Toward Better Horizontal Integration Among IoT Services

Several divergent application protocols have been proposed for IoT solutions. Each protocol focuses on a specific aspect of IoT communications. The lack of a protocol that can handle the vertical market requirements of IoT applications has resulted in a fragmented market between many protocols. In turn, this fragmentation is a main hindrance in the development of new services that require the integration of multiple IoT services to unlock new capabilities and provide horizontal integration among services.

Ala Al-Fuqaha, Abdallah Khreishah, Mohsen Guizani, Ammar Rayes, and Mehdi Mohammadi

Abstract

Several divergent application protocols have been proposed for Internet of Things (IoT) solutions including CoAP, REST, XMPP, AMQP, MQTT, DDS, and others. Each protocol focuses on a specific aspect of IoT communications. The lack of a protocol that can handle the vertical market requirements of IoT applications including machine-to-machine, machine-to-server, and server-to-server communications has resulted in a fragmented market between many protocols. In turn, this fragmentation is a main hindrance in the development of new services that require the integration of multiple IoT services to unlock new capabilities and provide horizontal integration among services.

In this work, after articulating the major shortcomings of the current IoT protocols, we outline a rule-based intelligent gateway that bridges the gap between existing IoT protocols to enable the efficient integration of horizontal IoT services. While this intelligent gateway enhances the gloomy picture of protocol fragmentation in the context of IoT, it does not address the root cause of this fragmentation, which lies in the inability of the current protocols to offer a wide range of QoS guarantees. To offer a solution that stems the root cause of this protocol fragmentation issue, we propose a generic IoT protocol that is flexible enough to address the IoT vertical market requirements. In this regard, we enhance the baseline MQTT protocol by allowing it to support rich QoS features by exploiting a mix of IP multicasting, intelligent broker queuing management, and traffic analytics techniques. Our initial evaluation of the lightweight enhanced MQTT protocol reveals significant improvement over the baseline protocol in terms of the delay performance.

Introduction

IoT devices can be classified into two major categories: resource-constrained devices and resource-rich devices. We define resource-rich devices as those that have the hardware and software capabilities to support the TCP/IP protocol suite. For devices that support the TCP/IP protocol suite, IoT applications are implemented on top of a variety of application-level protocols and frameworks, including the Constrained Application Protocol (CoAP), Representational State Transfer (REST), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), Message Queue Telemetry Transport (MQTT), MQTT for Sensor Networks (MQTT-SN), Data Distribution Service (DDS)[1], and others. On the other hand, devices that do not have the required resources to support TCP/IP cannot interoperate easily with resource-rich devices that support the TCP/IP suite. For example, microcontroller-based appliances and gadgets should have the capability to interoperate with other IoT elements that are TCP/IP enabled. Beyond the interoperability issues between devices that support TCP/IP and those that do not, IoT devices utilize a variety of protocols leading to a myriad of interoperability issues that limit the potential applications of the IoT. This fragmentation between the protocols utilized for communication within and across resource-constrained devices and resource-rich devices is not foreseen to change in the near future.

Furthermore, the communication patterns between the different entities that comprise the IoT can be classified into three main categories based on the nature of the communicating devices: machine-to-machine (M2M), machine-to-server (M2S), and server-to-server (S2S). M2M communications between IoT devices that can be resource-constrained or resource-rich typically require timely data delivery. Depending on the application, the response time might be required to be as low as 10 µs. DDS is an object management group (OMG) data-centric middleware standard for real-time M2M communication. DDS utilizes a broker-less architecture that relies on multicasting to offer various quality of service (QoS) guarantee and reliability control capabilities even in scenarios that involve massive fan-outs (i.e. a large number of destinations for a given message). On the other hand, M2S and S2S communication does not typically require the stringent delay performance needed for M2M communication. Therefore, IoT typically employs broker-based protocols like MQTT and AMQP to offload the IoT devices from having to handle a large number of upstream server requests. Moreover, M2S and S2S protocols, like MQTT, offer simplistic QoS services in the form of at-least-once, at-most-once, and exactly-once deliveries.

There are many middleware proposals in the literature that address the interoperation of the IoT protocols. The common design patterns used in these protocols are: publish/subscribe oriented, service oriented [1], virtual machine based [2], and software defined network/radio (SDN/SDR) based designs [3, 4].

MQTT-SN[2], which is the extension to MQTT for wireless sensor networks, uses the publish/subscribe model in the same way as in the baseline MQTT protocol. A main component of this

Ala Al-Fuqaha and Mehdi Mohammadi are with Western Michigan University.

Abdallah Khreishah is with the New Jersey Institute of Technology.

Mohsen Guizani is with the University of Idaho.

Ammar Rayes is with Cisco Systems.

