This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2547890, IEEE Access

SDN-based Application Framework for Wireless Sensor and Actor Networks

Jianguo Zhou, Hao Jiang, Jing Wu, *Member*, *IEEE*, Lihua Wu, Chunsheng Zhu, *Student Member*, *IEEE*, Wenxiang Li, *Member, IEEE*

***Abstract*—As a promising platform for implementing various applications, a wireless sensor and actor network (WSAN) consists of many sensor and actor nodes that can cooperatively handle complex tasks. However, many issues, including nodes' mobility, the heterogeneity of capacity, topology, and energy consumption, may bring severe challenges to efficient WSAN operation. Currently, the Software Defined Network (SDN) appears as a novel approach that is effective to manage and optimize networks in a programmable and centralized pattern. This paper studies the application framework and relevant methods for applying the SDN approach in a WSAN, with the objective of improving network's efficiency and scalability. The details of the framework include a three-layer structure, the relevant system entities, the enhanced protocol stack, and the programmable message types for cooperative communication and task execution among WSAN nodes. Based on this framework, this paper explores the relevant challenges and mechanisms for effective system management from many aspects, including mobility, energy saving, reliability maintenance and topology construction. This paper also proposes an optimization method for scheduling decomposed tasks to relevant nodes, with an example implemented by the Genetic Algorithm. Next, this paper demonstrates the typical application scenarios, including military, industry, transportation, and environmental disaster monitoring. Moreover, an indoor application scenario and an outdoor application scenario are presented to demonstrate the application of the SDN-assisted communication handoff. Last, the future trends and technical challenges for SDN in WSAN are discussed.**

***Index Terms*—Software defined networks; wireless sensor and actor networks; scalability; network management; task scheduling; protocol stack**

This work is part of the National Natural Science Foundation of China (NSFC, No. 61371126), the National High Technology Research and Development Program of China (863Program, No. 2014AA01A707), and the Applied Basic Research Programs of Wuhan (No. 2014010101010026).

This work is also supported by the Scientific Research Foundation for Returned Overseas Chinese Scholars of the State Education Ministry of China (the 50th, 2015), the Science and Technology Research Project of the Education Department of Hubei Province of China (No. Q20141110), and the Technique Innovation Project of the Next Generation Internet from CERNET (No. NGII20150301).

J. Zhou, H. Jiang, J. Wu and L. Wu are with the School of Electronic Information, Wuhan University, Wuhan 430072, China, and with the Collaborative Innovation Center for Geospatial Technology, Wuhan 430079, China. (e-mail: zjg@whu.edu.cn, jh@whu.edu.cn, wujing@whu.edu.cn, wlhcheers@126.com) H. Jiang is the corresponding author.

C. Zhu is with the Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC V6T 1Z4, Canada. (e-mail: cszhu@ece.ubc.ca)

W. Li is with the School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China (e-mail: liwx2006@hotmail.com).

I. INTRODUCTION

Wireless sensor and actor networks (WSANs) [1, 2] consist of a large number of sensor nodes and actor nodes. A WSAN can not only gather real-time information from the outside environment but also actively give responses to the environment for realizing complex tasks or applications. Thus, a WSAN is potentially valuable for applications in many fields, including transportation, industry, military, medical care and environmental protection [3]. Commonly, there exist three basic cooperative communication patterns for the nodes in a WSAN, including sensor to sensor (SS), sensor to actor (SA), and actor to actor (AA) [4, 5]. The SA communication emphasizes the selection of a practical path for packets transmission between sensors and actors, and the key lies in the selection of the proper sensor set and actor set. The AA communication emphasizes the decision of the nodes' actions and their procedures, and the key lies in properly arranging actors for task execution. These task-oriented communication patterns have many performance requirements [6], including:

1) Real-time action;
2) Low energy consumption;
3) The selection of proper nodes for eliminating redundant messages and executions [7];
4) The correct order of tasks execution;
5) The guarantee of the effective and reliable transmission of the sensing results and the execution results [8].

The nodes in WSAN operate in a distributed way, making it difficult to control the sensing objects, patterns, and the starting time for cooperative communication in a global view. The nodes can also lead to frequent communication conflicts that may cause high energy consumption, low reliability, and large delays [9]. In addition, the nodes' dynamic behaviors, including joining and quitting, are difficult to discern by other nodes. Moreover, it is difficult for all nodes to obtain a synchronized update of their states and functions [10]. These issues lead to worse scalability of network scale and performance.

The software defined network (SDN) [11, 12] stems from Clean Slate's project in Stanford University. Based on the idea of multi-layering, SDN separates the two functions of forwarding control and data transmission that are originally integrated in one node. In the control plane, the logically centralized and programmable SDN controller holds the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2547890, IEEE Access

information of global network, making it convenient for researchers and operators to manage the network and deploy new protocols. In addition, in the data plane, nodes implement only simple functions of packets forwarding, enabling the whole network to easily handle the increasing amount of traffic flow. There exist open and unified interactive interfaces between the two planes. The SDN controller can send unified forwarding rules to the nodes in the data plane through these interfaces, and these nodes simply handle task requests according to the relevant rules. As a result, SDN can exert effective control on the communication infrastructures and reduce the processing load of the forwarding nodes. In addition, with a programmable interface for function development, SDN can provide compatibility to traditional network and achieve real-time update and optimization for the network.

By applying the SDN approach to WSAN, we can achieve the optimal control and scheduling for the whole process of cooperative communication and task execution by relevant nodes, thereby improving efficiency/reliability and decreasing energy consumption in the WSAN. Moreover, the SDN controller can provide many functions for controlling mobile WSAN nodes and add new functions based on the open programming interface; thus, SDN can improve the scalability of WSAN, both in scale and in function. Currently, as few works exist on this topic, this paper explores the framework and methods for optimizing WSAN with SDN. The main contributions of this work are as follows.

1) For exploiting the merits of SDN to address the above challenges in WSAN, we propose an SDN-supported WSAN framework and describe the relevant details for implementation of the framework, including entities, protocol stack and the message type definition.

2) Based on the above framework, we explore the challenges and corresponding measures for optimizing system management from many aspects, including mobility, energy, reliability, and topology. We also design a task-driven scheduling strategy based on the Genetic Algorithm for different performance optimization objectives.

3) We propose the typical application scenarios and future technical challenges based on the above framework. We also present an indoor application scenario for video-on-demand service and an outdoor application scenario for monitoring water-quality, and apply the SDN control to achieve smart handoff of the forwarding nodes. The results show that the framework can effectively control the handoff process for the mobile client to connect to a new relay node.

The rest of this paper is organized as follows. Section II discusses the current SDN-relevant work in wireless sensor networks. Section III describes the framework entities, message types and protocol stack for the SDN-supported WSAN. Section IV explores the challenges and countermeasures for efficient system management. Section V explores the model and implementation of the SDN-based scheduling method. Section VI discusses the typical application scenarios, and introduces the experiments of the SDN-assisted connection handoff. Finally, section VII proposes the future trends and challenges for SDN in WSAN.

