employing a DTN [9] architecture. SimPhony [10] presents a mobile voice communication system built on a PDA, which supports one-to-one or one-to-many communication with voice messages. TeamSpeak [11] and Ventrilo [12] are popular Internet-based MVC systems frequently used under MMOG scenarios.

Most social settings that involve sizeable groups of people are also likely to involve significant numbers of simultaneous speakers. Researchers have revealed significant periods when several participants were simultaneously generating audio traffic [13], [14]. In general, voice patterns in MVC scenarios consist of talkspurts and silence periods. Farber et al. [15] show that talkspurt periods follow an exponential distribution in traditional telephony, while Papp and GauthierDickey [16] demonstrate that talkspurts follow a Weibull Distribution.

To better understand the patterns of collision rates in multi-party communication scenarios, we use three sets of traces from real multi-party IRC (Internet Relay Chat) text-chatting sessions [17].[1] The total size of the traces we analyzed is over 2 MB, consisting of 13.5 hours and 10747 lines of text-chat sessions. For our analysis, we define a *collision* under text-chat to have occurred when multiple users communicate within the same (short-scale) time window (the *collision time window*). Using this definition for a collision, we scan the text-chat session traces sequentially to determine where collisions occur. Figure 1 presents the results of our analysis. For each trace, we plot the regression curve for the number of collisions as we vary the number of actively participating users (denoted as *active users*). Since there is no notion of a session within the text-chat traces, we are unable to determine directly who the active users are. So, we define an *active user* as a person who communicates a minimum of 5 times during a 10 minute period. The collision time window used to generate the results is shown in the figure is 1 second long.



The lines in the figure are the regression curves of the number of collisions that occur in each of the three text-chat traces with a 1 second collision window. The regression curves illustrate the positive correlation between number of active users and text-chat collisions.

Figure 1. Collisions in Text-Chat.

[1] Lacking multi-party audio communication trace data, we use text chatting as a coarse approximator for audio chatting, with respect to the occurrence of collisions.

From the figure, we observe that for each set of traces, as the number of active users scales from 5 to 30, the number of collisions increases from 5 to 40. For each trace set, the plotted regression curve shows a clear positive correlation between number of active users and collisions. Although this may be acceptable in a text-chat scenario, such a trend would quickly become intolerable in a traditional MVC scenario.

III. Functional Overview

To illustrate the RoadSpeak concept, this section provides an example RoadSpeak scenario. This scenario is only one of a number of possible use cases and it is chosen to demonstrate how a typical user, Joe, might utilize RoadSpeak. [2]

Joe, a RoadSpeak user who currently participates in a Voice Chat Group (VCG) during his afternoon commute home, would like to start participating in a new VCG during his daily morning commute. From his home computer, he goes to the RoadSpeak portal, logs on using his account, and browses through the various VCGs available along the route he takes between home and work. He chooses to join a sports talk group active on his work route between 6 AM and 8 AM, since he is usually on the highway from 6:30 AM - 7:15 AM. Finally, he verifies that he has the most recent version of the RoadSpeak client installed on his smart phone.

The next morning, Joe gets into his car, sets his smart phone in its cradle, and puts on his hands-free kit. As he travels to work, the RoadSpeak client running on his smart phone tracks his location via the built-in GPS receiver, and when he enters the highway at 6:32 AM, the client contacts a RoadSpeak server and automatically adds Joe to the VCG. Joe receives an audible alert from his phone, notifying him that he is joining the chat group. He introduces himself to the group and begins listening to the current sports talk voice chat messages that his RoadSpeak client receives from the server. As he listens, he is happy to note that he is familiar with a number of the other participants who also belong to his regular afternoon VCGs.

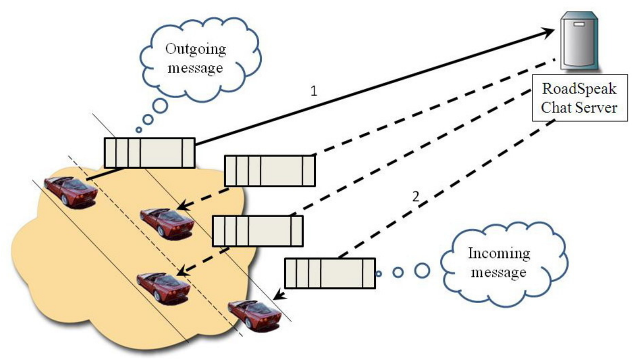
After a while, Joe intervenes and speaks to the group. The RoadSpeak client captures and transmits his voice message to the server. As the server continues to send out messages in the sequence it receives them, it sends Joe's message to the other participants in the group. Fran and Harry both hear Joe's question to the group and respond, about the same time. Under a typical MVC system, this would lead to an audio collision. Under RoadSpeak, it does not, since the messages are buffered at the clients and sent to the server once completed. The server serializes the messages into a sequential ordering and transmits them individually to the other group participants. Finally, as Joe leaves the highway, the RoadSpeak client running on his smart phone detects his departure and notifies him that he will leave the sports talk chat group, shortly. Joe transmits a farewell message to the group and continues along his route to work. [3]

