



A Data-Oriented M2M Messaging Mechanism for Industrial IoT Applications

DOI:

[10.1109/JIOT.2016.2646375](https://doi.org/10.1109/JIOT.2016.2646375)

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Meng, Z., Wu, Z., Muvianto, C., & Gray, J. (2017). A Data-Oriented M2M Messaging Mechanism for Industrial IoT Applications. *IEEE Internet of Things Journal*, 4(1), 236 - 246. <https://doi.org/10.1109/JIOT.2016.2646375>

Published in:

IEEE Internet of Things Journal

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



A Data-Oriented M2M Messaging Mechanism for Industrial IoT Applications

Zhaozong Meng, *Member, IEEE*, Zhipeng Wu, *Senior Member, IEEE*, Cahyo Muvianto, and John Gray

Abstract—Machine-to-Machine (M2M) communication is a key enabling technology for the future Industrial Internet of Things (IIoT) applications. It plays an important role in the connectivity and integration of computerized machines, such as sensors, actuators, controllers, and robots. The requirements in flexibility, efficiency, and cross-platform compatibility of the inter-module communication between the connected machines raise challenges for the M2M messaging mechanism toward ubiquitous data access and events notification. This investigation determines the challenges facing the M2M communication of industrial systems and presents a data-oriented M2M messaging mechanism based on ZeroMQ (ZMQ) for the ubiquitous data access in rich sensing pervasive industrial applications. To prove the feasibility of the proposed solution, the EU funded PickNPack production line with a reference industrial network architecture is presented, and the communication between a microwave sensor device and the Quality Assessment and Sensing (QAS) module controller of the PickNPack line is illustrated as a case study. The evaluation is carried out through qualitative analysis and experimental studies, and the results demonstrate the feasibility of the proposed messaging mechanism. Due to the flexibility in dealing with hierarchical system architecture and cross-platform heterogeneity of industrial applications, this messaging mechanism deserves extensive investigations and further evaluations.

Index Terms—IIoT, M2M, collaborative automation, messaging mechanism, ZMQ

I. INTRODUCTION

THE future IoT world is envisaged to be surrounded by various unobtrusive sensing and computing devices observing the ambient environment and providing adapted services to satisfy human's expectations. It is estimated that there will be over 50 billion devices connected to the Internet by 2020, and the rapid increase of the devices has started a new paradigm of computing in which all the objects become interactive [1][2]. In this new computing paradigm, machines talk to one another to achieve some tasks automatically and intelligently for different application scenarios, including manufacturing, healthcare, home automation, supply chain, retailing, etc. [3]. M2M communication allows the devices to

exchange information with each other in an autonomous way [4]. The overwhelming volume of data is generated and stored by the connected machines for human observation, analysis, and further knowledge discovery [5]. The effective use of the machine generated 'Big data' with Knowledge Discovery in Database (KDD) technique takes advantage of the collected data to optimize the system performance by creating valuable domain knowledge. The new computing paradigm, which features ubiquitous sensing, data interaction, data collection and data analysis is considered to be a promising solution to revolutionize the industrial applications by connecting the machines and allowing collaborative automation between machines and intelligent optimization of industrial processes [6][7]. The new emerging technologies are expected to reshape the future industrial systems, and new concepts are created, such as IIoT, Industry 4.0, smart manufacturing, digital manufacturing, and factory of things [8][9][10].

In the computerized IIoT systems, data transmission and data sharing between machines are critical parameters which affects the performance of the whole system, and investigations in this area are increasingly gaining research interests. The new emerging standards, protocols, and cost reduction of the commercially available sensing and communication modules has promoted the progress of M2M in industry applications [11][12]. However, the complex system structure and heterogeneity in hardware and software platforms have resulted in challenges for ubiquitous access and interoperable sharing of the machine generated data. The Quality-of-Information (QoI) in IIoT sensory environment becomes an issue as well [13][14]. Particularly, since the machines perform different functionalities on heterogeneous hardware and software platforms, an underlying standard messaging mechanism based on cross-platform technologies to support the communication between them becomes a major concern to the success of autonomous industrial systems.

This investigation aims to provide a generic and flexible M2M messaging mechanism to deal with the complex structure and heterogeneity problems of IIoT applications. It identifies the characteristics of IIoT applications regarding machine communication and then presents a data-oriented messaging mechanism based on the selected M2M messaging protocol. In order to evaluate the proposed solution, a microwave sensor machine for food quality inspection in a proposed reference industrial network architecture is implemented as a case study. The data acquisition and data sharing functionality of the proposed solution is implemented with the ZMQ messaging protocol. Test and evaluation of this application are conducted, and the results demonstrate the feasibility and performance of the designed M2M messaging mechanism for industrial applications.

This manuscript is submitted in June 2016. The work presented in this manuscript was supported by the European Commission under the Grant No. 311987.

Z. Meng, Z. Wu*, and J. Gray are with School of Electrical and Electronic Engineering, The University of Manchester, Oxford Road, Manchester M13 9PL, UK (e-mail: {zhaozong.meng, zhipeng.wu, john.gray.2}@manchester.ac.uk). C. Muvianto was with School of Electrical and Electronic Engineering, The University of Manchester, Manchester M3 9PL UK, and now with the University of Mataram, Indonesia (e-mail: mistiko_cahyo@yahoo.co.uk).

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The rest of this paper is organized as follows: Firstly, the enabling techniques are introduced, a review of related studies is presented, and some technical challenges are highlighted accordingly in Section II. Secondly, a ZMQ based data-oriented M2M messaging mechanism is developed with a reference industrial network architecture in Section III. Then, a case study on the flexible data communication of the microwave sensor controller in the PickNPack production line is presented in Section IV. Finally, conclusions are drawn and future work is suggested in Section V.

II. A REVIEW OF M2M COMMUNICATION TECHNIQUES

There is a significant difference between the M2M and traditional ‘Human-to-Human (H2H)’ Internet in the data transmission pattern. Therefore, it is important to determine the characteristics and potential challenges facing M2M communication systems. This section introduces the key enabling technologies of M2M communication and provides a review of the peer investigations. The state-of-the-art enabling technologies, the recent progress, and peer studies in M2M communication are summarized and presented. And then, some open issues and challenges facing this specific area are highlighted.

A. Enabling Technologies, Progresses, and Related Work

1) Enabling Technologies for M2M Communication

Although a wide range of research fields are related to M2M communication, they share the same underlying supporting technologies. Generally, M2M is leveraged by the rapid progress of technologies in the Information and Communication Technology (ICT) domain, including embedded electronics, software platforms, wireless communication standards, network infrastructures, communication protocols and messaging mechanisms. The supporting technologies in an industrial application context are summarized as shown in Fig. 1.

(1) Embedded Electronics

The embedded electronics based on the processing units such as Micro-Controller Units (MCU), Digital Signal Processors (DSP), ARM processors, Programmable Logic Controllers (PLC), and System-on-Chip (SoC) devices are important enabling techniques for M2M and IoT. They interface the physical world with computerized machines and make them visible and manageable by computer systems. Different from PC, the embedded electronics can be tailored to specific application scenarios to reduce the cost and promote the performance. Thus, handheld, miniaturized, energy-efficient, and low-cost devices appear and are used as an interface between the Information Technology (IT) infrastructure and sensor nodes. In addition, various sensors, Micro-Electro-Mechanical Systems (MEMS), Radio Frequency Identification (RFID), and Wireless Sensor Network (WSN) convert physical variables into computer understandable data [15], which is interfaced to the IT infrastructure for potential use. Then, robots and actuators interact with the physical world according to the computer command in turn.

