



A Framework For Integration of A Patients' Body Area Network With Iot

DAWOOD A. KHAN

Department of Computer Sciences, Univesity of Kashmir (North Campus)

Abstract

In this paper, we give a framework for integration of patients' Body Area Network with IoT. We also discuss the enabling technologies that may help with the proliferation of the IoT in health care, besides mitigating various interoperability challenges in a health care IoT. We use a health care use-case of an artificial pancreas for diabetic patients to discuss our framework. We describe the framework as a formal model of a health care IoT, which we map onto the components of a proposed end-to-end, closed-loop health-care IoT architecture. In this paper, we also discuss dependability in a health care IoT. As such, we describe why certification, standardisation, and dependability should be central for a health care IoT.



Article History

Received: 24 June 2017
Accepted: 12 July 2017

Keywords

IoT, diabetics, formal model, dataflow, Body Area Network, dependability.


Introduction

The term Internet of Things (IoT) refers to the interconnecting of physical devices over the internet. The basic idea is the integration of physical things (embedded devices with sensors and actuator) to the cyber-world (internet). An example of such a system in a health-care is a patient attached to various physiological sensors, which continuously and consistently monitor patients' vital signals. The sensors on the human body, form a Body Area Network (BAN), may measure various physiological changes in patients, in a location independent manner. The information collected by the sensors is transmitted to an external processing unit (e.g. a server in the cloud) via a gateway for storage, visualisation, analytics (i.e. a healthcare IoT system)

—for detection of some medical conditions. The patient data in a healthcare IoT system may be used to analyse and detect medical conditions and thus we can: mitigate frequent hospitalisations, and predict an onset of attacks in chronic patients. Moreover, healthcare IoT systems may be used to provide health services to retirement homes or homes for eldercare, geographically remote or inaccessible areas. Besides, the economic potential of IoT for healthcare is promising with a projected economic impact of about \$1.1 Trillion per year by 2025, see Exhibit 8 on page 39 of¹ for detail. Furthermore, adoption of IoT for healthcare may lead to a reduction of up to 20% in disease burde¹. Therefore, healthcare IoT not only has an economic potential but also —importantly— has

CONTACT Dr. Dawood A. Khan ✉ dawood.khan@uok.edu.in 📍 Department of Computer Sciences, Univesity of Kashmir (North Campus)

© 2017 The Author(s). Published by Enviro Research Publishers

This is an  Open Access article licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>), which permits unrestricted NonCommercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

To link to this article: <http://dx.doi.org/10.13005/ojcs/10.03.11>

a potential to directly impact a person's health by improving wellness (e.g. through analytics based feedback), monitoring (i.e. a person's physiological parameters), treatment (e.g. drug administration).

In this paper, we give a modular architecture of an end-to-end healthcare IoT and show an integration of patient Body Area Network (BAN) to IoT. We give a formal description of a healthcare system based on a case study. We also discuss the dependability requirements in a healthcare IoT system. The rest of this paper is organized as follows. In Section 2, we present a brief review of works done to integrate the IoT in healthcare. In Section 3, we present a problem description. In Section 4, we present a conceptual model of an artificial pancreas and its mapping onto physical components. In Section 5, we present a data-flow (Kahn Process Networks) based formal representation of the conceptual model of artificial pancreas. In Section 6, we present a mapping of a formal model (developed in Section 5) onto the IoT. In Section 7, we present dependability in IoT for healthcare domain. Finally, in Section 8, we conclude the discussion in this paper and give a perspective for future of healthcare IoT.

Literature review

In this section, we review the literature about healthcare IoT, particularly. As such we consider previous works, which are related to the topic of

this paper, in contrast with framework discussed in this work. In², authors describe various architectural elements of a general cloud-based IoT, which is a hybrid (public/private) cloud approach. The approach, therein, is user-centric and general. However, the work only discusses a high-level architecture, without discussing the IoT protocol stack or gateway architecture or available technologies. In³, authors describe an IoT application of ambient assisted living for elderly, which is a closed-loop system (with human-in-loop). However, the approach is based on RFID (a dumb node) devices connected to the Internet via a phone network; which limits the approach to urban areas only. The approach is closed-loop (human-in-loop), therefore may not scale to a large group of persons—we discuss a closed-loop approach based on AI and machine learning. In⁴, authors describe an IoT application for a rural healthcare system. However, the authors only describe a model for wireless communication channel and not an end-to-end solution. In⁵, authors describe an open-loop healthcare IoT system. However, the authors only describe a process model of open-loop systems and an IoT protocol stack for a node. The protocol choices for an IoT stack are not discussed; the server and the gateway architectures are not described. In⁶, the authors describe an IoT and cloud-based healthcare application. However, the authors only describe functional aspects of the system. Moreover, the described application is an

