A Framework for Opportunistic Networking in Spontaneous and Ephemeral Social Networks

Nicolas Le Sommer, Pascale Launay, Yves Mahéo

A Framework for Opportunistic Networking in Spontaneous and Ephemeral Social Networks

Nicolas Le Sommer
IRISA, Université de Bretagne-Sud
nicolas.le-sommer@univ-ubs.fr

Pascale Launay
IRISA, Université de Bretagne-Sud
pascale.launay@univ-ubs.fr

Yves Mahéo
IRISA, Université de Bretagne-Sud
yves.maheo@univ-ubs.fr

ABSTRACT

The C3PO project promotes the development of a new kind of social networks called Spontaneous and Ephemeral Social Networks (SESNs), dedicated to cultural or sports events. SESNs rely on opportunistic networks formed dynamically by the mobile devices of event attendees. This paper presents the framework developed in the C3PO project to support opportunistic communication in such challenged networks. This framework exploits the different wireless interfaces of the mobile devices (e.g., Bluetooth, Wi-Fi and Wi-Fi Direct) to interconnect them so as to form an opportunistic network.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Store and forward networks*

Keywords

Mobile Social Networking, Opportunistic Computing.

1. INTRODUCTION

A new era of communication is now underway with the massive use of online social networks and media. A multitude of more or less specialized software platforms allow people to interact and share information via the Internet. Although users are faced with a diversity of purposes and types of platforms (business-oriented, dedicated to music, books or sports, designed to share photos, videos or news...), the vast majority of these social networks are build on the same architectural model: a centralized service provider acts as a broker to filter and disseminate the data produced by the users. This architecture is not without drawbacks from the user's point of view. First, it induces a dependence on a centralized authority that controls the data management and exchange (more than often, data ownership is even transferred to the service provider). Second, this architecture requires that the user has a permanent access to the Internet.

Project C3PO[1] (Collaborative Creation of Contents and Publishing using Opportunistic networks) proposes to investigate multimedia content production and exchange in a new type of social networks that we call Spontaneous and Ephemeral Social Networks (SESNs). SESNs rely on a peer-to-peer distributed architecture formed spontaneously by mobile devices carried by people and, optionally, by fixed devices that can be deployed to support such networks. Devices are interconnected using their wireless interfaces (e.g., Wi-Fi or Bluetooth). Opportunistic networking techniques are devised in order to support the connectivity disruptions resulting from the mobility of users and the short communication range of the radio interfaces. By exploiting the store-carry-and forward principle, these techniques allow devices to exchange data even if they are out of the radio range of each others and no end-to-end path exist between them. Due to their spontaneous and ephemeral nature, SESNs are suited to produce multimedia reports on conferences and cultural or sport events, such as a marathon [4].

In this paper, we present the framework designed in project C3PO to support opportunistic networking in SESNs. This framework allows the building of a communication middleware that offers a message-oriented API to application programmers so that they can develop C3PO applications adapted to specific needs (for example an application may be dedicated to the emission, reception and presentation of the official results of a race). As a communication framework, the C3PO framework is intended to be specialized to form a communication middleware. This is done by implementing a small number of functions that constitutes a set of reactions to standard communication events. The programmer either develops original implementations of these functions, or uses the implementations provided in the C3PO toolkit.

The opportunistic network targeted by C3PO is a form of wide-area ad hoc network that exploits common device to device radio technologies available in smartphones and tablets. At present, it seems that there exists no unique technology that can be effectively used for this purpose. Every technology has its own limitations, for example the limited number of devices involved in a Wi-Fi Direct group, the short radio range of Bluetooth interfaces, or the necessity to acquire root privileges for setting a Wi-Fi ad hoc network. We address this difficulty by integrating in C3PO the possibility to exploit several technologies (Wi-Fi in AP mode, Wi-Fi Direct, Bluetooth...) to build so called micronets in which direct communication is possible, and aggregate these micronets in larger networks, called macronets. As the topology of the network may dynamically evolve, store-carry-and-forward communication is used inside and between macronets.

Although numerous works have been done in the last decade on opportunistic networking, with a strong focus on protocols and

---

[1] http://www.c3po-anr.fr/

routing mechanisms, few works share the two main characteristics of C3PO: first, proposing a middleware platform designed to be effectively used to build opportunistic applications with a clearly defined API, and second, managing to exploit real mobile ad hoc networks. Among them can be cited CAMEO [1], Haggle [8], DoD-WAN [5], 7DS [6], or the system described in [3].

