Improving Sensor-Cloud: Energy Efficiency, Security, Sensory Data Transmission, and Quality of Service

by

Chunsheng Zhu

B.E., Dalian University of Technology, 2010
M.Sc., St. Francis Xavier University, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

June 2016

Abstract

Recently, induced by incorporating 1) the ubiquitous data gathering capabilities of wireless sensor networks (WSNs) as well as 2) the powerful data storage and data processing abilities of cloud computing (CC), Sensor-Cloud is attracting growing attention from both academia and industry. However, Sensor-Cloud is still in its infancy and a lot of research efforts are expected to emerge in this area.

Improving Sensor-Cloud, this thesis first presents the important research issues that are yet to be widely investigated by other researchers regarding the energy efficiency, security, sensory data transmission and quality of service (QoS) of Sensor-Cloud, respectively. Further, our accomplished work regarding solving the identified research issues is described.

Particularly, two collaborative location-based sleep scheduling (CLSS) schemes are designed. Based on the locations of mobile users, CLSS dynamically determines the awake or asleep status of each sensor node to reduce energy consumption of the WSN integrated with mobile cloud. An authenticated trust and reputation calculation and management (ATRCM) system is introduced. ATRCM considers i) the authenticity of cloud service provider (CSP) and sensor network provider (SNP); ii) the attribute requirement of cloud service user (CSU) and CSP; iii) the cost, trust, and reputation of the service of CSP and SNP. A mechanism named TPSS is shown. TPSS consists of two main parts: 1) time and priority-based selective data transmission (TPSDT) for WSN gateway to selectively transmit sensory data to the cloud

and 2) priority-based sleep scheduling (PSS) algorithm for WSN to save energy consumption. Trust-Assisted Sensor-Cloud (TASC) is exhibited. In TASC, the sensory data is gathered and transmitted to cloud, by the trusted sensors (i.e., sensors which own trust values surpassing a threshold) in WSN. The sensory data is stored, processed and on demand delivered to users, by the trusted data centers (i.e., data centers which own trust values surpassing a threshold) in cloud.

The analytical and experimental results conducted in our work show that the proposed approaches can effectively alleviate the corresponding research issues, respectively. We hope our work can attract more research into Sensor-Cloud to make it develop faster and better.

Preface

About this thesis, there are a number of related publications as listed below. Chapter 2 is related to J1 and C1. Chapter 3 is related to J2 and C2. Chapter 4 is related to J3. Chapter 5 is related to J4 and C3. Chapter 1 and Chapter 6 are fully or partially related to all the listed publications.

With respect to all the listed publications, I am the principal contributor, regarding the needed devotion (reviewing related work, identifying research issues, proposing solutions, performing analysis, conducting evaluations, writing manuscripts). The contributions of my co-authors are shown as follows. My supervisor (i.e., Prof. Victor C. M. Leung), supervised all the listed publications as well as improved their writing. Dr. Hasen Nicanfar who was a PhD Student at The University of British Columbia, assisted me in revising J2, based on reviewers' comments about authentication and system security analysis. Dr. Zhengguo Sheng who was a Postdoctoral Fellow at The University of British Columbia, assisted me in analyzing the research issues in J3. Mr. Xiuhua Li who is a PhD Student at The University of British Columbia, assisted me in performing the mathematical analysis in J4, J5 and C7. Mr. Hai Wang who is a PhD Student at Toyota Technological Institute at Chicago in United States, assisted me in conducting the evaluations in J6, C5 and C6. The remaining co-authors (i.e., Prof. Laurence T. Yang, Prof. Lei Shu, Prof. Edith C.-H. Ngai, Prof. Joel J. P. C. Rodrigues, Prof. Takahiro Hara, Prof. Shojiro Nishio, Prof. Lei Wang, Prof. Wenxiang Li, Prof. Wei Chen, Prof. Hong Ji, Prof. Xi Li, Dr. Xiping Hu, Mr. Xiulong

Liu), helped me by providing suggestions about the introduction sections of the related publications.

Publications related to Chapter 1, Chapter 2 and Chapter 6

J1: Chunsheng Zhu, Victor C. M. Leung, Laurence T. Yang, and Lei 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. (One of the popular articles published in IEEE Transactions on Computers for Dec. 2015)

C1: Chunsheng Zhu, Victor C. M. Leung, Laurence T. Yang, Xiping Hu, and Lei Shu, "Collaborative Location-based Sleep Scheduling to Integrate Wireless Sensor Networks with Mobile Cloud Computing," in *Proc. IEEE Global Communications Conference Workshops (GLOBECOM WKSHPS)*, 2013, pp. 452-457.

Publications related to Chapter 1, Chapter 3 and Chapter 6

J2: Chunsheng Zhu, Hasen Nicanfar, Victor C. M. Leung, and Laurence 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.

C2: Chunsheng Zhu, Hasen Nicanfar, Victor C. M. Leung, Wenxiang Li, and Laurence T. Yang, "A Trust and Reputation Management System for Cloud and Sensor Networks Integration," in *Proc. IEEE International Conference on Communications (ICC)*, 2014, pp. 557-562.

Publications related to Chapter 1, Chapter 4 and Chapter 6

J3: Chunsheng Zhu, Zhengguo Sheng, Victor C. M. Leung, Lei Shu, and Laurence 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. (One of the popular articles published in IEEE Transactions on Emerging Topics in Computing for Dec. 2015)

Publications related to Chapter 1, Chapter 5 and Chapter 6

J4: Chunsheng Zhu, Xiuhua Li, Victor C. M. Leung, Laurence T. Yang, and Lei Shu, "Towards Trustable Sensor-Cloud: Trust-Assisted Sensor-Cloud," *To be submitted to a journal in IEEE Transactions*, 13 pages, double column, 2016.

C3: Chunsheng Zhu, Victor C. M. Leung, Laurence T. Yang, Lei Shu, Joel J. P. C. Rodrigues, and Xiuhua Li, "Trust Assistance in Sensor-Cloud," in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015, pp. 342-347.

Publications partially related to Chapter 1 and Chapter 6

J5: Chunsheng Zhu, Xiuhua Li, Victor C. M. Leung, Laurence T. Yang, Edith C.-H. Ngai, and Lei Shu, "Towards Pricing for Sensor-Cloud," *Submitted to a journal in IEEE Transactions*, 12 pages, double column, Nov. 2015.

J6: Chunsheng Zhu, Hai Wang, Xiulong Liu, Lei Shu, Laurence T. Yang, and Victor C. M. Leung, "A Novel Sensory Data Processing Framework to Integrate Sensor Networks With Mobile Cloud," *Accepted for publication in IEEE Systems Journal*, 12 pages, double column, Jan. 2014.

J7: Chunsheng Zhu, Victor C. M. Leung, Lei Shu, and Edith C.-H. Ngai, "Green Internet of Things for Smart World," *IEEE Access*, vol. 3, pp. 2151-2162, Nov. 2015. (Invited paper) (One of the popular articles published in IEEE Access/IEEE Xplore for Dec. 2015)

J8: Chunsheng Zhu, Laurence T. Yang, Lei Shu, Victor C. M. Leung, Takahiro Hara, and Shojiro 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. (Nominated by IEEE Industrial Electronics Society Technology News in Jul. 2015 issue)

J9: Chunsheng Zhu, Laurence T. Yang, Lei Shu, Victor C. M. Leung, Joel J. P. C. Rodrigues, and Lei 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.

L1: Chunsheng Zhu, Lei Shu, Xiping Hu, Laurence T. Yang, and Victor C. M. Leung, "Review of CKN based Sleep Scheduling in Duty-Cycled Wireless Sensor Network," *IEEE CommSoft E-letters*, vol. 1, no. 2, pp. 5-9, Dec. 2012. (Invited paper)

C4: Chunsheng Zhu, Victor C. M. Leung, Xiping Hu, Lei Shu, and Laurence T. Yang, "A Review of Key Issues that Concern the Feasibility of Mobile Cloud Computing," in *Proc. IEEE International Conference on Cyber, Physical and Social Computing (CPSCom)*, 2013, pp. 769-776.

C5: Chunsheng Zhu, Victor C. M. Leung, Hai Wang, Wei Chen, and Xiulong Liu, "Providing Desirable Data to Users when Integrating Wireless Sensor Networks with Mobile Cloud," in *Proc. IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2013, pp. 607-614.

C6: Chunsheng Zhu, Hai Wang, Victor C. M. Leung, Lei Shu, and Laurence T. Yang, "An Evaluation of User Importance When Integrating Social Networks and Mobile Cloud Computing," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 2935-2940.

C7: Chunsheng Zhu, Xiuhua Li, Victor C. M. Leung, Xiping Hu, and Laurence T. Yang, "Job Scheduling for Cloud Computing Integrated with Wireless Sensor Network," in *Proc. IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2014, pp. 62-69.

C8: Chunsheng Zhu, Victor C. M. Leung, Edith C.-H. Ngai, Laurence T. Yang, Lei Shu, and Xiuhua Li, "Pricing Models for Sensor-Cloud," in *Proc. IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2015, pp. 454-457.

C9: Chunsheng Zhu, Xi Li, Hong Ji, and Victor C. M. Leung, "Towards Integration of Wireless Sensor Networks and Cloud Computing," in *Proc. IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2015, pp. 491-494.

Table of Contents

List of Tables

List of Figures

Glossary

AO	Always On
ATRCM	Authenticated Trust and Reputation Calculation and Management
BitC	Body-in-the-Cloud
CC	Cloud Computing
CLSS	Collaborative Location-based Sleep Scheduling
CSP	Cloud Service Provider
CSU	Cloud Service User
CTASC	Collaborative TASC
EC-CKN	Energy-Consumption based Connected K-Neighborhood
GPS	Global Positioning System
GS	General Scheme
IaaS	Infrastructure as a Service
IEC	International Electrotechnical Commission
ISMS	Information Security Management System
ISO	International Organization for Standardization
ITASC	Independent TASC

LooCI	Loosely-coupled Component Infrastructure
MCC	Mobile Cloud Computing
MTASC	Mutual TASC
PaaS	Platform as a Service
PDFs	Probability Density Functions
PLA	Privacy Level Agreement
PSS	Priority-based Sleep Scheduling
PTP	Point vs Time & Priority
QoS	Quality of Service
SaaS	Software as a Service
SC	Sensor-Cloud
SCWTA	SC Without Trust Assistance
SLA	Service Level Agreement
SNP	Sensor Network Provider
SSC	Social-Sensor-Cloud
TA	Trust Agent
TASC	Trust-Assisted Sensor-Cloud
TCE	Trusted Center Entity
TPSDT	Time and Priority-based Selective Data Transmission
WSN	Wireless Sensor Network

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Victor C. M. Leung. He offered me constant guidance, support and encouragement regarding both my research and my life. Without his commitment, this thesis would be impossible. I learn a lot from him and I will never forget the effort and time he devoted. I hope he is healthy and happy every day.

I also want to thank my supervisory committee members, university examiners, external examiner, and co-authors. They provided me valuable advices about presenting and improving my work. It is an honor for me to have them and I am grateful to all of them.

In addition, I shall thank my colleagues and friends, for their kind help in my life. I wish them all the best.

Finally, I am greatly indebted to my family members, for their endless and unconditional love. I deeply appreciate them and I love them.

All the work about the thesis topic was supported by a Four Year Doctoral Fellowship from The University of British Columbia and by funding from the Natural Sciences and Engineering Research Council of Canada, the ICICS/TELUS People & Planet Friendly Home Initiative at The University of British Columbia, TELUS and other industry partners.

Dedication

To my family

Chapter 1

Introduction

1.1 Motivation

1.1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) [1] [2] [3] are networks consisting of spatially distributed autonomous sensors, which are capable of sensing the physical or environmental conditions (e.g., temperature, sound, vibration, pressure, motion, etc.). WSNs are widely focused due to their great potential in areas of civilian, industry and military (e.g., forest fire detection, industrial process monitoring, traffic monitoring, battlefield surveillance, etc.), which could change the traditional way for people to interact with the physical world. For instance, regarding forest fire detection, since sensor nodes can be strategically, randomly, and densely deployed in a forest, the exact origin of a forest fire can be relayed to the end users before the forest fire turns uncontrollable without the vision of physical fire. In addition, with respect to battlefield surveillance, as sensors are able to be deployed to continuously monitor the condition of critical terrains, approach routes, paths and straits in a battlefield, the activities of the opposing forces can be closely watched by surveillance center without the involvement of physical scouts.

1.1.2 Cloud Computing

Cloud computing (CC) [4] [5] [6] is a novel computing model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services). CC is featured by that users can elastically utilize the infrastructure (e.g., networks, servers, and storages), platforms (e.g., operating systems and middleware services), and softwares (e.g., application programs) offered by cloud providers in an on-demand manner. Not only the operating cost and business risks as well as maintenance expenses of service providers can be substantially lowered with CC, but also the service scale can be expanded on demand and web-based easy access for clients could be provided benefiting from CC.

By further integrating CC into a mobile environment, mobile cloud computing (MCC) [7] [8] [9] can offload much of the data processing and storage tasks from mobile devices (e.g., smart phones, tablets, etc.) to the cloud. MCC is widely considered to not only greatly relieve the processing, storage and energy capacity limitations of mobile devices, but also provide users with many new mobile services (e.g., mobile cloud learning, mobile cloud gaming, mobile cloud healthcare). For instance, regarding mobile cloud learning, traditional mobile learning may encounter various issues (e.g., high storage cost, low processing speed and limited education resources) on mobile devices. By moving the learning platform to the cloud, learners and teachers can achieve much faster processing speed as well as much richer learning and teaching resources in terms of available information, using just a simple client on the mobile device.

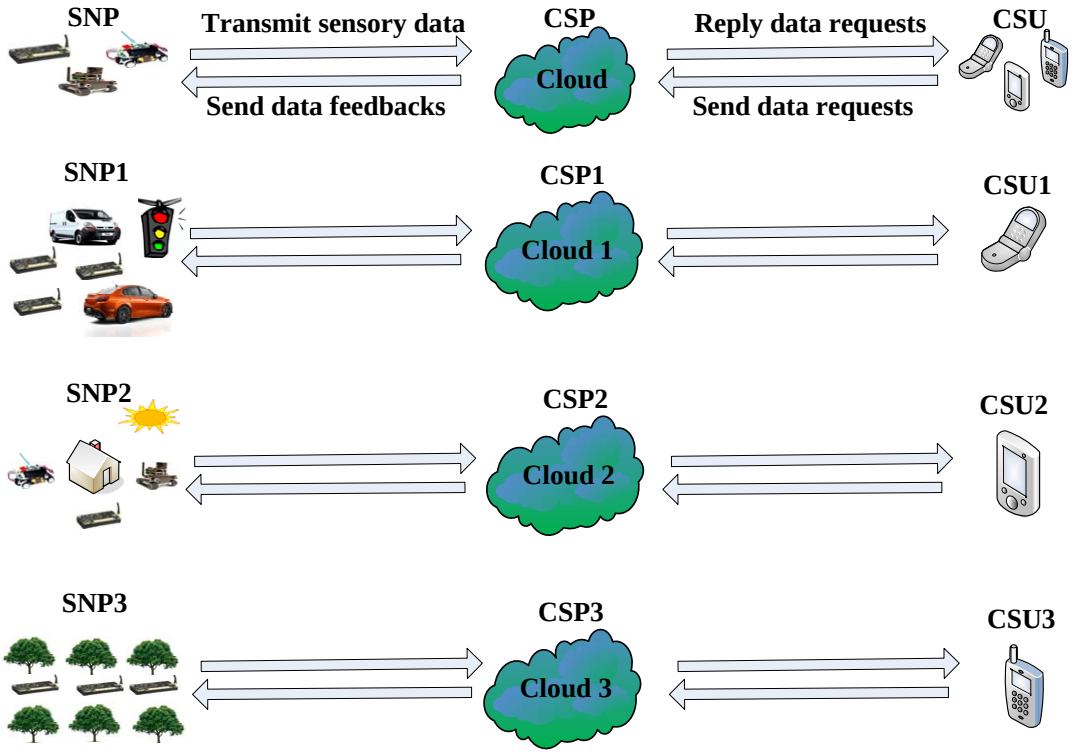


Figure 1.1: Examples of Sensor-Cloud

1.1.3 Sensor-Cloud

Defined by IntelliSys[1], Sensor-Cloud is "an infrastructure that allows truly pervasive computation using sensors as an interface between physical and cyber worlds, the data-compute clusters as the cyber backbone and the internet as the communication medium" [10] [11]. According to MicroStrains[2], Sensor-Cloud is "a unique sensor data storage, visualization and remote management platform that leverages powerful cloud computing technologies to provide excellent data scalability, rapid visualization, and user programmable analysis" [10] [11].

Attracting increasing interest from both academic and industrial communities, *Sensor-Cloud* (i.e., *WSN-CC integration*, *WSN-MCC integration*, *CC-WSN integration*, *MCC-WSN integration*, *SC*) [10] [11] [12] [13] [14] [15] [16] is actually a new paradigm, motivated by incorporating 1) the ubiquitous data gathering capabilities of WSNs as well as 2) the powerful data storage and data processing abilities of CC. Specifically, as shown in Fig. 1.1, sensor network provider (SNP), cloud service provider (CSP) and cloud service user (CSU) are included in Sensor-Cloud. The SNP offers different types of sensor nodes (e.g., static sensors, mobile sensors, video sensors) which form the WSN, collecting various sensory data (e.g., temperature, humidity, motion, sound, vibration, pressure) about the surrounding environment. The CSP provides the powerful cloud which is consisted of data centers, storing and processing the sensory data transmitted from the WSN. The CSU is the sensory data buyer, issuing data requests to the cloud on demand. In this whole process, SNPs act as the sensory data sources for CSPs. CSUs are the sensory data requesters for CSPs.

With Sensor-Cloud, there are a lot of advantages [10] [11] [12] [13] [14] [15] [16], benefiting the users

[1] http://www3.ntu.edu.sg/intellisys/index.html
[2] http://www.sensorcloud.com/system-overview

and the WSN as well as the cloud. For instance, the users can have access to their required sensory data from cloud anytime and anywhere if there is network connection, instead of being stick to their desks. The utility of WSN can be increased, by being able to serve multiple applications via the cloud. The services the cloud provides can be greatly enriched, by offering the services that the WSN provides (e.g., healthcare monitoring, environmental monitoring, forest fire detection, landslide detection, etc.). However, *Sensor-Cloud is still in its infancy and a lot of research efforts are expected to emerge in this area* [10] [11] [12] [13] [14] [15] [16].

1.2 Related Work

For the state of the art, current literatures about Sensor-Cloud mainly focus on the following two aspects: 1) improving the performance of WSN with cloud; 2) better utilizing the sensory data of WSN with cloud.

1.2.1 Improving the Performance of WSN with Cloud

About 1) improving the performance of WSN with cloud, a collaborative computing framework which integrates cloud and wireless body sensor networks is proposed in [17], reducing sensory data's transmissions and computation time to enhance the WSN's performance about detecting fall events and reconstructing 3-D motions. To effectively configure body sensor networks in an adaptive and stable manner by seeking the trade-offs among conflicting objectives (e.g., resource consumption and data yield), a cloud-integrated architecture named in Body-in-the-Cloud (BitC) is presented in [18]. For sharing the network resources among any two multimedia sensor nodes, a channel characterization scheme is shown in [19], combining a cross-layer admission control in dynamic cloud-based multimedia sensor networks. Aiming at improving the packet transmission error rate as well as the number of end-to-end hops of a WSN, an integration

architecture based on cloud and WSN is introduced in [20], which assumes that the cloud acts as a virtual sink with many sink points collecting the sensory data from sensors. About enhancing the latency performance as well as the memory of WSN, a framework to integrate WSN and cloud is designed in [21], combining a lightweight component model and a dynamic proxy-based approach. Lightweight component model utilizes the publicly available Loosely-coupled Component Infrastructure (LooCI) middleware for component management and dynamic proxies are added to the LooCI middleware to connect the sensors with the cloud.

1.2.2 Better Utilizing the Sensory Data of WSN with Cloud

Regarding 2) better utilizing the sensory data of WSN with cloud, a cloud-based platform (i.e., Wiki-Health) is introduced in [22], for storing, tagging, retrieving, analyzing, comparing and searching health sensory data. Similarly, the motivation of [23] is to design and develop a virtualized middleware platform based on cloud, to perform the collection, management and integration of sensory data from WSNs, while conducting the management of businesses enabled by the infrastructure of sensors. Moreover, considering the scenario that the cloud utilizes the sensory data to make real time alert in critical situations, an event matching algorithm based on subscriber category is proposed in [24], for distributing the sensory data to appropriate subscriber. In addition, to make adequate use of the sensory data gathered by a WSN, a framework to integrate CC and WSN is depicted in [25], where the requests of users are served via three service layers (i.e., Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)) either from an archive collecting sensory data from WSN to data centers, or from a live query which is issued to the corresponding WSN. For providing an effective and flexible security mechanism that guarantees confidentiality, integrity as well as fine grained access control to the e-health sensory data,

a secure and scalable cloud-based architecture is recommended in [26] to integrate e-health WSNs, by utilizing the cloud servers to ensure sensory data storage and sensory data encryption and decryption.

