Supporting Efficient Machine-to-Machine Communications in the Future Mobile Internet[1]

Jun Li, Yanyong Zhang, Kiran Nagaraja and Dipankar Raychaudhuri
WINLAB, Rutgers University, New Brunswick, NJ 08901

***Abstract* — Machine-to-machine (M2M) communication represents an important new class of applications for the future "Internet of Things (IoT)". The IoT scenario requires scalable connectivity to billions of embedded devices such as sensors, RFID's and machines. Current sensor networking solutions involve the use of gateways for hierarchical access to sensor devices, but this approach leads to specialized M2M systems which are not visible to the vast majority of Internet devices and applications. The MobilityFirst future Internet architecture provides a new framework for efficiently supporting sensor net and M2M scenarios through the use of an identity naming system for all network attached objects using the concept of a "globally unique identifier (GUID)". The use of GUID results in a "flat" network where sensors and other embedded devices are visible to all applications and devices on the Internet. In addition, GUID names make it possible to introduce context and content-aware services which are well suited for M2M scenarios where attributes such as location or type of content are more important than physical address for establishing network connectivity. This paper presents a discussion of how M2M service scenarios are supported in the MobilityFirst architecture, introducing the protocol layers and API's as they apply to sensors and embedded devices. A proof-of-concept prototype implementation is discussed and an application to a sensor scenario is described.**

Keywords- M2M communications, sensor networks, Internet of things, future Internet architecture

I. INTRODUCTION

Machine to machine (M2M) communications over the Internet is expected to grow rapidly as more and more applications involve interactions with physical world objects [4]. A traditional M2M application is normally designed based on a dedicated sensor network consisting of physical sensors or tags. The sensors or tags bear the identities of the physical objects and transmit their status accordingly. The information for those physical objects is usually constrained to the scope of the dedicated applications. This vertical M2M market model has existed for a long time dating back before the Internet was invented. Such an isolated vertical model, however, limits the accessibility of the sensor networks and the reusability of the middleware and application software. The vertical model is therefore inherently high in cost and low in efficiency.

As more M2M applications with vertical models become available online, more efficient networking models are needed in order to remove the constraints of vertical markets and open machines and sensors to all Internet connected devices and appliances. There are many solutions for the new model based on alternative kinds of new Internet technologies, including web services based on service oriented architecture (SOA) that provide generic APIs for objects in the cyberspace [2]. Internet of Things (IoT) [3] envisions that physical objects (things) become network objects in the cyberspace that can communicate with each others and be accessible across the full range of Internet applications and users.

This vision of IoT faces two major challenges. One is the universal identity of Things and the other is the standardizations of the data format for Things. We have witnessed efforts in standardization to address the second challenge e.g. SensorML [8] and TransducerML [9]. This paper will focus on the first challenge, i.e. the identification of things in the Internet. Normally, physical objects are identified at product level through attached tags (EPCglobal, Barcode, RFID) [10] and network objects are directly identified at application level through defined names (URI: URL or URN) [11]. In the current Internet, both of them cannot be addressed or identified directly by the core network which leaves the IoT just another type of web applications.

One approach is to extend IP addressing from computers to Things, i.e., assigning all things an IPv6 address. IPv6 [6] has a huge addressing space. It defines the lowest 6 – 8 bytes of addressing space as device ID, inherent from the 6 bytes of MAC addresses [7]. Although 6 bytes are long enough for identifying all physical objects connecting to the Internet, there are a few fundamental issues with directly using IPv6 addresses as object identities. Firstly, it does not support mobility. The devices' dynamic connectivity or lost connectivity and multi-homing cannot be handled efficiently in IPv6, and IP tunneling may be necessary between different operators. Secondly, it is inherently insecure at the network layer; neither the source nor the destination can prove their authorities through their identities (ID part in IPv6 address). Thirdly, it opts to a large overhead, as described in 6LoWPAN [12], for example. Most sensors / tags may not be able to run an IP stack directly due to their cost and energy constraints, therefore, there is no need to stick on IPv6 for Things' network identities.

