Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Conceptual model of real-time IoT systems*

Bo YUAN†1,2, De-ji CHEN†‡1,3, Dong-mei XU4, Ming CHEN5

1Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
2School of Electronics and Information Engineering, Jinggangshan University, Ji'an 343009, China
3MOE Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai 201804, China
4China Electronics Standardization Institute, Beijing 100007, China
5Chinesisch-Deutsche Hochschule for Angewandte Wissenschaften, Tongji University, Shanghai 201804, China
†E-mail: yuanbo@tongji.edu.cn; dejichen@tongji.edu.cn

Received Feb. 26, 2019; Revision accepted Apr. 7, 2019; Crosschecked Oct. 14, 2019

Abstract: We address a special kind of Internet of Things (IoT) systems that are also real-time. We call them real-time IoT (RT-IoT) systems. An RT-IoT system needs to meet timing constraints of system delay, clock synchronization, deadline, and so on. The timing constraints turn to be more stringent as we get closer to the physical things. Based on the reference architecture of IoT (ISO/IEC 30141), the RT-IoT conceptual model is established. The idea of edge subsystem is introduced. The sensing & controlling domain is the basis of the edge subsystem, and the edge subsystem usually must meet the hard real-time constraints. The model includes four perspectives, the time view, computation view, communication view, and control view. Each view looks, from a different angle, at how the time parameters impact an RT-IoT system.

Key words: Internet of Things (IoT); Real-time system; Conceptual model; View; Hard/Soft real-time
https://doi.org/10.1631/FITEE.1900115 **CLC number:** TP393

1 Introduction

Nowadays smart embedded devices are connected to the Internet; the rising Internet of Things (IoT) links the devices/applications that were previously isolated. In addition, embedded devices with real-time properties (e.g., strict timing and safety requirements) require a guaranteed interaction between cyber and physical worlds. These devices are used to monitor and control physical systems and processes in many domains, e.g., manned and unmanned vehicles (including aircraft, spacecraft, unmanned aerial vehicles (UAVs), and self-driving cars), critical infrastructures, and process control systems in industrial plants. Smart systems are embedded systems which include sensing, actuation, and control. Smart systems are the gate to a smart society. Billions of smart devices and systems will make up the IoT.

A real-time system is a special kind of computer system. Its design and development are different from those of conventional computer systems. The disciplines involved are real-time operating systems (OSs), embedded development process, etc. Multitasking is a main service provided by the real-time OS. The scheduling algorithm is applied in many IoT applications, and developers use a real-time OS in the application development of IoT (Abdelsamea et al., 2016). A real-time IoT (RT-IoT) system is first an IoT system; we can study it under the guidance of the IoT reference architecture (IoT-RA), document ISO/IEC 30141. To some extent, some IoT systems exhibit real-time aspects as they interact with the physical world. We use the term "real-time IoT" to indicate the

‡ Corresponding author

* Project supported by the MITT Intelligent Manufacturing Project of China, the Study of Interconnection Standard and Experimental Verification in the Intelligent Manufacturing Plant for Naval Architecture and Marine Engineering, and the Science and Technology Program of Jiangxi Province, China (No. 20161BBE50062)

ORCID: Bo YUAN, http://orcid.org/0000-0001-7916-6879; De-ji CHEN, http://orcid.org/0000-0002-7838-9576

IoT systems that are also real-time systems, i.e., behaving under timing constraints. Likewise, we need to pay special attention to an RT-IoT system; many of its important aspects are not sufficiently described in the IoT-RA. As a matter of fact, some categories of RT-IoT systems, such as Industrial IoT (IIoT) systems (Hossain and Muhammad, 2016) and cyber-physical systems (CPSs) (Lin and Shu, 2010), have drawn much interest. Jin (2017) explored which CPS research topics are related to the emerging information technology (IT) trends and investigated how CPS technologies have been implemented in the industry. Lu (2017) proposed a potential framework of CPS systematically. Many IoT practitioners ignore real-time aspects and consider IoT simply as an extension of the Internet.

