



Re-thinking voice communication

Why phone calls, voicemail and current radios are not enough

And

The need for a Multiple Conversation Management System

**by
Tom Katis and Matt Ranney
CommoTek, LLC**



The CommoTek Multiple Conversation Management System

Description

The core application creates a new mode of communication that spans the continuum from real-time to an archived and organized record of all communications designed for easy access and retrieval on the fly. It is a system that supports both synchronous and asynchronous communication seamlessly and allows conversations to transition between asynchronous and synchronous.

The core mission of the system is to enable individuals to manage a large number of active conversations as efficiently and effectively as possible – to do for voice and video communications what email, instant messaging, and devices like the Blackberry have done for written correspondence. Users can send quick voice or video messages at the click of a button to any number of users without necessarily interrupting other conversations or waiting for acknowledgement.

Critical Components – The primary categories:

1. A user interface to organize and prioritize current and past conversations, manage contacts and groups, and manage that user's interaction with the system
2. A protocol for the interaction of devices on a network which manages multiple simultaneous half-duplex and full-duplex, real-time and asynchronous messages with appropriate hierarchies of priority and authority
3. Stream and store protocol – streaming voice, video or other media as quickly as possible while still guaranteeing delivery, even when network conditions are poor
4. Gateways and interconnection points for interoperability with the Public Switched Telephone Network, tactical radio systems, VoIP systems, instant messaging, or other current communications systems
5. A robust cryptographic privacy and security system offering fine-grained control over viewing and authenticity of all communications
6. A communication system that combines some or all of the above components



Context

Much has been made of the transition from old circuit-switched telephone networks to VoIP (Voice over Internet Protocol) telephony. No longer does a telephone company need to have a dedicated line for two parties to talk with each other. Voice and video can be turned into a stream of data packets and sent over the Internet without a dedicated infrastructure for that call. However, this leap forward in design efficiency does not change the way calls are made at a human level. People still pick up a phone (or sign into Skype) and engage in a full-duplex, synchronous conversation.

It is the ambition of this paper, and of CommoTek LLC, to rethink and redesign this human aspect of communication and to design a technical infrastructure to enable it.

Today people have a relatively narrow range of options for interpersonal voice communication. Some of these include:

- Shouting (full-duplex, synchronous)
- A telephone phone call (full-duplex, synchronous)
 - Land line telephones
 - Mobile/Cellular telephones
 - Voice over IP/Skype
- Voicemail (one-way, asynchronous)
- Push To Talk, radio-like systems (half-duplex, synchronous)
 - Nextel Direct Connect
 - Tactical Radios (police, military, first responder, etc)

There is a much broader array of interpersonal text options, including:

- Written correspondence
- Email
- Text messaging (SMS)
- Telex and teletype
- Instant Messaging
- Message Boards
- Chat Groups
- Etc.

It is possible that the diversity of written messaging options is a result of the keyboard being the primary input device of personal computers (PCs). Regardless, we believe there are gaps in the currently offered forms of voice communication, and we offer an alternative.



Just as Instant Messaging (IM) has similarities to Email but is also a different and distinct form of messaging, the proposed system has similarities to some current technologies but constitutes a different, distinct, and new form of messaging.



Limitations of current systems

Radios:

Current radio systems allow for multiple parties to tune their radios to pre-determined channels to communicate between the individuals. This system can be effective, but it has some clear inefficiencies and limitations. If everyone is on the same channel, then everyone can hear everything that is going on. If the group gets too big, or if there are competing priorities, then communications on the channel become distracting and inefficient. With these systems, only one party may transmit at a time. This leads to problems when two parties attempt to transmit simultaneously, or if one party transmits accidentally, by leaning against the transmit button, for example.

To address these shortcomings, organizations (police, military, fire, etc...) typically assign separate channels to different teams or groups. This helps to make the communication of that team more efficient between the team members, but it also separates and isolates them from other teams or groups, making it very difficult to share information and communicate between groups. It typically necessitates leadership having multiple radios to communicate and coordinate between groups.