[1] http://www.omg.org/spec/DDS/1.2/

[2] http://mqtt.org/documentation 3 http://ict-icsi.eu/

architecture is the MQTT-SN gateway. Its main task is to translate MQTT and MQTT-SN messages between MQTT-SN clients and MQTT brokers. One main drawback of this model is that even the simple levels of QoS presented by MQTT are not supported. Things Broker [1] is a service oriented gateway that provides a RESTFul interface to smart objects by a set of Twitter-based abstractions and communication models. In [2], instead of gateways, the authors propose a solution to integrate smart resource-constrained objects into the Internet using virtual networks. This work can provide end-to-end communication between devices, but scalability and binding to specific protocols are the main challenges. The other method is the combination of the publish/subscribe model and the virtual machine-based design that has been used in the ICSI[3] project to implement a flexible IoT middleware for smart city environments.

Software-defined networking (SDN) and software-defined radio (SDR) have drawn the attention of researchers toward using these brand new technologies to cover different IoT communications. For example, an approach based on SDN was proposed for IoT tasks in [3]. In their approach, the authors developed a middleware with a layered IoT SDN controller to manage dynamic and heterogeneous multi-network environments. As another example, the work in [4] used SDR technology to build a communications infrastructure for IoT applications.

There are some attempts to extend CoAP targeting QoS features and conditional observation. Among them, in [5] the authors proposed an extension to the CoAP protocol to support QoS for timeliness based on delivery priorities. In this work an observer can request a level of priority that establishes the order of notification transmission. In conditional observation, a notification condition can be determined by the clients. An extension to XMPP was presented in [6] in which XMPP was improved by a topic based filter for enhanced performance. Figure 1 illustrates the recent research work toward the IoT gateways as well as enhancements to the current application protocols. The state of our proposed approach is highlighted so that it uses the publish/subscribe pattern in conjunction with DDS (multicasting) and MQTT (intelligent queue management) to provide flexible QoS features and to bridge the gap between the different protocols.

The dissimilar QoS constraints for M2M and M2S/S2S communication and the failure of the current protocols to cover a wide range of QoS requirements is the primary culprit behind protocol fragmentation in the IoT. This gloomy picture of protocol fragmentation and interoperation between IoT devices calls for an intelligent protocol that is capable of meeting the mentioned challenges and an intelligent protocol gateway that is capable of bridging the gap between different technologies. There are other protocols to address the QoS requirements in the Internet (e.g. SNMP), but these protocols are too heavy to be used in resource-constrained devices and are rarely used within the context of IoT. Moreover, the QoS features offered by other application protocols such as XMPP, CoAP, and AMQP are not broader than MQTT. Therefore, MQTT as a light protocol is a good candidate to be extended to address the aforementioned challenges. In addition, the Enhanced MQTT protocol can be used in the fog layer of IoT environments and in collaboration with other gateways to support a full range of QoS requirements.

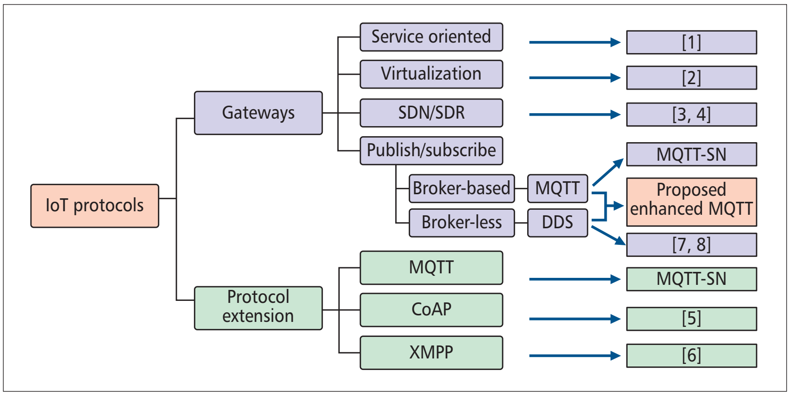

Figure 1. IoT gateway taxonomy, recent related literature and research position.

Although MQTT is considered to be a lightweight protocol, our extension is not going to make it heavy since our extension relies on multicasting to deliver M2M traffic and intelligent queue management for M2S/S2S communications. Therefore, our extension modifies MQTT while being backward compatible with the baseline protocol and without negatively impacting its clean and simple architecture.

In the next section we elaborate on the proposed intelligent IoT protocol gateway. Then in the next section we discuss the motivations and the details of our proposed approach for Enhanced MQTT relative to the current state-of-the-art research in the field.

Intelligent IoT Protocol Gateway

State of the Art

IoT applications bring about some horizontal integration situations in which a conversion between protocols is required. These situations usually happen when smart devices are talking to each other using different communication technologies (like Bluetooth and ZigBee) or when they are using different application protocols (like MQTT and CoAP). Several attempts have been made in the recent literature to address this issue. Paramount among these attempts is Ponte, which was initially developed as QEST and currently is under the Eclipse IoT project.[4] Ponte offers uniform open APIs for the programmer to enable the automatic conversion between the various IoT application protocols such as CoAP and MQTT.