IV. System Design

A. RoadSpeak Overview

Figure 2 illustrates the paths of voice chat messages in RoadSpeak. RoadSpeak clients send voice messages to the server over the network. The RoadSpeak server buffers incoming messages and transmits them to the other VCG participants. This buffering of messages at the server decouples the sender from the receivers, such that once a message has been submitted by a speaker, it is guaranteed to reach any actively logged in clients, ensuring message delivery.

The overall design of RoadSpeak is guided by two goals. The first goal is to *increase the scalability of MVC by providing interruption-free communication*. The second goal is to *enable a consistent user experience by enforcing fair and equal access to the voice chat channel*.



RoadSpeak clients send voice messages to the server through a wireless network (1). The server, after receiving messages from clients, transmits the messages to all clients in the voice chat group (2).

Figure 2. RoadSpeak Overview.

RoadSpeak provides interruption-free multiparty voice communication by enabling voice chat participants to submit messages whenever they like, and buffering voice chat messages. Once submitted, these messages are sequentially ordered by the RoadSpeak server, and delivered to RoadSpeak clients in order, for playback. With RoadSpeak, there can never be audio collisions.

To enable a consistent user experience, RoadSpeak enforces quotas on participant speaking time. This ensures that all participants are allowed a fair share of the voice chat channel, preventing any one user from dominating a group. A Quota Manager controls both a user's total speaking time and the individual lengths of each message sent by a user. RoadSpeak also provides an application-level flow-control

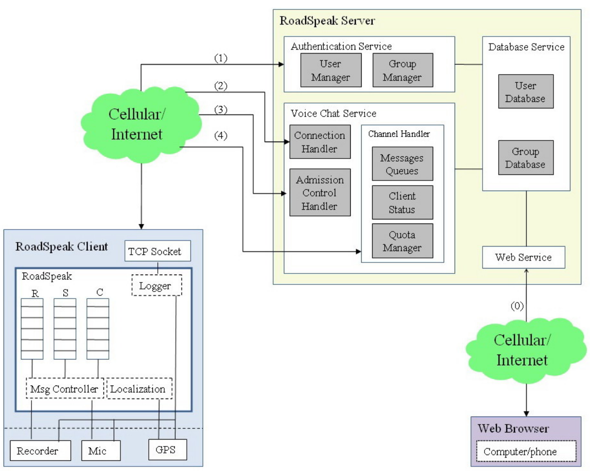
[2]Beyond this example, there are many other possible scenarios in which RoadSpeak could be found useful to a motorist (e.g., during an emergency, or to communicate interesting observances, etc.).

[3]In addition to the scenario described above, Joe could also join a VCG of interest while he is at his home or workplace, if the VCG (optionally) does not enforce location.

mechanism. From the perspective of the RoadSpeak server, this flow-control mechanism is used to adjust the message rates of clients. From the client perspective, flow-control allows a participant to discard recently created messages, prior to transmission.

B. RoadSpeak Architecture

The RoadSpeak architecture is shown in Figure 3 and is composed of two main components: the RoadSpeak server and client. The server handles authentication, access control, flow control, and message delivery. The client software, once downloaded and installed on a smart phone, handles all message capture and playback functions. Finally, there is a web portal through which a user connects to organize the groups they own, create new groups, and join existing groups owned by others. In the following subsections, we will describe each component in more detail.



RoadSpeak clients can use their browsers to access web server (0). Authentication service handles user authentication (1). (2), (3) and (4) handle group admission and chatting procedure.

Figure 3. RoadSpeak Architecture.

C. RoadSpeak Smart Phone Client

The RoadSpeak client was developed to be downloaded to smart phones, carried by users. The client software, after a successful log in, periodically reads a user's GPS location and automatically joins the user into an appropriate group, based upon the set of groups she has selected (from the web interface), her present location, and the current time. To do so, RoadSpeak performs geo-matching, which is further discussed in Section VIII-B.