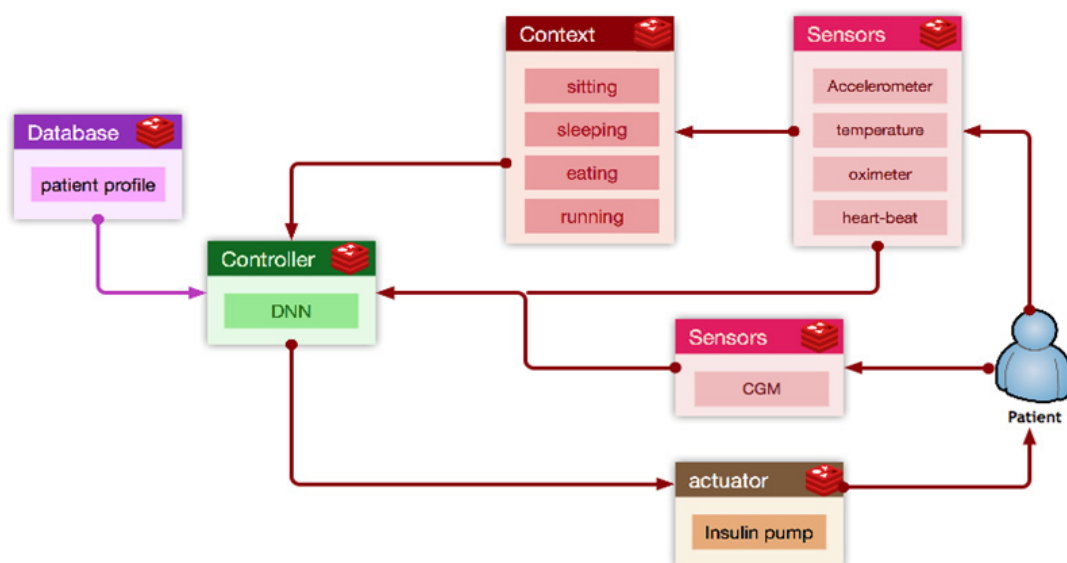


Fig. 1: Model of an Artificial Pancreas and its Components

open-loop or human-in-loop system. In⁷, the authors give a comprehensive description of an IoT and its application to various environments. The author concentrates on the business model, information package management, and architecture of the system. However, the author does not address the choice of protocols in an IoT protocol stack, gateway architecture, modelling and mapping of a problem in an IoT system. In⁸, authors describe an open-loop non-invasive glucose level sensing and monitoring in patients based on IoT. However, in contrast, we describe a closed-loop intelligent system, which has both sensing/monitoring capabilities and control capabilities. Some other works like:⁹⁻¹⁶ also contribute to the healthcare IoT, nevertheless, these works only describe certain limited aspects of an IoT healthcare system, for example gateway design¹¹, IoT and cloud¹⁰, security^{14,15}; and do not describe a complete architecture of an open source end-to-end healthcare IoT system.

Problem Description

As a use-case of a healthcare IoT, we choose to discuss the integration of a diabetic patient to a healthcare IoT. Diabetes is a metabolic disease in which body produces no insulin (Type-I) or produces inadequate insulin (Type-II) to regulate the Blood

Glucose Concentration (BGC). The unregulated BGC, as a result of diabetes, causes serious complications like renal failure, cardiovascular failure, neuropathy etc. in diabetic patients. Type-I diabetes is insulin dependent diabetes since pancreas secretes no insulin to regulate BGC. Type-II diabetes is managed by using oral medication, regulating dietary habit, exercises, along with frequent monitoring of BGC. However, Type-II patients, usually, in long-term may need to use insulin to regulate the BGC.

The BGC of a person (at a future instance of time) is dependent on factors like 1) a meal or bolus (a type of food, quantity taken etc.) person takes; 2) and on a workout (exercise, walking, etc.) a person does. Type-I diabetes patients require multiple daily BGC measurements and insulin injections, to regulate the BGC. As such, it becomes a tedious process to manage Type-I patients BGC; since this process requires frequent finger pricks to measure current BGC and then compute insulin dosage (based on factors describe above).

The integration of a diabetic (Type-I) patient to a healthcare IoT allows us to design an intelligent system which continuously monitors patients BGC