2. NETWORK ARCHITECTURE

A SESN is formed by users carrying off-the-shelf mobile devices (smartphones, tablets) communicating with each others thanks to short range wireless interfaces implementing communication standard such as Bluetooth or IEEE 802.11 (Wi-Fi). In most cases, the users that compose the SESN can move, and are not necessarily located in the same place at the same time. The topology of the network formed by their devices changes continuously and can be fragmented into distinct communication islands. Thus, the network is structured as a collection of what we call *micronets*, grouped together in independent *macronets*.

2.1 Micronets

We define a micronet as a subset of devices connected using a common communication technology, that are able to communicate directly with each others. A micronet is an abstraction of a piconet for Bluetooth, a basic service set (BSS) for Wi-Fi legacy, or a group for Wi-Fi direct.

The topology of a micronet does not matter, but, most commonly, a micronet is structured in a star topology, where devices communicate with each others through a central node called, for example, master for Bluetooth, group owner (GO) for Wi-Fi Direct or access point (AP) for Wi-Fi legacy.

We consider that devices in a micronet communicate *directly* as there exists a network layer that allows any device to address and send messages to any other device in its micronet, even if it is not the case at the MAC layer (messages being forwarded through a Bluetooth master, Wi-Fi Direct GO or Wi-Fi AP). We call a *one-hop neighbor* of a device any other device member of the same micronet, and a *micronet link* the (virtual) link between two one-hop neighbors.

Wireless communication technologies available for current off-the-shelf mobile devices allow only a limited number of devices to be interconnected and exchange data directly between them (e.g., at most 8 devices for a Bluetooth piconet). Consequently, micronets have to be interconnected in order to allow SESN users to communicate over the whole network.

2.2 Macronets

We define a macronet as a group of micronets interconnected through devices that are members of at least two micronets. A macronet can thus link together any number of devices provided they are close enough to each others and can form connected micronets. The resulting communication network is a collection of independent macronets.

Macronets bear some similarity to Wi-Fi Direct multi-groups as defined in [2] or to Bluetooth scatternets, composed of Bluetooth-enabled interconnected devices where some devices participate in two different piconets. A Bluetooth scatternet can thus link together more than 8 devices. However, the notion of macronet is wider than that of scatternet, as the micronets forming a macronet can use different communication technologies. For example, a macronet can be formed when a Bluetooth piconet and a Wi-Fi Direct group are linked together by a common device member of both micronets through its Bluetooth and Wi-Fi interfaces.

A device joining two micronets is called a *gateway* as it connects both micronets and is used for relaying data exchanged between them. We call a *two-hop neighbor* of a device any device member of another micronet having a common gateway with its own micronet.

Given the strong limit in the number of connections allowed in Wi-Fi-Direct, Wi-Fi soft-AP or Bluetooth (due to the specifications or noted in available implementations, particularly in Android), avoiding micronet link redundancy is a crucial point. In our framework implementation, we first ensure that there cannot exist redundant micronet links, for example one through the Bluetooth interfaces and one through a Wi-Fi Direct link. Moreover, we prevent a device from establishing a micronet link with a two-hop neighbor. This guarantees that two micronets in a macronet cannot be linked by more than one gateway device.

2.3 Inter and Intra-macronet Communication

It cannot be assumed that there is a common addressing scheme in a macronet nor that messages can be transmitted directly between the devices of a macronet, even in the case of a Bluetooth scatternet. So, an upper networking layer is needed to address devices and transfer messages between different micronets members of the same macronet. A possible approach would be to design a dynamic routing algorithm associated with an addressing scheme with a macronet scope, so that any message sent could be immediately routed to any other macronet node, and that some routing structure be built to broadcast messages in the macronet. We did not choose to follow this approach in the C3PO framework, but rather to rely on the store-carry-and-forward principle for both intra-macronet and inter-macronet communication. Indeed, temporaneous end-to-end paths in the whole SESN (i.e., between macronets) cannot be established; therefore, a store-carry-and-forward layer must be developed for communication between macronets, and this layer can also be used for exchanging messages inside a macronet. Besides the simplification of the framework, this option is likely to provide a better tolerance against network disruptions.

