Smart Forests: fire detection service

Guilherme Borba Neumann, Vitor Pinheiro de Almeida and Markus Endler
Department of Informatics
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro - Brazil
Email: [gneumann,valmeida,endler] @inf.puc-rio.br

***Abstract*—Smart Forest is a concept derived from the Internet of Things (IoT) and defines sections of a forest where remote sensing is applied to collect data about environmental conditions. One of the main objectives of Smart Forests is to detect wild fire at early stages. However, the required technology for such monitoring usually demands a complex and expensive sensor and network infrastructure and requires central processing capabilities for analyzing data from several thousands of sensors. The goal of this work is to propose a solution focused on Edge Computing, using the concept of Mobile Hubs (M-Hubs). The developed Fire Detection IoT prototype application is based on the ContextNet middleware, and uses Event Processing Agents (EPAs), running on smart phones carried by forest guards. Scalability tests up to 10,000 remote sensors connected per M-Hub were performed.**

I. Introduction

In the context of Internet of Things (IoT), the term smart things has become more and more common. The Smart Cities have been imagined as a technological solution for social and environmental problems. In many countries, a large range of applications are being developed for these purposes [1]. Smart Forests appears as an option for sustainable management of natural environments, including environmental quality studies, especially in regard to climate change, wildlife monitoring, as well as remote monitoring and early detection of natural disasters and fires.

In the United States, the U.S. Forest Service is investing time and money in a variety of tools for smart forests, collecting biotic and abiotic data, and using this information to protect those regions (https://smartforests.org/). In Brazil, a similar effort is being conducted by Instituto Nacional de Ciência e Tecnologia de Sistemas Micro e Nanoeletrônicos (INCT NAMITEC) and by the Instituto Florestal de São Paulo.

Smart objects (with temperature and humidity sensors), may be used in the fire detection routine and for notifying relevant entities about an event. Using wireless technologies of different ranges and capacities for detection and communication (e.g. WPAN, WLAN, WiMax, RFID (Radio-Frequency IDentification), Zigbee, 3G/4G) remote sensors deployed in the forest can generate a large amount of data about environmental conditions. It is also a reality to connect sensors to hardware like Arduino and Raspberry, to temporarily persist data. In agriculture, for example, smart things are present too, improving soil fertilization and water management. According to TongKe [2], who proposed modernization and informatization in China's agriculture, RFID and IoT technology can be combined, using cloud computing, to realize smart agriculture. The goal of this work is to propose an application focused on Edge Computing (participatory sensing), using the concept of Mobile Hubs (M-Hubs).

Section II will consider related work. Section III will expose the application idea based on Mobile Hubs and section IV will briefly show some results of Scalability tests. Finally, section V will explore the final remarks about this work.

II. Related Work

As mentioned before, the U.S. Forest Service is developing a Smart Forest Network to collect environmental data in order to better understand, control and protect the green areas. The main architecture of this system is composed by Wireless sensors, which send data to a central base, where it is analyzed and stored in a Web database free to public. The data is analyzed once a day, considered by them as a near real-time system [3].

On the other hand, the Factor-Link SISVIA project, that uses Libelium sensors, implements three main parts in the architecture: Wireless Sensor Network, Communications Network and Reception Center [4]. The data is collected by Waspmote hardware and analyzed in real time [4]. This approach requires a complex and expensive infrastructure: transmission towers, base stations and staff for maintenance.

However, these two projects implement centralized processing, creating a demand on receiving data from every site and processing that in the cloud. A first real-time Wireless sensor network to fire detection was proposed in 2005, using a neural network method [6], innovating the satellite-based monitoring in use that time to wildfire detection in forests. This paradigm was also present in [7].

The intention here is to think a network that can ensure shorter response time and better reliability, from edge computing perspectives [5], considering the participatory sensing strategy. The smart objects (basically smart-phones and sensors) are going to represent a simpler network to be implemented in a Forest. The cost is also very important to the application. Edge Computing can improve cost-benefit, when compared to centralized solutions.

Investigating participatory sensing using movable/replaceable sensors was the main goal in [8]. The idea is that people staying or passing through a space leave tiny and inexpensive sensors in points of interest, thereby exploring and engaging with their environment. Although they do not focus exclusively on forest monitoring, but advocate the general monitoring of environment conditions of any urban space, such as air pollution, noise, or water quality, from the perspective of sensors connectivity it shares almost the same challenges as our approach. However, instead of being interested in (sensor) data probing and transmission accuracy and completeness, the authors of [8] used only mock sensors - with no real data gathering and transmission - but just wanted to investigate how each stakeholder (i.e. cab drivers, student, adult, biker, homeless, etc.) perceived the need to monitor his/her environment and which aspect of it. This was used to identify which concerns and incentives leverage participatory sensing for city-wide grassroots activism. Translated to forest monitoring, and assuming that sensors were widely available and very inexpensive, it would mean how each person would choose places to leave/drop sensors for early detection of wildfire.