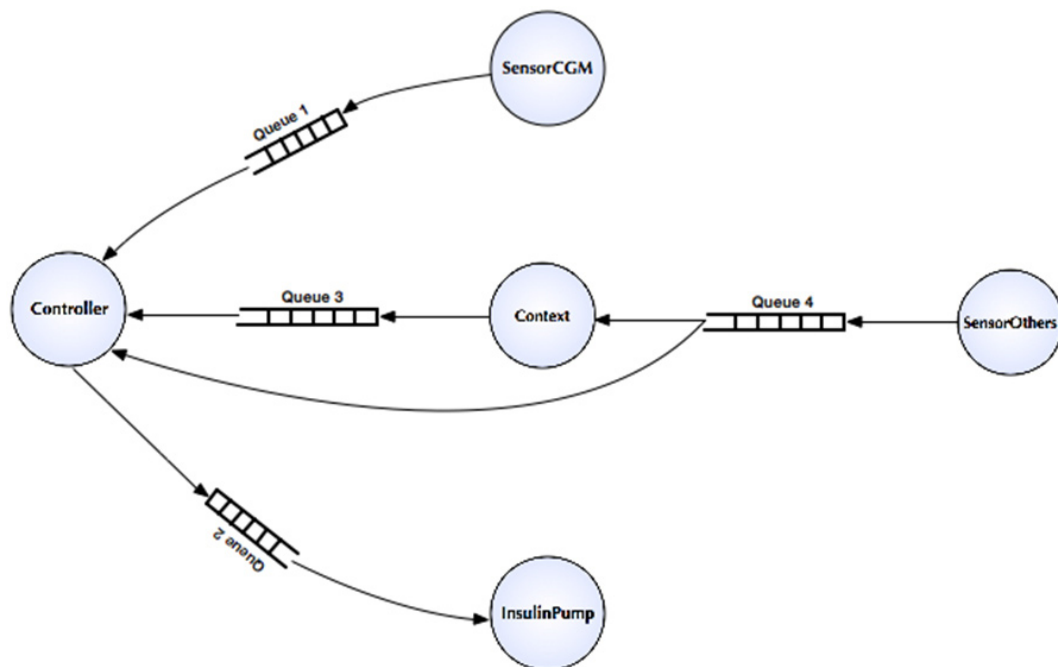


Fig. 2: KPN model of artificial pancreas deduced from Figure 1

and adjust the insulin dosage automatically; i.e. an IoT-based artificial pancreas. Functionally, IoT-based artificial pancreas shall mimic the working of the pancreas by regulating patients BGC in the normal zone (as specified for fasting and post-meals). The controller works in a closed-loop by regulating insulin-pump actuator, based on BGC from Continuous Glucose Monitor (CGM) sensor. The proposed system is composed of four parts: 1) sensor/actuator node 2) coordinator 3) communication channel 4) cloud based server for data analytics, machine learning, and feedback to the insulin pump, see Figure 1.

The key issues that we consider in this paper are:

1. To develop a formal model of an artificial pancreas, and map it to an IoT system,
2. The issues/properties that we should consider in a healthcare IoT.

Body Area Network

Body Area Network (BAN), see Figure 6: is a network of embedded systems which are capable of: sensing and/or actuation, computation, and communication (wired/wireless). These embedded devices are low-power battery operated, placed around a human body (epidermal or endodermal) to sense and act in a coordinated fashion—i.e. a node in BAN may be of wired or wireless type, however, we consider wireless BAN (WBAN) in this paper due to the need for the ambulatory, modularity, scalability (ease of adding new nodes without having to connect wires), and flexibility in a healthcare IoT systems.

We can use various topologies for networking the BAN nodes, such as Star, Cluster-tree, and Mesh. The Star topology has single point of failure, the central node (may be due to battery decay etc.), and as such the device network collapses.

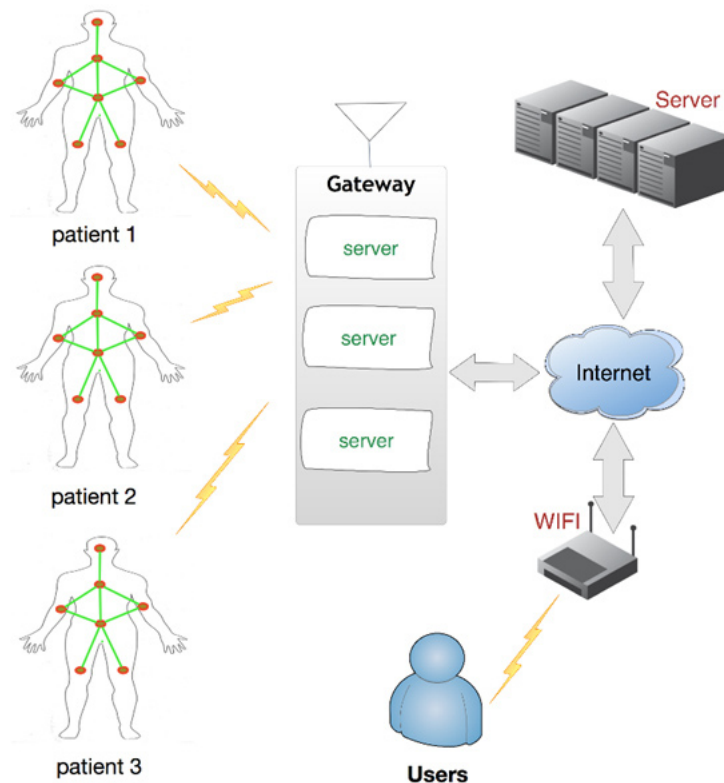


Fig. 3: entities and Components of an End-to-end Architecture For Artificial Pancreas System

Similarly, Cluster-tree may also suffer from branch pruning, due to the failure of intermediate cluster coordinators. Such failure may not cause any problem in certain networks, however, since we are developing a safety-critical application we need to use a robust network topology. Therefore, we consider mesh topology which provides robustness in terms of multiple available paths for routing packets in a network. The BAN designates one of its nodes as a BAN gateway-node (of BAN network) for external communication and all messages designated for an external network (through IoT gateway) are routed through the BAN gateway-node. For example, in Figure 6 we have seven nodes in a mesh topology and one of the nodes is acting as a BAN gateway to forward the message to an IoT gateway. Furthermore, since the number of nodes attached to a patient is not arbitrarily large, the routing algorithm for BAN can converge quickly without consuming too much energy—each node is capable of measuring multiple parameters, therefore, it is reasonable to consider that we have a fairly limited number of nodes attached to a patient.