(2) Software Platforms

Embedded Linux such as Ubuntu, mobile platform such as Android and iOS, WSN operating systems such as TinyOS [16], LiteOS, and RIOS are all essential technologies to the success of M2M applications. The embedded operating systems allow the machines to fulfill their tasks and communicate with the other devices based on standard protocols. These operating systems are normally micro-kernel architectures that provide multi-threading with standard API development in C/C++. Most of them support standard protocols or IoT protocols such as 6LoWPAN, IPv6, RPL, TCP, and UDP [17]. Since the distributed machines perform different tasks individually, the software platforms are the intermediary technology to integrate them with the IT infrastructure.

(3) Network Infrastructures

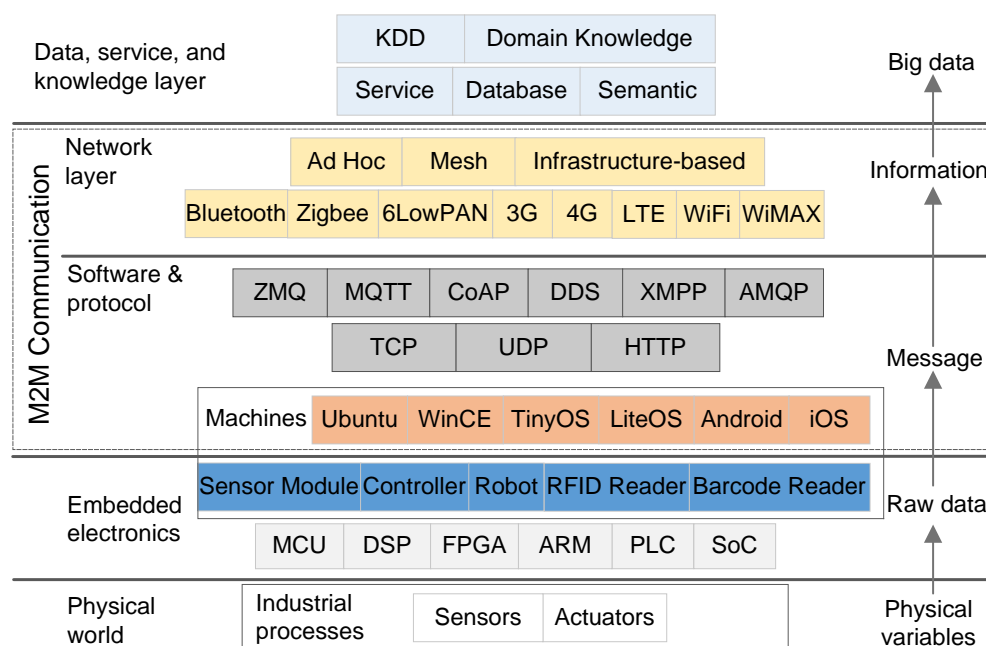


Fig. 1. Enabling Technologies of M2M Communications for Industrial Applications.

The wireless communication standards such as Bluetooth, Zigbee, 6LoWPAN, 3G, 4G, LTE, WiFi, and WiMAX [18], and the wireless infrastructure built with them have also promoted the progress of M2M communication. The strength of the wireless network is that it breaks the limitation of location and devices can be anywhere. To enable the messaging between machines, a connection between machines should be built with an appropriate model, such as Ad Hoc [19], mesh network, infrastructure-based network, etc. The maturity of network technologies and the wide support by various machines have provided flexibilities for the design and development of M2M communication-based systems.

(4) Messaging Mechanisms

Traditional H2H Internet and industrial communication between machines are largely direct use of HTTP and TCP/UDP. With these raw communication protocols, information transmission can also be achieved. But it is difficult and time-consuming to deal with the efficiency, failure handling, and reliability. In a post-Moore Law era, the distributed computing becomes the mainstream pattern to handle the pervasive computing devices. Therefore, messaging mechanisms are required to deal with the communication between the distributed nodes.

With respect to the communication level, many optional protocols, middleware, and API libraries are promising for the purpose of M2M messaging. Although they are based on different techniques and toward different application scenarios, the purpose is to gain flexibility in the interactions between connected machines. The main competitive protocols, middleware, and API libraries for M2M communication are summarized as follows:

- Message Queue Telemetry Transport (MQTT) - A simple and lightweight messaging protocol designed for resource-limited devices and low-bandwidth, high-latency networks with low-reliability needs.
- Constrained Application Protocol (CoAP) - A specialized web transfer protocol for use with constrained nodes and constrained networks in the IoT-based on Representational State Transfer (REST) model, which is suitable for M2M applications such as smart energy and building automation.
- Data Distribution Service (DDS) - A specific M2M middleware that provides scalable, real-time, dependable, high-performance and interoperable data exchange between machines. DDS can be used in certain IoT implementations.
- Extensible Messaging and Presence Protocol (XMPP) - A communication protocol for message-oriented middleware based on XML (Extensible Markup Language), which enables the near-real-time exchange of structured yet extensible data between any two or more network entities.
- Advanced Message Queuing Protocol (AMQP) - An open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing, reliability and security. Example AMQP based solutions are RabbitMQ and StormMQ.
- ZMQ - A high-performance asynchronous messaging library, aimed at use in distributed or concurrent applications. It provides a message queue without dedicated

message broker, which resembles that of Berkeley sockets for Inter-process Communication (IPC).

In addition to the above enabling techniques, the high-level data managing and processing methods such as semantic technique, cloud service, and data analysis technology all contribute to M2M enabled industrial applications. Integration of the data in both hardware and software infrastructure and potential use of the machine generated data are the major concerns in this specific area and the future of IIoT systems.

2) Existing M2M Messaging Techniques

The M2M messaging techniques are invented for M2M communication of different requirements and use scenarios. Some methods are suitable for handling lightweight messaging under varying levels of latency due to bandwidth constraints or unreliable connections, such as MQTT, some are suitable for resource-limited internet devices like WSN nodes in IoT applications, such as CoAP. Some others focus on data format, message orientation, queuing, routing, reliability, and security like AMQP. There are techniques strong in the capability to support service discovery across network domains, such as XMPP, which are well-suited for cloud computing where virtual machines and networks would present obstacles to alternative service discovery and presence-based solutions. In addition, some aim to simplify complex network programming for big data applications including financial trading, air-traffic control, smart grid management, etc. like DDS. There are also techniques providing easy connection, presence, and flexible development of application level functions like ZMQ.

Since each solution has its strengths and particularly suited fields of applications, the selection of the alternative techniques should be based on the requirements of the practical situations of the industrial applications.

3) Related Work

The industrial M2M market is undergoing a fast transformation as enterprises are increasingly realizing the value of connecting geographically dispersed people, devices, sensors and machines to corporate networks. Related work can be classified into two categories: (1) standard and protocol based studies, and (2) application-specific designs and developments.