The RoadSpeak client maintains three message queues, as shown in Figure 3: *(i)* the send queue, *(ii)* the receive queue, and *(iii)* the control queue. These queues are used to buffer outgoing voice messages, incoming voice messages, and flow control messages, respectively. As in other MVC systems, voice audio data is captured by the client through the smart phone microphone. Unlike other MVC systems, though, RoadSpeak buffers the voice data as messages, which are placed into the send queue to be reliably sent to the RoadSpeak server. This also allows a user to discard a newly captured message, prior to transmission. Any new voice messages received by a client are placed on the receive queue. The Voice Playback thread selects messages from the receive queue for playback to the user based upon the server-determined message playback schedule. This ensures that messages are played back to all users in the same order. Additionally, the RoadSpeak server can send *pause* and *resume* messages to clients. These higher priority messages are placed in the client control queue. A pause directs a client to cease all message sending to the server and a resume notifies the client to begin sending, once again.

D. RoadSpeak Server

The RoadSpeak server includes an Authentication Service and a Voice Chat Service, which are also shown in Figure 3. All persistent users and group data is stored in a database and accessed by both the Authentication Service and the Voice Chat service.

Authentication Service: Users can browse the RoadSpeak web server to create an account in the system ((1) in Figure 3) and log in. Both user account creation and login authentication are handled over a secure web protocol (i.e., HTTPS) by the User Manager.

A voice chat group is defined by its location and time interval (e.g., 10AM-12PM). Groups also have specific characteristics, such as public or private, open or moderated, and hidden or advertised. Public groups are open to all users, while private groups are restricted via access control. In an open group, members are free to chat, but in a moderated group, the group owner can restrict who may submit audio messages to the group. A hidden group is not publicly visible, while advertised groups are visible to all. Admission to a group is handled by the Group Manager.

Voice Chat Service: When a RoadSpeak client connects to a RoadSpeak server via TCP/IP ((2) in Figure 3), the Connection Handler spawns a new Admission Control Handler (ACH) and hands the connection to the ACH to perform Admission Control ((3) in Figure 3). Once the user drives within the location of a group she has previously registered with, the ACH hands the client connection off to the Channel Handler for the requested group ((4) in Figure 3).

A Channel Handler spawns one thread per client connection. All Channel Handlers share a global data structure, called the Message Queue. Messages in the Message Queue are maintained in the order in which they were received and are sent out to clients in order. The Receiver Handler puts each newly received message into the Message Queue and the Channel Handler is responsible to send all messages on the Message Queue to its associated client.

V. Implementation and Field Trial

A. Implementation

The RoadSpeak server is developed in Java and the client is a J2ME Midlet for Nokia N95 smart phones. The client can use either 3G or WiFi connectivity to access the server. The web portal provides group and user maintenance, and management functionality, and is developed using PHP and Javascript, for the Apache2 web server. RoadSpeak groups associated with a certain region are returned to a visualization service that displays the regions using the Google maps API with the JMaki toolkit. Screenshots of the web portal and the smart phone client are presented in Figure 4(a) and Figure 4(b), respectively. [4]

(a) Web interface screenshot of a user defining a new VCG.

(b) Client software executing on a Nokia N95 smart phone.

Figure 4. RoadSpeak Server and Client Interfaces.

B. Field Trial

We perform a field trial with real users, to better understand the system performance and human interactions in RoadSpeak. Our field trial consists of four participants (colleagues in our lab), each of whom was equipped with a Nokia N95 smart phone with 3G Internet access. The participants were briefly trained on how to use the RoadSpeak system. The experiment was performed when the whole group went to a restaurant for a celebratory dinner. While driving to the destination, the participants used RoadSpeak for communicating to each other.

Dataset Characteristics: The entire experimental trace consists of 72 voice messages over a duration of 1200 seconds. The combined speaking time (silence periods removed) is 514 seconds, and the average number of messages per user is 18. The maximum message length is 27 seconds and the average message length is 7 seconds, and most messages are between 4 and 7 seconds long.

To better understand human conversation patterns in the RoadSpeak system, we plot the speaking and playback message durations for the field study trace. Figure 5 presents the time series plot for the trace period of 800 to 1200 seconds. We choose this period since it is the most active period of the trace. For each individual graph, the y-axis represents the user identifier number, e.g. a peak at $y = 1$ denotes that user 1 is speaking. When $y = 0$, all users are silent. Finally, the elapsed session time is shown along the x-axis. Figure 5(a) presents the speaking time for all users, while each of the four subplots of Figure 5(b) presents the speaking time for an individual user ($y = -1$) and the message playback times ($y > 0$) for all users.

