

DEAF 911 – DIRECT, EQUAL TELEPHONE EMERGENCY ACCESS FOR THE DEAF

Thad Starner, Nirmal Patel, Zahoor Zafrulla, Ketaki Deo, Ed Price, and Harley Hamilton

ABSTRACT

Teletypewriter (TTY) technology enables the Deaf to access emergency call centers (PSAPs). However, carrying a mobile TTY to connect to a cellular phone just in case of emergency is onerous and rarely done. Yet current SMS (Short Message Service) is not suitable for emergency messaging as it is not supported, not real-time, not localizable by E911, and does not guarantee delivery. We present TTYPhone, a TTY emulation application that allows Deaf users to communicate directly with emergency operators. TTYPhone is a software-only application that is loaded on to a commercially-available cellular phone and is compatible with existing PSAP and cellular phone infrastructure.

KEYWORDS

TTY/TDD, Deaf, Wireless communications, 911, PSAP, emergency

INTRODUCTION

Not far from our research laboratory in Atlanta, GA, a man was shot outside a grocery store late at night. He died because the only person on the scene, a deaf man, could not contact emergency services quickly enough. Not being able to communicate with 911, the deaf man had to SMS a friend, who was hearing, to call 911. Using their E911 automatic location service, the emergency services quickly found the location of the hearing friend, which was, of course, not the location of the shooting. Emergency services were dispatched to this location, as required by law. The time taken to solve the confusion, to redirect the emergency vehicles, and to find the shooting victim was probably the difference between his life and death.

When encountering an emergency, it is crucial to get prompt access to emergency services. In the United States, Public Safety Answering Points (PSAPs), which can be contacted using 911, serve the purpose. While hearing people can speak to the emergency call operators with ease, deaf and hard of hearing people have severe limitations in doing so. The 1990 Americans with Disabilities Act mandated that every PSAP have the appropriate technology to handle incoming calls from TTYs, also known as Telecommunications Devices for the Deaf (TDD). Meanwhile, the Deaf have been quick to adopt additional technologies for communication such as SMS, Instant Messaging (IM), and Video Relay. However, these technologies generally are not supported at the PSAPs. This omission is due to the assumption that all incoming calls to a PSAP originate over the PSTN (Public Switched Telephone Network) or the voice channel of a cellular network. Even if SMS was accepted at 911 centers, it is incompatible with enhanced identity and location information (E911) (Saunders 2003). SMS delivery can be delayed significantly during normal operation and does not guarantee delivery. SMS also requires the user to type the entire

message and press send before the message is sent. All of these factors can lead to situations where an emergency message is never delivered to the PSAP. TTY, on the other hand, is real-time, transmitted character-by-character, is transferred over the voice channel, and should work with E911's location service the same as voice. Unlike SMS, a user who communicates over TTY to 911 knows if they are being "heard" and can interact more directly with the 911 operator. Additionally, by law, wireless carriers must be capable of transmitting 911 calls made using TTY devices. Thus, while more sophisticated methods are being proposed for Deaf access to future 911 services, TTY seems an obvious choice given current technology.

REGULATORY ISSUES

Access to emergency services is a right of every American. Several laws and rulemakings have been enacted throughout the years to ensure that people with disabilities have access to telephone services and emergency services. Sections 255 and 251(a)(2) of the Telecommunications Act of 1934, as amended, "require manufacturers of telecommunications equipment and providers of telecommunications services to ensure that such equipment and services are accessible to persons with disabilities, if readily achievable." This law sets in place the baseline legal requirements that apply to most telephony devices, including traditional landlines, cellular telephones, and interconnected voice over IP (VoIP).

The FCC rules on 911 require that wireless carriers must be capable of transmitting 911 calls from persons with speech or hearing disabilities through the use of other means such as TTY (47 C.F.R. §§ 0.131, 0.331, 1.3). At the time such rules were to take effect in 1997 it was determined that TTY would not properly work over the digital wireless systems being deployed in the United States (<http://www.fcc.gov/Bureaus/Wireless/Orders/1998/da982323.txt>). A series of waivers were put into place while technical solutions to this problem were developed. Eventually, in the Fourth Report and Order in the Matter of Revision of the Commission's Rules to Ensure Compatibility with Enhanced 911 Emergency Calling Systems, the FCC finally set a deadline of June 30, 2002 by which time wireless carriers must be capable of transmitting 911 calls made using TTY devices (6 years after the original rulemaking began - http://www.fcc.gov/Bureaus/Common_Carrier/Orders/2000/fcc00436.txt).