Artificial Pancreas Model

Artificial Pancreas (AP) is safety-critical system¹ and we specify AP as a set of components that are composable together. The conceptual block diagram of a proposed artificial pancreas is given in Figure 1. The model, in Figure 1, captures the underlying principle components of IoT, i.e. sensors, actuators, computation, and communication. The essence of this composable model is the ability to capture different requirements of components like high assurance, safety-critical, performance-critical, quality-of-service etc. at the design time. Therefore, this approach may allow us to capture and model

requirements of diverse IoT systems (a simple toaster to a safety-critical node).

In Figure 1, the sensor (CGM) reads the BGC from a patient and communicates it to the controller. The sensors (e.g. accelerometer, temperature, pulse oximeter, heartbeat etc.) read the parameters and communicate those to the controller and context component. The context component, based on the sensor values, computes the context of the patient (i.e. if the patient is walking, sleeping, sitting, eating, etc.) using machine-learning algorithms¹⁷. The computed context is useful as the future BGC is human activity dependent, and as such is sent to the controller. The controller based on the inputs from context component, database component (which stores a patient's reference parameters), sensor component (CGM) computes the appropriate set point for the actuator component (insulin pump). Thereby, the controller controls the BGC of a patient in an intelligent (human independent) closed-loop manner.

Here we may use a Deep Neural Networks (DNN) controller as the control algorithm [18, 19] for the AP. The choice of DNN controller is based on the facts that: 1) integration of diabetic patients into IoT will generate big data, since there are large number of diabetic patients world wide 2) we can train better DNN controllers using these data (initially we only gather data for data analytics and to train DNN controller) 3) we can use powerful computational resources in the cloud to train the DNN 4) such controller can be robust and capable of approximating dynamic behaviour of the system to desired level of accuracy, assuming large number of patients—given these facts, we use computational resource in cloud¹⁹. Therefore, such a controller may perform better given diverse data to

Table 1: Mapping of khan process network (KPN) of figure 1 for program p of system ap. Also shows mapping on o physical resources for relisation of an ap

Hardware Mapping	Process	Function Mapping
Embedded node	Sensor CGM	X1 = Sensor CGM()
Embedded node	Insulin Pump	Insulin Pump(X2)
Embedded node(s)	Sensor Other	X4 = Sensor Other()
Server Cloud	Controller	X2 = Controller(X1, X3, X4)
Server Cloud	Context	X3 = Context(X4)

train the controller—yet it will act in a personalised manner, based on a patient specific set-points in the database component. Moreover, the data analytics (point 2) allows us to combine, integrate, and analyse all the data, which can provide better insight (visualisations, hidden patterns, unknown dependencies etc.) for healthcare providers and policy makers.

An IoT-based AP is a distributed system, as described in Figure 1, the mapped components run on different networked computational resources. The interaction between these computing resources (referred to as a node in rest of the paper) is coordinated by message passing.

KPN based formal model an Artificial Pancreas

We now describe a KPN model— which is a data-flow model— for correct specification of an AP, to capture the structural (i.e. physical infrastructure like nodes, network etc.) and for behavioural (virtual infrastructure like protocols, tasks, processes etc.) semantics. KPN, see²¹, model consists of processes and channels. The components of a KPN are represented as a directed graph $G=(V,E)$, where

nodes V is a set of all processes and arcs E is a set of all unidirectional FIFO (First-In-First-Out) communication channels, see Figure 2—note that we have already identified the components of an AP in Section 4. KPN, see²⁰, is a parallel computation model (with a deterministic behaviour) where any application can be modelled as a set of concurrent independent processes with unbounded FIFO buffers in between. The processes read their inputs from a FIFO and write their outputs to a FIFO, connected to a process in graph G .

The writes to a channel are non-blocking (i.e. process continues to write data into FIFO, unconditionally) and the reads are blocking (i.e. read is conditional; the unavailability of data in FIFO causes read to wait until the data is available in FIFO). The processes in KPN may or may not have an input or an output channel (i.e. no read or write); for example, SensorOther¹ and SensorCGM processes do not read from any input channel, and InsulinPump process does not write to any output channel. The mapping function, see²¹, of a program (say P) associated with AP, is given in Table 1—along with their mappings onto a hardware component.