There are many Standard Development Organizations (SDOs) working towards system framework and business models called Manufacturing Reference Architectures (MRAs), such as European Telecommunications Standards Institute (ETSI) M2M, Smart Manufacturing Leadership Coalition (SMLC) open manufacturing platform, oneM2M reference architecture, and Microsoft Discrete Manufacturing Reference Architecture (DiRA). ETSI M2M service architecture is the first step towards a universal M2M platform, which already provided a good maturity level on unified communication capabilities and protocols [20]. SMLC is a US initiative on next-generation manufacturing, the goal of which is seamless manufacturing execution through a cloud-based, open-architecture manufacturing infrastructure and marketplace supporting real-time applications that optimize manufacturing systems [21]. SMLC is building an open smart manufacturing platform for collaborative industrial networked information applications. OneM2M Global Initiative currently works on a

harmonized reference architecture, which integrates previous work such as the aforementioned by ETSI, as well as other standard organizations, such as Open Mobile Alliance (OMA) and BroadBand Forum (BBF) [22]. In addition, Microsoft Discrete Manufacturing Group inaugurated the DiRA, a cloud-based framework that connects smart devices across manufacturing networks, which emphasizes user interfaces, enterprise-class social-computing solutions, smart connected devices, and security-enhanced solutions [23].

Further to SDO based solutions, investigations are carried out in both academia and industry. Lo et al. [24] proposed a cellular-centric M2M service architecture combining ETSI and 3rd Generation Partnership Project (3GPP) architectures. This proposal used M2M relay nodes as data connector to implement a tunnel-based aggregation scheme which merges data from several machines destined to the same tunnel exit-point. In order to solve the server congestions in random access channels of cellular systems for M2M communication, Lien et al. [25] proposed a cooperative Access Class Barring (ACB) to corporate 3GPP ACB among Base Stations (BS). Mougy, et al. [26] proposed an application and context-aware framework for resource management in dynamic wireless M2M networks. This framework was composed of a software platform and a reasoning engine. The software platform was designed to be context-aware and used multi-level adaptations to support the dynamic collaboration of devices. The problem of uplink scheduling of M2M devices in Long-Term Evolution (LTE) network was studied by Lioumpas and Alexiou [27], coverage and capacity of M2M in specific realm of LTE were studied by Ratasuk, Tan and Ghosh [28], a trade-off design in M2M communication from a power and energy optimal design perspective was investigated by Dhillon et al. [29], large-scale M2M communication and the performance was discussed by [30], and trust and security issues were addressed in the work published by Bartoli et al. [31][32]. To serve dynamic applications across different platforms, Liu et al. [33] proposed a naming, addressing, and profile server (NAPS) as a middleware to bridge different platforms in IoT sensory environments.

In addition to studies on standard and framework, there are also studies focusing on applications-specific developments which utilize the M2M communication for practical implementations. M2M systems in daily operations can be found in several application scenarios, such as remote monitoring, track and trace, facility management, and smart metering [34].

Mougy et al. [26] proposed an application and context-aware framework for resource management in dynamic wireless M2M networks composed of a software platform and a reasoning engine. Niyato et al. [12] proposed an M2M Home Energy Management System (HEMS) with a network architecture to collect status and power consumption demand from home appliances. Zhang et al. [35] proposed an M2M network architecture for services in home surroundings, such as mobile healthcare, home automation, security, appliance monitoring, lighting, and entertainment. This investigation focused on the QoS management in home M2M network considering the visual and multimedia requirements and proposed a cross-layer joint admission and rate control design for QoS-aware multimedia sharing. Jung et al. [36] presented an M2M

healthcare solution which integrated the smartphone and IPv6 techniques in a WSN for health monitoring and to providing effective healthcare service. Jung and Chun [37] then introduced a global M2M healthcare system to monitor patients' health condition with non-intrusive low-power embedded wearable sensors over 6LoWPAN. Arce et al. utilized the ZMQ for the data exchange between GPU and CPU to forecast high frequency financial time series using parallel Feed Forward Neural networks (FFN) with Computer Unified Device Architecture (CUDA). ZMQ was used to receive chunks of data input features and publish the output features to minimize the overhead and system call [38]. ZMQ as a communication middleware was used for communication between embedded processor and Linux-based control board to design energy management applications toward optimal energy distribution, redundancy control, and abstraction of the inventors [39]. Shakhimardanov, et al. [40] introduced a pragmatic model, Protocol Stack View (PSV), for the analysis of distributed robotic software to deal with the non-functional aspects such as network scalability and predictability of system behavior.

The investigations on M2M communication will increasingly gain research interests in the envisioned future. However, to the best of our knowledge, there are seldom comprehensive designs and experimental studies applying M2M messaging technologies to deal with the complex relationships of heterogeneous machines and the flexibility and efficiency of data transmission between them in industrial applications. The increasing number of machines being connected to the networks and the requirements to interact with each other becomes a common problem facing the M2M communication in terms of speed, reliability, light-weightness, and cross-platform interoperability.

B. M2M Communication towards Industrial Applications

Compared with H2H Internet communication, M2M is featured with more communication nodes, lower bandwidth, real-time processing, and many nodes are required to be energy efficient [32]. Since data is the most important concern in M2M systems, the unobtrusive collection, reliable transmission, and effective use of machine-generated data are the main tasks to achieve. The M2M systems are highly dependent on the IT infrastructure for data collection, sharing, and decision making. In terms of data communication for industrial applications, the specificities of M2M communication are:

- Heterogeneity in hardware and software platforms
- Frequent notification and peer machine observation
- Collaborative automation between machines
- Real-time event handling and data processing
- Data of various format and size to transmit

Therefore, the communication between the connected machines is expected to be:

- Cross-platform interoperable with basic standards
- Machine discovery and presence capability
- Flexible data interaction and event notification
- Reliable, efficient, and fast in speed
- Failure handling and recovery capability

Since more and more computing devices such as sensor nodes, RFID readers, smartphones, and laptops are integrated

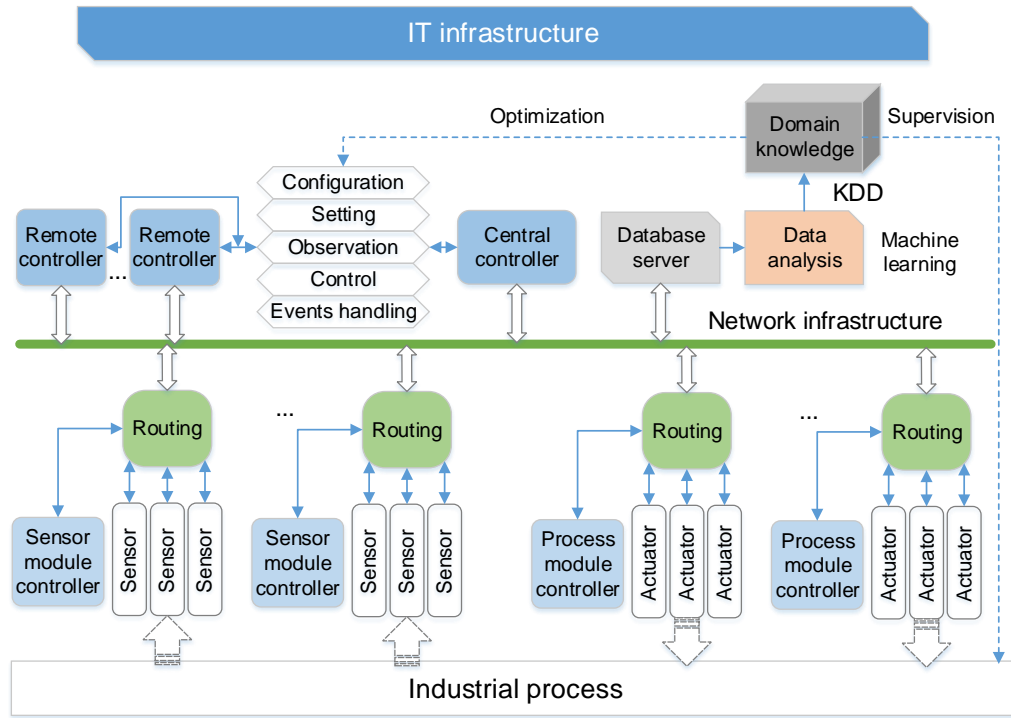


Fig. 2. A Reference M2M Network Architecture for Industrial Applications.

into the IT infrastructure, the heterogeneity in both hardware and software and distribution of these devices are common problems for this particular research area. Therefore, generic and cross-platform solutions for efficient data interaction between machines are expected.