PERCEIVED NEED

From the responses collected for the Wireless RERC's Survey of User Needs (SUN) [WirelessRERC 2007], of the total number of people who indicated a physical or cognitive limitation, 16% reported themselves as deaf (235 out of 1452 respondents). 88% of deaf respondents own or have access to a wireless communications device, and 57% use TTY. Interestingly, even with today's options of SMS and IM, 25% of the deaf respondents indicated that TTY was one of the five most important features in a wireless device. Deaf respondents in the age ranges of 25-44 and 45-64 showed much higher rates of current use of TTY and desire for TTY in a mobile device than those in the 18-24 and 65+ age group. Interestingly, hard of hearing respondents (32%, or 463, of the 1452 with a disability) also showed interest in TTY. 12% of the hard of hearing respondents use TTY, and 10% indicated that TTY was an important

feature in wireless devices. All of the more free-form comments that mentioned TTY in the survey were in response to the question “What would make your wireless device easier to use?”:

- “Email and text become the same no matter if deaf or hearing and make cell phone so I can use it as TTY incoming or outgoing.”
- “Have people easy to call me, tty cellphone, not tty and cellphone separately who has space or time to hook them all up?”
- “Having tty capability so i can direct call someone else who has a tty”
- “If I could use it as a TTY or relay device.”

Clearly, even without mentioning the case of calling 911 while mobile, there is a desire in the Deaf and hard of hearing communities to have TTY as a feature on wireless devices.

Currently, most mobile phone manufacturers provide TTY adapters for their mobile phones. However, according to our contacts in industry, they are rarely sold. Most mobile TTYs are significantly larger than current smart phones (see Figure 1) and still require a mobile phone to operate. Why would a Deaf user carry two devices when one mini-QWERTY style device (e.g. Sidekick, Blackberry, etc.) will provide similar communication services in a cheaper and more mobile package? Even the access to the larger (and presumably faster) keyboard of a TTY seems to be of little benefit. Experienced mini-QWERTY users can sustain a 60 word per minute (wpm) typing rate, which is considered more than sufficient for secretarial work (Clawson et al. 2005). Thus, the Deaf community has little incentive to carry a TTY just in case they need to make an emergency call while mobile. Integrating TTY into current mobile devices seems a more reasonable approach. Indeed, organizations serving the Deaf community, such as Gallaudet’s Technology Access Program, have advocated such an approach for several years (Harkins 2007).

We developed TTYPhone as part of our Deaf 911 effort in the Wireless RERC to demonstrate that a TTY could be emulated on today’s smart phones. TTYPhone requires no change to current 911 call centers and requires only that a software application be installed on the user’s phone. In the following sections, we discuss TTYPhone’s development, testing to date, and our future work in development and user studies.

SYSTEM DEVELOPMENT

Hardware Platforms

TTYPhone was initially developed in Symbian C++ and installed on a Panasonic X700 phone running Symbian OS 2nd Edition [Zahoor]. At the time, this combination of software and hardware was one of the few that allowed programmers to access the voice stream. Mobile phone service providers and manufacturers may be concerned that such access might lead to applications that violate the user’s privacy and thus limit third party capabilities. Unfortunately, even the Panasonic phone only allowed processing the voice stream in batch. Thus, our original

system would send a default message to the PSAP, listen for returning TTY signals from the PSAP for 30 seconds, decode and display any characters sent in that 30 seconds, wait 30 seconds for the user to type a message, encode and send those characters, and repeat the process. Needless to say, such a system is not sufficient.

Fortunately, the OpenMoko Neo Freerunner mobile phone became commercially available. This open software and open hardware Linux-based GSM/GPRS phone platform allows us to directly control the voice stream (and almost everything else) through a standard programming interface. Thus, we could write a truly interactive application using standard programming languages (in this case, we chose C++ and Python for speed and convenience).

Operation

Figure 2 shows a conceptual overview of the TTYPhone prototype. Figure 3 shows three screens on the OpenMoko mobile phone that show the prototype in action. When the user first contacts 911, the application sends an introductory message (Figure 4 shows our acoustic TTY receiving the introductory message). For now, the message is “HELLO I AM DEAF” but will eventually include more information to help guide a 911 operator’s response. After playing the message, TTYPhone listens to capture any characters typed by the 911 operator. When the user wants to type a character, he uses the on-screen mini-QWERTY virtual keyboard. Currently, the system must stop listening to the incoming signal to send a character typed by the user. While the Openmoko hardware and software allows us to remove this limitation in theory, normal TTY has this same (half-duplex) restriction. Thus, TTYPhone provides functionality similar to current desktop TTYS. Interaction takes place in real time, and the message is transmitted and received on a per character basis. Communication continues until the deaf user hangs up the call. Standard TTY etiquette can be adhered to during the conversation, such as using “GA” to indicate Go Ahead and “SK” for stop keying. Currently, TTYPhone provides a soft key shortcut for GA for the convenience of the user. Other soft key shortcuts are easily added.