Radios also require synchronous conversations. All parties need to be actively listening and participating. If a radio user doesn't hear a message clearly for whatever reason, he or she has no choice but to transmit to the entire channel asking for the message to be repeated.

Telephones and Voicemail:

Making a successful telephone call to someone involves the establishment of a live, synchronous, full-duplex conversation. If one party is not available for a dedicated conversation, the call will not be successful. Calls can be one-to-one or conference calls. It takes time to initiate a call, wait for a person to hear the phone, answer the call, exchange greetings, and begin a conversation.

Voicemail adds the ability to leave an asynchronous message for someone when an attempted phone call wasn't successful. However, the overall process of delivering this one-way voice message is very time-consuming. The calling party spends substantial time waiting for the call to be set up, more time for the called party to be notified, and still more time to listen to the called party's greeting. The called party then must establish a call to his or her voicemail system, often requiring password authentication, then navigate through a series of voice prompts to receive the original communication.



How the CommoTek solution is different:

This is a new system and method for engaging in conversations and/or managing multiple simultaneous conversations using a variety of media types (voice, video, text, location, data, etc.). Users can engage in conversations by sending short voice messages to designated recipients at the click of a button. Depending on preferences and priorities, the recipient might hear the message in real time, or they might simply be notified that a message is ready for retrieval. Users could transition from a series of back and forth transmissions to a synchronous, full duplex conversation and then back again.

Users of the system have the control to interact with their most important conversations in real-time without missing any messages from their other conversations. Users can be notified of urgent or high priority communications from a conversation that they are not currently listening to. All messages from all conversations are stored for retrieval and review at any time. The system is designed to make the sending of messages and their retrieval easier, faster and better organized and to allow an easy transition from sending messages to a full conversation and back.

Features:

- All media that is sent or received by the device is recorded for time-shifting, conversation management, and archival purposes. Media is available for listening or viewing while it is being received, if desired.
- Sent messages may be delivered asynchronously if the recipient is unavailable or network conditions are poor at the time.
- If the priorities and preferences are appropriate, a message will be heard or displayed in real-time by the recipient.
- If the priorities or preferences are set differently, the message will be available for listening or viewing at a later time.
- Each conversation is allowed to send media to a receiving device at any time, and the receiving device associates these messages with conversations appropriately, regardless of the current actions of the receiving user.
- Every device is technically capable of talking to all of the other devices on the network.
- The network could be the Internet, but it could also be any number of other types of network including but not limited to a cell phone network, a private network, etc.
- Groups of users can be designated for group conversations (both synchronous and asynchronous).
- A user can assign priorities and preferences for his/her interaction with other users and groups.
- A user can designate a priority level for a message (e.g. designate a message as urgent).



- Messages are organized into conversations. Conversations can spawn sub-conversations in a hierarchy. Users, Messages, and Conversations all have associated priorities. The receiving user can use these priorities along with their own preferences to build a completely prioritized Conversation Management System
- All Conversations are fundamentally full-duplex. Any party may send a message at any time, even while another party is also sending a message. This eliminates the “one transmitter at a time” problem with traditional radio systems.
- All Conversations are fundamentally asynchronous. However, if two parties are actively engaged in a given Conversation and their delay between transmissions is small, then their experience will be one of a synchronous conversation. If either party delays a transmission for whatever reason, the conversation may drift towards an asynchronous experience. This model is similar to the way Instant Messaging (IM) conversations take place.
- Messages may be time-shifted to allow a user to “catch up” on a conversation if they missed some messages or were involved in another conversation. Older messages may be replayed at any time at potentially higher speeds. Users may easily skip forward and backward through their messages and within individual messages. The catch up process may be configured on a message-priority basis to potentially skip messages below a certain threshold. This time-shifting process eliminates users having to ask their entire channel to repeat a message.
- Each Conversation has a variable number of Contacts associated with it. When sending a message to that Conversation, a user could be sending to a single other party or a large group.
- Devices on the network typically communicate with a Server, but they may communicate directly with other Devices under certain circumstances.
- A Device may also be a Server for other Devices, typically in a vehicle-based or fixed site Device with extra power and storage.