In addition to Ponte, the Eclipse IoT Project includes other sub-projects that facilitate IoT application development such as Kura, Eclipse SCADA, Eclipse SmartHome, and Krikkit. Kura provides a Java/OSGi-based container for M2M applications running in service gateways and facilitates I/O access, data services, watchdog, network configuration, and remote management for M2M applications. Eclipse SCADA's focus is

[3] http://ict-icsi.eu/

[4] https://projects.eclipse.org/projects/iot

	Gateways	Application protocols					Management		Connectivity			
		RESTful HTTP	COAP	MQTT	XMPP	DDS	OMA-DM	BBF TR-069	Cellular	Zigbee	Bluetooth	WiFi
Practice	Ponte	✓	✓	✓								
	oneM2M	✓	✓	✓			✓	✓	✓	✓	✓	✓
	SmartM2M	✓	✓				✓		✓	✓	✓	✓
	Intel IoT Platform			✓			✓	✓	✓	✓	✓	✓
	LWM2M								✓	✓		✓
Research	IoT communication gateway [9]								✓	✓		
	IoT gateway centric model [10]	✓								✓		✓
	HTTP-CoAP crossprotocol proxy [11]	✓	✓									
	Semantic gateway [12]		✓	✓	✓							
	Proposed enhanced MQTT	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓

Table 1. IoT gateways and their supported protocols.

to provide a common communication mechanism as well as post-processing and visualization of the data. Eclipse SmartHome aims to build a heterogeneous environment to integrate different protocols and standards and to bring a uniform access while supporting different kinds of interactions. The Krikkit architecture uses the publish/subscribe model by which data acquiring is possible by registering rules or policies on edge devices like sensor gateways.

An ongoing standard specification, oneM2M[5], aims to provide a common M2M/IoT service platform that supports secure, reliable, and efficient operations of M2M/IoT services. oneM2M tries to leverage existing specifications and bridge the gap between the current M2M service layer protocols. oneM2M utilizes REST to represent and manage smart things while using the common application protocols including HTTP, CoAP, OMA-DM, and MQTT to facilitate interworking with non-oneM2M systems. For device management, oneM2M relies on the OMA-DM and BBF TR-069 specifications. In order to have a heterogeneous environment for different access technologies like ZigBee, BACnet, or Bluetooth, oneM2M has started to provide device abstraction and semantic interworking.

The SmartM2M standard is developed by ETSI to provide a horizontal M2M service platform for developing network-independent services and deploying vertical applications.

Intel also announced the launch of Intel IoT[6] middleware, connectivity, and security components to simplify deploying IoT applications. The Intel IoT gateway enables connectivity up to the cloud and down to the smart devices. In terms of connectivity and communications, Intel IoT Platform supports ZigBee, Cellular 2G/3G/4G, Bluetooth, Serial, USB, VPN, Wi-Fi, and MQTT. Its device management module consists of OMA-DM, BBF TR-069, and web-based configuration interfaces.

In addition to the aforementioned projects, academic research projects and proposals have presented partial solutions or addressed a specific application to the problem. They may also be limited to specific hardware devices or rely on resource-rich TCP/IP based devices. For example, the gateway that is proposed in [9] supports the conversion between ZigBee and GPRS protocols in wireless sensor networks and mobile communication networks. The Light-weight M2M[7] (LWM2M) device management protocol has been developed to ease unified remote device management. This protocol is also limited to those devices that support IP.

In [10] the authors proposed a wireless gateway by which mobile clients are able to interact with M2M devices. Their proposed gateway just supports REST queries in the application layer and provides dynamic device discovery, connection management of non-smart things, and binding metadata to sensor and actuator measurements. Other conversions like HTTP to CoAP [11] are also presented in the literature. In [12] the authors proposed a gateway and Semantic Web enabled IoT architecture to bring the interoperability of messaging protocols such as XMPP, CoAP, and MQTT using a multi-protocol proxy. Table 1 compares recent implementations of IoT gateways and their coverage of different IoT protocols.

The anecdotal data that we have obtained so far about the diverse needs of IoT applications and the capabilities and restrictions of the underlying hardware reveals that the existing IoT gateway solutions like Ponte or Intel IoT Platform are not sufficiently extensive to bridge the gap between the different IoT protocols, and therefore a more intelligent and comprehensive solu-

