Context-aware File Sharing for Opportunistic Networks

Marco Conti, Franca Delmastro, and Andrea Passarella
IIT Institute - CNR
via G. Moruzzi,1 - 56124 Pisa, Italy
email{firstname.lastname@iit.cnr.it}

***Abstract*—Opportunistic networks are mainly characterized by nodes intermittently connected among them. Available applications designed for mobile ad hoc networks are not suitable for such an environment since we cannot assume to have a stable path between pairs of nodes. Network protocols and applications themselves must be enhanced to exploit all possible communication opportunities to deliver messages on the network. In this demo we present an enhanced file sharing application based on the exchange of context information between nodes. In this case the context is defined as a combination of the user personal information, interests, and social relationships in order to implement cooperative downloading mechanisms. Besides reducing the impact of intermittent connectivity and high mobility on multi-hop communications, exploiting context also allows us to avoid flooding, thus resulting in a very efficient approach.**

I. Introduction

Opportunistic networks represent one of the current evolutions of ad hoc networks. No assumption is made on the connectivity of couples of nodes wishing to communicate. In this scenario intermittent connectivity is very likely to occur and nodes must exploit all the available opportunities to get in touch and exchange information. Since it is not possible to inherit the main principles of MANET protocols, new solutions have to be designed at all layers of the stack to develop efficient and disconnection-tolerant services for opportunistic environments. To this aim, the use of context information, in terms of users' habits and their social relationships, could be a great improvement both to discover all the available communication opportunities and select the best ones to reach a specific destination, and to optimize applications' features. Within the Haggle Project a context-based forwarding protocol (HiBOp [1]) has been designed and evaluated through simulations considering also the impact of users' mobility on its performance [2]. Simulations results show that using context information highly reduces resource consumption (e.g. memory, bandwidth, and consequently energy), slightly increasing message delays compared with dissemination based solutions like Epidemic Routing [3]. Starting from these results we decided to exploit additional context information to enhance the main features of a real attractive application for mobile users: File Sharing (FS).

FS belongs to the set of content sharing applications that are extremely appreciated both in wired and wireless networks. Nowadays this paradigm is evolved into a new model called User-Generated Content in which users are not only passive content consumers, but also active content producers (YouTube and Flickr are examples of this kind of services on Internet). Mobile devices further encourage this paradigm, since users are able to generate and share content anytime and anywhere. Thus, using FS as an example, we can show how using context information enhances the application's features guaranteeing an efficient service for opportunistic networks. Specifically, using the interests of the local user and its neighbors and the related list of shared files can allow the prediction of possible users' requests in the next future, optimizing network communications. In this demo we present a prototype of a context-aware opportunistic FS application showing the advantages of using context to reduce messages dissemination and guarantee the service reliability in case of intermittent connectivity and high mobility of nodes.

II. Context-aware Opportunistic File Sharing

Users essentially need to communicate and exchange data with other users with the same interests. For this reason, FS applications represent one of the most used services both in wired and wireless networks. In the last few years a lot of studies on p2p systems on MANETs and related services have been conducted, providing optimized solutions for mobile users [4]. These solutions are essentially based on the use of routing protocols that allow the message delivery only in case a stable path between the source and the destination node exists at the moment of the initial sending. Obviously, nodes mobility affects the system performance in terms of packet loss and delays due to network topology reconfigurations and the establishment of new routes. Thus, these solutions are not suitable for opportunistic networks.

However, in an opportunistic scenario nodes mobility can be exploited to enable communications between nodes currently not in range. Very simple approaches adopt scoped flooding techniques (e.g. Epidemic routing), but they easy exhaust network and devices' resources. Thus, more resource efficient solutions are required. To this aim, considering users' behaviour models is necessary for the system design. Specifically, users interactions and their mobility models can be exploited as context information to improve the application's efficiency, optimizing local resource consumption and reducing the proactive dissemination of requests between nodes.

A. Context-awareness in opportunistic networks

Our definition and exploitation of context information is based on social network models [5] and small-world theories [6]. In typical social network models users are grouped in communities, and nodes of the same community have strong social links between each other. Some nodes also have social links outside their ”home” community, modelling social relationships with users of different groups. Small-world theories have shown that these ”external” links act as shortcuts and enable communications across the network with a small number of hops (6 hops in the ”classical” small-world models).

This model of users behaviour can be exploited to spread context information in the users community in order to optimize opportunistic network protocols and applications. For example, data can be replicated only on nodes whose context tells that they are in touch with users interested in that type of data. We have exploited these ideas to design HiBOp context-aware forwarding protocol for opportunistic networks [1].