1.3 Research Issues

To the best of our knowledge, concerning Sensor-Cloud, *the following important research issues are yet to be widely investigated by other researchers*, in terms of the *energy efficiency, security, sensory data transmission and quality of service (QoS)* of Sensor-Cloud, respectively.

1.3.1 Research Issues about Energy Efficiency of Sensor-Cloud

- *MCC applications are often utilized in a location specific way* [27] [28] [29]. For example, the online work schedule application might be useful when the mobile user is on the way to work, but not when the mobile user is in a restaurant in the evening. Similarly, the traffic news application may be accessed by the mobile user to obtain the traffic information of a certain region before the mobile user actually gets there, while it is unlikely that the mobile user will always pay attention to the traffic news regardless of his or her current location. Also, thinking about a tourism navigation application which guides the mobile user to walk directly to the specific sightseeing place, such an application might be favorable when the mobile user is in fact in or near the tourism area. Whereas, it is not necessarily accessed when the mobile user is at home. In short, the current locations of mobile users usually determine the specific data mobile users might request.

- *Most sensors are usually equipped with non-rechargeable batteries with limited energy* [30] [31] [32]. If the sensor nodes continuously transmit the collected data to the cloud, the energy of these sensor

nodes will be depleted quickly and the lifetime of the WSN will be short.

1.3.2 Research Issues about Security of Sensor-Cloud

- *Authentication of CSPs and SNPs:* Malicious attackers may impersonate authentic CSPs to communicate with CSUs, or fake to be authentic SNPs to communicate with CSPs. Then CSUs and CSPs cannot eventually achieve any service from the fake CSPs and SNPs respectively. In the meantime, the trust and reputation of the genuine CSPs and SNPs are also impaired by these fake CSPs and SNPs.

- *Trust and reputation calculation and management of CSPs and SNPs:* Without trust and reputation calculation and management of CSPs and SNPs, it is easy for CSU to choose a CSP with low trust and reputation. Then the service from CSP to CSU fails to be successfully delivered quite often. Moreover, CSP may easily select an untrustworthy SNP that delivers the service that the CSP requests with an unacceptable large latency. In addition, the untrustworthy SNP probably may only be able to provide the requested service for a very short time period unexpectedly.

These two issues not only seriously impede the CSU from obtaining the desirable service they want from the authentic CSP, but also prevent the CSP from obtaining the satisfied service from the genuine SNP.

1.3.3 Research Issues about Sensory Data Transmission of Sensor-Cloud

In these potential applications of Sensor-Cloud (e.g., ubiquitous healthcare monitoring, environmental monitoring for disaster detection, agriculture and irrigation control, earth observation, transportation

and vehicle real-time visualization, tunnel monitoring, wildlife monitoring) [10], quite a number of them actually require the WSN to reliably offer sensory data that are more useful to the cloud based on the requests of the mobile users.

Take smart house monitoring as an instance, although various monitored information about the whole house gathered by the strategically deployed video sensors, image sensors and other types of sensors can be offloaded to the cloud to let the owner of the house or other authenticated and permitted people conveniently access their desired sensory data with the mobile devices (e.g., smart phones, tablet computers), it is expected that videos from some locations (e.g., storage room) are of little interest, while videos from other locations (e.g., front door, back door, windows) are considered to be more important to make sure that there is no unexpected intrusion into the house.

Thus, not all the sensory data are useful (i.e., actually utilized) for the cloud to satisfy user requests, while transmitting these data (i.e., multimedia data) to the cloud will use substantial network bandwidth. From this point, we can observe that 1) *sensory data that are more useful to the mobile users should be offered from WSN to cloud*. On the other hand, to perform the goal of monitoring the house intelligently, the WSN needs to successfully gather and transmit the collected information (e.g., videos, images) to the cloud continuously, which means that 2) *the sensory data should be reliably offered from the WSN to the cloud*.

1.3.4 Research Issues about QoS of Sensor-Cloud

Enhancing quality of service (QoS) of Sensor-Cloud with trust assistance: QoS is always a fascinating and valuable topic, as QoS (e.g., throughput, response time) always plays a vital role for users to eventually use the service (e.g., service provided by Sensor-Cloud). Trust is supposed to enhance the performance

of an existing system, by assigning trust value to the entity of the system. Potential way to improve the QoS of Sensor-Cloud (e.g., with trust assistance) is always worth to be explored.

1.4 Contributions of Thesis

Improving Sensor-Cloud, this thesis further describes our *accomplished work, aiming at tackling the identified research issues*. The performed analytical and experimental results demonstrate that the approaches proposed in our work can effectively mitigate the corresponding research issues respectively. We hope our work can attract more research into Sensor-Cloud, to make it develop faster and better. The detailed contributions of this thesis are summarized as follows.

- In Chapter 2, to solve the *identified research issues about energy efficiency of Sensor-Cloud*, we propose *two collaborative location-based sleep scheduling (CLSS) schemes* for WSNs integrated with MCC. Considering the locations of mobile users, the awake or asleep state of sensor nodes in the integrated WSN are dynamically changed in the CLSS schemes to reduce the energy consumption. Specially, CLSS1 focuses on maximizing savings in energy consumption of the integrated WSN and CLSS2 considers also the scalability and robustness of the integrated WSN. These schemes are further evaluated analytically and by simulations to show that they can prolong the lifetime of the integrated WSN while satisfying the data requests of the mobile users.

- In Chapter 3, for solving the *identified research issues about security of Sensor-Cloud*, we first analyze the authentication of CSPs and SNPs as well as the trust and reputation about the services of CSPs and SNPs. Further, we propose *an authenticated trust and reputation calculation and management (ATRCM) system* for CC-WSN integration. Particularly, considering (i) the authenticity of CSP

and SNP; (ii) the attribute requirement of CSU and CSP; (iii) the cost, trust and reputation of the service of CSP and SNP, the proposed ATRCM system achieves the following three functions: a) authenticating CSP and SNP to avoid malicious impersonation attacks; b) calculating and managing trust and reputation regarding the service of CSP and SNP; c) helping CSU choose desirable CSP and assisting CSP in selecting appropriate SNP. Detailed analysis and design as well as further functionality evaluation results are presented to demonstrate the effectiveness of ATRCM, followed with system security analysis.

- In Chapter 4, motivated by solving the *identified research issues about sensory data transmission of Sensor-Cloud* for supporting these WSN-MCC integration applications that require WSN to reliably collect and send data that are more useful to the mobile users to cloud, we first identify the critical issues that affect the usefulness of sensory data and reliability of WSN, and then propose *a WSN-MCC integration scheme named TPSS*. Specifically, TPSS includes two parts: 1) TPSDT (Time and Priority based Selective Data Transmission) for WSN gateway to selectively transmit sensory data that are more useful to the cloud, considering the time and priority features of the data requested by the mobile user; 2) PSS (Priority-based Sleep Scheduling) algorithm for WSN to save energy consumption so that it can gather and transmit sensory data in a more reliable way. Analytical and experimental results demonstrate the effectiveness of TPSS, in enhancing the usefulness of sensory data and reliability of WSN for WSN-MCC integration.

- In Chapter 5, triggered by solving the *identified research issues about QoS of Sensor-Cloud*, we introduce *TASC (Trust-Assisted SC)*, enhancing the QoS that the sensory data is achieved by users from SC. Specifically, the trusted sensors (i.e., sensors which own trust values surpassing a threshold) in WSN are utilized by TASC, for gathering and transmitting the sensory data to the cloud. Then

the trusted data centers (i.e., data centers which own trust values surpassing a threshold) in cloud are used by TASC, for storing and processing the sensory data as well as further on demand delivering the sensory data to users. Furthermore, regarding the application of TASC, we discuss the methods to compute the trust values of sensor nodes and data centers in TASC. In addition, concerning the design of TASC, we present three types of TASC (i.e., ITASC (independent TASC), CTASC (collaborative TASC), MTASC (mutual TASC)). With extensive analysis and numerical results, it is shown that the throughput and response time that the sensory data is achieved by users from SC can be greatly enhanced with TASC, in contrast to SCWTA (SC without trust assistance).

1.5 Organization of Thesis

For the rest of this thesis, Chapter 2 shows our work regarding solving the identified research issues about energy efficiency of Sensor-Cloud. The MCC-WSN integration model and the proposed CLSS schemes are shown, followed with the related theoretical analysis and numerical results. Chapter 3 describes our work to solve the identified research issues about security of Sensor-Cloud. The system model is described, followed with the discussion about the authentication of CSPs and SNPs as well as the trust and reputation about the services of CSPs and SNPs. Then the design, functionality evaluation results and security analysis about the proposed ATRCM system are shown. Chapter 4 presents our work motivated by solving the identified research issues about sensory data transmission of Sensor-Cloud. The critical issues that affect the usefulness of sensory data and reliability of WSN are discussed. Then the WSN-MCC integration system model and TPSDT as well as PSS are presented, followed with the introduction about the proposed TPSS scheme and its evaluations. Chapter 5 introduces our work triggered by solving the identified research issues about QoS of Sensor-Cloud. The system model and the proposed TASC are

introduced. With that, the trust values computation in TASC is discussed and three types of TASC (i.e., ITASC, CTASC and MTASC) are designed. Further, the analysis and numerical results about TASC and SCWTA are shown. Chapter 6 concludes the thesis and discusses the future work.

Chapter 2

Collaborative Location-based Sleep Scheduling for Wireless Sensor Networks Integrated with Mobile Cloud Computing

2.1 MCC-WSN Integration Model

The MCC-WSN integration model in this chapter is presented as follows and the main notations used in this chapter are summarized in Table 2.1 and Table 2.2.

2.1.1 Overall System Model

There is one cloud c and we assume that there are M mobile users (i.e., $u_1, u_2, \cdots, u_M$) as well as M multihop WSNs (i.e., $wsn_1, wsn_2, \cdots, wsn_M$). Each WSN acts as a sensory data source for the cloud to reply the data requests issued by each corresponding mobile user. We suppose that the mobile device used

Table 2.1: Main notation definitions in Chapter 2

Symbol	Definition
$\|\ \|$	Number of elements in a set
$\mathcal{A}$	Area of WSN
B	Set of links
c	Cloud
d	Transmission distance
e_a	Energy consumption of power amplification of one byte to cover a 1 m distance
e_r	Energy consumption of receiving a byte
e_t	Energy consumption of transmitting a byte
E_o	Node initial energy
E_R	Energy consumption of receiving a packet
E_T	Energy consumption of transmitting a packet
h	Packet length
i	Sensor node
I	Set of sensor nodes
k	k in EC-CKN
L	Mobile user location list
L_h	Mobile user location history list
L_p	Mobile user predication location list

Table 2.2: Main notation definitions continued in Chapter 2

Symbol	Definition
M	Number of mobile users or WSNs
N	Total number of sensor nodes in WSN
NL	Network lifetime
NL_0	Network lifetime of AO
NL_1	Network lifetime of CLSS1
NL_2	Network lifetime of CLSS2
NWR	Network work rate
NWR_0	Network work rate of AO
NWR_1	Network work rate of CLSS1
NWR_2	Network work rate of CLSS2
$p_{XY}(x, y)$	Independent probability distribution of events
q	Number of packets transmitted from a node to each neighbor node if an event is detected
s	Base station
t	Time epoch interval
t_r	Transmission radius
T	Time
TP	Time epoch
u	Mobile user
λ	Average event rate
ρ	Node density

by the mobile user has the global positioning system (GPS) and the mobile user utilizes the StarTrack service in [33]. There is a base station s with unlimited energy supply, serving as the gateway between each WSN and the cloud. Time T is divided into time epochs and the interval of each time epoch TP is t.

2.1.2 Overall WSN Model

The multihop WSN is uniformly randomly deployed with N sensor nodes in a two dimensions area $\mathcal{A}$. The whole network is modeled by a graph $G = (I, B)$, where $I = \{i_1, i_2, \cdots, i_N\}$ is the set of sensor nodes and $B = \{b_{(1,2)}, b_{(2,3)}, \cdots, b_{(N-1,N)}\}$ is the set of links. Each node has the same transmission radius t_r. Any two sensors i_i and i_j are neighbors if they are within the transmission range of each other. Any two sensors i_i and i_j are 2-hop neighbors if $b_{(i_i,i_j)} \notin B$ and there exists another node i_w satisfying $b_{(i_i,i_w)} \in B$, $b_{(i_w,i_j)} \in B$ or $b_{(i_j,i_w)} \in B$, $b_{(i_w,i_i)} \in B$.

2.1.3 WSN Energy Model

The energy consumed by a sensor to transmit and receive one byte, and power-amplify each transmitted byte to cover the distance of 1 m are e_t mJ, e_r mJ and e_a mJ/m^2, respectively. The energy model is the first order model shown as follows in [34] [35] [36]. Including the packet header and body containing the sensed data, the consumed energy to transmit and receive a packet of length h bytes over distance d are E_T, E_R:

$$E_T = e_t \cdot h + e_a \cdot h \cdot d^2 \tag{2.1}$$

$$E_R = e_r \cdot h \tag{2.2}$$

2.1.4 WSN Event Model

Generally, an event which has a characterizable distribution in space and time, is sensed when a sensor node receives a signal with power above a predetermined threshold. Assume that the temporal event behavior over the entire sensing region $\mathcal{A}$ is a Poisson process with an average event rate λ. In addition, given that the spatial distribution of events is characterized by an independent probability distribution given by $p_{XY}(x, y)$. Then the probability that an event occurring in A_i is detected by a sensor node i is as follows [37] [38].

$$p_{ei} = \frac{\int_{A_i} p_{XY}(x, y)}{\int_{\mathcal{A}} p_{XY}(x, y)} \tag{2.3}$$

2.2 Proposed CLSS Schemes

In this section, we first show the mechanisms to obtain the mobile user location list and then present our proposed CLSS schemes.

2.2.1 Mobile User Location List

Mobile User Location History List

To achieve the location list L of mobile user u, the location history of u is extracted by the cloud c based on the StarTrack service. Specifically, StarTrack in [33] is a mobile client application and it periodically

captures the user's current location (e.g., with GPS) and relays the location information to the StarTrack server which runs as a service in the cloud c. Further, the StarTrack server processes these location data and decomposes them into various tracks (i.e., discrete representations of trips taken by the mobile user). The points of these tracks are operational and retrievable through a high-level application programming interface and they make up the location history list named as L_h.

Mobile User Predication Location List

To obtain the mobile user predication location list L_p, we utilize the following method that is similar with the Place Transition Graph utilized as in [27]. The key idea is that the future locations of the mobile user would be associated with the frequently visited locations of the mobile user, thus it is likely that the future track of the mobile user will be constituted by these frequently visited locations. For instance, if a mobile user goes to restaurant A and gym B from office C very often, it is obvious that the mobile user will go to gym B from restaurant A, or go to restaurant A from gym B someday in the future.

Particularly, we compute a frequently visited location list L_f first. This L_f is obtained by iterating over all the retrieved tracks and selecting the end points of the retrieved tracks of the mobile user. Then L_f is updated by further removing the end points of the tracks that only appear once. With that, an adjacency matrix in which the numbers of rows and columns correspond to the number of the elements in the updated L_f is constructed. Finally, the match of each element in the row and the column except the match with two same points becomes a new track (i.e., the prediction track). All points without repetition excluding the starting and end points of the prediction tracks constitute the mobile user predication location list L_p.

The mobile user location history list L_h and mobile user predication location list L_p constitute the

Pseudocode of CLSS1 scheme
Step 1: Cloud c obtains mobile user u's current location l_u.
Step 2: If $l_u \in L$, c sends flag A to base station s. Otherwise, c sends s flag Z.
Step 3: s broadcasts flag to sensor nodes.
Step 4: Run Step 5 at each node i.
Step 5: If node i receives flag A, remain awake. Otherwise, go to sleep.

location list L of the mobile user.

2.2.2 CLSS Schemes

There are two collaborative location-based sleep scheduling (CLSS) schemes for the integrated WSN and the pseudocodes of these two CLSS schemes (i.e., CLSS1 and CLSS2) in each time epoch TP are as shown above and below.

CLSS1

Regarding CLSS1 scheme, cloud c first obtains the current location l_u of mobile user u (Step 1 of CLSS1). Then according to whether l_u is in the location list L or not, a flag A or Z is sent to base station s by cloud c (Step 2 of CLSS1). Base station s further broadcasts the flags. At last, each sensor node i determines its awake or asleep state according to the flag it receives in each time epoch TP (Step 3 to Step 5 of CLSS1).

CLSS2

In terms of CLSS2, the first 4 steps are the same as that of CLSS1. The difference between CLSS2 and CLSS1 lies in Step 5. In Step 5 of CLSS2, when sensor node i receives flag Z, i will be sleep scheduled using the energy-consumption based connected k-neighborhood (EC-CKN) sleep scheduling scheme [36].

Pseudocode of CLSS2 scheme

Step 1: Cloud c obtains mobile user u's current location l_u.
Step 2: If $l_u \in L$, c sends flag A to base station s. Otherwise, c sends s flag Z.
Step 3: s broadcasts flag to sensor nodes.
Step 4: Run Step 5 at each node i.
Step 5: If node i receives flag A, remain awake. Otherwise, run Step 6 to Step 12.

Step 6 to Step 12 are the pseudocodes of EC-CKN scheme

Step 6: Get the current residual energy $Erank_i$.
Step 7: Broadcast $Erank_i$ and receive the ranks of its currently awake neighbors N_i. Let R_i be the set of these ranks.
Step 8: Broadcast R_i and receive R_j from each $j \in N_i$.
Step 9: If $|N_i| < k$ or $|N_j| < k$ for any $j \in N_i$, remain awake. Go to Step 12.
Step 10: Compute $C_i = \{j | j \in N_i \text{ and } Erank_j > Erank_i\}$.
Step 11: Go to sleep if both the following conditions hold. Remain awake otherwise.

- Any two nodes in C_i are connected either directly themselves or indirectly through nodes within i's 2-hop neighborhood that have $Erank$ more than $Erank_i$.
- Any node in N_i has at least k neighbors from C_i.

Step 12: Return.

Regarding EC-CKN, the current residual energy rank (i.e., $Erank_i$) of each node i is obtained first (Step 6 of CLSS2) and the subset C_i of i's currently awake neighbors that have $Erank > Erank_i$ is computed (Step 10 of CLSS2). Before a node i can go to sleep in each time epoch TP, the following two conditions should hold: (1) all nodes in C_i are connected by nodes with $Erank > Erank_i$ (2) each of its neighbors owns at least k neighbors from C_i (Step 11 of CLSS2).

2.3 Theoretical Analysis

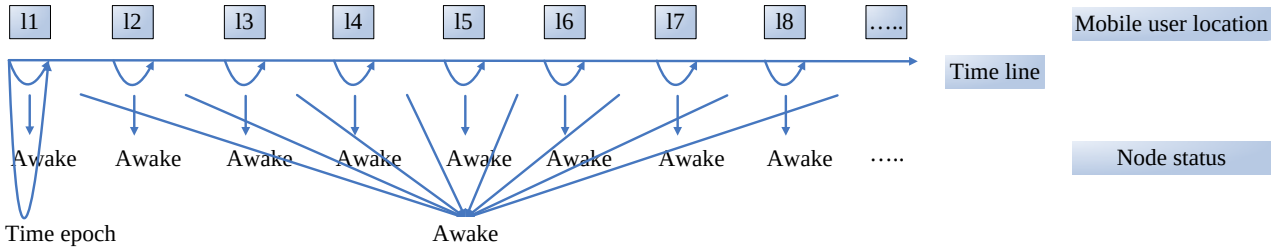


Figure 2.1: Example of time line of always on (AO) scheme

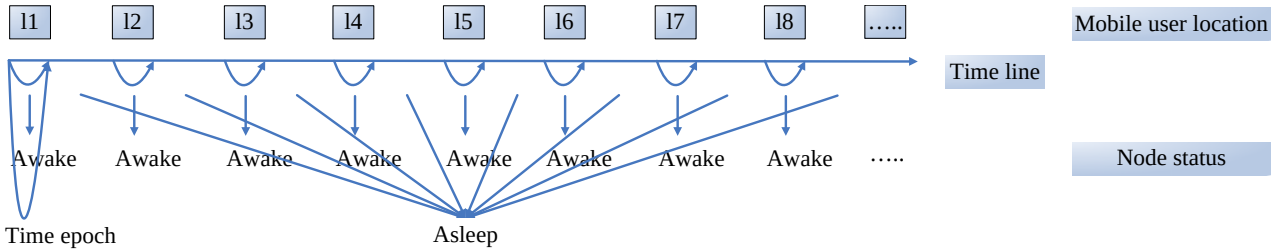


Figure 2.2: Example of time line of CLSS1

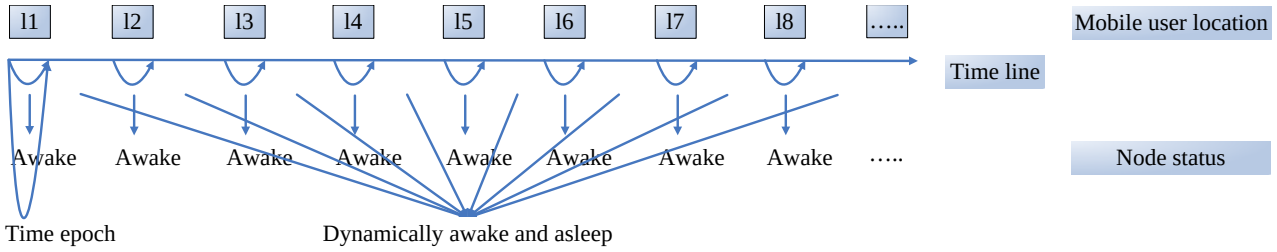


Figure 2.3: Example of time line of CLSS2

2.3.1 Time Line

Fig. 2.1 shows the example of the time line of the always on (AO) scheme, in which sensor nodes are always working to transmit the sensory data to the cloud. Fig. 2.2 and Fig. 2.3 present the examples of the time line of CLSS1 and CLSS2 schemes, respectively. Based on the graphical depictions in these three figures and the algorithmic descriptions in Section 2.2, we can observe the following properties about these three schemes in each time epoch TP: 1) In AO, no matter the mobile user's current location is in L or not, the sensor node is always awake; 2) About CLSS1, when the location of the mobile user is in L, the sensor nodes will be awake. Otherwise, the sensor nodes will be asleep to save energy; 3) Regarding CLSS2, if the mobile user's current location is in L, the sensor nodes will also be awake. Otherwise, the sensor nodes are dynamically asleep and awake for saving energy based on EC-CKN.