This investigation therefore proposes a data-oriented M2M messaging mechanism for data integration between machines in industrial applications from a software architecture point of view. This messaging mechanism focuses on the machine presence and discovery, M2M messaging model, and its potential use for industrial applications as a cross-platform interoperable tool for data collection, ubiquitous data sharing, and events notification.

III. DATA-ORIENTED M2M MESSAGING MECHANISM

A. A Reference Industrial Network Architecture

In order to investigate the M2M communication and satisfy the requirements identified in Section II, a reference industrial network architecture is proposed following the modular design, which divides the machines into sub-groups of different hierarchies. As shown in Fig. 2., low-level machines in the industrial fields are divided into sensor groups and actuator groups. The middle-level machines are the sensor module controllers, and process module controllers. Then, the high-level are central controller, remote controllers, and data management systems. This reference architecture focuses on the collection, transmission, and usage of data towards ubiquitous data access and intelligent optimization of the industrial processes.

In this reference architecture, the low-level electronics are embedded into the industrial systems for data collection and command execution. The collected data is sent to and stored in the database server, which is used to discover high-level domain knowledge to supervise the industrial processes. The

observation of machines, utilization of data, and high-level supervision constitute a closed-loop optimization of the industrial processes.

The machines in the three-layer hierarchy topology are devices, module controllers, and high-level controllers. The devices including sensors and actuators interact with industrial processes. The module controllers manage devices in specific modules, and high-level controllers are responsible for system configuration, remote or central control, and data analysis.

The communications between the sensor nodes and sensor module controller, sensor module controller and central controller, actuator nodes and process controller, and process controller and central controller are all achieved with the ZMQ messaging mechanism. Data is transmitted with message envelope, and heartbeat messages are sent frequently to notify the liveness of connections. Failure of connection is returned to notify the nodes and controllers concerned.

The following sub-sections illustrate the functionality pattern, heartbeats for machine presence and discovery, and messaging model of ZMQ based messaging mechanism, and explains why ZMQ is suitable to implement the proposed reference system architecture.

B. ZMQ Messaging Mechanism

ZMQ is an efficient and light-weight embeddable library or ‘message-oriented middleware’ for messaging between machines, which is capable of intervention when something goes wrong, and is fault-tolerant, low in cost, and much faster than common instant messaging [41]. Especially, it has the following features:

- It handles I/O asynchronous, in background thread;
- Components can come and go dynamically and ZMQ will automatically reconnect;
- It queues messages automatically when needed;
- It has ways of dealing with over-full queues;

TABLE I
ZMQ PATTERNS [42]

PATTERNS	Description
Lazy pirate	Reliable request-reply from client side
Simple pirate	Reliable request-reply using load balancing
Paranoid pirate	Reliable request-reply with heartbeating
Majordomo	Service-oriented reliable queuing
Titanic	Disk-based/disconnected reliable queuing
Binary star	Primary-backup server failover
Freelance	Broker-less reliable request-reply
Lazy pirate	Reliable request-reply from client side

- Applications talk to each other over arbitrary transports: TCP, multicast, in-process, inter-process;
- It lets route messages using a variety of patterns;
- It lets create proxies to queue, forward, or capture messages with a single call;
- It handles network errors intelligently, by retrying automatically in case where it makes sense;
- It does more with less power consumption;
- Supported more than 40 programming languages and multiple operating systems.

These attributes made ZMQ a promising candidate for machine messaging in IIoT scenarios, where machine discovery and presentation, flexible messaging, efficiency, and failure handling and recovery are the major concerns.

1) ZMQ Patterns

ZMQ is an embeddable networking library but acts like a concurrency framework. It provides sockets that carry atomic messages across various transports like in-process, inter-process, TCP, and multicast. The different patterns of ZMQ are listed in TABLE I.

For an industrial application which features machine collaboration, the Paranoid Pirate Pattern (PPP) is a promising candidate for request-reply with heartbeating. The diagram of this pattern is given in Fig. 3. In this pattern, the sensor nodes are the workers, the module controllers are the queues, and central controllers are the clients in the figure. The data transmission between them is in a reliable request-reply way with heartbeating. Each node including workers, queue, and clients notifies their presence with heartbeats. The queue forwards messages between workers and clients. Therefore, the workers can talk to the clients through the queue to update data,

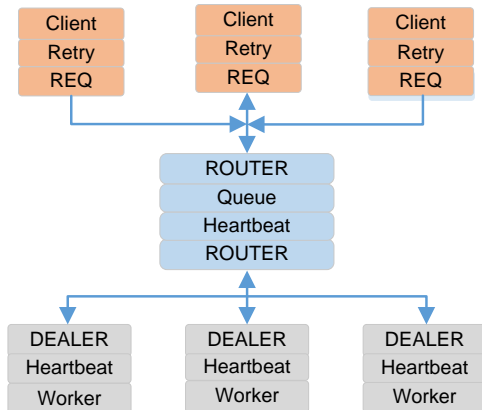


Fig. 3. Functional Diagram of ZMQ PPP Pattern.

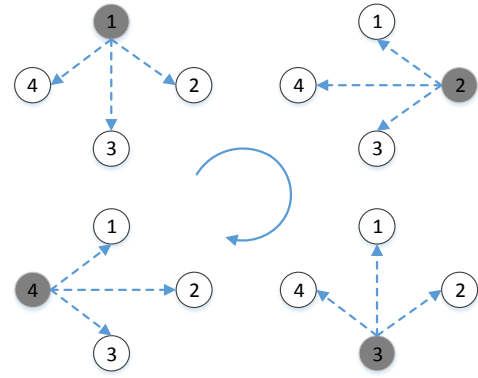


Fig. 4. Heartbeating in a Group with Four Nodes.

and the clients can also talk to the workers for observation or command execution.

2) Heartbeating

Heartbeating solves the problem of knowing whether a peer is alive or dead. This is not an issue specific to ZMQ. TCP has a long timeout (15 minutes by default), which means it can be imposed to know whether a peer has died or has been disconnected. The timing of communication between machines is a critical issue for some application scenarios. There are three ways of heartbeating: shrugging it off, one-way heartbeats, and ping-pong heartbeats and the adopted pattern employs the second. The heartbeat messages flow asynchronously in both directions, and either peer can decide the other is 'dead' and stop talking to it.

As shown in Fig. 4., the connected nodes send heartbeating messages to the peers continuously with specific intervals. The peers receive the heartbeating messages to confirm whether a node is alive or not. With this method, any abnormality of a node can be determined by the peers in the network. Following is how the heartbeats are handled to the queue:

- Calculate when to send the next heartbeat, which is a single variable because worker talks to queue only.
- In the `zmq_poll` loop, whenever defined time interval passes, a heartbeat is sent to the queue.

The code snippet in List 1. is the essential heartbeating function for the workers in the PPP pattern.

