SOFTWARE DEFINED HEALTHCARE NETWORKS

LONG HU, MEIKANG QIU, JEUNGEUN SONG, M. SHAMIM HOSSAIN, AND AHMED GHONEIM

ABSTRACT

With the increasingly serious problem of the aging population, creating an efficient and real-time health management and feedback system based on the healthcare Internet of Things (HealthIoT) is an urgent need. Specifically, wearable technology and robotics can enable a user to collect the required human signals in a comfortable way. HealthIoT is the basic infrastructure for realizing health surveillance, and should be flexible to support multiple application demands and facilitate the management of infrastructure. Therefore, enlightened by the software defined network, we put forward a smart healthcare oriented control method to software define health monitoring in order to make the network more elastic. In this article, we design a centralized controller to manage physical devices and provide an interface for data collection, transmission, and processing to develop a more flexible health surveillance application that is full of personalization. With these distinguished characteristics, various applications can coexist in the shared infrastructure, and each application can demand that the controller customize its own data collection, transmission, and processing as required, and pass the specific configuration of the physical device. This article discusses the background, advantages, and design details of the architecture proposed, which is achieved by an open-ended question and a potential solution. It opens a new research direction of HealthIoT and smart homes.

INTRODUCTION

The urban population has increased sharply in recent years, and this trend will continue for several years. Based on the healthcare Internet of Things (HealthIoT), this article designs an emotionally interactive system with an intelligent mechanism based on the cross-network cooperation of cloud and body area networking (BAN) [1], which can comprehensively perceive the physical and mental conditions of the user through wearable computing and BAN.

At present, HealthIoT is still in its initial phase of development and deployment. However, there is no doubt that the importance of HealthIoT in future daily life will increase extensively just like the Internet today. Although the Internet has achieved great success and changes in many aspects, it is still confronted with some problems. On one hand, intelligent control is realized by various route and management protocols, and is embedded into each single router, which makes change difficult. Since the architecture of the Internet is rigid, its development is quite slow. Moreover, depending on the interface of the supplier makes infrastructure management complex and error-prone. On the other hand, it only offers best effort service, so it holds back the development of highly personalized applications and cannot meet specific requirements for service quality and user experience.

To support sustainable development and facilitate flexible management with diverse application demands, the design of HealthIoT architecture should avoid the above problems in the future. From the IoT evolution point of view, following are some trends need to be considered for the design of future HealthIoT.

Sharing of Physical Infrastructure: Sharing means that the physical facilities in the bottom layer support multiple applications in multiple categories. The popularization of cloud computing shows the trend of sharing the infrastructure very well. Through cloud computing, application developers deploy their applications to the cloud data center, rather than establishing their own physical infrastructure. In addition to cloud computing, there is also a trend of sharing the network infrastructure, such as base stations and access points. Sharing physical infrastructure has a common advantage of reducing initial and maintenance costs. In order to get this advantage, we anticipate sharing the IoT infrastructure.

Rise of Software Defined Architecture: Software definied networking (SDN) makes flexible network control possible by separating the control and data planes. Enlightened by this advantage, SDN has been extended to the mobile access network [2] and 5G wireless network [3, 4]. The physical infrastructure will be more and more complex in the IoT age, so it is necessary to realize flexible control and management of IoT infrastructure through the SDN concept.

Popularity of the Application Programming Interface: Providing the application programming interface (API) is the trend in sharing the physical infrastructure. Cloud suppliers such as Google APP Engine provide such APIs, and network controllers such as OpenDaylight [5] also provide the northbound interface to develop the control

Long Hu and Jeungeun Song are with Huazhong University of Science and Technology.

Meikang Qiu is with Pace University and Columbia University.

M. Shamim Hossain and Ahmed Ghoneim are with King Saud University.

Ahmed Ghoneim is also with Menoulia University.

1536-1284/15/$25.00

The standard API reduces the complexity and the development cycle of deploying the new application, and meanwhile the sharing of physical infrastructure reduces the OAM costs to a large extent. These characteristics enable SDHN to support various application demands efficiently, so as to make the intelligent remote health surveillance possible.

