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, [2]ITEC, Federal University of Pará, [3]CSD, UCLA
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 help 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, taking advantages of the social daily routines of users. As we want to exploit the social proximity and interactions existing among users, SocialDTN includes a social-aware opportunistic routing proposal, *dLife*, instead of the well-known (but social-oblivious) *PROPHET*. Simulations show the potential of *dLife* for our needs. Additionally, some preliminary results from field experimentations are presented.

Categories and Subject Descriptors

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

General Terms

Design, Experimentation

Keywords

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

1. INTRODUCTION

The number of computers and Internet users increase everyday as they play an important role not only in the educational development of people, but also in the inclusion of

people into society. However, due to being geographic dispersed from the major cities (e.g., Amazon regions), some people are completed left 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 paradigm, DTN 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.

One can find different implementations of the DTN architecture [1] that can be employed in such scenario, such as DTN2, IBR-DTN and Bytewalla. However, as social excluded communities do not count with any network infrastructure, but still present social proximity and interactions patterns (e.g., due to daily habits and routines), these implementations do not answer our needs as they do not operate over Bluetooth in order to capture social proximity, do not employ routing approaches able to exploit social interation patterns, and depend on some infrastructure (e.g., AP) to allow user devices to communicate.

Thus, we present SocialDTN, an implementation of the DTN architecture and Bundle Protocol [2] 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, in order to increase the probability of message delivery while avoiding waste to network and storage resources. Social proximity relates to how physically close users are, and how much time they spend together. This means that 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 achieved with *dLife* [3, 4], which computes the social weight among users considering their social interactions and the importance that users have in different periods of their daily routines.

We present some simulation results considering the social-aware *dLife* [3] and the social-oblivious *PROPHET* [5] in a trace scenario aiming to show how useful social proximity is for SocialDTN. Also some preliminary results of SociaDTN based on field experiments are presented.

This work is part of a joint project between SITILabs and Federal University of Pará (UFPA) called DTN-Amazon.

This project aims to develop networking technology to mitigate the effects of digital divide and social exclusion in the riverside communities close to the UFPA campus in Belém, Pará, Brazil.

The remaining of this paper is structured as follows: We present the SocialDTN implementation in Section 2. Section 3 provides a description of the simulation and field experimentation results. On Section 4 we conclude our arguments and and present some future steps of the DTN-Amazon project.

2. SOCIALDTN

Fig. 1 presents the SocialDTN implementation. It implements the DTN architecture based on RFC 4838 [1] and RFC 5050 [2] for basic DTN and bundle exchange operations. As mentioned earlier, current implementatons of the DTN archictecture (i.e., DTN2, IBR-DTN, Bytewalla) do not comply with our main requirements: i) exploit social proximity and interactions between users; and ii) be independent of any network infrastructure. With SocialDTN social proximity is exploited by using Bluetooth as it is a close-range communication technology, but none of such implementations are functional over this wireless technology. What is more, these implementations depend somewhat of some infrastructure (e.g., an AP) to aid the communication between endpoints, which is not feasible (due to its nonexistence) for the targeted communities.

Figure 1: Overview of SocialDTN

Thus, the Bluetooth Convergence Layer (BCL) was implemented to allow SocialDTN to operate over the Bluetooth technology and to allow exploitation of the aforementioned social proximity. With BCL, nodes can straightforwardly exchange bundles through the Bluetooth interface without the need of a structured WiFi network. SocialDTN takes advantages of some native Bluetooth functionality already implemented in Android, such as the discovery of neighbors and the possibility of storing information about them: BCL comprises a Service Discovery Protocol (SDP) for sensing the medium and the Serial Port Profile (SPP) for data exchange. Moreover, it uses RFCOMM as transport protocol and runs on top on the Logical link control and adaptation protocol (L2CAP), which interfaces with the Host Controller Interface (HCI).

Since we want the exchange of bundles to happen mostly between socially well-connected users, we want to make use of these social interactions. This leads us to the opportunistic routing component, which is responsible for deciding what is the best way for a bundle to reach its destination. The routing agent of SocialDTN is implemented based on the specification of the social-aware opportunistic proposal

dLife [3, 4]. It is important to note that *PROPHET* [5] could have been used; however, it is a social-oblivious proposal[1], and does not explore the social interactions as we intend to use in SocialDTN. In Section 3, we show why *dLife* answers our needs better than *PROPHET*.