From Figure 5(a), we observe that audio collisions occur whenever multiple lines overlap. Although this is only a four-user experiment, we observe that there are numerous audio collisions. As we will demonstrate through simulation in Section VII, when the number of users increases beyond four, audio collisions occur much more frequently.

From Figure 5(b), we observe the beneficial effects of collision-free communication provided by RoadSpeak. For the $y > 0$ section of each subplot, we see that there are no occurrences of overlapping lines. We also observe, by comparing the $y > 0$ and $y < 0$ sections of each subplot, that users are free to submit new voice messages, without causing interruptions to the flow of the voice-chat session.

VI. RoadSpeak Evaluation

The goal of this evaluation is to study the performance of RoadSpeak, addressing the following two questions:

- What is the base performance of RoadSpeak under typical client workload with different network connectivity (Section VI-A)?
- How does the performance of RoadSpeak scale as more clients are added to the system, under more realistic workload (Section VI-B)?

We perform two types of experiments in evaluating RoadSpeak: single client microbenchmarks and multiple client emulation. For our emulation platform, we used a Intel Dual Core 1.86GHz Linux PC with 1 GB of RAM.

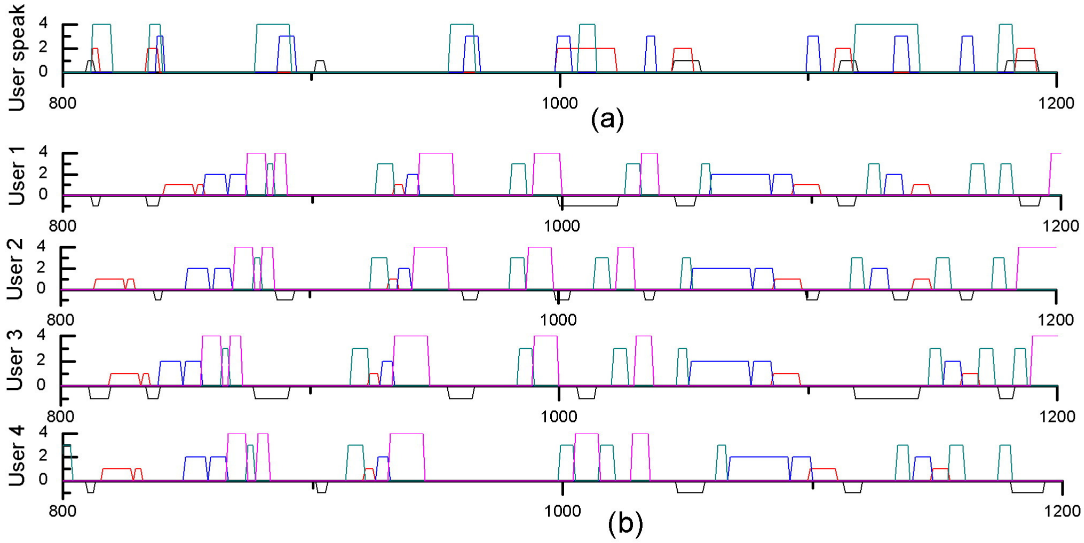
A. Single Client Microbenchmarks

The single client microbenchmark experiments measure the performance of RoadSpeak running over a 3G cellular network (Verizon 1xEVDO), as well as a WiFi network (802.11g). We define the end-to-end message transmission latency as the time taken to transmit a message from one RoadSpeak client to another RoadSpeak client through the RoadSpeak server. We measure the latency to transmit a 1-second long message, and a 5-second long message. Each result reported is the mean of 10 measurements. The average message transmission latency over WiFi and 3G network connections is reported in Table I.

B. Multiple Client Emulation

For the multiple client emulation experiments we use a set of scripted RoadSpeak clients with synthetically generated

[4] RoadSpeak server and client software distributions are available for download at http://www.roadspeak.com.



Visualization of a four user RoadSpeak conversation. The x-axis is elapsed time (sec). In (a), a line with $y = n$ represents the time when User n is speaking. In (b), a line with $y = n$ represents message playback time from speaker n, and $y = -1$ represents when the respective user speaks.

Figure 5. RoadSpeak Conversation Visualization.