3. C3PO FRAMEWORK

3.1 Overview of the Architecture

The C3PO framework is a Java-based framework dedicated to opportunistic networking. It has been designed to be highly configurable and extensible. It provides application developers with an API offering two distinct application-level communication paradigms, namely a topic-based publish/subscribe paradigm, and a channel-based send/receive paradigm. Its architecture, illustrated in Figure 1, is organized in five main modules.

Module 1 and 2 are related to the management of micronets and macronets. Module 1 manages wireless interfaces and sets up communications in micronets. It defines abstract neighbor discovery managers and network managers, that are respectively responsible for discovering neighbor devices (i.e., devices in radio range) to build micronets and for managing communications to support data exchanges between devices inside a micronet. Module 2 defines opportunistic networking mechanisms to perform data forwarding inside and between macronets. It includes a cache of messages, a message forwarder and abstract forwarding strategies in order to implement the store-carry-and-forward principle. This decoupling is motivated by the fact that the network topology built in order to support networking for SESNs allows direct communications inside micronets, but does not provide any inter-micronet addressing and networking layer, either inside or between macronets.
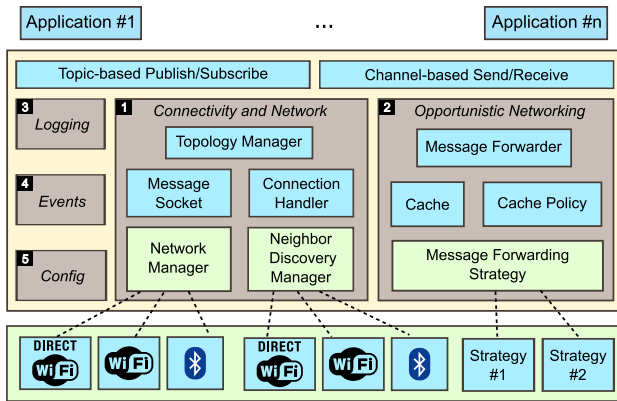
Figure 1: General architecture of the C3PO framework

Figure 2: Hierarchy of message forwarding strategies.

Finally, the three other modules (i.e., modules 3, 4 and 5) composing the framework define functionalities that are respectively dedicated to the configuration of the framework, to the management of the events produced by the framework, and to log the traces that are generated by the framework (e.g., contact and message exchanges traces). Modules 1 and 2 are original parts of the framework, and therefore they are detailed in the remainder of this section.

It is worth mentioning that several concrete implementations of the abstract functionalities defined in the C3PO framework are provided in a toolkit. These implementations can be used as they are, or can be extended by developers in order to adapt them to their own needs. For instance, this toolkit provides, for the Bluetooth, Wi-Fi Legacy and Wi-Fi Direct communication modes, specific network managers and neighbor discovery managers. Other technologies (such as ZigBee for example) could be included by providing specific implementations of these abstract managers. Similarly, the C3PO framework does not implement a given message forwarding strategy, but instead defines a Java interface that must be implemented by concrete classes. Some classes are already provided in the C3PO toolkit. Developers can use these strategies directly, can extend them for their specific purposes, or can define new ones from scratch.

3.2 Application Programming Interface

Two distinct application-level communication paradigms are provided by the framework: a topic-based publish/subscribe model, and a channel-based send/receive model.

The topic-based publish/subscribe communication model makes it possible to develop applications that can publish multimedia contents on specific topics, and subscribe in order to receive the contents related to given topics. The model implemented in the C3PO framework relies on a purely peer-to-peer decentralized approach. Topic names are assumed to be already known by the application as in C3PO, a given topic is generally pre-defined or handled by a single application module that performs both production and consumption. The subscription and the publication are local to each device. Thanks to the store-carry-and-forward principle, contents published in a topic by publishers are disseminated opportunistically in the communication network by mobile devices, being either devices hosting subscribers for this topic or ordinary intermediate devices , and are thus delivered to the topic subscribers.