Industry 4.0 systems, Industrial Internet systems, CPS, and the so-called IT+OT (operation technology) systems are different kinds of RT-IoT systems; they all exhibit real-time requirements. In the “Implementation Proposals for Germany Industry 4.0,” Industry 4.0 (I4.0) was described as follows: “Essentially, Industry 4.0 means to integrate the cyber-physical system into the manufacturing and logistic industry and use the Internet of Things and Internet of Services in the production process. This will create profound influence on the value realization, business mode, downstream services and work organization.” In the White Paper for the Framework of the Cyber-Physical System of the National Institute of Standards and Technology, it was pointed out that the cyber-physical system and the Industrial Internet (II) are similar/relevant concepts. There is a large part of overlay in the aspects of the cyber-physical system and IoT. CPS is the next generation embedded system; an embedded system is a real-time system. We can thus say CPS is a real-time system of connected things, i.e., an RT-IoT system.

2 Real-time IoT system

2.1 Real-time system

A real-time system is one whose basic specification and design correctness arguments must include its ability to meet its timing constraints (Kim and Sang, 1995). This implies that its correctness depends not only on logical correctness, but also on timeliness of its actions. To function correctly, it must produce a correct result within a specified time, called deadline. In these systems, an action performed too late (or too early) may be useless or even harmful, even if it is functionally correct. If timing requirements coming from certain essential safety-critical applications are violated, the results can be catastrophic. They may cause serious damage to the system or to its environment, including injury or even death of people involved. These are called hard real-time systems. By contrast, there are applications that have deadlines which are noncritical. For example, one can define a transmission as failure if voice packets are not delivered by a certain deadline during a teleconference session. However, in this application, such a failure will not be catastrophic. This type of system is called a soft real-time system.

2.2 Conceptual model

A new method for energy-saving and emission-reduction (ESER) life cycle assessment (LCA) based on IoT and bill of material (BOM) was proposed in Tao et al. (2014). A four-layer (i.e., perception access layer, data layer, service layer, and application layer) ESER LCA system based on IoT and BOM was designed and presented, as well as the key technologies and functions in each layer. Catarinucci et al. (2015) proposed a novel, IoT-aware, smart architecture for automatic monitoring and tracking of patients, personnel, and biomedical devices in hospitals and nursing institutes. A real-time information capturing and integration architecture of the Internet of Manufacturing Things (IoMT) was presented to provide a new paradigm by extending the techniques of IoT to the manufacturing field (Zhang et al., 2015). Based on the IoT-RA in the International Electro Technical Commission (IEC) standard 30141, we define our conceptual model.

In IEC standard 30141 (ISO/IEC, 2018), an IoT system is grouped into six domains. They are user domain (UD), physical entity domain (PED), sensing & controlling domain (SCD), operations & management domain (OMD), access & communication domain (ACD), and application & service domain (ASD).

Fig. 1 is the conceptual model of an RT-IoT system. First, an RT-IoT system is still an IoT system. Its conceptual model is extended from that of the six-domain model. A typical RT-IoT system is a system of

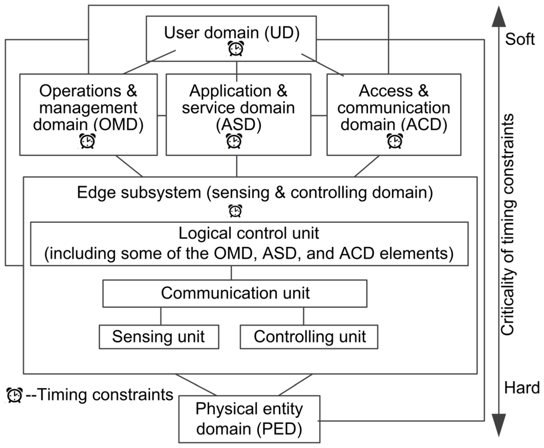
User domain (UD)
Operations & management domain (OMD)
Application & service domain (ASD)
Access & communication domain (ACD)
Edge subsystem (sensing & controlling domain)
Logical control unit (including some of the OMD, ASD, and ACD elements)
Communication unit
Sensing unit
Controlling unit
--Timing constraints
Physical entity domain (PED)
Soft
Criticality of timing constraints
Hard

Fig. 1 Conceptual model of a real-time Internet of Things (IoT) system

systems. The subsystem operates independently. As an IoT subsystem, the edge subsystem includes some elements of other domains. Timing constrains are the major characteristics of an RT-IoT system.