2.3.2 Preliminaries

From [36] [39] [40], we can determine that for a uniformly randomly deployed WSN with N nodes in a two dimensional area $\mathcal{A}$, the node density is $\rho = \frac{N}{\mathcal{A}}$.

The expected distance to the nearest neighbor for each sensor is as follows [36] [39] [40].

$$E_d = \frac{1}{2\sqrt{\rho}} \tag{2.4}$$

The expected number of neighbors for each sensor is as follows [36] [39] [40].

$$E_{nb} = \rho\pi {t_r}^2 \tag{2.5}$$

In addition, from [37] [38], we can obtain that the expected number of events that occur in time interval t at node i is as follows.

$$E_{ne} = \lambda \cdot p_{ei} \tag{2.6}$$

Let q be the number of packets transmitted from a node to each neighbor node if an event is detected, then the expected total number of transmitted packets for a node i is as follows.

$$Q = E_{ne} \cdot q \cdot E_{nb} = \lambda \cdot p_{ei} \cdot q \cdot E_{nb} \tag{2.7}$$

2.3.3 Network Lifetime

There are different definitions of network lifetime [41] [42], since when the network is considered non-functional depends on the specific application. It could be calculated as the instant when the first sensor

node dies, or when a percentage of sensor nodes have exhausted their energy, or when area coverage is no longer available.

In this chapter, we assume that the initial energy reserve of each sensor is E_o. With the time interval t, the network lifetime NL can be defined as follows, in which x is unknown.

$$NL = xt \tag{2.8}$$

In the following sections, we first analyze the packet transmissions during each time epoch TP of AO and CLSS1 as well as CLSS2 schemes. Further, we deduce the average energy consumption of a sensor node in each time epoch TP and then determine the corresponding network lifetime.

Network Lifetime of AO

For AO scheme, each sensor node is always awake in each time epoch TP, no matter the mobile user's location is in the location list L or not. Then the average energy consumption of a sensor node in each time epoch TP for AO is:

$$E_0 = \frac{(E_T + E_R)Q + (E_T + E_R)Q}{2} = E_T \cdot Q + E_R \cdot Q \tag{2.9}$$

The network lifetime of AO is:

$$NL_0 = \frac{E_o}{E_0} \cdot t = \frac{E_o}{E_T \cdot Q + E_R \cdot Q} \cdot t \tag{2.10}$$

Network Lifetime of CLSS1

For CLSS1 scheme, during each time epoch TP, base station s broadcasts the flags to sensor nodes first. The energy consumption of the sensor node for this part is generally $E_T E_{nb} + E_R$. If the sensor should be awake after receiving flag A, the energy consumption of a sensor node being awake for transmitting data packets is $(E_T + E_R)Q$. If the sensor can be asleep after receiving flag Z, then there is no energy consumption.

Hence, the average energy consumption for a sensor node during each time epoch TP for CLSS1 is

$$\begin{aligned} E_1 &= E_T E_{nb} + E_R + \frac{(E_T + E_R)Q}{2} \\ &= E_T \cdot (E_{nb} + \frac{Q}{2}) + E_R \cdot (1 + \frac{Q}{2}) \end{aligned} \tag{2.11}$$

In addition, the network lifetime of CLSS1 is

$$\begin{aligned} NL_1 &= \frac{E_o}{E_1} \cdot t \\ &= \frac{E_o}{E_T \cdot (E_{nb} + \frac{Q}{2}) + E_R \cdot (1 + \frac{Q}{2})} \cdot t \end{aligned} \tag{2.12}$$

Network Lifetime of CLSS2

Regarding CLSS2 scheme, base station s also first broadcasts the flags to sensor nodes during each time epoch TP. The energy consumption of the sensor node for this part is generally $E_T E_{nb} + E_R$ as well. The energy consumption of a sensor node if it remains awake to transmit data packets is also $(E_T + E_R)Q$ if the sensor receives flag A. When the sensor receives flag Z, the sensor is dynamically asleep and awake. Specifically, each sensor broadcasts $Erank_i$ and receives the ranks of its currently awake neighbors. The

energy consumption for broadcasting and receiving packets in this part is $(E_T + E_R)E_{nb}$. Also each sensor broadcasts R_i and receives R_j from each $j \in N_i$. The energy consumption for this part is $(E_T + E_R)E_{nb}$ as well. If the sensor is determined to be awake after satisfying the corresponding condition, let the energy consumption of such an awake node during data transmission be $(E_T + E_R) \cdot Q^{'}$, where $Q^{'}$ is the expected total number of transmitted packets for an awake node in an EC-CKN based network. If the sensor determines to be asleep after meeting the corresponding requirement, then there is no energy consumption.

Thus the average energy consumption for a sensor node during each time epoch TP for CLSS2 is

$$\begin{aligned} E_2 =& E_T E_{nb} + E_R + \frac{(E_T + E_R)Q}{2} \\ &+ \frac{(E_T + E_R)E_{nb} + (E_T + E_R)E_{nb} + \frac{(E_T+E_R)Q^{'}}{2}}{2} \\ =& E_T(2E_{nb} + \frac{Q}{2} + \frac{Q^{'}}{4}) + E_R(1 + E_{nb} + \frac{Q}{2} + \frac{Q^{'}}{4}) \end{aligned} \tag{2.13}$$

Then, the network lifetime of CLSS2 is

$$\begin{aligned} NL_2 &= \frac{E_o}{E_2} \cdot t \\ &= \frac{E_o}{E_T(2E_{nb} + \frac{Q}{2} + \frac{Q^{'}}{4}) + E_R(1 + E_{nb} + \frac{Q}{2} + \frac{Q^{'}}{4})} \cdot t \end{aligned} \tag{2.14}$$

In the following, we analyze $Q^{'}$. Let E_{anb} be the expected number of awake neighbors for each sensor after EC-CKN sleep scheduling and $f = \frac{E_{anb}}{E_{nb}} < 1$, we can further deduce that

$$Q^{'} = fQ \tag{2.15}$$

The analysis of E_{anb} is as shown in [36].

$$E_{anb} \leq \frac{cklnN}{\xi} \tag{2.16}$$

where c is a constant > 96, k is the parameter in the EC-CKN, and $\xi \geq 4(k + lnN)$.

2.3.4 Network Work Rate

The network work rate NWR is defined as the number of time epochs during which the network needs to work (i.e., always on or sleep scheduled), divided by its network lifetime. In this chapter, we assume that during the network lifetime NL, the number of time epochs when the mobile user's location is in the mobile user location list L is $|L|$ and the number of time epochs when the mobile user's location is not in the mobile user location list L is $NL - |L|$ ($|L| < NL$).

Network Work Rate of AO

As the network is always on in AO no matter the mobile user's location is in L or not during its network lifetime, thus the network work rate of AO (i.e., NWR_0) is as follows.

$$NWR_0 = \frac{|L| + NL_0 - |L|}{NL_0} = 100\% \tag{2.17}$$

Network Work Rate of CLSS1

Since the network lifetime of CLSS1 is NL_1 and the network resulting from CLSS1 only needs to be always on when the location of the mobile user is in L, the network work rate of CLSS1 (i.e., NWR_1) is as follows.

$$NWR_1 = \frac{|L|}{NL_1} \tag{2.18}$$

Network Work Rate of CLSS2

As the network lifetime of CLSS2 is NL_2 and the CLSS2 based network has to be always on if the mobile user's location is in L or sleep scheduled otherwise, the network work rate of CLSS2 (i.e., NWR_2) is as follows.

$$NWR_2 = \frac{|L| + NL_2 - |L|}{NL_2} = 100\% \tag{2.19}$$

2.3.5 Comparison of CLSS1 and CLSS2

The comparison of CLSS1 network lifetime (i.e., NL_1) and CLSS2 network lifetime (i.e., NL_2) as well as the contrast of CLSS1 network work rate (i.e., NWR_1) and CLSS2 network work rate (i.e., NWR_2) are shown as follows.

$$
\begin{aligned}
\frac{NL_1}{NL_2} &= \frac{E_o}{E_T \cdot (E_{nb} + \frac{Q}{2}) + E_R \cdot (1 + \frac{Q}{2})} \cdot t \div \\
&\frac{E_o}{E_T(2E_{nb} + \frac{Q}{2} + \frac{Q^{'}}{4}) + E_R(1 + E_{nb} + \frac{Q}{2} + \frac{Q^{'}}{4})} \cdot t \\
&= \frac{E_T(2E_{nb} + \frac{Q}{2} + \frac{fQ}{4}) + E_R(1 + E_{nb} + \frac{Q}{2} + \frac{fQ}{4})}{E_T \cdot (E_{nb} + \frac{Q}{2}) + E_R \cdot (1 + \frac{Q}{2})} \\
&= \frac{E_T(\frac{2E_{nb}}{Q} + \frac{Q}{2Q} + \frac{fQ}{4Q}) + E_R(\frac{1}{Q} + \frac{E_{nb}}{Q} + \frac{Q}{2Q} + \frac{fQ}{4Q})}{E_T \cdot (\frac{E_{nb}}{Q} + \frac{Q}{2Q}) + E_R \cdot (\frac{1}{Q} + \frac{Q}{2Q})} \\
&= \frac{E_T(\frac{1}{2} + \frac{f}{4}) + E_R(\frac{1}{2} + \frac{f}{4})}{E_T \cdot \frac{1}{2} + E_R \cdot \frac{1}{2}} \quad (Q \to \infty) \\
&= \frac{E_T(2 + f) + E_R(2 + f)}{E_T \cdot 2 + E_R \cdot 2} \quad (Q \to \infty)
\end{aligned}
\tag{2.20}
$$

Referring to equation (2.20) about $\frac{NL_1}{NL_2}$, we can further obtain

$$
\begin{aligned}
\frac{NWR_1}{NWR_2} &= \frac{|L|}{NL_1} \div \frac{|L| + NL_2 - |L|}{NL_2} \\
&= \frac{|L|}{|L| + NL_2 - |L|} \cdot \frac{NL_2}{NL_1} \\
&= \frac{|L|}{NL_2} \cdot \frac{E_T \cdot 2 + E_R \cdot 2}{E_T \cdot (2 + f) + E_R \cdot (2 + f)} \quad (Q \to \infty) \\
&= \frac{|L|}{NL_2} \cdot \frac{E_T + E_R}{E_T \cdot (1 + \frac{f}{2}) + E_R \cdot (1 + \frac{f}{2})} \quad (Q \to \infty)
\end{aligned}
\tag{2.21}
$$

2.3.6 Theorems

Theorem 2.3.1. *In terms of CLSS2, a sensor node i will always have at least min(k, o_i) awake neighbors after sleep scheduling, if it has o_i neighbors in the original sensor network.*

Proof: For CLSS2, if the mobile user's location $l_u \in L$, flag A will be sent from cloud c to base station

s. Then every sensor node i will receive flag A and be awake. In this case, i will have o_i awake neighbors after sleep scheduling.

In the case that a node i receives flag Z from the base station s, we perform the following analysis [43] [44] [45]. If $o_i < k$, all of i's neighbors should be awake (Step 9 of CLSS2) and i will have o_i awake neighbors. Otherwise if $o_i \geq k$, we prove the theorem by contradiction. First, we assume that a node i will not have at least k awake neighbors after running CLSS2, i.e., we can suppose that the m'th ($m \leq k$) lowest ranked neighbor (e.g., v) of i, decides to sleep. Then C_i will own at most $m-1$ nodes which are neighbors of i. But since $m-1 < k$, v cannot go to sleep according to Step 11 of CLSS2. This is a contradiction. In other words, the k lowest ranked neighbors of i will all be awake after running CLSS2. Thus, i will always have at least k awake neighbors.

Theorem 2.3.2. *In terms of CLSS2, running the scheme creates a connected network if the original sensor network is connected.*

Proof: Regarding CLSS2, if the mobile user's location $l_u \in L$, flag A will be sent to base station s. Further, every sensor node receives flag A which indicates that they should all be awake. Then the resultant network is connected.

Otherwise if base station s issues flag Z to sensor node i, we prove this theorem by contradiction [43] [44] [45]. First, we assume that the output network after running CLSS2 is not connected. Then the deleted nodes (asleep nodes determined by CLSS2) are put back in the network in descending order of their ranks. Let i be the first node making the sensor network connected again. Note that by the time we put i back, all the members of C_i are already present and nodes in C_i are already connected by nodes with $Erank > Erank_i$. Let v be a node that was disconnected from C_i but now becomes connected to C_i by i. However, this contradicts the fact that i can sleep only if all its neighbors (including v) are

connected to $\geq k$ nodes in C_i (Step 11 of CLSS2).

Theorem 2.3.3. *With CLSS schemes, the WSN lifetime of the MCC-WSN integration will be prolonged while the data requests of mobile users will still be satisfied.*

Proof: About CLSS1, sensor nodes will go to sleep if they do not receive flag A from the base station s during the time epoch (Step 5 of CLSS1). This actually indicates that a lot of energy consumption will be saved, since usually sensor nodes will transmit a lot of data packets to the base station if they are not in the asleep status. Even though there is extra flag broadcast energy consumption, these energy consumption are negligible, compared with the data transmission energy consumption when sensor nodes are awake. Thus, the WSN lifetime of MCC-WSN integration will be prolonged with CLSS1.

This also can be easily observed, by comparing equation (2.11) and equation (2.9) shown as follows, since $E_{nb} << Q$.

$$E_T \cdot (E_{nb} + \frac{Q}{2}) + E_R \cdot (1 + \frac{Q}{2}) < E_T \cdot Q + E_R \cdot Q \tag{2.22}$$

$$\frac{E_o}{E_T \cdot (E_{nb} + \frac{Q}{2}) + E_R \cdot (1 + \frac{Q}{2})} \cdot t > \frac{E_o}{E_T \cdot Q + E_R \cdot Q} \cdot t \tag{2.23}$$

Then,

$$NL_1 > NL_0 \tag{2.24}$$

Concerning CLSS2, if sensor nodes do not receive flag A from base station s during the time epoch,

they will be sleep scheduled according to the EC-CKN scheme (Step 5 of CLSS2). In this case, similar with CLSS1, quite an amount of data transmission energy consumption will also be saved, as only a subset of sensor nodes will be awake under some certain conditions (Step 11 of CLSS2). Hence, the WSN lifetime of MCC-WSN integration will also be extended with CLSS2.

This could be obtained from equation (2.13) and equation (2.9), since $E_{nb} << Q$ and $Q' < Q$.

$$E_T(2E_{nb} + \frac{Q}{2} + \frac{Q'}{4}) + E_R(1 + E_{nb} + \frac{Q}{2} + \frac{Q'}{4}) < E_T \cdot Q + E_R \cdot Q \tag{2.25}$$

$$\frac{E_o}{E_T(2E_{nb} + \frac{Q}{2} + \frac{Q'}{4}) + E_R(1 + E_{nb} + \frac{Q}{2} + \frac{Q'}{4})} \cdot t > \frac{E_o}{E_T \cdot Q + E_R \cdot Q} \cdot t \tag{2.26}$$

With that,

$$NL_2 > NL_0 \tag{2.27}$$

Regarding the data requests of mobile users for MCC-WSN integration with CLSS1, since the mobile applications are location specific and all sensor nodes will be awake if the mobile user's location $l_u \in L$, all sensory data required by the mobile users will be obtained.

In terms of utilizing CLSS2 for WSN to integrate with MCC, from Theorem 2.3.2, we can achieve that a connected network will always be produced with CLSS2 if the original sensor network is connected. In addition, this connected network is regardless of the mobile user's location l_u. Furthermore, from Theorem 2.3.1, we can obtain that each sensor node i will always own at least min(k, o_i) awake neighbors

with CLSS2, if it has o_i neighbors in the original sensor network. This indicates that the k actually could be tuned to offer enough awake sensor nodes, which ensure sufficient area coverage or target coverage according to the requirement of specific WSN application in CLSS2. Hence, the sensory data required by the mobile users for MCC-WSN integration with CLSS2 can also be achieved.

Theorem 2.3.4. *With CLSS1 scheme, the WSN work rate of the MCC-WSN integration will be enhanced. With CLSS2 scheme, the WSN work rate of the MCC-WSN integration remains unchanged.*

Proof: This theorem can be easily analyzed as follows, based on equation (2.17) to equation (2.19).

$$\frac{|L|}{NL_1} < \frac{|L| + NL_0 - |L|}{NL_1} < \frac{|L| + NL_0 - |L|}{NL_0} = 100\% \tag{2.28}$$

$$\frac{|L| + NL_2 - |L|}{NL_2} = 100\% = \frac{|L| + NL_0 - |L|}{NL_0} = 100\% \tag{2.29}$$

Thus,

$$NWR_1 < NWR_0 \quad \& \quad NWR_2 = NWR_0 \tag{2.30}$$

Theorem 2.3.5. *The network lifetime of CLSS1 is larger than that of CLSS2, while the network work rate of CLSS1 is shorter than that of CLSS2.*

Proof: Referring to equation (2.20) and equation (2.21), with $|L| < NL_2$, we can easily achieve the following.

$$\frac{NL_1}{NL_2} > 1 \quad \& \quad \frac{NWR_1}{NWR_2} < 1 \tag{2.31}$$

Then,

$$NL_1 > NL_2 \quad \& \quad NWR_1 < NWR_2 \tag{2.32}$$

The theorem is obtained.

2.3.7 Summary

From the above analysis, we can obtain the following insights.

- 1) Both CLSS1 and CLSS2 have longer network lifetime than AO.
- 2) CLSS1 has lower network work rate than AO and CLSS2 has the same network work rate as AO.
- 3) CLSS1 has longer network lifetime and lower network work rate than CLSS2.
- 4) CLSS1 maximizes the energy saving, while CLSS2 is scalable and robust. Reason: When the location of the mobile user u is not in L, all sensor nodes go to sleep in CLSS1 while sensor nodes are dynamically awake and asleep according to EC-CKN in CLSS2. EC-CKN sleep scheduling based network can still gather data and the k in EC-CKN can be tuned to achieve satisfied area coverage or target coverage. This can ensure that the WSN still provides sufficient data to cloud c, even when the location of u is not in L.

2.4 Numerical Results

We compare the WSN lifetime of MCC-WSN integration with CLSS and without CLSS (i.e., AO) schemes, as the data requests of mobile users will be both satisfied with either CLSS schemes or AO scheme based on the previous analysis. Due to that the network lifetime of the non-rechargeable WSN is generally

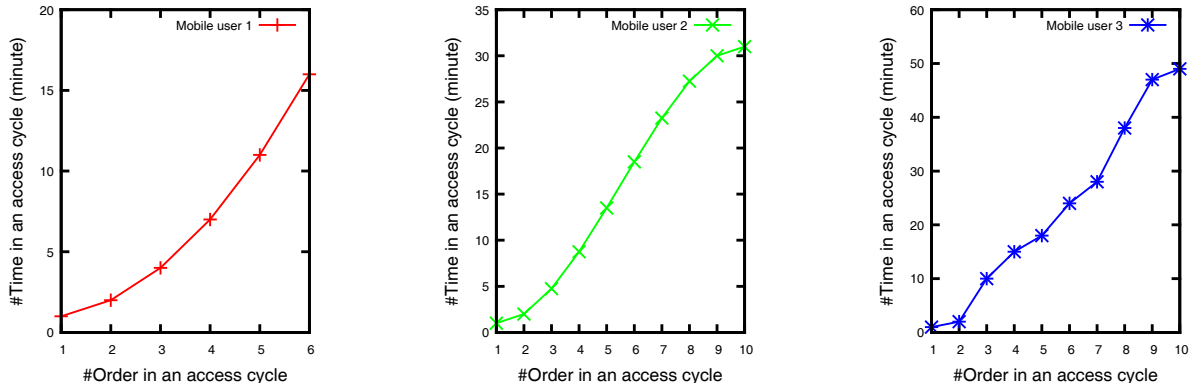
much shorter than the lifetime of the cloud, the WSN lifetime of the MCC-WSN integration could also be taken as the lifetime of MCC-WSN integration. Specifically, when the integrated WSN is not functional, the mobile user may no longer obtain the required sensory data from the cloud any more. Then the integration framework of MCC-WSN is considered not working any more. In addition, the WSN work rate of MCC-WSN integration with CLSS schemes and AO scheme are compared.

