Backup and Bypass: Introducing DTN-based Ad-hoc Networking to Mobile Phones

Omar Mukhtar
Helsinki University of Technology
Networking Laboratory
omar.mukhtar@tkk.fi

Jörg Ott
Helsinki University of Technology
Networking Laboratory
jo@netlab.tkk.fi

ABSTRACT

Today's powerful networked personal computing devices offer a solid technical basis for mobile ad-hoc networking in support of consumer applications independent of operator networks. However, running Internet protocols directly on top ad-hoc routing protocols such as AODV requires a sufficient node density to establish end-to-end paths. In contrast, Delay-tolerant Networking (DTN) allows to exploit device capabilities also in sparse environments. In this demonstration paper, we present a DTN prototype for mobile phones as the most widespread platform for (delay-tolerant) ad-hoc networking and show a sample application that allows bypassing cellular operator infrastructure—with a fallback option in case DTN fails to deliver the information in time.

Categories and Subject Descriptors

C.2.6 [**Computer Communication Networks**]: Internetworking; D.4.4 [**Operating Systems**]: Communications Management

General Terms

Performance, Design, Reliability, Experimentation

Keywords

Delay-tolerant Networking, DTN, Mobile Ad-hoc Networking, Symbian, Mobile phone

1. INTRODUCTION

Mobile ad-hoc networking has been studied for many years but, despite its potential for consumer devices, its practical application is yet largely restricted to closed sensor networks and other controlled (e.g., tactical) deployments. While consumer applications using ad-hoc networks have been [1] and continue to be investigated [6], the focus of wireless communcations for consumers is clearly on infrastructure-based networking, using cellular networks (GPRS, UMTS), WLAN, or WiMAX.

However, infrastructure-based (cellular) communication may be suboptimal for at least two reasons: 1) Network coverage may not be available (e.g., in remote regions), yet, communication applications should be able to work as long as users are sufficiently close to each other to communicate indirectly, using ad-hoc networking as a *backup*. 2) Even if network coverage is available, communication may often be localized to some extent (e.g., for coordination among parties at a conference) so that using an expensive infrastructure (both in terms of cost and routing efficiency) should be avoided. This makes it desirable to be able to *bypass* infrastructure networks if the communicating peers can do so, yet allow them to fall back to the infrastructure if ad-hoc communication fails.

With the increasing penetration of personal devices capable of wireless communications (often supporting several link layer technologies), exploiting mobile ad-hoc networking among those devices becomes feasible. However, in the real-world of consumer communications, as opposed to well-deployed sensor networks or meshed infrastructures, traditional ad-hoc networking itself is usually not sufficient because the density of nodes may be far too low to establish a network layer end-to-end path between two communicating peers. Even if a sufficient number of mobile devices is around, their owners may not be willing to cooperate to save their own resources e.g., energy) or because they fear misuse; device heterogeneity may inhibit interworking; and radio range and interference may limit communications.

This suggests asynchronous information exchange using Delay-tolerant Networking (DTN), briefly introduced in section 2, to enable communications—even though the asynchronous interaction paradigm inherent to DTN does not allow addressing all aspects of interpersonal communications. We use mobile phones as experimentation platform: they are the most widespread mobile personal devices, users tend to always carry them, they are probably the only devices virtually always turned on, and they already form a center for personal communications. In this paper, we present a prototype DTN implementation for mobile phones that run the Symbian operating system. Our prototype uses Bluetooth and WLAN interfaces to enable ad-hoc communication independent of wireless operator infrastructure. We introduce Symbian and our implementation architecture in section 3 and present application considerations and our demonstration scenario in section 4. Section 5 summarizes our findings and suggests some future work.

2. DELAY-TOLERANT NETWORKING

Supporting communications in situations with intermittent connectivity is one goal of delay-tolerant networking (DTN) [5]. DTN concepts have been applied to various application scenarios, including sensor networks and intermittently connected communication infrastructures for remote areas. DTN uses asynchronous communications (modeled after email): Rather than relying on end-to-end communications using (small) packets, DTN endpoints exchange

REALMAN'06, May 26, 2006, Florence, Italy.
ACM 1-59593-360-3/06/0005.

messages of of arbitrary size (*bundles*) forwarded by DTN routers hop-by-hop from the source to the destination. Conceptually, DTN operates above the transport layer and may interconnect different internets with arbitrary underlying protocol stacks [5]. A *custody transfer* mode allows ensuring reliable delivery while delegating the responsibility to the next capable DTN router. Status reports may convey information about the delivery progress of a bundle and include forwarding and custody notifications from intermediate routers as well as receipts from the receiver to achieve end-to-end semantics [11].

DTN endpoints and applications are identified by *endpoint identifiers* specified in a URI-style format: *scheme:scheme-specific-part*. *scheme* defines the scope within which the *scheme-specific part*, i.e., the actual address, is interpreted. Any syntactically correctly formed URI scheme is acceptable; but only the "dtn:" scheme that takes an arbitrary string as address part is defined for DTN so far. For our experiments, we use both "tel:" and "mailto:" URIs for user identification.