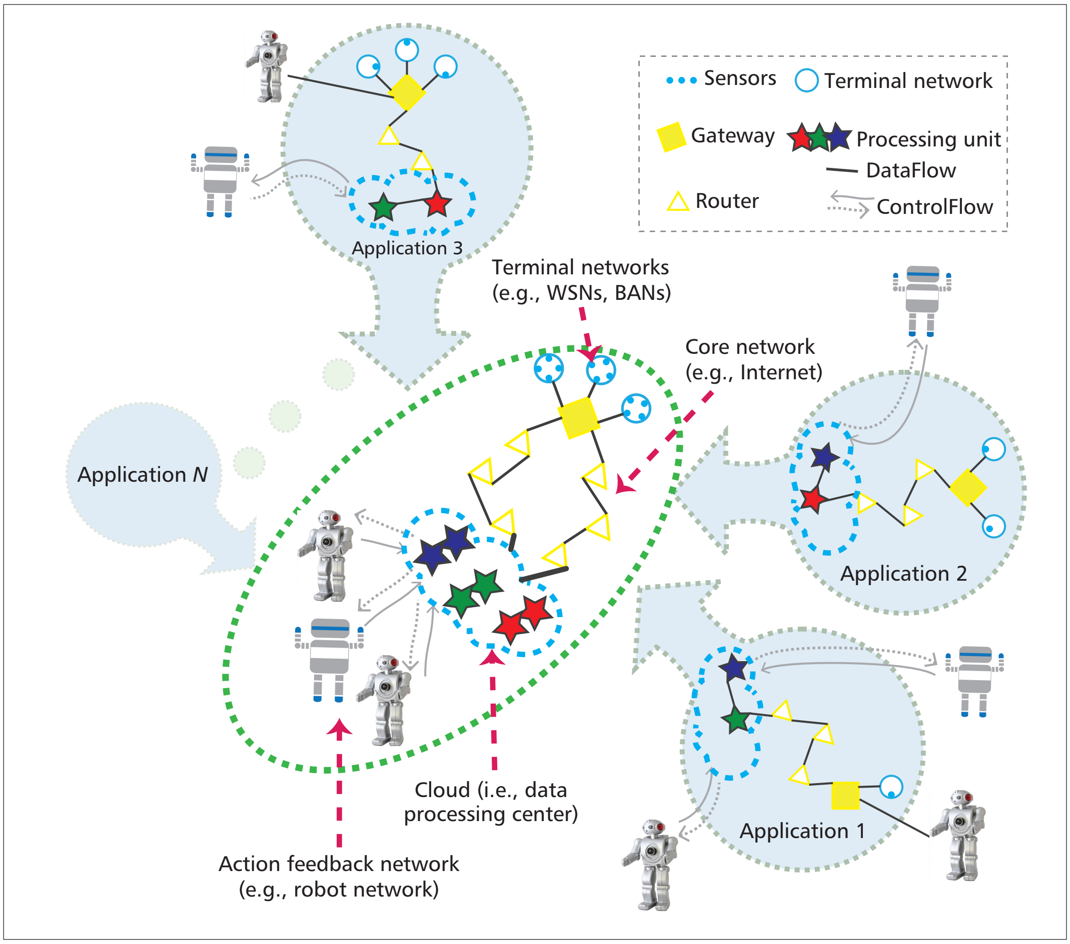

Figure 1. Illustration of software-defined health monitoring and heathcare.

application program. In addition to making the sharing of physical infrastructure possible, an API also conceals the complexity and isomerism of the physical infrastructure, which significantly reduces the difficulty of application program development and shortens the time to market for the new application. This trend shows that IoT, especially the sensor platform, should provide an API for the application program to develop their abilities in a flexible and efficient way.

Enlightened by SDN, this article proposes an intelligent health surveillance oriented software defined healthcare networking (SDHN) structure. In keeping with SDN, SDHN also separates the control logic from the physical device function through a logical and centralized controller managing the equipment through a standard interface. Specifically, SDHN expands the software defining method from the network devices to wearable devices, sensor platforms, robots, and cloud, and supports remote health surveillance and intelligent healthcare applications in combination with the application service interface (API) by providing data collection, transmission, and processing. Figure 1 shows the conceptual use of this architecture. The physical infrastructure is composed of the robot, BAN, sensor platform, router, and remote cloud server. Based on this architecture, multiple healthcare monitoring applications are deployed, and each application customizes its own data collection, transmission, and processing by the service interface. The standard API reduces the complexity and development cycle of deploying the new application; meanwhile, the sharing of physical infrastructure reduces the operation, administration, and maintenance (OAM) costs [6] to a large extent. These characteristics enable SDHN to support various application demands efficiently, making intelligent remote health surveillance possible.

Problem Statement

Figure 2 shows sensory data collected by several sensing infrastructures, such as a wireless sensor network, a body area network, and a robot. The sensing parameters are divided into body signals and ambient environmental information, for example, ECG, blood oxygen, respiration, environmental temperature, and environmental noise. The whole system can be divided into three subsystems logically, such as data collection, transmission, and processing. Specifically, for the purpose of hybrid human vital signs with environmental context modeling, different sensors are deployed on the human body and ambient environment, aimed at acquiring the significant signals of the human body and the environmental data. Then the collected data are transmitted to the remote cloud server for storage and processing. Generally, data collected by body sensors are first transmitted to the user's mobile phone by BAN, while the environmental monitoring sensor transmits the data to a gateway at the edge of a wireless sensor network (WSN). The gateway further transmits the data to the remote server by a wireless or wired network. Data processing may occur in the whole uplink, including filtering the unwanted data in the sensor node [7], compressing and encrypting the data in the gateway, further analyzing the collected data at the server side to gain statistical information, and so on.

At present, the application oriented method can be used for developing these three subsystems. In other words, the application developer need to customize the body sensor platform, environmental monitoring sensor platform, gateway, core network, and remote cloud server according to specific application demands. Usually, the developer should buy or develop a sensor platform in accordance with the application demands. It generally includes the sensor of acquiring the demanded data, wireless module of the data transmission, energy supply module, and micro-controller to coordinate the periphery module and execute the function of data processing. The firmware also needs to be customized for this specific application program. Although the application oriented method above seems to be intuitive, it has many drawbacks. We summarize these below.

Excessively High Capital and Maintenance Costs: Each application should deploy and manage its own sensor platform, which needs huge investment, deployment, and maintenance investment in hardware. Nevertheless, as a matter of fact, if the required data types of various applications are the same, sharing the same sensor platform is possible. Even if the required data are different, many modules in a sensor platform, such as wireless, energy supply, and microcontroller, can be shared to reduce the overall fees.

Unadaptable and Inflexible Change in Application Demands: In this method, the infrastructure and application are closely coupled. For example, the intelligence of the application program is static in the sensor platform, gateway, and server. Any change relevant to the application program needs the secondary development or secondary defined physical structure, which is complex, error-prone, and sometimes impractical. In addition, the deployment of a sensor platform is also inflexible in this case. When each application needs to deploy its own sensor platform, gateway, and remote server from the beginning, the total time for introducing a new application is not trivial. A lengthy development cycle and high investment will certainly be barriers to deploying the new application on a large scale, hence suppressing potential application innovation.

Inefficient Resource Utilization: When the applied control logic is embedded into hardware devices, it is very difficult to improve resource utilization by dynamic optimization of data collection, transmission, and processing. For example, when there is no method to dynamically control the data collection and transmission in the sensor platform, they will continuously transmit the data to the remote server, even if the data are unwanted at certain times, and waste the energy of the sensor platform and network bandwidth.