2.4.1 Evaluation Setup

In this evaluation, inspired by the behaviors that mobile users utilize the mobile application, we focus on the following three representative mobile users utilizing the StarTrack service, capturing and relaying the mobile user's location information to the StarTrack server in the cloud.

- 1) For mobile user 1, we assume that the mobile user accesses the mobile application with a cycle. For each cycle, the mobile application is accessed from every 1 minute to every 5 minutes and there are total 6 accesses as shown in Fig. 2.4(a).
- 2) Regarding mobile user 2, we suppose that the mobile user utilizes the mobile application also with a cycle. In this cycle, mobile user accesses the mobile application according to a parabola with maximum value which is 5 (i.e., every 5 minutes) and minimum value which is 1 (i.e., every 1 minute). There are total 10 accesses for this cycle as shown in Fig. 2.4(b).
- 3) With respect to mobile user 3, the mobile user utilizes the mobile application with a random interval between 1 minute and 10 minutes as shown in Fig. 2.4(c).

The track of each mobile user is observed and recorded to obtain a database, containing the current locations of mobile users as well as the location lists of mobile users. This database is to be further



(a) (b) (c)

Figure 2.4: Access behavior of mobile user 1 (a), mobile user 2 (b) and mobile user 3 (c)

Table 2.3: Evaluation parameters in Chapter 2

Parameter	Parameter value
Network size	600×600 m^2
Number of sensor nodes	100-1000
k in EC-CKN	1
Average event rate	50 times/minute
Initial energy	100000 mJ
Transmission energy	0.0144 mJ
Reception energy	0.00576 mJ
Transmission amplifier energy	0.0288 nJ/m^2
Transmission radius	60 m
Packet length	12 bytes
Number of packets	1000
Time epoch interval	1 minute

utilized by the WSN simulator NetTopo [3] [46] to conduct the simulation of each integrated WSN for each mobile user. The WSN size is 600×600 m^2. The number of randomly uniformly deployed sensor nodes ranges from 100 to 1000 (every time increased by 100). The value of k in EC-CKN is 1, which is the minimum value of k in EC-CKN. For every number of deployed sensor nodes, 100 different network topologies are generated with 100 different seeds and the average event rate is 50 times/minute. The initial energy of each node is 100000 mJ. The energy consumed by a sensor to transmit and receive one byte, and power-amplify each transmitted byte to cover the distance of 1 m are 0.0144 mJ, 0.00576 mJ and 0.0288 nJ/m^2 respectively [40] [44]. The transmission radius of each node is 60 m. The length of each packet is 12 bytes and each node transmits 1000 packets to each neighbor node in each time epoch which lasts 1 minute. Table 2.3 presents the evaluation parameters for each mobile user.

2.4.2 Evaluation Results

Fig. 2.5 depicts the theoretical analysis of network lifetime and network work rate of CLSS1 and CLSS2 as well as AO schemes for mobile user 1, mobile user 2 and mobile user 3. The simulation analysis of network lifetime and network work rate of CLSS1 and CLSS2 as well as AO schemes for mobile user 1, mobile user 2 and mobile user 3 are shown in Fig. 2.6. From Fig. 2.5 and Fig. 2.6, we can first observe that the theoretical results approximate the simulation results very much, which proves that our model is ponderable for analyzing the sleep scheduling mechanisms for WSNs to integrate with MCC.

Specifically, Fig. 2.5(a) to Fig. 2.5(c) and Fig. 2.6(a) to Fig. 2.6(c) graph the network lifetime of CLSS1 and CLSS2 as well as AO schemes for different mobile users, in theory and in simulation respectively. From these six graphs, we can obviously observe that the WSN lifetime of MCC-WSN integration

[3] NetTopo (online at http://sourceforge.net/projects/nettopo/) is an open source software on SourceForge for simulating and visualizing WSNs.

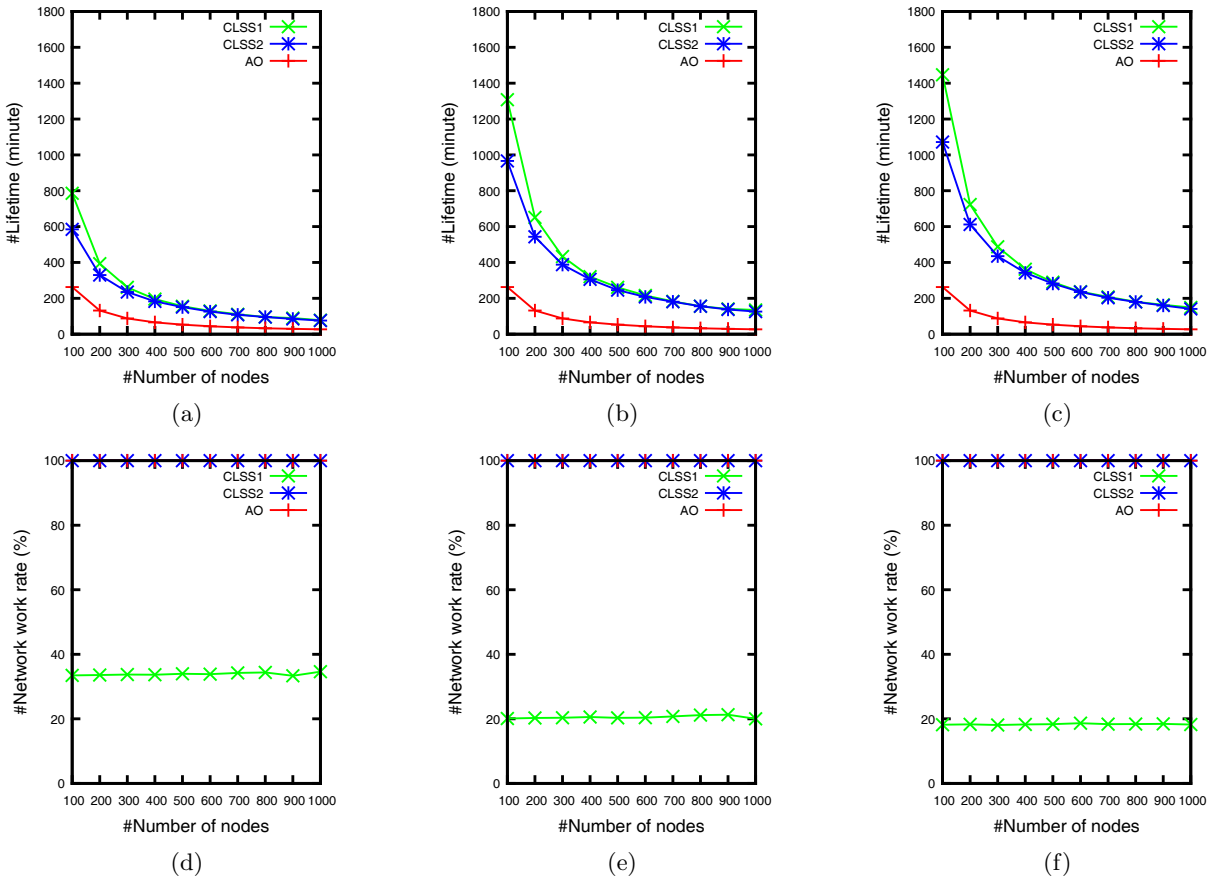


Figure 2.5: Theoretical analysis: network lifetime of CLSS1, CLSS2 and AO schemes for mobile user 1 (a), mobile user 2 (b) and mobile user 3 (c); network work rate of CLSS1, CLSS2 and AO schemes for mobile user 1 (d), mobile user 2 (e) and mobile user 3 (f). Both CLSS1 and CLSS2 have longer network lifetime than AO. CLSS1 has lower network work rate than AO and CLSS2 has the same network work rate as AO. CLSS1 has longer network lifetime and lower network work rate than CLSS2.

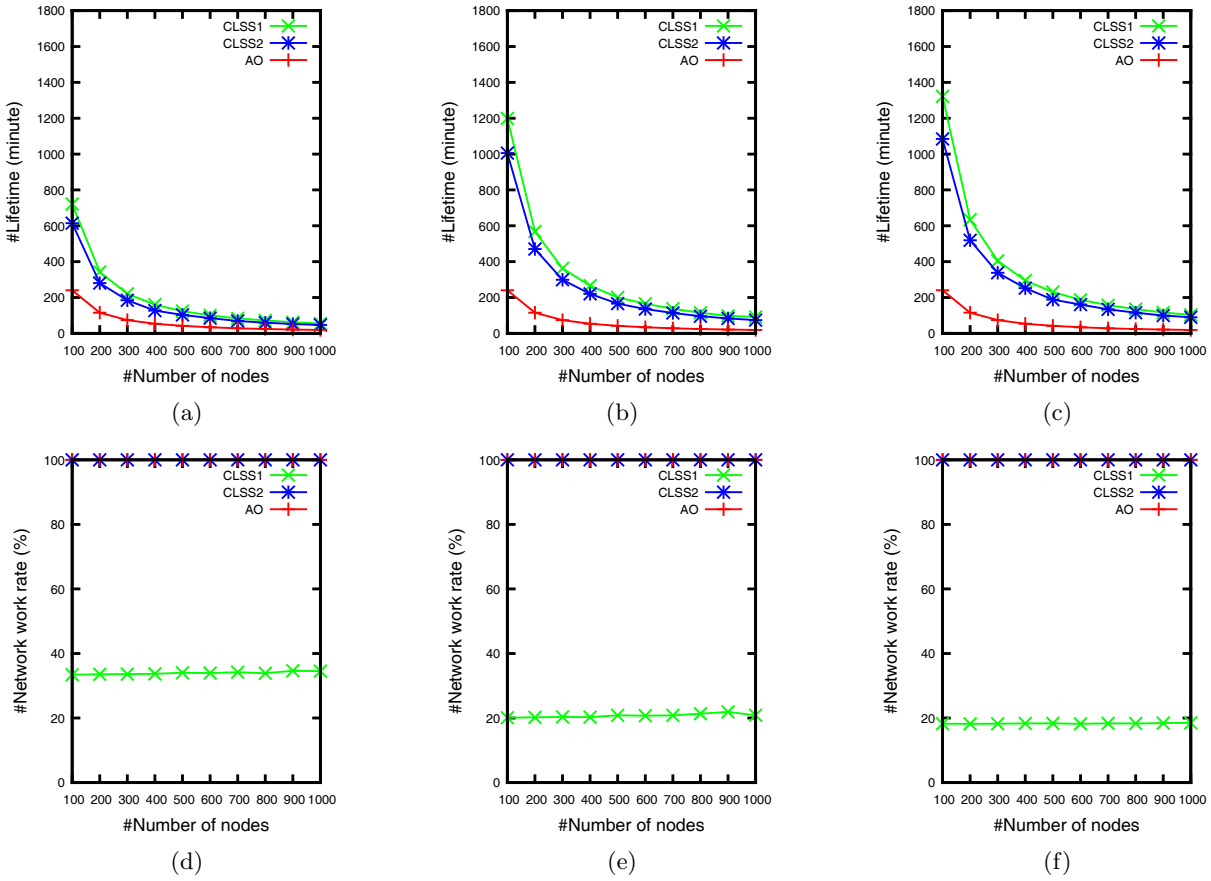


Figure 2.6: Simulation analysis: network lifetime of CLSS1, CLSS2 and AO schemes for mobile user 1 (a), mobile user 2 (b) and mobile user 3 (c); network work rate of CLSS1, CLSS2 and AO schemes for mobile user 1 (d), mobile user 2 (e) and mobile user 3 (f). Both CLSS1 and CLSS2 own longer network lifetime than AO. CLSS1 achieves lower network work rate than AO and CLSS2 achieves the same network work rate as AO. CLSS1 owns larger network lifetime and lower network work rate than CLSS2.

are greatly enhanced with either CLSS1 or CLSS2, compared with that with AO scheme. Moreover, the network work rate of CLSS1 and CLSS2 as well as AO schemes in theory and in simulation for various mobile users are depicted from Fig. 2.5(d) to Fig. 2.5(f) and Fig. 2.6(d) to Fig. 2.6(f), respectively. These six pictures demonstrate that the WSN work rate of MCC-WSN integration with CLSS1 scheme is much lower than that with AO scheme, while the WSN work rate of MCC-WSN integration with CLSS2 scheme is the same as that with AO scheme.

In addition, with respect to CLSS1 and CLSS2, compared with CLSS2, the network lifetime with CLSS1 presented in Fig. 2.5(a) to Fig. 2.5(c) and Fig. 2.6(a) to Fig. 2.6(c) is longer. The network work rate in CLSS1 is lower than that in CLSS2, which are shown in Fig. 2.5(d) to Fig. 2.5(f) and Fig. 2.6(d) to Fig. 2.6(f). However, CLSS2 is more scalable and robust to unexpected mobile user data requests compared with CLSS1, since a subset of sensor nodes in CLSS2 still form a connected network and transmit the collected sensory data to the cloud, even when the location of the mobile user is not in the location list.

2.4.3 Summary

From the above evaluation results in both theory and simulation, we can achieve the following.

- 1) In terms of CLSS schemes and AO scheme, all the theoretical results approximate all the simulation results very much.
- 2) Comparing CLSS schemes and AO scheme, both CLSS1 and CLSS2 have longer network lifetime than AO. In addition, CLSS1 has lower network work rate than AO and CLSS2 has the same network work rate as AO.

- 3) Comparing CLSS1 and CLSS2, CLSS1 owns larger network lifetime and lower network work rate than CLSS2.

Chapter 3

An Authenticated Trust and Reputation Calculation and Management System for Cloud and Sensor Networks Integration

3.1 System Model

The system model in this chapter is shown as follows, while the main used notations in this chapter are summarized in Table 3.1 and Table 3.2.

- There are multiple CSUs, CSPs and SNPs. The number of CSU, CSP and SNP are N_u, N_c and N_k, respectively. $CSU_{Set} = \{CSU_1, CSU_2, \ldots, CSU_{N_u}\}$. $CSP_{Set} = \{CSP_1, CSP_2, \ldots, CSP_{N_c}\}$. $SNP_{Set} = \{SNP_1, SNP_2, \ldots, SNP_{N_k}\}$.

- Each CSU, CSP and SNP have several attributes. Particularly, the data service requested and required by the CSU owns the following attributes: data service pay (DSP); data type (DT); data size (DS); data request speed (DRS); data service time (DST). The cloud service provided and managed by each CSP has the following characteristics: cloud service charge (CSC); cloud operation

Table 3.1: Main notation definitions in Chapter 3

Symbol	Definition
COC	Cloud Operation Cost
CPS	Cloud Processing Speed
CRT	Cloud Response Time
CSC	Cloud Service Charge
CSN	Cloud Server Number
CSP	Cloud Service Provider
CSP_{Set}	Set of CSPs
CSS	Cloud Storage Size
CST	Cloud Service Type
CSU	Cloud Service User
CSU_{Set}	Set of CSUs
DRS	Data Request Speed
DS	Data Size
DSP	Data Service Pay
DST	Data Service Time
DT	Data Type
PLA	Privacy Level Agreement
SLA	Service Level Agreement
SNC	Sensor Network Coverage
SNL	Sensor Network Lifetime
SNN	Sensor Node Number
SNOC	Sensor Network Operation Cost
SNP	Sensor Network Provider
SNP_{Set}	Set of SNPs
SNRT	Sensor Network Response Time
SNSC	Sensor Network Service Charge
SNSP	Sensor Network Service Pay
SNT	Sensor Network Throughput
ST	Sensor Type
TCE	Trusted Center Entity

Table 3.2: Main notation definitions continued in Chapter 3

Symbol	Definition
C_{bc}	Acceptable range for C_c
C_{bk}	Acceptable range for C_k
$\|C_{bc}\|$	Interval of C_{bc}
$\|C_{bk}\|$	Interval of C_{bk}
C_c	CSC-DSP
C_k	SNSC-SNSP
ct_c	Certificate of CSP
ct_k	Certificate of SNP
R_c	Reputation value of service provided by CSP
R_k	Reputation value of service provided by SNP
R_{sc}	Minimum acceptable reputation value for service of CSP
R_{sk}	Minimum acceptable reputation value for service of SNP
T_{cu}	Trust value of service from CSP to CSU
T_{kc}	Trust value of service from SNP to CSP
T_{scu}	Minimum acceptable trust value of service from CSP to CSU
T_{skc}	Minimum acceptable trust value of service from SNP to CSP
α_c	Weight with respect to the importance of C_c
α_k	Weight with respect to the importance of C_k
β_c	Weight with respect to the importance of T_{cu}
β_k	Weight with respect to the importance of T_{kc}
γ_c	Weight with respect to the importance of R_c
γ_k	Weight with respect to the importance of R_k

cost (COC); sensor network service pay (SNSP); cloud service type (CST); cloud server number (CSN); cloud storage size (CSS); cloud processing speed (CPS); cloud operation time (COT); cloud response time (CRT). The sensor network offered and managed by each SNP is with the following properties: sensor network service charge (SNSC); sensor network operation cost (SNOC); sensor type (ST); sensor node number (SNN); sensor network coverage (SNC); sensor network throughput (SNT); sensor network lifetime (SNL); sensor network response time (SNRT).

- There is a trust value (i.e., T_{cu}) of each service from each CSP to each CSU and there is a trust value (i.e., T_{kc}) of each service from each SNP to each CSP. In addition, there is a reputation value (i.e., R_c) of each service provided by each CSP and there is a reputation value (i.e., R_k) of each service provided by each SNP.

- Each CSU owns a minimum acceptable trust value (i.e., T_{scu}) of each service from each CSP to the CSU. Moreover, each CSP has a minimum acceptable trust value (i.e., T_{skc}) of each service from each SNP to the CSP. Similarly, each CSU owns a minimum acceptable reputation value (i.e., R_{sc}) with respect to each service of each CSP. And each CSP has a minimum acceptable reputation value (i.e., R_{sk}) in terms of each service of each SNP.

- There is a cost difference (i.e., C_c) between the CSC of CSP and DSP of CSU for each service, i.e., C_c=CSC-DSP.

- There is a cost difference (i.e., C_k) between the SNSC of SNP and SNSP of CSP for each service, i.e., C_k=SNSC-SNSP.

- Each CSU owns an acceptable range (i.e., C_{bc}) about C_c. In addition, each CSP owns an acceptable range (i.e., C_{bk}) about C_k. The interval of C_{bc} and C_{bk} are $|C_{bc}|$ and $|C_{bk}|$, respectively.

- Each CSU has three weights (i.e., α_c, β_c and γ_c) in terms of the importance of C_c, T_{cu} and R_c, while $\alpha_c + \beta_c + \gamma_c = 1$. Similarly, each CSP owns three weights (i.e., α_k, β_k, γ_k) about the importance of C_k, T_{kc} and R_k, while $\alpha_k + \beta_k + \gamma_k = 1$.

3.2 Authentication of CSP and SNP as well as Trust and Reputation of Service of CSP and SNP

In this section, we first discuss the authentication of CSP and SNP. With that, we give some preliminaries about service level agreement (SLA) and privacy level agreement (PLA), followed with the preliminaries of trust and reputation and the preliminaries of trusted center entity (TCE). Finally, we discuss and analyze the trust and reputation with respect to the service of CSP and SNP respectively.

3.2.1 Authentication of CSP and SNP

In this chapter, as the key of our work is to enable CSU to choose the authentic and desirable CSP as well as assist CSP in selecting genuine and appropriate SNP, we focus on the authentication of CSP and SNP rather than the authentication of CSU. Specifically, the CSP needs to prove its authenticity to CSU and SNP has to show its authenticity to CSP. Here, ISO/IEC 27001 certification [47] [48] is applied to authenticate CSP and SNP, as it is an internationally recognized information security management system (ISMS) standard by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). It requires that the information management of an organization (e.g., CSP or SNP) meets (i) the organization's information security risks are systematically examined; (ii) a coherent and comprehensive suite of information security controls is designed and implemented to solve

those risks that are deemed unacceptable; (iii) an overarching management process is adopted to ensure that the information security controls continue to satisfy the organization's information security needs on an ongoing basis. Particularly, it provides confidence and assurance to trading clients of the organization, as the security status of the organization is audited to be qualified, by issuing a certificate with the ISO/IEC 27001 certification. After CSP and SNP are certificated with ISO/IEC 27001, they obtain the certificates (i.e., ct_c and ct_k) respectively.