The point-to-point communication paradigm using the concept of channel is intended for applications that allow users to communicat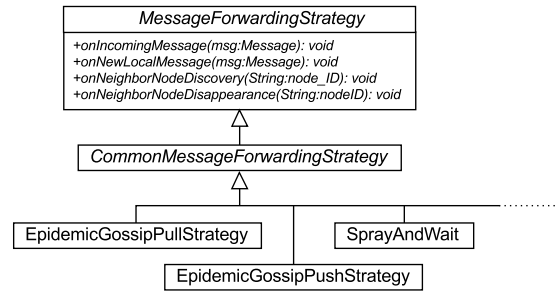e with each others by sending messages addressed to specific recipients. In the framework, a channel between two devices is identified by the addresses of the devices and a channel ID. Messages sent through a channel are opportunistically forwarded by intermediate devices towards their destination according to one of the message forwarding strategies implemented in the framework. Primitives similar to the ones used for Java sockets are available to the programmer (open, accept, receive...). Asynchronous receiving is also possible by registering a message listener associated with a channel.

3.3 Opportunistic Networking

The C3PO framework implements the store-carry-and-forward principle. Different types of messages are currently considered: application-level messages and several types of control messages such as beacon messages that are used to exchange the neighbor lists, gossiping messages, acknowledgements, and drop messages that make it possible to implement network healing protocols in order to reduce the number of messages that are disseminated in the network. The messages that are published or sent by the applications using the previous described API are exchanged by mobile devices according to a given forwarding strategy. Strategies are reactive and based on four main events: the emission of a message by a local application ; the reception of a message from the network ; the discovery and connection of a new neighbor device ; the disappearance and disconnection of a neighbor device.

Message forwarding strategies must implement the Java interface *MessageForwardingStrategy* defined in Figure 2. They can be defined by implementing this interface, or by extending an existing strategy, such as the *CommonMessageForwardingStrategy*. For instance, the common message forwarding strategy implements functionalities to store messages in the local cache, to decrease the number of hops in messages that are received, as well as to exchange neighbor lists between connected devices to build a two-hop neighborhood. By extending this common strategy, developers can reuse these functionalities directly, and thus they do not need to redefine them in their own strategies. So the implementation of new strategies is greatly simplified. These new strategies could implement forwarding techniques relying on content-based, topic-based, community-based, contact frequency or social-based criteria [7]. Strategies relying on the computation of spanning trees could also be implemented in the C3PO toolkit in order to perform an efficient broadcast of data within a macronet. In the current toolkit, only controlled epidemic forwarding strategies are provided. They have the advantage to be simple to implement and to be robust to disconnections, and meet the needs of the C3PO project, in which the digital contents produced in the social networks must be replicated on the mobile devices in order to ensure their local access by users.

EpidemicGossipPullStrategy
extends: CommonForwardingStrategy

onIncomingMessage(msg):
    parent.onIncomingMessage(msg) *// store* DATA *message in the local cache*
    switch (msg.type)
        case MESSAGE_ID_CATALOG:
            CATALOGS.update(msg.source, msg.catalog)
            missing_msg = diff(LOCAL_CATALOG, msg.catalog)
            forward(msg.source, missing_msg, MESSAGE_PULL)
        endcase
        case MESSAGE_PULL:
            forward(msg.source,CACHE.get(msg.id_pull_message), DATA)
        endcase
        case DATA:
            LOCAL_CATALOG.update(msg.ID)
            CATALOGS.get(msg.source).update(msg.ID)
            forward('*', LOCAL_CATALOG, MESSAGE_ID_CATALOG)
        endcase
    endswitch

onNewLocalMessage(msg):
    parent.onNewLocalMessage(msg) *// store* DATA *message in the local cache*
    LOCAL_CATALOG.update(msg.ID)
    forward('*', msg, DATA)

onNeighborDiscovery(nodeID):
    parent.onNeighborDiscovery(nodeID) *// exchange neighbor list*
    forward(nodeID, LOCAL_CATALOG, MESSAGE_ID_CATALOG)

onNeighborDisappearance(nodeID):
    parent.onNeighborDisappearance(nodeID)
    CATALOGS.remove(nodeID)

Algorithm 1: Example of a gossiping-based message forwarding strategy provided by the C3PO toolkit.

One of them is a gossiping-based message forwarding strategy that allows to replicate the messages on all the nodes forming the opportunistic network, provided these nodes have enough space in their local cache to store these messages. Compared to a plain flooding (that blindly transmits all the messages to neighbors), it reduces drastically the number of messages, by maintaining on each device a view of the content of the caches of its neighbors, through the exchange of catalogs of message IDs.

The development of the strategy consists in the implementation of the four event reactions described in Algorithm 1. As it can be seen, the code remains quite simple, a significant part of it being inherited from the *CommonForwardingStrategy* parent class.