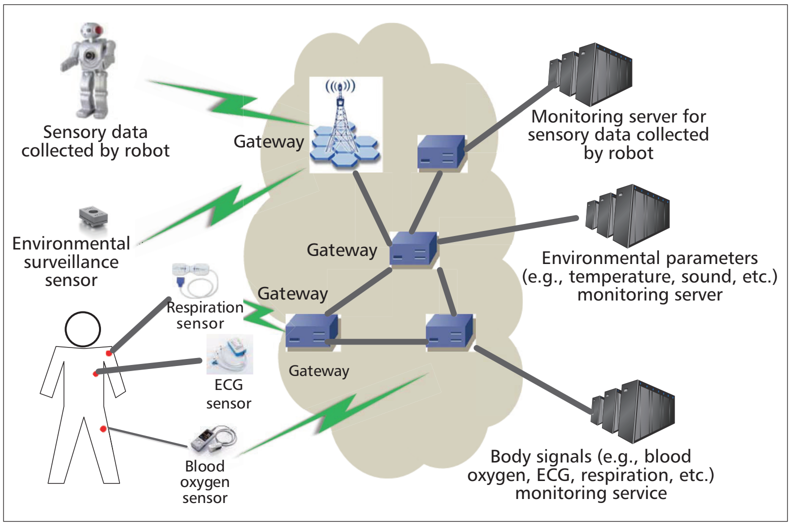

Figure 2. Illustration of urban sensing applications, which include three subsystems of data acquisition, transmission and processing.

After comparing the drawbacks of current technology, we are motivated to design SDHN from the aspects of data collection, transmission, and processing, respectively.

Architecture

In this article, we put forward the SDHN architecture, as shown in Fig. 3. SDHN is composed of three layers: physical infrastructure layer, control layer, and application layer.

Physical Infrastructure Layer: This layer is made up of all kinds of physical equipments, including a WSN (e.g., body sensor platform and environmental monitoring sensor platform), core network (e.g., gateway, base station, switch/router), cloud, and robot network. These equipments possess the basic function and resource to sense urban data, transmit the data from one node to another, and process the data to extract the requested information. However, they cannot decide what to do. On the contrary, they leave the decision to the control layer, and interact by the standard interface, such as the southbound interface via SDN.

Control Layer: The control layer acts as the medium between the physical infrastructure layer and application layer. On one hand, the control layer manages the physical equipments with different characteristics and functions by different southbound interfaces. On the other hand, the control layer offers the service for the application layer through the northbound interface and API. For the urban sensing application, the control layer will provide the data collection, transmission, and processing service. We explain the details of this service in the following chapters.

Application Layer: In this layer, the developers use the provided APIs to set up the remote health surveillance and smart healthcare application. In particular, they can customize the data collection, transmission, and processing, Under the SDHN, each sensor platform is equipped with more than one same or different sensors, and is shared by various application programs. For example, a sensor platform may include the temperature sensor and noise sensor at the same time, and accordingly the total sensor platforms to be deployed are significantly reduced.