3. RESULTS

This section is divided into two parts: first, to show that the social-aware *dLife* [3, 4] is able to perform better than the social-oblivious *PROPHET* [5]; second, to show our preliminary field experiments.

3.1 Social-aware vs. Social-oblivious

We want to exploit social proximity and interactions with SocialDTN. Thus, we carried out simulations on the Opportunistic Network Environment (ONE) simulator using the CRAWDAD traces of Cambridge that comprises contact information of 36 students carrying these devices throughout their daily routines [6]. Our goal is to show that *dLife* can capture this social proximity, and can perform better than *PROPHET* [5] justifying our choice for the former for routing data in socially excluded communities, which have similarities to a university small community without Internet access (e.g. short geographic dispersion, small number of people).

A total of 6000 messages are generated with size ranging from 1 kB to 100 kB. The source/destination pairs remain the same for the simulations of both proposals. Since different applications (e.g., email, asynchronous chat) are expected to run over the proposed module, we set message Time-To-Live (TTL) in 1, 2, 4 days, 1 week and 3 weeks. The results are presented with 95% confidence interval and in terms of average delivery probability (ratio between the number of delivered messages and total number of created messages) and cost (number of replicas per delivered message).

In Figure. 2 and Figure 3, we can observe that *dLife* has better overall performance than *PROPHET*. Regarding delivery, both proposals are affected due to the very low number of contacts (average 32 per hour) in the scenario. Thus, determining social weight/node importance (*dLife*) and delivery predictability of nodes (*PROPHET*) takes longer to offer both proposals a stable view of the network in terms of the utility functions that both use. Yet, *dLife* has a subtle advantage over *PROPHET*.

When it comes to cost, clearly we see the advantage of relying on social awareness. The advantage of *dLife* over *PROPHET* is of approximately 25 times better: *dLife* creates an average of 24.56 replicas to perform a successful delivery, while *PROPHET* creates 538.37 replicas.

With these results, we can see that *dLife* indeed captures the social proximity we want to exploit in the context of DTN-Amazon: the lower number of replicas suggests that only well socially-connected nodes exchange content. This allows us to have content easily going from the UFPA campus to the riverside communities through nodes which are really involved in the the process of digital/social inclusion.

---

[1]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 much more elaborated and defined based on utility functions able to identify/classify individuals or groups of these, i.e., interests, social ties, levels of social interactions.

Figure 2: Delivery performance with human traces

Figure 3: Cost performance with human traces

3.2 Field Experimentations

SocialDTN first deployment tests consider 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 present in the experiments. Nodes counted with a 10MB storage and message size varied between 1KB and 1MB as to represent the different applications and data types (e.g., asynchronous chats, email, video sharing) that are expected to be used.

The goal of these tests are to: i) evaluate the BCL implementation and generate the first DTN-Amazon contact traces; and ii) test if the implementation of *dLife* complies with its specification [4]. Both BCL and *dLife* have shown to be sound: with the former we are able to detect the social proximity among nodes and explore their social interactions with the latter. Additionally, we can collect trace information that can be used in the ONE simulator.

Another experimentation (cf. Fig. 4) involved a health agent that acts in the riverside community of Combu Island. As SocialDTN "knows" this agent has a strong interaction towards this community, the device he carried received a video[2] about dengue prevention while he was visiting the Bettina Ferro hospital in the UFPA campus. On the way 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. 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 Con-

---

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

Figure 4: SocialDTN usage by Riverside community

vergence Layer and a new opportunistic routing protocol, *dLife*. 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 University campus.

As shown by our experimental work, SocialDTN is able to explore the social proximity and interactions and to operate with no dependence upon network infrastructure.

As future work, we plan to: i) have BCL supporting multiple connections; ii) include other opportunistic routing solutions, such as *Epidemic* and *PROPHET*; iii) expand the experimentations in the riverside communities; iv) release the collected social traces to the scientific community.

5. REFERENCES

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

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

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

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

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

[6] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/content (v. 2006-09-15)." Downloaded from http://crawdad.cs.dartmouth.edu/, 2006.