III. Application

In this paper we use WPAN, Bluetooth Low Energy sensors. Bluetooth sensors were chosen because are cheaper than Wi-Fi ones, making possible a bigger number of sensors in the forest - Bluetooth Low Energy provides a theoretical range over 100 meters. The smart-phones receive data from them and, after processing the data, send it to the Smart Forest Server, to be stored and to notify entities about the events occurred. The sensors are located at tree tops, and also at undergrowth, to easily detect abruptly temperature and humidity changing due fire. Every guard has a smart-phone, and almost every visitor. An application that provides park information about trails, fauna and vegetation, may be offered in association with Mobile Hubs to incentive people install that.

The proposal of this work is to process data from sensors in the edges. ContextNet Middleware [9] was chosen to attempt this purpose, since it has Mobile Hubs implementation available. A first JAVA application was created to detect fire and fire risk in a green area, more appropriated to Natural Parks where Internet connection (4G) is able and where park guards' and visitors' smart-phones (hubs) are accessible.

The ContextNet uses a Scalable Data Distribution Layer (SDDL) [10] to communicate in the remote network (Wi-Fi and 4G). Two protocols are present in SDDL: OMG Data Distribution Service for Real-Time Systems (DDS) [11] in a Publish-Subscribe asynchronous communication and MR-UDP to robust communication between mobile nodes and Gateway [10,12].

The project's architecture is basically composed of Physical devices - Sensor Tags [13] -, Mobile Hubs (M-Hubs) [14], Entities (Actuators) and the Smart Forest Server (Figure 1). The steps below explain the way Fire Detection Application works.

1. Sensor Tags send data via Bluetooth Low Energy to M-Hubs installed in guards' and visitors' smart-phones.
2. M-Hub connects with Smart Forest Server (SFS), and receives the Complex Event Processing (CEP) rules [15, 16]. The CEP rules implemented in this work are related with temperature and humidity. In these rules, the temperature data from each sensor is used to generate averages each 10 seconds, and then, the higher one is sent to the SFS. In the case of humidity, all averages are sent to the SFS each 10 seconds, because the relative humidity does not vary in the same way as temperature.
3. Data from sensors are updated each 20,000 milliseconds.
4. Data from sensors are processed in M-Hubs and are sent to the SFS throughout a Gateway. When no network is available (neither Wi-Fi nor 4G), the data is stored in a buffer and it is sent later to the SFS.
5. In the SFS, data from distinct types are received: event data and sensor data. The sensor data is persisted to future statistics analysis and the event data is immediately analyzed to notify the interested entities: Fire Department and/or Supervisor - Supervisor entity is the person responsible for the protected area (when applicable).

Temperature above 100°C in Forests and above 300°C in camping area are notified to Supervisor and to Fire Department. Temperature above 100°C in camping area is notified only to Supervisor. According to the Laboratório de Incêndios Florestais, from Universidade Federal do Paraná, "The average temperature in a fire depends on several factors, such as propagation speed, type of fuel and time of burning. However, most surveys indicate maximum temperatures between 600 and 800°C, although sometimes they are below 300°C or above 1000°C" [17]. When relative humidity is below 35 Supervisor is also notified about fire risk.

M-Hubs executes in Android System with four java threads: The Connection Server that executes MR-UDP, sending, receiving and buffering messages; the Sensory, Presence and Actuation (S2PA) Service responsible for connecting mobile objects (Sensor Tags) through WPAN technologies; the Location Service that uses GPS information to notify Connection Server the location; and the Energy Manager, that regulates the operation frequency to other threads depending on the device energy level. In M-Hubs there is an agent from CEP called Mobile-EPA (Event Processing Agent), using the language Asper, responsible for the processing in Edge Computing.

Another important concept in ContextNet is the ClientLib [14]. ClientLib is a library used for asynchronous communication between mobile clients. It encapsulates the MR-UDP protocol to a nodeConnection interface, manages a list of Gateways IDs available and realizes handovers when necessary. The ClientLib is present in M-Hubs and in the Entities Supervisor and Fire Department. That is important to note that only Server is attached with an IP and port, other entities use UUID, a universal identifier used in ContextNet.

As part of the Fire Detection Application, it is possible to determinate the location in which the fire event has been detected and take different actions depending on the context (using the GPS coordinates sent from M-Hub). The ContextNet libraries and the M-Hub application can be downloaded from http://www.lac.inf.puc-rio.br/dokuwiki/. The Fire Detection App is available at https://github.com/neumannguib/fire_app.

