SocialDTN: a DTN Implementation for Digital and Social Inclusion

Waldir Moreira[1], Ronedo Ferreira[2], Douglas Cirqueira[2],
Paulo Mendes[1] and Eduardo Cerqueira[2,3]
[1]SITILabs/University Lusófona, Lisbon, Portugal, [2]ITEC/Federal University of Pará, Belém, Brazil
[3]CSD/University of California, Los Angeles, USA
waldir.junior@ulusofona.pt, ronedo@aitinet.com, douglas.cirqueira@gmail.com,
paulo.mendes@ulusofona.pt, cerqueira@ufpa.br

ABSTRACT

Despite of the importance of access to computers and to the Internet for the development of people and their inclusion in society, there are people that still suffer with digital divide and social exclusion. Delay/Disruption-Tolerant Networking (DTN) can provide the technical support for the digital/social inclusion of these people as it allows opportunistic and asynchronous communication, which does not depend upon networking infrastructure. We introduce SocialDTN, an implementation of the DTN architecture for Android devices that operates over Bluetooth and aims to take advantages of users' social daily routines to improve data forwarding. As we want to exploit the social proximity and interactions existing among users, SocialDTN considers a social-aware opportunistic routing proposal, *dLife*, instead of the well-known (but social-oblivious) *PROPHET*. Some preliminary field experimentations are explained based on the direct delivery of content.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *network communications, store and forward networks, wireless communication*

Keywords

delay/disruption-tolerant networking; social proximity; Amazon riverside communities; digital/social inclusion

1. INTRODUCTION

The number of personal networked devices and Internet users increases everyday. This fact may have an important impact on the educational development of socially excluded people, and on their inclusion into society. Social exclusion is accentuated by the geographic isolation of some regions (e.g., Amazon regions), leaving some people completely out of the digital world.

Due to its capability of coping with intermittent connectivity and long delays, and due to the employment of the store-carry-and-forward approach, the implementation of the DTN paradigm based on low cost personal networked devices can be used to mitigate this digital divide. It can allow users of isolated regions (e.g., riverside communities of the Amazon region) to opportunistically and asynchronously access the Internet and communicate with other users inside and outside such regions.

We can find different implementations of the DTN architecture [1], such as DTN2, IBR-DTN, and Bytewalla. However, as social excluded communities do not count with any network infrastructure, such implementations are not suitable in our target scenario as they somewhat use infrastructure (e.g., AP as DTN endpoint/gateway/router in DTN2 and Bytewalla) to allow devices to communicate. Moreover, DTN solutions should exploit the social proximity and interactions patterns (e.g., due to daily habits and routines) that is present in these isolated communities in order to mitigate the lack of a network infrastructure. Although Bluetooth is supported by DTN2 and could be emulated with IBR-DTN, they do not exploit this technology to capture social proximity metrics. Bluetooth is only used by prior implementation for the exchange of data among neighbor nodes.

Thus, we present SocialDTN, an implementation of the DTN architecture [1] and Bundle Protocol [8] for Android devices that takes advantage of the social proximity and daily routines of users to exchange bundles, even in the absence of any network infrastructure. In order to exploit social proximity and interactions between users, SocialDTN introduces a Bluetooth Convergence Layer aiming to consider only socially well-connected users (i.e., devices) in the exchange of bundles. The goal is to increase the probability of message delivery while avoiding waste of network and storage resources. As social proximity relates to how physically close users are and how much time they spend together, Bluetooth is the most indicated technology to exploit such proximity as it supports only close range communication.

Additionally, the exchange of bundles solely between socially well-connected users is to be achieved with *dLife* [5], which computes the social weight among users based on their interactions and the importance that users have in different periods of their daily routines. This explains why we use *dLife* over the social-oblivious *PROPHET* [3] in the context of this work. For a better understanding on the functioning and performance of *dLife*, we point the reader to [6, 4].

LCDNet'13, September 30, 2013, Miami, Florida, USA.
ACM 978-1-4503-2365-9/13/09.
http://dx.doi.org/10.1145/2502880.2502892.

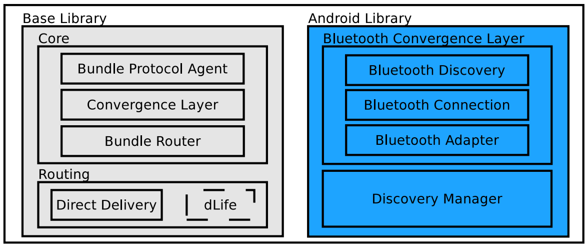

Figure 1: SocialDTN architecture

This work is a joint project between SITILabs and Federal University of Pará (UFPA) called DTN-Amazon, in which we are developing networking technology to mitigate the effects of digital divide and social exclusion in the riverside communities close to the UFPA campus in Belém, Brazil.