II. Related Work

Currently, there are only a few research works in the literature on applying the SDN approach in a wireless sensor network (WSN). [13] took a radical, yet backward and peer compatible, approach to tackle the problems inherent to WSN, proposed a Software-Defined architecture for WSN, and addressed the key technical challenges for its core component, i.e., Sensor OpenFlow. This work represented the first effort that synergized the software-defined networking and WSN.

For intelligent system management in WSN, [14] designed a general framework of the software defined wireless sensor networks. In this work, the SDN controller was implemented in the base station. Further, [14] discussed some important issues to be handled in the future, including the synchronization of the network states, the necessity of the distributed controller, the security of the central controller and the applicability of Openflow [15].

[16] implemented a software defined approach in the environment of Internet of Things (IoT). This approach was meant to provide preset service grades for different IoT tasks in a heterogeneous environment. Further, the SDN controller, as shown in Fig. 1, was built based on a reactive middleware, i.e., the Multinetwork INformation Architecture (MINA) that realized a close loop of "observe, analysis, and adaptation". The SDN controller can schedule different flows based on different task grades and heterogeneous Ad hoc paths, and improve the utilization of the service resources in IoT based on a modified intelligent algorithm. The relevant prototype system had been applied in automatic driving, smart grids, and electronic tolls.

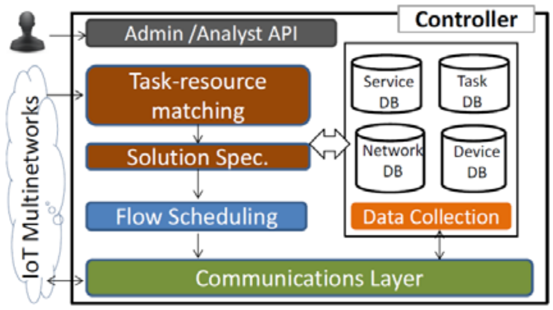


Fig. 1. The IoT controller Architecture [16]

[17] analyzed the future prospect of applying SDN in the IoT; the relevant application scenarios included multimedia delivery, smart home, mobile network, virtualization, security, reliability, and recovery. The relevant challenges for implementing the applications were also discussed, including the function integration of the control plane and the data plane, the balance of functions in hardware and software, security, the transition from current architecture to the SDN-supported

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2547890, IEEE Access

architecture, and the location and number of the SDN controller's deployment.

In [18], a SDN-based Sleep Scheduling algorithm (labeled as SDN-ECCKN) was designed to manage the energy consumption of the network. In this algorithm, each computation was implemented in the SDN controller rather than the sensors themselves, and there existed no broadcasting messages between any two nodes, which was the main characteristic of the traditional EC-CKN method. The evaluation results of the SDN-ECCKN method showed its advantages in energy management, and the relevant performance indexes included network lifetime, the number of solo nodes, and the number of live nodes in the network.

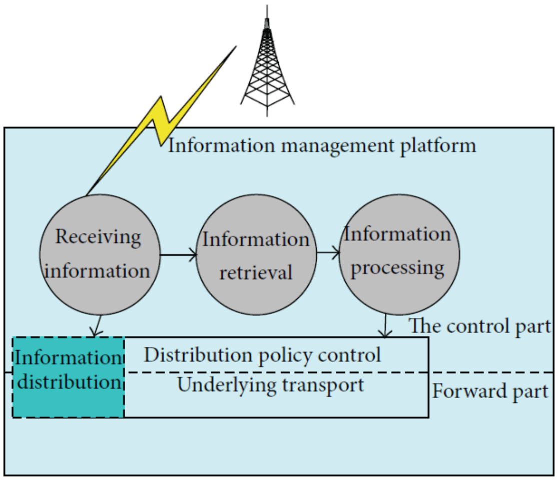


Fig. 2. A typical WSN based on the OpenFlow network structure [19]

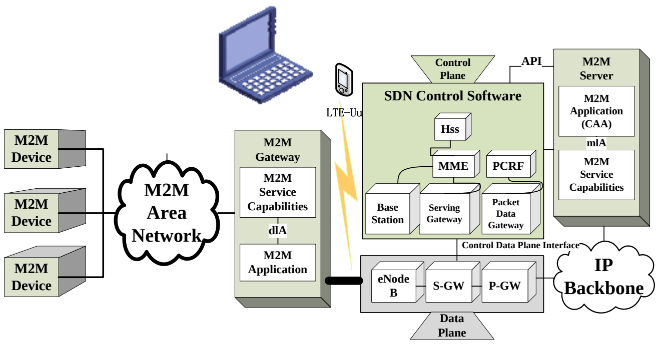


Fig. 3. The SDN controlled cellular framework [20]

Considering the issues of load balance and energy consumption, [19] proposed a WSN optimization method based on the Openflow, as shown in Fig. 2. In this method, the whole WSN was divided into several clusters, each of which owned a cluster head. The functions of the global SDN controller included making decisions of forwarding strategies, controlling each cluster head, and building paths.

For the SDN environment, [20] proposed a cellular network architecture that was monitored by the M2M techniques, as shown in Fig. 3. In this architecture, M2M facilities provided the context information and sent the sensing signals, and the SDN controller was in charge of redirecting the users' data flow and the flexible reconfiguration of resources.

Considering two use cases, i.e., SDN enabled IoT gateways and eNodeB, [21] described how to use SDN effectively to address the challenges faced by service providers in controlling the dynamic nature of current networks. These methods and other relevant services with the help of North Bound APIs can be applied in various scenarios, including enterprise gateways, wireless networks, service providers, and data centers, to dynamically program networks to make them faster and agile.

A TinyOS-based SDN framework that enabled multiple controllers for WSN, i.e., TinySDN, was presented in [22]. The framework was composed of two main components: the *SDN controller node*, where the control plane was programmed, and the *SDN-enabled sensor node*, which included an SDN switch and an SDN end device. The TinySDN was developed and implemented to be hardware independent. Simulation experiments were conducted on the COOJA simulator, and the results concerning the memory footprint and delay were given.

Currently, there is little work on SDN design in WSAN. This paper explores the SDN-based approaches to resolve the problems of communication conflicts and scalability.

III. ARCHITECTURE AND FUNCTIONS

A. Description of the framework and the entities

For applying SDN to WSAN, three principles should be observed while considering the vertical scalability.

Principle 1: The control functions should be reserved in the control plane, except for those that could promote the processing efficiency and adapt to the hardware and software requirements in the data plane to accord with the evolutionary trend of SDN.

Principle 2: The control functions cannot change the basic process of the data plane to ensure the generality of network devices in SDN.

Principle 3: Collecting statistics in the data plane cannot affect the accuracy or validity and cause high load in the control plane.

The three-layer SDN-based WSAN framework is shown in Fig. 4, and its feature is represented by the control plane between the data plane and the application plane. In the control plane, the SDN controller instead of the distributed WSAN nodes makes decisions for controlling packet forwarding.

The relevant entities from WSAN are described below.