[5] http://www.onem2m.org

[6] http://www.intel.com/content/www/us/en/internet.of-things

[7] http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2mobject-registry

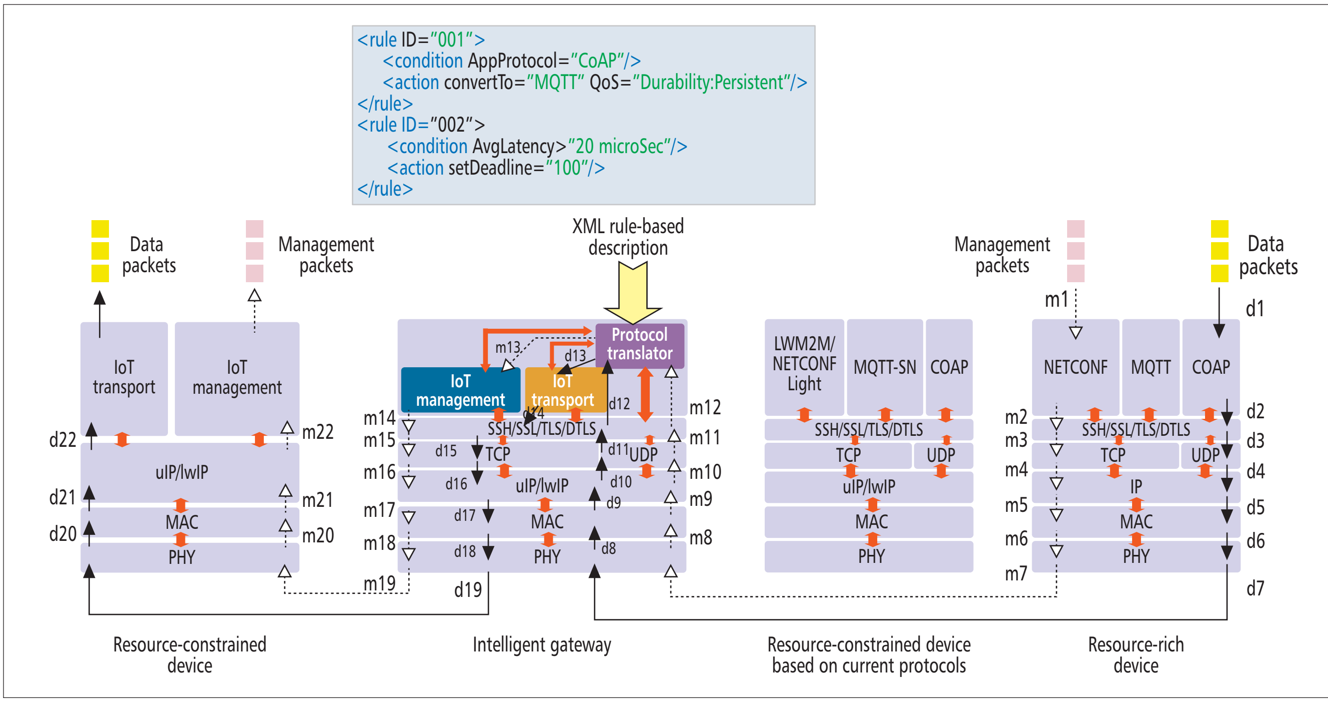

Figure 2. The architecture of a rule-based Intelligent IoT gateway.

tion is needed. For example, Ponte's feature that automatically converts between a number of protocols comes at a price due to longer packets and verbose communication. Moreover, as with many other protocol gateways, Ponte assumes that the underlying devices rely on TCP/IP. Intel IoT Platform, although supporting a wide range of transmission protocols, currently does not provide full conversion capabilities between application protocols.

Functionalities of the Intelligent Gateway

Approaches like that followed by Ponte assist programmers to write a single version of the same application while supporting different protocols, but the programmers have no control over the underlying wire-protocol, and consequently the degradation of performance and efficiency will be uncontrolled. A more important drawback is that resource-constrained devices have no chance to be integrated with other objects in this solution, bringing to mind that they are considered to be "second-class citizens." Therefore, we are motivated by the following three main observations to illustrate the need for a new intelligent IoT gateway.

- IoT gateways should include mechanisms by which programmers are allowed to control the wire protocol. Application-agnostic message passing for resource-constrained devices causes unnecessary packet exchanges. In such environments, an intelligent gateway provides facilities for programmers to optimize the performance by controlling the underlying wire protocol based on the specific application.
- Resource-constrained devices should be treated the same as resource-rich devices. An intelligent gateway should support true interoperability between these two types of IoT devices.
- The introduction of an intelligent protocol gateway into the IoT has the potential to open up new opportunities to reduce the market fragmentation between IoT protocols.

The aforementioned observations provide strong evidence that an intelligent IoT gateway is essential to IoT development so that it offers "smart" services and it is deeply re-programmable through a high-level rule-based language written by the programmer. It is remarkable that our proposal for deeper re-programmability of the IoT gateway through a rule-based language not only does not conflict with the current interoperability and management standardization efforts, but also complements the existing protocols like IEEE 1905.1, LWM2M, and NETCONF Light. While the intelligent gateway can utilize default templates of rules out-of-the-box for convenience, the deep rule-based programmability of the gateway allows application developers to exploit application-specific communication patterns to achieve more efficient protocol translations. Therefore, application developers have to write the application protocol translation rules once in return for more efficient protocol translations.

The overall concept of the proposed gateway is demonstrated in Fig. 2. Here, the protocol stack that is installed on resource-constrained devices relies on the current technologies pertaining to resource-constrained devices that utilize the intelligent gateway. The flow sequence of data packets and the management packets are depicted in the figure by labels d_i and m_i, respectively. In the gateway, the logic of the rules is applied to the stream of data and management packets (i.e. d12 and m12 in Figure 2) leading to generate appropriate data and management packets (i.e. d13 and m13 in Figure 2). Autonomic management and data aggregation services are also possible by feeding rules to the

Research	Pub/sub	QoS-enabled	Application domain
QoS-aware platform [7]	✓	✓	WSN (healthcare)
Sensor network pub/sub protocol [8]	✓	×	WSN
Decentralized pub/sub model [13]	✓	✓	Peer-to-peer
Real-time pub/sub service [14]	✓	✓	Real-time applications
DDS/SDN based communication [15]	✓	×	Real-time applications

Table 2. Summary of recent research utilizing the DDS protocol.

gateway while the gateway can generate and transmit new data and management packets from the received ones. Inside the gateway, encapsulation and decapsulation of data packets are carried out by a lightweight protocol named "IoT transport," while "IoT Management," another lightweight protocol, does the same for management packets.