Encoding and Decoding Baudot

TTY uses a standard called Baudot for transmitting letters over the voice channel. This standard can trace its history to a five positive and five negative element alphabet used for telegraphy by Weber and Gauss in Goettingen in 1833 (Vansize 1901). Baudot consists of MARKS and SPACES transmitted at 45.5 baud, sending each character in 176ms. A MARK corresponds to a “1” and is indicated by a tone at 1400Hz. A SPACE is represented by a “0” and a 1800Hz tone. Each character has a 5-bit code with one leading SPACE bit and 1.5-2 trailing MARK bits. Figure 5 shows a spectrogram of a TTY message (“HI”) captured using the Openmoko phone. In testing, we found the number of trailing MARK bits to be variable, with several manufacturers tending to include many more than necessary. Figure 5 shows that the TTY signal is surprisingly clear given that it has been transmitted over a cellular network (AT&T in Atlanta, GA).

To send Baudot, we store an audio file in the TTYPhone application for each possible TTY character. The initial message is constructed by concatenating the audio files for the

appropriate TTY characters. This combined audio file is saved and played when the application first connects to 911. For each subsequent character typed, the corresponding audio file is played to the voice stream. TTYPhone removes the sidetone (a low level introduction of what is heard in the microphone into the earpiece) and disconnects the Openmoko's microphone to limit interference in transmission. These operations can be done quickly and under program control.

Receiving TTY characters is more complicated. Fortunately, the Openmoko allows the elimination of sidetone and provides a relatively clean incoming voice stream audio signal. In theory, we could perform a fast Fourier transform (FFT) on the incoming audio to determine which frequencies are present. Unfortunately, an FFT is computationally expensive. However, the Goertzel algorithm is an optimized version of the FFT which calculates the signal power at only a single frequency (Jacobsen & Lyons 2003). Since TTYs only use 1800Hz and 1400Hz for signaling, we can save significant computing power by just calculating the signal power at these two frequencies. Most mobile phones do not have floating point units (FPUs), and floating-point heavy tasks such as an FFT or the Goertzel algorithm can be quite slow. By using integer approximations we were able to achieve run times that were several orders of magnitude faster than the straightforward implementation. Currently, TTY decoding on the Openmoko runs two times faster than real-time.

Voice and Hearing Carryover

Given the excellent control the Openmoko provides for the voice streams, we will investigate reconnecting the microphone when TTYPhone is not transmitting a character so that the 911 operator can monitor the caller's ambient environment and get a better awareness of the emergency situation. This feature will also allow VCO (Voice Carryover), where the caller could speak as well as type to the PSAP. Similarly, we will investigate playing the incoming audio to the Openmoko's external speaker (as opposed to keeping the interface silent). This feature would allow HCO (Hearing Carryover), where a caller who is unable to speak could type messages to the PSAP and receive responses in speech. Unfortunately, it is unclear if current PSAP hardware would support this feature, and, furthermore, such a loud device may be unsuitable for some emergencies (for example, a call during an on-going home break-in).

One benefit for providing both VCO and HCO is that TTYPhone could be made to work transparently whether or not the user was deaf. When a 911 call is placed, a mini-QWERTY keyboard is displayed on the Openmoko's touch screen. When the user speaks, TTYPhone simply forwards the voice streams as normal. If the user starts typing with the keyboard, TTYPhone transmits Baudot codes. In theory, this approach could be used on many of today's smart phones. Such a system could benefit the hearing when they have an emergency where they need to be silent. For example, two kidnap victims in Atlanta were stored in the trunk of their car as their kidnappers drove them to different cash machines around the city. The hearing victims had to SMS friends to contact police to come to their aid (Haines 2007). With TTYPhone, the victims could have maintained silence while contacting the police directly and allowing the police to trace their movement using E911.

INITIAL TESTING AND DEMONSTRATIONS

To date, we have performed informal testing against two acoustically-coupled and two direct connect TTYs. The acoustically-coupled TTYs were stand alone systems whereas the direct connect TTYs were part of the standard hardware at PSAP call centers. We tested against Zetron PSAP hardware at Georgia Tech's emergency call center (Figure 6) and Positron PSAP hardware at the city of Orlando's emergency call center.