3.2.2 Preliminaries of SLA and PLA

An SLA [49] [50] is a negotiated agreement between two or more parties, in which one is the customer and the others are service providers. In short, it is a part of a service contract, in which a service is formally defined. SLA specifies the levels of availability, serviceability, performance, operation and other attributes of the service. Usually, an SLA addresses the following segments about a service: definition, performance measurement, problem management, duties, warranties, termination. The subject of SLA is the result of the service received by the customer.

An PLA [51] is an agreement to describe the level of privacy protection that the CSP will maintain. Thus it is an appendix to the SLA between CSU and CSP. The SLA between CSU and CSP provides specific parameters and minimum levels on other performance (e.g., cloud processing speed, cloud operation time) of the cloud service, while PLA addresses information privacy and personal data protection issues about the cloud service.

3.2.3 Preliminaries of Trust and Reputation

Defined by Merriam Webster's Dictionary, trust is "*assured reliance on the character, ability, strength or truth of someone or something*" and reputation is "*overall quality or character as seen or judged by people in general*". However, trust and reputation are multidisciplinary concepts with different definitions and evaluations in various fields (e.g., psychology, sociology, economics, philosophy, wireless networks) [52] [53] [54]. For example, in the scenario of wireless communications, "Trust of a node A in a node B is the subjective expectation of node A receiving positive outcomes from the interaction with node B in a specific context". Also, "Reputation is the global perception of a node's trustworthiness in a network".

Generally, to evaluate trust from an entity (e.g., A or *trustor*) to another entity (e.g., B or *trustee*), A needs to gather evidence (e.g., honest, selfish, malicious behaviors), representing the satisfaction, about B either through direct interaction or information provided by third-parties [52], [53] [54]. With that, trustor (A) maps the gathered information from the evidence space to the trust space through a predefined mapping function and an aggregation function to obtain the trustworthiness value of trustee (B). Specifically, the trustworthiness obtained by mapping evidences from direct interaction is known as *direct trust*, while the trustworthiness achieved through mapping evidences from third-parties is *indirect trust*. Furthermore, a trustor can bring into account *recent trust*, which reflects only the recent behaviors, as well as *historical trust*, which is built from the past experiences and it reflects long-term behavioral pattern. For instance, using indirect trust and historical trust helps trustor to protect trust evaluation (and trust system in general) from attacks such as good mouthing and bad mouthing, or sudden selfishness of a trustee. More discussion about these terms and definitions can be found in our references, for instance in [55]. In addition, to evaluate reputation about a trustee (e.g., B), the aggregated trust opinion of a group of entities are usually taken to represent the reputation value [56] [57].

A widely used way to map the observed information from the evidence space to the trust space is the beta distribution [58] [59] [60] illustrated as follows. Let s and f represent the (collective) amount of positive and negative feedbacks in the evidence space about target entity, then the trustworthiness t of a subject node is then computed as $t = \frac{s+1}{f+s+2}$.

3.2.4 Preliminaries of TCE

In this chapter, based on the five main roles (e.g., *cloud customer, cloud provider, cloud broker, cloud auditor and cloud carrier*) in CC [61], we assume that the role of the cloud auditor is assigned to TCE. Furthermore, we assume that TCE consists of multiple entities in various locations with a shared and secured database, e.g., in a data center. Specifically, the duties of TCE are introduced as follows.

- Duty 1) Receiving the copies of signed SLAs and PLAs from CSUs, CSPs and SNPs.
- Duty 2) Receiving the feedbacks from CSUs about the services of CSPs and receiving the feedbacks from CSPs about the services of SNPs, based on signed SLAs and PLAs.
- Duty 3) Auditing whether received copies are genuine as well as auditing whether received feedbacks that are to be utilized to calculate T_{cu}, T_{kc}, R_c and R_k are genuine, by security audit, privacy impact audit and performance audit, and etc. [62].
- Duty 4) Calculating and managing (i.e. storing and updating) T_{cu}, T_{kc}, R_c and R_k, with the genuine historical feedbacks received from CSUs about the services of CSPs and the genuine historical feedbacks from CSPs about the services of SNPs based on genuinely signed SLAs and PLAs.
- Duty 5) Replying T_{cu}, T_{kc}, R_c and R_k values if these values are requested by CSUs or CSPs.

- Duty 6) Auditing whether the T_{cu}, T_{kc}, R_c and R_k values received by CSUs and CSPs are genuine, by security auditing, privacy impact auditing and performance auditing, and etc. [62].

- Duty 7) Monitoring the process of the proposed ATRCM system to detect misbehaviors of CSUs, CSPs or SNPs that affect the process of ATRCM.

3.2.5 Trust of Service of CSP

From Fig. 1.1, we can obtain that the fulfillment of service of CSP needs to receive and store the raw sensory data from SNP first. Then CSP processes the raw sensory data and stores the processed sensory data. Finally, CSP transmits the processed sensory data to CSU on demand. In this process, there are various types of trust (e.g., cloud data storage trust, cloud data processing trust, cloud data privacy trust, cloud data transmission trust) which might concern the CSU to choose the service of CSP. Furthermore, for various CSUs, the types of trust that they concern are different.

In this chapter, we assume that the following three types of trust about CSP concern the CSU to choose the service of CSP and we further show how they are calculated.

- i) Cloud data processing trust: This trust is related to whether cloud processes the raw sensory data with error. TCE has a database which dynamically stores the non-error number (i.e., S_{c1}) and error number (i.e., F_{c1}) of data processing of each service from CSP to the CSU in the history, with the feedbacks about the historical SLAs regarding the service. The trust value of cloud data processing trust (i.e., T_{c1}) is calculated by TCE via equation (3.1).

$$T_{c1} = \frac{S_{c1} + 1}{F_{c1} + S_{c1} + 2} \tag{3.1}$$

- ii) Cloud data privacy trust: This trust is about whether the sensory data stored on cloud can be accessed by others. Based on the feedbacks about previous PLAs regarding the service, assume the number that the sensory data accessed by others with respect to each service from CSP to CSU in the history stored on TCE database is F_{c2}. As CSU is generally sensitive about the data privacy, the trust value of cloud data privacy trust (i.e., T_{c2}) is presented by TCE through equation (3.2).

$$T_{c2} = \begin{cases} 1, & F_{c2} = 0 \\ 0, & F_{c2} > 0 \end{cases} \tag{3.2}$$

- iii) Cloud data transmission trust: This trust is with respect to whether the data transmission from CSP to CSU is successful. Using the feedbacks of previous SLAs regarding the service, with the success number (i.e., S_{c3}) and failure number (i.e., F_{c3}) of data transmission of each service from CSP to the CSU in the history on TCE database, the cloud data transmission trust (i.e., T_{c3}) is shown by TCE as per equation (3.3).

$$T_{c3} = \frac{S_{c3} + 1}{F_{c3} + S_{c3} + 2} \tag{3.3}$$

In summary, with respect to T_{cu} value calculation, the trust value T_{cu} of each service from CSP to CSU is calculated by TCE with a combination function (i.e. CF) of three-dimensional trust (i.e., cloud data processing trust, cloud data privacy trust and cloud data transmission trust), as per equation (3.4).

$$T_{cu} = CF(T_{c1}, T_{c2}, T_{c3}) \tag{3.4}$$

Specifically, about CF, there are many different ways to combine multi-dimensional trust. For example, a probabilistic trust model based on the Dirichlet distribution to combine multi-dimensional trust is shown in [63], by estimating the probability that each contract dimension will be successfully fulfilled as well as the correlations between these estimates. In addition, an MeTrust model is presented in [64], enabling each user to choose a dimension as a primary dimension and put different weights on different dimensions for trust calculation.

In this chapter, we assume that these three types of trust (i.e., cloud data processing trust, cloud data privacy trust and cloud data transmission trust) are considered with equal weight and then the minimum trust value in these three trust values is taken as T_{cu}, through equation (3.5).

$$T_{cu} = Minimum\{T_{c1}, T_{c2}, T_{c3}\} \tag{3.5}$$

3.2.6 Reputation of Service of CSP

In this chapter, based on the feedbacks of previous SLAs about the service, we assume that if the CSU chose the service of the CSP, then it means that the CSU somehow trusted that CSP and decided to use the service of the CSP. Let us assume that the number of CSUs that chose the service of the CSP is CN_c and the number of CSUs that needed the service to receive from a CSP is $N_u^{'}$ ($N_u^{'} \leq N_u$). Then the reputation value (i.e., R_c) of the service of the CSU is calculated by TCE following [56] [57] via equation (3.6).

$$R_c = \frac{CN_c}{N_u^{'}} \tag{3.6}$$

3.2.7 Trust of Service of SNP

Based on Fig. 1.1, we can also observe that the service of SNP requires the sensor nodes to be deployed first and then sense, store and process data to achieve data collection. At last, the collected sensory data are transmitted from SNP to the CSP.

Similarly, in this chapter, we assume that the following four kinds of trust about SNP consist of the trust of service of SNP in the above process.

- i) Sensor data collection trust: This trust concerns whether the sensor network collects the required sensory data with error. Utilizing the feedbacks of previous SLAs regarding each service, given that the non-error number and error number of data collection of each service from SNP to CSP in the history on the TCE database are S_{k1} and F_{k1}, respectively. The trust value of sensor data collection trust (i.e., T_{k1}) is calculated by TCE as follows.

$$T_{k1} = \frac{S_{k1} + 1}{F_{k1} + S_{k1} + 2} \tag{3.7}$$

- ii) Sensor network lifetime trust: This trust aims to analyze whether the lifetime of the real deployed sensor network matches the sensor network lifetime the SNP demonstrates, as energy consumption is the primary concern of sensor network. Assume that the matching number and non-matching number of the sensor network lifetime of each service from SNP to CSP in the history recorded by TCE are S_{k2} and F_{k2} respectively, with the feedbacks of historial SLAs regarding each service. The sensor network lifetime trust (i.e., T_{k2}) is shown by TCE as follows.

$$T_{k2} = \frac{S_{k2} + 1}{F_{k2} + S_{k2} + 2} \tag{3.8}$$

- iii) Sensor network response time trust: This trust researches whether the response time of the real deployed sensor network matches the sensor network response time the SNP demonstrates, since the response time of sensor network is with quite uncertainty due to various factors (e.g., sensor dies, bad weather). TCE records the matching number (i.e., S_{k3}) and non-matching number (i.e., F_{k2}) of the sensor network response time of each service from SNP to CSP in the history with feedbacks about previous SLAs. The sensor network response time trust (i.e., T_{k3}) is obtained by TCE as follows.

$$T_{k3} = \frac{S_{k3} + 1}{F_{k3} + S_{k3} + 2} \tag{3.9}$$

- iv) Sensor data transmission trust: This trust cares whether the data transmission from SNP to CSP is successful or not. TCE owns a database which dynamically stores the success number (i.e., S_{k4}) and failure number (i.e., F_{k4}) of data transmission of each service from SNP to the CSP in the history, based on the feedbacks of previous SLAs regarding each service. The sensor data transmission trust value (i.e., T_{k4}) is presented by TCE as follows.

$$T_{k4} = \frac{S_{k4} + 1}{F_{k4} + S_{k4} + 2} \tag{3.10}$$

In summary, concerning T_{kc} value calculation, we also assume that these four types of trust (i.e., sensor data collection trust, sensor network lifetime trust, sensor network response time trust and sensor data transmission trust) are considered equally and the minimum value of these four trust values is taken as the trust value T_{kc} of the service from SNP to CSP, calculated by TCE as follows:

$$T_{kc} = Minimum\{T_{k1}, T_{k2}, T_{k3}, T_{k4}\} \tag{3.11}$$

3.2.8 Reputation of Service of SNP

About R_k value calculation, with the feedbacks of previous SLAs about the service, given that if the CSP chose the service of an SNP, then it also means that the CSP somehow trusted the SNP and decided to use the service of the SNP. Further, denote that the number of CSPs that chose the service of the SNP is CN_c and the number of CSPs that required the service to receive from a SNP is $N_c^{'}$ ($N_c^{'} \leq N_c$), the reputation value of the service of the SNP is calculated by TCE following [56] [57] as follows.

$$R_k = \frac{CN_k}{N_c^{'}} \tag{3.12}$$

3.3 Proposed ATRCM System

3.3.1 System Overview

The proposed authenticated trust and reputation calculation and management (ATRCM) system is introduced from the following three parts:

- Part 1) Authentication flowchart of CSP and SNP;
- Part 2) Trust and reputation calculation and management flowchart between CSU and CSPs;
- Part 3) Trust and reputation calculation and management flowchart between CSP and SNPs.

Specifically, Part 1) shown in Table 3.3 aims at a) identity authentication of CSP and SNP to avoid malicious impersonation attacks, based on the certificate of ISO/IEC 27001 certification [47] [48] illustrated in Section 3.2. In addition, Part 2) and Part 3) presented in Table 3.4 and Table 3.5 focus on

Table 3.3: Authentication flowchart of CSP and SNP

Step	CSUs	CSPs	SNPs
Start			
1		Provide ct_c to CSU	Provide ct_k to CSP
2	Check ct_c and filter CSPs	Check ct_k and filter SNPs	
End			

Table 3.4: Trust and reputation calculation and management flowchart between CSU and CSPs

Step	CSPs	CSU	TCE	CSU
Start				
1	Provide attributes	Checks CSPs attributes and filters CSPs		
2		Issues requests to TCE	Replies T_{cu}	Checks T_{cu} and filters CSPs
3		Issues requests to TCE	Replies R_c	Checks R_c and filters CSPs
4		Calculates C_c		
5		Checks ct_c and chooses the service of CSP and informs TCE		
6		Checks ct_c and sends feedbacks	Updates T_{cu} and R_c	
End				

Table 3.5: Trust and reputation calculation and management flowchart between CSP and SNPs

Step	SNPs	CSP	TCE	CSP
Start				
1	Provide attributes	Checks SNPs attributes and filters SNPs		
2		Issues requests to TCE	Replies T_{kc}	Checks T_{kc} and filters SNPs
3		Issues requests to TCE	Replies R_k	Checks R_k and filters SNPs
4		Calculates C_k		
5		Checks ct_k and chooses the service of SNP and informs TCE		
6		Checks ct_k and sends feedbacks	Updates T_{kc} and R_k	
End				

(b) calculation and management of trust and reputation with respect to the service of CSP and SNP as well as (c) helping the CSU choose desirable CSP and assisting the CSP in selecting appropriate SNP, considering the attribute requirement of CSU and CSP as well as cost, trust and reputation of the service of CSP and SNP.

3.3.2 Authentication Flowchart of CSP and SNP

- Step 1: CSPs provide the certificate ct_c to CSU and CSU checks whether the signature of the certificate is valid and whether the certificate is revoked. CSU filters the CSPs that are not qualified.
- Step 2: SNPs offer the certificate ct_k to CSP and CSP checks whether the signature of the certificate is valid and whether the certificate is revoked. CSP filters the SNPs that are not qualified.

3.3.3 Trust and Reputation Calculation and Management Flowchart between CSU and CSPs

- Step 1: CSU checks whether the characteristics of CSPs satisfy the attribute requirement of CSU. Filter the CSPs that are not satisfied.

$$\begin{cases} CST \supseteq DT \\ CSS \geq DS \\ CPS \geq DRS \\ COT \geq DST \end{cases} \tag{3.13}$$

- Step 2: CSU issues requests to TCE and achieves the T_{cu} value of the service from CSP to the CSU. CSU checks whether the T_{cu} value is greater than or equal to the T_{scu} value. Filter the CSPs that are not satisfied.

$$T_{cu} \geq T_{scu} \tag{3.14}$$

- Step 3: CSU issues requests to TCE and achieves the R_c value of the service offered by the CSP. CSU checks whether the R_c value is greater than or equal to the R_{sc} value. Filter the CSPs that are not satisfied.

$$R_c \geq R_{sc} \tag{3.15}$$

- Step 4: CSU calculates the C_c value between CSC of CSP and DSP of CSU and checks whether the C_c value is within the C_{bc} range. Filter the CSPs that are not satisfied.

$$C_c \in C_{bc} \tag{3.16}$$

- Step 5: CSU checks whether ct_c is revoked and chooses the service offered by the CSP with the maximum M_c and informs TCE about signed SLA or PLA.

$$M_c = -\alpha_c \cdot \frac{C_c}{|C_{bc}|} + \beta_c \cdot T_{cu} + \gamma_c \cdot R_c \tag{3.17}$$

- Step 6: CSU checks whether ct_c is revoked before using the service from the CSP. CSU sends feedbacks about the service of the CSP to TCE based on PLA and SLA after the termination of service. TCE stores and updates the T_{cu} value as well as the R_c value with the equations illustrated in Section 3.2.

3.3.4 Trust and Reputation Calculation and Management Flowchart between CSP and SNPs

- Step 1: CSP checks whether the characteristics of SNPs satisfy the attribute requirement of CSP. CSP also checks whether the characteristics of SNP satisfy the attribute requirement of CSU. Filter the SNPs that are not satisfied.

$$\begin{cases} ST \supseteq DT \\ SNC \supseteq DS \\ SNT \geq DRS \\ SNL \geq DST \end{cases} \tag{3.18}$$

$$\begin{cases} CST \supseteq ST \\ CSS \geq SNC \\ CPS \geq SNT \\ COT \geq SNL \end{cases} \tag{3.19}$$

- Step 2: CSP issues requests to TCE and receives the T_{kc} value of the service from SNP to the CSP. CSP checks whether the T_{kc} value is more than or equal to the T_{skc} value. Filter the SNPs that are not satisfied.

$$T_{kc} \geq T_{skc} \tag{3.20}$$

- Step 3: CSP issues requests to TCE and receives the R_k value of the service offered by the SNP. CSP checks whether the R_k value is more than or equal to the R_{sk} value. Filter the SNPs that are not satisfied.

$$R_k \geq R_{sk} \tag{3.21}$$

- Step 4: CSP calculates the C_k value between SNSC of SNP and SNSP of CSP and checks whether the C_k value is within the C_{bk} range. Filter the SNPs that are not satisfied.

$$C_k \in C_{bk} \tag{3.22}$$

- Step 5: CSP checks whether ct_k is revoked and chooses the service offered by the SNP with the maximum M_k and informs TCE about signed SLA or PLA.

$$M_k = -\alpha_k \cdot \frac{C_k}{|C_{bk}|} + \beta_k \cdot T_{kc} + \gamma_k \cdot R_k \tag{3.23}$$

- Step 6: CSP checks whether ct_k is revoked before utilizing the service of the SNP. After the end of service, CSP sends feedbacks about the service of SNP to TCE based on SLA and PLA. TCE stores and updates the T_{kc} value and the R_k value with the equations presented in Section 3.2.

Note: In the aforementioned steps, during the utilization of the service, the ct_c of the chosen CSP or the ct_k of the selected SNP may still be revoked. Furthermore, the T_{cu} or the R_c of the service of the chosen CSP may be lower than T_{scu} or R_{sc}, respectively. Similarly, the T_{kc} or the R_k of the service of the selected SNP is possible to be lower than T_{skc} or R_{sk} respectively. In such cases, the system flowcharts are performed again to enable the CSU to choose a new CSP or make the CSP select a new SNP. In addition, although the check of ct_c, T_{cu} and R_c is the duty of CSU as well as the check of ct_k, T_{kc} and R_k is the duty of CSP, TCE can support these duties for CSU and CSP as well if necessary.

3.4 Evaluation of System Functionality

In this section, we evaluate whether our proposed ATRCM system can fulfill the predetermined functions: a) authenticating CSP and SNP to avoid malicious impersonation attacks; b) calculating and managing trust and reputation regarding the service of CSP and SNP; c) helping CSU choose desirable CSP and assisting CSP in selecting appropriate SNP, based on (i) the authenticity of CSP and SNP; (ii) the attribute requirement of CSU and CSP as well as (iii) the cost, trust and reputation of the service of CSP and SNP.

3.4.1 Evaluation Setup

To perform the evaluation, all the three aimed functions are analyzed based on the flowcharts and processes of the corresponding functions. Particularly, the third function is evaluated utilizing two representative case studies to demonstrate the effectiveness of ATRCM. Case study 1 involves a small quantities of CSUs, CSPs and SNPs, while case study 2 involves a large number of CSUs, CSPs and SNPs. The evaluation processes of the third function shown in these two case studies are universal for CSUs, CSPs and SNPs with other attributes and parameters.

3.4.2 Evaluation Results

Authenticating CSP and SNP

With respect to the authentication of CSP and SNP, Part 1) authentication flowchart of CSP and SNP shown in Section 3.3 presents the detailed steps.

