DT-Talkie: Interactive Voice Messaging for Heterogeneous Groups in Delay-Tolerant Networks

Md. Tarikul Islam
Department of Communications and Networking, Helsinki University of Technology

***Abstract*—Walkie-talkie kind of communication has been gaining attention as a service in cellular networks. Traditional walkie-talkie like services require network infrastructure for its operation. While infrastructure-less operation in mobile ad-hoc networks has been explored in the past, such approaches attempt to establish an end-to-end path for communications, which may be unstable and exhibit bad performance over multiple hops; and often the node density may be insufficient to establish such paths in the first place. In this demo, we apply the concept of asynchronous message passing using delay-tolerant networking to audio messages which get recorded and then transmitted via store-carry-and-forward from the source to the destination. We present an implementation of the DT-Talkie on the Nokia Internet tablets, leveraging the DTN reference implementation developed in the concept of the DTN Research Group.**

***Index Terms*— DTN, Mobile Ad-hoc networking, messaging**

I. Introduction

Walkie-talkie style communication over cellular networks is gaining popularity with increasing number of mobile devices around the world. Walkie-talkie like cellular phone service provides a way to communicate one-to-one and one-to-many. There is a version of walkie-talkie style communication called "Push-to-talk over Cellular" (PoC) that uses SIP and RTP protocols to communicate over Internet architecture in the walkie-talkie fashion.

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.

The traditional walkie-talkie like cellular phone services over either cellular networks or operator independent wireless networks are infrastructure based. But the mobile users may travel around infrastructure-less environment while talking. These 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 these services may not work in the environment that suffers from high delay, high error rates and low data rates.

Delay Tolerant Networking (DTN) is an approach that uses store-carry-forward message switching mechanism to overcome the problems associated with intermittent connectivity, long or variable delay, low bit rates, high error rates and frequent network partitions. The IRTF DTNRG has proposed bundle protocol which overlays on top of heterogeneous link layer technologies. The bundle protocol defines a series of contiguous data blocks as a bundle. Each bundle is variable in size that carries application layer data. In DTN architecture, a bundle node is an entity that can send and receive bundles. Each bundle node has three components: one bundle protocol agent, zero or more convergence layer adapters and an application agent [3]. All nodes in the DTN are identified by a unique endpoint identifier (EID) that has a general structure: <scheme name>:<scheme-specific part, or "SSP">.

In this paper we describe a prototype application called DT-Talkie which enables mobile users to communicate over challenged environments in the walkie-talkie fashion. DT-Talkie is implemented for Maemo based Nokia Internet Tablet. We are porting DT-Talkie application to smartphone, Linux and Mac based PC. DT-Talkie supports both one-to-one and group communication. Section 2 describes the DT-Talkie architecture. The DT-Talkie implementation is presented in Section 3, followed by the demonstration setup in Section 4.

II. System Architecture

The general architecture of DT-Talkie application is shown in Figure 1. DT-Talkie application comprises of three modules: GUI, Audio C/P (Capture/Playback) and Bundle S/R (Send/Receive). dtnd runs as a background process to provide the bundle protocol services to the application by means of dtnd APIs.

