Ad Hoc Networks 25 (2015) 330–345

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Interest-awareness in data dissemination for opportunistic networks

Radu-Ioan Ciobanu [1], Radu-Corneliu Marin, Ciprian Dobre *, Valentin Cristea

Faculty of Automatic Control and Computers, University Politehnica of Bucharest, 313, Splaiul Independentei, Bucharest, Romania

ARTICLE INFO

Article history:
Received 4 March 2014
Received in revised form 19 July 2014
Accepted 23 July 2014
Available online 7 August 2014

Keywords:
Opportunistic
Dissemination
Interests
Social
Prediction

ABSTRACT

Generally, data dissemination in opportunistic network uses flooding to ensure that content is spread to interested subscribers. However, this might lead to congestion and high overhead, so alternative solutions are required. In this article, we propose leveraging context information such as node interests, social connections and predictions based on contact history, in order to decrease congestion without affecting the network's hit rate and delivery latency. Thus, we first propose a basic interest-based dissemination algorithm, to show that nodes tend to group together based on interests. Then, we present ONSIDE, an algorithm that also uses other types of context information to select the nodes that act as forwarders. Finally, we propose five heuristics for sorting the messages in a node's memory, and show how each of them affects hit rate, delivery latency and congestion.

1. Introduction

The emergence and wide-spread of new-generation mobile devices, as well as the increased integration of wireless technologies such as Bluetooth and WiFi, create the premises for new means of communication and interaction, challenge the traditional network architectures and are spawning an interest in alternative, ad-hoc networks such as opportunistic networks (ONs). ONs consider node mobility an opportunity to exploit, in environments where human-carried mobile devices act as network nodes and are able to exchange data while in proximity. Node mobility creates contact opportunities among nodes, which can be used to connect parts of the network that are otherwise disconnected. Whenever a destination is not directly accessible, data is opportunistically forwarded between nodes, so the peers can bring the message closer to the destination.

An application scenario for ONs is represented by *data dissemination*. Data dissemination implies that the nodes in a network subscribe to certain channels, which publish data. Whenever data is published by a channel, nodes that are subscribed to it must receive the data. Thus, nodes play the roles of "publisher" and "subscriber" at the same time. While in regular networks there is a central entity that keeps track of which channel each node is subscribed to, since ONs are fully decentralized and composed of highly mobile nodes, the subscriptions of a network participant are only known to itself and to nodes it comes in contact with during the ON's lifetime. Thus, to cope with the locality of information, many dissemination strategies flood the network, by sending all channel data and subscription information to every peer a node comes in contact with. However, previous studies show that this not only leads to very high bandwidth usage, but also potentially to

* Corresponding author. Address: University Politehnica of Bucharest, 313, Splaiul Independentei, Office EG403, Sector 6, 060042 Bucharest, Romania. Tel.: +40 745 174 359; fax: +40 318 163 028.

E-mail addresses: radu.ciobanu@cti.pub.ro (R.-I. Ciobanu), radu.marin@cti.pub.ro (R.-C. Marin), ciprian.dobre@cs.pub.ro (C. Dobre), valentin.cristea@cs.pub.ro (V. Cristea).

[1] Address: University Politehnica of Bucharest, 313, Splaiul Independentei, Office EG403, Sector 6, 060042 Bucharest, Romania. Tel.: +40 724 225 933.

http://dx.doi.org/10.1016/j.adhoc.2014.07.004
1570-8705/

congestion [1]. We expand on this, and propose a method to help alleviate such problems, without affecting the hit rate.

As previous studies show, ON nodes tend to encounter other nodes with common interests with a high rate [2–4]. We believe this happens because humans generally form social groups (i.e. communities) based on similar tastes or preferences. For example, work colleagues, which are in contact for 8–9 h a day, have common interests regarding the domain they work in. In the same community, people sharing common interests (such as sports) are more likely to "bond" together. This implies that, for the period when a mobile device-carrying person is at work, the corresponding ON node will interact with other nodes with common interests. However, this situation is not only valid for work hours. After work or in weekends, people meet their friends, and it is very likely that these friends are also interested in the same topics. Moreover, participating in an event such as a music concert or a football match leads to even more encounters with nodes with common interests.