Acoustic-coupled TTYs

To our surprise, the decoding of characters sent from the acoustic TTYs had significantly fewer errors than the transmission of characters from the Openmoko to the acoustic TTYs. After several experiments, we realized this effect was partially due to ambient noise captured by the acoustic coupler. Simply covering the acoustic TTYs with a jacket improved the results. After some tuning of the system, we could both transmit and receive with some reliability. While no wired or wireless TTY system will be 100% reliable due to the vagaries of line conditions, achieving 1% error or less seems a reasonable goal and is in-line with previous work (Dörbecker 2001). Formal testing will begin this Fall. We are investigating tools and scripts designed at Gallaudet to determine if they can be re-implemented on our phone platform for testing (Norman 2001).

Georgia Tech 911 Center

Georgia Tech's Zetron PSAP automatically detects incoming TTY signals and displays the text on a monitor (normally used for displaying the location information associated with the call). When testing against this hardware, we discovered that the spectrograms from both the transmitted and received signals were significantly cleaner than those with stand-alone TTYs. Transmission of the initial message worked relatively well, but some errors still remain. Reception worked fairly cleanly. Surprisingly, character by character transmission from the OpenMoko to the PSAP showed that, after pausing, the first character in a group was often not received by the PSAP. This artifact may be due to how we retrieve the TTY audio files. The system processing required to pull the first character's information from flash memory may slow the audio playback sufficiently that the first few bits are garbled. A simple solution should be to pull the audio files into main RAM when the TTYPhone application is started. We are capturing spectrograms made from the Openmoko's transmissions to further investigate the issue.

Another improvement was discovered from capturing and comparing the spectrograms produced by the Ultratec TTY and the Zetron. The Ultratec adds several more 1s to the end of each character than the specification suggests. Our characters, on the other hand, mostly use silence between the characters. Upon reflection, keeping a signal inserted into the channel may help stabilize any auto gain control (AGC) in the system from amplifying background noise to the point where it falsely triggers bits when no signal is being transmitted. Furthermore, in decoding characters, these extra 1s may also provide clean signal strength references for systems that adapt to the quality of the signal. Modifying our transmission code to add extra 1s seemed to improve signal quality. However, when we added too many 1s, the Zetron TTY

recognized them as the “Letter Shift” or “Figure Shift” characters, causing the Zetron to switch into a different display mode (which the 911 operator did not know how to reset without restarting the call). Thus, we need to tune the system to provide the best clarity while avoiding this difficulty.

NENA2009 Demonstration

During a presentation to the National Emergency Number Association in Orlando, Florida (NENA2009), we placed a live call to the Georgia Tech 911 center to demonstrate transmission and reception of TTY from the Openmoko phone. We also connected our Ultratec TTY to the hotel’s phone lines so that attendees could test the system themselves. While still informal, this test showed that we have not overly tuned our system to AT&T’s network in Atlanta (versus Orlando).

While in Orlando we had the opportunity to test TTYPhone against the city’s Positron-based PSAP. Again, the spectrograms of the signal seemed cleaner than those captured from the acoustic TTYs. However, we quickly discovered a bug in our decoding software that seemed to be triggered by the good signal-to-noise ratio in the signal received! While this bug prevented a two-way demonstration, we captured representative samples of the TTY signals so as to better tune the decoding system in the future.

Interestingly, Orlando’s PSAP TTY was configured to automatically send a default “orlando police what is your emergency q ga” message upon recognizing a TTY call. Since normal TTY is half-duplex, having TTYPhone and the PSAP both send introductory messages could cause both messages to be garbled or interleaved. Outgoing characters from the PSAP are lowercase while incoming characters are uppercase, which helps with the readability of interleaved messages. However, depending on whether sidetone is used on the PSAP hardware, garbling could still be a problem. However, this issue is the same as on current landline TTYs, and users can employ normal TTY etiquette to resolve the problem. Given the separate channels for transmission and reception on mobile phones, TTYPhone could support full duplex operation. However, it is unclear if current PSAP hardware can support full duplex TTY, either landline or wireless.

FUTURE WORK

Our testing against Positron and Zetron PSAP hardware shows that 911 call centers can have a wide variety of procedures and defaults in addition to having small variants on the Baudot signals itself. In addition, cellular networks can have different means of transporting Baudot to better survive radio transmission.