	Message Length	
Connection Type	1 Second	5 Seconds
WiFi	0.03(0.00)	0.12(0.01)
3G	1.24(0.01)	2.95(0.34)

Message transmission latency in seconds over WiFi and 3G for differing message lengths. Standard deviations are reported in parentheses.

Table I
MESSAGE TRANSMISSION LATENCY (MICROBENCHMARK).

workload (around 5 messages per minutes), to evaluate how the server and client queues scale. We ported RoadSpeak to PCs for this set of experiments, and emulate 3G and WiFi network connections by injecting a synthetic delay before transmission and reception of every message.

In the first experiment, we fix the number of speaking clients to 10. From Figure 6, we observe that the effect of message length on the overall message transmission latency is linear. Apart from *short*, *normal*, and *long* messages, we also consider a *mixed* workload scenario where each message is chosen to be short, normal, or long with equal probability. Since long messages dominate the transmission time, the mixed workload scenario shows similar results as the long message scenario. Message transmission delay can be further broken down into four components: *(i)* server queue delay, *(ii)* client send queue delay, *(iii)* client receive queue delay, and *(iv)* propagation delay.

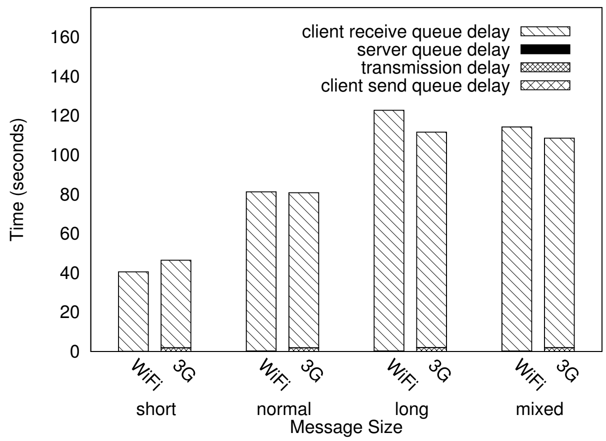
To evaluate the impact that increasing the number of speaking clients has on message transmission latency, we fix the message length to the worst case value, i.e., long messages, and vary the number of clients from 2-30 speakers. From Figure 7, we observe that message latency initially grows linearly until the number of users is increased to 10, then grows sub-linearly. Since the rate of message playback for a client is fixed, once the workload approaches this capacity, system performance stabilizes.

Finally, from both Figures 6 and 7, we observe that the overall message transmission time is dominated by the client receive queue time. This is the duration of time that messages are buffered in the receiver before playback. The RoadSpeak prototype allows clients the ability to control the client message queue in two ways. First, it allows a sender to *cancel* a message, if the sender realizes that she no longer wants to send out a particular message. Second, a receiver may *skip* a message, for example, an uninteresting message or if the receiver's queue is too long. The results in Figures 6 and 7 represent the worst case system performance, and clients never *cancel* or *skip* messages.

From the results of these experiments, the client receive queue delay appears, at first, to be very large when compared to a what a person would experience in a typical conference call. In fact, though, when comparing RoadSpeak queue times and conference call waiting times, we have to also consider that a conference call does not allow a users to concurrently submit messages. Instead, a user will buffer her messages in her head, while waiting for her turn to speak. RoadSpeak, on the other hand, allows a user to speak her messages as they occur to her, and edit them while they are still buffered in queue. In Section VII-B, we show a comparison between conference calls and RoadSpeak in terms of the time it takes to have a message heard by all other parties on the call, which we believe to be the most relevant metric to understand the actual user experience.

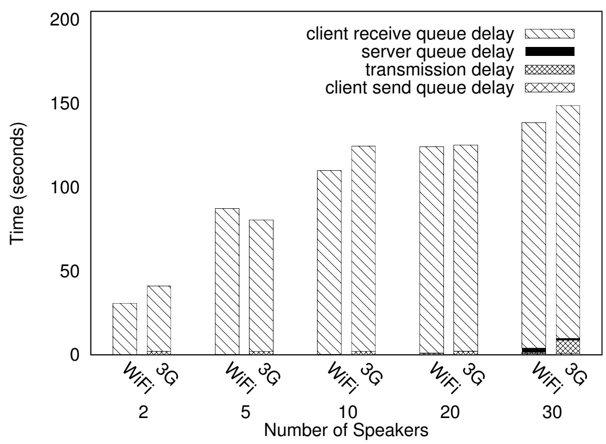
VII. SIMULATION STUDY

The performance of RoadSpeak depends on various factors, including the number of users, behavior characteristics of users, human-driven flow control and system-initiated



Message transmission latency (in seconds) over 3G and WiFi as message length is varied. Group size is fixed to 10.