Each domain and each functional component must have a sense of time. Their local clock must be synchronized with a single master clock in the system. On top of this, the timing constraints can be defined and satisfied. This SCD interacts directly with the physical world, and usually is of hard real-time constraint. The timing constraint softens as we move further away from the PED.

2.3 Examples of real-time IoT systems

IIoT systems are typical RT-IoT systems. The II Consortium has launched many test beds (http://www.iiconsortium.org/test-beds.htm). They are representative of RT-IoT systems. There are also hard RT-IoT systems in smart homes. For example, if smoke detectors are part of an IoT system, they must be real-time systems. When fire breaks out, the smoke detectors must send the alarm information to people within the area and firefighters and police in distant places. The fire-fighting equipment must be activated.

3 Characteristics of an RT-IoT system

3.1 Real-time computing

Embedded real-time systems are mostly in the form of small scales and various types. Therefore, there is no uniform system standard corresponding to them. There have been few uniform standards. For computer operating systems, IEEE 1003 (ISO/IEC 9945), Standardized Portable Operating System Interface of UNIX (POSIX) Specification, defines the portable operating system interface, whereas there is no standard for real-time systems. Many commercial or open-source real-time operating systems support priority and common process scheduling algorithms, but calling interface and parameter combinations are not the same. For RT-IoT systems, versatility and interactivity become especially important. Without standards in this area, the development and operation of RT-IoT, a large-scale heterogeneous system, is difficult.

3.2 Real-time network

The support for real-time communication networks is lacking, and its importance has been increasingly recognized. In 2015, the Internet Engineering Task Force (IETF) established the DetNet Working Group. DetNet means a deterministic network, providing the worst-case bounds of delay, packet loss, and jitter, to determine a definite delay. The IEEE 802.1 Time-Sensitive Task Group (TSN TG) also researches network support for time parameters.

3.3 Real-time control

In contrast, the standardization work for real-time control is relatively mature. The standards in the industry automation field have been very well segmented and specific. For example, various control algorithm parameters in the process industry are standardized. However, there are few uniform standards across the industries. The size, complexity, and popularity of RT-IoT systems are evoking the experts from different fields to research the internal and external interconnections among IoT systems.

3.4 Feedback loop

Although Lindberg and Arzen (2010) proposed a feedback control method for the cyber-physical system, they assumed that the task is running continuously until the resource is lacking, and that it is suitable for the processing of streaming media. In most cases, an RT-IoT system involves control loops. Gao and Lu (2012) built a new task model for hard RT-IoT systems with feedback tasks and non-feedback tasks, and presented a new response time analysis method. Sensor measurement of the physical world is used in the decision-making controller module, which drives the actuator to modify the physical entities. The changes of these physical entities are measured by the sensors again, thus forming a continuous closed loop. This is common in most automation systems, programmable logic controllers (PLCs), or distributed control systems (DCSs). People have been developing and practicing automation systems for a long time. Yet, a common standard that allows the IoT systems to interface with automation systems such as PLC or DCS is missing with regard to IoT systems, unnecessarily placing the RT-IoT system in isolated information islands. Fig. 2 shows a typical feedback loop in the IoT PED domain.

3.5 Complexity

An RT-IoT system is usually very complex, being a system of subsystems. A common language is needed to exchange data between different systems and different subsystems, which is one of the basic requirements. IoT is a very complex heterogeneous network, including the connections among various types of networks through various communication technologies (Xu LD et al., 2014).

In Fig. 3, the left-hand side is the domain model from ISO/IEC 30141. An RT-IoT system may contain multiple sensing and controlling units, being a system of systems; these multiple units work cooperatively to provide services. The subsystem units are shown as dots on the right-hand side in Fig. 3. Each unit works independently for a specific task; meanwhile, they work coordinately via other shared domains.

4 Viewpoints

In this section we look at an RT-IoT system from different perspectives and describe its properties and requirements.