Following the main principles of HiBOp, context information can be divided into two main categories: the *current context* of the local user, and the *context evolution* over time. The current context contains information about the user itself, and it can be customized depending on the application. Each user is identified by a set of personal information (e.g. name, address, city, job, hobbies) that can be enriched with additional information like preferred file categories, genres, and topics, but also the file list he/she wants to share in this particular case. Each node maintains this information in a local data structure called Identity Table (IT). By exchanging ITs during the neighbor discovery procedure, each node knows the interests of all its neighbors and the related file lists, and it can add them to its current context. However, since ITs can contain a huge quantity of data, it is not efficient to periodically broadcast them on the network. Thus, each node periodically announces its presence in the network broadcasting a unique identifier that can be represented by the hash of its IT. Then, when it receives an unknown identifier from one or more neighbors, it broadcasts its IT.

The current context gives thus only an istantaneous picture of the network and the social interactions of the local user. However, since these conditions can change over time, it is necessary to maintain also information about past experiences and encounters of the user as the evolution over time of the general context. For example, in HiBOp framework context information seen in the past is stored in a specific data structure together with associated indexes like the frequency of spotting data and its redundancy.

B. Context-aware File Sharing application

Starting from this definition of context, we can thus analyse how this information is exploited by the opportunistic FS application to improve its basic features. Let us consider two social communities of users as shown in Figures 1 and 2. Generally, users belonging to the same community share some common interests, and a single user can belong also to more than one community. In addition, users can also have social

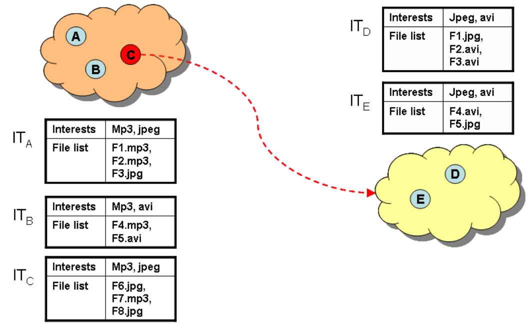

Fig. 1. Context-aware opportunistic FS: discovery of the current context.

relationships outside their main community and occasionally ”visit” other communities. Exploiting this users behavioural model, we present a simple example to better understand the service features. Looking at Figure 1 we can note that in the two communities there are users sharing some common interests: on the one hand, users A and C are interested in collecting mp3 and jpeg files, while B is interested in mp3 and avi files; on the other hand both users D and E are interested in mp3 and avi. Assuming that all nodes in each community are able to comunicate to each other, after the neighbor discovery phase each of them knows the interests and the file list of the others, updating thus their current context. In addition, supposing that user C originally belongs to both communities, it is highly probable that he/she moves towards the other community and can thus share additional files. When he/she reaches the other community, it broadcasts its IT and collects information from the new neighbors. At this point he/she is able to select the files he/she is interested in and download them on its device (see Figure 2). However, looking at the previously stored context information, the application running on node C is also able to identify additional files in which the users of the other community are probably interested. The selection of these additional files mainly depends on two factors: local resource availability (e.g. memory and battery capacity) and the *utility* of those data both for the local user and the social contexts he/she is in touch with. Through a probabilistic evaluation of these parameters, the application is able to define a cooperative downloading scheme in which single nodes, in addition to downloading data they are interested in, can also prefetch data probably interesting for other members of different communities. In this way, the application tries to predict future requests from the other users. If the subject of the request is available on the local node it can directly serve it, otherwise it has to forward the request to the best destination it currently knows. Specifically, in case the prediction fails, the receiving node is able to detect if some of the known nodes is currently sharing the requested file, directly looking at the file lists stored in its context data structures, and the probability of successfully download the file from them.

As shown in Figure 2, when user C reaches the new community, it becomes aware of the files shared by the new neighbors. Since he/she is directly interested in mp3 and jpeg files, he/she decides to download first of all F1.jpg from D

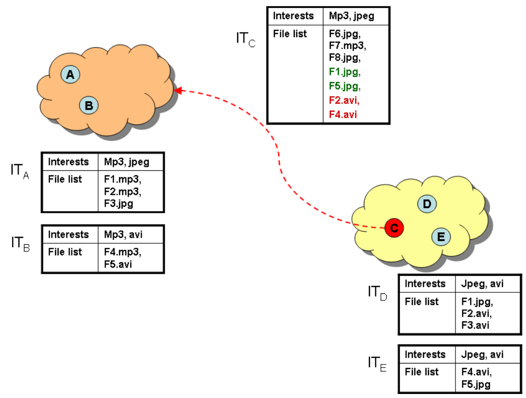

Fig. 2. Context-aware Opportunistic FS: cooperative downloading in a new context.

and F5.jpg from E. Then, he/she decides to dedicate part of its remaining memory to download avi files that can be interesting for the users belonging to the other community (in this case F2.avi and F4.avi). In this way, when C comes back to his/her original community is able to serve requests for those specific files, avoiding the requests dissemination over the network and reducing the delay to satisfy them.