Figure 6. Effect of Message Length on Transmission Latency.



Message transmission latency (in seconds) over 3G and WiFi as group size is varied. Message length is set to *long* messages.

Figure 7. Effect of Group Size on Transmission Latency.

flow control. To better understand the impact of each of these factors, we build a model of human conversation to feed into an MVC simulator. We use this simulator to compare the performance of RoadSpeak with traditional MVC systems, which we refer to as conference calls. Note that we do not simulate network conditions since we want to compare the base-case (ideal) performance of RoadSpeak with other multiparty voice communication systems.

A. MVC Simulator

There are two approaches to building an MVC simulator, a micro-simulator that takes into account each participant's speaking pattern and interactions among the participants, or a macro-simulator that uses aggregate statistics about the conversations. We build a micro-simulator because we are interested in the effects of interruption-free communication on the experience of individual users.

Simulation Scenario	Message Length	
	Conference Call	RoadSpeak
Talkspurt	$\lambda = 2.30$, $k = 1.18$	$\lambda = 7.6$, $k = 2.6$
Silence	$\lambda = 13.53$, $k = 0.60$	$\lambda = 58.0$, $k = 1.7$

Weibull distribution parameters for RoadSpeak and Conference Call simulations.

Table II
SIMULATION PARAMETERS.

1) Simulation Parameters:

- *The talkspurt period:* (ON period, or message length) This parameter represents the length of talkspurt and follows a Weibull distribution. A talkspurt period follows and is followed by a silence period.
- *The silence period:* (OFF period) This parameter represents the length of each silence period and also follows a Weibull distribution. A silence period similarly follows and is followed by a talkspurt period.
- *Aggressiveness:* This is an adaptive parameter that defines the willingness of a speaker to intervene in the current conversation. Each speaker is given an initial aggressiveness probability generated as a uniformly random value between 0 and 1. If a speaker loses a collision round, her aggressiveness is increased as a function of her current aggressiveness. Once a speaker completes her current message, her aggressiveness is reset to its original value.

Talkspurts and silence period modeling: Prior research on multiparty communication suggests that the distribution followed by talkspurts and silence periods follows a Weibull distribution [16]. We analyzed the logs from our field trial (described in Section V-B) and confirmed that the talkspurts and silence periods fit a Weibull distribution. We use the parameters from Papp and GauthierDickey [16] for conference calls, and compute the best-fit parameters empirically from our field trial for RoadSpeak. Table II lists the best-fit parameters we derived for talkspurt and silence periods, as well as the parameters we use for conference calls. The model shows some interesting trends in human conversation patterns while driving. For example, 90% of talkspurts are shorter than 11 seconds, whereas 90% of silence periods are shorter than 117 seconds. Silence periods are much longer than talkspurts because during driving, people need to concentrate on the road ahead, watch traffic lights, etc.

A simulation begins by defining *number_of_speakers* in the conversation. Each speaker is assigned an *aggressiveness* value. The length of talkspurt and silence are generated from a Weibull distribution. We assume that at any second, only one speaker can occupy the channel.

2) Simulation Scenarios: RoadSpeak, RoadSpeak w/o Flow Control, and Conference Call are simulated.[5]

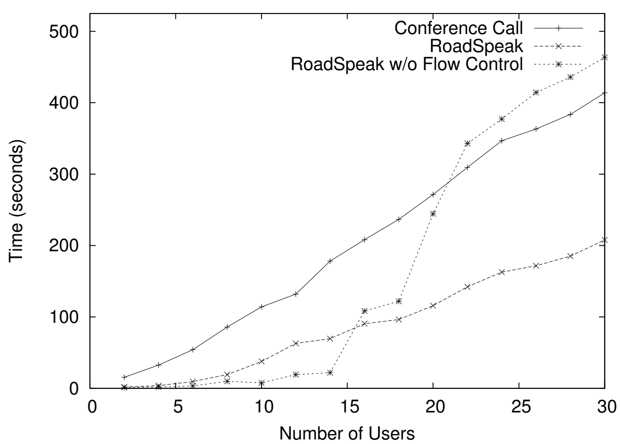
In a conference call scenario, when Joe intends to speak,

[5]For simplicity, we use Joe and Tom to represent our two speakers.

he first checks the channel to see if another user is speaking. If Tom is speaking, Joe checks if he is more aggressive than Tom. If so, he stops Tom and starts speaking, otherwise he goes back to waiting until he has the next chance to speak. It is unlikely for someone to interrupt a speaker when he has just started to speak. We therefore define a parameter *speaking_constraint_time*, which allows the current speaker to continue speaking for *speaking_constraint_time* seconds before he can be interrupted by other speakers.