The light protocol stack presented by the intelligent gateway relies only on uIP/lwIP protocols without the need for other transportation and security protocols on the resource-constrained device. Security services can also be delegated to the gateway so that confidentiality, authenticity, and integrity traffic can be exchanged with the gateway and not directly with resource-constrained devices.

By supporting rule-based language, the programmer will have to describe the transformations in a high-level language by which the gateway should perform the required conversion. The advantage of this approach comes from the flexibility of the gateway by which a programmer can examine and fulfill wire protocol optimizations at the cost of describing required transformations in a high-level domain-specific language. The programmer can also use standard transformation templates instead of specifying application-specific rules. In this case, the wire protocol messaging will be less efficient but yet will be similar to that of Ponte.

The proposed intelligent gateway can also be used for localized autonomic management of the IoT devices automatically. Considering a real deployment scenario consisting of thousands or millions of IoT nodes, it is vital to have self-management fault, configuration, accounting, performance, and security (FCAPS) capabilities. Moreover, a deeply re-programmable gateway can also provide the option to accomplish data and flow aggregation and consequently reduce the number of flows handled by the network elements and to achieve high performance. As another benefit, the existence of several gateways within the IoT can also be utilized for load balancing of the potentially massive IoT traffic among the available gateways.

Enhanced MQTT

As discussed above, DDS is a quintessential IoT protocol that employs a broker-less publish/subscribe architecture to meet the stringent real-time constraints for M2M communications. Nevertheless, it is only utilized in academic labs and, to the best of our knowledge, has not been used in IoT released products. In the following subsections we review recent implementations of DDS, followed by our proposal for the Enhanced MQTT and its motivations and opportunities.

DDS in the Literature

In recent years DDS has drawn the attention of many IoT researchers to design and develop more reliable and real-time IoT systems. Most of the studies focus on the QoS features of the DDS protocol [7, 13, 14]. In [13] a decentralized content-based publish/subscribe model based on DDS is introduced in which a peer-to-peer communication is guaranteed to have individual QoS requirements based on the bandwidth constraints. Exploiting the DDS protocol for publish/subscribe communication to support various QoS levels for real-time applications has been proposed in [14] as well. The Sensor Network Publish-Subscribe protocol (SNPS) [8] has been proposed to use the DDS concepts in WSNs, but it does not have the complete QoS features offered by DDS. Achieving QoS and reliability requirements in WSNs for healthcare environments has been investigated in [7] in which DDS has been used as the main building block. The feasibility of managing DDS communications dynamically using SDN is described in [15].

The common drawback of these research activities is that they do not support all types of IoT communications (i.e. M2M, M2S, and S2S). Specifically, they assume a real-time environment for their system, while real-time response is not necessary everywhere. Moreover, the current proposals do not provide sufficient QoS support. The characteristics of these attempts that are leveraging DDS in their core have been summarized in Table 2.

Broker-based architectures like MQTT and AMQP can offload devices from having to handle a large number of upstream server requests. However, this relief comes with the cost of losing quality of service and reliability, which are vital in most IoT applications.

Enhanced MQTT: The Big Picture

In this research we propose to revisit the MQTT protocol with the aim of providing it with a hybrid architecture that would allow it to operate in various M2M as well as M2S and S2S scenarios. Therefore, we provide a solution that stems the root cause of the protocol fragmentation issue encountered within the context of the IoT. A horizontal integrated IoT framework needs to handle a wide range of different communication and application protocols and to facilitate their interworking. To realize such a framework, we target the MQTT protocol by improving it to support rich QoS features, intelligent broker queue management, and traffic analytics techniques. The hybrid architecture will allow the protocol to seamlessly utilize direct and broker-less multicast communication for M2M communication while utilizing the broker for M2S and S2S communications. This hybrid architecture would allow MQTT to extend its role in the IoT to handle M2M communications and would allow for multiple MQTT brokers to cooperate to deliver better QoS and reliability capabilities. Beyond this architectural change to MQTT, we believe that the QoS features currently offered by MQTT are very limited and there is a need to extend these features significantly to support various development and deployment scenarios. Also, MQTT does not offer a management interface that allows for the prioritization, preemption, and collection of analytics on the IoT traffic going through the broker. A full set of traffic analysis services is to be supported by the broker. The proposed architecture considers a *subscribers table* and a *management table* inside each broker. Brokers are registered with the desired topics in the subscribers table. The management table tracks the statistics of each topic in terms of QoS and reliability metrics. Through these extensions, our proposed Enhanced MQTT remains backward compatible with MQTT.

Figure 3 depicts the proposed new MQTT capabilities. These capabilities can be summarized as follows.