DTN routing and forwarding offers particular challenges due to the delay tolerance of communications. While (Internet and) ad-hoc routing protocols determine an end-to-end path when they have to forward a packet (and may report an error if none is found), DTN routing must also account for potential or known future paths so that the complexity of forwarding decisions increases [7]. Furthermore, if paths are only available intermittently, no continuous exchange of routing information is possible. And, for links that become available only opportunistically, like with ad-hoc networking, predictions are difficult. For mobile ad-hoc networking environments, DTN routers rely on various kinds of information replication for forwarding bundles to maximize delivery probability and minimize transit time. Such forwarding schemes, for example, follow the concept of *epidemic routing* [12] of which numerous variations have been developed such as [9] [8]. In our demonstrator, we use simple epidemic routing based upon flooding with a finite lifetime for messages. Nodes keep a history of discarded messages so that they will accept every message only once. Purging messages occurs after timeout or when successful delivery of a message is indicated, i.e, when the corresponding status report passes by.

3. IMPLEMENTATION ARCHITECTURE

The demonstration setup comprises two applications described below, developed for Symbian-based mobile phones. The Symbian OS is a non-preemptive multitasking OS with a large set of APIs and library functions. It does not encourage use of threads and offers event-handling system in the form of active objects. Symbian-based mobile phones provide a variety of communication interfaces including WLAN, Bluetooth, infrared, GPRS and services such as telephony and messaging. Symbian offers a generic socket-based interface to access many of the underlying link layers. It also offers efficient resource and memory management via cleanup stack which works similar to a garbage collector. Their broad availability and their rich communication interfaces make Symbian OS phones a good choice for an implementation platform.

3.1 Bundle Protocol Application

The Bundle Protocol [11] specifies operations and messaging of nodes participating in the overlay delay-tolerant network. This specification is implemented by the Bundle Protocol Application (BPA). The different logical components of the BPA are shown in figure 1 below.

BPA Core Logic. The BPA core logic component implements the bundle protocol processing functionality including its state machine and is responsible for all bundle transactions. It accepts local messages for DTN-based transmission, encapsulates them, composes and originates all outgoing bundles, and decomposes incoming bundles to extract application data. It also processes, stores, and replicates bundles in transit.

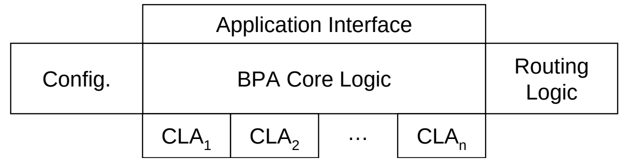

Figure 1: Software architecture of the bundle application

Configuration Interface. The BPA application can operate in a number of modes, including a bundle gateway or bundle router [2] as defined by means of the configuration interface. Configuration parameters include the endpoint ID, naming conventions for bundle nodes; the available types of convergence layer interfaces, routing types, etc. At this point, configuration parameters are loaded at application startup only.

Routing Logic Interface. The routing logic component is responsible for determining the route of an outgoing bundle. At the moment, we use only a simple epidemic routing scheme. But this component is kept extensible to allow incorporating other routing schemes via its generic interface to the core logic component.

Convergence Layer Adapters (CLAs). The CLAs are the components, which implement a generic interface for the underlying communication infrastructure. There may be one or more CLAs associated with one BPA. The convergence layer protocols and interfaces are not defined by the bundle protocol specification itself. Rather, auxiliary documents can specify logic and interfaces of such layers [3]. So far, our BPA supports TCP (via WLAN/GPRS) and Bluetooth convergence layers but is easily extensible to other convergence layer types due to the generic framework architecture. In particular, we are presently investigating the use of cellular (multimedia) messaging as further CLA.

Application Layer Interface. The BPA works like a session layer above the transport and below the application layer offering services to application programs to send and receive application data via a delay-tolerant network. The ALI component provides a socket-based interface to such programs to connect to the BPA which works like a server program. The generic socket-based interface makes it independent of the particular platform and device. Any application running on any device on that network can contact the BPA and use its services.

3.2 Demonstrator Application

Mobile phones are used for synchronous (audio or video phone calls, conferences) as well as asynchronous communications (text and multimedia messaging). We are considering signaling-specific optimizations for dealing with disconnections in synchronous communications elsewhere (see [10] for an outline) where do not assume an ad-hoc end-to-end path for conversational real-time communications either. In this paper, we focus on asynchronous applications such as the transmission of text messages, still images, and audiovisual clips via DTN—as an alternative to SMS and MMS services.

We have designed a small application which uses BPA services for communication and interfaces to the user via a small GUI. On the sending side, when invoked, the application checks whether new outgoing items were created by the user (e.g., a new image or video captured) and, if so, offers the item for transmission to the user. The user can specify the intended recipient (by selecting her phone number or email address from the address book) and decide which item(s) to send along. The user can further specify which interfaces shall be tried in which order and how long the user is willing to wait for successful DTN delivery before falling back to infrastructure communications, e.g., for (multimedia) messaging. The application takes the responsibility for the user to try the optimal (with respect to cost, etc.) underlying network. It will wait as necessary and choose the best available connection and send the user data whenever the network is available, based on user's preferences and configurations.