Based on the flowchart, we can observe that if a malicious attacker impersonates the authentic CSP

or authentic SNP, then it needs to own the ct_c certificate or the ct_k certificate first. If it cannot provide a certificate, then it is not a genuine organization. In addition, even if the malicious attacker further a) offers a fake certificate (e.g., fct_c or fct_k) or b) provides a real but revoked certificate (e.g., rct_c or rct_k), it still cannot launch the impersonation attacks, since CSU and CSP check whether the signature of the certificate is valid and whether the certificate is revoked.

Thus, we can achieve that our proposed ATRCM system is able to prevent malicious impersonation attacks, by enforcing the CSP or SNP providing a valid certificate. Meanwhile, as the valid certificate of CSP and SNP are obtained through ISO/IEC 27001 certification, the CSU will start trading with CSP and CSP will begin trading with SNP, with more confidence and assurance.

Calculating and Managing Trust and Reputation of Service of CSP and SNP

For the calculation and management of trust and reputation with respect to the service of the CSP and SNP, the detailed processes are illustrated in Section 3.2.

Particularly, calculation and management of trust regarding the service of the CSP are based on cloud data processing trust (i.e., T_{c1} shown in equation (3.1)), cloud data privacy trust (i.e., T_{c2} shown in equation (3.2)) and cloud data transmission trust (i.e., T_{c3} shown in equation (3.3)). The minimum value of T_{c1}, T_{c2} and T_{c3} is the trust value of the service of the CSP. Moreover, the history that CSUs chose the service of the CSP and the history that CSUs needed the service to receive from a CSP are utilized to calculate and manage the reputation about the service of the CSP (i.e., R_c shown in equation 3.6).

Furthermore, calculating and managing the trust of the service of the SNP take sensor data collection trust (i.e., T_{k1} presented in equation (3.7)), sensor network lifetime trust (T_{k2} presented in equation (3.8)), sensor network response time trust (i.e., T_{k3} presented in equation (3.9)) as well as sensor data

transmission trust (i.e., T_{k4} presented in equation (3.10)) into account. The trust value of the service of the SNP is the minimum value of T_{k1}, T_{k2}, T_{k3} and T_{k4}. Finally, the calculation and management of the reputation of the service of the SNP (i.e., R_k presented in equation (3.12)) are based on the history that CSPs selected the service of the SNP and the history that CSPs required the service to receive from a SNP.

From the above analysis, we can obtain that the proposed ARTCM system is capable of calculating and managing the trust and reputation about the service of CSP and SNP.

Helping CSU Choose Desirable CSP and Assisting CSP in Selecting Appropriate SNP

Regarding helping CSU to choose desirable CSP as well as assisting CSP in selecting appropriate SNP, Part 2) Trust and reputation calculation and management flowchart between CSU and CSPs and Part 3) Trust and reputation calculation and management flowchart between CSP and SNPs shown in Section 3.3, present the detailed mechanisms to validate our demonstration. Specifically, from equation (3.17) and equation (3.23), we can see that the cost and trust as well as the reputation of the service of CSP and SNP are utilized for CSU and CSP to make the corresponding choice.

Case Study 1: In the following sample case study, there are three CSUs, four CSPs and five SNPs. With the filter process of the Step 1 of Part 2) and Part 3), we assume that one CSP and two SNPs are filtered out as their attributes do not satisfy the requirements. Then there are three CSUs, three CSPs and three SNPs, in which all characteristics of CSPs satisfy the attribute requirement of CSUs and all characteristics of SNPs satisfy the attribute requirement of CSPs.

In the following, Table 3.6 shows the detailed parameters with respect to CSUs and qualified CSPs about C_c, T_{cu}, R_c, C_{bc}, T_{scu} and R_{sc}, which will be used from Step 2 to Step 5 of Part 2). And table 3.7

Table 3.6: Parameters of CSUs and qualified CSPs

	C_c	T_{cu}	R_c	C_{bc}	T_{scu}	R_{sc}
$CSU_1 \leftrightarrow CSP_1$	-10	0.7	0.8	[-30, 30]	0.5	0.5
$CSU_1 \leftrightarrow CSP_2$	-15	0.8	0.7	[-30, 30]	0.5	0.5
$CSU_1 \leftrightarrow CSP_3$	-20	0.9	0.6	[-30, 30]	0.5	0.5
$CSU_2 \leftrightarrow CSP_1$	-15	0.7	0.8	[-30, 30]	0.5	0.5
$CSU_2 \leftrightarrow CSP_2$	-20	0.8	0.7	[-30, 30]	0.5	0.5
$CSU_2 \leftrightarrow CSP_3$	-25	0.9	0.6	[-30, 30]	0.5	0.5
$CSU_3 \leftrightarrow CSP_1$	-20	0.7	0.8	[-30, 30]	0.5	0.5
$CSU_3 \leftrightarrow CSP_2$	-25	0.8	0.7	[-30, 30]	0.5	0.5
$CSU_3 \leftrightarrow CSP_3$	-30	0.9	0.6	[-30, 30]	0.5	0.5

Table 3.7: Parameters of qualified CSPs and SNPs

	C_k	T_{kc}	R_k	C_{bk}	T_{skc}	R_{sk}
$CSP_1 \leftrightarrow SNP_1$	-10	0.8	0.7	[-30, 30]	0.5	0.5
$CSP_1 \leftrightarrow SNP_2$	-15	0.7	0.6	[-30, 30]	0.5	0.5
$CSP_1 \leftrightarrow SNP_3$	-20	0.6	0.5	[-30, 30]	0.5	0.5
$CSP_2 \leftrightarrow SNP_1$	-15	0.8	0.7	[-30, 30]	0.5	0.5
$CSP_2 \leftrightarrow SNP_2$	-20	0.7	0.6	[-30, 30]	0.5	0.5
$CSP_2 \leftrightarrow SNP_3$	-25	0.6	0.5	[-30, 30]	0.5	0.5
$CSP_3 \leftrightarrow SNP_1$	-20	0.8	0.7	[-30, 30]	0.5	0.5
$CSP_3 \leftrightarrow SNP_2$	-25	0.7	0.6	[-30, 30]	0.5	0.5
$CSP_3 \leftrightarrow SNP_3$	-30	0.6	0.5	[-30, 30]	0.5	0.5

Table 3.8: Weight set 1 of CSUs and corresponding choices

	α_c	β_c	γ_c	*Choice*
CSU_1	1/3	1/3	1/3	CSP_3
CSU_2	1/2	1/4	1/4	CSP_3
CSU_3	1/5	2/5	2/5	CSP_3

Table 3.9: Weight set 1 of qualified CSPs and corresponding choices

	α_k	β_k	γ_k	$Choice$
CSP_1	1/3	1/3	1/3	SNP_1
CSP_2	1/2	1/4	1/4	SNP_1
CSP_3	1/5	2/5	2/5	SNP_1

Table 3.10: Weight set 2 of CSUs and corresponding choices

	α_c	β_c	γ_c	$Choice$
CSU_1	1	0	0	CSP_3
CSU_2	0	1	0	CSP_3
CSU_3	0	0	1	CSP_1

Table 3.11: Weight set 2 of qualified CSPs and corresponding choices

	α_k	β_k	γ_k	$Choice$
CSP_1	1	0	0	SNP_3
CSP_2	0	1	0	SNP_1
CSP_3	0	0	1	SNP_1

presents the detailed parameters regarding qualified CSPs and SNPs, that will be utilized from Step 2 to Step 5 of Part 3) about C_k, T_{kc}, R_k, C_{bk}, T_{skc} and R_{sk}. Moreover, two typical weight sets about α_c, β_c, γ_c as well as α_k, β_k and γ_k are used to validate the effectiveness. In weight set 1, CSUs and CSPs take C_c, T_{cu} and R_c all into account. For weight set 2, CSUs and CSPs only consider one of C_k, T_{kc} and R_k.

Weight set 1 of CSUs and the corresponding choices with respect to CSPs are shown in Table 3.8. Meanwhile, weight set 1 of qualified CSPs and the corresponding choices with respect to SNPs are shown in Table 3.9. With equation (3.17) and equation (3.23), we can get that CSU_1, CSU_2 and CSU_3 all choose CSP_3 as shown in Table 3.8. In addition, CSP_1, CSP_2 and CSP_3 all select SNP_1 as presented in Table 3.9.

Furthermore, Table 3.10 and Table 3.11 present weight set 2 of CSUs and the corresponding choices with respect to CSPs as well as weight set 2 of qualified CSPs and the corresponding choices with respect to SNPs, respectively. Similarly, based on equation (3.17) and equation (3.23), we can obtain that CSU_1 and CSU_2 select CSP_3 while CSU_3 chooses CSP_1 as presented in Table 3.10. Meanwhile, CSP_1 chooses SNP_3 while CSP_2 and CSP_3 both select SNP_1 as shown in Table 3.11.

Case Study 2: In the following sample case study, there are one hundred CSUs, one hundred and fifty CSPs and two hundred SNPs. With the filter process of the Step 1 of Part 2) and Part 3), we suppose that fifty CSPs and one hundred SNPs are filtered out as their characteristics are not satisfied. Then there are one hundred CSUs, one hundred CSPs and one hundred SNPs, in which all characteristics of CSPs satisfy the attribute requirement of CSUs and all characteristics of SNPs satisfy the attribute requirement of CSPs.

In the following, the detailed parameters with respect to CSUs and qualified CSPs about C_c, T_{cu}, R_c, C_{bc}, T_{scu} and R_{sc} are randomly initialized and they will be utilized from Step 2 to Step 5 of Part 2).

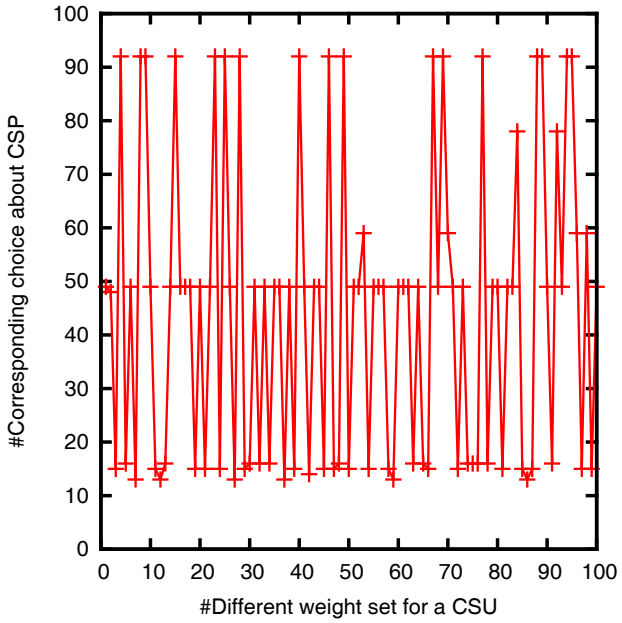


Figure 3.1: Different weight set for a CSU and corresponding choice about CSP

Similarly, the detailed parameters about qualified CSPs and SNPs are randomly initialized and they will be used from Step 2 to Step 5 of Part 3) about C_k, T_{kc}, R_k, C_{bk}, T_{skc} and R_{sk}. In addition, one hundred different weight sets about α_c, β_c, γ_c as well as α_k, β_k and γ_k are randomly initialized to validate the effectiveness.

Different weight sets for a CSU and the corresponding choices regarding CSPs are shown in Fig. 3.1. Meanwhile, different weight sets for a qualified CSP and the corresponding choices regarding SNPs are shown in Fig. 3.2. With equation (3.17) and equation (3.23), we can get that the CSU can choose CSP and CSP can choose SNP as shown in Fig. 3.1 and Fig. 3.2, respectively.

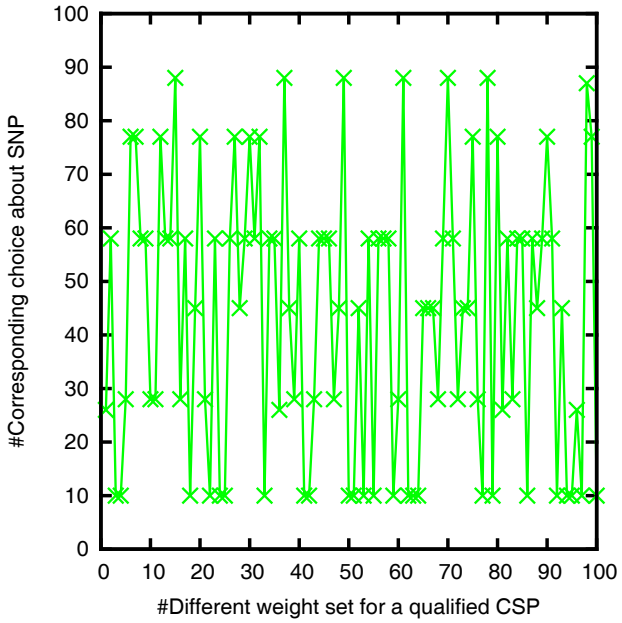


Figure 3.2: Different weight set for a qualified CSP and corresponding choice about SNP

Summary

From the above evaluation results, we can observe that our proposed ATRCM system is indeed able to assist the CSU in selecting authentic and desirable CSP as well as help CSP choose authentic and appropriate SNP, considering (i) the authenticity of CSP and SNP and (ii) the attribute requirement of CSU and CSP as well as (iii) the cost, trust and reputation of the service of CSP and SNP.

Moreover, we can deduce that different weight sets do not always change the corresponding results for CSU to choose CSP, by comparing the weight sets and corresponding choices about CSPs in Table 3.8 with that in Table 3.10 and observing the corresponding choices about CSPs with different weight sets

shown in Fig. 3.1. Similarly, the corresponding choices for CSP to select SNP are not always be affected by changing weight sets, comparing the weight sets and corresponding choices about SNPs in Table 3.9 with that in Table 3.11 and observing the corresponding choices about SNPs with different weight sets presented in Fig. 3.2.

3.5 Analysis of System Security

In this section, we analyze our proposed ATRCM system from the view of security by providing a few adversary models, in which we follow Dolev-Yao approach [65]. Particularly, we analyze whether ATRCM is immune to the following four attacks [66] [67] (i.e., good mouthing attack, bad mouthing attack, collusion attack and white-washing attack).

3.5.1 First Adversary Model: Good Mouthing and Bad Mouthing Attacks

Mechanism

The adversary (e.g., a malicious CSU or a malicious CSP) provides a malicious feedback about its experience with another party (a CSP or a SNP). For example, a malicious CSU provides a malicious feedback about its experience with a CSP or a malicious CSP provides a malicious feedback about its experience with a SNP, even if the experience actually does not exist. The feedback could be wrong positive feedback (i.e., good mouthing attack) or wrong negative feedback (i.e., bad mouthing attack).

Initial Capability

The adversary knows the operational mechanism of ATRCM system and is free to produce wrong feedbacks about any service of any CSP or any SNP.

Capability during the Attack

During the attack, the adversary is able to send feedbacks periodically to the TCE about the experience via a secure communication.

Discussion

The good mouthing and bad mouthing attack cannot maliciously subvert the trust or reputation value as the adversary wishes, because of the following:

- False feedbacks from the adversary to the TCE will not be utilized to calculate the trust or reputation value, as whether the feedbacks received by TCE are genuine are audited by TCE.
- The trust and reputation of a CSP or a SNP on providing a service are largely dependent on the historical feedbacks of previous SLAs or PLAs about the service, meaning that the historical trust values can effectively maintain the trust or reputation.

3.5.2 Second Adversary Model: Collusion Attack

Mechanism

The adversaries (i.e., malicious CSUs or malicious CSPs or malicious SNPs) collude other parties mutually (e.g., a malicious CSU collude a malicious CSP or a malicious CSP collude a malicious SNP) and

participate in events that generate real positive feedbacks for the colluding participants.

Initial Capability

The adversaries know about the operational mechanism of ATRCM system and they are free to collude any CSP or any SNP mutually.

Capability during the Attack

During the attack, the adversaries are able to to change their colluding parties dynamically, without noticing TCE.

Discussion

From [68], since the colluders are synthesizing events that create verifiable feedbacks between CSUs and CSPs or between CSPs and SNPs in a collective way, they are able to improve their trust and reputation values faster than the honest participants or counter the effects of possible negative feedbacks. Thus, it is hard to mitigate the collusion attack, without detecting and reacting to the groups of colluders who interact exclusively with each other, while discovering these colluders that are formulated as discovering a clique of a certain size within a graph is known to be NP-complete and only heuristic-based solutions have been proposed.

On the other hand, to launch the collusion attack, since TCE is supposed to be informed about the signed SLAs or PLAs and TCE is capable of monitoring the process of system (i.e., TCE has the role of the cloud auditor), TCE should sense the service delivery at the minimum. Therefore, in case of this attack, the colluding participants should initially report dummy SLAs or PLAs to TCE followed by bogus

feedbacks, and then do not actually deliver the service. However, if the service delivery is not actually performed, TCE will detect this. Thus, TCE can detect this attack and further filter out the bogus feedbacks.

Note: In case of the first and second adversary models, since in the trust and reputation management system *punishment* is a normal action after finding a malicious entity, the TCE punishes the attacker. For instance, the TCE can filter out the feedbacks initiated by an adversary after finding an attack lunched by the adversary. Then, the TCE can decrease the services provided to the adversary to punish the adversary. Therefore, these attacks are costly for the adversary and the high cost can prevent the adversary to perform the attacks.

3.5.3 Third Adversary Model: White-Washing Attack

Mechanism

The adversary (e.g., a malicious CSP or a malicious SNP) resets a poor trust or reputation, by rejoining the system with a new identity and a fresh trust or reputation.

Initial Capability

The adversary knows the operational mechanism of ATRCM system, and is free to re-enter the system at any time with a new identity and a fresh trust or reputation.

Capability during the Attack

During the attack, the adversary is able to switch their identities dynamically, without informing TCE.

Discussion

The white-washing attack cannot mislead the honest customers by resetting a poor trust or reputation as the adversary wishes, because of the following:

- In case of a malicious CSU, the adversary can rejoin the system only to lunch other attacks such as bad mouthing attack. However, since the trust and reputation evaluation of CSU is not within our system targets, rejoining the system does not affect the trust or reputation value of the CSU. In fact, these two values are not utilized by ATRCM system.

- When a malicious CSP or a malicious SNP rejoins the system as a new identity, it needs to be authenticated by the CSU or the CSP based on the ISO/IEC 27001 certification, then the CSU or the CSP will know its original identity and rejoining purpose.

- The trust and reputation are different in the ATRCM system in terms of newcomers and participants that have shown good behaviors for a long time. Thus, it is hard to cheat the honest customers by letting them easily choose newcomers.

- Finally, even if the adversary resets its negative trust value and restarts as a fresh entity, in return the adversary loses its reputation completely (as per equation (3.6) and equation (3.12)). Furthermore, the reputation is a positive value all the time, and resetting it puts the adversary in a vulnerable and risky position of not being selected by any customer for a long time (as per equation (3.17) and equation (3.23)).

Chapter 4

Towards Offering More Useful Data Reliably to Mobile Cloud from Wireless Sensor Network

4.1 Usefulness of Sensory Data and Reliability of WSN

4.1.1 Usefulness of Sensory Data

In the context of WSN-MCC integration, this chapter considers the usefulness of sensory data according to whether the sensory data offered by the WSN is eventually utilized by the cloud to satisfy the data requests from mobile users. We observe that characteristics of user data requests are key factors that affect the usefulness of sensory data.

User Data Request Characteristics

The behavior that a user issues data requests to the cloud is usually characterized by *Time* [69] [70]. For example, it is very common in our daily life that "usually someone (say, Bob) will do something (say,

watch a talk show) with some mobile device during some time period (e.g., 2:00 pm to 3:00 pm)". In addition, the content of the data requested by a user usually has *Priority* [70] [71]. For instance, although the traffic information of the whole city has some value to a certain extent, a lot of mobile users may be more interested in the traffic information of the downtown than the traffic information in a quiet countryside during the same time period. Meanwhile, a substantial number of mobile users may only be interested in the traffic information of a certain set of places (e.g., company, living residence, restaurant, school) among all the places in the city.

In short, the transmitted data from the WSN to the cloud may not be fully utilized by the cloud to satisfy the mobile users' data requests, as mobile users generally issue data requests for some certain contents for a specific time period. Thus, it is not necessary for the WSN to always transmit all the sensory data to the cloud, since it is not efficient and it also increases the transmission bandwidth requirement and exacerbates the network traffic.

However, all sensory data still need to be able to be collected by sensors, as mobile users may request data from any sensor at any time, though with highly varied probabilities. In this chapter, since usually the data request behaviors of mobile users are characterized by time and priority and there are various factors (e.g., communication interference, network congestion, limited bandwidth) [72] [73] that affect the data gathering and data transmission latency, we ignore the issue regarding latency of gathering and transmitting data to satisfy data requests from mobile users.

4.1.2 Reliability of WSN

In WSN-MCC integration, one aspect of WSN reliability relates to whether the WSN is continuously able to gather and transmit the sensory data to the cloud successfully. We observe the following critical issues

concerning the reliability of WSN.