• Allow for broker-less multicast communication in support of M2M communication (cf. interaction d on Fig. 3). The broker-less multicast communication will allow MQTT to be used for M2M communication as well and to enhance its delay and fan-out performance.

• Allow multiple brokers to receive multicast communication from IoT devices in support of reliability (cf. interaction d on Fig. 3). The multicast communication will allow multiple MQTT brokers to listen to the device generated traffic allowing for failure recovery in the event of broker failures.

• Allow brokers to move subscribers to other brokers in support of QoS (cf. interaction c on Fig. 3). This will allow for QoS features that are beyond the naïve QoS features currently offered by MQTT.

• Allow brokers to re-prioritize (i.e. preempt) the distribution of MQTT packets in support of QoS (cf. interaction a on Fig. 3). This will enable the inclusion of rich QoS features even in deployments that do not involve multiple brokers.

• Allow subscribers to register for and obtain analytics on the IoT traffic going through the broker (cf. interaction d on Fig. 3). By far this is the most important feature that is enabled by our proposed approach, as it would allow for a new class of IoT applications that are otherwise infeasible using current protocols such as MQTT. The new class of applications enabled by our proposed protocol allows the IoT infrastructure itself to evolve based on collected analytics. For example, the collected analytics can be used as the input to a mathematical formulation or a reenforcement learning strategy that would reconfigure the number of brokers and their locations to obtain better QoS support for the given application. Therefore, the application designers do not have to worry about the optimal deployment for the given application offline; instead they can have their application monitor its own performance and consequently evolve its configuration.

We have performed a simulation experiment to compare the performance of our proposed model to the baseline MQTT model in terms of queuing delay. We compared the queue time only since it is a bottleneck in broker-based approaches, and our proposal is an attempt to resolve this issue while leaving all the other parameters intact (i.e. processing, transmission, and propagation delays). Figure 4 illustrates the results of this evaluation. In our model, M2M and M2S traffic streams are separated to be handled differently. M2M traffic is directly delivered to the destination using multicasting, while M2S traffic goes through a queuing system with four different levels of priorities. Figure 4 reveals that the baseline MQTT has resulted in more queueing delays compared to our proposed model, even though the service rate was increased by 5 percent and 10 percent, respectively, compared to the service rate used in our proposed model. On the other hand, the proposed model shows a decrease in the queueing time when the fraction of M2M traffic constitutes a higher percentage of the overall IoT traffic.

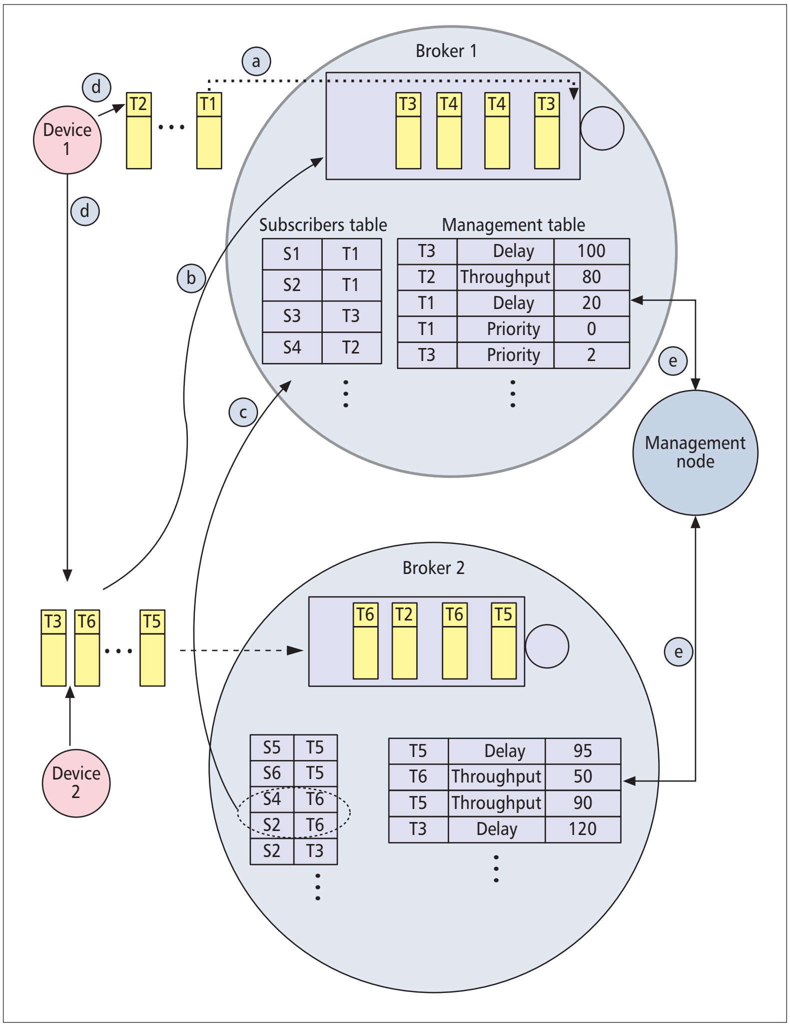

Figure 3. Enhancements of broker functions.