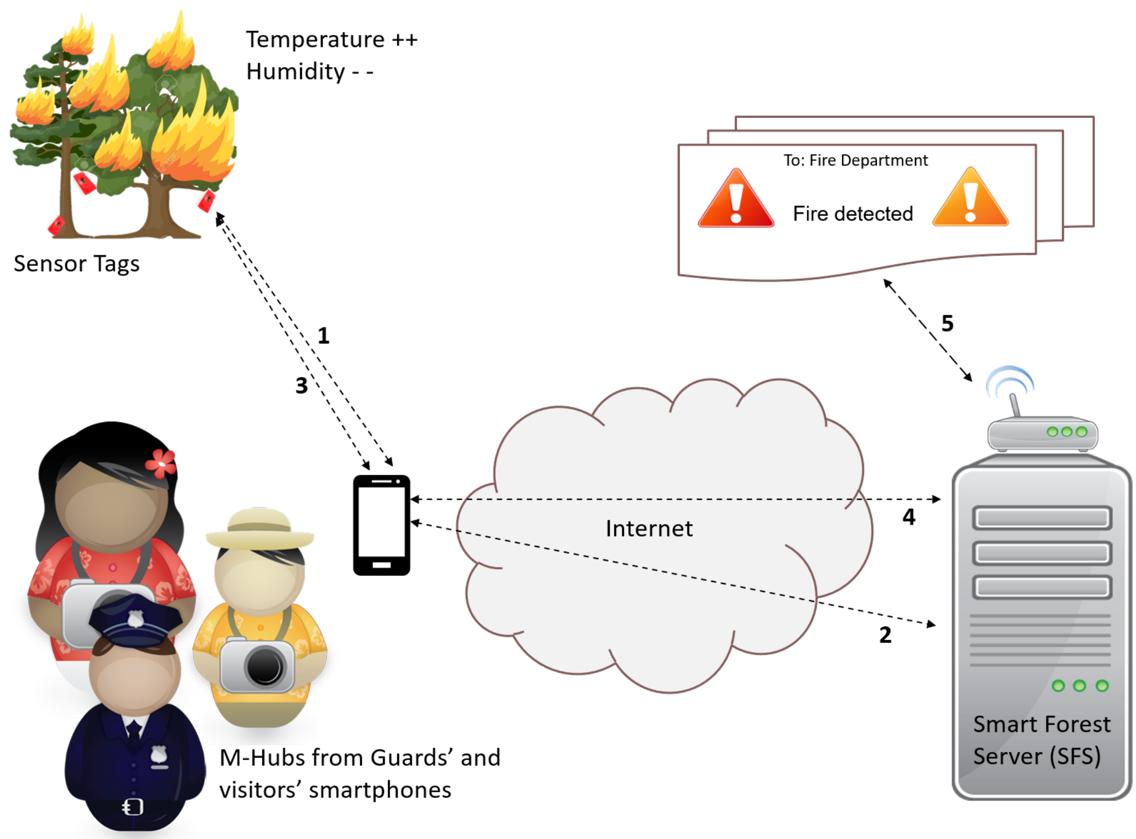

Fig. 1. Fire detection using M-Hubs and ContextNet middleware

TABLE I
SCALABILITY RESULTS FROM MOBILE HUBS INCREASING THE NUMBER OF MOBILE OBJECTS PER M-HUB

Number of Mobile Objects	Number of Event packages received	Number of Sensor Data packages received
10	39	310
100	40	3,011
1,000	43	23,557
1,500	40	28,154
2,000	40	26,360
2,500	40	25,005

IV. PERFORMANCE EVALUATION

The application was tested for scalability, in terms of number of connected mobile objects (Sensor Tags), using a sensor data simulator (https://bitbucket.org/endler/app-mobile-objects-simulator). Temperature and humidity data were simulated changing the amount of data sent each 10,000 milliseconds. A modified Mobile Hub was used to receive the simulated sensors throughout wireless connection, in an Android System 5.1, Moto E. The sensors simulator was run in an Intel(R) Celeron(R) 3205U @ 1.50GHz, 1501 MHz, 2 Cores and 4GB RAM in Ubuntu 16.4.

A single Mobile Hub could connect to 2,500 simulated mobile objects (table 1). Each mobile object had temperature and humidity data. After receiving data from all mobile objects, the M-Hub processed the data looking for events - the CEP rules were executed at the M-Hub. So, all sensor data was sent to the SFS as well as event data. It is worth mentioning that before updating the data each 10,000 milliseconds, the M-Hub processed an initialization to become ready to start collecting the data from all simulated mobile objects. Each test was run in intervals of 300 seconds, measured through time-stamps available in M-Hub packages sent to the SFS. These results are shown in the table 1.