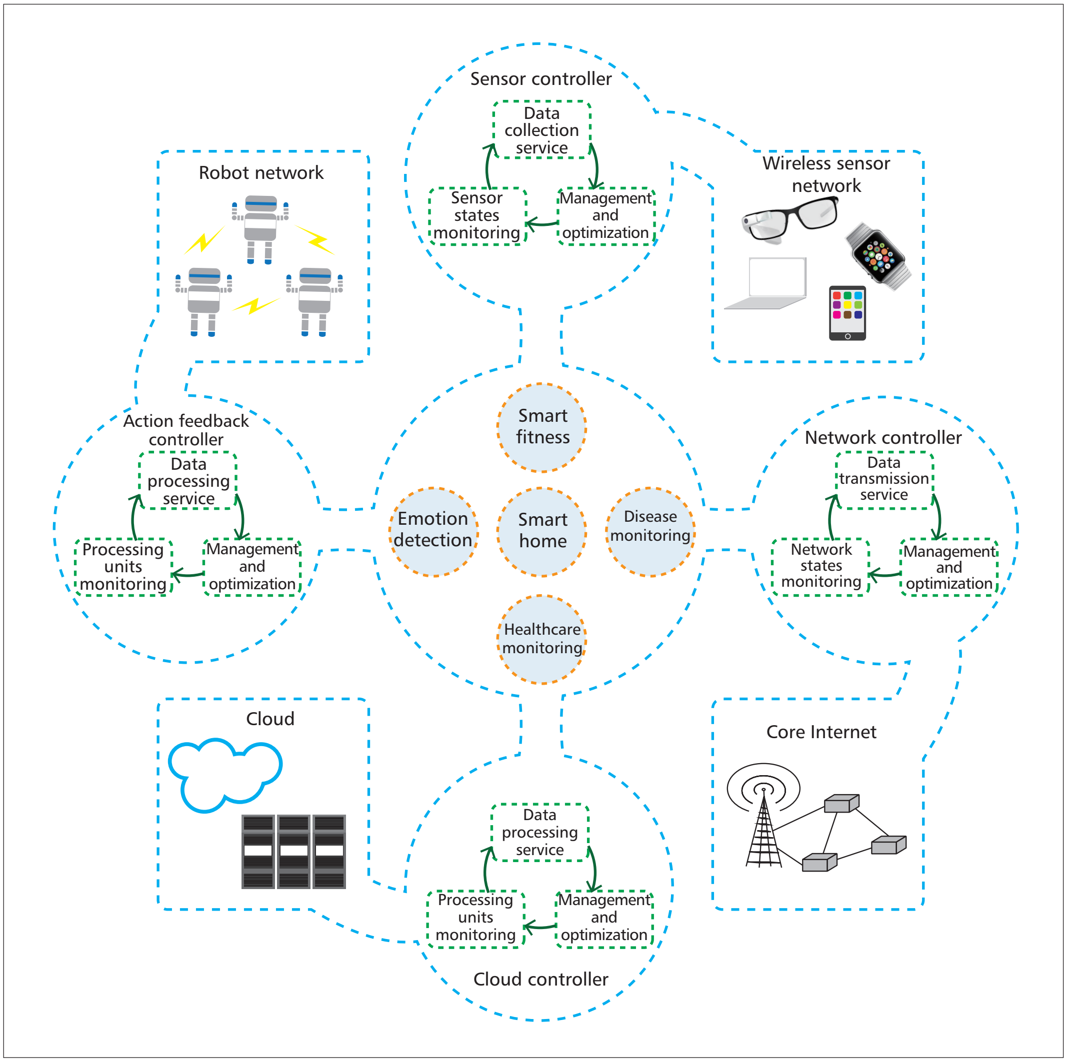

Figure 3. Architecture of software-defined healthcare networks.

without worrying about the change in configuration demand of the physical equipment, which greatly simplifies the development of a new application. In addition, the physical infrastructure is shared by various application programs, so the total costs and maintenance costs are reduced.

Sensor Platform and Data Collection Service

Figure 3 shows the diagram of data collection, transmission, processing, and action feedback, which includes four kinds of services, that is, data collection, data transmission, data processing, and action feedback. Different from conventional IoT, the action feedback component is added in SDHN for healthcare, especially for emotional care [8]. The action feedback controller can send a demand to a robot to care for a user's emotions.

To designate their data demands, data collection service systems provide APIs for each application program. The controller automatically configures the sensor platform to collect the required data. The data designation includes some general attributes, such as data type, target geographic area, and duration. For example, as shown in Fig. 4a, an application will request the ECG signal of user A, and also request the environmental temperature data. What is more, the data attribute can be set in accordance with the application program, such as setting the sampling rate for blood oxygen and noise data. The data collection service can provide APIs for the application program to query the attributes of valid data, such as data type, geographic area, and optional attributes of each type of data.

Under the SDHN, each sensor platform is equipped with more than one of the same or different sensors, and is shared by various application programs. For example, a sensor platform may include both a temperature sensor and a noise sensor, and accordingly the total sensor platforms to be deployed are significantly reduced. Finally, the overall investment in hardware, deployment, and maintenance is reduced. The sensor controller has an overall view of the sensor platform at the bottom layer. Specifically, it knows the position of each sensor platform and the embedded sensor. Due to the overall view, the sensor controller can dynamically activate or deactivate the sensor, customize its configuration to meet the application demands, and meanwhile reduce the energy consumption.

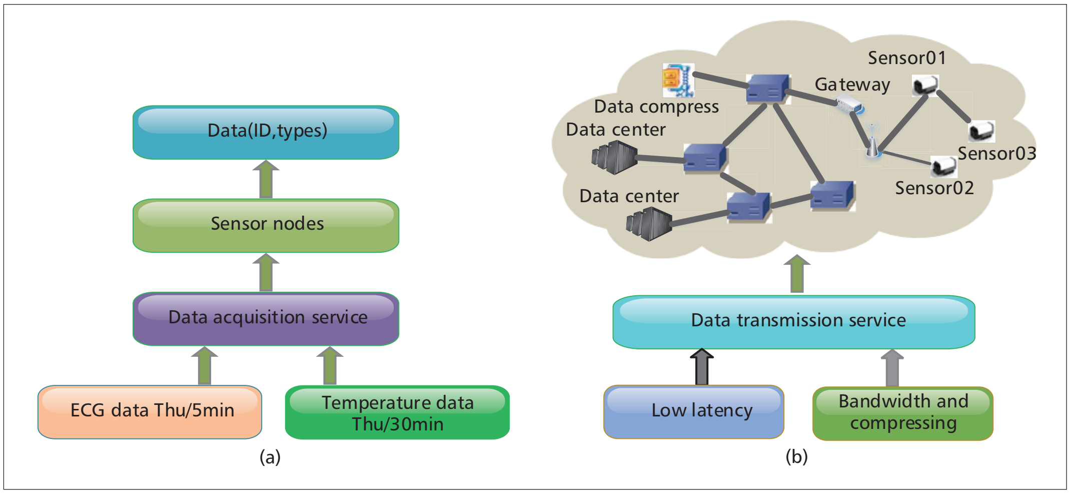

Figure 4. Illustration of data acquisition and transmission in SDHN: a) data acquisition service; b) data transmission service.

On the basis of the collected overall network view, the controller guides the data packet to different destinations, dynamically schedules the flow to meet the requirements of the application for network quality, and meanwhile optimizes the usage of the network resources.

Network and Data Transmission Service

The network transmits the data from the sensor platform to the cloud server. Since the application may prefer to select different cloud data centers, it should be able to designate the destination of data transmission. Besides, the application may have specific performance requirements for data transmission. For example, an intelligent transportation application program for providing a path planning proposal must know the current traffic load, so low-latency data transmission is required [9].

The data transmission service provides APIs for the application program to designate its demands. It mainly includes two dimensions: destination and quality of service (QoS) parameter. An IP address can be used for designating the destination, and meanwhile several options involving basic transmission, latency sensitivity transmission, and bandwidth guarantee transmission can be provided for matching QoS requirements. Generally, the basic transmission is made in a best effort manner. Latency-sensitive transmission possesses high priority during flow scheduling. The controller reserves the bandwidth for bandwidth guaranteed transmission. Moreover, with the propulsion of network functions virtualization (NFV), the network will also provide the path data processing service, such as data compression and encryption. Specifically, the data transmission service API will also allow the application to designate the service chain, such as the pipeline of the virtual network function, where a specific flow needs to pass. Figure 4b describes two examples of the data transmission service request.