Thus, in this article we begin by proposing a basic interest-based dissemination algorithm, showing that it reduces congestion without affecting hit rate, when compared to flooding and contact history-based methods. Afterward, we present *ONSIDE* (*O*pportu*N*istic *S*ocially-aware and *I*nterest-based *D*iss*E*mination), a dissemination strategy that leverages information about a node's social connections, interests and contact history, in order to decrease network overhead and congestion, while not affecting the network's hit rate and delivery latency. This is done by carefully selecting the nodes that act as forwarders, instead of simply flooding every node. Finally, we show that the order in which messages are dropped, when a node's buffer is full and it needs to download a new message, has an effect on various ON-specific metrics. We propose five heuristics for sorting the messages in a node's memory, and show how each of them affects the output. All our solutions are compared with various existing algorithms, using three ON mobility traces.

2. Related work

Since data dissemination presumes sending a message to multiple nodes, Epidemic [5] is one of the simplest strategies previously proposed. The algorithm simply floods the network with a message, until it reaches all interested subscribers. When two Epidemic nodes meet, they exchange all the messages they carry. This way, assuming that a node can store an unlimited number of messages in its data memory, the maximum hit rate of the network is guaranteed. However, flooding the network with messages can very easily lead to congestion and high overhead, especially in dense networks with high activity.

Although users' interests have not been employed by too many dissemination solutions in ONs, routing algorithms that take advantage of this information have been proposed over the years. Moghadam and Schulzrinne's interest-aware algorithm [2] is able to analyze a user's history of cached data in order to obtain his interests. Using these interests, the algorithm is able to decide whether a document carried by an encountered node should be downloaded or not when a contact occurs. When a node *A* meets a node *B*, it has to decide which documents should be forwarded to *B*. This is done by mapping each document carried by *A* into *B*'s interest space and applying cosine similarity. If the result is higher than a predefined threshold, then that document is transferred to *B*. Thus, documents are only spread to nodes that are interested in their content, which in turn can forward them further on to nodes with similar interests. By using this algorithm, the number of data exchanges in the network decreases dramatically, since a node can only carry a document that it is interested in. Another algorithm for data forwarding in ONs which is based on the assumption that nodes with common interests tend to meet each other more often than regular nodes, is SANE [3]. Similarly to the previous algorithm, nodes with SANE have their interests defined by profiles represented as m-dimensional vectors in an interest space. When two nodes meet, they each compute cosine similarities between their own interest profiles and the relevance profile of each message in the other node's memory. If the cosine similarity for a message is higher than a download threshold, then that message is downloaded. SANE is slightly different than the previous algorithm because it uses two thresholds: one for deciding whether a node is interested in a message, and one for deciding whether a node should download a message. The first threshold is higher than the second one, and is the equivalent of the threshold from the previous algorithm, where a node would only download a document if it was interested in its content. Thus, SANE might lead to more message exchanges in the network, but also to higher hit rates and lower delivery latencies. However, the disadvantage of these two solutions is that they are very restrictive in terms of message exchanges, since data is forwarded only to interested nodes. Thus, if a node never encounters another node that is interested in a data item it stores, then that item might not reach any interested node. This is why, as we show in Section 3, ONSIDE leverages other nodes that are not necessarily interested in a data item, but have a high chance of encountering other nodes that are.

Algorithms that combine interest information with other context knowledge (such as social connections) have also been proposed [6,7]. One example is ML-SOR [8], which exploits three social network layers: the contact network, the online social network and the interest network. The contact network is the proximity graph created through contacts between devices, while the online social network is extracted from virtual contacts. A tuple of multiple such social network layers is defined as a multi-layer social network. ML-SOR thus extracts social network information from multiple contexts, and analyzes encountered nodes in terms of node centrality, tie strength and link prediction on different social network layers. When an ML-SOR node *A* encounters a node *B*, it computes a social metric called *MLS* for all the messages in *B*'s memory, both from its own standpoint, as well as from *B*'s. If *MLS* is higher for node *A*, then it sends a download request for that particular message. The social metric is computed based on three components: *CS*, *TSS* and *LPS*. *CS* represents the