

D-Book: A Mobile Social Networking Application for Delay Tolerant Networks

Russell J. Clark
College of Computing
Georgia Institute of
Technology
Russ.Clark@gatech.edu

Evan Zasoski
College of Computing
Georgia Institute of
Technology
Evan.Zasoski@gmail.com

Jon Olson
College of Computing
Georgia Institute of
Technology
jsolson@gatech.edu

Mostafa Ammar
College of Computing
Georgia Institute of
Technology
ammar@cc.gatech.edu

Ellen Zegura
College of Computing
Georgia Institute of
Technology
ewz@cc.gatech.edu

ABSTRACT

We describe a mobile application for social networking that demonstrates the features of *DTN#*, a .Net-based implementation of the DTN bundle specification and testbed for research in challenged networks. The application, dubbed D-Book, includes the ability for users to create, modify, and share profiles much like the popular web-based social networking services.

Categories and Subject Descriptors

H.4.3 [Information Systems Applications]: Communications Applications

General Terms

Design, Experimentation

Keywords

DTN, application, development, social networking

1. INTRODUCTION

The unique problems related to challenged networks have become an important research area in recent years [3]. Defined as either Delay Tolerant or Disruption Tolerant Networks, these so-called DTNs have several unique characteristics including excessive delays and partial disconnections. The Delay Tolerant Networking Research Group within the IETF has created several important reference documents in the DTN area including the DTN Architecture [2] and the Bundle Protocol Specification [4].

Along with this standardization work has come several implementation projects. The DTN2 reference architecture available at www.dtnrg.org provides a research and development platform for Linux/Unix OS variants. The SPINDLE [1] DTN system is being developed at BBN under the DARPA Disruption Tolerant Networking program and builds upon DTN2. This paper describes the design and

demonstration of *DTN#*, a new research and development implementation for Windows .Net.

2. DTN#

DTN# is a new, independent implementation of the DTN Bundle Specification [4] developed to support DTN research and application development. *DTN#* was developed for the .Net Platform and is being used on numerous devices including Windows laptops and desktops as well as Windows Mobile portable devices. This project was motivated in part by the desire to deploy DTN applications on commonly available mobile devices supporting the Windows Mobile platform.

The design goals for *DTN#* include a modular design, interoperability with other implementations (e.g. DTN2, SPINDLE), a simple application API, and minimal system resource requirements. In particular, it was important not to require a backend database server to facilitate deployment on limited resource devices. The overall architecture for *DTN#* is depicted in Figure 1.

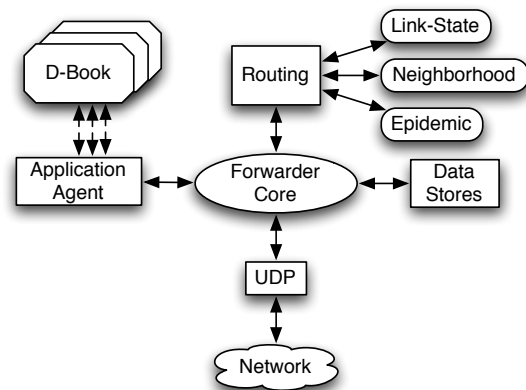


Figure 1: DTN# Architecture

The heart of the *DTN#* architecture is the *Forwarder Core* which is deployed on all participating DTN nodes, re-

ardless of which applications are running. The forwarding function is separated from the *Routing* function to facilitate work on new routing protocols. As a research platform, *DTN#* is designed to support multiple routing protocols and allow students to easily test new routing concepts. The specific routing approaches considered to date are Neighborhood Router (i.e., hosts to which we are directly connected are routable), an Epidemic router, and a Link-State router. A number of probabilistic routing schemes are also being considered. For the current project, a significant extension was made to support multicast delivery in *DTN#*. This work includes a multicast group registration mechanism as well as extensions to the forwarder to support multicast.

In addition to the demonstration application described in the next section, other applications developed with *DTN#* include a sample chat application and an image sharing application known as DEPICT.

3. D-BOOK FEATURES

The D-Book Social Networking Application was developed as part of the master's degree project for one of the authors (Zasoski). It is a distributed mobile application that allows users to create, modify, and share profiles much like with the popular on-line social networking site Facebook

The goals of the application are to:

- Create an application that will showcase various aspects of disruption tolerant networking in a manner that works well from both the standpoint of a demonstration and as a long-running application.
- Provide a foundational application onto which other, smaller applications can easily be attached, allowing for a central jumping off point for *DTN#* applications, ideas, and, concepts which may not be of sufficient scope to warrant their own stand-alone projects.