MobilityFirst proposes an innovative solution for the future Internet architecture. The core technique is the separation of naming and addressing. Every network object is assigned a global unique identification (GUID), which is independent of its addressing scheme (IP or

[1] Research Supported by NSF Future Internet Architecture (FIA) program under grant #CNS- 1040735

other sort) and locations (home, visit, disconnect, multi-home). Like current Internet, MobilityFirst is mainly designed for computers, with emphasis on mobile though. However, it provides a good foundation for IoT. This paper will discuss how MobilityFirst works for sensors and sensor data access / distribution.

The rest of this paper is organized as follows. In section II, we give the MobilityFirst background and architecture. In section III, we discuss the issues with existing M2M applications over Internet and provide a solution using MobilityFirst approach. In section IV, we present the protocol design of sensing data access / delivery. In section V. we present our prototype system setup and implementation. Finally, we give the conclusion and the future work.

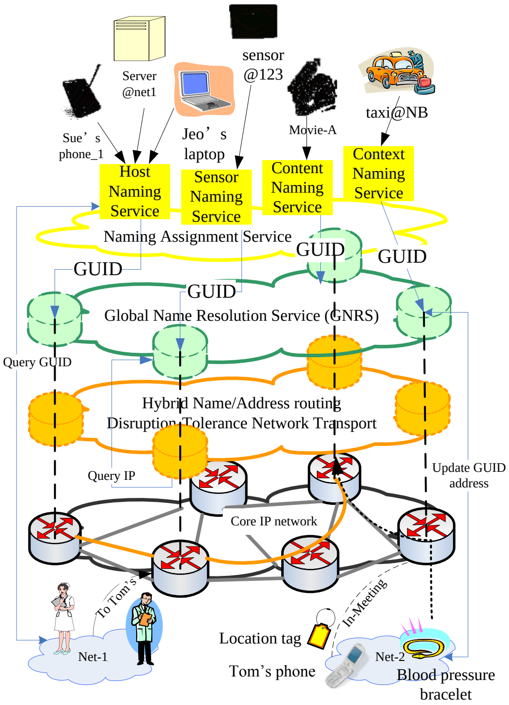

Figure 1. MobilityFirst Architecture

II. MobilityFirst Architecture

Today, the number of mobile phones is 4 times of the number of personal computers (4 Billions vs. 1 Billions) and according to a CISCO report [1], the mobile data will surpass fixed data over Internet by 2014. In addition, there will be 10 billions of sensors / tags by 2020 connecting to the Internet, representing the status of physical world.

Current Internet, designed 50 years ago, is used to connect fixed computers. MobilityFirst [13], a mobile centric future Internet architecture proposal, addresses the following fundamental issues for future Internet:

First, mobility becomes the norm of network entities. On the one hand, desktops will soon disappear in the households. On the other hand, standalone servers will be replaced by data centers – the computing cloud. A mobile terminal will have dynamic network access points, sometimes disconnected and sometimes connected to multiple networks simultaneously (multi-homing).

Second, network security becomes increasingly important as the current Internet may be paralyzed in case of cyber wars. Although the privacy and content protection can be handled at the application layer, the Internet protocol (IP) is inherently weak against address spoofing and denial of service (DoS) attacks. Security is especially important for sensors / M2M communication system. It must be an embedded component separated from the devices’ addresses [4].

Third, due to the mobility feature of future networking entities, the connectivity may not be available all the time. Traditional TCP/IP transport, designed for communication between fixed devices requiring end-to-end connectivity, cannot work efficiently in a mobile environment. A disruptive tolerant network (DTN) transport must be supported by future Internet routers.

Finally, future Internet is not only connecting computers but also connecting objects in other forms, such as sensors, content. It should provide a flexible way for the core network to route data from/to various contexts of connected objects. Apparently overloading IPv6 addresses is not a good way as we discussed earlier.

The same requirements for mobile devices on mobility, security and disruptive tolerant transport apply to the sensors or actuators in M2M applications.

Figure 1. shows the architecture of MobilityFirst network. On top of a networking layer (IP), there are three functional layers; each being described as follows:

A. Name Assignment Service (NAS)

The first principle of MobilityFirst is the *name and address separation*, i.e., giving every network object a global unique identification (GUID) independent of its network address, which can be dynamic (mobility), multiple (multi-homing) or not directly reacheable (e.g. sensors).