In each scenario, we vary the number of speakers from 2 to 30. For each test, we mark half of the speakers as aggressive speakers and the other half as passive speakers. We chose 30 as the maximum group size since the conversation patterns are revealed at and beyond 20 speakers per group. In the Conference Call scenario, more than 80% of the voice chat messages were interrupted by others when the number of speaker in the group is beyond 10, which makes it very difficult for a speaker to complete a message.

B. Simulation Results

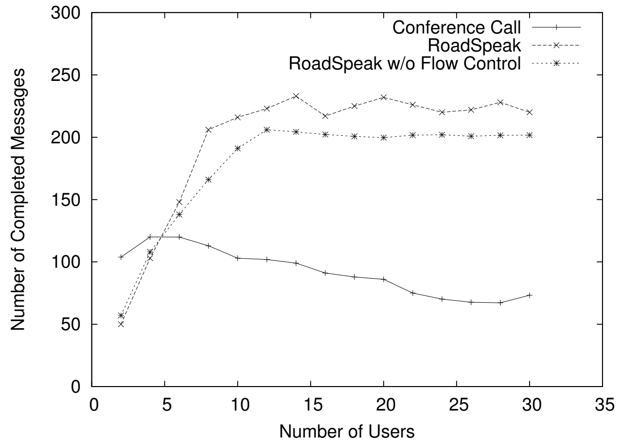


Message completion time (in seconds), as group size is varied for: Conference Call, RoadSpeak, and RoadSpeak w/o Flow Control.

Figure 8. Message Completion Time.

From Figure 8 we observe that, for all three scenarios, as the number of speakers increases, message completion time increases. The time for RoadSpeak, as well as Conference Call, increases linearly, while RoadSpeak w/o Flow Control increases dramatically after the number of speakers in the group exceeds 15. RoadSpeak performs better than Conference Call in all cases. RoadSpeak also performs better than RoadSpeak w/o Flow Control beyond 15 speakers, due to the fact that flow control enables stable scaling, preventing server overload and subsequent performance degradation.

Finally, we compare the message completion rates for the three scenarios in Figure 9. From the figure we observe that, as the number of speakers in a group increases, both RoadSpeak cases demonstrate a higher message completion rate



Message completion rate for: Conference Call, RoadSpeak, and RoadSpeak w/o Flow Control.

Figure 9. Message Completion Rate.

than Conference Call (beyond 5 speakers). As the number of speakers increases, the rate of audio collisions for Conference Call also increases, causing the message completion rate to drop dramatically. RoadSpeak also consistently outperforms RoadSpeak w/o Flow Control, maintaining a higher message completion rate, throughout. Finally, the message completion rate for both RoadSpeak cases is ultimately limited by the size of the client receive queue capacity, once the rate of concurrent message generation exceeds 60 seconds worth of messages generated per minute.

VIII. DISCUSSION

A. RoadSpeak vs. Conference Call Systems

RoadSpeak allows interruption-free communication, and employs voice message buffering and flow control to enable the system to scale to large groups. On the other hand, classical MVC systems, such as conference call systems, allow audio collisions, and shift the onus onto the speakers (or a human moderator, if available) to resolve these collisions. There is an inherent tradeoff between RoadSpeak and conference call systems. The interruption-free model employed by RoadSpeak can often result in out-of-order conversations, since old messages may remain in the system. This is because the RoadSpeak server, unlike a human moderator, cannot understand the semantics of a message. Conference calls, on the other hand, do not scale well and can experience high collision rates with large groups. We believe that this tradeoff brings RoadSpeak and conference call systems to different domains. RoadSpeak is more applicable for entertainment or socializing purposes by large numbers of users, while conference call systems are more suitable for business usage with a limited number of participants.