1) **Sink node**. The node resides in the data plane and is in charge of message collection and data fusion. Based on its protocol set and interface set, the whole WSAN is divided into many communication clusters.

2) **Sensor node**. The node resides in the data plane. The functions of the node include sensing the states of task execution and environment as well as forwarding the state messages to relevant actors.

3) **Actor node**. The node resides in the data plane. The functions of the node include receiving the state information from sensor nodes, executing cooperatively or independently, and returning the execution results to upper layers.

The relevant entities from the users are described below.

4) **Tasks**. A task is generated in the application plane and is sent to the control plane. The task's attributes are designated by the users, including volume, type, delay limit, etc.

5) **Subtasks.** Each task is composed of several subtasks. As the elementary component of a task, each subtask is handled by an actor node.

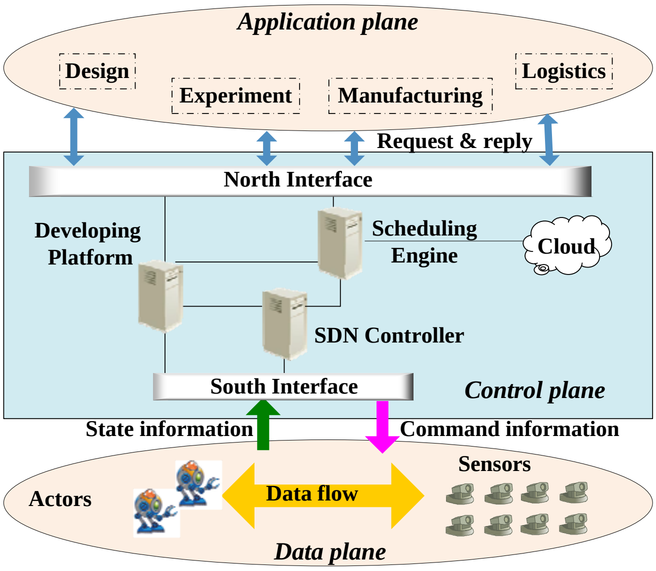


Fig. 4. The WSAN framework based on SDN

The relevant entities from the control plane are:

6) **North interface**. The interface receives and analyzes the incoming task requests from the application plane, and decomposes each task into several relevant subtasks according to specified process. The north interface can also return the response of task execution to the application plane.

7) **Scheduling engine**. Based on the task/subtask demands in the application plane and the states of nodes' load in the data plane, the scheduling engine selects the proper numbers and types of sensors or actor nodes and sets their cooperation behaviors for the desired overall performance.

8) **Cloud intelligence**. This entity provides the knowledge base and the decision function for decomposing tasks in the north interface and scheduling subtasks in the scheduling engine.

9) **SDN controller**. Based on the decisions from the scheduling engine, the functions of the SDN controller include controlling the transmission flows in the data plane and the execution of system management functions.

10) **South interface**. This interface analyzes various commands for tasks execution, communication and sensing from the scheduler and generates the control messages for relevant sensor and actor nodes. In addition, the south interface should accept and analyze the state information of task execution (including relevant facilities, products and progress) and network (including available protocols, links, interfaces and nodes).

11) **Developing platform**. With a friendly and flexible programming interface, the developing platform enables the system developer or administrator to design and improve the functions of relevant entities according to users' specific requirements.

In this way, the entities in the data plane need not care about the discovery, decision and maintenance of the end-to-end route. Thus, the large volume of data traffic is confined in the data plane as much as possible. According to the collected statistics on traffic flow, the SDN controller can obtain a global view of network states and select the desired disjoint transmission paths for all on-going task instances; thus, it can decrease the communication conflicts and the redundant packets transmissions.

B. Message types

The detailed message types for implementing the SDN-based WSAN framework are as follows.

1) **User's request**. Holding the description of tasks, the user's request is designated by the user and sent to the control plane.

2) **Result response**. The result response is sent from the control plane to the users and holds the response of task execution.

3) **Execution state**. The execution states hold the states of the tasks/subtasks and relevant entities, i.e., sensors and actors. In addition, these states are sent from the entities in the data plane to the south interface periodically or on-demand.

4) **Network state**. The network states hold the states of the entities for communication, traffic flows and the relevant communication resources, i.e., protocols, interfaces, bandwidths and energy. In addition, these states are sent from the entities in the data plane to the south interface periodically or on-demand.

5) **Control information**. The control information is sent from the SDN controller to the entities in the data plane, including the settings of the participants, the starting time and the pattern for vital actions (communication, sensing and execution), the modifications on the node's routing table and the settings of node's states (working or sleeping).

6) **Data**. The data are the information for task execution among the sensors and actors.

The developing platform can dynamically define the message types and set their parameters as needed.

C. Protocol stack

The traditional structure of the protocol stack [23] for WSAN is shown in Fig. 5. In this structure, the cooperation plane interacts with the five protocol layers and makes decisions on how the nodes react to commands. The management plane also interacts with the five protocol layers with the responsibility for system maintenance, and provides the necessary information for cooperation among the nodes. The five-layer protocol stack is implemented in each WSAN node for communication. Moreover, every node should

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2547890, IEEE Access

implement the relevant functions of distributed management and cooperation.

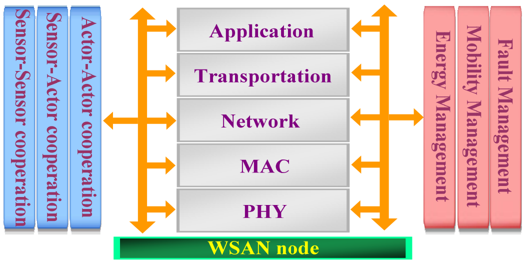


Fig. 5. The traditional WSAN protocol stack

Considering the design and introduction of the SDN's control plane, we can simplify the architecture and the functions of the vast number of WSAN nodes and improve the scalability of the application system. The relevant architecture of the protocol stack is shown in Fig. 6. The main feature of this architecture is the division of functions at the network layer of the WSAN node. We classify these functions into two categories, including the local functions for the WSAN nodes and the global functions for the SDN controller.

1) ***Local functions at the network layer*** include the segmentation and merging of data packets, activating or terminating network connection, service ordering, and error detection.

2) Based on the information of nodes' states and the network topology, ***global functions at the network layer*** include route discovery, route selection, the maintenance of routing table, and flow control.

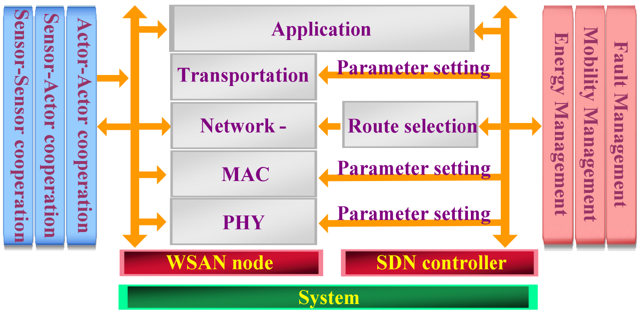


Fig. 6. The SDN-based protocol stack for WSAN

Based on the collected state information of task execution and network, the SDN controller can dynamically set the routing table for relevant relay nodes with fast convergence speed and small cost. Moreover, the SDN controller can select the appropriate parameters for other protocol layers, and implement all functions of network management. This function division at the network layer can simplify the functions and the design of WSAN nodes, and the lightweight protocol stack of WSAN nodes can yield improved network performance.