3.4 Network Topology Management

The first module (identified by number 1 in Figure 1) manages the connectivity between devices and the communications inside micronets. It is composed of a set of Network Managers and Neighbor Discovery Managers, one for each short range communication interface used by the device. We currently consider Bluetooth, Wi-Fi (in access point mode) and Wi-Fi Direct interfaces, for which concrete implementations are provided in our toolkit. Neighbor Discovery Managers implement a proactive discovery of mobile devices in the interface radio range and produce neighbor discovery or disappearance events managed by the module 4 of the framework. Network Managers catch those events and connect together devices that have been discovered, relying on the Topology Manager that prevents it from creating one-hop and two-hop redundant communication links between devices. Network Managers produce connection and disconnection events. Two devices, when they are connected together, exchange their neighbors list, thus allowing to implement a two-hop discovery. When the topology changes, the nodes broadcast these changes to the nodes they are connected to.

Communications between devices are performed by Connection Handlers through Message Sockets. Instances of Connection Handler and Message Socket classes are created for each communication interface, and thus provide a uniform way to manage the communications between the devices independently of the underlying technology. The nodes are identified by Message Sockets using a unique identifier as a string of 6 bytes encoded in Base 62.

In order to support new kinds of connectivity types such as for instance ZigBee, developers can add specialized Network Manager and Neighbor Discovery Manager classes. The rest of this module remains unchanged.

4. CONCLUSION

In this paper, we have introduced a new type of social networks dedicated to happenings, namely spontaneous and ephemeral social networks (SESNs). A SESN has the characteristic of relying on the opportunistic network formed by the devices of the happening attendees. We have shown how such an opportunistic network can be formed using the Bluetooth, Wi-Fi and Wi-Fi Direct interfaces of standard mobile devices, assembled in a novel network architecture. We have also presented the framework we have developed in project C3PO to support opportunistic communication in SESNs. This framework is able to manage the different wireless interfaces of mobile devices so as to form an opportunistic network dynamically. It is designed to be highly configurable and extensible, namely making it possible to define various message forwarding strategies.

5. REFERENCES

[1] V. Arnaboldi, M. Conti, and F. Delmastro. CAMEO: A Novel Context-Aware Middleware for Opportunistic Mobile Social Networks. *Pervasive and Mobile Computing*, 2013.

[2] C. E. Casetti, C.-F. Chiasserini, L. Curto Pelle, C. Del Valle, Y. Duan, and P. Giaccone. Content-centric Routing in Wi-Fi Direct Multi-group Networks. In *16th International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, June 2015.

[3] O. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, L. Pajevic, and G. Karlsson. A Mobile Peer-to-Peer System for Opportunistic Content-Centric Networking. In *2nd ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds*, pages 21–26, New Delhi, India, Aug. 2010. ACM.

[4] F. Laforest *et al*. C3PO: a Spontaneous and Ephemeral Social Networking Framework for a Collaborative Creation and Publishing of Multimedia Contents. In *International Conference on Selected Topics in Mobile and Wireless Networking*, pages 214–219, Rome, Italy, Sept. 2014. Elsevier.

[5] Y. Mahéo, N. Le Sommer, P. Launay, F. Guidec, and M. Dragone. Beyond Opportunistic Networking Protocols: a Disruption-Tolerant Application Suite for Disconnected MANETs. In *4th Extreme Conference on Communication*, pages 1–6, Zürich, Switzerland, Mar. 2012. ACM.

[6] A. Moghadam, S. Srinivasan, and H. Schulzrinne. 7DS - A Modular Platform to Develop Mobile Disruption-Tolerant Applications. In *2nd International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE, Sept. 2008.

[7] V. F. S. Mota, F. D. Cunha, D. F. Macedo, J. M. S. Nogueira, and A. A. F. Loureiro. Protocols, Mobility Models and Tools in Opportunistic Networks: A Survey. *Computer Communications*, 48:5–19, July 2014.

[8] J. Su, J. Scott, J. Crowcroft, E. de Lara, C. Diot, A. Goel, M. How Lim, and E. Upton. Haggle: Seamless Networking for Mobile Applications. In *9th International Conference on Ubiquitous Computing*, volume 4717 of *LNCS*, pages 391–408, Innsbruck, Austria, Oct. 2007. Springer.