GUID:	Public Key	Seq number

Figure 2. A GUID structure

A GUID can be formed as [owner publicKey + sequence number] as shown in Figure 2. The owner of the network object can obtain a public / private key pair for his/her network objects and certify them an authority. Since there is a creation and management cost for PKI key pair, an owner may use one key pair for multiple network objects he owns with sequence numbers

Name assignment service (NAS) is an owner selected service to publish a GUID for network objects he owns. It maps a human readable semantic string, for example, “temperature of New York central park” to a GUID. The binding between the human readable string, such as keywords, and the GUID should be signed by the private key corresponding to the public key used by the GUID.

As shown in Figure 1, each network object, such as computer, mobile phone, sensor, content or context, can be assigned a GUID and published at NAS. People who intend to access a network object can lookup its GUID at NAS and then use the GUID to access it accordingly.

Since GUID contains a public key of the owner, it provides the ability to self verify its authentication and data integrity.

B. Global Name Resolution Service (GNRS)

A network object with a given GUID may be connected to the MobilityFirst core network at different network locations due to multi-homing, terminal mobility or network mobility. Global name resolution service (GNRS) is a distributed fast GUID to network address (e.g. IP) mapping. GNRS is similar to today's DNS service which maps a URL to an IP address, but it requires much faster responses at sub 100ms time scale. The key challenge of the MobilityFirst project is to implement the GNRS to meet this goal [14].

C. Hybrid Name and Address Routing

MobilityFirst supports hop-by-hop transport. Data is divided into chunks and each chunk, with a destination GUID, is delivered hop-by-hop in a storage aware core network[15]. A GUID might be resolved through GNRS at the source (early binding) or on the way towards the destination (late binding). The ability of the MobilityFirst network to route on GUIDs (names) makes it possible for data being distributed upon knowledge other than network addresses. This is especially useful in a disruptive tolerant network (DTN) transport when the destination is not at a known network location.

III. Sensors in MobilityFirst Networks

Internet of Things intends to connect all physical world objects and make them accessible by the applications in the cyberspace. However, most of real world objects are not able to communicate with each others. Wireless sensor network technology is the way to make things "intelligent" and interact with each others. For example, James' mug with an RFID tag indicate its location and the same mug with a temperature sensor integrated with the RFID tag can indicate not only on its location but also if there is hot coffee in the mug.

A. Unified Identity of Sensor Data

The design goal of MobilityFirst is to provide a unified network interface for sensors – every sensor connected to the MobilityFirst network can be accessed by a third party application over the Internet through a GUID.

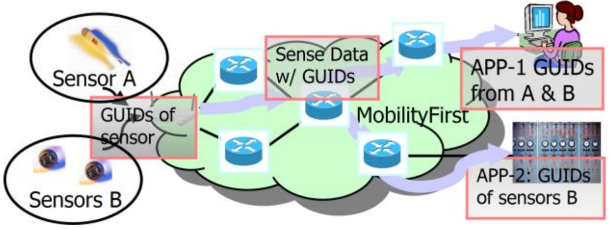

Figure 3. Accessing sensors through GUIDs

This implies that the sensor data from two sensors at different sensor networks are routed the same way by MobilityFirst network, just like IP packets are routed the same way in the Internet, regardless what application they belong to. As shown in Figure 3, sensor data from sensor A and sensors B are routed through MobilityFirst routers to APP-1 that uses both sensors A and B; and to APP-2 that uses only sensors B. In this case, sensor data from sensors B can be multicast toward APP-1 and APP-2.

B. M2M: from Vertical to Flat

As we have shown in Figure 1, GUIDs can be used to identify hosts, sensors, content as well as contexts. A sensor network may contain many layers from physical, data aggregation, processing, distribution and application layers. Figure 4 shows that depending on the sensor application requirements, sensors can be exposed to the MobilityFirst core network at different layers via GUIDs.