IV. Challenges and Countermeasures for System Management

To realize centralized SDN control functions in a distributed WSAN environment, it is vital to exert effective management and optimization with consideration of multiple issues, including mobility, energy consumption, reliability maintenance, security, heterogeneity, the SDN controller's load and topology. These management functions are all implemented in the SDN controller.

A. Mobility management

The mobility of nodes in WSAN can affect packets transmission and task execution [24]. Mobility management conducted by the scheduling engine and the SDN controller provides stable services under a dynamic network environment. The challenges on mobility management are as follows:

1) How to discern the nodes that newly enter the WSAN, including their attributes and functions, and how to manage the new nodes, including function registration;

2) How to handle the leaving nodes without adverse influence on task execution.

The SDN controller provides countermeasures for these problems, and the typical processing steps are as follows.

1) Each new node detects some neighboring nodes and sends the JOINREQ messages to them at the interval of T. These messages consist of the information on the functions, current energy level and other characteristics of the new node.

2) Neighboring nodes forward these messages to the SDN controller.

3) When the SDN controller receives these messages, it begins to authenticate the new node.

4) If the SDN controller accepts the node, it then registers the node, sends the JOINACK message to the new node, and omits the subsequent JOINREQ messages from the new node.

5) When the new node receives the JOINACK message, it stops sending the JOINREQ message and begins to operate normally.

6) If the SDN controller refuses to accept the request, then the SDN controller sends the JOINDEC message to the new node.

7) When the new node receives the JOINDEC message, it stops sending the JOINREQ message and leaves the network.

If a node is going to leave the network, it sends the LEAVEREQ message to the SDN controller. When the SDN controller receives this message, it attempts to select another node with similar functions to replace the role of the leaving node for task execution. Subsequently, it sends the LEAVEACK message to the leaving node. The node leaves the network as it receives the LEAVEACK message. The message sequence chart for successful joining and leaving is shown in Fig. 7.

B. Energy management

The nodes in WSAN, especially the sensor node, carry limited energy. Due to the difference in communication distance, transmission rate, task volume, and the pattern of task execution, there are obvious distinctions on the patterns of energy consumption among the nodes. The distributed energy optimization method on each WSAN node cannot achieve the globally optimal performance, so it is advisable to adopt the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2547890, IEEE Access

SDN controller for the global balance of energy consumption [25]. The detailed mechanisms that are implemented through the scheduling engine and the south interface include:

1) Collecting the characteristics of the patterns of energy consumption and the remaining energy for each node;

2) Balancing the load of communication and task execution;

3) Controlling the working/sleeping state of the key nodes with low energy level.

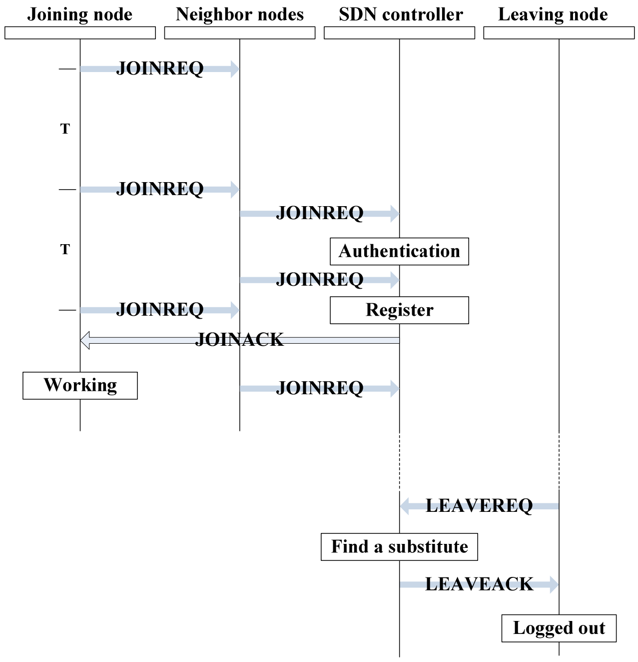


Fig. 7. The message sequence chart for successful joining and leaving

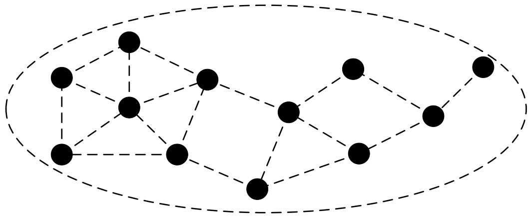
C. Topology management

The executions of different tasks require different parameters of network connectivity, sensing coverage and actuation coverage for the optimal WSAN performance. Thus, we require a proper topology structure for various types of tasks. The construction of the desired topology is implemented by the SDN controller. As shown in Fig. 8, the popular topology structures include flat topology, hierarchy topology, and overlay topology.

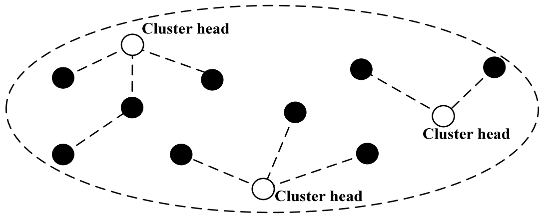
The flat topology is easy to deploy, and leads to no network performance bottleneck. However, the flat topology yields limited scalability of network scale and can be applied for cooperative nodes within only a short distance, e.g., the scenario of deploying a WSAN in the workshop for product assembling.

The hierarchical topology can provide satisfactory scalability of network scale, but the traffic may concentrate in the cluster head node, which is vulnerable to attack or easily becomes the performance bottleneck of the whole network. This topology can be applied in scenarios of large area, e.g., the intelligent transportation control system in cities.

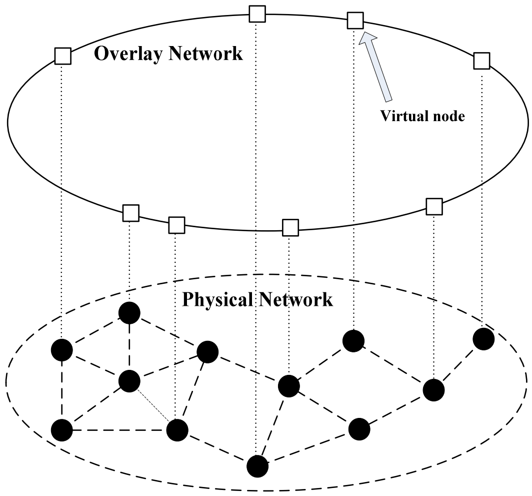
Based on the physical network, a virtual overlay topology can be constructed according to a certain specification. This topology can provide smart adaptability and flexibility for the definition and the implementation of various functions, and can be applied in scenarios of a dynamic combination of various actor nodes.