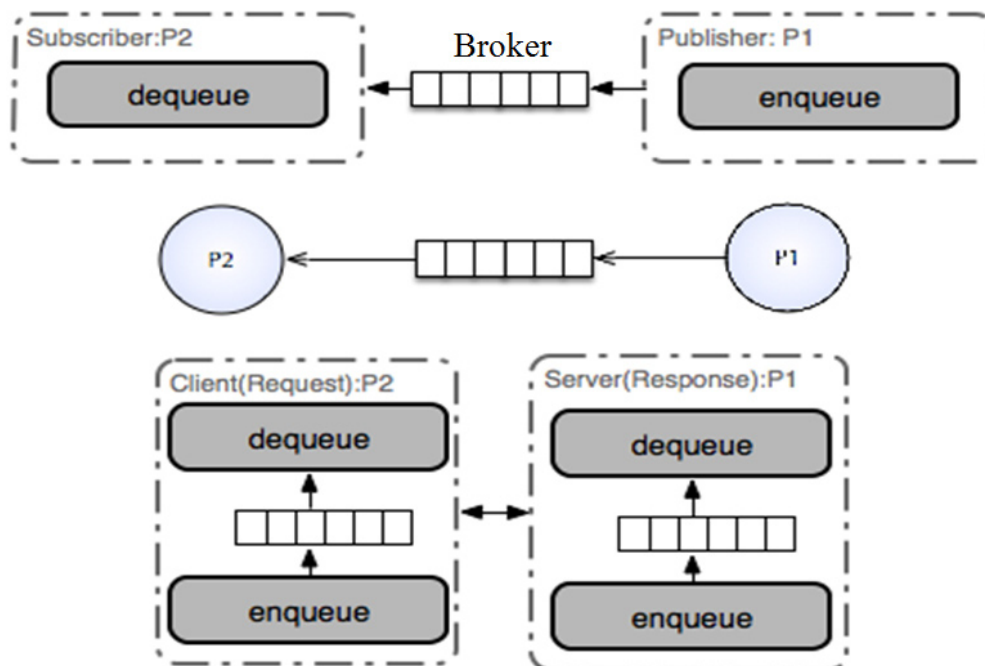


Fig. 4: Example mapping of a KPN network to P/S and R/R messaging models

The SensorOther could be mapped to one node or many nodes; for a multi-node case each node senses a unique or a group of characteristic (e.g. temperature, pressure etc.). Here X1, X2, X3, X4 values are on channels which are generated by processes: SensorCGM, Controller, Context, SensorOther respectively, and put in corresponding Queues: Queue1, Queue2, Queue3, and Queue4. In²¹, authors prove various important properties of a program P, like boundedness, monotonicity, and determinism, which are important for us, since we are redesigning a safety-critical system. Moreover, the processes in a KPN model are autonomous, as such the processes need not know about other processes. The processes that communicate with each other simply do so using a queue.

IoT Implementation description of an AP

A general architecture of a healthcare IoT system is depicted in Figure 3. The BAN on patients is connected to a gateway, which is connected to the Internet. This allows BAN nodes to communicate measurements to a server on the Internet via a gateway node (which understands communication protocol of both BAN and Internet). We now consider the mapping KPN model onto an IoT.

The formal model of an artificial pancreas we described abstracts many details like underlying network protocols used, type of communication hardware etc. KPN provides us with a simple read and write based formalism for processes of a system. We map the abstract read and write formalism of KPN onto an IoT protocol stack, which is standardised—this is possible if they have an interface similar to KPN's read and write formalism. The standard application layer IoT protocols (like mqtt, mqtt-sn, CoAp, ZeroMq etc.) use such a formalism as a messaging model¹ called: Publish/Subscribe (P/S) and Request/Response (R/R); for communication between interested entities. These messaging models function by decoupling the sender from the receiver.

The use of a particular messaging model depends on an application, and the adoption depends on a matrix given in Table 2, see²². For example, we may use R/R messaging model for a CGM sensor node as it needs to reliably stream the data to a

server, and we may use P/S messaging model for a temperature sensor as we may sample aperiodically to communicate the data to a server. Nevertheless, some IoT systems may require hybrid messaging models. Therefore, we use P/S messaging model for nodes, which do not need reliability and timeliness, and R/R messaging model for nodes, which do not need mobility and adaptability.

The possible solutions for a hybrid messaging model are: 1) use of R/R adaptation layer for P/S, see²²; 2) allow nodes to implement either P/S or R/R messaging model and use a gateway implementing a multiple-protocol stack (i.e. both P/S and R/R) joining two messaging paradigms, e.g. emqttd²³ implements both; 3) use only one of the messaging model (R/R or P/S)—depending on the features which are required most. The first two choices seem reasonable and shall be explored further.

The node, in P/S messaging model, publishes (sends) a message to a central entity called a broker, where it is stored against a topic in the broker. A broker is a software module, which sits in between a publisher and subscriber. A Broker receives all the messages from the publishing nodes and sends them to the nodes subscribing the topic, see Figure 6. The node, which is interested in receiving a message from some node, subscribes to the topic published by that node. In P/S messaging model, the publisher does not have an acknowledgement-path, as the publisher does not know if the subscriber has received a message or not—P/S is also, appropriately, called fire and forget messaging model and is 1-to-many cast. Therefore, P/S messaging model is suited for nodes that require mobility and communicate asynchronously with a gateway, see Table 2.

The R/R messaging model is a client-server model. A client requests information from a server and server replies. In R/R messaging model a client is any entity that makes a request (which could be a node or a gateway or a server) and a server is any entity, which replies (which could be a node or a gateway or a server). Therefore, R/R messaging model is more like a one-to-one model, and well suited for nodes that lack mobility and communicate synchronously with a gateway, see Table 2.