In order to realize the data transmission service, the network also follows the SDN architecture. The repeater device is programmable. For example, OpenFlow starts, and the controller takes charge of realizing the flow control and scheduling. Specifically, on the basis of the collected overall network view, the controller guides the data packet to different destinations, dynamically schedules the flow to meet the requirements of the application for network quality, and meanwhile optimizes the usage of the network resources.

Data Processing Service

Software defined data processing shall be described by an example of emotion recognition and interaction.

First, which devices should carry out data preprocessing for emotional big data? As we know, the data come from sensors, social networks, and even human face video recorders. Shall those data be preprocessed locally or remotely in a cloud data center?

Second is emotional big data analysis. In the traditional method, it is done in the data center intuitively, because the data center has very strong processing capacity. However, sometimes a cloudlet [10] and even a mobile device can be processed locally to save much data transmission.

Action Feedback Service

After emotion recognition, the system should console the user, so there will be a series of actions. For example, a robot can do a set of actions to put me in a good mood. Or, our room now has a projector, and I miss my child very much, so I can play a video of my child on the projector. Therefore, the peripheral hardware resources for emotion soothing should be cognized; then the hardware equipment plays the medium with the corresponding feature in order to soothe the current emotion of the user.

From the task designed above, we can see that high flexibility is needed to configure the resources, including which type of task is processed by which type of equipment, and whether SDN can meet the demand of separating the control plane from the data plane with high flexibility.

Design Issues

Southbound Interface Design

In order to realize SDHN, the southbound interface will be designed for the interaction of the controller and the physical infrastructure. Some interfaces are already designed for this challenge. For forward devices, OpenFlow is the most widely used interface and abstracts the forward behavior between the heterogeneous switch and the router. For the server, interfaces are typ-

In SDHN, the application performance and control flexibility depend on the interaction performance between the control layer and the physical layer, such as the communication delay. Moreover, the control layer must be strong enough to work normally through all kinds of problems.

ically dependent on the cloud control system. However, the device heterogeneity is generally processed by middleware. Compared to the forward facilities and the server, the design of the southbound interface for the sensor platform is more complicated due to higher device heterogeneity. Moreover, since the energy of the sensor platform is limited, the energy cost of interacting with the controller shall be minimized.

In the literature, it is thought that the southbound interface is built for the sensor platform, so we propose a combination of the abstract strategy and middleware. First of all, a sensor platform is provided for abstraction of the data collection, processing, and transmission, and the operation of the controller is separated from the sensor platform. Second, before the standardization of the abstract, the actual control interfaces of different sensor platforms and the robot platform are switched to each other. The middleware can be adopted for the transmission. Especially for saving the energy of the human body sensor platform, the middleware can be placed in the controller. When the sensor platform is not activated, the interaction frequency of the controller with the sensor platform shall be reduced. Although these advantages exist, more discussion and study shall be conducted on the design and execution of the abstraction and middleware.

The design of one logically centralized control layer shall realize three objectives: high expansibility, high performance, and high robustness. First, as time goes by, the number of physical devices and applications will increase, In order to support them, the control layer shall be expanded at the same time. In addition, in SDHN, the application performance and control flexibility depend on the interaction performance between the control layer and the physical layer, such as the communication delay. Moreover, the control layer must be strong enough to work normally through all kinds of problems.

The deployment of multiple controllers is a generic method of achieving these goals. On one hand, the controller can be copied to increase its robustness; on the other hand, each controller can manage some of the devices. Therefore, the control layer shall be expanded by increasing the quantity of the controllers. Moreover, the controller can be placed in different locations, so the average communication delay of the physical device can be reduced.

Conflict Resolution and Optimization of the Sensor Platform

In SDHN, the human body sensor and the environmental monitoring sensor platform shall be shared through different applications, which may cause potential configuration conflict. For example, the noise data sampling rate a user needs is once every five minutes, while another user may also need the noise data, but the sampling rate he/she needs is different from the former. When the conflict is sent, the sensor controller needs to decide which request of the application is accepted, and how to resolve the conflict and minimize the energy consumption of the sensor platform.

In order to avoid this kind of conflict, the controller can assign one sensor to only one application. The strategy is simple but inefficient, since sometimes the sensor can be shared by applications with different settings. For example, application A needs to set the sampling rate to be once every five minutes, and application B needs to set the sampling rate to once every seven minutes. The controller can control the sensor to generate the sampling time sequence as: 5 min, 7 min, 10 min, 14 min, 15 min, and so on. When expanding the method under a general condition, the controller builds a resolver for each sensor; the resolver records the current configuration of each sensor and converts the service needs into the sensor configuration when receiving it. The resolver generates a suitable configuration according to the current and new configurations, or refuses the configuration needs if not coordinated.

Traffic Scheduling Supporting Service Quality

In SDHN, in order to guarantee the quality, the network controller can provide the data transmission of the end-to-end service. However, several challenges must be solved before realizing this capability. First of all, the number of queues of the forwarding device for executing the OoS is limited, so it is hard to support a large amount of QoS needs. Second, it is a challenge to design efficient traffic scheduling in a large-scale network to meet QoS needs [11].

Now we explain two strategies to deal with the above challenges. First, reduce the number of queues of the forwarding device through quantification of the QoS needs according to the statistical data of the needs. Second, consider the available number of queues of each forwarding device during traffic scheduling. Since these strategies further increase the complexity of traffic scheduling, some approximation algorithms shall be developed to solve the problem of traffic scheduling efficiently [12].