(a) Flat topology



(b) Hierarchical topology



(c) Overlay topology

Fig. 8. The typical topology maintained by the SDN controller in WSAN

D. Reliability management

Many factors that affect communication exist in the wireless environment of a WSAN, so it is necessary to take measures to guarantee the system reliability. The recovery from failure by the distributed WSAN nodes leads to complex interactions among the nodes with large delay and cost. By means of two levels of reliability, i.e., ***the communication reliability*** and ***the task reliability***, the SDN controller attempts to guarantee reliability with high efficiency. For communication reliability, there are such traditional methods as redundant links and backup nodes. For task reliability, the SDN controller provides the preservation and recovery of sensing information for the case of unreliable sensors, and provides the preservation and recovery of the task context for the case of unreliable actors. These two levels of reliability maintenance can be applied to tasks of different levels of importance.

V. Software defined scheduling method

A. Performance metrics

In our system, the subtasks are sent to the nodes in the WSAN for cooperative execution. For the optimal distribution of traffic flows and satisfactory QoS or QoE, the following issues should be considered in subtask scheduling.

1) The cooperative utilization of resources, i.e., the efficient utilization of energy and communication capacity for a robust WSAN.

2) The cooperative task allocation addresses task description, decomposition, allocation, scheduling and execution.

3) The cooperative information processing addresses the data fusion by sensor nodes for reducing the communication cost.

Given N nodes and M subtasks in the system, we should pay attention to the following properties or performance metrics of relevant entities.

1) VT_j: The volume (measured in kilobytes) of transmission for the j-th subtask.

2) VP_j: The volume (measured in kilobytes) of processing for the j-th subtask.

3) E_i: The current energy level (measured in joules) of the i-th node.

4) DT_i: The average transmission delay (measured in seconds) for one kilobyte of data in the i-th node.

5) ET_i: The average energy consumption (measured in joules) for transmitting one kilobyte of data per meter in the i-th node. Note that the energy consumption increases roughly by d^{α}, where d is the distance, and α is the attenuation factor.

6) DP_i: The average processing delay (measured in seconds) for one kilobyte of data in the i-th node.

7) EP_i: The average energy consumption (measured in joules) for processing one kilobyte of data in the i-th node.

8) $\boldsymbol{F_i}$: The function vector $\{x_1, x_2, \ldots x_M\}$ for the i-th node has the value of x_j=1 if this node can handle the j-th subtask, otherwise x_j=0, $1 \le j \le M$.

All these metrics can be presented in the developing platform as the relevant parameters of tuning for the optimal performance.

B. Scheduling algorithm

Based on the states of network and task execution, the scheduling engine designates a dynamic workgroup for each task. The workgroup consists of such key information as the group members of WSAN, the starting time and sequence of cooperative packet transmission, the content of messages, the starting time and sequence of task executions, the content of executions, task volume, the QoS constraints and the energy consumption constraints. In essence, the scheduling is a resource allocation problem for the specified optimization objectives, including the overall energy consumption, the overall processing cost, the overall processing delay, the number of concurrent communications and other comprehensive performance indexes. We can define the scheduling strategies and relevant parameters with the developing platform in the control plane. The scheduling method can be implemented by intelligent algorithms (e.g., the Genetic Algorithm and the Particle Swarm Optimization) or a heuristic algorithm [26]. An example of implementation by the Genetic algorithm is illustrated as follows.

1) **The definition of the fitness functions**: With different optimization objectives such as the maximization of the network lifetime and the highest processing efficiency, i.e., the shortest processing delay, we define different fitness functions as follows.

(1) The balance of node's lifetime

With the collected states of N WSAN nodes in the system, the SDN controller estimates the remain energy level and the lifetime L_i of the i-th node, based on the above-mentioned metrics. In addition, the SDN controller estimates the lifetime variation $\triangle L_i$ for all nodes after this task execution. The mean value of the lifetimes after task execution is

$$\overline{L} = \sum_{i=1}^{N} (L_i - \Delta L_i) \tag{1}$$

We minimize the standard variation of the lifetime for all nodes to achieve the longest lifetime of the whole WSAN; thus, the fitness function can be described as

$$F_{lifetime} = \sqrt{\frac{N}{\sum_{i=1}^{N} (L_i - \Delta L_i - \overline{L})^2}} \tag{2}$$

(2) The shortest processing delay

Given m subtasks to be processed and a scheduling scheme, the SDN controller estimates the processing delay T_k (including the task-execution delay in the node and the message-transmission delay to the next node) required for handling the k-th subtask with the collected network state, $1 \le k \le m$. Because we attempt to minimize the sum of these delays, the fitness function can be described as

$$F_{delay} = \frac{1}{\sum_{k=1}^{m} T_k} \tag{3}$$

2) **Chromosome encoding**: We arrange all tasks and the corresponding subtasks in a chromosome. Each gene represents a subtask, and the value of the gene is the ID of the WSAN node. Thus, each chromosome represents a complete scheme for all pending tasks. An example is shown in Fig. 9.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2547890, IEEE Access

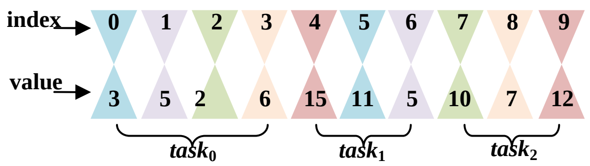


Fig. 9. Example of chromosome encoding

3) **The generation of the initial population**: Randomly construct *N* chromosomes as the initial population. Generally *N* is within the range of [50,120].

4) **The evaluation of the fitness value**: With predefined fitness functions representing different optimization objectives, we evaluate the fitness value of each chromosome.

5) **Chromosomes selection**: Based on the threshold of fitness function, excellent chromosomes are selected from the parent generation. Here, we adopt the proportional selection method [27], in which the ratio of the fitness value for a chromosome to the sum of fitness values for all chromosomes is considered as the selection probability.

6) **Chromosomes crossover**: The crossover is the most vital operation for producing a new generation and embodies the idea of information exchange. We perform the crossover for the gene segment of each task and select the starting position for the crossover in each segment randomly.

7) **Chromosomes mutation**: With a random small probability, we select some genes among all the chromosomes and change their values to embody the effect of mutation.

8) **The generation of new population**: After the aforementioned stages, we evaluate the values of the fitness functions for all chromosomes, select *N* chromosomes with the highest values of the fitness function as the new population, and then go to stage 5) for the next iteration.

When the number of generations is large enough, or the algorithm converges, we terminate the algorithm and output the chromosome with the highest value of fitness functions as the selected scheduling scheme.

VI. Applications

A. Typical application scenarios

The SDN technique shows promise for use in many application scenarios of the WSAN.

