Ad Hoc Networks 18 (2014) 85–101

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

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

Efficient naming, addressing and profile services in Internet-of-Things sensory environments

Chi Harold Liu [a,*,1,2], Bo Yang [b,2], Tiancheng Liu [a,2]

[a] *Internet-of-Things Infrastructure Management, IBM Research, China*
[b] *Emerging Technology Institute, IBM China Development Laboratory, China*

ARTICLE INFO

Article history:
Received 26 July 2012
Received in revised form 8 December 2012
Accepted 22 February 2013
Available online 15 March 2013

Keywords:
Internet-of-Things
Naming and addressing
Middleware
System designs and implementation

ABSTRACT

We present a naming, addressing, and profile server (NAPS) as a middleware to bridge different platforms in Internet-of-Things (IoT) sensory environments. Given massive amount of heterogeneous devices deployed across different platforms, NAPS serves as the key module at the back-end data center to aid the efficient upstream sensory data collection, content-based data filtering and matching, and downstream efficient control by applications. While previous research efforts only focus on a specific standard or protocol, we aim to design a middleware component servicing dynamic application needs, and sensors/actuators deployment and configurations across different platforms. Specifically, we propose a complete design of NAPS, including its key functionalities, system flows, interfaces, and individual module design. We further propose a unique device naming and addressing convention, and show its applicability to a few widely-used standards and protocols. We also propose an efficient identifier generation scheme; and we demonstrate a full implementation of the above designs with a case study, including a service registration portal. Finally, performance evaluation is done against the system throughput.

1. Introduction

The Internet-of-Things (IoT) is a novel paradigm rapidly gaining wide attention from academia, industry and government agencies [1,2]. Its basic concept is that "… things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts…" [3]. These global *networked* things include Radio Frequency Identification (RFID [4]) tags, ZigBee sensors, smart phones, etc. An example is that the US Department of Environmental Protection deploys thousands of water pollutant and water level sensors in Hoover Dam to periodically monitor its water quality and potential hazardous scenarios.

Nevertheless, the lack of a *de facto* standard architecting a naming, addressing and profile server (NAPS), as a middleware [5] interoperable with heterogenous platforms has become a key limiting issue on its proliferation to deployment [6]. The research community are hearing the strong desire from application developers to avoid learning heterogenous communication/networking protocols in use, but be provided a homogeneous naming and addressing convention, so that they are able to retrieve the data from sensors and control the actuators of different platforms and network domains. Towards this end, a higher layer of device naming-addressing mapping should be provided to integrate with legacy systems and different platforms. As for device naming, the convention should

[*] Corresponding author. Tel.: +86 (0) 10 58748833; fax: +86 (0) 10 58748330.

E-mail addresses: chiliu@ieee.org (C.H. Liu), boyang@cn.ibm.com (B. Yang), liutc@cn.ibm.com (T. Liu).

[1] Diamond Building 19A, Zhongguancun Software Park, Haidian District, Beijing 100193, China.

[2] IBM Software Group, 3/F Ring Building, Zhongguancun Software Park, Haidian District, Beijing 100193, China. Tel.: +86 (0) 10 58748433; fax: +86 (0) 10 58748330.

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.02.008

contain key elements of device meta-data, such as device type and domain information; while for addressing, its format allows the granularity of efficient accessibility and addressability to the physical world. Profile services are also needed to aid the application query and system configurations, like device status and presence. Furthermore, sensing tasks are always achieved by a group of devices with similar sensing capabilities, and thus NAPS should provide device group management functionalities, such as to create, update, read, and delete (CURD) groups (and its tree-structured subgroups). In this way, application development logic is greatly simplified where only a device group name is needed and NAPS handles the internal mapping. As a middleware, it should extend its usability by providing abundant external interfaces.

