Applied Soft Computing 35 (2015) 708–716

Contents lists available at ScienceDirect

Applied Soft Computing

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

Fuzzy decision method to improve the information exchange in a vehicle sensor tracking system

Guillermo Cueva-Fernandez[a,*], Jordán Pascual Espada[a], Vicente García-Díaz[a], Ruben Gonzalez-Crespo[b]

[a] Department of Computer Science, University of Oviedo, c/ Calvo Sotelo s/n, 33007 Oviedo, Asturias, Spain
[b] International University of la Rioja, Spain

ARTICLE INFO

Article history:
Received 20 September 2014
Received in revised form 22 January 2015
Accepted 23 January 2015
Available online 7 February 2015

Keywords:
Vehicle sensor
Fuzzy logic
Fuzzy decision making
Tracking
Resource management
Bandwidth

ABSTRACT

Internet of Things is based on the identification of real-world objects in a unique way that interconnects them by means of communication interfaces. Such a simple idea allows the emergence of a huge number of new applications in almost any domain of knowledge. One of the most prominent areas of application is road vehicles, which currently have, on average, more than 50 sensors inside and which information can be accessed through a standard protocol. With this, vehicles have become real smart objects that can interact with other objects or any software system. To allow that, our previous work focused on proposing and developing the Vitruvius platform, where users with no programming knowledge can design and quickly generate Web applications based on the real-time data consumption from interconnected vehicles. The problem is that the sending of such information is out of control, being unable to filter out when the best time to send the information is and what information should be sent at any moment in order to minimize the resource consumption of the mobile device that acts as a bride between the vehicle and the database in which all the information is stored. Thus, in this work we propose a fuzzy algorithm that allows to optimize the resources that are used by real-time applications that constantly send data while maintaining data quality, contextualized in vehicle sensor tracking systems and the applications that can be built above them.

1. Introduction

Internet of Things (IoT) is a concept proposed by Ashton [1] that tries to identify real-world objects in a unique way, creating a graph-like structure with all the related information. The main idea is to integrate several technologies such as identification and tracking methods, wired and wireless sensors and actuators, communication protocols or distributed intelligence [2], being the "smart" objects the building blocks for the IoT [3]. Those objects can be used to create a large amount of different applications such as for example: (1) helping people with disabilities with assistance and support to achieve a good quality of life [4]; (2) efficient smart homes with ubiquitous services [5]; (3) therapy management in ambient assisted living taking into account different parameters such as illness, treatments, physical and psychological stress, drugs, etc. [6]; or (4) efficient logistics management [7]. Smart objects could be any classic real-world object with a digital component equipped with sensors, actuators and communication interfaces, usually Wi-Fi [8], Bluetooth or IrDA. Sensors and actuators are very important since they are used to gather information from the real world for different tasks of monitoring and react according to the information obtained.

An industry in which the sensors are becoming of vital importance is the transport industry. Modern vehicles have a large number of sensors that show important information about the current state of the vehicle (e.g., vehicle speed, fuel pressure, throttle position, engine RPM, engine coolant temperature, etc.). For that, the OBD-II (On Board Diagnosis) standard port is used [9], at least in all new passenger road vehicles in EU and USA. It provides a set of common mandatory sensors, opening the possibility of including brand-specific sensors. It makes it possible to directly join together the IoT and vehicles (e.g., He et al. worked on a multi-layered vehicular data cloud platform to provide services such as intelligent parking [10]).

Traditional logic deals with binary sets of values and is used in a huge amount of applications. However it is not always the best solution since the values can only be two, true or false. Thus, fuzzy logic [11] deals with reasoning what is approximate rather than

* Corresponding author. Tel.: +34 985103397.
E-mail address: guillermo.cueva.fernandez@gmail.com (G. Cueva-Fernandez).

http://dx.doi.org/10.1016/j.asoc.2015.01.066
1568-4946/

exact, by means of a truth value that ranges on a scale between 0 (completely false) and 1 (completely true). This idea is widely used in many fields such as artificial intelligence and control systems when there is no clear separation between what is true and what is false. The same idea could be applied to determine if it might be a good time to transmit data, and what data, through a computer system. As an example, with traditional logic we could say: "if there is new data and speed is high then send information to database". However, the real world is not usually so easy to manage and decide what is new, or what is a high value. It is not usually a binary issue since it can depend on various factors and, at the same time, in different degrees.

The goal of this work is to propose a fuzzy algorithm to reduce the resources that are used by real-time applications that constantly send data through any interface. This is expected to optimize their operation. The focus is to know when the best moment occurs to send the information and what information should be sent. We think that our previous work, the Vitruvius platform, is a clear example of a scenario in which this proposal could be useful since it is based on a real-time applications in which data (currently each vehicle has an average of more than 60 sensors on board) is constantly sent (between one and four seconds) through a resource-constrained device as the smartphone of any user is.