4.1 Time view

Palattella et al. (2013) advocated the use of an IEEE/IETF standardized IoT architecture along with a recently introduced data-centric scheduling algorithm known as the traffic-aware scheduling algorithm (TASA). Applying graph theoretical tools to the multi-channel, time-synchronized, and duty-cycled nature of TASA, they rigorously derived the minimum number required for active slots (impacting end-to-end delays) and network duty-cycle (impacting lifetime).

Fig. 4 is the time view of an RT-IoT system. The components of an RT-IoT system should have a common understanding of the clock, although the precision may differ. Due to the high bandwidth requirements and stringent delay constraints of multiuser wireless video transmission applications, ensuring that all video senders have sufficient transmission opportunities before their delay deadlines is a long-term research problem. Xu J et al. (2014) proposed a novel solution that addresses this problem without detailed packet-level knowledge, which is unavailable

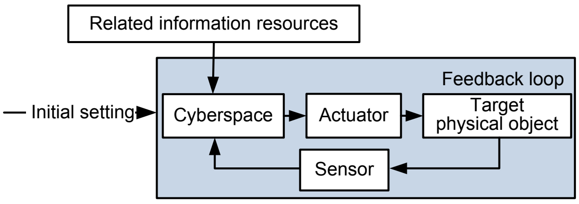

Fig. 2 Feedback loop

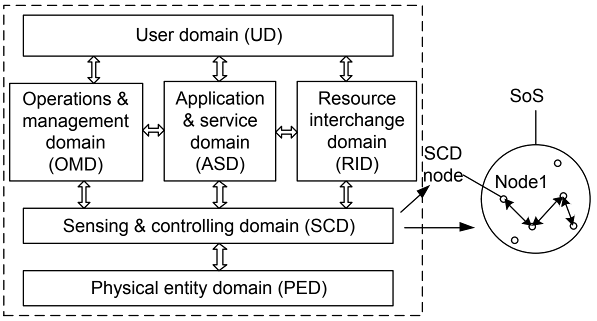
User domain (UD)
Operations & management domain (OMD)
Application & service domain (ASD)
Resource interchange domain (RID)
Sensing & controlling domain (SCD)
Physical entity domain (PED)
SCD node
SoS
Node1

Fig. 3 Internet of Things (IoT) reference architecture

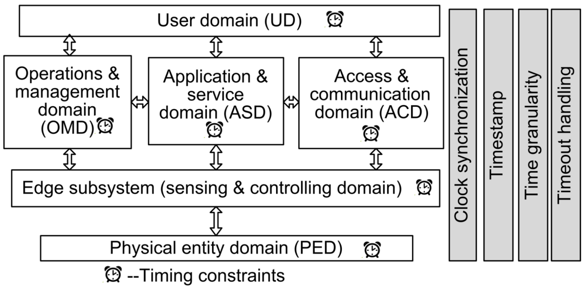
User domain (UD) ⏰
Operations & management domain (OMD) ⏰
Application & service domain (ASD) ⏰
Access & communication domain (ACD) ⏰
Edge subsystem (sensing & controlling domain) ⏰
Physical entity domain (PED) ⏰
⏰ --Timing constraints
Clock synchronization
Timestamp
Time granularity
Timeout handling

Fig. 4 Time view of a real-time Internet of Things (RT-IoT) system

at resource allocation time (i.e., prior to the actual compression and transmission). Instead, the authors translated the transmission delay deadlines of each sender's video packets into a monotonically decreasing weight distribution within the considered time horizon. Higher weights were assigned to the slots that had higher probabilities of deadline-abiding delivery.

4.1.1 Clock synchronization

In an RT-IoT system, all functional components must have a clock. The clocks are synchronized to a single master clock. The master clock may or may not be synchronized to the global standard time. There may be different synchronicity and/or drifting rates among different clocks, and the whole system meets the timing constraints.

4.1.2 Timestamp

Any task of an RT-IoT system is likely to be under strict timing constraints, so it is necessary to specify the time of completion of each step in the process, such as sensing, control, transmission, and computing. Therefore, a timestamp mark is needed in the output results of each unit to clarify the processing of each step. This is a necessary condition for realizing the RT-IoT system.