A P/S messaging model is homomorphic to KPN model. Consider a queue between two communicating processes P1 and P2 in a KPN model of Figure 5. We can map P1 (a publisher) to a function which enqueues message and we can map P2 (a subscriber) to a function which dequeues a message, from the same queue on a broker. As such, the publishers/subscribers are autonomous, like the processes in KPN model.

An R/R messaging model is also homomorphic to KPN model. However, compared to P/S messaging model, the queue is distributed between a client (Request) and a server (Response). Consider R/R case for processes P1 and P2 of KPN model in Figure 5. We assume synchronous communication channel is set up between nodes hosting processes P1 and P2—in the case of a failure a client node retries to establish a communication channel. We also assume that message transactions and queue buffers are atomic, and queues in client and server are sufficiently large. As such, we get two cases:

- P2 (client) sends a request (analogous to read of KPN model) to P1 (server). As P1 (server) receives a request it dequeues a message from its queue and sends (i.e. response) the message through the established communication channel to P2 (client). As such, read (request in this case) retains the blocking property of a KPN model—read is only blocked when a server is empty.
- P1 (client) sends a request (analogous to write of KPN model) to P2 (server). As P2 (server) receives a request it responds with an OK¹ message to P1 (client). P1 (client), after receiving the OK message, dequeues a message from its queue and writes (i.e. response) that message through the established communication channel to P2 (server). Therefore, a write (request in this

case) retains the non-blocking property of a KPN model—write is never blocked since the OK message has been received.

Using the approach describe above, enables us to map our formal model to an IoT messaging model and choose a messaging model for individual nodes depending on the requirements (see Table 2). For example, IoT protocols like mqtt, mqtt-SN use P/S messaging model and CoAp uses R/R messaging model, thus we can select an IoT protocol stack for a node based on the properties discussed in this section.

IoT Node architecture

We consider that the nodes are physical devices: which have a computational capability, measure some physical/chemical property of a patient, have some networking capability to communicate measurements, and are battery powered. Moreover, a gateway is also a node, which manages a group (cluster) of nodes and acts as an interface to other network types. A node

may act as a router (in multi-hop networks) and simply pass the messages to a destination node. The architecture of a generic node is depicted in Figure 7.

The nodes can have varying computational capabilities, form factor, power consumption, communication capabilities, cost etc.—which depend on the selection of modules used by the node designers. The choice of selecting a particular node implementation (i.e. a hardware platform and software configuration running on top of it) depends on the problem at hand, which an IoT designer will choose based on the application requirements, and the IoT stack required.

The ambulatory nodes are suitable in a healthcare IoT. As such, we may choose a low-power node that communicates wirelessly and which run an IoT

Table 2: Properties of Publish/Subscribe and Request/Response Messaging Patterns²²

Property	Efficiency	Mobility	Adaptability	Reliable delivery	Security	Timeliness	Operation mode	#of Receiver
P/S	Partial	Yes	Yes	-	partial	-	async	1
R/R	Partial	-	-	Yes	Yes	Yes	sync	1

protocol stack; for example, mqtt-SN based wireless sensor nodes. The node runs some applications; for example, a node runs an application that periodically measures BGC of a patient and communicates the measurement to the server.

Dependability

Dependability, see²⁴, is the system property that integrates such attributes as reliability, availability, safety (described above), security, survivability, maintainability. The dependability is a much-desired property in a healthcare environment—both for hardware and software components of IoT. Lack of reliability (correct continuous service) in a healthcare systems could potentially kill a patient; for example, consider a patient attached to an insulin pump (actuator node in BAN) and the node not being able to receive commands from server as such intended system behaviour is not reached—due to software, hardware or network failure. Nevertheless, achieving reliability is not sufficient, as satisfying availability is necessary as well. For example, the highly reliable system with low availability (a system working for only a limited period) will only issue commands to an insulin pump during those working periods; a patient can die due to lack of input during the periods of non-availability.

Having a reliability and availability is not sufficient. The healthcare IoT system, also, requires confidentiality (only authorised access to information)—since we are dealing with personal patient data—and integrity (only authorised system modification)—illegal modification of patient dosage can potentially kill a patient. Lastly, maintainability is also an important requirement of IoT systems in general, as it allows us to modify the system (SW/HW components), thereby removing potential issues which could have an effect on our system like security, reliability, availability etc.

In order for our system to have these dependability properties, we need to design IoT system components to be dependable from the very start, and composability of these components will make our system dependable. Therefore, certain techniques have to be followed to develop a dependable IoT components. These techniques are 1) fault prevention (preventing the fault from happening); 2) fault tolerance (function even in presence of faults); 3) fault removal (reduce faults); 4) fault forecasting (estimating present, future faults)—interested reader should see⁶⁴ for detailed discussion.