1) **Military control**: In the multi-layer military control system, with the cooperation of the scheduling engine, the SDN controller can sense and control the states of unmanned aerial vehicles, ships, ground vehicles and crew via satellite network and ground communication infrastructure. The main feature is the quick response to commands for efficient cooperative action (including attack, advance, and retreat).

2) **Environmental disaster monitoring**: In the case of extinguishing field fire, several unmanned aerial vehicles can sense the dynamic fire states, wind-force, wind direction, the state of crowd and other weather conditions. As a result, the SDN controller can make accurate decisions and control the actions of staffs and facilities, including rescue, firefighting, and retreat.

3) **Industry process management**: In the field of product production, the SDN controller can timely obtain the states of raw material supply, facilities operation and task execution. As a result, the system can provide efficient raw material delivery, facility selection and process control, and find the abnormal conditions of facilities and inferior products.

4) **Transportation management**: In the peak traffic period of cities, the SDN can dynamically set the traffic signals and make a globally optimal decision via fusion of the data from the monitoring equipment and avoid the aggravation or transfer of traffic congestion. When a crowd gathers in the metro or park, the SDN controller can make an accurate decision of early warning and control the exit or passage for public safety, if necessary.

B. Indoor experiment of the SDN-based video transmission

We have implemented an SDN-based mobility management system for an indoor mobile multimedia service network as shown in Fig. 10. In this network mobile clients can roam among different access points.

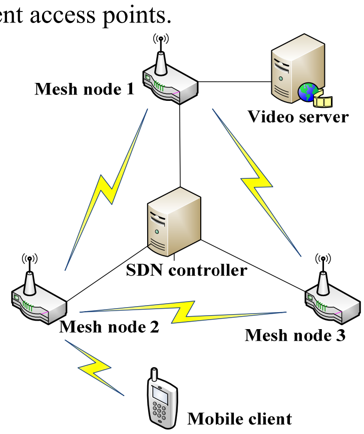


Fig. 10. The testbed of SDN-based roaming for indoor video transmission

In this system, the backbone of the wireless mesh network provides the data transmission path, and the wireless mesh nodes build a secure control path with the SDN controller. All packet transmissions are controlled by the OpenFlow with the support from each mesh node. When the signal strength for a link is below the threshold, the access point begins to disassociate with the client, and the SDN controller begins to reconnect the client with a new access point. For the fluency of transmission during the access point handoff, the Floodlight that runs in the SDN controller builds the forwarding rules for all mesh nodes and controls the volume and paths for all traffic flows.

The key techniques for implementation of the testbed in Fig. 10 are as follows:

1) Transplanting OpenvSwitch to the routing system OpenWrt [28].

2) Multiple-SSID virtual NIC interfaces.

3) The Floodlight-based control of data flows.

We apply this testbed to video-on-demand service and analyze the network performance. The settings are:

1) The wired connection between the Video server and node 1.

2) The mobile client obtains video via the VLC [29] player and builds a wireless connection with node 2.

3) The wired connection between each mesh node and the SDN controller.

4) There are 30 connection handoffs between node 2 and node 3 as the video is being transmitted to the mobile client.

The Wireshark [30] that runs in the mobile client obtains the exact time of packet arrival, and calculates the time T_{itv} required for data recovery after handoff, i.e., the arrival time of the first received packet after handoff minus the arrival time of the last received packet before handoff.

We observe that the average time required for video recovery is approximately 6.7 seconds. The time increases to above 8 seconds for 7 times among the 30 handoffs. Thus, the SDN-based mesh network is proven to support the function of transparent roaming for mobile devices. However, the drawback is the relatively long time for data recovery in spite of the continuity of transmission. One countermeasure is the buffering size for more than 8 seconds [31]. In addition, higher capacity of hardware is also helpful.

B. Outdoor experiment of the SDN-based sensor data collection

As shown in Fig. 11, we apply the SDN approach in a hybrid network for water quality monitoring in the Three Gorges, China. The hybrid network is composed of a wireless mesh network as the backbone, including a mesh node installed in a vessel with the implementation of the SDN controller, several fixed mesh nodes installed in the shoreside of the river for data transmission in the data plane, and several wireless sensor networks as the subnets that stem from each mesh node for data collection in the data plane.

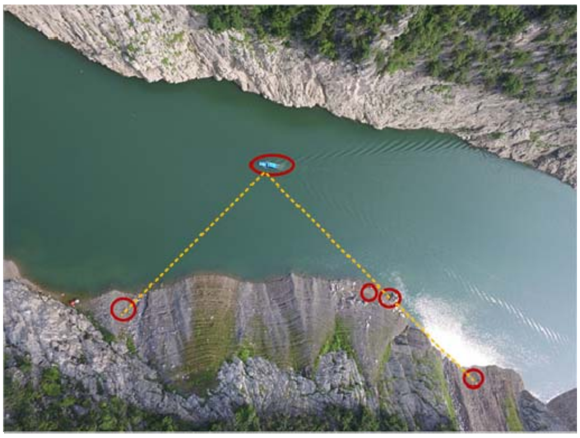


Fig. 11. The experiment of SDN-based hybrid network for water quality monitoring

The vessel moves along the river and thus roams with frequent communication handoffs among all fixed mesh nodes. Each time, the collected data of water quality are forwarded from the sensors in the vessel to the remote server through the vessel mesh node and one fixed mesh node. When the signal strength for the working link between the vessel node and the fixed mesh node is below the threshold, the mesh nodes actively invoke the re-association operation. The Floodlight-based SDN controller makes decisions of packets forwarding for the wireless mesh networks to enable the backbone mesh network to select the proper path and guarantee smooth communication.

In the experiment, the data regarding water quality and GPS is collected every 15 seconds. Based on the Wireshark that runs in the vessel node, the analysis result shows that the time required for recovery from handoff is approximately 6 seconds. Thus, the server cannot lose data because of roaming handoff.

VII. FUTURE TRENDS AND CHALLENGES

WSAN will be applied for many applications in the future. As the applications expand, there exist several evolving trends:

1) **Big data and cognitive methods**

As the information technology develops, data will emerge with prominent characteristics in variety, volume, velocity, value and veracity [32, 33], especially for multimedia data or the data from complex processes [34]. There exist many technical challenges for unstructured or structured data, including incompleteness [35], heterogeneity, redundancy, privacy [36, 37] and delay sensitivity. Thus, it is necessary to design effective cognitive methods for supporting SDN control. We can also predict the entities' behaviors and evaluate the services with relevant cognitive methods [38]. The objects that are involved in cognition include the users' behaviors, the states of the tasks and the state of the network.

2) **The evolution of the SDN platforms**

With the rapid expansion of the scope and level of WSAN applications, the effective organization and combination of different WSAN platforms become a critical issue. In the data plane of the propose framework, the sensor and actor nodes should be efficiently employed by the SDN controller with relevant methods for deployment, sharing, and scheduling to save the cost for initial deployment and later maintenance in an individual WSAN.