4.1.3 Time granularity

Different IoT systems can have different time granularity. A driverless car must react to external events in microseconds, whereas a smart building adjusts room temperature in minutes. Within an RT-IoT system, the actuator may act in milliseconds, but the self-diagnostics routine needs to run intermittently in the background.

4.1.4 Timeout handling

While an RT-IoT system must finish all tasks by the deadlines, often than not, a task might miss the deadline. One major feature of a real-time system is its component handling timeouts. The results can be rolling back the modification, opening a relieving valve, rebooting the system, etc.

4.2 Communication view

Li et al. (2018) reviewed the state of the art of the fifth-generation (5G) IoT, key enabling technologies, and main research trends and challenges in 5G IoT. Deterministic communication is essential for providing real-time support. Of the networks in an IoT system, field-level networks are usually fieldbuses and usually provide hard real-time support. Soft real-time support is expected at the administrative level. IETF 6TiSCH (Richardson et al., 1997; Salman et al., 2010; Thubert et al., 2013) provides a set of link layer interfaces to the higher IPv6 layer to support the time-slotted medium access control (MAC) layer, which reduces collisions, saves energy, and enables deterministic industrial applications. Devices will require intelligent routing protocols achieve intelligent device-to-device (D2D) communication. Bello and Zeadally (2016) and Jung and Lee (2017) presented an overview of how intelligent D2D communication can be achieved in IoT.

Fig. 5 is the communication view of an RT-IoT system. The networks at the proximity network layer are usually fieldbuses. The networks at the access network, service network, and user network layers may or may not provide real-time support. As technology develops, we see more deterministic networks being deployed there. The network traffic must be managed so that all traffic flows smoothly. There is no blocking for time-sensitive data. Real-time data must be delivered before it expires. Network redundancy is the means to provide guaranteed data delivery. Data must be prioritized and treated differently by the network. High priority data gets transmitted earlier and faster for high priority tasks.

4.3 Control view

Fig. 6 is the control view of an RT-IoT system. An RT-IoT system has stricter security requirements than a general IoT system, especially physical safety. The reliability of terminals in RT-IoT is the foundation of system reliability. Many terminals have been in a state of unmanned monitoring for a long time. Many devices are in a bad environment, including high temperature, coldness, poor ventilation, damp, strong electromagnetic interference, and power input interference. In the design process, to ensure that the terminal can faithfully execute the instruction of the system, the reliability of the terminal itself should be guaranteed. At the same time, it has strong adaptability to the environment and interference. Due to the simultaneous existence of certainty and uncertainty in the process of interaction with the physical environment, the traditional deterministic control mode and feedback control mode exist in the IOT control system. The ability of a control system or its components to continue to perform their essential functions when errors and failures occur is required. In the feedback control of RT-IoT, due to the influence of external environment interference, system errors, etc., the processing and communication data may produce some errors. The system needs to be inclusive of these errors, ensuring that the system can be operated in the correct logic within a certain range of errors.

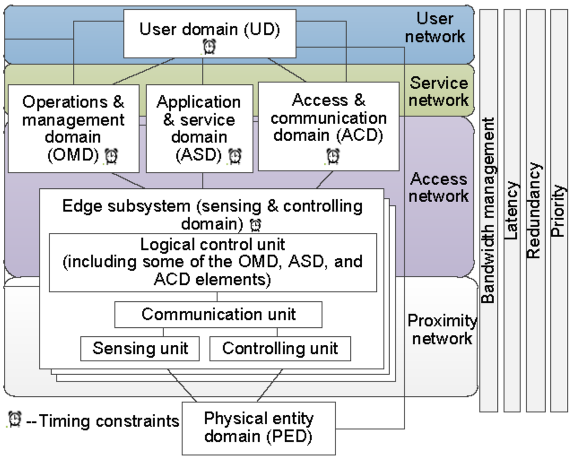
User domain (UD)
User network
Service network
Operations & management domain (OMD)
Application & service domain (ASD)
Access & communication domain (ACD)
Access network
Edge subsystem (sensing & controlling domain)
Logical control unit (including some of the OMD, ASD, and ACD elements)
Communication unit
Sensing unit
Controlling unit
Proximity network
Bandwidth management
Latency
Redundancy
Priority
--Timing constraints
Physical entity domain (PED)