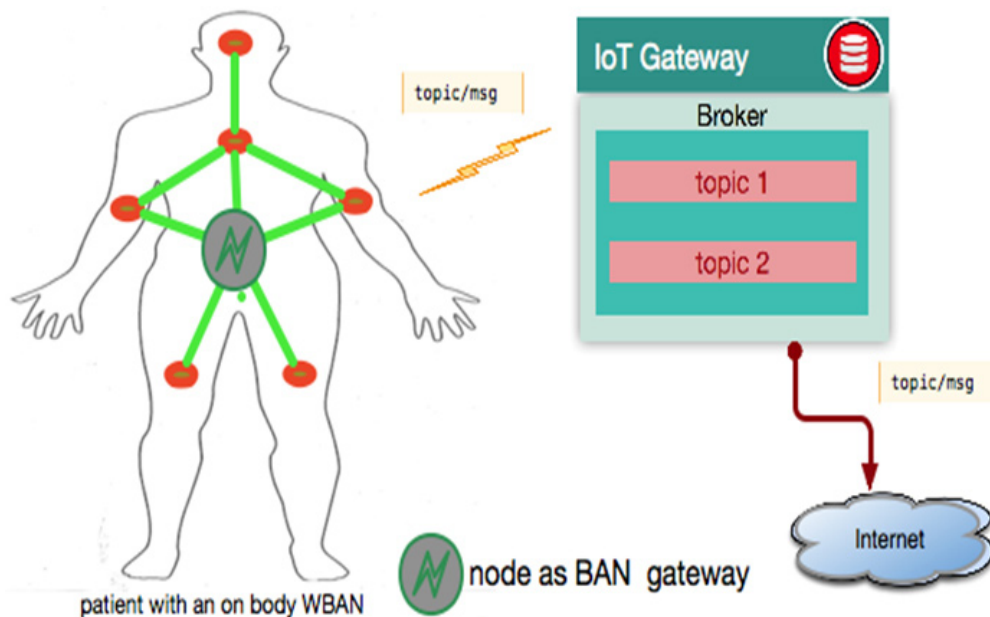


Fig. 5: Example of a BAN with 7 nodes (circles are nodes) networked together (lines show wireless links). Also showing how A ban –gateway is connected to IoT through AN IoT-gateway via wireless link

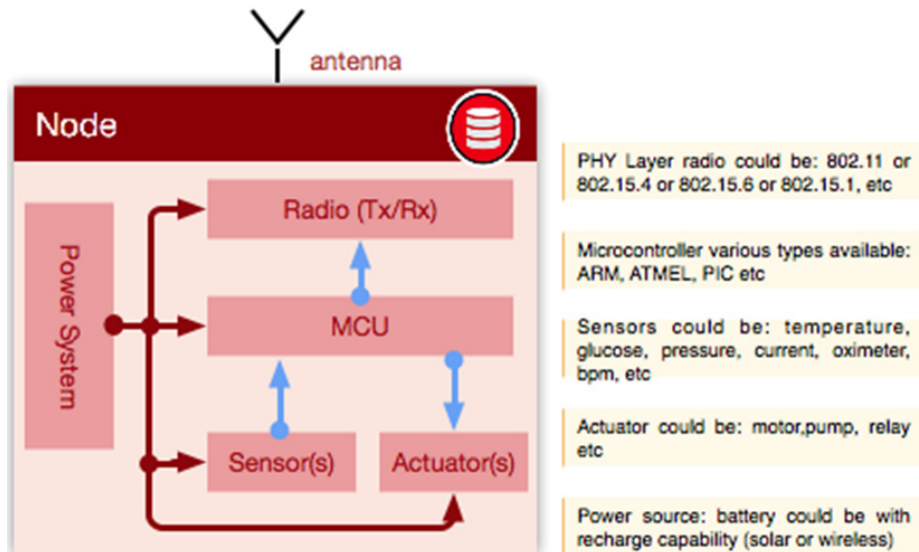


Fig. 6: Architecture On A generic NODE in an IoT, used in a ban and in a gateway

Conclusion

In this paper, we presented a formal healthcare IoT framework. This approach enables us to present an innovative convergence of scalable and interoperable system modules; since standardisation facilitates interoperability. We, also, described the patient BAN and its integration to the IoT. We, also, presented an architecture of an end-to-end healthcare IoT, along with its components. Moreover, we presented a model that can be used in a closed-loop to control the actuators on a patients (for example, to control insulin) using machine-learning algorithms and learn patients' physical activity context (for example, walking, sleeping etc.). The closed-loop (human independent) solution requires minimum intervention.

The adoption of IoT in healthcare can be an enabling factor that can mitigate challenges healthcare and disease management. Nevertheless, IP-network may be a limiting factor that may regress the proliferation of IoT. This is due to the fact that we have limited knowledge about how IP-network (bandwidth limited) behaves under the tremendous network load that unprecedented number IoT nodes may generate. Moreover, configuration management of hardware and software (like firmware, calibration bug-fixes etc.) in IoT is going to be a challenge due to the heterogeneity of hardware and software; which is important as these may lead to dependability issues in an IoT system. As such, in coming years dependability is going to be a key challenge in IoT for researchers in industry and academia.