The heartbeat time is controlled by the timestamp of the system returned by function `zclock_time()`. With the above method, the heartbeat can be sent regularly but reduces the too many or too few, which may result in network congestion or disconnection. If workers die one by one, the client will

List 1. Time Interval-based Heartbeating.

```
uint64_t heartbeat_at = zclock_time() +
HEARTBEAT_INTERVAL;
while (true) {
    ...
    int rc = zmq_poll(items, 1, HEARTBEAT_INTERVAL*
        ZMQ_POLL_MSEC);
    ...
    //Send heartbeating to queue if it's time to
    if (zclock_time() > heartbeat_at)
    {
        heartbeat_at=zclock_time()+HEARTBEAT_INTERVAL;
        //Send heartbeat message to queue
        ...
    }
}
```

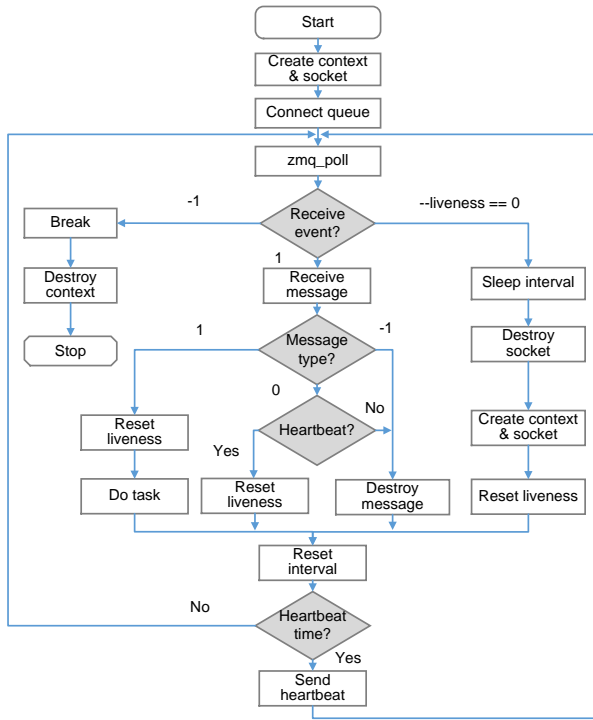


Fig. 5. The Flowchart of ZMQ PPP Worker.

eventually give up after attempting several attempts to connect. If the queue is stopped and restarted, both the client and workers will reconnect and carry on. No matter what is done to the queue and workers, the client will never get an out-of-order reply.

The heartbeating allows the machines to present themselves and discover the peer machines. The heartbeating, failure handling and recovery are essential to the maintenance of network connected systems.

C. ZMQ based Messaging Model

ZMQ is a very complicated messaging queuing mechanism for secure and fast data transmission between machines especially the thread communication, data queuing, and concurrency. But the fundamental and design of ZMQ is not the focus. This investigation employs this advanced technology to deal with the data transmission of the distributed machines. The PPP pattern includes workers, queue, and clients to perform the robust and reliable queuing for data transmission. In this subsection, the principle of the ZMQ message queuing is introduced with a PPP worker. The flowchart diagram of the worker is given in Fig. 5.

As shown in the diagram, the worker monitors the state of the queue and sends out heartbeats according to the time variables *liveness* and *heartbeat_at*. The queue works normally if *liveness* is greater than 0. Otherwise, disconnect, destroy context, create a new context, and connect again. It also monitors the time, and the heartbeat will be sent when the interval between *heartbeat_at* and present time is greater than the heartbeat interval. With this method, both the worker and queue can observe the state of each other, and disconnect and rebuild when the connection collapses. Industrial application tasks can be done in the 'Do task' step in Fig. 5 to gather data, process data, and send data to the data management nodes, or to execute received commands with actuators.

According to the above description, this ZMQ based messaging can be easily customized to fit the proposed reference industrial system architecture to deal with the communication between devices, module controllers, and high-level controllers.

IV. CASE STUDY

In order to demonstrate the feasibility of the presented solution, a quality inspection microwave sensor of food manufacturing production line in the PickNPack project is implemented as a case study. The EU PickNPack project aims to integrate the state-of-the-art sensors and robots with emerging ICT technologies to build a flexible automatic food manufacturing production line. There are seven modules in PickNPack line: thermoformer, pickrobot, Quality Assessment and Sensing (QAS), printing, laser sealing & cutting, packrobot and cleaning. The devices are connected to the network for machine collaboration and data integration towards high-level automation.

The microwave sensor is an integral part of the PickNPack production line for online food quality inspection. It scans food materials under test with a 32-channel transmitter-receiver antennas array, which can perform line scanning when food products move through it. The sensor nodes communicate with the QAS module controller which manages 5 sensors in total (Microwave, RGB, 3D, hyper-spectral, and X-ray) for multi-facet evaluation of food quality, and sends the measured features of food products to it. The module controller then talks to the line controller for quality rating and automatic label printing, etc. In this section, the communication between microwave sensor, QAS module controller, and central controller are implemented with ZMQ following the proposed reference architecture.

A. Design and Implementation

For the implementation of the system, the microwave sensor controller for data acquisition and signal processing, the module controller to integrate different sensors, and the central controller for remote monitoring, control, and data analysis are integrated. In practical M2M systems such as an industrial production line, sensor controller can be some light-weight embedded electronics and sensor nodes, and central controller can be replaced with smartphones, tablets, and mini-PCs. With the ZMQ PPP pattern, the sensor controller, module controller, and central controller are the worker, queue, and client respectively. The connectivity and messaging between them are as shown in Fig. 6.

In this system, microwave sensor controller application is developed with Matlab 2013a. Module controller application and central controller applications are developed with Visual

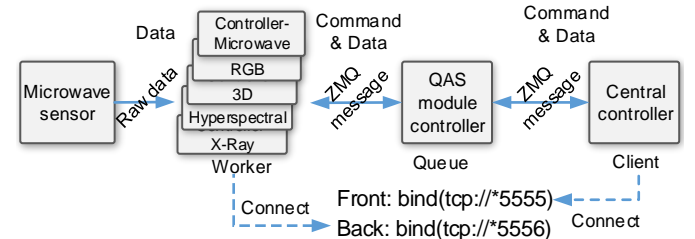


Fig. 6. The Connectivity and Messaging of the QAS System.