Some potential features of Multiple conversation management

- Users may send messages with multiple media types: voice, video, text, location, or raw data instead of only voice.
- Users may attach images or other data files to their messages.
- Different audible tones may be played before listening to or displaying messages depending on whether they are live or recorded, who they are from, priority and other factors.
- The CommoTek network may be interfaced with a legacy radio net allowing full two-way conversations between users of that net and CommoTek users. Users on the legacy side would be limited to the capabilities of their existing equipment. However, CommoTek users would enjoy all of the benefits of the CommoTek system for messages originated on a legacy net.



- The CommoTek network may also be interfaced with the Public Switched Telephone Network. This would allow a seamless transition from existing PSTN-based devices like mobile phones to the CommoTek system.
- The CommoTek network may also be interfaced with VoIP systems.
- The network can have the ability to signal the presence or absence of individuals on the network, as well as the ability to notify if users are no longer actively listening on a given conference call channel.
- Messages may be left for people not on the net to be picked up the next time the individual signs onto the net. This process will likely happen automatically when a message is being sent, but some sort of audible or visible feedback will alert the sender that this message may not be received immediately.
- Individuals on a conference call or tactical net who have shifted their attention away from the device can be signaled or “rung” to notify them that their attention is needed back on the call. This indication may convey multiple levels of urgency.
- The system will be designed from the ground up with a cryptographically sound architecture:
 - Messages are secure through the network from sender’s device to receiver’s device.
 - Non-alterable signatures on messages to ensure authenticity
 - If devices are lost, they may be completely erased or disabled remotely.
 - Network will have to have a hierarchy of trust – both for tactical nets to exclude unauthorized users and also to give network operators the ability to control message spam and other undesirable activities.

UI design – possible design elements include

- Applying IM-style functionality to voice or video messages, but with a seamless transition to real-time conversations
- Ability to sort by:
 - Conversation
 - Sender
 - Receiver
 - Priority
 - Time
 - Group
 - Etc.
- Organization and designation of groups
- The rules and hierarchies of groups
- Transmission control: voice activated, PTT, PTT lock, and other options
- Indication of which individual is transmitting on a party / conference call
- Display of incoming calls and emergency calls and notifications
- Incorporation of text, images and video messages
- The ability to use audible tones to designate status, etc.
 - Different tones for live vs recorded messages



- Different tones for each party on a party line/ conference call line
 - Different rings for different people and varying levels of criticality
 - The potential use of Morse Code tones as alerts
- Voice, video, text, position, or any mixture may be used. The actual communication payload will be referred to as “media”
- ability to type or input a subject or other tag or title before sending an initial message or the ability to add the same after a message has been sent
- pre recorded messages and or responses – e.g. a customer service rep or others may want to save time by pre-recording or otherwise preparing messages that they believe they will need in the future
- with Voice Recognition software, transmitted voice messages can be translated to text for easier organizing, searching and review



Some examples of potential use

Executive Communications

An executive comes out of an important client meeting and wants to communicate five different critical messages to different groups of people (20 people in total) that include his secretary, his CEO, two of his teams, and another key manager working on the project. In today's world, he would have limited options including:

- Place twenty different phone calls to all recipients, one at a time, each call resulting in either a live conversation or a voicemail.
- Take the time to coordinate five conference calls with the required parties, if indeed that is even possible.
- Take the time to type out emails to the different groups.

Using the CommoTek system he could pull out his CommoTek enabled phone or device and make five transmissions. He would not need to wait to see if someone was available. He would not need to coordinate a meeting or conference call. He could select the individual or group he wanted to pass a message to, press a button and transmit the message. Individuals receiving the message could listen to it when they are ready and respond either with their own transmission or with a quick text message. Each of the five groups could engage in sub-conversations around this initial message. The executive could participate in any or all of these based on his time and preferences.

A Conference Call

Peter and Paul work for Company A and are negotiating a deal with Company B. Representatives of both Companies are together on a conference call negotiating terms. Normally, Peter and Paul are in the same physical location and will frequently press the mute button to make comments and ask questions that they don't want the other parties to hear. However, if they are each calling in to a conference call number from different locations, their options are limited to:

- Calling each other on another phone (cell phone, etc) to have a side conversation.
- Using their Internet-connected computers and having an IM conversation while on the conference call.