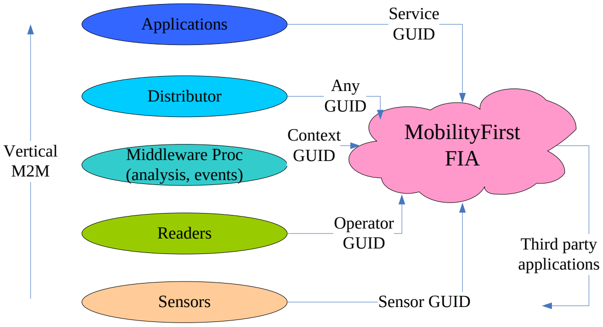

Figure 4. M2M application from vertical to flat

No matter at which layer the sensor data is pushed or pulled to the MobilityFirst network, the corresponding GUIDs are used to uniquely identify the data. From the MobilityFirst network point of view, the vertical sensor networks become transparent and flat to any third party M2M application, allowing an easy integration of sensor data at different levels and share the software/network resources in the value chain.

C. Inherent Security Feature

One of the main reasons of the isolation between traditional M2M applications is the security concern. A sensor network owner will not share his data unless there is mean for access control. On the other hand, a sensor application won't use a sensor data unless its integrity can be guaranteed. Both capabilities are only possible through an end-to-end secure session in the traditional Internet architecture, while MobilityFirst provides a way to self-certifying both source and sink in an open network because every GUID contains a public key.

IV. System Functions and Protocol Design

A. Sensor Application Scenario

Through one application scenario below, we describe the details of how sensors are integrated to MobilityFirst future Internet architecture.

Imagine there are a number of temperature sensors in New York central park at different locations. They can offer different kinds of temperature data at different time intervals. There could be thousands of applications that would like to use the sensor data in many different ways.

Some applications may want to have the raw data continuously reported at a fixed period. Some other applications may want to get temperatures from multiple sensors simultaneously.

B. Web Service API

Nowadays, if the owner of sensors wants to share the data and make them available to Internet applications, he needs to build a web application and provide a web service API for the sensors. The sensor data may be shared through a P2P network if the identities of sensors can be maintained. The sensor application needs to subscribe the P2P network for the data.

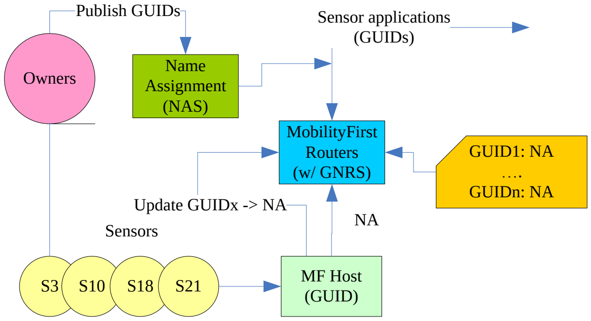

Figure 5. MobilityFirst system functions for sensors

C. MobilityFirst Approach

The MobilityFirst approach can offload the burden on the data owners and application developers to the core network. Figure 5 shows the system functional entities involved in the solution for the above scenario. The following steps are performed to deliver / access sensing data from the sensors to Internet (third party) applications:

1) Owners publish sensors (S3/S10/S18/S21) with GUIDs at a name assignment service (NAS). For example, “temperature at New York central park” maps to one of GUIDs, i.e. GUID_s. A schema of the data format of the sensor data identified by GUID_s can be obtained as well.

2) Each sensor network needs at least one MF (MobilityFirst) host program. It first opens an MF socket with GUID_s, performing an association of GUID_s to the MF core network. The edge MF router updates the GNRS with the mapping from GUID_s to the network address (NA, IP or other sort) of the MF host machine.

3) The MF host program can either wait for Internet applications to pull the data of GUID_s or simply push the data into the MF network. Since the MF core network supports hop-by-hop transport and data caching, the sensor data identified by GUID_s may be stored in the core network waiting for any application to pickup.

In this section, we will describe the primitives of the MF protocol and how it applies to sensor applications.

D. MobilityFirst Interfaces

The interfaces of MF networks are illustrated in Fig 6.