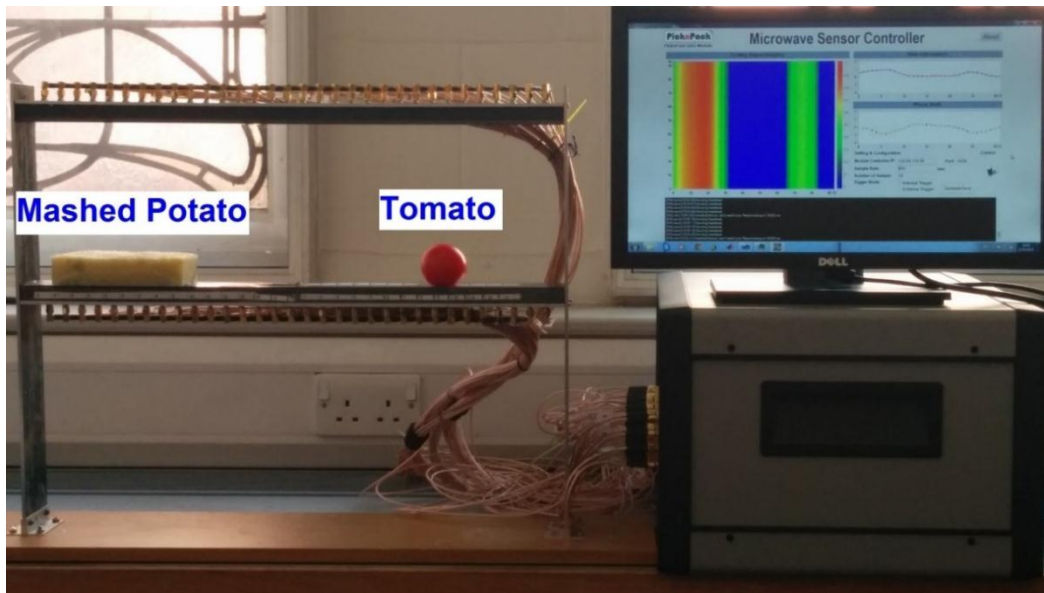


Fig. 7. The Microwave Sensor in Operation

C# .NET 2010. Since the instrument control toolbox of Matlab 2013a supports the development of NI DAQ module inside the microwave sensor controller box, it is a convenient way to implement and evaluate the mathematical methods with Matlab. Therefore, in this proof of concept development stage, Matlab 2013a is selected to develop applications for data acquisition, processing, and ZMQ data transmission on sensor controller. The IP address and the COM ports of PPP Queue need to be assigned for the communication workers and clients. As shown in Fig. 6, port 5556 and 5555 are assigned to bind workers and clients.

Since the structure of the result data generated by microwave sensor is not complicated, raw string format is used for data transmission. For complicated data structures, cross-platform interoperable tools, such as JavaScript Object Notation (JSON) and XML can be used for data presentation.

B. Test and Evaluation

With the designed architecture and ZMQ messaging mechanism, the microwave sensor, QAS module controller, and central controller are successfully implemented. Three identical DELL desktop PCs with Intel Core™i7-3770 CPU @3.4GHz, 16.0G RAM, and 64bit Windows 7 Enterprise operating system are used to build up the system. One PC works as the sensor controller running Matlab application, and the other two work

as the queue and client running Visual C# .NET applications. The PCs communicate with a WiFi network.

During the implementation of the system the following objectives were achieved:

- (1) The sensor controller, QAS module controller, and central controller can talk to each other flexibly for event notification and data sharing;
- (2) Central controller can access sensor data streaming in real-time at microwave sensor's working speed of 100 frames/second;
- (3) The sensor controller, QAS module controller, and central controller can observe the states of the others and be observed with heartbeats;
- (4) The worker, queue, and client work with different application platforms and the data transmissions between them are stable and reliable.

The microwave sensor device in operation is shown in Fig. 7. The microwave sensor inspects the quality of food according to the gain attenuation and phase shift when microwave penetrates the food objects. The sharing of the microwave sensor data between the microwave sensor controller, QAS module controller, and central controller is shown in Fig. 8. In Fig. 8., (a) gives the sensor controller results in Matlab GUI, (b) gives the QAS module controller displaying the data from sensor controller in Visual C# .NET application, and (c) presents the central controller displaying the data accessed from

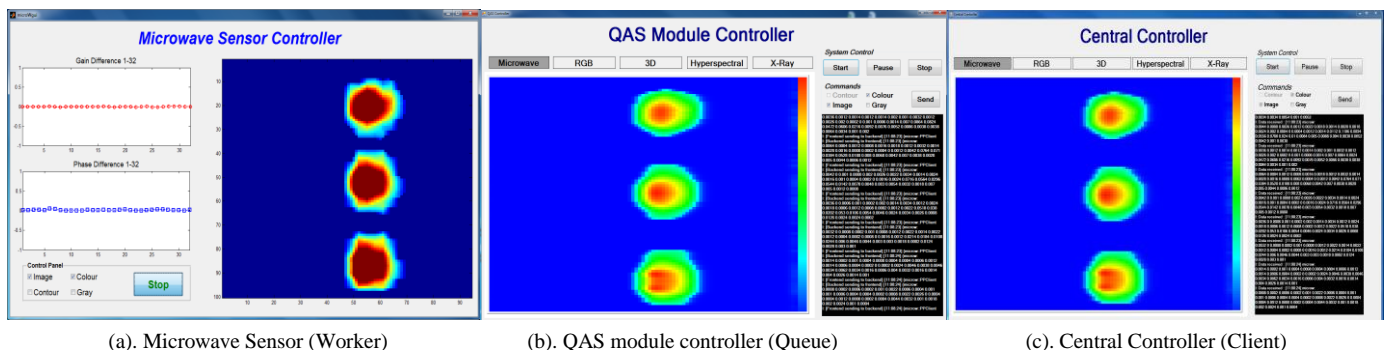


Fig. 8. Data Sharing between Microwave Sensor, QAS Module Controller, and Central Controller.

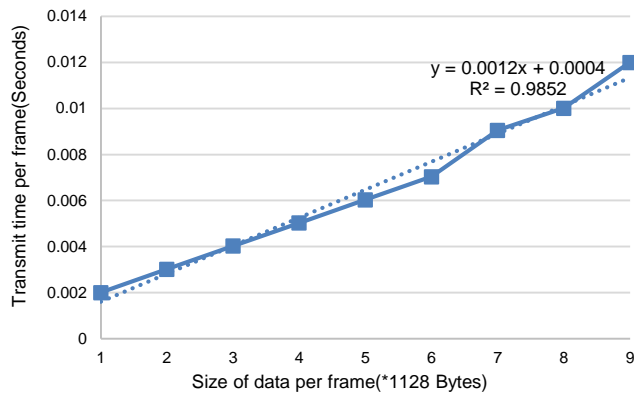


Fig. 9. Data Transmission Time with the Increase of Data Size.

sensor controller through QAS module controller for observation purpose. The QAS module controller and central controller can connect to the sensor module devices and access the raw data or measurement results for online real-time decision making or for further use. Database server can also behave as a client to access data from devices through their module controller. Therefore, the ubiquitous data access between multiple machines is made possible.

In order to evaluate the efficiency of the designed system, it is tested with the following methods:

(1) With fixed number of workers and clients, the increase of data transmission time with the size of data is tested.

(2) With fixed size of data, the increase of data transmission time with the number of workers is tested.

The data used for testing is the microwave gain attenuation and phase shift of the 32 channels in the microwave sensor, which is converted into string from Matlab arrays of double precision. The size of one sample is 1128 Bytes. The test results are given in Fig. 9 and Fig. 10, respectively.

Fig. 9 shows that the data transmission time per frame increases with the size of the data to be transmitted. The transmission time and the size of data are largely in linear relationship. When the size of data is at 1.128 KB, the transmission can reach a speed of 500 frames/second. However, when the size is 11.28KB, the speed is about 80 frames per second. This speed of interest for many industrial applications.

Fig. 10 shows the data transmission time per frame increases with the number of workers, which is largely in polynomial function trend. When there are 5 workers, the data transmission speed for 1.128KB is over 25 frames/second. However, when there are 20 workers, the frame-rate is only 4, which becomes quite slow. One apparent reason is that, the 20 nodes are running on one PC and the runtime environments of the nodes consume too much computation resource of the PC. Performance can be improved if each node is running on an individual PC. Another reason lies in the time delay of data processing algorithm and string processing of microwave sensor controller with Matlab platform.

The implementation of the microwave sensor in PickNPack QAS module proves the feasibility of ZMQ based messaging mechanism for flexible presence, messaging, and ubiquitous data access between the machines for industrial applications.