Using the CommoTek system you aren't stuck in one "call". A "conference call" is just a conversation that you have currently designated as your Priority Group. You can be engaged in a full-duplex, synchronous conversation with the whole group, while at the same time periodically pressing a button to transmit messages to just one of the parties, or to a totally unrelated party. If Peter transmits a message to Paul, Paul won't be interrupted from his conversation. He will see that he has a message from Peter, and he can choose to listen to it (or read it if it is a text message) when there is a lull in the conversation. Paul can speed up Peter's voice to minimize the time he is away from the primary conversation. On returning from Peter's message, Paul may speed up the primary conversation until he is caught up to what is currently being said.



A Fire Department

At a large fire there could be multiple teams attacking different issues. Some teams could be focused on rescue. Others could be focused on fighting the fire. Others could be focused on trying to keep the fire from spreading. Still others could be giving medical aid to victims. There would likely be a leadership element coordinating and overseeing the activities of the different groups including the arrival of med-evac helicopters, ambulances, and others.

If all of the firefighters and paramedics in the different groups are on the same channel then that channel becomes crowded and chaotic. Transmissions get stepped on when more than one person is transmitting at a time, and everyone becomes less efficient. For teams that are involved in dangerous activities, such as those inside a burning building, their priority is to communicate instantly and effectively with their nearby teammates. Having to share their channel with non-critical communications coordinating actions outside the building is not only inefficient but dangerous. A possible solution is to give different teams their own channels. This fixes the problem of overcrowded channels, but it creates a new problem of putting up barriers to communication between the different teams. Should the rescue team be on the same channel as the fire fighting team or the paramedics team? What channel does the Fire Chief listen to? Does he have a different radio for each team? Can he listen to them all simultaneously? What if he is in the middle of one conversation with the helicopter while someone fighting the fire inside the building is trying to have an urgent conversation on a different channel?

With the CommoTek System everyone can be organized into multiple different groups, with each group having its own priority level. For example, all of the firefighters inside the building can have their own priority group. The leader of that team can be giving updates to the Fire Chief without the whole team hearing, or he can broadcast updates to everyone. Any urgent transmissions by his team members inside the building (look out, the floor is going to collapse!) override his non-essential side conversations and are played immediately. The push to talk buttons could be designed to have multiple levels of priority (e.g. push it in half way for a normal priority transmission and all the way for an urgent or emergency one).

If the Fire Chief wants to know what is going on in the building but it isn't an emergency, he can transmit a message to the team leader at normal priority (below the priority of his team members in the building). The team leader will hear or see that he has a transmission from the chief and can listen to it. He can start listening to it whenever he wants, including before the Chief has finished transmitting.

If, on the other hand, the Fire Chief has an urgent message (e.g. the building is going to collapse. Everybody get out!), he can transmit at the highest priority level to everyone at once and it will override every other conversation.



One problem that radios can have is bad reception, especially in buildings and urban settings. This can be helped by having repeaters spread throughout cities and buildings, or even the construction of a mesh network, but it is still possible for there to be dead spots without reception. With the CommoTek system if someone was in a dead spot when a critical message was transmitted they would get that message (and all others that they missed) when they came back into coverage. The messages would be prioritized, so they could receive the most important ones first.

An Infantry Platoon Commander

Obvious. See Fire Department example above. Replace fire suppression with fire teams. Continue to rescue wounded. Add neutralization of hostile forces. Q.E.D.



Description of the Prototype

Goals:

- Prove that fundamental concepts are valid
- Understanding the system's usability and "flows"
- A flexible platform for UI components that support easy prototyping and reconfiguration.
- Understand the human factors involved with participating in multiple simultaneous conversations.
- Test new networking technology.