The remaining of this paper is structured as follows. Section 2 briefly presents the related work. The SocialDTN implementation is presented in Section 3. Section 4 provides our first experimentations to test the Bluetooth Convergence Layer (BCL) and *dLife*, as well as to check the behavior of SocialDTN on a direct contact delivery scenario. Finally, Section 5 concludes the paper and presents some future steps of the DTN-Amazon project.

2. RELATED WORK

There are different implementations of the DTN architecture from which the ones most related to this work are DTN2, IBR-DTN, and Bytewalla. As SocialDTN, these implementations aim at provide communication means in challenging scenarios (e.g., intermittent/disruptive communication, long delays). However, SocialDTN is meant to exploit social proximity and interaction patterns between users' devices, which can be achieved through the use of Bluetooth technology. Additionally, as the target communities have no available infrastructure, its operation shall be independent of any infrastructure, which is not possible when using DTN2, IBR-DTN, and Bytewalla implementations.

DTN2[1] is the result of joint efforts within the Delay Tolerant Networking Research Group. This DTN reference implementation is flexible and can be used in various DTN-related experiments and real-world deployment. Despite the fact that DTN2 provides support for Bluetooth communication (Bluetooth RFCOMM Convergence Layer), this implementation is not readily available for operation over Android devices. Bluetooth is not used to capture social proximity and DTN2 does not use a social-aware routing protocol.

IBR-DTN first emerged to transform access points into DTN nodes to support mobile applications [2]. Compared to DTN2, IBR-DTN is a lightweight version that aims at efficient resource utilization. This implementation later evolved [7] to allow operation over Android devices. SocialDTN could be based on IBR-DTN as it complies with Android and can emulate Bluetooth rates (through IEEE 802.15.4). However, we do not want to achieve Bluetooth rates, but to capture social proximity through this technology.

Regarding Bytewalla[2], it is designed to run over Android devices to allow the asynchronous exchange of information between these devices and access points. Its latest version also allow the use of GPRS to reach the Internet. Bytewalla could have been the basis for SocialDTN, but Bluetooth has only been considered for neighbor discovery. As routing in SocialDTN is based on *dLife*, we need other functionalities (e.g., neighbor gone) besides the capability to exchange data over Bluetooth that is currently not supported in Bytewalla.

[1]https://sourceforge.net/projects/dtn/
[2]http://csd.xen.ssvl.kth.se/csdlive/content/dtn-bytewalla

3. SOCIALDTN

As mentioned earlier, current implementations of the DTN architecture (i.e., DTN2, IBR-DTN, Bytewalla) do not fully comply with our main requirements: i) exploit social proximity and interactions between users by using Bluetooth; and ii) be independent of any network infrastructure, since there is no infrastructure in the targeted communities.

Fig. 1 presents an overview of the SocialDTN architecture. It implements two libraries, namely the base library and the android library. For the sake of simplicity, we highlight only the most important parts of each library. The base library comprises core and routing functionalities. The core functionalities are based on the DTN architecture [1] for basic DTN operation. These functionalities implement:

- Bundle Protocol Agent [8] for bundle exchange-related operations, such as sent/received bundle notifications, carried bundles, and storage capacity of node;
- Convergence Layer that allows the interfacing between the bundle layer and the underlying Bluetooth RFCOMM. It defines the adapter and connection information (i.e., sockets, UUID, peering device) necessary for a bundle transfer;
- Bundle Router that provides the basic functions over the link, as well as connection information (i.e., link states), to be extended and used by the routing functionality.

The routing functionality refers to how forwarding takes place. At the moment SocialDTN can perform only Direct Delivery. However, our goal is to have bundles being exchanged also with users that are socially well-connected with the destination, and not only to have bundles being delivered directly to the destination. Thus, version 2.0 of SocialDTN will include the social-aware opportunistic proposal *dLife* [5] as its routing functionality. It is important to note that *PROPHET* [3] could have been used; however, it is a social-oblivious proposal[3] that does not exploit social interactions as we intend to use in SocialDTN. In our previous work [6, 4],we show why *dLife* answers our needs better than *PROPHET*.

Regarding the Android library, it extends the base library in order to cope with the Android environment. It defines:

- Bluetooth Convergence Layer (BCL), which allows SocialDTN to operate over the Bluetooth technology to exploit the aforementioned social proximity. BCL provides the Bluetooth Discovery, Connection, and Adapter mechanisms that get relevant information about neighboring peers and the connections between them (e.g.,

[3]Some authors may argue that *PROPHET* is a social-aware solution as it considers the history of encounters. However, we believe that social-aware solutions are based on utility functions able to identify/classify individuals or groups of these, i.e., interests, social ties, levels of social interactions.

