Campus++: a publish-subscribe architecture for intermittently connected 802.15.4 networks

Donato Battaglino *, Lorenzo Bracciale*, Andrea Detti *, Francesca Lo Piccolo *,
Andrea Bragagnini†, Maura Santina Turolla† Nicola Blefari-Melazzi *
*DIE, Universitá di Roma “Tor Vergata”, Rome, Italy
{donato.battaglino,lorenzo.bracciale,andrea.detti,francesca.lopiccolo,blefari}@uniroma2.it
†Telecom Italia, Turin, Italy
{andrea.bragagnini,maurasantina.turolla@telecomitalia.it}@telecomitalia.it

***Abstract*—The aim of this extended abstract is to present Campus++, a location-based publish-subscribe system for intermittently connected delay tolerant networks, exploiting IEEE 802.15.4 devices, and taking into due account the severe constraints deriving from their physical characteristics. We describe our proposed architectural model and how we implemented our solution in a real test-bed.**

We point out that our system can be easily adapted to operate in a fully distributed, infrastructure-less way, allowing free communications e.g. in disaster areas or in areas in which ”usual” communications means are either non existent or intentionally made unavailable.

I. Introduction

The small size and power consumption of IEEE 802.15.4 devices allows embedding them in GSM/UMTS U-SIM cards and/or SD cards. The availability of such technology for data exchange within mobile phones is very useful to complement GSM/UMTS services, providing location-based or proximity services, such as chat and advertisements in a commercial center, configuration data, micro-payments, access control. In addition, once that we have the availability of a free communication radio link, we can enlarge the assortment of offered services, supporting not only direct data exchanges between two users within the 802.15.4 connectivity range, but also communication among intermittently connected users.

This is a typical scenario of so-called Delay Tolerant Networks (DTN) [4] (we do not think to a mobile ad-hoc network scenario, as the small range of IEEE 802.15.4 devices makes difficult providing end-to-end paths among any two nodes).

We argue that a communication paradigm well suited to this service environment is publish-subscribe [8]. Besides, this paradigm can satisfies the requirements of a community of users such as the one of a university campus, complementing other services such as voice and Internet access.

In this paper we present a location-based publish-subscribe system, named CAMPUS++, that enables a community of users equipped with IEEE 802.15.4 devices (e.g., mini SD, ZSIMs [1]) to exchange messages regarding specific “Topics“. We call these messages data-samples; users that send data-samples regarding a topic are publishers of that topic; users interested in receiving data-samples of a topic are subscribers of that topic. Users exploit services by running Campus++ software on their PDAs or mobile phones. The PDA application allows visualizing the list of topics in the system and the subscribed-to topics and published data-samples. The delivery of data-samples toward subscribers is carried out by a Delay Tolerant Network (DTN) formed by 802.15.4 devices belonging to users, and by infrastructure nodes. The spread of information only occurs within a limited service area, hence CAMPUS++ allows implementing several location-based services, without necessarily resort to external localization infrastructures such as GPS, which, in addition, can not be used indoor and could pose some privacy issues.

II. System architecture and protocol design

The Campus++ system architecture is composed of 802.15.4 mobile user devices, one or more 802.15.4 infrastructure nodes called ”way-servers”, and one administrative server (Figure 1).

Way-servers are placed in strategic locations crossed by users when they enter the service area. The roles of a way-server are: i) to be the publisher of control-data; ii) to provide mobile terminals with a loose system clock reference; iii) to inform the administrative server about the current status of systems parameters, such as the number of users, the number of subscribers per-topic, etc.

