DT-Talkie: Push-to-Talk in Challenged Networks

Md. Tarikul Islam
Department of Communications and Networking
Helsinki University of Technology
mtislam@netlab.tkk.fi, mtislam@cc.hut.fi

1. INTRODUCTION

Push-to-talk (PTT) technology is gaining popularity with increasing number of mobile devices around the world. PTT is similar to walkie-talkie experience that provides a way to communicate one-to-one and one-to-many. PTT is commonly employed in the wireless cellular phone services that use a single button to switch between voice transmission mode and voice reception mode. Half-duplex mode of communication is the base of PTT where only one party can speak at a time and others can listen. There is a version of PTT called “Push-to-talk over Cellular” (PoC) that is based on 2.5G or 3G packet-switched networks. PoC uses SIP and RTP protocols to communicate over Internet architecture in the walkie-talkie fashion. Open Mobile Alliance (OMA) has specified an open standard of PoC [4].

PoC service in the operator independent wireless networks (e.g. WLAN) has received significant attention. In [1], Wu et al. implement a PoC client in the WLAN environment with some variation from the OMA PoC specifications.

The traditional PoC services over the aforementioned communication networks, either cellular networks or operator independent wireless networks, are infrastructure based. But the mobile users talk with each other in the similar fashion of walkie-talkie in an environment where infrastructure may not be available. The PoC services also rely on the traditional Internet protocols that require end-to-end path for communication. But the current Internet protocols may not work in the scenario of intermittent and opportunistic connectivity. Moreover the PoC services may not work in the environment that suffers from high delay, high error rates and low data rates. So an alternative solution is needed to communicate in the challenged scenarios.

Delay Tolerant Networking (DTN) is an approach that overcomes the problems associated with intermittent connectivity, long or variable delay, low bit rates, high error rates and frequent network partitions. DTN (modeled after email) allows communication in the above extreme conditions through using store-carry-forward message switching mechanism. The IRTF DTNRG has proposed bundle protocol to be used in challenged networks. The bundle protocol which overlays on top of heterogeneous region specific lower layers to support communication across challenged environments. The bundle protocol defines a series of contiguous data blocks as a bundle. Each bundle is variable in size that carries application data. In DTN architecture, a bundle node is an entity that can send and receive bundles. Each bundle node has three conceptual components: a bundle protocol agent, zero or more convergence layer adapters and an application agent [3]. The DTNRG has also provided a reference implementation that implements most aspects of the DTN bundle protocol [2].

In this paper we describe a prototype application called DT-Talkie which enables mobile users roaming around challenged network to communicate with each other through sending and receiving voice messages in the PTT fashion. If user A wants to talk with user B, user A presses the PTT button and starts talking. When he finishes, he presses the button again. Then DT-Talkie forwards his voice message over the DTN infrastructure as a bundle destined to user B. On the other end, user B can hear the voice that is sent from user A as soon as the bundle is received. User B communicates with user A using the same approach. DT-Talkie is implemented for Maemo based Nokia Internet Tablet. In the present implementation, DT-Talkie supports one-to-one communication. Section 2 describes our delay tolerant PTT application architecture. The details of the implementation are presented in Section 3, followed by the demonstration setup in Section 4.

2. SYSTEM ARCHITECTURE

In the OMA specified PoC architecture the endpoints use traditional Internet multimedia protocols to send and receive voice messages over wireless networks. Typically the PoC session is established using SIP and the voice messages are transmitted back and forth between the endpoints using RTP. Such protocols require stable end-to-end connection to communicate between the endpoints. But the mobile nodes roaming around extreme environments might never establish an end-to-end path and therefore will not be able to send and receive voice messages.

To get over the requirement of stable end-to-end path, we employ message-oriented DTN approach in the DT-Talkie application that enables mobile endpoints to exchange voice messages asynchronously in the challenged environments. The voice messages sent from the sender are pushed into the DTN infrastructure and forwarded towards the recipient using store-carry-forward mechanism.

All nodes in the DTN are identified by a unique endpoint identifier (EID), which conforms to the Uniform Resource Identifier (URI). Each EID can be characterized as having the general structure: <scheme name>:<scheme-specific part, or "SSP">. "dtn:" is the one default scheme specified in the bundle protocol that takes an arbitrary string as SSP. When an application wishes to receive bundles, it registers to a particular EID with its local bundle agent.