Depletion of Sensor Energy

Generally, sensors will deplete their limited battery power by performing data sensing, processing and transmission after a certain period of time, as they are often equipped with batteries that are not rechargeable and battery replacement may also be impractical [74] [75]. Particularly, the sensors close to the gateway are serving as intermediary nodes that forward most packets to the gateway on behalf of the source nodes. Therefore they may deplete their energy sooner than other sensors and form holes in the WSN where no data can be collected for the cloud, or cause the WSN to be disconnected.

Failures in Sensory Data Transmissions

The data transmissions from one sensor to another sensor and from the WSN to the cloud may encounter failures or losses, due to various factors such as network congestion, limited bandwidth or interference [72] [76]. In such cases, if the WSN does not perform data retransmission, then the cloud cannot obtain the sensory data coming from the WSN. In this chapter, we consider that this reliability issue is always overcome through data retransmissions.

Limit in Storage Space for Sensory Data

As stated in [77], data storage is a very serious issue for WSN, since a large volume of collected data needs to be archived for future information retrieval. In addition, when sensors are deployed to gather multimedia data such as images or videos that usually have large sizes, this further aggravates the demand on sensory data storage space. If the sensors do not have available storage space to store the sensed data,

then the cloud cannot obtain any sensory data, even if the sensors have enough residual energy to gather and transmit data and the data transmissions from WSN to cloud are successful. In this chapter, we assume that sensors have sufficient storage space.

In this chapter, we do not consider the sensory data transmission failure and sensory data storage space limit issues for reasons given above, but instead focus on the sensor energy depletion issue, which strongly affects the reliability of a WSN.

4.2 WSN-MCC Integration System Model

The WSN-MCC integration system is modeled and analyzed in this chapter based on the following assumptions.

- There is one cloud C and M mobile users (i.e., $U = (u_1, u_2, \ldots, u_M)$) as well as M WSNs (i.e., $WSN = (wsn_1, wsn_2, \ldots, wsn_M)$). Each WSN gathers and transmits data to the cloud to satisfy the data requests from each corresponding mobile user.
- Each WSN consists of one gateway g as well as N sensor nodes (i.e., $I = (i_1, i_2, \ldots, i_N)$).
- Each gateway g is externally powered with an unlimited energy supply. Each sensor node i has a limited energy supply powered by a non-rechargeable and non-replaceable battery, which has an initial energy e_o and a residual energy e_i.
- Time is divided into Z time periods (i.e., $T = (t_1, t_2, \ldots, t_Z)$).

Table 4.1: Example of point vs time & priority (PTP) table

Point of Interest	9 am-10 am	10 am-11 am	11 am-12 pm	12 pm-1 pm	...
i_1	10%	5%	20%	15%	...
i_2	20%	5%	0%	15%	...
i_3	20%	10%	0%	15%	...
i_4	10%	10%	0%	10%	...
i_5	20%	20%	0%	15%	...
i_6	10%	20%	30%	15%	...
i_7	5%	20%	40%	5%	...
i_8	0%	5%	0%	5%	...
...	...	...	...	...	...

4.3 TPSDT and PSS

In this section, we present and discuss the proposed 1) TPSDT mechanism for WSN gateway to selectively transmit sensory data that are more useful to the cloud and 2) PSS mechanism for WSN to save energy consumption so that it can gather and transmit data in a more reliable way.

4.3.1 TPSDT

The difference between our proposed TPSDT and other selective data transmission methods (e.g., [78] [79] [80]) in WSN is that TPSDT is the first method for WSN gateway to selectively transmit data to the cloud, considering the time and priority characteristics of the data requested by the mobile user. These characteristics are recorded in a Point vs Time & Priority (PTP) table maintained in the cloud for each mobile user, where each point corresponds to a sensor node and the time reflects the specific time period and the priority reflects the probability that the mobile user requests data from the corresponding sensor node during that time period.

PTP Table

Based on the time and priority features illustrated in Section 4.1 about mobile user data requests, we consider that the cloud is able to analyze the historical behaviors of mobile user data requests and then maintain a PTP table of each mobile user with respect to time and priority for sensor nodes of interest.

An example of this PTP table reflecting the interest of a mobile user is shown in Table 4.1. Specifically, the probability that the data requests correspond to each point of interest, as shown in the PTP table, represents the priority of the requested data to the mobile user. A higher probability connected to a given point in a specific time period means that the mobile user is more interested in that point and is more likely to issue data requests for the point in that specific time period.

Assume the number of data requests issued for a point of interest (e.g., sensor node i) during each specific time period t in the history is r_i^t. In addition, given that the number of data requests issued to all points for each specific time period t in the history is R^t. The probability (i.e., p_i^t) that the data requests concern the sensor node i in each specific time period t is calculated as follows.

$$p_i^t = \frac{r_i^t}{R^t} \tag{4.1}$$

In addition, for the whole WSN-MCC integration, there are Z time periods and N sensor nodes. Thus,

$$1 = \sum_{i=i_1}^{i=i_N} p_i^t \quad (t = t_1, t_2, \ldots, t_Z) \tag{4.2}$$

This PTP table obtained for each mobile user is updated dynamically by the cloud C and sent to the gateway g of each corresponding WSN.

Details of TPSDT

With the PTP table, the process of our proposed TPSDT for each WSN gateway to selectively transmit data that are more useful to the cloud is shown as follows.

- 1) Each gateway g sets a timer, which records the current time.
- 2) For each time period t, each gateway g sends the sensory data to the cloud C, according to the start time and end time of t in the PTP table.
- 3) Particularly, for the transmitted data content, each gateway g sends the sensory data gathered by each sensor node in order, according to the priorities (i.e., probabilities in the PTP table). The sensory data gathered by the points of interest with larger priorities are sent first, followed with sensory data collected by those with lower priorities. The sensory data coming from the points of interest with no priority (i.e., probability is 0%) in the PTP table are not transmitted.

4.3.2 PSS

The difference between our proposed PSS and other sleep scheduling algorithms (e.g., [43] [44] [45] [81]) in WSN is that PSS first incorporates the time and priority characteristics of the data requested by the mobile user into the WSN sleep scheduling process to gather and transmit data for the cloud, with PTP table.

Design Factors

The design of the proposed PSS algorithm considers the following three factors: 1) the points of interest (i.e., sensor nodes of interest in WSN) in the PTP table with probability larger than 0% should be awake

Pseudocode of PSS algorithm

First: Run the following at gateway g during each time period t.

Step 1: Gateway g obtains PTP table.
Step 2: If $p_i^t > 0$, g sends flag A to node i.
Step 3: Run the second part at each node i.

Second: Run the following at each node i during each time period t.

Step 1: Get the current residual energy rank e_i.
Step 2: Broadcast e_i and receive the energy ranks of its currently awake neighbors N_i. Let E_i be the set of these ranks.
Step 3: Broadcast E_i and receive E_j from each $j \in N_i$.
Step 4: If $|N_i| < k$ or $|N_j| < k$ for any $j \in N_i$, remain awake. Go to Step 7.
Step 5: Compute $C_i = \{j | j \in N_i \text{ and } e_j > e_i\}$.
Step 6: Go to sleep if both the following conditions hold. Remain awake otherwise.
- Any two nodes in C_i are connected either directly themselves or indirectly through nodes within i's 2-hop neighborhood that have e more than e_i.
- Any node in N_i has at least k neighbors from C_i.
- It does not receive flag A.

Step 7: Return.

in each time period t, since mobile user requires sensory data gathered by these sensor nodes; 2) the whole sleep scheduled network should be connected so that data transmissions from sensor nodes to gateway can be performed; 3) only a subset of all sensor nodes should be awake in each time period t to reduce energy consumption - the sensor nodes that are scheduled to be awake should generally have more residual energy than the nodes that are scheduled to be asleep, so that network lifetime could be further prolonged.

Details of PSS

Considering the above three design factors, the pseudocode of the proposed PSS algorithm is as shown above.

Analysis of PSS

Property 1: The PSS algorithm guarantees that sensor nodes required to satisfy the anticipated data requests of mobile users are awake.

Discussion: We discuss this property by observing the execution process of PSS. Particularly, with respect to the sensor nodes from which mobile users require data, these sensor nodes are actually the points of interest in the PTP table with the corresponding probabilities larger than 0% (i.e., $p_i^t > 0$). From step 2 of the first part of PSS, we can observe that sensor node i will receive flag A if $p_i^t > 0$. Further, with step 6 of the second part of PSS, sensor nodes that receive the flag A cannot be asleep. In other words, sensor nodes receiving flag A will all be awake so as to gather and transmit data requested by the mobile users.

Property 2: PSS algorithm maintains a connected network if the original network is connected.

Discussion: We discuss this property by contradiction [43] [44] [45]. Given that the sleep scheduled network after running the PSS is not connected. Then, we put the deleted nodes (asleep nodes determined by PSS) back in the network, in descending order of their energy ranks. Let i be the first sensor node making the network connected again. Note that by the time we put i back, all the members of C_i are already present and sensor nodes in C_i are already connected since they are connected by nodes with $e > e_i$. Let v be a node that was disconnected from C_i but now is connected to C_i by i. Then, this contradicts the fact that i can sleep only if all its neighbors (including v) are connected to $\geq k$ nodes in C_i (Step 6 of the second part of PSS).

Property 3: The PSS algorithm prolongs the network lifetime compared with the always-on WSN.

Discussion: We discuss this property by analyzing the execution results after running PSS algorithm. First, from the entire steps of PSS, we can observe that after PSS, there is a subset of sensor nodes that

determine to be awake and there is another subset of sensor nodes that are asleep. As only a subset of sensor nodes need to be awake, the energy consumption will be saved and the network lifetime will be prolonged, compared with the always on WSN scheme in which all sensors are always awake.

Second, based on step 6 of the second part of PSS, we can observe that after PSS, the asleep sensor nodes satisfy that 1) Any two nodes in C_i are connected either directly themselves or indirectly through nodes within i's 2-hop neighborhood that have e more than e_i. This constraint means that the awake nodes own more residual energy than the asleep sensor nodes after sleep scheduling. By letting the sensor nodes with more residual energy rather than the sensor nodes with less residual energy be awake to perform data sensing, data storage and data processing, the network lifetime is further prolonged.

4.4 Proposed TPSS Scheme

4.4.1 Overview

Fig. 4.1 shows the proposed TPSS scheme to gather and transmit sensory data for WSN-MCC integration, towards reliably offering data which are more useful to the mobile users from WSN to cloud. The detailed steps of TPSS for each WSN to gather and transmit sensory data for each corresponding mobile user are depicted as follows.

- 1) Sensor nodes determine their awake and asleep states with PSS.
- 2) Sensor nodes sense the environmental data with a set frequency and store the sensory data as well as process the sensory data.
- 3) Sensor nodes send the processed sensory data to the gateway g with the many to one and hop

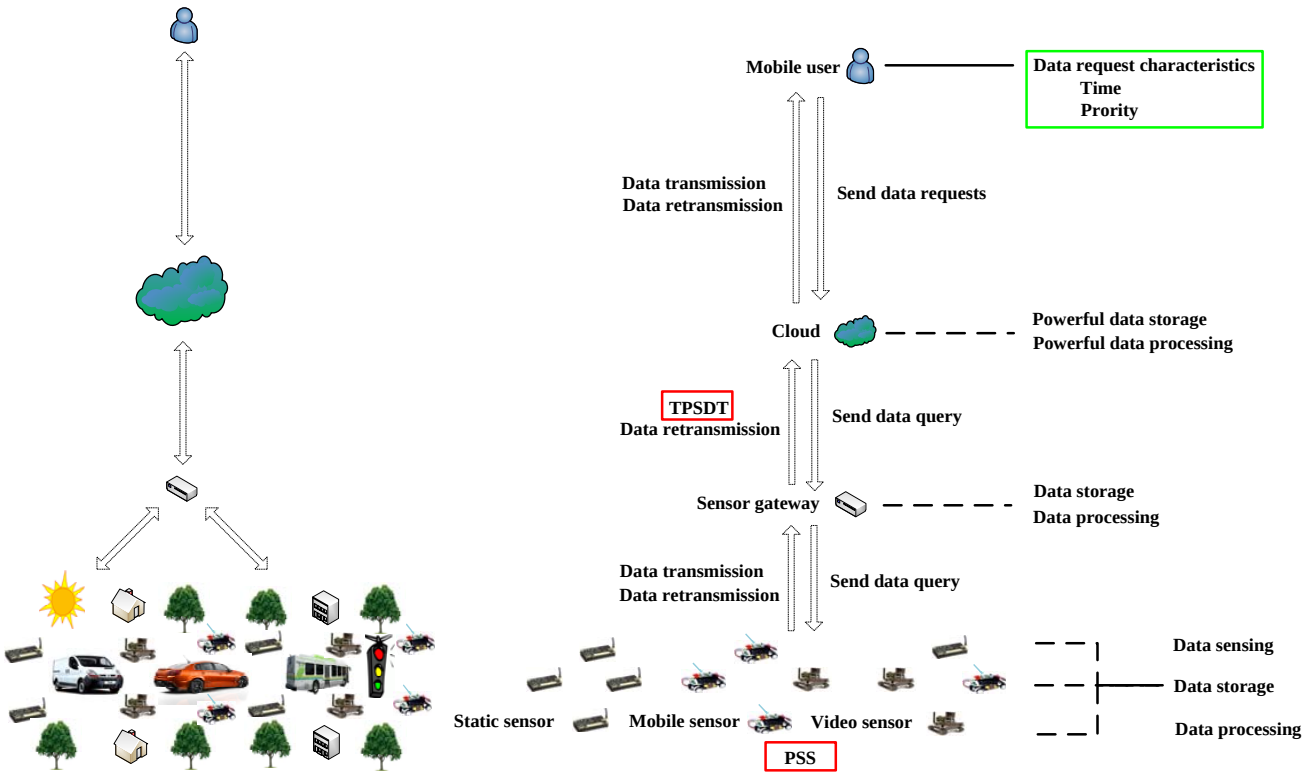


Figure 4.1: Proposed TPSS scheme to gather and transmit sensory data for WSN-MCC integration

by hop pattern.

- 4) Gateway g stores the received sensory data and then processes the sensory data.
- 5) Gateway g selectively transmits the sensory data to the cloud C with TPSDT.
- 6) Cloud C further stores and processes the received sensory data.
- 7) If data transmission from i to g or g to C experiences data losses or failures, i or g performs data retransmission until the data transmission is successful.
- 8) Mobile user u issues data requests to cloud C and cloud C transmits the requested sensory data to the mobile user u.
- 9) If data transmission from C to u encounters data losses or failures, C performs data retransmission until the data transmission is successful.
- 10) Cloud C dynamically updates the PTP table with equation (4.1) if the time and priority features of the requested data of the mobile user are changed and sends the updated PTP table to gateway in each time period t.

4.4.2 Scheme Characteristics and Analysis

Fig. 4.2 shows the general scheme (GS) to gather and transmit sensory data for WSN-MCC integration. Comparing Fig. 4.2 and Fig. 4.1, based on the above introductions, we can see that our proposed TPSS shares the same technique with GS (i.e., data retransmission) to mitigate data transmission losses or failures in sensory data transmissions for improving the reliability of WSN during WSN-MCC integration.

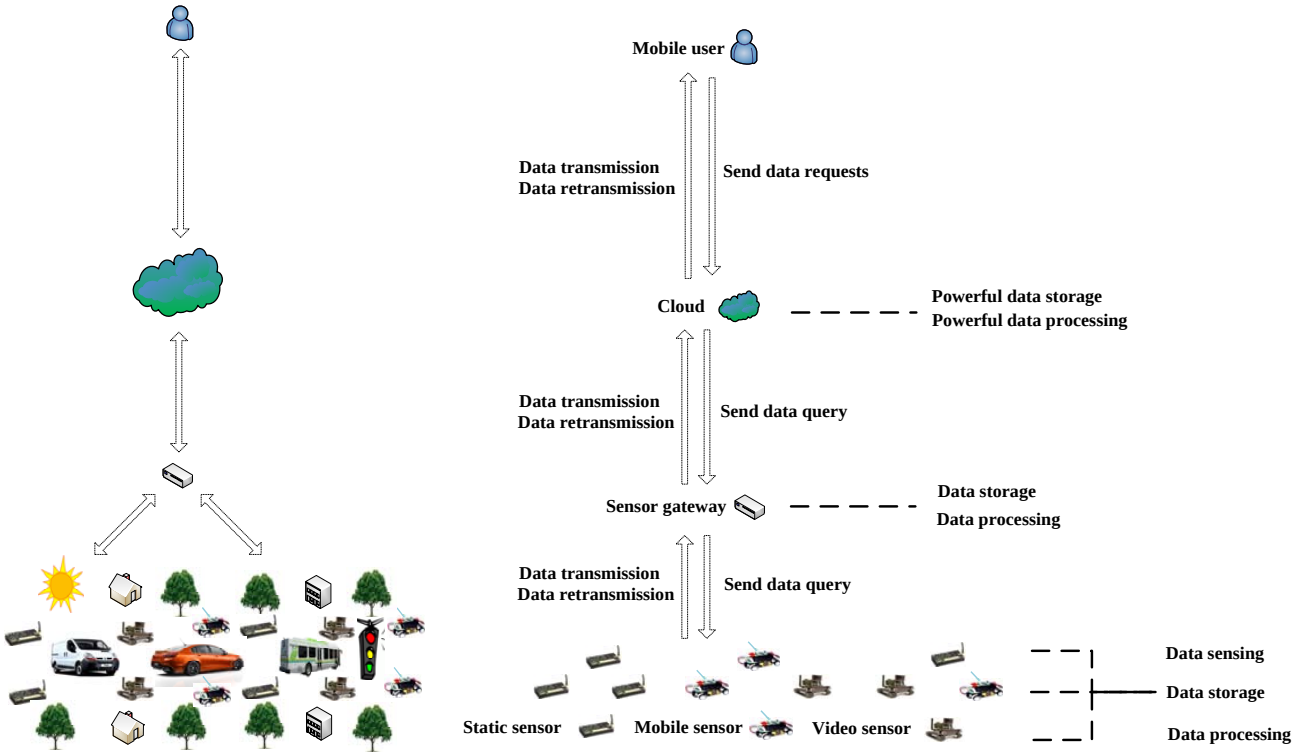


Figure 4.2: General scheme (GS) to gather and transmit sensory data for WSN-MCC integration

In addition, we can observe that TPSS differs from GS to gather and transmit sensory data for WSN-MCC integration, with respect to the following two aspects.

TPSDT for WSN Gateway

In our proposed TPSS, the gateway g selectively transmits the sensory data to the cloud C with TPSDT.

This design is to enhance the usefulness of sensory data, since TPSDT data transmission is based on the PTP table deduced from the time and priority features of the data requested by the mobile user. Thus, normally the successfully transmitted sensory data to the cloud will all be utilized to answer mobile user data requests.

In the case that the mobile user u issues data requests for sensory data currently not stored in the cloud C in the time period t, as the PTP table is dynamically updated with equation (4.1) if the time and priority features of the requested data of the mobile user are changed in t (Step 10 of the TPSS), running the PSS algorithm with the updated PTP table in t makes the sensor nodes from which mobile user requires data awake (property 1 of PSS). In other words, the cloud is capable of answering the unexpected data requests by dynamically updating PTP table.

Moreover, as the sensory data are selectively transmitted from WSN gateway to the cloud with TPSDT, the bandwidth requirement and network congestion are reduced meanwhile. This can also enhance the reliability of WSN, as it alleviates the data transmission loss failure or loss issue to some extent.

PSS for WSN

In our proposed TPSS, the sensor nodes dynamically determine their awake and asleep states according to PSS.

This design can greatly improve the reliability of WSN, since PSS can greatly save the energy consumption and prolong the network lifetime (property 3 of PSS) so that WSN can gather and transmit data longer. Specifically, when sensors normally work for a certain period of time, the energy of the sensors will be consumed very fast and sensor nodes will die and cannot work any more. With PSS, sensor nodes are dynamically awake and asleep and only a subset of sensor nodes with more residual energy are required to be awake in each time period, this will greatly alleviate the sensory energy depletion issue that seriously affects the reliability of WSN and the lifetime of WSN will be greatly enhanced.

4.5 Evaluations

4.5.1 Evaluation Setup

To perform evaluations, we compare our proposed TPSS with GS to gather and transmit sensory data for WSN-MCC integration. Then we analyze whether our proposed TPSS is effective in enhancing the usefulness of sensory data and reliability of WSN.

WSN-MCC Integration System

We assume that there is one cloud and 10 mobile users and 10 WSNs. Each WSN gathers and transmits data to the cloud, enabling it to reply to data requests from each corresponding mobile user. We further assume that each WSN consists of one gateway, which has an unlimited energy supply, and 100 sensor nodes, each of which has an initial energy of 100000 mJ. Time is divided into 24 time periods, each one hour long. The detailed evaluation parameters are summarized in Table 4.2.