Figure 7 demonstrates how Baudot may be transcoded into the Cellular Telephone Text Modem (CTM) standard before transmission over a wireless GSM link [Dorche]. CTM is transcoded back to Baudot before the signal reaches the wired telephone networks. We suspect that the chipset in the Openmoko is automatically recognizing the Baudot tones we are injecting in the voice stream and is encoding the Baudot characters in CTM. If so, this transcoding would explain the better accuracies we have been seeing compared to older tests at Gallaudet (Norman 2001) and may indicate that CTM transcoding is now part of standard cellular phone

GSM chipsets. Even without such standard transcoding, TTYPhone could be modified to directly generate and/or receive CTM codes, thus (seemingly) ensuring compatibility with current GSM networks (note that more advanced networks support older versions for compatibility, and many mobile phones allow the user to select which wireless standard they wish to use). However, CDMA networks use a different standard for transmitting Baudot. We are currently exploring how to test TTYPhone against traditionally CDMA-based networks.

In order to ensure accurate transmission, we need to test TTYPhone against various combinations of wireless networks and PSAP hardware. Even on a given carrier, regional networks are heterogeneous. Fortunately, after our NENA2009 presentation, most of the major PSAP hardware vendors and representatives from all of the major networks offered their assistance. Several managers of state-wide 911 centers volunteered their PSAPs for testing as well. We are currently planning on how to best leverage these offers.

In the near future, we must optimize TTYPhone's decoding and transmission systems both for accuracy and speed. We plan to use machine learning and pattern recognition techniques to automatically adjust to line conditions. Of particular interest, we expect to recover signal power at 1600Hz (midway between a "0" and "1" tone) so as to better determine when a TTY signal is present compared to noise that was amplified by AGC. We are also examining different TTYs on the market to determine what encoding optimizations and variations are common.

Beyond the purely technical problems of implementing TTYPhone on a mobile device, we have encountered several important usability issues. For example, when should TTYPhone be started? Should it be whenever the phone calls 911? What if a hearing person uses the phone to call 911? Perhaps the phone should only start TTYPhone when the user tries to SMS 911. Or perhaps a different number, say 912, should be used to trigger the application. The most simple solution, suggested in a previous section, may be to have TTYPhone always run for 911 calls, keeping VCO and HCO on so that the phone behaves normally for a hearing user and still provides a keyboard for typing.

In our discussions at NENA2009, many PSAP managers stated that they received TTY calls very rarely. Several PSAPs, including our local one, were interested in working with the project just to have TTY training for their operators. This opportunity may allow us to perform usability tests on TTYPhone from the 911 operator's perspective. Perhaps an introductory message from TTYPhone can help guide 911 operators who are unaccustomed to TTY calls to prompt for appropriate information. Conversely, perhaps we can include several iconic buttons on the TTYPhone interface that transmit pre-formatted answers to common requests from 911 operators (e.g., "Do you need an ambulance, police car, or firetruck?").

We also wish to perform user studies and iterative development with the Deaf community to ensure that the interface is intuitive and appropriate for their needs. One possible advanced feature is to match queries from the 911 operator, who has a typical script to follow, with appropriate signed phrases. These phrases could be displayed with the text message as it is received. While the phrase matching could cause confusion, it could help to address English-as-

a-second-language issues for the Deaf community. User studies can help determine if such an aid is useful or a distraction.

CONCLUSION

We have described TTYPhone, a software-only application that runs on a commercially available cellular phone to help the deaf community access emergency services without requiring changes to current PSAP infrastructure. We have demonstrated TTYPhone in a number of forums and have gained significant experience in implementing a system that can work for a wide range of users. We plan on continuing development of the application and working with industry in order to improve access to emergency services for both the hearing and the Deaf communities.

ACKNOWLEDGEMENTS

We would like to express much gratitude to Richard Ray, Steve O’Conor, and Paul McLaren for arranging testing at the Orlando 911 center and Walter Warner who provided access to Georgia Tech’s PSAP. Thanks also to John Morris who provided the statistics from the Wireless RERC’s Survey of User Needs. This work is supported, in part, by the Wireless RERC and the National Science Foundation (NSF) under Grant No. 0812281. The Rehabilitation Engineering Research Center for Wireless Technologies is sponsored by the National Institute on Disability and Rehabilitation Research (NIDRR) of the U.S. Department of Education under grant number H133E060061. The opinions contained in this paper are those of the author and do not necessarily reflect those of the U.S. Department of Education or NIDRR. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

REFERENCES

- Cagle, S. J. (1991). GA and SK etiquette—Guidelines for telecommunications in the Deaf community. Bowling Green, OH: BG Press.
- Clarkson, E., Clawson, J., Lyons, K., and Starner, T. (2005). An empirical study of typing rates on mini-QWERTY keyboards. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*, 1288-1291.
- Dörbecker, M., Hellwig, K., Jansson, F., & T. Frankkila. (2001). The Cellular Text Telephone Modem – The solution for supporting text telephone functionality in GSM networks. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 1441-1444.