The general architecture of delay-tolerant PTT application DT-Talkie is shown in Figure 1. Two entities run on the box of the end users. One is the DT-Talkie application that comprises of three modules – GUI, Audio C/P (Capture/Playback) and Bundle S/R (Send/Receive). The other one is the dtnd that runs as a background process to provide the bundle protocol services to the application by means of dtnd APIs.

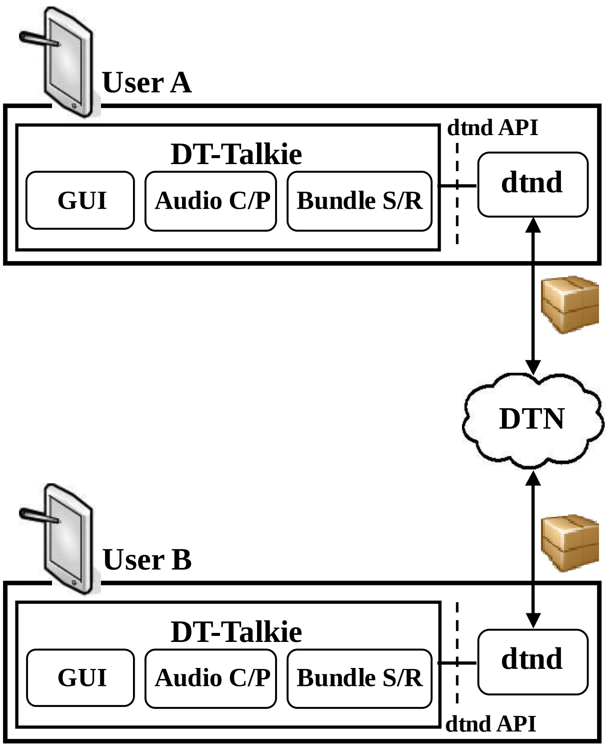

Figure 1: DT-Talkie Architecture

We describe the architecture in Figure 1 with the aid of a scenario. Suppose, user A wants to talk with user B in the PTT fashion. User A selects the EID of user B, presses the PTT button (in this case, we use full screen hard button of the Internet Tablet) and starts talking. DT-Talkie Audio C/P module captures voice from the audio source (e.g. microphone) of user A in the PCM format, encodes the PCM voice and saves encoded bytes in the local temporary file. When user A presses the button for second time, voice capturing is stopped. Then DT-Talkie Bundle S/R module encapsulates encoded audio bytes in the DTN bundle payload that was saved in the temporary file. Finally Bundle S/R module sends the bundle over the DTN infrastructure destined to user B.

In the side of user B, DT-Talkie Bundle S/R module receives the bundle that comes over DTN and decapsulates the bundle. Then the module gets the encoded audio bytes from the bundle payload and saves them in a local temporary file. Finally DT-Talkie Audio C/P module decodes the encoded audio file and starts playback in the audio sink (e.g. speaker) of user B. The aforementioned approach is applicable for user B if he wants to answer user A. DT-Talkie GUI module creates the main window and draws several widgets over that window. It also handles the event when the user interacts with the widgets and presses the hard keys.

3. IMPLEMENTATION

We implement delay-tolerant PTT application DT-Talkie for Nokia Internet Tablet that is based on Maemo platform. The Maemo platform is itself based on the GNU/Linux operating system and the GNOME desktop. There is an open source, GTK based UI toolkit called Hildon that provides a finger friendly interface. Hildon is standard on the Maemo platform used by the Nokia Internet Tablets.

We use DTN2 as the bundle protocol agent in our implementation of DT-Talkie. DTN2 is the reference implementation of the bundle protocol developed by DTNRG. DTN2 supports basic DTN functionality, including the application API, return receipts, static routing and flood routing. DTN2 includes support for TCP, UDP, Bluetooth, Ethernet and Sneakernet convergence layers. DTN2 also supports several link types such as always-available links, on-demand links, opportunistic links and links with scheduled contacts. In DTN2, SSP of the "dtn" scheme takes the canonical form: "dtn://<router identifier>/<application tag>", Where router identifier is a DNS-style hostname string and application tag is any string of URI-valid characters. In our implementation, "host.dtn" is used as router identifier and "dttalkie" is used as application tag (e.g. dtn://Nokia-N810.dtn/dttalkie) to identify the endpoints.