B. Geo-Matching

Motorists need to focus on the road while they are driving, so the system cannot expect users to manually select a VCG group. To aid a motorist, geo-matching and spatial-temporal enforcement mechanisms are used to: *(i)* provide automatic group joining, and *(ii)* only let valid users currently driving on the proper roadway join a group constrained to that roadway. When a user logs into the RoadSpeak system from her smart phone, the RoadSpeak client connects to the server to retrieve the waypoints for each group that user has registered and stores the waypoints in an R-Tree in local smart phone storage. Periodically, the RoadSpeak client reads GPS locations and compares these coordinates with the waypoints stored in the R-Tree to determine if the motorist's current location matches the spatial-locality preferences of any of her preferred groups. When a match occurs, the client automatically joins the user to the group.

C. Spam

Unwanted messages such as advertisements and spam is another potential issue in RoadSpeak. Reputation systems are often useful in large online communities in which users may frequently have the opportunity to interact with other users they have no prior experience with. In such a situation, it is helpful to base the decision whether or not to interact with that user on the prior experiences of other users. Reputation systems may also be coupled with an incentive system to reward good behavior and punish bad behavior. RoadSpeak can establish and utilize a reputation system to prevent unwanted messages. Users of RoadSpeak can be assigned with reputations at the beginning. Low ratings from other users may diminish his reputation. When the reputation of a user is below a threshold, RoadSpeak should not publish the messages from him, or should revoke his admission to the group. We leave the addition of a reputation system for RoadSpeak as future work.

D. Driver Distraction

Although it is important to consider the safety of drivers who use cell phones while driving, we believe that, with advances in automated vehicular control (e.g., automatic cruise control, obstacle avoidance, etc.), this issue will be diminished in the near future. We consider detailed discussion about this topic to be out of the scope for this paper.

IX. Conclusion

RoadSpeak is a mobile multiparty voice communication system that allows motorists to socialize with each other along popular roadways based upon their interests. We implemented a prototype of RoadSpeak for smart phones, as well as a custom MVC simulator for comparing the scalability of our system with traditional MVC systems. Our experimental results show that the use of voice chat message buffering and interruption-free communication allows RoadSpeak to scale to large groups of users.

X. Acknowledgement

We thank the anonymous reviewers for their insightful comments and suggestions. This work is supported in part by the National Science Foundation through grant numbers CNS-0831268 and CNS-0520123.

References

[1] "Citizens' Band Radio." [Online]. Available: http://en.wikipedia.org/wiki/Citizens_Band_Radio

[2] "Trafficflow." [Online]. Available: http://en.wikipedia.org/wiki/Traffic_flow

[3] "MySpace." [Online]. Available: http://www.myspace.com

[4] "Facebook." [Online]. Available: http://www.facebook.com

[5] "LinkedIn." [Online]. Available: http://www.linkedin.com

[6] S. Smaldone, L. Han, P. Shankar, and L. Iftode, "RoadSpeak: Enabling Voice Chat on Roadways Using Vehicular Social Networks," in *SocialNets'08: Proceedings of the 1st Workshop on Social Network Systems*, 2008.

[7] "Push to talk over cellular." [Online]. Available: http://en.wikipedia.org/wiki/Push_to_talk

[8] M. T. Islam, A. Turkulainen, T. Karkkainen, M. Pitkanen, and J. Ott, "Practical Voice Communications in Challenged Networks," in *1st Extreme Workshop on Communication*, 2009.

[9] "DTN." [Online]. Available: http://en.wikipedia.org/wiki/Delay-tolerant_networking

[10] V. Lakshmipath and C. Schmandt, "Simphony: A voice communication tool for distributed workgroups," *BT Technology Journal*, vol. 22, no. 4, pp. 52–61, 2004.

[11] "Teamspeak." [Online]. Available: http://www.teamspeak.com

[12] "Ventrilo." [Online]. Available: http://www.ventrilo.com

[13] J. Bowers, J. Pycock, and J. O'Brien, "Talk and embodiment in collaborative virtual environments," in *SIGCHI Conference on Human Factors in Computing Systems*, 1996.

[14] P. Boustead, F. Safaei, and M. Dowlatshahi, "DICE: Internet Delivery of Immersive Voice Communication for Crowded Virtual Spaces," in *VR'05: IEEE Virtual Reality 2005 Proceedings*, 2005.

[15] J. Farber, "Network Game Traffic Modelling," in *1st workshop on Network and System Support for Games*, 2002.

[16] G. Papp and C. GauthierDickey, "Characterizing multiparty voice communication for multiplayer games," in *SIGMETRICS'08: International Conference on Measurement and Modeling of Computer Systems*, 2008.

[17] "Freenode IRC." [Online]. Available: http://freenode.net/