Harkins, J. (2006). "Pager" Access to 9-1-1: Weighing Short-term and Long-term Solutions [slides]. Retrieved from Gallaudet University Technology Access Program. Web site: http://tap.gallaudet.edu/Presentations/Harkins%20NAD%202006_files/index.html

Haines, E. (2007). Kidnapped Atlanta Lawyers Freed After Text-Messaging for Help. The Associated Press. July 5. Web site: <http://www.law.com/jsp/article.jsp?id=1183453584114>

Jacobsen, E. & Lyons, R. (2003). The Sliding DFT, IEEE Signal Processing Magazine, March, 74-80.

Sauders, C. (2003). Bridging SMS and E911. Retrieved June 1, 2009, from Instant Messaging Planet Web site: <http://www.instantmessagingplanet.com/wireless/article.php/3069641>

Vansize, W. B. (1901). A New Page-Printing Telegraph. Transactions (American Institute of Electrical Engineers) 18: 22.

Williams, N., Tucker, P. E., & Harkins, J. (2001). Wireless TTY Tools [user manual]. Retrieved June 1, 2009 from Gallaudet University Technology Access Program. Web site: <http://tap.gallaudet.edu/ttytools>

Wireless RERC. (2007). Survey of User Needs.

Zafrulla, Z., Etherton, J., & Starner, T. (2008). TTY phone: direct, equal emergency access for the deaf. ACM SIGACCESS conference on Computers and accessibility (ASSETS), 277-278.

Address of project lead:

Thad Starner

85 5th St. Room #239

Atlanta, GA 30332

USA



Figure 1: An Ultratec Minicom iV acoustically coupled TTY and a standard office phone used for initial testing.

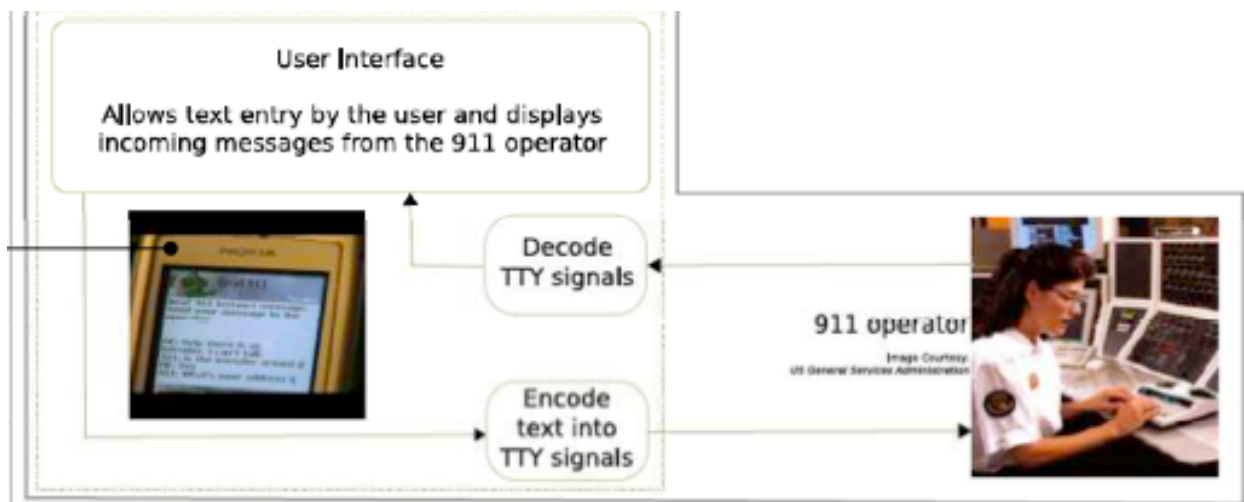


Figure 2: Overview of the TTYPhone application. The user dials 911 and an introductory message is played. The 911 operator, recognizing a TTY call, switches the PSAP hardware to TTY mode. The user types a character, and the corresponding Baudot tones are played over the voice channel. The 911 operator's TTY receives the tones and displays the character. When the 911 operator types, the appropriate tones are played, and TTYPhone detects the tones and displays the appropriate character on the screen.