Given the large scale of WSAN, many SDN controllers must work collaboratively for high scalability and reliability of applications. The early works include the Kandoo [39] in vertical fashion, together with the Onix [40] and the HyperFlow [41] in horizontal fashion. In the future, the SDN controller cloud [42] may be presented to provide robust services for network control. The relevant issues include the resource virtualization in the south interface and service publishing in the north interface [43].

Alternatively, it is necessary to design methods to transform the traditional WSAN to the Sensor and Actor Cloud [39] with the capacity and intelligence from the SDN controllers. Thus, by the optimal configurations for various physical resources and their collaboration behaviors, we can handle large number of tasks and realize different applications with satisfactory performance.

REFERENCES MONTH

[1] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio and L. T. Yang, "A survey on communication and data management issues in mobile sensor networks," Wireless Communications and Mobile Computing, vol. 14, no. 1, pp. 19-36, Jan. 2014.

[2] C. Zhu, V. C. M. Leung, L. Shu and E. C.-H. Ngai, "Green Internet of Things for smart world," IEEE Access, vol. 3, pp. 2151-2162, Nov. 2015.

[3] N. Sabri, S. A. Aljunid, M. S. Salim, R. Kamaruddin, R. B. Ahmad and M. F. Malek, "Cognitive wireless sensor actor network: An agricultural perspective," International Journal of Innovative Computing, Information and Control, vol. 10, no. 2, pp. 631-658, Apr. 2014.

[4] M. Erol-Kantarci and H.T. Mouftah, "Wireless multimedia sensor and actor networks for the next generation power grid," Ad Hoc Networks, vol. 9, no. 4, pp. 542-551, Jun. 2011.

[5] Al-Anbagi I., Erol-Kantarci M. and Mouftah H. T., "Priority-and delay-aware medium access for wireless sensor networks in the smart grid," IEEE Systems Journal, vol. 8, no. 2, pp. 1-11, Jun. 2014.

[6] C. Zhu, L. T. Yang, L. Shu, V. C. M. Leung, T. Hara and S. Nishio, "Insights of Top-k query in duty-cycled wireless sensor networks," IEEE Transactions on Industrial Electronics, vol. 62, no. 2, pp. 1317-1328, Feb. 2015.

[7] C. Zhu, L. T. Yang, L. Shu, V. C. M. Leung, J. J. P. C. Rodrigues and L. Wang, "Sleep scheduling for geographic routing in duty-cycled mobile sensor networks," IEEE Transactions on Industrial Electronics, vol. 61, no. 11, pp. 6346–6355, Nov. 2014.

[8] C. Zhu, X. Li, V. C. M. Leung, X. Hu and L. T. Yang, "Job scheduling for cloud computing integrated with wireless sensor network," in Proc. of 6th IEEE International Conference on Cloud Computing Technology and Science, 2014, pp. 62-69.

[9] K. Wang, Y. Shao, L. Shu, G. Han and C. Zhu, "LDPA: a local data processing architecture in ambient assisted living communications," IEEE Communications Magazine, vol. 53, no. 1, pp. 56-63, Jan. 2015.

[10] L. Shu, L. Wang, J. Niu, C. Zhu and M. Mukherjee, "Releasing network isolation problem in group-based industrial wireless sensor networks," IEEE Systems Journal, vol. PP, no. 99, pp. 1-11, Oct. 2015.

[11] Nunes B. A. A., Mendonca M., Nguyen X.-N., Obraczka K. and Turletti T., "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE Communications Surveys and Tutorials, vol. 16, no. 3, pp. 1617-1634, Sep. 2014.

[12] Bertaux L., Medjiah S., Berthou P. and Abdellatif S., "Software defined networking and virtualization for broadband satellite networks," IEEE Communications Magazine, vol.53, no.3, pp. 54-60, Mar. 2015.

[13] T. Luo, H.-P. Tan and Quek T.Q.S., "Sensor OpenFlow: Enabling Software-defined wireless sensor networks," IEEE Communications Letters, vol.16, no.11, pp. 1896-1899, Nov. 2012.

[14] A. De Gante，M. Aslan and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in Proc. of 27th Biennial Symposium on Communications, 2014, pp. 71-75.

[15] A. Lara, A. Kolasani and B. Ramamurthy, "Network innovation using OpenFlow: A survey," IEEE Communications Surveys and Tutorials, vol. 16, no. 1, pp. 493-512, Mar. 2014.

[16] Qin Z.J., Denker G., Giannelli C., Bellavista P. and Venkatasubramanian N, "A software defined networking architecture for the internet-of-things," in Proc. of IEEE/IFIP Network Operations and Management Symposium, 2014, pp. 1-9.

[17] V. C. A. Leonardo, B. P. Alberto, B. L. L. Isabel and G. V. L. Javier, "SDN: Evolution and opportunities in the development IoT applications," International Journal of Distributed Sensor Networks, vol. 2014, ID. 735142, pp. 1-10, 2014.

[18] Wang Y., Chen H., Wu X and Shu L., "An energy-efficient SDN based sleep scheduling algorithm for WSNs," Journal of Network and Computer Applications, vol. 59, pp. 39-45, Jan. 2016.

[19] Han Z.J. and Ren W., "A novel wireless sensor networks structure based on the SDN," International Journal of Distributed Sensor Networks, vol. 2014, ID. 874047, pp. 1-7, 2014.

[20] Savarese G., Vaser M. and Ruggieri M., "A software defined networking-based context-aware framework combining 4G cellular networks with M2M," in Proc. of 16th International Symposium on Wireless Personal Multimedia Communications, 2013, ID. 6618626.

[21] Tadinada V. R., "Software defined networking: Redefining the future of internet in IoT and cloud era," in Proc. of 2nd International Conference on Future Internet of Things and Cloud, 2014, pp. 296-301.

[22] D. O. B. Trevizan, M. C. Borges and G. L. Batista, "TinySDN: Enabling multiple controllers for software-defined wireless sensor networks," in Proc. of IEEE Latin-America Conference on Communications, 2014, pp. 1-6.

[23] Zhang S.Q., Dong Y.J., Dong Z.Y. and Cui Y., "A universal platform for wireless sensor and actor network applications," in Proc. of IEEE International Conference on Progress in Informatics and Computing, 2010, vol. 2, pp. 1024-1028.

[24] E. Canete, J. Chen, R. M. Luque and B. Rubio, "NeuralSens: A neural network based framework to allow dynamic adaptation in wireless sensor and actor networks," Journal of Network and Computer Applications, vol. 35, no. 1, pp. 382-393, Jan. 2012.

[25] He S., Li X., Chen J., Cheng P., Sun Y. and S.-R. David, "EMD: Energy-efficient p2p message dissemination in delay-tolerant wireless sensor and actor networks," IEEE Journal on Selected Areas in Communications, vol. 31, no. 9, pp. 75-84, Sep. 2013.

[26] S. Khan, M. Bilal, M. Sharif and F. A. Khan, "A solution to bipartite drawing problem using Genetic Algorithm," in Lecture Notes in Computer Science vol. 6728, pp. 530-538, 2011.