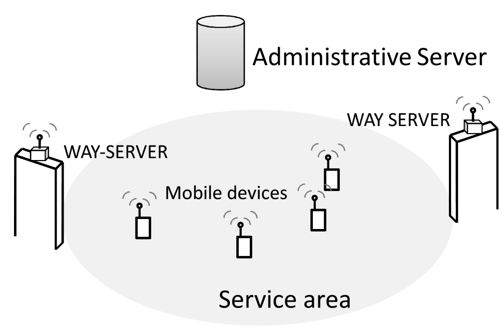

Fig. 1. System architecture, composed of an administrative server, one or more way-servers and several mobile devices

As regards the DTN routing paradigm, we choose ”Spray and Wait”[4] because of its simplicity, which suits the characteristics of IEEE 802.15.4 devices, and because of its native support for point to multi-point communication. Spray and Wait works as follows: when a new data-sample is published, it is replicated on L different nodes of the network, including the source, and this process is called spray phase. Subscribers can retrieve the data-sample when they are in direct radio contact with one of these L nodes. Thus, the more the replicas in the system, the smaller the average time between the publication of a data-sample and its reception by subscribers. However, as the memory that nodes made available for storing replicas becomes full, the spray diffusion phase gets slower, because the probability that a node can not store replicas increases. We studied this trade-off in [6], where we derived guidelines to determine the optimal number of replicas per topic. A simple rule of thumb is to use the 80% of all the memory and to share this memory among data-samples, proportionally to the popularity of the topic the data-sample belong to; i.e. data-samples of popular topics have to be replicated more times.

We point out that "Spray and Wait" is not a fully-fledged protocol but only a routing scheme, thus, as we can see in Figure 2, we designed a publish-subscribe, receiver-driven full protocol that works as follows: a node pulls data-samples when it either wants to retrieve subscribed-to data-samples or when it wants to replicate data-samples. To this end, nodes periodically emit Hello messages, which advertise the stored data-samples and if some of them need to be replicated [4].

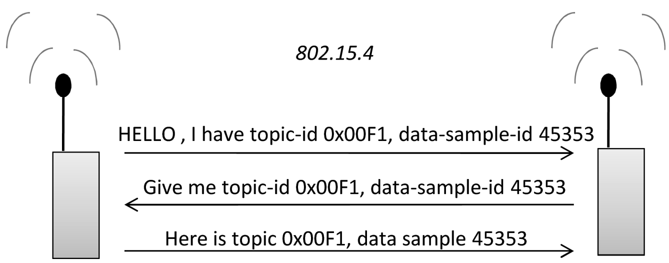

Fig. 2. Example of message exchange for data-sample retrieval

Moreover, since the available memory of nodes is limited, oldest data-samples have to be removed from the system when new data-samples are published on the same topic. This means that we need a way to distinguish newer data-samples from older ones. To support *data obsolescence* we loosely synchronize all devices with the clock reference provided by way-servers and use the publishing time to set the data-sample-id. A node that is replicating a data-sample marks it as "removable" if the node sees a neighbor node with a newer data-sample of the same topic. Removable data-samples will be indeed removed from nodes' memory if space is needed for new data-samples, so that the system memory is efficiently used.

Time reference and other *data control information* are periodically distributed in the network by using the same publish-subscribe mechanism. To this end we define a special topic called "built-in topic", whose publishers are only the way-servers and whose subscribers are all user nodes. Built-in topic data-samples are replicated in an epidemic manner on all the nodes of the network. When a node periodically receives the control updates published on the built-in-topic, the node sets itself as being inside the service area and activates the DTN functionality (i.e. it starts participating to the replication process). Conversely, if a node does not receive the periodic update published on the built-in-topic, the node declares itself as out of the service area and switches off the DTN functionality.

Finally, to make possible the DTN routing we need a way to establish connections between intermittently connected nodes. Unfortunately the IEEE 802.15.4 Standard does not provide any kind of "*ad-hoc*" mode but it requires one and only one PAN coordinator for each personal area network (PAN). Given that security support is not (yet) considered in our scenario, and that we use unique 64bit extended addresses (i.e. MAC addresses), we do not find any reason to follow the rule of having a unique PAN coordinator. Thus, we force all nodes of the system to be PAN Coordinators.