CommoTek Concepts:

- **Device** – This is the end user component of the system. Users authenticate with the Device, and the Device in turn registers this fact with the Server. For the purposes of the Prototype, a User may only ever be actively logged in to a single Device. Future versions will support simultaneous multi-device access.
- **Server** – The infrastructure of the Network. Server nodes contain authentication information, Presence data, Message Archives, and anything else that might be necessary to facilitate communications between Devices.
- **Node** – any type of device that is part of the CommoTek network, including Devices, Servers, or hybrids. For the Prototype, only pure Devices and Servers will be implemented.
- **Contact** – Users of the system may be contacted on their Devices. Users manage lists of Contacts that either uniquely identify other Users or describe multi-user groups called Channels or Groups. Users are themselves Contacts. Contacts send Messages to other Contacts.
- **Channel or Group** – A type of Contact that may refer to multiple Contacts. When Messages are sent to the Channel or Group, all Channel members will receive them. Contacts may join and leave certain channels based on the security settings of that Channel.
- **Message** – An individual unit of communication from one Contact to another. Each message contains some sort of Media.
- **Media** – audio, video, text, position, or other data that one Contact sends to another inside a Message.
- **Conversation** – A grouping of Messages between Contacts on a Device. Users generally interact with their Device using Conversations, reviewing or adding to a Conversation as desired. When new Messages arrive, they either define a new Conversation, or they are added to an existing one.
- **Store and Stream** – tentative name for the system by which Messages are relayed between Nodes, storing the entire Message while making the Media available for rendering while it arrives. While not strictly "streaming", Store and Stream generally conveys the intent, although a better, more trademark-able term is desirable.



Prototype Features:

- Will use a PC running Windows as a Device. A mic and speaker are required. Devices talk to each other by way of a Server and some sort of IP network. Future versions will support direct Device-Device communication.
- The Server component will run on a datacenter collocated Linux server for the Prototype.
- All communications are delivered reliably to the intended recipient using the CT “Store and Stream” technology. This means that all messages will be stored on both clients and servers, though they will be available for consumption or propagation while the message is being received.
- The only supported Media type for this prototype will be voice. Future versions will support text, video, position, and raw data Media.
- Voice communication will be done only when the user presses a “transmit” button, sometimes called “push to talk” (PTT).
- Incoming Messages may be received while a user is transmitting. These Messages may come from a User on the same Channel, or from a different Channel. The audio Media in the Message may or may not be played immediately upon reception.
- The Device may have multiple Conversations, with new Conversations automatically added as necessary. All sent and received Messages will be automatically sorted into a Conversation.
- All received Messages in all Conversations may be sorted, reviewed, and replayed at any time.
- The prioritization of multiple Conversations is an active area of design. The current plan for the prototype is to build a simple unique priority list. Each Conversation will be assigned a unique priority. Higher priority Conversations will alert their new Messages first. An Alert Threshold may be set, above which new Messages will be played automatically. For example, if the AT is set at 3, Conversations 1, 2, and 3 will have their new Messages played immediately, in priority order. If the user was reviewing Conversation 6, and a P3 Message came in, the P6 message would stop, and the P3 would immediately begin playing. If a P1 message came in while that P3 message was playing, then the P1 would start. After the P1 finished, then the P3 would finish, then the P6 would finish. This sounds complicated, but building the prototype will help us better understand how to best design this.
- While users are engaging in PTT-style transmissions, they may essentially be involved in a “full duplex” conversation. This means that while receiving a Message, one party can send begin sending another Message. The prototype will not have any echo cancellation, and it is likely that audio feedback will result if two parties have their transmit buttons depressed simultaneously. Future versions will require echo cancellation.
- The prototype will not use any kind of encryption, however the final version will make use of a robust and highly secure cryptographic security system.