As previously explained, the possible choices between the files a user can decide to download on its local device mainly depends on the resource availability and the data utility. Intuitively, the main idea is to select those files that maximize the utility in terms of utility for the local user (e.g. data belonging to the same category he/she is interested in) and utility for the users belonging to the different contexts the user is in touch with. We can thus define the following utility function:

$$U = u^{(l)} + \sum_i \omega_i u_i^{(c)} \quad (1)$$

where $u^{(l)}$ is the utility for the local user, $u_i^{(c)}$ is the utility for the i-th context the user is in contact with, and ω_i is a cooperation index (a real number between 0 and 1), which defines the willingness of the user to cooperate with the i-th context (i.e., to spend own resources to increase data availability for that context). Note that, by using cooperation indexes greater than 0, we can avoid the selfish users behaviour in which they tend to maximize their own utility. In addition, the local user is aware of the available files in its current context and can directly request to its neighbors those he/she is currently interested in, avoiding unnecessary preventive downloads. Another condition for the utility computation is represented by the current context of the local user. Since it is highly probable that users currently visiting the same community are in communication range, or can directly download the files they are interested in with few-hops connections, we can decide to assign a lighter weight to the current context of the local user, so that he/she can privilege the other contexts he/she is in touch with.

After this preliminary analysis it is necessary to define the parameters the utility function depends on. Generally, we consider the *access probability* as the probability the local node has to successfully download the specific file, that is strictly related to the *context stability* as a function of the time the local node would spend in the current context and the mobility model of its neighbors. Furthermore, we must consider the advantage the users of the examined context can take from the cooperative downloading of the file by the local user. In addition to this utility function, we must consider the local resources availability that can be evaluated for each possible download in the current context. A detailed probabilistic analysis of this *context-aware utility* is currently a work in progress. In this demo we show some simple examples of the application in which a simplified elaboration of the utility function is considered, in order to highlight the main advantages of using context information to enhance basic features of a file sharing service in opportunistic networks.

III. Demo Highlights

During this demo attendees will be able to run the FS application on laptops, downloading files from neighbors belonging to the same community and moving from a community to another. This demo will show how such kind of application is enjoyable and attractive for users even in an opportunistic environment. From the technical point of view, the demo will show the reliability of the application in case of intermittent connectivity, storing requests that cannot be istantaneously satisfied due to the disconnection of the destination node or the movement of the local user from a community to another, always exploiting context information. In addition, it will show the reduction of requests dissemination on the network thanks to the elaboration of context.

IV. Demo Requirements

We will provide all the hardware necessary for the demo. Specifically, we will provide a set of laptops for attendees to enjoy the demo, and to illustrate the advantages of this application. The demo will setup a standard 802.11 ad hoc network characterized by two groups of nodes not directly connected. We will establish possible 1-hop communications by using iptables firewall, and we will configure the ad hoc network through proper channel selection, use of dedicated ESSID and encryption keys. This will allow our demo not to interfere with any other demo running on 802.11. We do not foresee to need any particular support by the organization, apart from a stand where to place a poster and power sockets to plug the laptops in.

References

[1] C. Boldrini, M. Conti, I. Iacopini, and A. Passarella, "HiBOp: a History-based Routing Protocol for Opportunistic Networks," in *Proc. of IEEE WoWMoM 2007*, Helsinki, Finland.

[2] ——, "Impact of Social Mobility on Routing Protocols for Opportunistic Networks," in *Proc. of IEEE AOC Workshop in conjunction with IEEE WoWMoM 2007*, Helsinki, Finland.

[3] A. Vahdat and D. Becker, "Epidemic Routing for partially connected ad hoc networks," in *Tech. Rep. CS-2000-06*, CS Dept., Duke University, 2000.

[4] M. Conti, F. Delmastro, and G. Turi, *Peer-to-Peer Computing in Mobile Ad Hoc Networks*, P. Bellavista and A. Corradi, Eds. The Handbook of Mobile Middleware, Auerbach Publications, 2007.

[5] D. Watts, *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press, Princeton, 1999.

[6] J. Kleinberg, "The Small Worlds Phenomenon: An Algorithmic Perspective," in *Proc. of 32nd ACM Symposium on Theory of Computing*, 2000.