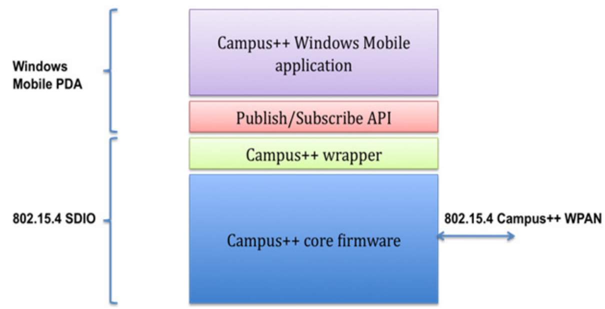

Fig. 3. Campus++ mobile node software

III. Implementation of the System

We implemented the DTN and publish-subscribe functionality on the TI CC2430 System on Chip (SoC)[5], which is an IEEE 802.15.4-capable device shipped with 128 KB of flash memory and 8KB of RAM. We have chosen to implement our system inside the firmware chip so as to use the same code for different nodes of the Campus++ architecture (way-server, user nodes, other nodes based on CC2430), which could have different OSs (e.g., Windows Mobile, Android), or could be stand-alone nodes not necessarily embedded in other devices. For instance, in the case of mobile phones this chip is embedded in the ZSIM [1]; in the case of PDAs this chip is embedded in a SD card (ZSD), and in the case of way-servers this chip is a stand-alone node.

Figure 3 outlines the software architecture of a Campus++ mobile node. Campus++ services are provided to users through a Windows Mobile application, which interacts with the TI CC2430 firmware by means of a publish/subscribe API. The publish/subscribe API sends/receives to/from the firmware serialized data-units that are deserialized/serialized by a wrapper. Finally, the wrapper exchanges data with the Campus++ core firmware, that implements DTN and publish-subscribe functionality.

Figure 4 shows the core firmware architecture. Campus++ protocols rely upon the Texas Instrument implementation of the IEEE 802.15.4 MAC (TI-MAC) and use IPv6 as network layer. To transport IPv6 datagrams within the small 802.15.4

PDU, we use the 6loWPAN adaptation layer [7]; this is documented in our previous work [2]. On top of the transport protocol (UDP in our case), we develop the Campus++ protocol, which is composed of three modules: i) Campus++ process event, which implements the state machine of the protocol and the main Campus++ features; ii) Data Sample Memory, which handles the memory where the replicas of data samples spread by the Spray and Wait DTN routing algorithm are stored; iii) Time Reference Management, which is responsible for the time reference synchronization, necessary for properly handling data obsolescence. The total code-size of the firmware is about 60 KB.

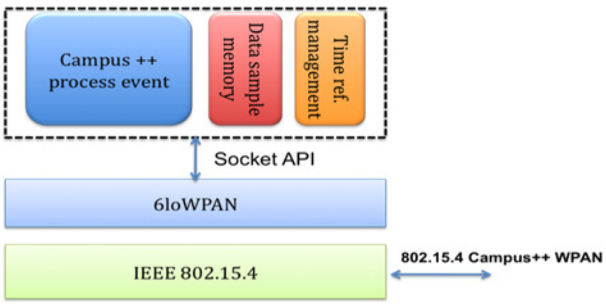

Fig. 4. Campus++ core firmware architecture

IV. Description of the implemented Demo

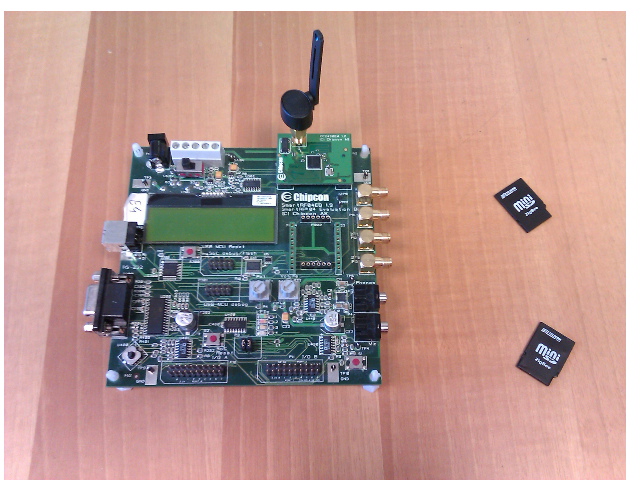

Fig. 5. Basic elements of the Campus++ architecture: on the left side, a TI CC2430EB playing the role of way-server; on the right side, two 802.15.4 enabled SD cards.