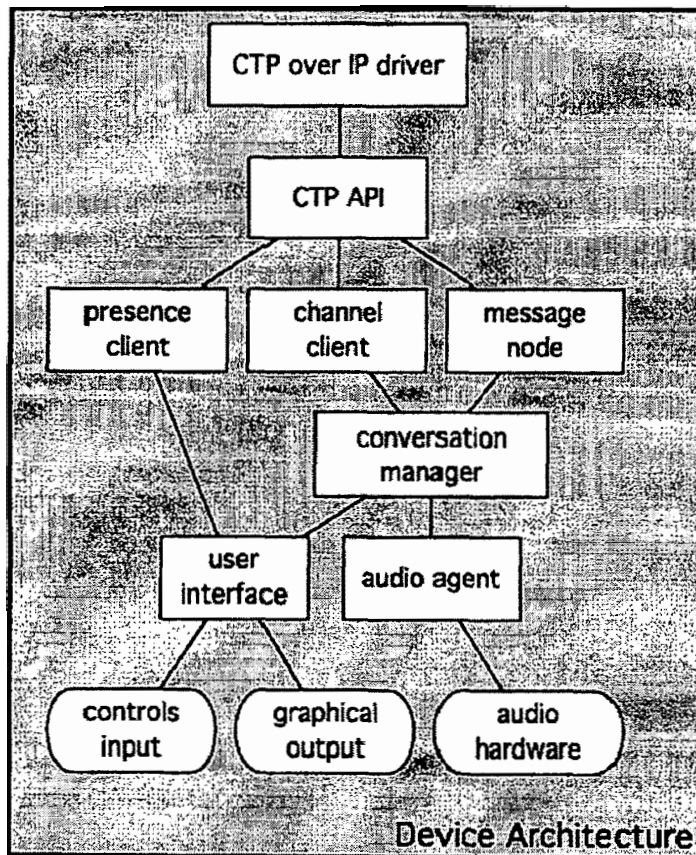


Server software components:

- Channel Server – manages subscriptions from device clients. Devices may create new Channels, see the list of Channels, and subscribe to Channels. Some Channels are created as private by the initiator so that only the Channel members may see and subscribe to it.
- Presence Manager – maintains reachability information for all Device clients.
- Message Router – this tells a sending Device how to reach all of the recipient Contacts for a given Message.
- Message Switch – this is the Store and Streamer. It accepts a Message from a sending Device, and then starts sending that Message to any and all recipients. It tracks the delivery status of the message for all recipients. After all recipients have accepted the Message, the Message Switch may store the message in a Message Archive.
- Message Archive – Stores fully delivered Messages for later retrieval.

Client software components:

- Presence Client – maintains and monitors connectivity via the Presence Manager on the Server. The prototype will publish presence information to all devices on the network, but future versions will have this integrated with the security system so that a “private” mode is possible.
- Channel Client – talks to Channel Server to manage participation in various Channels.
- Message Node – talks to Message Switch on server, accepting and transmitting Messages. Messages are sent to switch while they are being generated locally. Messages may be played locally while they are being received from Switch.
- Conversation Manager – handles priority and sorting of incoming Messages. The User Interface sets the priority of all Conversations. Each incoming Message is part of a Conversation. These Messages may be played immediately, after a currently playing Message finishes, or only upon user request, depending on configuration.
- Audio Agent – talks to PC speaker and mic and the Conversation Manager. When user sends a message, Audio Agent records and encodes the signal from the mic, then relays it to the Conversation Manager. When Conversation Manager determines that a message should be played, it relays it to the Audio Agent.
- User Interface – allows user to join the network, join Channels, define Contacts, initiate Conversations, accept invitations, prioritize Conversations, and likely other operations.



Notes:

- While this prototype will build a Windows client and a Linux server, this operating systems decision does not preclude support for other platforms in the future.
- In future versions, it is likely that all nodes in the network may have components of both client and server.



Images of some potential prototypes:

One of the major goals of the prototype is to experiment with different User Interfaces. Here are some of the early mock ups. These mock ups are based on a touch screen design:

Image 1) Active Call

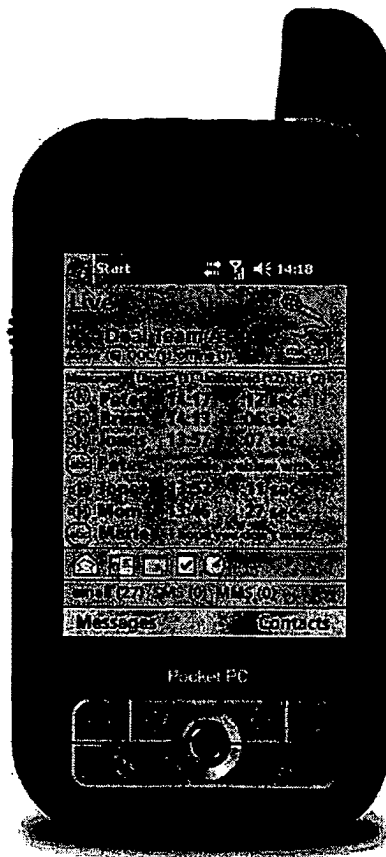




Image 2) Ready but Inactive

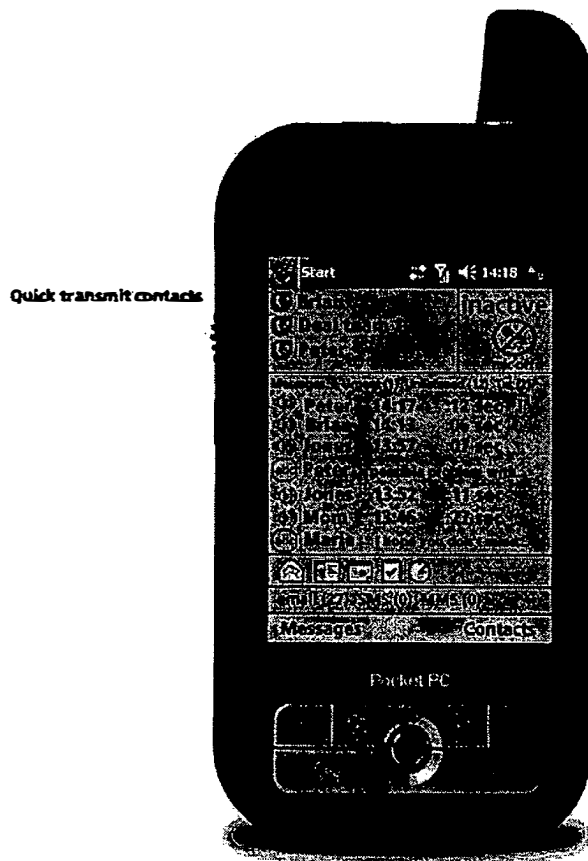




Image 3) listening to a message while on a group call





Image 4) sending a message while on a separate group call

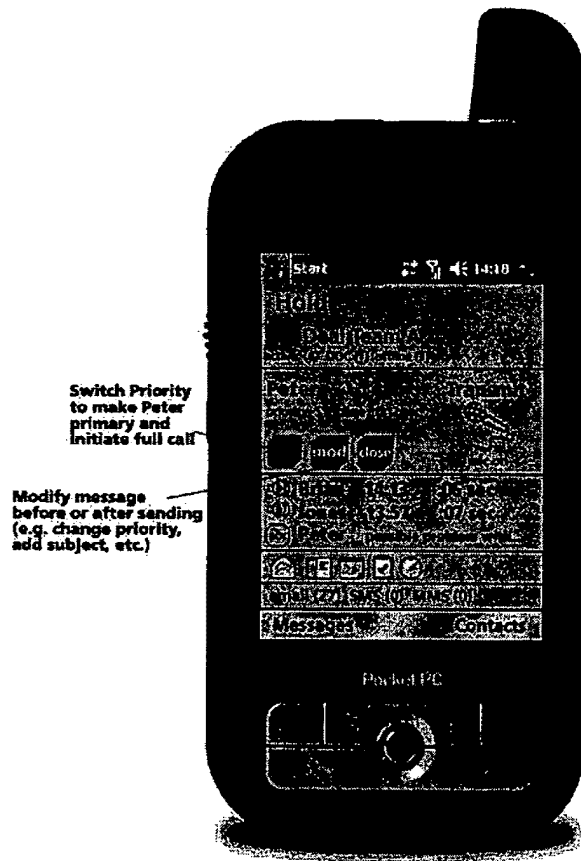




Image 5) Sending a message