Resource Mapping of a Cloud Data Center

In SDHN, the cloud controller needs to decide how to map the application service demand to the physical device. For example, first, the stored data is allowed to be applied considering the cloud; then the data is processed by renting a virtual machine, and the cloud controller needs to decide where to store the data and which server is used for hosting the virtual machine for later data processing. Generally, there are several expected targets, such as to increase the income of the cloud supplier through acceptance of more service needs, to save energy through use of fewer servers, and to balance the load of servers through equal mapping of the service demand to different servers [13]. These limits include the server capacity, storage capacity, and the types the software/platform/virtual machine can bear.

One challenge when executing the above optimization is that different targets are contradictory. Therefore, the targets cannot be realized at the same time. For example, energy saving and load balancing are targets that are contradictory.

The balance between the contradictory targets shall be realized by developing an efficient heuristic method or an approximation algorithm. For example, weigh the load balance and save energy by activating or closing servers through use of a strategy based on the threshold value, that is, activate more servers when the average load exceeds the preset maximum value and close some servers when the load is smaller than the preset value.

Testbed

The testbed should exhibit the following capability of SDHN:

- It can control the collection frequency and the data rate of various body sensors.
- The platform can control the data transmission.
- As for action feedback, it can control the robot-to perform specific combination of actions or movements to achieve mental healthcare or emotional care [14].

In the data center, the testbed should analyze the physical condition and emotions of users aiming at life sign modeling and health big data analytics so that the emotions of users can be detected and comforted by controlling a robot in a software defined fashion.

Under the framework of SDHN, let the software defined robot (SD-Robot) denote a robot supported by the intelligence generated from the healthcare big data analysis via a data center and cloud. In SDHN, the cloud knows the needs of the user and the hardware resources of the environment where the user stays. Therefore, SD-Robot can be controlled to do the most suitable movements through the software defined method, and the surrounding hardware devices of the user can be adjusted to the most suitable status to be applicable to the current mood of the user.

In order to compare with SDHN, we make a comparison between the SDHN and the ordinary scheme. Traditionally, the advanced intelligence of the emotion interaction robot mainly comes from local processing, including some intelligence by mining and analyzing data collected locally. Let A-Robot denote a robot in the traditional scheme.

The intelligence of A-Robot is mainly limited by the following aspects:

- Due to its battery, it cannot conduct a series of data collection and task analysis, which would consume most of its power.
- The capability of communication and network access are limited.
- Its mobility is limited.
- Its capability of computation is limited.

Various factors limit the intelligence of A-Robot at the local level, which makes it not as strong as SD-Robot, especially when the problem of robot-building cost must be considered during realization. Therefore, some of the intelligence of SD-Robot is completed in the cloud. Except for saving computation cost, big data clouds can utilize long-term stored data and know the needs of various users and the applied QoS; these factors enable the decision made by SD-Robot to be high-grade.

How Should the Intelligence of the SD-Robot Be Allocated?

For example, for emotion big data, there are data collection, data analysis, and data feedback. So where are these three processes conducted?

- Data collection: The mobile phone and the robot can collect the data, and the backstage data center can also collect the network data or the data of the database through the network.
- Data analysis: This can be done on the mobile phone or the robot, or processed on the cloud.
- Emotion feedback: This involves how to feed back, by the mobile phone, cloudlet, or data center, to generate some videos and push them to users for feedback.

According to the above, consider the following problems.

The Span of the System: First, the system has to span various hardware platforms, such as comfortable wearable devices, the Long Term Evolution (LTE) base station, the cloud data center, the robot, and some surrounding hardware resources for emotional feedback.

The Centralized Control of the System: Since there are so many platform spans, how to optimize and schedule these resources becomes the key content.

More importantly, we need to consider various applications and users who have different requirements for QoS and quality of experience (QoE) [15]. SD-Robot needs to design a set of schemes for each type of user separately. Therefore, strategies shall be generated in the control layer according to the requirements of different users, and it is necessary to consider how to avoid conflict with other users. The control scope of the control layer is divided into three layers: data collection, data analysis, and visualization and data feedback, wherein the data collection and feedback controls the human body sensor, the environmental sensor, the mobile phone, and the robot, and the data analysis and visualization control the cloud to conduct personalized machine learning and the consequent presentation.

For example, the control layer needs to collect some important indicators, such as time, place, and body status, to determine whether to use a robot or a smartphone to collect the information of human and environment and what kind of information to collect according to the information, and to upload the collected information to the machine learning layer of the cloud. The machine learning layer processes the data, such as common categorization and clustering, and feeds back the result to the control layer. The control layer feeds back the result to the smartphone or the robot for comforting the emotions of users according to the results of the categorization or clustering and information like time and location.

Figure 5a shows the work comparison of hours between A-Robot and SD-Robot. We consider three different scenarios. In the first scenario, both of them are in standby mode. In the second scenario, both of them are in half-load work mode, which means that they just move around and do some data collection and transmission work. In the last scenario, all of them are in full-load mode to do maximum work such as moving, interactiing, data collection, transmission, and

In the data center, the testbed should analyze the physical condition and the emotions of users aiming at the life sign modeling and the health big data analytics so that the emotions of the users can be detected and comforted by controlling the robot in a software defined fashion.