[27] I. H. Toroslu and Y. Arslanoglu, "Genetic algorithm for the personnel assignment problem with multiple objectives," Information Sciences, vol.177, no.3, pp.787-803, Feb. 2007.

[28] The OpenWrt [Online]. Available: https://openwrt.org/

[29] The VLC player [Online]. Available: http://www.videolan.org/

[30] The Wireshark [Online]. Available: https://www.wireshark.org/

[31] Hu H., Wen Y.G., Chua T.-S., Huang J., Zhu W.W. and Li X.L., "Joint content replication and request routing for social video distribution over Cloud CDN: A community clustering method," IEEE Transactions on Circuits and Systems for Video Technology. Accepted, 2016.

[32] F. Tao, Y. Cheng, L.D. Xu and L. Zhang, "CCIoT-CMfg: Cloud computing and internet of things-based cloud manufacturing service system," IEEE Transactions on Industrial Informatics, vol.10, no.2, pp.1435-1442, May. 2014.

[33] H. Hu, Y.G. Wen, T.-S. Chua and X.L. Li. "Toward scalable systems for big data analytics: A technology tutorial," IEEE Access. vol.2, pp. 652-687, Jul. 2014.

[34] Hu H., Wen Y.G., Gao Y., Chua T.-S. and Li X.L., "Toward an SDN-enabled big data platform for social TV analytics," IEEE Network, vol. 29, no. 5, pp. 43-49, Sep. 2015.

[35] C. Zhu, H. Wang, X. Liu, L. Shu, L. T. Yang and V. C. M. Leung, "A novel sensory data processing framework to integrate sensor networks with mobile cloud," IEEE Systems Journal, vol. PP, no. 99, pp. 1-12, Jan. 2014.

[36] C. Zhu, H. Nicanfar, V. C. M. Leung and L. T. Yang, "An authenticated trust and reputation calculation and management system for cloud and sensor networks integration," IEEE Transactions on Information Forensics and Security, vol. 10, no. 1, pp. 118-131, Jan. 2015.

[37] C. Zhu, V. C. M. Leung, L. T. Yang, L. Shu, J. J. P. C. Rodrigues and X. Li, "Trust assistance in sensor-cloud," in Proc. IEEE Conference on Computer Communications Workshops, 2015, pp. 342-347.

[38] W.X. Li, V. C.M. Leung, C.S. Zhu and Y.J. Ma, "Scheduling and routing methods for cognitive radio sensor networks in regular topology," Wireless Communications and Mobile Computing. vol. 16, pp. 47-58, Jan. 2016.

[39] Yeganeh S.H. and Ganjali Y., "Kandoo: A framework for efficient and scalable offloading of control applications," in Proc. of the ACM SIGCOMM Workshop on HotSDN, 2012, pp. 19-24 .

[40] Koponen T, Casado M, Gude N, Stribing J, Poutievski L, Zhu M, Ramanathan R, Iwata Y, Inoue H, Hama T and Shenker S, "Onix: A distributed control platform for large-scale production networks," in Proc. of the 9th USENIX Conference on Operating Systems Design and Implementation, 2010, ID 1-6.

[41] Tootoonchian A and Ganjali Y, "HyperFlow: A distributed control plane for OpenFlow," in Proc. of the USENIX Internet Network Management Conference on Research on Enterprise Networking, 2010, pp. 3-3.

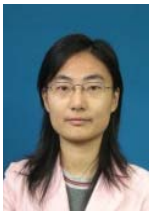
[42] C. Zhu, V. C. M. Leung, L. T. Yang and L. Shu, "Collaborative location-based sleep scheduling for wireless sensor networks integrated with mobile cloud computing," IEEE Transactions on Computers, vol. 64, no. 7, pp. 1844 -1856, Jul. 2015.

[43] C. Zhu, Z. Sheng, V. C. M. Leung, L. Shu and L. T. Yang, "Towards offering more useful data reliably to mobile cloud from wireless sensor

network," IEEE Transactions on Emerging Topics in Computing , vol. 3, no. 1, pp. 84-94, Mar. 2015.

Jianguo Zhou was born in 1965. He received his B. Eng degree of computer and automation at the Sichuan University, China in 1988. He received his M. Eng degree of radio electronics and Ph. D degree of communication and information system at the Wuhan University, China in 1991 and 2013 respectively. He is currently an associate professor at the Wuhan University, China. He serves as the senior member of China Institute of Communications. His research interest includes computer networks and the spatial information networks. He has published more than 50 papers in different journals and conferences.

Hao Jiang was born in 1976. He received his B. Eng degree of communication engineering at the Wuhan University, China in 1999. He received his M. Eng degree and Ph. D degree of communication and information system at the Wuhan University, China in 2001 and 2004 respectively. He undertook his post-doc research work in the LIMOS, Clermont-Ferrand, France from Oct. 2004 to Oct. 2005, and was a visiting professor at the University of Calgary, Canada and ISIMA, B. Pascal University, France. He is currently a professor at the Wuhan University, China. His research interest includes mobile Ad hoc network and mobile Big Data. He has published more than 60 papers in different journals and conferences.



Jing Wu (M'12) was born in 1981. She received her B. Eng degree of communication engineering at the Wuhan University, China in 2002, and received her Ph. D degree of communication and information system at the Wuhan University, China in 2007. She undertook her post-doc research work in the LIMOS, Clermont-Ferrand, France from Nov. 2009 to Nov. 2010. She is currently an assistant professor at the Wuhan University, China. Her research interest includes the spatial information network and the network protocols verification. She has published more than 40 papers in different journals and conferences.

Lihua Wu received his Bachelor's degree of communication engineering at the Wuhan University, China in 2009. He is currently a Ph.D. candidate in the School of Electronic Information, Wuhan University, China. He was also a visiting scholar in the Department of Electrical and Computer Engineering, the University of Calgary, Canada. His research interests are in the area of complex networks, big data, mobile internet, and wireless communication networks.

Chunsheng Zhu (S'12) received the B.E. degree in Network Engineering from Dalian University of Technology, China, in 2010. He obtained the M.Sc. degree in Computer Science from the St. Francis Xavier University, Canada, in 2012. He is currently a PhD Student with the Four-Year-Fellowship in the Department of Electrical and Computer Engineering at the University of British Columbia in Canada from Sep. 2012. He has around 40 papers published or accepted by refereed international journals (e.g., IEEE Transactions on Industrial Electronics, IEEE Transactions on Computers, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Emerging Topics in Computing and IEEE Systems Journal) and conferences (e.g., IEEE Globecom and IEEE ICC). His research interests are in the areas of wireless sensor networks and mobile cloud computing.

Wenxiang Li (M'14) was born in 1979. He received his B. Eng degree of communication engineering at the Wuhan University, China in 2001. He received his M. Eng degree and Ph. D degree of communication and information system at the Wuhan University, China in 2004 and 2010 respectively. He is currently an associate professor at the Wuhan University of Science and Technology, China, and was a visiting professor at the University of British Columbia, Canada. His research interest includes cognitive wireless networks and cloud computing. He has published more than 50 papers in different journals and conferences.