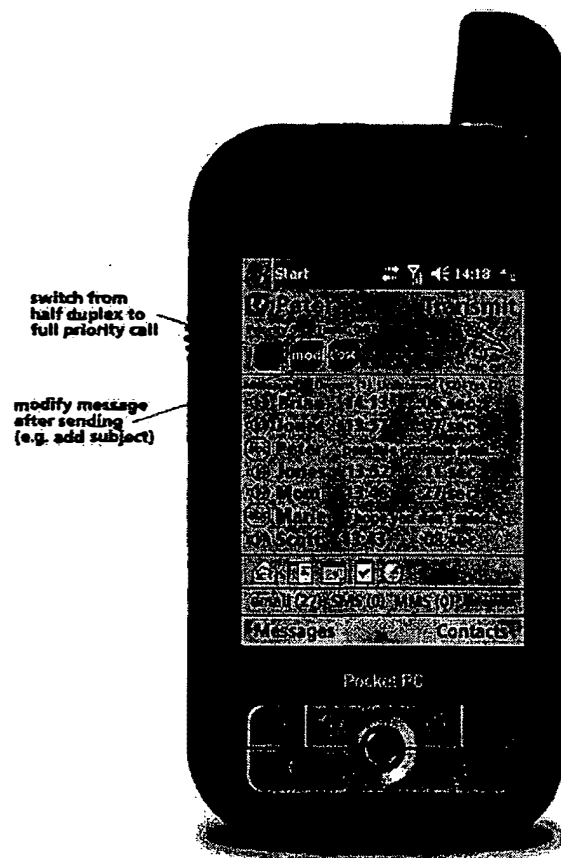
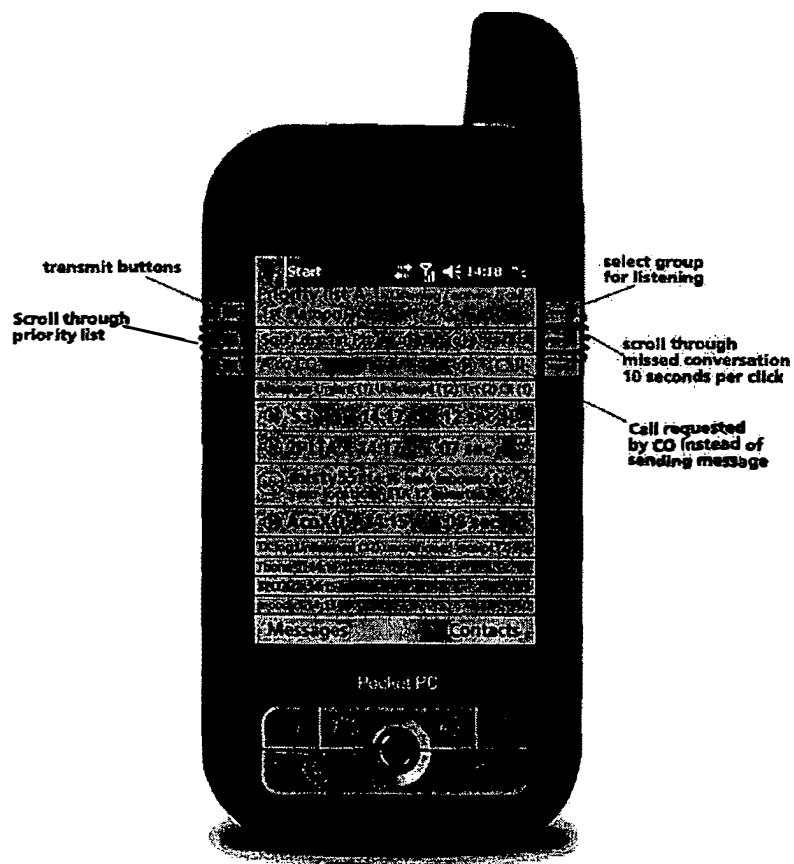




Image 6) potential tactical version





Conclusion

The CommoTek Multiple Conversation Management System is a novel and revolutionary advance in communications for humans.

Written by Tom Katis and Matt Ranney
for
CommoTek LLC
970 West Broadway
Suite E-321
Jackson, WY 83001-9475