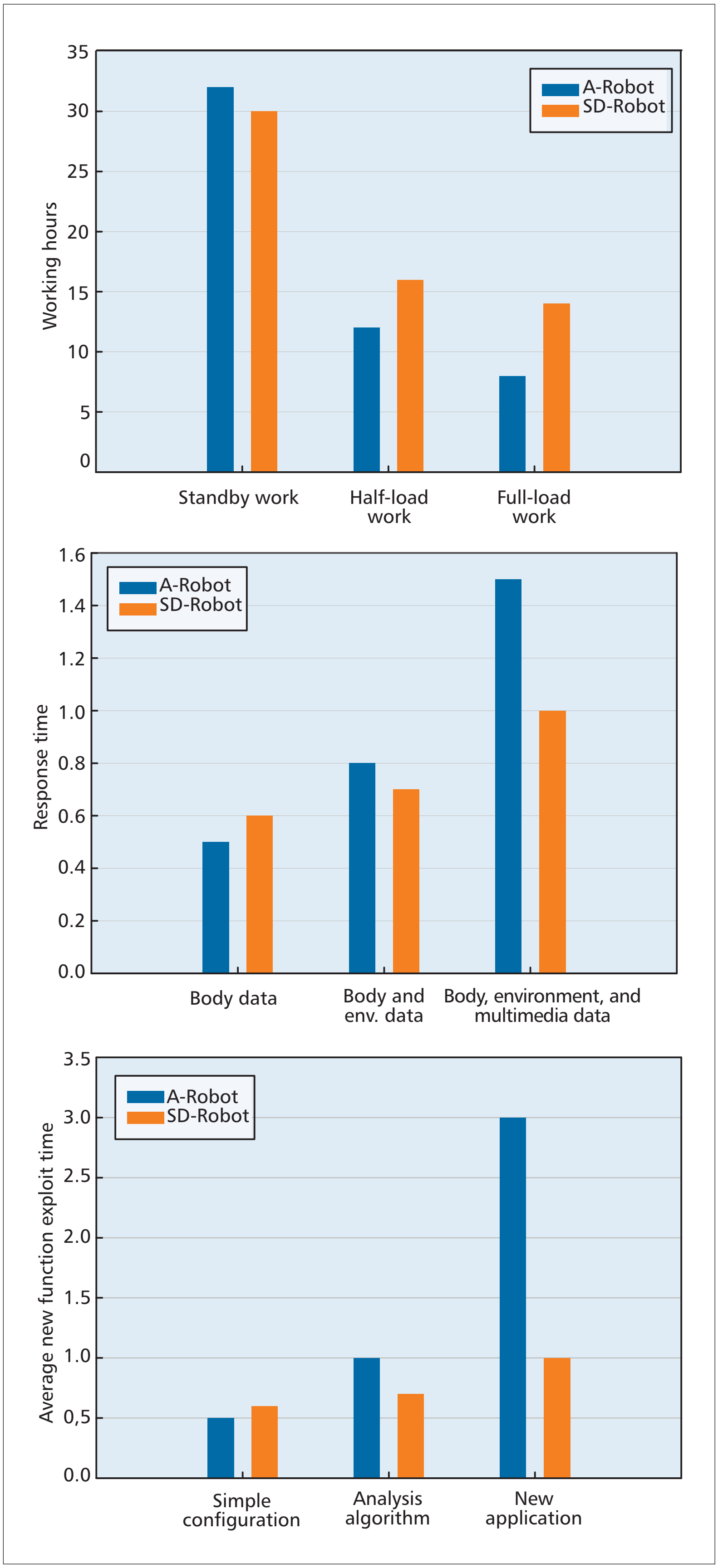

Figure 5. Illustration of the performance of both robots: a) comparison of work hours between SD-Robot and A-Robot; b) comparison of response time between SD-Robot and A-Robot; c) comparison of average new function deployment time between SD-Robot and A-Robot.

processing, and so on. It is shows that SD-Robot can work longer than A-Robot, especially in complex and compute-intensive conditions.

Figure 5b illustrates the response time, which means the computing capacity between A-Robot and SD-Robot with three different types: the first scenario is that both of them are just collecting some simple body sensor data and giving the right feedback upon request; the second scenario is both of them collecting body sensor data and other environment data and giving the feedback after having been activated as soon as possible. The last scenario is that all of them are collecting body data, other environment data, and other complex multimedia data such as a user's speech data, facial expression, and video data. with the right feedback after being activated. It shows that SD-Robot can perform more efficiently and work much better than A-Robot.

Figure 5c indicates the average new function deployment time between A-Robot and SD-Robot in three different conditions. The first scenario is that some simple component updating and configuration need to be added into both A-Robot and SD-Robot. It shows that both of them take almost the same time as A-Robot takes 0.5 day and SD-Robot takes 0.6 day. The second scenario means adding some normal analysis algorithms into both robots. Figure 5c shows that A-Robot needs almost 1 day but SD-Robot just takes 0.7 day. A more important point is that the last conditions show the biggest diversity between them: as a complex new application is added into the two robots, A-Robot needs almost three days to function well, but through our new proposed SD-Robot, it takes just one day to equip well due to the software defined-based architecture, which can update application requirements directly in the cloud center. There is no doubt that SD-Robot performs much better than A-Robot. In conclusion, SDHN architecture has more flexibility than traditional HealthIoT architecture. It can greatly reduce the overall system's operational complexity development cycle of new functions.

Conclusion

This article focuses on the design of the elastic HealthIoT structure, which deals with both intelligent health monitoring and emotional care. Specifically, we propose a HealthIoT framework with software defined capability by separating the application from the underlying physical infrastructure. With this framework, healthcare services can be customized to exhibit their own data collection, transmission, processing, and emotion feedback through well defined APIs, and enable multiple applications to exist in a shared infrastructure to reduce the total capital and maintenance costs. Therefore, the framework enables elastic control and management of the physical infrastructure and speeds up the innovation of various healthcare applications.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia, for funding this work through research group project no. RGP-VPP-229. Jeungeun Song is the corresponding author.

References

[1] M. Chen, "NDNC-BAN: Supporting Rich Media Healthcare Services via Named Data Networking in Cloud-Assisted Wireless Body Area Networks," *Info. Sciences*, vol. 284, no. 10, Nov. 2014, pp. 142–56.

[2] K. Zheng *et al*., "Challenges of Massive Access in Highly Dense LTE-Advanced Networks with Machine-to-Machine Communications," *IEEE Wireless Commun*., vol. 21, no. 3, 2014, pp. 12–18.