Fig. 5 Communication view of a real-time Internet of Things (RT-IoT) system

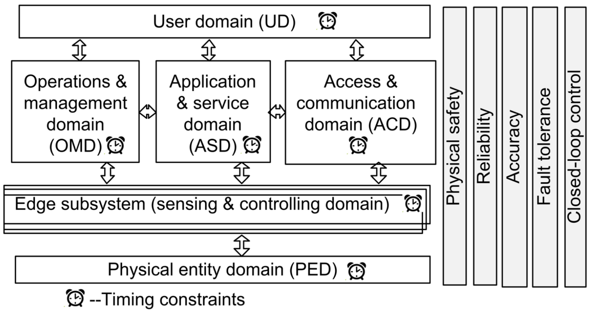
User domain (UD)
Operations & management domain (OMD)
Application & service domain (ASD)
Access & communication domain (ACD)
Edge subsystem (sensing & controlling domain)
Physical entity domain (PED)
--Timing constraints
Physical safety
Reliability
Accuracy
Fault tolerance
Closed-loop control

Fig. 6 Control view of a real-time Internet of Things (RT-IoT) system

4.4 Computation view

Fig. 7 is the computation view of an RT-IoT system. Most RT-IoT systems are implemented using a set of periodic (e.g., fixed temporal septation between consecutive instances) or sporadic (e.g., the tasks that can make an execution request at any time, but with a minimum inter-invocation interval) tasks (Mok, 1983; Buttazzo, 2008).

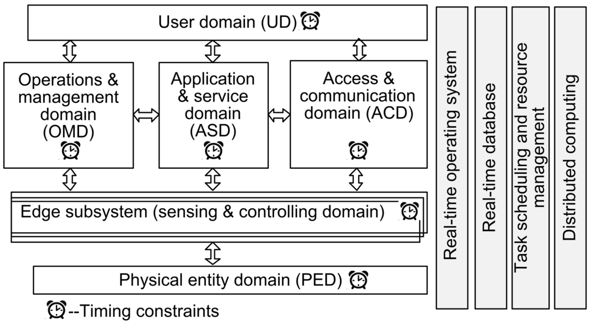
User domain (UD)
Operations & management domain (OMD)
Application & service domain (ASD)
Access & communication domain (ACD)
Edge subsystem (sensing & controlling domain)
Physical entity domain (PED)
--Timing constraints
Real-time operating system
Real-time database
Task scheduling and resource management
Distributed computing

Fig. 7 Computation view of a real-time Internet of Things (RT-IoT) system

For instance, a sensor management task that monitors the conveyor belt in a manufacturing system needs to be periodic, but the tasks that monitor the arrival of automated cars at traffic intersections are sporadic. Application tasks in the RT-IoT nodes are often designed based on Liu and Layland's model (Liu and Layland, 1973; Davis, 2014). The computation tasks in an RT-IoT system are real-time tasks. Li et al. (2014) proposed a three-layer quality of service (QoS) scheduling model for service-oriented IoT.

4.4.1 Real-time task model

A real-time task is modeled with different parameters, including computation time, period, deadline, and sometimes start time.

4.4.2 Real-time operating systems

Other than very simple ones, most of the embedded systems employ real-time operating systems, which are equivalent to Windows or Linux, but smaller, and provide more real-time support. Real-time operating systems will play an important role in any RT-IoT system.

4.4.3 Real-time databases

As systems grow, data management becomes a bigger challenge. Real-time databases are designed for real-time applications.

4.4.4 Task scheduling and resource management

A set of real-time tasks must be scheduled in a way that none will miss deadlines. Real-time scheduling involves single-CPU and multi-CPU, and is over networks. Shared resources complicate the scheduling problem. Real-time task and scheduling have been studied extensively. Transactional real-time tasks are often encountered in an RT-IoT system.

4.4.5 Distributed computation

As a system of systems, an RT-IoT system must coordinate the timing constraints among distributed components. The operating system, database, scheduling, etc. need to be adjusted accordingly.