In the case of 5,000 and 10,000 mobile objects, the M-Hub could not initialize. Also, can be noted that after 1,500 mobile objects, the number of sensor data received started decreasing. Anyway, it means that each M-Hub could be used as a data collector and edge processor for up to 1,500 mobile objects, which is sufficient for the participatory servicing of Fire Detection.

V. CONCLUSION

This paper aimed to introduce the idea of implementing a low-cost infrastructure for Smart Forests using mobile objects and mobile hubs to detect fire. The same approach can be extrapolated to a great variety of applications in Forests: water, soil and air quality; fauna dynamics; endangered species studies; etc. Mobile Hub appears as an option for edge computing in Smart Environments. Preliminary results showed that M-Hubs can connect to 2,500 mobile objects, receiving data from temperature and humidity and processing all these data (looking for events) before notifying Smart Forest Server.

A lot of effort need to be done, such as testing new technologies and possibilities to generate shorter response time and reliability needed in a Smart Forest. In practice, it would be interesting to use sensors powered by solar panels and also to use Arduino-like platforms on the trees to collect data when no person is close enough to the sensors. The data stored in those hardware could be collected periodically by Guards, in predetermined routes, or also by drones [18]. Drones and guards routine need to be created in order to generate as big as possible range of data collected from all sites.

Focusing on edge computing, some functions may be moved to Mobile Hubs, like Awareness Context and notification for actuation. In this case, the mobile hub that detected fire could notify close mobile hubs about the current event - besides notifying the Smart Forest Server -, to better control the incident.

Finally, it is also important to think locally. The technology implemented will depend on the context and on the money available for that. Many times, a simple application can solve the problem.

Acknowledgment

The authors would like to thank the LAC (Laboratory for Advanced Collaboration) team, who supported us on Context-Net middleware.

References

[1] J. Colding and S. Barthel, “An urban ecology critique on the Smart City mode”, Journal of Cleaner Production,vol. 164, pp. 95-101, oct. 2017.

[2] F. Tongke,“Smart Agriculture Based on Cloud Computing and IOT”, J. Converg. Inf. Technol., vol. 8, no. 2, 2013.

[3] U. S. Forest Service, Smart Forests for the 21 st Century: A New Initiative to Develop EFR Cyber Infrastructure.

[4] J. Solobera, “Detecting Forest Fires using Wireless Sensor Networks”, Libelium, 2010. [Online]. Available: http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, Edge Computing: Vision and Challenges, IEEE INTERNET OF THINGS J., vol. 3, no. 5, 2016.

[6] L. Yu and N. Wang,“Real-time Forest Fire Detection with Wireless Sensor Networks,”IEEE, pp. 12141217, 2005.

[7] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo,“Forest Fire Detection System based on Wireless Sensor Network,”IEEE Conf. Ind. Electron. Appl., pp. 520523, 2009.

[8] S. Kuznetsov and E. Paulos,“Participatory Sensing in Public Spaces: Activating Urban Surfaces with Sensor Probes”, in Proc DIS 2010, ACM Press, 2010, pp. 2130.

[9] M. Endler, G. Baptista, L. D. Silva, R. Vasconcelos, M. Malcher, V. Pantoja, V. Pinheiro, and J. Viterbo,“ContextNet: Context Reasoning and Sharing Middleware for Large-scale Pervasive Collaboration and Social Networking,” ACM/USENIX Middlew. Conf., 2011.

[10] L. David, R. Vasconcelos, L. Alves, R. Andr, and M. Endler, “A DDS-based middleware for scalable tracking, communication and collaboration of mobile nodes,” J. Internet Serv. Appl., vol. 4, no. 1, p. 16, Jul. 2013.

[11] G. Pardo-Castellote,“OMG data-distribution service: architectural overview,” in 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings., pp. 200206.

[12] L. D. Silva, M. Endler, and M. Roriz, “MR-UDP: Yet another Reliable User Datagram Protocol, now for Mobile Nodes,” PUC-Rio, 2013.

[13] Texas Instruments, “TI Designs Multi-Standard CC2650 SensorTag Design Guide”, 2015.

[14] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. J. Da Silva E. Silva, “The Mobile Hub concept: Enabling applications for the Internet of Mobile Things,” in 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2015, pp. 123128.

[15] D. C. Luckham, *The power of events: an introduction to complex event processing in distributed enterprise systems*. Boston, USA: Addison-Wesley Longman Publishing Co., 2001.

[16] G. Cugola and A. Margara, “Processing Flows of Information: From Data Stream to Complex Event Processing”, ACM Comput. Surv., 2012

[17] Floresta WebSite, 2017, “Incêndios Florestais”, [Online]. Available: http://www.floresta.ufpr.br/firelab/incendios-florestais/.

[18] B. Olivieri and M. Endler, “DADCA: An Efficient Distributed Algorithm for Aerial Data Collection from Wireless Sensors Networks by UAVs” in MSWiM17, 2017.