Motivation and Scenarios

The proposed QoS and reliability capabilities of MQTT can be utilized in a variety of IoT applications. As a matter of fact, our proposal was initially motivated by an actual need that appeared in an interdisciplinary study that we are involved with to alleviate some of the feed-

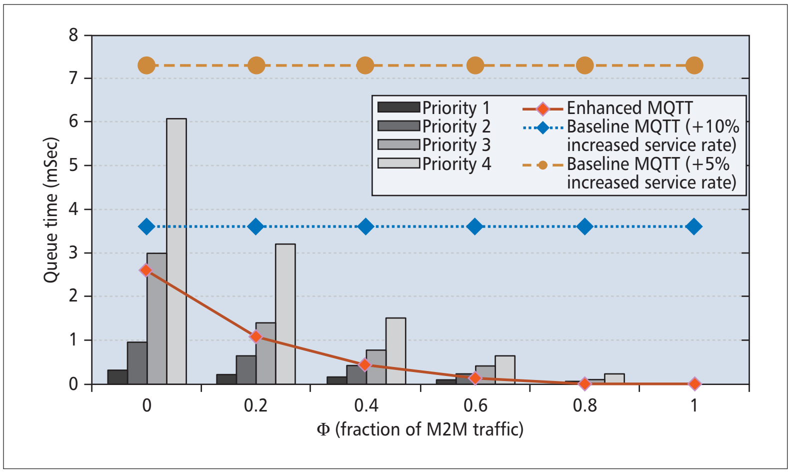

Figure 4. The Enhanced MQTT protocol compared to the baseline MQTT protocol.

ing difficulties experienced by adults with essential tremors, Parkinson's disease, and Alzheimer's disease. As an example of M2M communication with stringent delay requirements, in order to allow adults with essential tremors or Parkinson's disease to eat without spilling food, a glove can be equipped with tiny vibrating motors to counteract the hand movement instability measured by the accelerometers. In this application, the accelerometer sensors and vibrating motors have to communicate with the minimum delay possible to deliver the required functionality.

The monitoring of the patients' vital signs in a nursing home environment provides a quintessential scenario that requires brokers to deliver collected M2M data efficiently and reliably to different servers (i.e. M2S communication with stringent delay requirements). In this scenario, the vital sign measurements must be delivered to multiple nursing stations with minimum delay for accurate visualization and actuation of smart infusion pumps and other bed-size medical equipment even in the presence of broker failures. Another interesting scenario within a nursing home environment includes the use of accelerometers to monitor the eating behavior of Alzheimer's patients and activating audio-visual cues recorded by a close relative through a wearable device. These cues aim to stimulate the patient to eat better, exploiting the fact that most Alzheimer's patients continue to remember close relatives during the different stages of the disease.

Beyond a nursing home environment, a scenario that can benefit from the proposed enhancements includes exchanging time difference of arrival (TDoA) measurements between embedded devices deployed within the infrastructure, and wearable devices to provide precise localization details to blind and visually impaired people to navigate an indoor environment. In this scenario, the wearable device can also communicate with minimum delay to obtain the best estimated location and provide tactile cues to the patient to navigate to their destination through querying the meta-data associated with the location fingerprints.

Conclusion and Future Directions

In this article the restrictions and shortages of the current application protocols involved in IoT systems have been elaborated. We then identified the major driving forces for progression toward an intelligent IoT gateway to overcome the fragmentation and enable the efficient integration of horizontal IoT services. The opportunities that will be made possible by this evolution have been discussed as well.

To offer a solution that stems the root cause of this protocol fragmentation issue, an enhanced version of the MQTT protocol is proposed in this article that alleviates the deficiency of the existing MQTT protocol, especially in support of QoS and reliability. The enhanced protocol is strengthened by an intelligent queuing mechanism and real-time multicasting so that it is able to handle the different forms of communications required by IoT applications including M2M, M2S, and S2S. The performance of the current work is promising to be better than existing published results. Furthermore, the proposed protocol will contribute to minimizing the myriad of protocol options that application developers have to evaluate before developing IoT services and applications.

There are many potential future evaluation studies of the proposed Enhanced MQTT. In our future work we plan to evaluate our proposed extension in a wide range of deployment, emulation, and simulation scenarios. The application of the Enhanced MQTT model to implement real-time IoT systems is another potential direction for this research. Investigating other IoT challenges like security, scalability, availability, and management are also in our future research plan.

References

[1] R. A. Perez de Almeida *et al.*, "Thing Broker: A Twitter for Things," *Proc. 2013 ACM Conf. Pervasive and Ubiquitous Computing Adjunct Publication*, Zurich, Switzerland, 2013, pp. 1545–54.

[2] I. Ishaq *et al.*, "Internet of Things Virtual Networks: Bringing Network Virtualization to Resource-Constrained Devices," *2012 IEEE Int'l. Conf. Green Computing and Commun. (GreenCom)*, 2012, pp. 293–300.

[3] Z. Qin *et al.*, "A Software Defined Networking Architecture for the Internet-of-Things," *2014 IEEE Network Operations and Management Symp. (NOMS)*, 2014, pp. 1–9.