[3] C. Lai *et al*., "A Buffer-Aware HTTP Live Streaming Approach for SDN-Enabled 5G Wireless Networks," *IEEE Network*, vol. 29, no. 1, 2015, pp. 49–55.

[4] X. Ge *et al.*, "5G Wireless Backhaul Networks: Challenges and Research Advances," *IEEE Network*, vol. 28, no. 6, 2014, pp. 6–11.

[5] J. Medved *et al*., "Opendaylight: Towards a Model-Driven SDN Controller Architecture," *Proc. 15th IEEE WoWMoW*, 2014, pp. 1–6.

[6] D. Wu, Z. Xue, and J. He, "iCloudAccess: Cost-Effective Streaming of Video Games from the Cloud with Low Latency," *IEEE Trans. Circuits Sys. Video Tech*., vol. 24, no. 8, 2014, pp. 1405–16.

[7] J. Wan *et al*., "Cloud-Enabled Wireless Body Area Networks for Pervasive Healthcare," *IEEE Network*, vol. 27, no. 5, 2013, pp. 56–61.

[8] M. Chen *et al*., "EMC: Emotion-Aware Mobile Cloud Computing in 5G," *IEEE Network,* vol. 29, no. 2, Mar. 2015, pp. 32–38.

[9] Y. Li *et al*., "Contact-Aware Data Replication in Roadside Unit Aided Vehicular Delay Tolerant Networks," *IEEE Trans. Mobile Comp.*, DOI: 10.1109/TMC.2015.2416185, 2015.

[10] M. Chen *et al*., "On the Computation Offloading at Ad Hoc Cloudlet: Architecture and Service Models," *IEEE Commun. Mag*., vol. 53, no. 6, June 2015, pp. 18–24.

[11] J. Wan *et al*., "Context-Aware Vehicular Cyber-Physical Systems with Cloud Support: Architecture, Challenges, and Solutions," *IEEE Commun. Mag*., vol. 52, no. 8, 2014, pp. 106–13.

[12] C. Tsai *et al.*, "Data Mining for Internet of Things: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 1, 2015, pp. 77–97.

[13] X. Ge *et al*., "Spectrum and Energy Efficiency Evaluation of Two-Tier Femtocell Networks with Partially Open Channels," *IEEE Trans. Vehic. Tech*., vol. 63, no. 3, 2014, pp. 1306–19.

[14] M. Chen *et al*., "AIWAC: Affective Interaction through Wearable Computing and Cloud Technology," *IEEE Wireless Commun.*, vol. 22, no. 1, Feb. 2015, pp. 20–27.

[15] K. Zheng *et al*., "Qualtiy-of-Experience Assessment and Its Application to Video Services in LTE Networks," *IEEE Wireless Commun*., vol. 22, no. 1, Feb, 2015, pp. 70–78.

Biographies

LONG HU is a visiting student at the Department of Electrical and Computer Engineering, University of British Columbia. He is a Ph.D candidate in School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), China, since 2014. His research includes the Internet of Things, machine-to-machine communications, body area networks, body sensor networks, RFID, e-healthcare, and mobile cloud computing.

MEIKANG QIU [SM'07] (mqiu@pace.edu) received his B.E. from Shanghai Jiao Tong University, China and his Ph.D. in computer science from the University of Texas at Dallas. He is an associate professor of computer engineering at Pace University and an adjunct professor at Columbia University. He has published 10 books, 300 peer-reviewed journal and conference papers (including 130 journal articles, 170 conference papers, and 40+ IEEE/ACM Transactions papers), and three patents. He won the *ACM Transactions on Design Automation of Electrical Systems* 2011 Best Paper Award. Currently he is on the Editorial Boards of a dozen journals, including *IEEE Transactions on Computers* and *IEEE Transactions on Cloud Computing*. He has been the General Chair/Program Chair of 10+ IEEE/ACM conferences, including IEEE HPCC/ICESS/CSS 2015, IEEE CSCloud 2015, and IEEE BigDataSecurity 2016. He won the Navy Summer Faculty Award in 2012 and the Air Force Summer Faculty Award in 2009. His research is supported by the U.S. Government (e.g., through NSF) and companies such as Nokia, TCL, and Cavium.

JEUNGEUN SONG (jeungeunsong@163.com) is a Ph.D candidate at the School of Computer Science and Technology, HUST. Her research interests include emotion detection, healthcare monitoring, SDN, healthcare big data, and more.

M. SHAMIM HOSSAIN (SM'09) is an associate professor with the Department Software Engineering, King Saud University, Riyadh, Saudi Arabia. He has authored or coauthored more than 100 publications, including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. He has served as a Co-Chair, General Chair, Workshop Chair, Publication Chair, and Technical Program Committee Chair for over 12 IEEE and ACM conferences and workshops. He served as a Guest Editor for *IEEE Transactions on Information Technology in Biomedicine*, the *International Journal of Distributed Sensor Networks*, and the *International Journal of Multimedia Tools and Applications* (Springer). He currently serves on the Editorial Board of the *International Journal of Multimedia Tools and Applications*, and as a Lead Guest Editor for *IEEE Transactions on Cloud Computing*, *Future Generation Computer Systems* (Elsevier), *Computers and Electrical Engineering* (Elsevier), and *Cluster Computing* (Springer).

AHMED GHONEIM received a Ph.D. degree from the University of Magdeburg, Germany, in the area of software engineering in 2007. He is currently an assistant professor with the Department of Software Engineering, King Saud University. His research activities address software evolution, service oriented engineering, software development methodologies, quality of service, network-centric computing, and human-computer interaction.

With this framework, healthcare services can be customized to exhibit their own data collection, transmission, processing and emotion feedback through the well-defined APIs and enable multiple application to exist in a shared infrastructure so as to reduce the total capital and the maintenance cost.