Figure 2: SocialDTN usage by Riverside community

UUID, EID, sockets). Moreover, with BCL, nodes can straightforwardly exchange bundles through Bluetooth without the need to use a structured WiFi network.

- Discovery Manager that manages the life cycle of Bluetooth discovery process. SocialDTN relies on Bluetooth technology for exchanging data while keeping track of neighboring peers. This manager attempts to make the most out of the Bluetooth hardware. It ensures that SocialDTN discovers nearby nodes and sends data whenever is required without compromising (i.e., taking over) the hardware that is also shared by other active applications running in the device.

4. FIELD EXPERIMENTATION

SocialDTN first deployment test encompass seven Android devices carried by students during their daily routine activities in the UFPA campus for five days. A traffic generator was implemented to run in each device and to create a load of 6 messages/hour, towards the other six nodes. Nodes counted with a 10 MB storage and message size varied between 1 KB and 1 MB as to represent the different applications and data types (e.g., asynchronous chats, email, video sharing) that are expected to be used.

This field experiment aims to: i) evaluate the BCL implementation and generate the first DTN-Amazon contact traces; and ii) evaluate if the implementation of *dLife*, to be later integrated in SocialDTN v2.0, complies with its specification [5]. The used BCL and *dLife* implementations have shown to be sound: with the former we are able to detect social proximity among nodes; with *dLife*, we are able to determine the level of social interactions. Additionally, we can collect trace information that can be used in the ONE simulator. At the moment, BCL and *dLife* are undergoing revision as they are not suitably coping with concurrent connections.

Another experimentation (cf. Fig. 2) aimed to test the suitability of SocialDTN v1.0 to allow isolated people to get educational information. This experiment involved a health agent that acts in the riverside community of Combu Island. By using SocialDTN v1.0 (with direct delivery as routing functionality), SocialDTN provides this agent with useful information to be carried to this community. By using SocialDTN, the agent's device carries a video[4] about dengue prevention that he received while he was visiting the Bettina Ferro hospital in the UFPA campus. On the way back to the community, the agent checks for any information he should be using in his visit. The ability of using videos does increase the efficiency of the agent actions as the population in this area is illiterate.

[4]https://www.youtube.com/watch?v=JUDZ8hMnZeI

5. CONCLUSIONS AND FUTURE WORK

This work presents a new instantiation of the DTN architecture, called SocialDTN, which currently implements RFC 4838 and RFC 5050, as well as a new Bluetooth Convergence Layer. SocialDTN is being developed in the context of the DTN-Amazon project, aiming to promote the social/digital inclusion of the Amazon riverside communities in the vicinity of the UFPA campus.

As shown by our experimental work, SocialDTN is able to operate with no dependence upon network infrastructure.

As future work, we plan to: i) integrate *dLife* to SocialDTN v2.0 and compare it against the direct delivery of SocialDTN v1.0; ii) expand the experimentations in the riverside communities; iii) release the collected social traces to the scientific community.

Acknowledgment

Thanks are due to Dórian Langbeck for his co-authorship on this paper and implementation contributions, to the National Council for Scientific and Technological Development (CNPq), to the Amazon Research Foundation (FAPESPA), and to FCT for financial support of the User-Centric Routing project (PTDC/EEA-TEL/103637/2008) and the PhD grant (SFRH/BD/62761/2009) of Waldir Moreira.

6. REFERENCES

[1] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and Weiss, H. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), Apr. 2007.

[2] Doering, M., Lahde, S., Morgenroth, J., and Wolf, L. Ibr-dtn: an efficient implementation for embedded systems. In *Proceedings of the third ACM workshop on Challenged networks* (New York, NY, USA, 2008), CHANTS '08, ACM, pp. 117–120.

[3] Lindgren, A., Doria, A., and Schelén, O. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev. 7*, 3 (July 2003), 19–20.

[4] Moreira, W., and Mendes, P. Social-aware opportunistic routing: The new trend. In *Routing in Opportunistic Networks*, I. Woungang, S. K. Dhurandher, A. Anpalagan, and A. V. Vasilakos, Eds. Springer New York, 2013, pp. 27–68.

[5] Moreira, W., Mendes, P., Ferreira, R., Cirqueira, D., and Cerqueira, E. Opportunistic routing based on users daily life routine. Internet Draft, draft-moreira-dlife-02, work in progress, Apr. 2013.

[6] Moreira, W., Mendes, P., and Sargento, S. Opportunistic routing based on daily routines. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a* (2012), pp. 1–6.

[7] Morgenroth, J., Schildt, S., and Wolf, L. A bundle protocol implementation for android devices. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (New York, NY, USA, 2012), Mobicom '12, ACM, pp. 443–446.

[8] Scott, K., and Burleigh, S. Bundle Protocol Specification. RFC 5050 (Experimental), Nov. 2007.