IPv4, IPv6 and Dynamic Name Service (DNS) are usually considered as the candidate standard for naming and addressing, however due to the lack of communication and processing capabilities of many small and cheap devices (like RFID tags) it is quite challenging to connect every "thing" with an IP. Furthermore, with the increasing amount of end devices, even IPv6's address space may not enough. On the other hand, industry standards have put much effort in each application domain. EPCglobal [7] uses a 96-bit binary sequence to identify each RFID tag, and the object name service (ONS) for URL translation. OPC-UA [8] defines client–server based models for industrial production line solutions, where an abstract address space is formed by a mesh topology. In it, each node represents a sensor in the production stage and the edge between two nodes represents the stage-by-stage relationship during the production. As an overall service architecture, ETSI [9] proposed a solution interworking with 3GPP machine type communication (MTC) standard [10], to support machine-to-machine (M2M) communications when upgrading from traditional cellular networks where each device is with a unique international mobile subscriber identity (IMSI) and is IP addressable. Furthermore, as a service layer architecture, it defines a variety of service capabilities (SCs) including a Network Reachability, Addressing, and Repository (NRAR) SC. However, it has no technical details this far. Our goal in this work is to work with any service platforms as a middleware at the back-end data center. Therefore, all these efforts pay attention only to a specific network or application domain, however not applicable as a *common platform* managing different technologies and standards.

1.1. Scope and assumptions

Motivated by these facts, Fig. 1 shows an overall architecture from the physical phenomenon all the way up to the data center, considered in this paper. Devices such as sensors and actuators sense/control the physical world, which are interconnected either wirelessly or wired through a variety of access network technologies. A few known examples are cellular networks (2G/3G/LTE), IEEE 802.11/802.15 series of standards for WiFi, ZigBee, and Bluetooth, RFID readers and tags, and wireline technologies like power-line communications (PLC), etc. Usually there exists a gateway interconnecting the access networks and the backbone Internet.

Data are then routed either through the carrier public network or IoT private network. For the former, standard like 3GPP MTC is defined to upgrade the existing backbone cellular network to manage M2M devices. For the latter, most service layer architectures, like EPCglobal RFID architecture or OPC-UA client–server model, leverage existing transport layer protocols such as CoAP [11] over UDP, and HTTP over TCP. In the service layer, ETSI M2M service architecture can interwork with 3GPP MTC via interworking function (IWF) that enables seamless integration of M2M SC layer with cellular MTC devices. That is, M2M SCs can invoke and leverage cellular MTC functions to optimize and support better M2M services. Meanwhile, cellular MTC functions can be enhanced for M2M SCs. Companies like InterDigital and Juniper Networks have this kind of solution [12,13].

Towards this end, there is lack of a common platform interoperable with different platforms to hide this heterogeneity and provide a transparent naming service to applications. We therefore design an IoT - application infrastructure (IoT-AI) and its management platform (out of scope of this paper), as shown in the figure. The key technical entablements of IoT-AI are: application gateway (AG), NAPS and its service registration portal (Portal), and real-time operational database (RODB). AG coordinates the data filtering and processing, and control message delivery based on a uniform device naming and addressing convention in NAPS. The goal is to have applications access devices across different platforms without knowing their languages in detail, but focusing on the development logic only. The position of NAPS extends the functionality comparable to DNS in the Internet, to the profile services such as storage and query. We next present three assumptions of this work.

First is service discovery. Since the scope of NAPS is a middleware component at the back-end data center to hide the heterogenous protocols and standards, here we assume that service discovery has already been successfully performed by each platform individually, and stored in our NAPS repository. Examples are service discovery server enhanced from ETSI M2M service architecture by InterDigital [12], discovery service set in OPC-UA standard, and protocols like Universal Plug and Play (UPnP) [14], etc.

Second is the authentication, authorization and accounting (AAA). Although it is not the focus of this work, the design can largely leverage the Network Security Capability (NSEC) SC in ETSI M2M service architecture. It uses a key hierarchy, composed of root key, service key, and application keys. Root key is used to derive service keys through authentication, and key agreement between the device or gateway and the M2M SCs at the M2M Core. The application key, derived from service key, is unique as per M2M application. Issues like distributed denial-of-service (DDoS) attack will be discussed in Section 7.

Finally, we assume that wireless imperfection like packet errors and interference have been handled by the communication stack of each access networks. Solutions from PHY layer techniques (e.g., antenna techniques, modulation and coding) and MAC/network layer protocols (e.g.,