5 Challenges in implementing an IoT system as a real-time application

Implementing a real-time system is challenging. Conventional software needs only to make sure it is functionally correct. A real-time software system must also complete its tasks in a timely manner according to the system's latency requirements; it must be developed with timing constraints considered in the initial stage. Because some developers of an IoT system usually do not have background training about real-time systems, they build it as a conventional computer system. When it comes to timing constraints, it is simply hoped that the computer runs fast enough, or when a critical task is ready the task instructions better get hold of the CPU. To make matters worse, current IoT guides, including the IoT-RA, rarely involve the real-time aspects. An IoT system developed under the guidance of the IoT-RA will have difficulty in guaranteeing timely completion of its tasks.

We give a real-world example. A Chinese company has implemented an IoT system for electric car charging stations. After arriving at a charging station, the driver swipes his/her credit card to prepay for a desired charging time. After prepaying, the outlet is enabled and the driver starts the charging. When the time is up, the outlet is turned off. The billing system runs in the cloud. Unfortunately, in the system implementation the charge control algorithm also runs in the cloud. When the charge time is up, a command from the cloud will turn the outlet off. Because the Internet is not real-time or has a long time delay due to Internet traffic congestion, sometimes the command from the cloud to the outlet is delayed or lost, giving the driver additional free charging time. Functionally, the IoT system is correct. The design is flawed and thus not able to guarantee real-time processing. A correct design can embed the total charging time in the command that turns the outlet on and let the outlet turn off by itself when the prepaid charging time is reached. This insight cannot be learned in most IoT implementation guidelines. Putting in other words, the vendor of the flawed IoT system could have legitimately claimed that it is implemented under the guidance of the IoT-RA.

The problem does not stop here. The IoT-RA is the result of studying IoT use cases. The documents of IoT use cases collected do not pay much attention to real-time systems. As far as we know, there is almost no serious time constraint.

6 Conclusions

We discussed mainly the real-time issue of the IoT system. First, we described the basic concepts of a real-time system and IoT, and then put forward the concept of RT-IoT and listed the basic elements. The challenge of IoT as a real-time system and the necessity of its framework have been analyzed, and the state of the art on the real-time property of IoT has been introduced. The RT-IoT emphasizes the time constraints in its implementation. In the RT-IoT, it is possible to put forward specific time constraints for each unit (domain or subsystem). Therefore, when designing and implementing the RT-IoT system, relevant time factors need to be considered.

Compliance with ethics guidelines

Bo YUAN, De-ji CHEN, Dong-mei XU, and Ming CHEN declare that they have no conflict of interest.

References

Abdelsamea MHA, Zorkany M, Abdelkader N, 2016. Real time operating systems for the Internet of Things, vision, architecture and research directions. World Symp on Computer Applications & Research, p.72-77. https://doi.org/10.1109/WSCAR.2016.21

Bello O, Zeadally S, 2016. Intelligent device-to-device communication in the Internet of Things. *IEEE Syst J*, 10(3): 1172-1182. https://doi.org/10.1109/JSYST.2014.2298837

Buttazzo GC, 2008. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications.

Springer Publishing Company, New York.

Catarinucci L, de Donno D, Mainetti L, et al., 2015. An IoT-aware architecture for smart healthcare systems. *IEEE Int Things J*, 2(6):515-526. https://doi.org/10.1109/JIOT.2015.2417684

Davis RI, 2014. A review of fixed priority and EDF scheduling for hard real-time uniprocessor systems. *SIGBED Rev*, 11(1):8-19. https://doi.org/10.1145/2597457.2597458

Gao ZG, Lu HJ, 2012. Research on schedulability analysis for hard real-time systems in Internet of Things. *Telecommun Sci*, 28(1):92-97 (in Chinese). https://doi.org/10.3969/j.issn.1000-0801.2012.01.018

Hossain MS, Muhammad G, 2016. Cloud-assisted industrial Internet of Things (IIoT)-enabled framework for health monitoring. *Comput Netw*, 101:192-202. https://doi.org/10.1016/j.comnet.2016.01.009

ISO/IEC, 2018. Internet of Things (IoT)—Reference Architecture. ISO/IEC 30141:2018. National Standards of America.