To this end, D-Book offers the following set of features:

- Users can own and edit a personal profile, the master copy of which is stored on the user's own mobile device. The current profile contains information about the user's basic personal information, contact information (if desired), and interests but can be expanded easily and arbitrarily to contain any information that a developer may wish it to contain.
- Users can search for other profiles based upon common interests or other search criteria. Searches are distributed throughout the entire system much like a Gnutella network using *DTN#*'s multicast functionality.
- Search results are cached for later access. This means that even if the primary device is inaccessible, matching profiles may be retrieved from devices that have cached a copy of that profile.
- Users can "friend" other users, essentially subscribing to their profile. Changes to friended profile are automatically flagged and downloaded.
- This is facilitated through the use of heartbeats, which send out basic information about profile changes and

status once every minute. Heartbeats are used to determine the "goodness" of a cached profile. A stored profile whose heartbeat can be heard and which says that the cached profile is up to date is good. One whose heartbeat cannot be heard is unsure. And once a heartbeat is heard which says that the cached copy is out of date, that copy is bad.

- Users can send one to one messages to other users on the system. The messaging system is already set up to be expanded to allow for one to all and one to many messages, but this has not yet been implemented.
- Developers can easily create and attach new applications to the D-Book application. These applications may function as stand-alone applications or they may make use of the information stored in the cached and locally owned profiles, including creating new types of information to store and update as they see fit.

4. D-BOOK ARCHITECTURE

The design of the D-Book application largely follows the Model-Controller-View school of design. The models are the profiles, messages, and applications, the controller is the *SocialBackend* class, and the view is the GUI.

The model portion of the D-Book application is divided into three Microsoft Visual Studio projects: They are the *Dtn.Social.Profiles*, *Dtn.Social.Messaging*, and *Dtn.Social.Applications*

The *Dtn.Social.Profiles* project defines the structure and content of a user profile and contains all of the classes relevant to creating, altering, and serializing these profiles.

The *Dtn.Social.Messaging* project contains the classes which define the formats of the messages to be exchanged between users. The project contains the abstract class *Message*, the enumeration *MessagingType*, and the only (currently) implemented message type, *OnetoOneMessage*. This can be extended to leverage multicast for on-to-many user messages.

The *Dtn.Social.Applications* project contains the classes pertaining to the extensible applications functionality. This project defines the abilities of developers to create new applications and tie them into the D-Book application with what is hoped to be a minimum of effort.

New applications can be developed by creating a new class which inherits from the abstract class *Application* and implements all of its abstract methods and fields. The new application should then be added to the Applications data structure in *Dtn.Social.Backend.SocialBackend* (discussed below), which is handled in the *fillApplications()* function.

The *Dtn.Social.Backend* project contains the primary control functionality for the entire application. It stores and manages the profiles, the messages, and the applications. It is responsible for searching for profiles both locally and across the network, it handles all incoming bundles, and it handles the messaging functionality.

Searching is a central feature of the D-Book application. The primary user interface provides a search form where users can enter the criteria from which they would like to identify other users. The Backend then creates a search request bundle containing the criteria as well as the user's *SerialNum* and the current *searchSerial*. The search request is then broadcast out to all nodes listening on the search channel through the multicast functionality.

Any node receiving the query will then search its own profile, its friends, and its cached profiles to see if any of them contain a match to the criteria. For every match found, the node creates a search response bundle containing the querying node's *serialNum* and *searchSerial* for verification purposes as well as a copy of the matching profile. As in a Gnutella network, search responses tend to be returned in bursts rather than all at once, so it behooves the searcher to periodically recheck the search results.

Search requests are propagated to all possible nodes using the multicast features of the *DTN#* forwarder. A multicast group Endpoint Identifier (EID) was created for users of the D-Book application and all D-Book users subscribe to this EID.

On top of the D-Book application itself, there are several supplementary applications for various purposes. These include the following.

- **PictureApp** This application is a sample and tutorial of how to create new applications that tie into the D-Book application. It allows users to upload images, tag them with search criteria, and search for images based upon their criteria.
- **Dtn.Applications.Social.desktop** A desktop friendly version of the D-Book application. This is primarily used for for testing and debugging.

- **Dtn.Applications.Social.ProfMake** A desktop based profile-making utility. This allows users to pick a profile picture and set all of basic fields. Since profile creation is typing intensive, allowing users to create them with a utility that allowed for the use of a keyboard seemed like a natural choice.

5. DEMONSTRATION SPECIFICS

The application is being demonstrated at CHANTS using a Windows XP laptop and two Dell Axim devices running Windows Mobile 5. The devices communicate using WiFi.

The demonstration includes exercising the features of the application including searching via multicast and changing a user's profile and observing how the change is propagated to that user's "friends".

6. REFERENCES

- [1] R. K. et al. The SPINDLE disruption-tolerant networking system. In *Proceedings of IEEE MILCOMM 2007*, Orlando, FL, USA, Oct. 2007.
- [2] V. C. et al. RFC 4838: Delay-tolerant network architecture, Apr. 2007. Status: INFORMATIONAL.
- [3] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of ACM SIGCOMM 2003*, Karlsruhe, Germany, Aug. 2003.
- [4] K. Scott and S. Burleigh. RFC 5050: Bundle protocol specification, Nov. 2007. Status: EXPERIMENTAL.