In the demo scenario, we use three mobile nodes (i.e. PDAs) and one way-server. We put a mobile node (node A) in a room together with the way-server; we put another mobile node (node B) outside that room so that A and B can not communicate directly between each other. A third mobile node (node C) keeps on moving inside and outside the room, allowing A and B to communicate between themselves, by carrying data-samples generated by node A to node B and viceversa. The Windows Mobile Application allows users to subscribe and un-subscribe to some of the available topics in the system, to publish new data-sample, to be notified when new data-samples of a subscribed topic are retrieved, and to visualize all the retrieved data-samples (Figure 6).

Data-samples retrieval can be performed for two different reasons: i) to provide users with data-samples of topics they are interested in, or ii) for replication purposes, independently of the user subscription status, and taking into account only the available memory. In the latter case, data-samples are not passed to the user application running in the PDA but are temporarily stored on the SD plugged in that PDA and equipped with CC2430, until a newer retrieved data-sample will take its place.

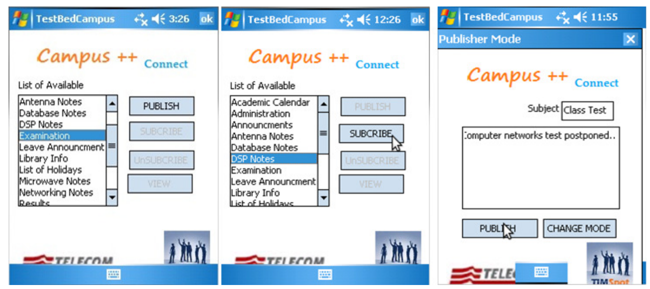

Fig. 6. Screen-shots of the Campus++ Windows Mobile application.

V. Conclusion

In this demo we show a system that provides location-based publish-subscribe services by exploiting a DTN networking paradigm and 802.15.4 technology. We believe that such an approach could be integrated in opportunistic community tools, e.g. [9] [10], so as to extend their functionality in terms of multi-hop connectivity, avoiding the need to explicitly send user location to other nodes.

Acknowledge

This work has been carried out in cooperation with and with funding from Telecom Italia.

References

[1] M. Turolla, E. Alessio, *ZSIM enabling innovative services to improve quality of life*, white paper on line available at www.zigbee.org/imwp/download.asp?ContentID=10403

[2] F.Lo Piccolo, D. Battaglino, L. Bracciale, A. Bragagnini, M.S. Turolla, N. Blefari Melazzi, *Towards fully self-configuring and IP-enabled IEEE 802.15.4 LR-WPANs*, in the proceeding of Med Hoc Net 2010

[3] RFC 4944, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks.*

[4] T. Spyropoulos , K. Psounis , C. S. Raghavendra, *Efficient routing in intermittently connected mobile networks: the multiple-copy case*, IEEE/ACM Transactions on Networking (TON), v.16 n.1, p.77-90, February 2008

[5] CC2430 web site, http://focus.ti.com/docs/prod/folders/print/cc2430.html

[6] A. Detti, D. Battaglino, L. Bracciale, F.Lo Piccolo, A. Bragagnini, M.S. Turolla, N. Blefari Melazzi, *A topic-based, publish-subscribe architecture for intermittently connected 802.15.4 networks* to be presented in Mobile Summit 2011 and available for peer review at http://stud.netgroup.uniroma2.it/ lorenzo/mobilesummit.pdf

[7] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, RFC 4944, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks.*

[8] P. T. Eugster, P. A. Felber, R. Guerraoui, A. Kermarrec, *The Many Faces of Publish/Subscribe*, in ACM Computing Surveys, V. 35 - I. 2, 2003

[9] LoKast mobile application, http://www.nearverse.com/lokast/

[10] Nokia Instant Community, http://research.nokia.com/news/9391