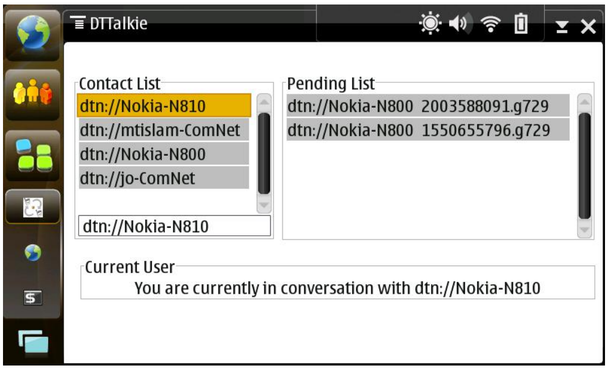

Figure 2: DT-Talkie Screenshot

In the GUI module of DT-Talkie we use GTK+ and Hildon framework to draw the main window and other widgets. There is an example screenshot depicted in Figure 2. In the main window there is a frame "Contact List" through which the user can select EID of other user to talk with, a frame "Pending List" that shows the voice messages which have not been played back yet and a frame to see the other person who is currently in conversation in the one-to-one communication scenario. There is a text entry box beneath the contact list to input EID of a user that is not available in the list.

We use GStreamer framework in the DT-Talkie Audio C/P module to capture and playback voice messages. G.729 codec is used to encode the captured PCM raw voice from audio source and decode to playback in the audio sink. DTN2 APIs are used in the Bundle S/R module of DT-Talkie for sending and receiving bundles over DTN architecture.

4. DEMONSTRATION SETUP

In the demonstration setup, we use two Wi-Fi and Bluetooth enabled Nokia Internet Tablets (N800 and N810). The latest release of DTN2 (version 2.5.0, released in October 2007) is ported to those devices. We use Bluetooth and TCP convergence layer in the demonstration scenario depicted in Figure 3 to transfer voice messages over DTN infrastructure using WLAN interface. For DTN message forwarding, flooding mechanism is used where a node transmits incoming voice messages to all the links except the one from which the voice message is arrived. Thus the voice messages are flooded throughout the network and finally reach the destination. Since the voice messages are flooded throughout the network using the flooding mechanism, DT-Talkie can be easily adapted for group communication. For persistent storage service, we use local file system that is used to store bundles, network state information (e.g. routing tables) and application state information (e.g. registrations). Voice messages are deleted immediately after playing back so that the resource constrained Internet tablets do not run out of storage.

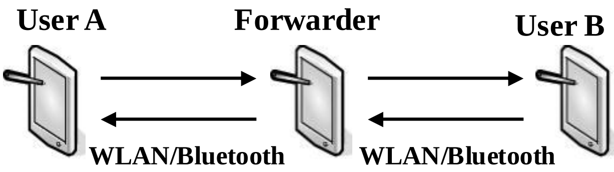

Figure 3: Demonstration Scenario

In Figure 3, we incorporate an intermediate mobile node that serves as forwarder for both endpoints. DT-Talkie and dtnd are running in both the devices of User A and User B. Only dtnd is running in the forwarder's device. We create artificial disruptions to get the DTN experience through disconnecting the link between the forwarder and User B. When User A sends a voice message destined to User B, the message is queued in the forwarder's device. After a while, we enable the connection between the forwarder and User B. The forwarder then forwards the message to User B as soon as the link becomes available between itself and User B.

5. REFERENCES

[1] Wu, L.-Y., Tsai, M.-H., Lin, Y.-B., Yang, R.-S. A Client-Side Design and Implementation for Push to Talk over Cellular Service. Wireless Communications and Mobile Computing, 55(1): 380-383, 2006.

[2] DTN2 Reference Implementation, http://dtnrg.org/wiki/Code

[3] K. Scott, S. Burleigh, "Bundle Protocol Specification", RFC 5050, 2007

[4] OMA Enabler Release Definition for Push-to-Talk over Cellular, Candidate Version 1.0 – 29 April 2005, OMA-RELD-PoC-V1 0-20050429-C, 21p.

APPENDIX

Equipments needed for the demo installation:

- 3 Nokia Internet Tablets
- 1 WLAN access points
- Optionally Internet connection with at least one public IP address to talk to remote user.
- 4 power outlets
- Poster wall + 2 m2 table surface for all devices.

Setup Time: About half an hour