4. DEMONSTRATION SCENARIO

The demonstrator focuses on the prototype implementation and targets mainly two aspects of DTN-based ad-hoc networking: 1) the suitability of asynchronous interpersonal communications using the DTN specification implemented in today's commercial mobile phones with a suitable integration into the phone environment and 2) the interoperability of different types of phones Symbian phones using our prototype and a laptop-based system running a different DTN implementation [4].

Figure 2 shows our setup: We use a Nokia 9500 communicator, an N90 mobile phone, and (optionally) a 770 Internet Tablet. The laptop is an IBM ThinkPad T41p running MS Windows XP and cygwin. Interaction between devices is supported using (a) WLAN ad-hoc mode, (b) WLAN infrastructure mode in case a common access point or enhanced service set is available (which may yield quite extensive coverage in case of campus networks), (c) Bluetooth communications, or, as a fallback, (d) cellular operator services.[1]

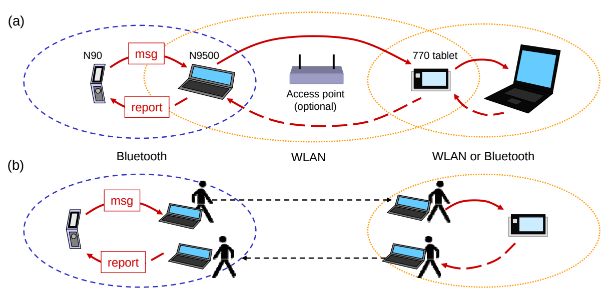

Figure 2: Demonstrator setup

Both types of mobile phones are equipped with digital cameras and allow recording still and moving images, composing text messages, and displaying the respective message contents. Both may serve as information source, sink, and DTN router. The same applies to the laptop except that it will not be used to capture video material. Forwarding messages will be demonstrated (a) by overlapping radio ranges of different hops (using different link layers) and (b) by means of physical carriage in a users device until another device comes in reach. In both cases, a confirmation will be returned upon successful DTN-based delivery. If success is not reported within the specified period, a fallback to operator infrastructure services is initiated. This is only done by the original message sender in order not incur cost on third parties. Message lifetimes and confirmations clean up the bundles stored in transit.

[1] We have not yet implemented the interfaces of the BPA to the mobile messaging services and plain (GPRS-based) cellular Internet access.

5. CONCLUSION

DTN-based ad-hoc networking enables interactions between mobile users based upon commodity devices and thus allows for ad-hoc communication in spite of the potentially low density of interoperable devices. Our prototype integrates the—widely used—mobile messaging applications with DTN-based ad-hoc networking and thus paves the way for autonomous operation of mobile devices: to cope with lack of or simply bypass infrastructure networks. Our implementation is available for download under GPL license[2] and thus allow other applications to use its services providing an abstraction from lower layers and disruptions. While our present demonstrator clearly focuses on the engineering aspects, next steps include further considerations on consumer applications coupled with investigations of DTN routing: taking into account user behavior, typical message sizes, and device capabilities but also addressing the options arising from closer integration of ad-hoc and infrastructure-based operation.

6. REFERENCES

[1] Homepage of FleetNet. http://www.fleetnet.de/, 2003.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H.Weiss. Delay-Tolerant Network Architecture. Internet Draft draft-irtf-dtnrg-arch-05.txt, Work in progress, March 2006.

[3] M. Demmer. Protocol for the DTN TCP Convergence Layer. Available via the code Wiki at http://www.dtnrg.org/, February 2006.

[4] DTN Reference Implementation, version 2.2.0. Available from http://www.dtnrg.org/wiki/Code, March 2006.

[5] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. *Proceedings of ACM SIGCOMM 2003, Computer Communications Review*, 33(4):27–36, August 2003.

[6] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket Switched Networks and Human Mobility in Conference Environments. In *ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005.

[7] S. Jain, K. Fall, and R. Patra. Routing in Delay Tolerant Networks. In *Proceedings of the ACM SIGCOMM 2004 Conference, Portland, OR, USA*, 2004.

[8] A. Lindgren and A. Doria. Probabilistic Routing Protocol for Intermittently Connected Networks. Internet Draft draft-lindgren-dtnrg-prophet-02.txt, Work in Progress, March 2006.

[9] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *The First International Workshop on Service Assurance with Partial and Intermittent Resources*, 2004.

[10] J. Ott, D. Kutscher, and O. Bergmann. SIP Voice Services for Intermittently Connected Users in Wireless Networks. In *Upperside Conference Wi-Fi Voice 2004*, May 2004. Available at http://www.drive-thru-internet.org/pub/wifivoice2005-ott-kutscher.pdf.

[11] K. L. Scott and S. C. Burleigh. Bundle Protocol Specification. Internet Draft draft-irtf-dtnrg-bundle-spec-04.txt, Work in progress, November 2005.

[12] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Tech Report CS-200006, Duke University, April 2000.

[2] http://www.netlab.tkk.fi/~jo/dtn/