1) Su: Sensor-User Interface

From the network connectivity point of view, this is a proprietary interface for the wireless sensor network and the MF host is a gateway from one type of connectivity to another. However, from identity point of view, the MF host is transparent, which makes sensors directly visible by MF core network through GUIDs. Unlike in traditional M2M systems, application gateways are used to expose sensor data that is invisible by core network.

At this interface, MF protocol defines how sensors register their GUIDs at this interface regardless of the underlying network connections.

2) Mu: MobilityFirst user to network interface

An MF host runs MF protocols, which makes GUID associations, sends or gets data blocks to / from a network object identified by a GUID.

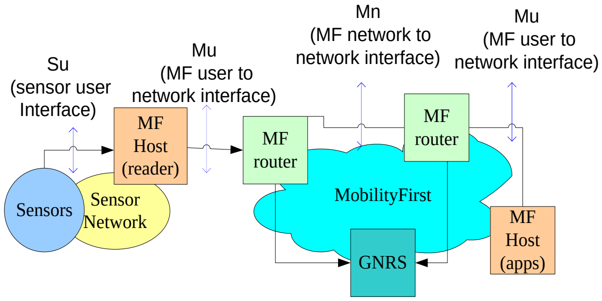

Figure 6. Sensor network interface to MobilityFirst

3) Mn: MobilityFirst network to network interface

It is the interface between MobilityFirst routers. It runs routing protocols, hop-by-hop transport and also GNRS lookup / update on behalf of MF hosts.

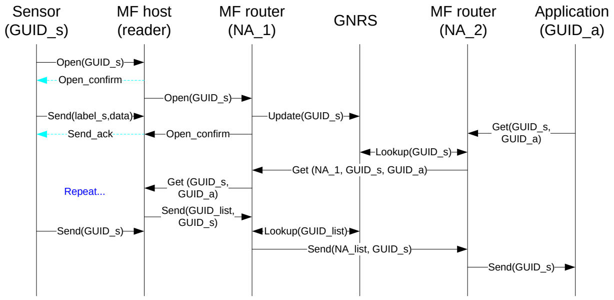

Figure 7. Signaling sequence of a sensor app scenario

Figure 7 gives the signaling sequence for a process of sensor data access/distribution as follows

1）A sensor connects to the MobilityFirst network and makes itself available for other applications.

a) *Open* (GUID_s) is the registration request from the sensor to Reader. It can be sent periodically to refresh reader. The *Open_confirm* (GUID_s, label_s) returns a shorten ID for the sensor to reduce the overhead. We note this function can be performed out of band if sensor network physical connectivity does not allow this process.

b) *Send* (label_s, data) is sending sensor data with a shortened ID. An MF host is able to maintain a table to map label_s to GUID_s.

c) *Open* (GUID_s) is the request to update GUID_s to GNRS through MF routers. Afterwards, GNRS will contains a mapping GUID_s –> NA.

2） The application accesses sensor data through the MobilityFirst network.

a) *Get* (GUID_s, GUID_a) is sent by GUID_a indicating the query for data of GUID_s.

b) MF routers send *Lookup* (GUID_s) to find the NA (network address) of GUID_s.

c) *Get* (NA, GUID_s, GUID_a) is sent by the edge MF router on behalf of the application (GUID_a).

d) *Get* (GUID_s, GUID_a) is sent by the destination MF router to the MF host that hosts sensor GUID_s.

3) The MF host distributes sensor data through the MobilityFirst network

a) *Send* (GUID_list, GUID_s) contains sensor data of GUID_s and transports toward multiple destinations in GUID_list (GUID_a is one on the list).

b) The MF router sends *Lookup* (GUID_list) to find the NA_list of GUID_list.

c) *Send* (NA_list, GUID_s) multicasts sensor data to a list of destinations.

V. Prototype System Setup

We have implemented a prototype of the MobilityFirst network architecture that enables network presence and reachability for sensors using GUIDs. Our prototype network consists of Click-based MobilityFirst routers that route GUID-addressed packets to network entities using a reliable hop-by-hop data transport. Apart from running a storage-aware control protocol to determine routes, each router also interacts with a GNRS server (optionally running on the router itself) to enable dynamic GUID-to-location binding for mobile entities. Network entities (i.e., MF hosts) run a corresponding protocol stack and API to allow applications to interact with the network. Currently, the host protocol stack and API have been implemented for both Linux and Android hosts and support access to Ethernet, WiFi and 4G-WiMAX networks. We have tested and validated our prototype implementation on the ORBIT[16] experimental platform that supports evaluation on a variety of network types up to a scale of 400 nodes.