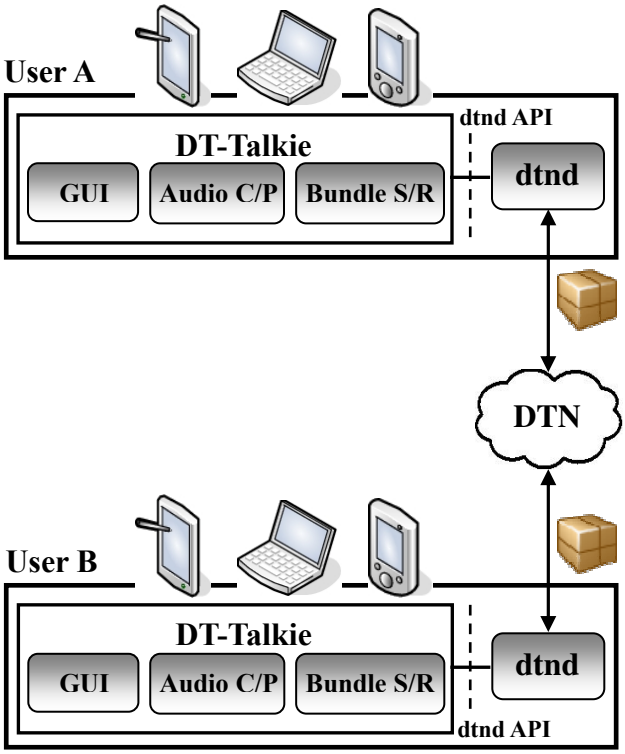

Fig. 1. DT-Talkie Architecture

We describe the architecture in Figure 1 with the aid of a one-to-one communication scenario. Suppose, user A wants to talk with user B in the walkie-talkie fashion. User A selects the EID of user B, presses a button 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. In the scenario of group communication, DT-Talkie multicast a voice message from User A to a group of mobile endpoints.

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.

III. Implementation

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 [2]. DTN2 supports TCP, UDP, Bluetooth, Ethernet and Sneakernet convergence layers. DTN2 includes support for several link types such as always-on links, on-demand links, opportunistic and scheduled links. 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.

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. We use GStreamer framework in the DT-Talkie Audio C/P module to capture and playback voice messages. DTN2 APIs are used in the Bundle S/R module of DT-Talkie for sending and receiving bundles over DTN architecture.

IV. Demonstration Setup

In the demonstration setup, we use four Nokia Internet Tablets, two laptops and one smartphone. The latest release of DTN2 (version 2.6.0, released in July 2008) is ported to those devices. Flooding protocol is used to forward voice messages. For persistent storage service, we use local file system that is used to store bundles.

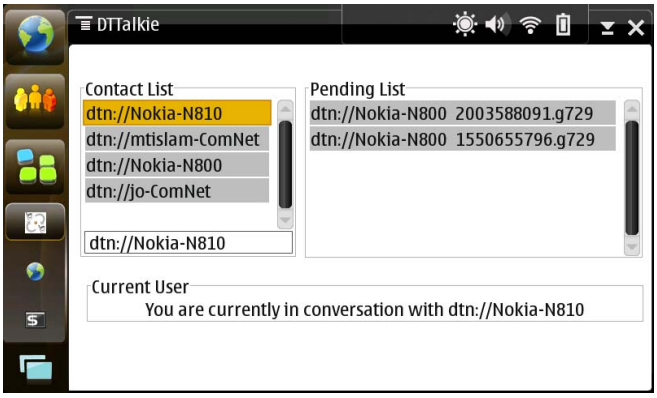

Fig. 2. DT-Talkie Screenshot

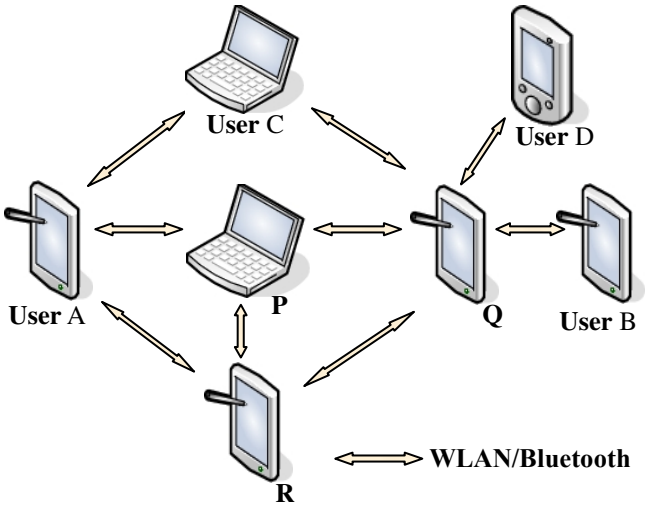

Fig. 3. Demonstration Scenario

In Figure 3, DT-Talkie and dtnd are running in the devices of User A, B and C. Only dtnd is running in the devices labeled as P, Q and R. We create artificial disruptions to get the DTN experience through disconnecting the link between the node Q and User B. When User A sends a voice message destined to User B, the message is queued in the node Q. After a while, we enable the connection between the node Q and User B. The node Q then forwards the message to User B as soon as the link becomes available between itself and User B. In the same setup, we also demonstrate the scenario of group communication through sending voice message from User A to User B, C and D.

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] DTN Ref Implementation, http://dtnrg.org/wiki/Code

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