C. Application of the Proposed Messaging Mechanism

This ZMQ based M2M messaging mechanism is specially designed for industrial applications, where machines feature

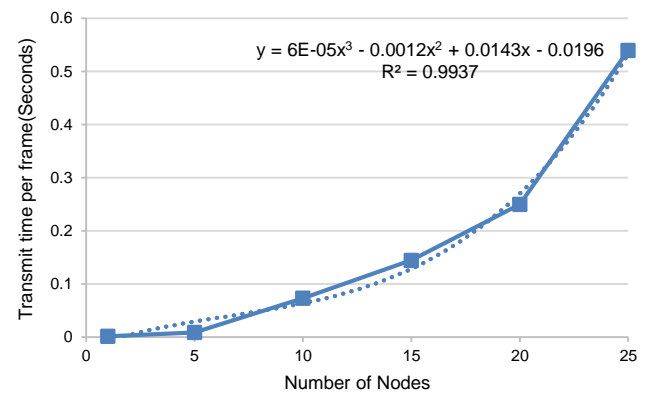


Fig. 10. Data Transmission Time with Increase of Number of Nodes.

complex relationship and heterogeneity in hardware and software platforms.

The implementation of the messaging mechanism involves the following steps: (1) To design of the machine hierarchy and system architecture according to the requirements; (2) To define the roles of the machines and their functionalities, connections, messaging events, and data models; (3) To implement of the ZMQ libraries to build the connectivity; and (4) Implementation the application-specific functionalities. The efficient machine discovery, present, and messaging, and the broadly support by mainstream software platforms make functionality and data the major concerns and the development relatively convenient.

V. CONCLUSION

The machines for future industrial systems will feature higher quality connectivity, smarter, and more interactive. The ubiquitous data access and efficient data integration between these distributed machines becomes a common problem for both academia and industry. This investigation highlights the major concerns in building a distributed system in a systematic manner, provides a data-oriented M2M communication mechanism with a reference industrial system architecture, and gives a proof-of-concept implementation and experimental evaluation.

In the system architecture proposed according to the specificity of M2M communication, the focus is the messaging mechanism between the machines for event and command notification and data sharing. In the case study, command notification and data sharing in continuous streaming are implemented. With real time data acquisition, the implemented microwave sensor module has achieved the cross-platform interoperability, fast data sharing, and machine state monitoring, which serves as a foundation for proposed reference industrial system architecture.

From the work presented in this investigation, we can draw the following conclusions.

(1) The proposed ZMQ based data-oriented messaging mechanism is a promising tool to deal with machine connectivity, machine presence and discovery, and messaging to allow the ubiquitous data access and data interaction for rich sensing IoT application scenarios, especially hierarchical and modularized IIoT systems.

(2) The ZMQ based data-oriented messaging and the reference architecture give a comprehensive picture of the

M2M industrial applications. The microwave sensor, QAS module, and central controller of PickNPack project constitute a miniature system of digital manufacturing, which have demonstrated the concept.

(3) ZMQ messaging mechanism is based on low-level TCP or UDP sockets and is widely supported by various software platforms. Its flexibility, efficiency, and cross-platform capability allow the implementation on various powerful computers and light-weight devices to be relatively convenient.

In the envisaged future M2M enabled IoT era, physical environment will be merged with the digital world and machine needs to interact with each other intelligently. A generic data-oriented architecture with fast, efficient, cost effective, and cross-platform interoperable messaging mechanism is a key technology to accommodate the increasing number of machines, and enable the data transmission and interaction between them. The future work of this investigation will focus on the optimization of ZMQ messaging mechanism to make it more convenient to build connection and more suitable for semantic understanding between machines.

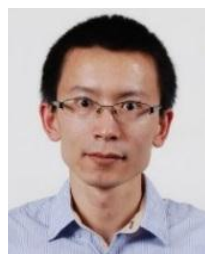
ACKNOWLEDGEMENT

The work presented in this paper is supported by the European Commission through PickNPack project (project code: 311987). The authors would also like to thank Dr. Niels Wouters and Professor Wouter Saey at KU Leuven, Belgium for their support in the integration of microwave sensor with QAS module.

REFERENCES

- [1] M. A. Feki, F. Kawsar, M. Boussard, and L. Trappeniers, "The Internet of Things: The Next Technological Revolution," *Computers*, vol. 46, no. 2, pp. 24–25, 2013.
- [2] W. Lumpkins, "The Internet of Things Meets Cloud Computing," *IEEE Consum. Electron. Mag.*, vol. 2, no. 2, pp. 47–51, 2013.
- [3] Z. Meng and J. Lu, "A Rule-based Service Customization Strategy for Smart Home Context-aware Automation," *IEEE Trans. Mob. Comput.*, vol. 15, no. 3, pp. 558–571, 2016.
- [4] J. N. Al-Karaki, K. C. Chen, G. Morabito, and J. De Oliveira, "From M2M Communications to the Internet of Things: Opportunities and Challenges," *Ad Hoc Networks*, vol. 18, pp. 1–2, 2014.
- [5] R. Schniederman, "Machine-to-Machine Connectivity is Booming," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 10–13, 2013.
- [6] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-Oriented Data Storage Framework in Cloud Computing Platform," *Ind. Informatics, IEEE Trans.*, vol. 10, no. 2, pp. 1443–1451, 2014.
- [7] L. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2242, 2014.
- [8] S. Savazzi, V. Rampa, and U. Spagnolini, "Wireless Cloud Networks for the Factory of Things: Connectivity Modeling and Layout Design," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 180–195, 2014.
- [9] C. Perera, C. H. Liu, and S. Jayawardena, "The Emerging Internet of Things Marketplace from an Industrial Perspective: A Survey," *IEEE Trans. Emerg. Top. Comput.*, vol. 3, no. 4, pp. 585–598, 2015.
- [10] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things From Industrial Market Perspective," *IEEE Access*, vol. 2, pp. 1660–1679, 2014.
- [11] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward Intelligent Machine-to-Machine Communications in Smart Grid," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 60–65, 2011.
- [12] D. Niyato, L. Xiao, and P. Wang, "Machine-to-Machine Communications for Home Energy Management System in Smart Grid," *IEEE Commun. Mag.*, vol. 49, no. April, pp. 53–59, 2011.
- [13] C. H. Liu, J. Fan, J. W. Branch, and K. K. Leung, "Toward QoI and Energy-Efficiency in Internet-of-Things Sensory Environments," *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 4, pp. 473–487, 2014.
- [14] Z. Sun, C. H. Liu, C. Bisdikian, J. W. Branch, and B. Yang, "QoI-aware energy management in internet-of-things sensory environments," in *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2012, pp. 19–27.
- [15] M. T. Lazarescu, "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 3, no. 1, pp. 45–54, 2013.
- [16] P. Alagupandi, R. Ramesh, and S. Gayathri, "Smart Irrigation System for Outdoor Environment using Tiny OS," in *2014 International Conference on Computation of Power, Energy, Information and Communication*, 2014, pp. 104–108.
- [17] M. Weyrich, J. Schmidt, and C. Ebert, "Machine-to-Machine Communication," *IEEE Softw.*, vol. 31, no. 4, pp. 19–23, 2014.
- [18] Y. Zhang, L. Sun, H. Song, and X. Cao, "Ubiquitous WSN for Healthcare: Recent Advances and Future Prospects," *Internet Things Journal, IEEE*, vol. 1, no. 4, pp. 311–318, 2014.
- [19] N. A. Surobhi and A. Jamalipour, "A Context-aware M2M-based Middleware for Service Selection in Mobile Ad-Hoc Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3056–3065, 2014.
- [20] J. Latvakoski, M. Alaya, H. Ganem, B. Jubeh, A. Iivari, J. Leguay, J. Bosch, and N. Granqvist, "Towards Horizontal Architecture for Autonomic M2M Service Networks," *Futur. Internet*, vol. 6, no. 2, pp. 261–301, 2014.
- [21] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart Manufacturing, Manufacturing Intelligence and Demand-Dynamic Performance," *Comput. Chem. Eng.*, vol. 47, pp. 145–156, 2012.
- [22] M. Glaab, W. Fuhrmann, J. Wietzke, and B. Ghita, "Toward Enhanced Data Exchange Capabilities for the oneM2M Service Platform," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 42–50, 2015.
- [23] Microsoft Discrete Group Manufacturing, "A Discrete Manufacturing Reference Architecture (DIRA) Framework Whitepaper," 2011.
- [24] A. Lo, Y. W. Law, and M. Jacobsson, "A Cellular-centric Service Architecture for Machine-to-machine (M2M) Communications," *IEEE Wirel. Commun.*, vol. 20, no. 5, pp. 143–151, 2013.
- [25] S. Y. Lien, T. H. Liao, C. Y. Kao, and K. C. Chen, "Cooperative Access Class Barring for Machine-to-Machine Communications," *IEEE Trans. Wirel. Commun.*, vol. 11, no. 1, pp. 27–32, 2012.
- [26] A. El Mougy, A. Kamoun, M. Ibnkahla, S. Tazi, and K. Drira, "A Context and Application-aware Framework for Resource Management in Dynamic Collaborative Wireless M2M Networks," *J. Netw. Comput. Appl.*, vol. 44, pp. 30–45, 2014.
- [27] A. S. Lioumpas and A. Alexiou, "Uplink Scheduling for Machine-to-Machine Communications in LTE-based Cellular Systems," *2011 IEEE GLOBECOM Work. GC Wkshps 2011*, pp. 353–357, 2011.
- [28] R. Ratasuk, J. Tan, and A. Ghosh, "Coverage and Capacity Analysis for Machine Type Communications in LTE," in *2012 IEEE 75th Vehicular Technology Conference*, 2012, pp. 1–5.
- [29] H. S. Dhillon, S. Member, H. C. Huang, S. Member, H. Viswanathan, and R. A. Valenzuela, "Power-Efficient System Design for Cellular-Based Machine-to-Machine Communications," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 11, pp. 5740–5753, 2013.
- [30] I. Stojmenovic, "Machine-to-Machine Communications with In-network Data Aggregation, Processing and Actuation for Large