[4] Y. H. Lin *et al.*, "Wireless IoT Platform Based on SDR Technology," *2013 IEEE and Internet of Things (iThings/CPSCom) Green Computing and Commun. (GreenCom), IEEE Int'l. Conf. and IEEE Cyber, Physical and Social Computing*, 2013, pp. 2245–46.

[5] A. Ludovici *et al.*, "Adding QoS Support for Timeliness to the Observe Extension of CoAP," *2012 IEEE 8th Int'l. Conf. Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2012, pp. 195–202.

[6] R. Klauck and M. Kirsche, "Chatty Things Making the Internet of Things Readily Usable for the Masses with XMPP," *2012 8th Int'l. Conf. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2012, pp. 60–69.

[7] A. Agirre *et al.*, "QoS Aware Platform for Dependable Sensory Environments," *2014 IEEE Int'l. Conf. Multimedia and Expo Wksps. (ICMEW)*, 2014, pp. 1–5.

[8] K. Beckmann and M. Thoss, "A Wireless Sensor Network Protocol for the OMG Data Distribution Service," *2012 Proc. 10th Wksp. Intelligent Solutions in Embedded Systems (WISES)*, 2012, pp. 45–50.

[9] Q. Zhu *et al.*, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things," *2010 IEEE/IFIP 8th Int'l. Conf. on Embedded and Ubiquitous Computing (EUC)*, 2010, pp. 347–52.

[10] S. K. Datta, C. Bonnet, and N. Nikaein, "An IoT Gateway Centric Architecture to Provide Novel M2M Services," *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 514–19.

[11] A. Castellani, T. Fossati, and S. Loreto, "HTTP-CoAP Cross Protocol Proxy: An Implementation Viewpoint," *2012 IEEE 9th Int'l. Conf. Mobile Adhoc and Sensor Systems (MASS)*, 2012, pp. 1–6.

[12] P. Desai, A. P. Sheth, and P. Anantharam, "Semantic Gateway as a Service Architecture for IoT Interoperability," *CoRR*, vol. abs/1410.4977, 2014.
[13] M. A. Tariq *et al.*, "Meeting Subscriber-Defined QoS Constraints in Publish/Subscribe Systems," *Concurrency and Computation: Practice and Experience*, vol. 23, 2011, pp. 2140–53.
[14] X. Lu *et al.*, "A Novel QoS-Enabled Real-Time Publish-Subscribe Service," *Int'l. Symp. Parallel and Distributed Processing with Applications*, 2008, *ISPA '08*, pp. 19–26.
[15] L. Bertaux *et al.*, "A DDS/SDN Based Communication System for Efficient Support of Dynamic Distributed Real-Time Applications," *2014 IEEE/ACM 18th Int'l. Symp. Distributed Simulation and Real Time Applications (DS-RT)*, 2014, pp. 77–84.

Biographies

Ala Al-Fuqaha [S'00, M'04, SM'09] (Ala.Al-Fuqaha@wmich.edu) is an associate professor and director of the NEST Research Lab in the Computer Science Department at Western Michigan University. His research interests fall in the areas of vehicular networks, dynamic spectrum access etiquettes in cognitive radio networks, smart services in support of the Internet of Things, and planning of software defined networks (SDN). He is currently serving on the editorial board of multiple journals.

Abdallah Khreishah [M] (abdallah@njit.edu) is an assistant professor in the Department of ECE at the New Jersey Institute of Technology. His research interests are in the areas of visible-light communication, green networking, wireless networks, and network securitys. H received his Ph.D. degree in electrical and computer engineering from Purdue University in 2010. While pursuing his Ph.D. studies, he worked with NEESCOM. He is the chair of the North Jersey IEEE EMBS chapter.

Mohsen Guizani [S'85, M'89, SM'99, F'09] (mguizani@ieee.org) is a professor and chair of the ECE Department at the University of Idaho. His research interests include wireless communications and mobile computing, computer networks, cloud computing, cyber security, and smart grid. He currently serves on the editorial boards of several international technical journals. He is the author of nine books and more than 400 publications. He is a senior member of ACM.

Ammar Rayes (rayes@cisco.com), Ph.D., is a distinguished engineer at the CTO Office of Cisco Services Technology Group focusing on IoT, analytics and smart services. He is the Founding President of the International Society of Service Innovation Professionals. Ammar has authored more than 100 papers and patents on advances in telecommunications-related technologies. He is the editor-in-chief of the journal *Advances of Internet of Things*, and he has served as an associate editor of *ACM Transactions on Internet Technology* and the *Journal of Wireless Communications and Mobile Computing*.

Mehdi Mohammadi [S'14] (Mehdi.Mohammadi@wmich.edu) is a Ph.D. student in the Department of Computer Science, Western Michigan University (WMU), Kalamazoo, MI, USA. His research interests include Internet of Things, Future Internet, software-defined networking, data mining, and natural language processing. He served as a reviewer for the Wiley journal *Security and Wireless Communication Networks*. He is a recipient of the Graduate Doctoral Assistantship from the WMU Libraries, as well as four travel grants from the National Science Foundation.