Jin HK, 2017. A review of cyber-physical system research relevant to the emerging IT trends: Industry 4.0, IoT, big data, and cloud computing. *J Ind Integr Manag*, 2(3): 1750011. https://doi.org/10.1142/S2424862217500117

Jung H, Lee IH, 2017. Performance analysis of three-dimensional clustered device-to-device networks for Internet of Things. *Wirel Commun Mob Comput*, 2017: 9628565. https://doi.org/10.1155/2017/9628565

Kim YK, Sang HS, 1995. Predictability and consistency in real-time database systems. In: Son SH (Ed.), Advances in Real-Time Systems. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Li L, Li SC, Zhao SS, 2014. QoS-aware scheduling of services-oriented Internet of Things. *IEEE Trans Ind Inform*, 10(2):1497-1505. https://doi.org/10.1109/TII.2014.2306782

Li SC, Xu LD, Zhao SS, 2018. 5G Internet of Things: a survey. *J Ind Inform Integr*, 10:1-9. https://doi.org/10.1016/j.jii.2018.01.005

Lin F, Shu SL, 2010. A review on cyber-physical systems. *J Tongji Univ*, 38(8):1243-1248 (in Chinese). https://doi.org/10.3969/j.issn.0253-374x.2010.08.026

Lindberg M, Arzen KE, 2010. Feedback control of cyber-physical systems with multi resource dependencies and model uncertainties. 31st IEEE Real-Time Systems Symp, p.85-94. https://doi.org/10.1109/RTSS.2010.14

Liu CL, Layland JW, 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J ACM*, 20(1):46-61. https://doi.org/10.1145/321738.321743

Lu Y, 2017. Cyber physical system (CPS)-based Industry 4.0: a survey. *J Ind Integr Manag*, 2(3):1750014. https://doi.org/10.1142/S2424862217500142

Mok AK, 1983. Fundamental Design Problems of Distributed Systems for the Hard-Real-Time Environment. Massachusetts Institute of Technology, Cambridge, MA, USA.

Palattella MR, Accettura N, Grieco LA, et al., 2013. On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH. *IEEE Sens J*, 13(10):3655-3666. https://doi.org/10.1109/JSEN.2013.2266417

Richardson M, Watteyne T, Thubert P, et al., 1997. IPv6 over the TSCH Mode of IEEE 802.15.4e. https://www.ietf.org/proceedings/95/slides/slides-95-6tisch-0.pdf

Salman N, Rasool I, Kemp AH, 2010. Overview of the IEEE 802.15.4 standards family for low rate wireless personal area networks. 7th Int Symp on Wireless Communication Systems, p.701-705. https://doi.org/10.1109/ISWCS.2010.5624516

Tao F, Zuo Y, Xu LD, et al., 2014. Internet of Things and bom-based life cycle assessment of energy-saving and emission-reduction of products. *IEEE Trans Ind Inform*, 10(2):1252-1261. https://doi.org/10.1109/TII.2014.2306771

Thubert P, Watteyne T, Palattella MR, et al., 2013. IETF 6TSCH: combining IPv6 connectivity with industrial performance. Proc 7th Int Conf on Innovative Mobile and Internet Services in Ubiquitous Computing, p.541-546. https://doi.org/10.17487/RFC7554

Xu J, Andrepoulos Y, Xiao YZ, et al., 2014. Non-stationary resource allocation policies for delay-constrained video streaming: application to video over Internet-of-Things-enabled networks. *IEEE J Sel Areas Commun*, 32(4): 782-794. https://doi.org/10.1109/JSAC.2014.140410

Xu LD, He W, Li SC, 2014. Internet of Things in industries: a survey. *IEEE Trans Ind Inform*, 10(4):2233-2243. https://doi.org/10.1109/TII.2014.2300753

Zhang YF, Zhang G, Wang JQ, et al., 2015. Real-time information capturing and integration framework of the Internet of Manufacturing Things. *Int J Comput Integr Manuf*, 28(8):811-822. https://doi.org/10.1080/0951192X.2014.900874