- Scale Cyber-Physical Systems,” *IEEE Internet Things J.*, vol. 1, no. 2, pp. 122–128, 2014.
- [31] A. Bartoli, J. Hernández-Serrano, M. Soriano, M. Dohler, A. Kountouris, and D. Barthel, “Secure Lossless Aggregation over Fading and Shadowing Channels for Smart Grid M2M Networks,” *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 844–864, 2011.
- [32] I. Cha, Y. Shah, A. U. Schmidt, A. Leicher, and M. V. Meyerstein, “Trust in M2M Communication,” *IEEE Veh. Technol. Mag.*, vol. 4, no. 3, pp. 69–75, 2009.
- [33] C. H. Liu, B. Yang, and T. Liu, “Efficient Naming, Addressing and Profile Services in Internet-of-Things Sensory Environments,” *Ad Hoc Networks*, vol. 18, pp. 85–101, 2014.
- [34] R. Bruns, J. Dunkel, H. Masbruch, and S. Stipkovic, “Intelligent M2M: Complex Event Processing for Machine-to-Machine Communication,” *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1235–1246, 2015.
- [35] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, “Home M2M Networks: Architectures, Standards, and QoS Improvement,” *IEEE Commun. Mag.*, vol. 49, no. April, pp. 44–52, 2011.
- [36] S. J. Jung, R. Myllyla, and W. Y. Chung, “Wireless Machine-to-Machine Healthcare Solution using Android Mobile Devices in Global Networks,” *IEEE Sens. J.*, vol. 13, no. 5, pp. 1419–1424, 2013.
- [37] S. J. Jung and W. Y. Chung, “Non-Intrusive Healthcare System in Global Machine-to-Machine Networks,” *IEEE Sens. J.*, vol. 13, no. 12, pp. 4824–4830, 2013.
- [38] P. Arce, C. Maureira, R. Bonvallet, and C. Fernández, “Forecasting High Frequency Financial Time Series Using Parallel FFFN with CUDA and ZeroMQ,” in *2012 9th Asia-Pacific Symposium on Information and Telecommunication Technologies*, 2012, pp. 1–5.
- [39] F. Kanayama, Y. Nishibayashi, Y. Yonezawa, and Y. Doi, “Development of Autonomous Power Electronics Products with Communication Middleware,” in *2014 IEEE International Conference on Smart Grid Communications*, 2014, pp. 61–66.
- [40] A. Shakhimardanov, N. Hochgeschwender, M. Reckhaus, and G. K. Kraetzschmar, “Analysis of Software Connectors in Robotics,” in *IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 1030–1035.
- [41] M. Süstrik, “ZeroMQ,” *Archit. Open Source Appl.*, vol. 2, pp. 1–13, 2015.
- [42] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. Sebastopol, USA: O’Reilly Media, Inc., 2013.



Z. Meng (M’16) received his B.Sc. degree on Instrumentation and Measurement from Sichuan University, Chengdu, China, in 2006, M.Sc. on Instrumentation and Measurement from Beihang University, Beijing, China, in 2009, and the Ph.D. on computer science from University of Huddersfield, Huddersfield, UK, in 2014. He then joined

the University of Manchester, the UK as a research associate in June 2014.

His research interests include embedded electronics and real-time software, sensors, instrumentation and measurement, pervasive and mobile computing, M2M communications and digital manufacturing.



Z. Wu (M’95-SM’05) obtained his BSc degree on Radio Technology from the Northeast University, China in 1983, and Ph.D. degree on Antennas and Propagation from the University of Birmingham in 1988. He was also awarded a DEng degree by the University of Birmingham in 2003. During 1988–1991, he was appointed as a Research Fellow at the University of Birmingham. In 1992, he joined UMIST as a lecturer. He now serves as a professor of antenna and propagation at the University of Manchester.

His research interests include microwave sensors, RF and microwave tomography, WSN, IoT, and techniques and systems for industrial applications, etc.



C. Muvianto received his Ph.D. degree in the University of Manchester, UK in 2012. He then worked as a research associate in University of Manchester from 2013 to 2014. He is now a lecturer in University of Mataram, Indonesia.

His research interests include electromagnetic field, antennas and propagation, communication systems, RF and microwave devices, and microwave sensors and systems.



J. Gray received his Ph.D. in Queens University Belfast, UK. He spent five years in the aircraft industry before being appointed a lecturer in Electrical Engineering at the University of Manchester where he specialized in aspects of control and instrumentation. In 1979 he was appointed to a Chair of control engineering at the University of Salford and his work on integrated systems engineering led to the establishment of the DTI’s National Advanced Robotics Research Centre at Salford in 1987. He now serves as a professor in robotics and systems engineering at the University of Manchester.

His research interests include robotics, automation in food manufacturing, digital manufacturing, etc.