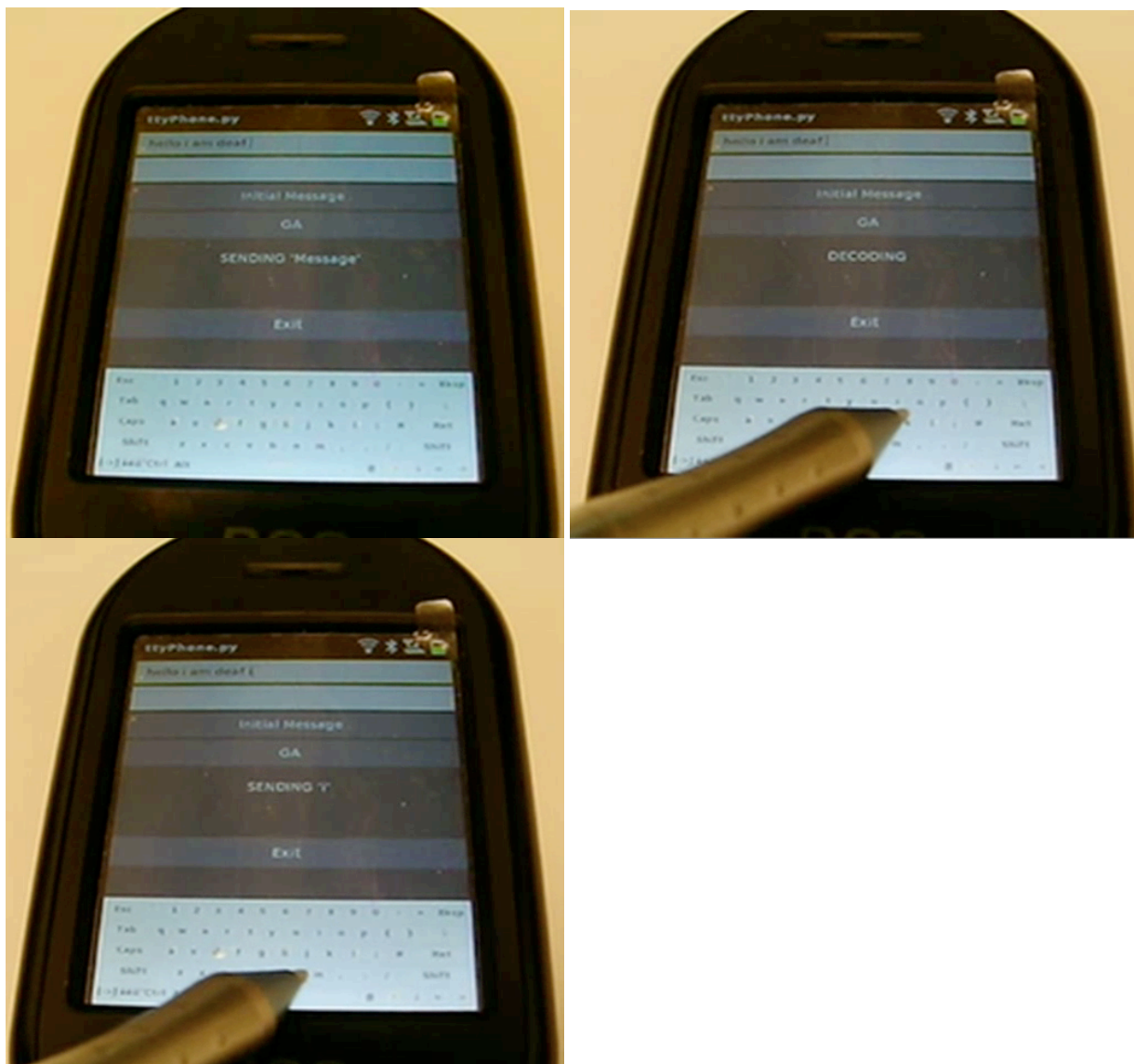


Figure 3: When the user dials 911 and starts the application, TTYPhone sends an initial message to the PSAP (left). TTYPhone then switches into decoding mode to listen for any TTY characters that are sent from the PSAP to the user (right). Whenever the user types a character, decoding is momentarily stopped, and the typed character is sent to the PSAP (bottom).



Figure 4: TTYPhone's initial message is displayed on the TTY called using the OpenMoko cellular phone.