The rest of this paper is organized as follows: Section 2 includes the background of related works, Section 3 describes the proposal and the approach followed to meet the goal, Section 4 shows the results and Section 5 discloses the conclusions.

2. Background

There is a growing concern about the amount of resources spent by computer systems [12]. Thus, new concepts like green computing progressively arise [13]. It refers to the study and use of environmentally sustainable computing and is a broad field that tries to efficiently and effectively manage computers, servers, communication systems or devices such as monitors, printers, disks, etc. It is such an important need today that many government agencies have implemented standards and regulations that encourage green computing (e.g., Energy Star program includes efficiency requirements for computer equipment since 2006 [14]).

However, it also includes the design and programming of good algorithms [15] to reduce the use of resources such as battery consumption, bandwidth of the network or CPU cycles. All of this leads to energy and monetary savings. As an example, an average Google search releases between 0.2 [16] and 7 [17] grams of carbon dioxide, depending on the source consulted.

There are a large number of works in which different algorithms have been applied to achieve resource waste reduction. For example, Mezmaz et al. [18] investigated the problem of scheduling precedence-constrained parallel applications on heterogeneous computing systems, proposing a parallel genetic algorithm that takes into account the energy consumption of the system. Other authors like Berl et al. [19] identified some of the research challenges that need to be addressed regarding cloud computing environments for energy efficient processing. However, the focus of this work is on resources for mobile devices which have some specific needs, and are greatly related to the search for energy savings.

From the point of view of mobile computing, there are special challenges [20] such as: (1) short duration of the batteries, which although they have increasingly greater capacity, they also have higher consumption because the resources used by devices (e.g., due to the increasing size of the screens); (2) data connections are usually much more constrained and limited than data wiring connections in a fixed place; and (3) total computational resources that are available are generally much lower than those available in a laptop or desktop computer. The following sections will deepen in those topics.

2.1. Focusing on battery usage

Batteries and their low duration have always been one of the weaknesses of mobile devices. Thus, to compensate the increasing power of such devices, batteries have been increasing their size too. However, that does not seem to be enough and various additional electronic gadgets have appeared to help keep active the devices for more time. For example, adapters to increase the battery capacity, and therefore the size of the mobile device, or external battery chargers that can be carried to charge the mobile device even where there is no power supply in which to be connected. However, that does not seem to be a global solution to the low duration problem of batteries and is one of the reasons why there are multiple research works focusing on mitigating the effects of the limitation of current batteries for mobile devices.

For example, Pathak et al. [21] claim there are lots of different types of bugs faced by millions of devices (energy bugs) that caused an unexpected amount of high energy consumption. Perrucci [22] says that mobile phones should switch the network in dependency of the service (e.g., 3G networks need more energy for text messaging and voice services but 2D networks need more for downloading large amount of data) to save energy. Others like Lin et al. [23] designed an adaptive location service that helps to reduce the high battery drain caused by GPS receivers.

2.2. Focusing on network traffic

Although mobile devices have progressively bettered, with faster and with higher capacities, Internet connections (e.g., in many cities there are free WiFI areas for people), etc., we are still far from achieving full and unlimited connectivity. We depend on the rates offered by operators, which tend to be expensive and with many limitations from the point of view of the bandwidth and the amount of data that can be sent/received, especially when compared to the connections that can exist in fixed locations such as offices or homes. This makes reducing the amount of data that is sent and received a priority in many scientific papers. Although it is not always necessarily true, it is important to note that reducing the amount of data that is sent and the CPU cycles that are needed are likely to reduce the battery used.

Some interesting works include the one by Pering et al. [24] in which they propose a method to enable a device to automatically switch between multiple radio interfaces such as WiFI and Bluetooth in order to optimize the transmission. Marsan & Meo [25] evaluate the amount of energy that can be saved by using all networks when required, but progressively disconnecting networks when traffic decreases. Bojic et al. [26] proposed an agent-based mechanism for auto-tuning mobile networks with the aim of achieving energy savings when users try to access them. Ma et al. [27] created a tool for refactoring traditional websites to be adapted to smartphones taking into account, among other things, that under limited networks users might just want to load their desired contents, to reduce fee and speed up loading time.

2.3. Focusing on CPU cycles

Related to all the above aspects and many others is the CPU consumption of mobile devices. Currently, these devices are full-fledged computers with very powerful features but, mainly due to the cost of small components, they are far from what other more conventional computers could provide. This means that we have to be careful with application designs to get applications running