References

- 1 Manyika, James(ed). The Internet of Things: mapping the value beyond the hype. McKinsey, 2015.
- 2 Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, no. 7 (2013): 1645-1660.
- 3 Dohr, Angelika, Robert Modre-Osprian, Mario Drobits, Dieter Hayn, and Günter Schreier. The Internet of Things for Ambient Assisted Living. *ITNG 10* (2010): 804-809.
- 4 Rohokale, Vandana Milind, Neeli Rashmi Prasad, and Ramjee Prasad. A cooperative Internet of Things (IoT) for rural healthcare monitoring and control. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless*

- VITAE), 2011 2nd International Conference on, pp. 1-6. IEEE, 2011.
- 5 Bui, Nicola, and Michele Zorzi. Health care applications: a solution based on the internet of things. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, p. 131. ACM, 2011.
- 6 Doukas, Charalampos, and Ilias Maglogiannis. Bringing IoT and cloud computing towards pervasive healthcare. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pp. 922-926. IEEE, 2012.
- 7 Pang, Zhibo. Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being. (2013). Thesis.
- 8 Istepanian, R. S. H., Sijung Hu, N. Y. Philip, and Ala Sungoor. The potential of Internet of m-health Things "m-IoT" for non-invasive glucose level sensing. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5264-5266. IEEE, 2011.
- 9 Belli, Laura, Simone Cirani, Luca Davoli, Lorenzo Melegari, Màrius Mònton, and Marco Picone. An Open-Source Cloud Architecture for Big Stream IoT Applications. *Interoperability and Open-Source Solutions for the Internet of Things* (2015): 73-88.
- 10 Raggett, Dave. COMPOSE: An Open Source Cloud-Based Scalable IoT Services Platform. *ERCIM News* 101 (2015): 30-31.
- 11 Desai, Pratikkumar, Amit Sheth, and Pramod Anantharam. Semantic gateway as a service architecture for IoT interoperability. In *2015 IEEE International Conference on Mobile Services*, pp. 313-319. IEEE, 2015.
- 12 Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.P., Riahi, M., Aberer, K., Jayaraman, P.P., Zaslavsky, A., Žarko, I.P. and Skorin-Kapov, L., 2015. Openiot: Open source internet-of-things in the cloud. *Interoperability and Open-Source Solutions for the Internet of Things*, pp.13-25.
- 13 Hiremath, Shivayogi, Geng Yang, and Kunal Mankodiya. Wearable Internet of Things: Concept, architectural components and promises for person-centered healthcare. In *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, pp. 304-307. IEEE, 2014.
- 14 Tarouco, Liane Margarida Rockenbach, Leandro Márcio Bertholdo, Lisandro Zambenedetti Granville, Lucas Mendes Ribeiro Arbiza, Felipe Carbone, Marcelo Marotta, and José Jair Cardoso de Santanna. Internet of Things in healthcare: Interoperability and security issues. In *2012 IEEE International Conference on Communications (ICC)*, pp. 6121-6125. IEEE, 2012.
- 15 Valera, Antonio J. Jara, Miguel A. Zamora, and Antonio FG Skarmeta. An architecture based on internet of things to support mobility and security in medical environments. In *2010 7th IEEE Consumer Communications and Networking Conference*, pp. 1-5. IEEE, 2010.
- 16 Mainetti, Luca, Luigi Patrono, and Antonio Vilei. Evolution of wireless sensor networks towards the internet of things: A survey. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pp. 1-6. IEEE, 2011.
- 17 Mannini, Andrea, and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors* 10, no. 2 (2010): 1154-1175.
- 18 El Youssef, Joseph, Jessica Castle, and W. Kenneth Ward. A review of closed-loop algorithms for glycemic control in the treatment of type 1 diabetes. *Algorithms* 2, no. 1 (2009): 518-532.
- 19 Cong, Shuang, and Yanyang Liang. PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. *IEEE Transactions on Industrial Electronics* 56, no. 10 (2009): 3872-3879.
- 20 Stefanov, Todor, Claudiu Zissulescu, Alexandru Turjan, Bart Kienhuis, and Ed Deprette. System design using Kahn process networks: the Compaan/Laura approach. In *Design, Automation and Test in Europe Conference and Exhibition, 2004.*

- Proceedings*, vol. 1, pp. 340-345. IEEE, 2004.
- 21** Kahn, Gilles, and David MacQueen. Coroutines and networks of parallel processes. (1976): 20.
- 22** Rodríguez-Domínguez, Carlos, Kawtar Benghazi, Manuel Noguera, José Luis Garrido, María Luisa Rodríguez, and Tomás Ruiz-López. A communication model to integrate the request-response and the publish-subscribe paradigms into ubiquitous systems. *Sensors* 12, no. 6 (2012): 7648-7668.
- 23** Emqtt.io, Feng At. EMQ - The Massively Scalable Open Source MQTT Broker. EMQ - The Massively Scalable Open Source MQTT Broker. Accessed September 10, 2016. <http://emqtt.io/>.
- 24** Avizienis, A., J. C. Laprie, and B. Randell. Fundamental Concepts of Dependability (LAAS-CNRS Report No. 01145). (2001).