A sensor network consisting of temperature sensors and readers with 2.4GHz wireless RFID interfaces is used. It will be integrated to a MF host program using the MF protocol stack and API implementations on a Linux host, where it can both register a sensor on the MF network and to relay messages from readers containing the sensor data. A typical Linux host we use for a MF host on the ORBIT platform has an Intel i7 2.93GHz processor, 3GB RAM, an Atheros WiFi (a/b/g) card and an Intel 6050 WiMAX/WiFi card. While large scale evaluations are pending, we believe this set up for a sensor network can quite easily support hundreds to a few thousand sensor nodes that periodically (on the order of seconds) publish data on to a MobilityFirst network. When such a host also handles requests directly from interested hosts, the architecture will need to be scaled by possibly distributing sensors among a number of reader hosts.

We plan to demonstrate two scenarios. One is an application queries the temperature of the “central park”, it chooses one GUID_s fit its criteria, then send the query to the network interface of GUID_s and get an instant temperature data. The other is an application send the query including its own GUID_a to subscribe the data of GUID_s, which is continuously sent to GUID_a periodically.

VI. Conclusions And Future Work

MobilityFirst can offer a solution for Internet of Things to effectively distribute sensor data over the core network with routing on GUIDs. It can also inherit the nice mobility and security features of GUID. More validations on the prototype system functionality are expected and the future work also includes further promote GUIDs on context identification and provide means to integrate various kinds of sensor network middleware to MobilityFirst core networks at different levels.

REFERENCES

[1] Cisco visual networking index: Global mobile data traffic forecast update, 2009-2014.

[2] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess and D. Savio, “Interacting with the SOA-Based Internet of Things”, IEEE Trans. On Services Computing, pp 223-235.

[3] Luigi Atzori, Antonio Iera, Giacomo Morabito, “The Internet of Things: A survey”, Computer Networks, May 2010.

[4] Geng Wu; Talwar, S.; Johnsson, K.; Himayat, N.; Johnson, K.D. “M2M: From mobile to embedded internet”, Communications Magazine, IEEE, April 2011; Volume: 49 Issue:4

[5] Babar, S.; Stango, A.; Prasad, N.; Sen, J.; Prasad, R. “Proposed embedded security framework for Internet of Things (IoT)”, Wireless VITAE, Feb. 2011, Chennai, India

[6] S. Deering and R. Hinden, “Internet Protocol, version 6 specifications” (RFC 2460), December 1998

[7] R. Hinden and S. Deearing “IP version 6 Addressing Architecture” (RFC 4291), February 2006

[8] Mike Botts, OpenGIS “Sensor Model Language (SensorML) Implementation Specification”, July 2007

[9] Steve Havens, OpenGIS “Transducer Markup Language (TransducerML) Implementation Specification”, May 2006

[10] GS1 Standard EPCglobal “Tag Data Standard v.1.6” Sept 9, 2011

[11] T.Berners-Lee at el. “Uniform Resource Identification (URI): Generic Syntax” (RFC 3986), January, 2005

[12] N. Kushalnagar at el., “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) : Overview, Assumptions, Problem Statement , and Goals” (RFC4919), August 2007

[13] MobilityFirst: NSF FIA project. http://www.nets-fia.net/

[14] T. Vu, A. Baid, Y. Zhang, T.D.Nguyen, J. Fukuyama, R.P. Martin and D. Raychaudhuri “DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet”, Technical Report WINLAB – TR-397, Fall, 2011

[15] S. Nelson, G. Bhanange and D. Raychaudhuri, “GSTAR: Generalized Storage-aware Routing for MobilityFirst in the Future Mobile Internet”, MobiArch’11, June, 2011

[16] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramach, H. Kremo, R. Siracusa, H. Liu, and M. Singh. “Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols”. In Proceedings of the WCNC, 2005.