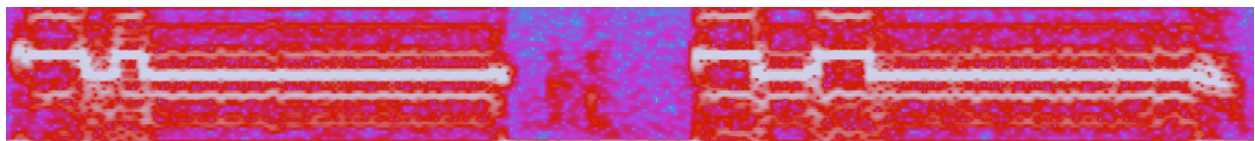


Figure 5: A spectrogram showing the message "HI" as transmitted from the Ultratec TTY to the OpenMoko phone using AT&T's GSM network in Atlanta, Georgia. Each character has a 5-bit code with one leading SPACE (0) bit and 1.5-2 trailing MARK (1) bits. The Ultratec puts extra MARK bits at the end for clarity. A 0 is indicated by a tone at 1800Hz, and a 1 is marked by a 1400Hz tone. "H" is transmitted as 00101, and "I" is transmitted as 01100.



Figure 6: Early TTYPhone tests at Georgia Tech's PSAP.

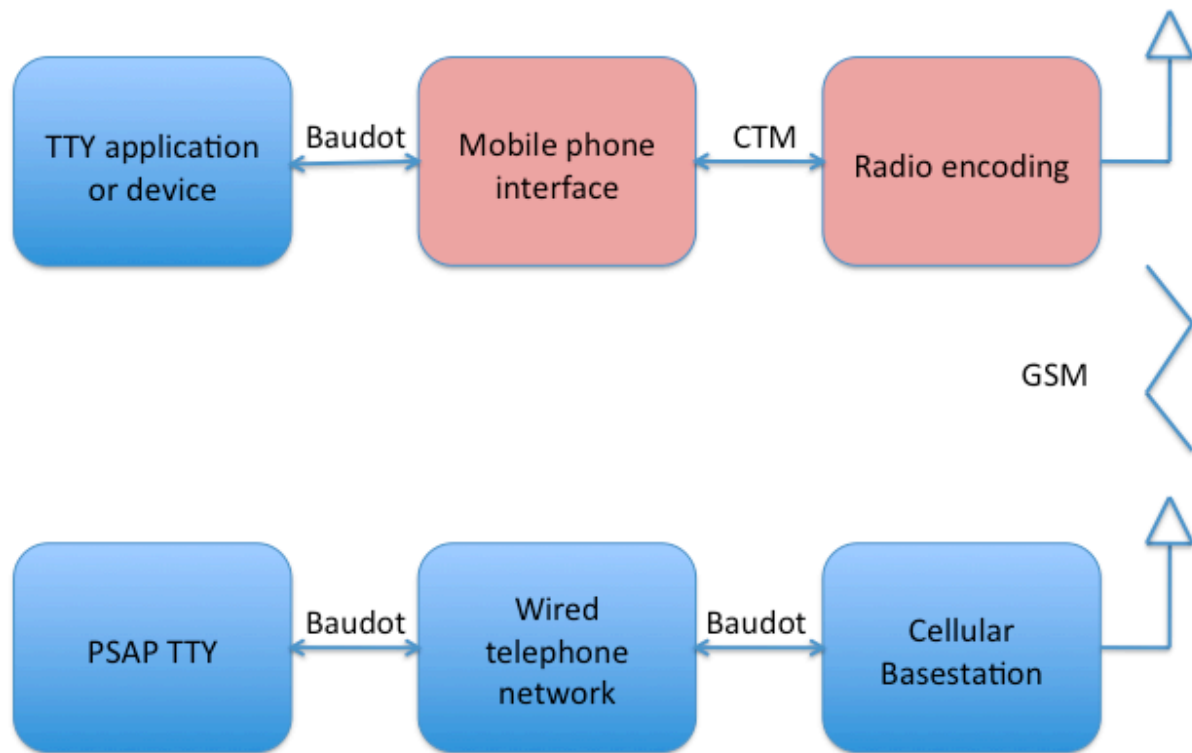


Figure 7: An example of how Baudot may be translated to an intermediate format for transmitting over the cellular infrastructure. In this case, Baudot is transformed to the Cellular Telephone Text Modem (CTM) standard in the mobile phone itself before the audio encoded into GSM radio signals for transmission (image based on Dörbecker et al. 2001).