Table 4.2: Evaluation parameters in Chapter 4

Parameter	Parameter value
Number of clouds	1
Number of users	10
Number of WSNs	10
Number of sensor nodes	100
Number of gateways	1
Initial sensor energy	100000 mJ
Time period	1 hour
Network size	800×600 m^2
Default transmission radius	60 m
Transmission energy	0.0144 mJ
Reception energy	0.00576 mJ
Transmission amplifier energy	0.0288 nJ/m^2
Packet length	12 bytes
Number of packets	1000
k in PSS	1

Usefulness of Sensory Data

The usefulness of sensory data offered from a WSN to a cloud is evaluated, by analyzing how much the offered data from the WSN to the cloud is utilized to answer mobile user data requests, as illustrated in Section 4.1.

To analyze the usefulness of sensory data, the mobile user data request features (i.e., time and priority of the data requested by the mobile user) have to be obtained first. Regarding this, we perform the following experiment. We use a database resulting from 10 mobile users watching a surveillance video for three consecutive weeks from 10:00 am to 4:00 pm. For each day, we observe the time and priority of the data requested by the mobile users and they are transformed to different PTP tables. Then the same 10 mobile users watch the same surveillance video for another three consecutive weeks from 10:00

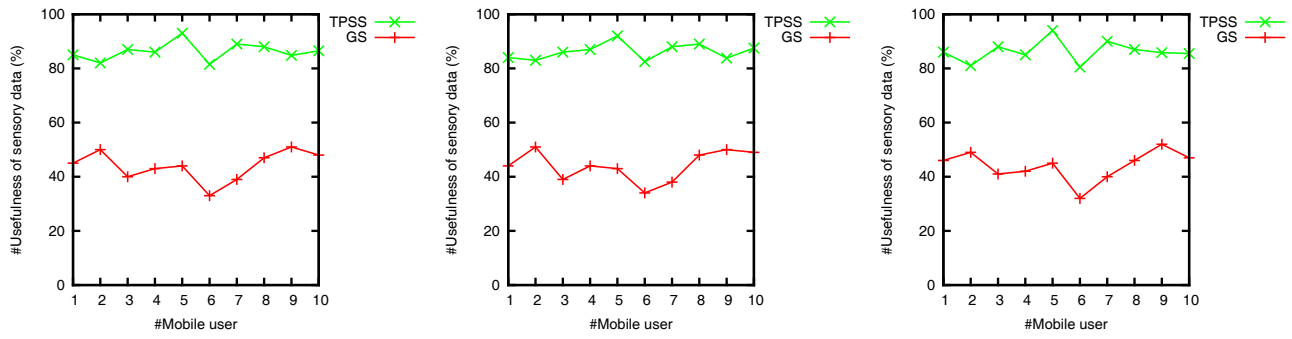
am to 4:00 pm and we analyze the usefulness of sensory data offered by TPSS and GS with the following assumptions.

For TPSS, we assume that the PTP tables are maintained by the cloud and further utilized by WSN gateways to transmit sensory data to the cloud with TPSDT for the 10 mobile users. For GS, we assume that the WSN gateways will transmit all sensory data to the cloud for the 10 mobile users, without utilizing the PTP tables. With that, the usefulness of sensory data which is actually the utility of the sensory data offered from the WSN to the cloud for each mobile user, could be respectively obtained for TPSS and GS as follows. Regarding TPSS, we analyze the percentage that the time and priority of the requested data of each mobile user observed in the previous three consecutive weeks are also observed in the subsequent three consecutive weeks. Regarding GS, we directly obtain the utility of the sensory data offered from the WSN to the cloud for each mobile user, by comparing the whole surveillance video with the content that each mobile user requests in the subsequent three consecutive weeks. The average utility of the sensory data offered from WSN to the cloud for each mobile user in each week is taken as the average usefulness of sensory data for each mobile user in each week.

Reliability of WSN

The reliability of WSN is evaluated by analyzing how long the WSN is able to gather and transmit sensory data to the cloud, as illustrated in Section 4.1. Specifically, we observe the network lifetime of WSN. In this chapter, the network lifetime of WSN is the time from the instant of network deployment to the instant when the first sensor node runs out of energy [42].

To analyze the reliability of WSN, we obtain the network lifetime of WSN in NetTopo [46], with each PTP table for each mobile user in the subsequent three consecutive weeks. For both TPSS and GS, the



(a) (b) (c)

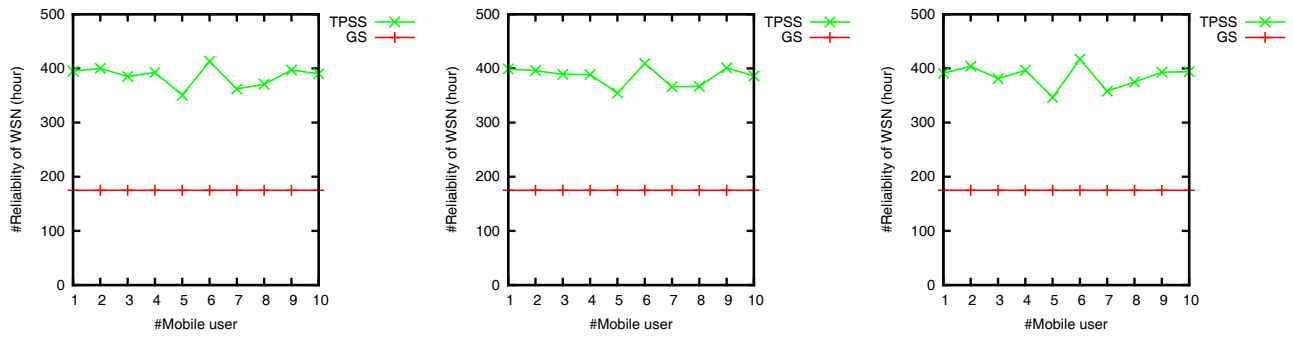
Figure 4.3: Average usefulness of sensory data for each mobile user in week 1 (a); in week 2 (b) and in week 3 (c)

network size is 800×600 m^2 and the default transmission radius is 60 m. The energy consumed by a sensor to transmit and receive one byte are, respectively, 0.0144 mJ and 0.00576 mJ [45] [75] [82]. The energy consumed by a sensor to power-amplify each transmitted byte to cover the distance of 1 m is 0.0288 nJ/m^2 [45] [75] [82]. The packet length is 12 bytes and there are 1000 packets transmitted during the communication time of each node [45] [75] [82]. The k in PSS is 1, which is the minimum value of k in PSS. The average value of the network lifetime of WSN gathering and transmitting data for each mobile user in each week is taken as the average reliability of WSN for each mobile user in each week.

4.5.2 Evaluation Results

The evaluation results with respect to usefulness of sensory data and reliability of WSN for each mobile user in each week are shown in Fig. 4.3 and Fig. 4.4, respectively.

From Fig. 4.3, we can observe that, averaging over all mobile users, around 85% of the sensory data sent to the cloud with TPSS are useful to the mobile users, whereas only around 45% of the sensory data



(a) (b) (c)

Figure 4.4: Average reliability of WSN for each mobile user in week 1 (a); in week 2 (b) and in week 3 (c)

sent to the cloud with GS are useful to the mobile users. The same results are obtained for each of the three weeks observed. This demonstrates that TPSS greatly improves the usefulness of sensory data due to the fact that mobile users generally request data over time according to the PTP tables.

From Fig. 4.4, we can observe that the reliability of WSN is also greatly enhanced with TPSS comparing GS. Particularly, the three sub-figures show that over each of the three weeks observed, the reliability of WSN with GS is around 180 hours for all the mobile users, while the reliability of WSN with TPSS varies among different mobile users but averages to around 400 hours.

In summary, TPSS substantially outperforms GS in terms of usefulness of sensory data and reliability of WSN. Moreover, for different mobile users with various data request characteristics indicated by different PTP tables, the usefulness of sensory data varies considerably for both TPSS and GS, as indicated in Fig. 4.3, while only the reliability of WSN for TPSS changes with mobile users as shown in Fig. 4.4.

Chapter 5

Trust-Assisted Sensor-Cloud

5.1 System Model

The system model in this chapter is summarized as below.

- A WSN includes one sink node sk, one source node se and η normal sensor nodes (i.e., $WSN = sk, se, i_1, i_2, i_3, \ldots, i_\eta$). The data from se is gathered and transmitted to the sk via normal sensor nodes and the data rate in the WSN is r kbps.
- A cloud includes ϕ data centers (i.e., $Cloud = j_1, j_2, j_3, \ldots, j_\phi$).
- δ Users (i.e., $Users = u_1, u_2, u_3, \ldots, u_\delta$) request the sensory data from cloud. The sensory data is transmitted from the WSN to the cloud.
- Time consists of τ time epochs (i.e., $Time = t_1, t_2, t_3, \ldots, t_\tau$). For each time epoch t, each sensor node i has a trust value $vs^t(i)$ and each data center j owns a trust value $vd^t(j)$.

5.2 Proposed TASC

About this section, some preliminary information regarding trust [52] [55] [83] is introduced first, followed with the overview about the proposed TASC.

5.2.1 Preliminaries about Trust

Trust is "assured reliance on the character, ability, strength or truth of someone or something", defined in Merriam Webster's Dictionary. In fact, trust is with different definitions and evaluations regarding various areas (e.g., wireless networks, sociology, philosophy, psychology, economics) [52] [83]. For example, the trust from node A to node B in terms of wireless communications, is the subjective expectation that node A achieves desirable outcomes from node B based on their interactions.

Generally, to perform the evaluation of trust from one subject (named as *trustor*) to another subject (named as *trustee*), the evidences (e.g., honesty, selfishness, vicious behaviors) reflecting the satisfaction regarding the trustee are needed. The evidences are either information based on direct interactions, or information coming from third-parties [55] [83]. Specifically, obtained through mapping evidences originated from direct interactions, the trustworthiness is *direct trust*. Achieved through mapping evidences offered by third-parties, the trustworthiness is *indirect trust* or *recommendation based trust*. Furthermore, defined as a weighted combination of direct and indirect trust, the recent behaviors are shown by *recent trust*. Defined with an exponential averaging update function of previous recent trusts, the long-term behaviors based on the previous experiences are presented by *historical trust*. Derived from both recent and historical trust, the expected performance of the target is demonstrated by *expected trust*.

After the trustor gathers various evidences about the trustee, the trustor achieves the trustworthiness value of the trustee, by using a function to map the collected evidences to the trustworthiness [52] [83].

For instance, the beta distribution presented as below is one kind of function to compute trustworthiness. Specifically, given that the accumulative amounts of positive and negative outcomes from the trustee in the evidences are Δ and Λ, the the trustee's trustworthiness v is computed as $v = \frac{\Delta+1}{\Lambda+\Delta+2}$.

5.2.2 Overview of TASC

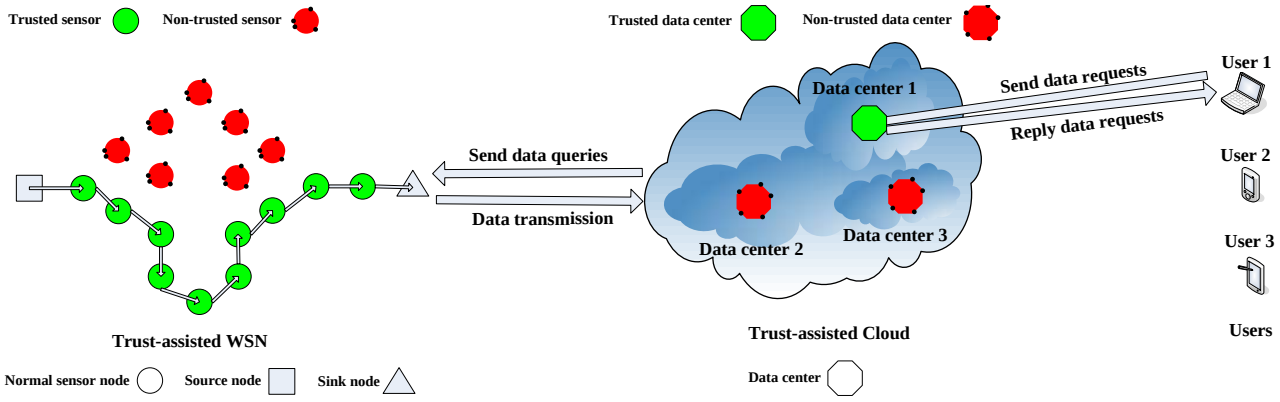


Figure 5.1: An instance about TASC

Fig. 5.1 shows an instance of TASC. With respect to TASC, the sensory data is gathered and transmitted to the cloud, by the trusted sensors (i.e., sensors which own trust values surpassing a threshold) in WSN. Then the sensory data is stored, processed and on demand delivered to the users, by the trusted data centers (i.e., data centers which own trust values surpassing a threshold) in cloud. In particular, 1) trust-assisted WSN and 2) trust-assisted cloud, are included in TASC.

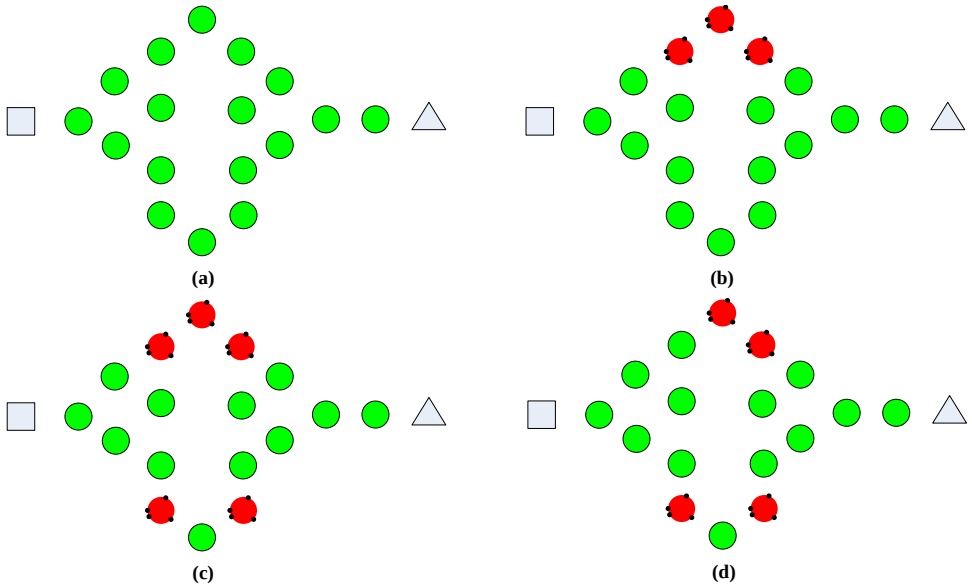


Figure 5.2: An instance about states of trust-assisted WSN

5.2.3 Trust-Assisted WSN

About trust-assisted WSN, the same trust value vs_o is owned by each sensor i initially. After a couple of time epochs (e.g., in the tth time epoch), due to that every sensor's behavioral pattern (e.g., data gathering history, data transmission history) is various generally, every sensor's trust value $vs^t(i)$ is changed in general. Specifically, higher trust values should be associated with the sensors that have less negative behavioral patterns. In addition, lower trust values should be associated with those sensors that have more negative behavioral patterns.

Moreover, the status that a sensor is trusted or not dynamically changes with time, as presented in Fig. 5.2 concerning trust-assisted WSN's states. Namely, previous non-trusted sensors can turn into trusted sensors, while previous trusted sensors can turn into non-trusted sensors. However, in terms of using sensors, trusted sensors are utilized for gathering and delivering sensory data to the cloud.

5.2.4 Trust-Assisted Cloud

Regarding trust-assisted cloud, initially, the same trust value vd_o is owned by each data center j. After a certain period (e.g., in the tth time epoch), the behavioral pattern (e.g., data storage history, data processing history, data delivery history) of every data center is different generally. Therefore, every data center's trust value $vd^t(j)$ is changed in general. Particularly, higher trust values should be associated with the data centers which have less negative behavioral patterns. Meanwhile, lower trust values should be associated with those data centers which have more negative behavioral patterns.

Furthermore, the status that a data center is trusted or not also changes with time, as shown in Fig. 5.3 regarding trust-assisted cloud's states. In other words, multiple non-trusted data centers and multiple trusted data centers could exist. When data centers are used, trusted data centers are used for storing,

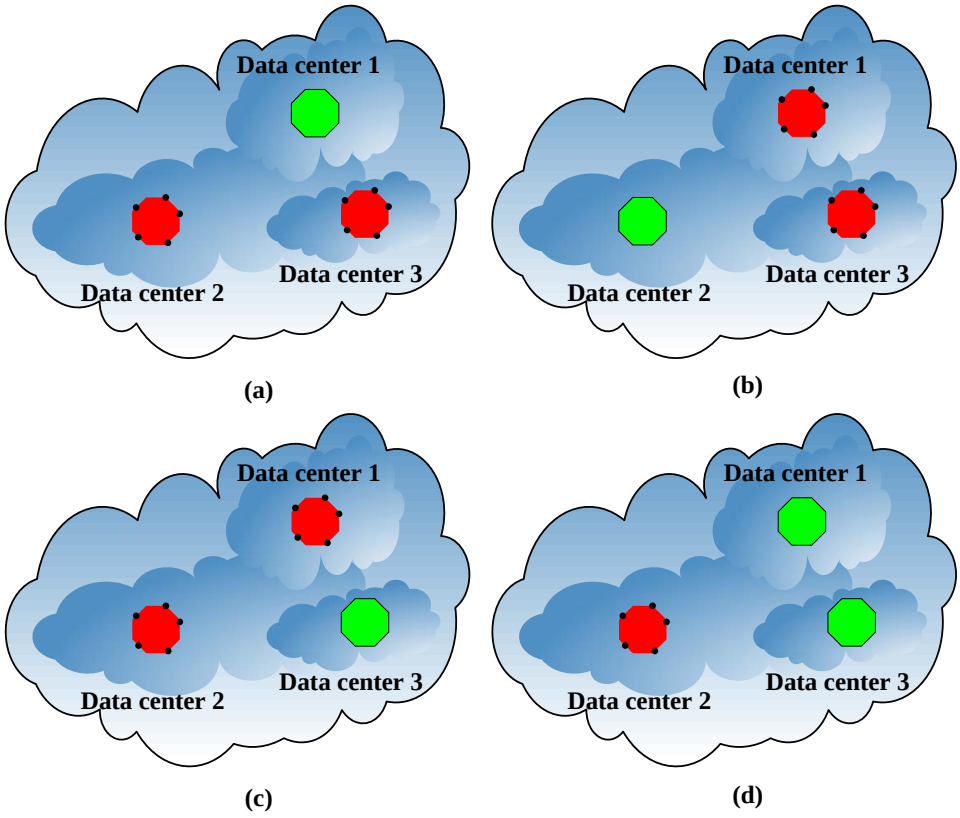


Figure 5.3: An instance about states of trust-assisted cloud

processing and then on demand delivering the sensory data to users.

5.3 Trust Values Computation, ITASC & CTASC & MTASC

In this section, the methods to compute trust values of sensor nodes and data centers in TASC are discussed first, followed with the introduction regarding the proposed three types of TASC (i.e., ITASC, CTASC and MTASC).

5.3.1 Trust Values of Sensor Nodes and Data Centers

For the state of art, there are various methods [52] [53] [54] that can be applied, to calculate the trust values of sensor nodes and data centers. Specifically, trust computation approaches can be summarized into *distributed trust computations* in which every sensor node or data center computes its own value of trust on its neighbors; or *centralized trust computations* where a central agent manages or helps the sensor node or data center in trust computations. For distributed trust computations, *direct trust*, *indirect trust* or a hybrid of direct trust and indirect trust (i.e., *recent trust*, *historical trust* or *expected trust*) is utilized by the trustor to calculate the trust value of the trustee. Regarding centralized trust computations, usually a TA (trust agent) that can be accessed by all sensor nodes or data centers in the community, is assumed. Then the trust values are computed by the TA for the whole group. Or the TA assists the sensor nodes or the data centers in their trust computations, by offering the initial trust values on target sensor nodes or data centers. In addition, one or many TAs might exist, based on the size of the WSN and cloud.

In this chapter, since different trust value computation methods have their own advantages and disadvantages, the historical trust that presents the long-term behavior is taken, as an instance to compute

the trust value $vs^t(i)$ of the sensor node i and the trust value $vd^t(j)$ of the data center j. Particularly, the following approaches [55] are shown, to compute the sensor nodes' trust values and data centers' trust values.

Computation of Trust Values of Sensor Nodes

Assume that up to z transactions in the tth time epoch, the historical trust which agent ps (i.e., a sensor node in WSN) owns regarding agent qs (i.e., another sensor node in WSN) is $vs_z^t(ps, qs)$. In addition, the forgetting factor (discounting older experiences) is $\rho s(0 \leq \rho s \leq 1)$ and $vs_0^0(ps, qs) = 0$.

$$vs_z^t(ps, qs) = \frac{\rho s \times vs_{z-1}^t(ps, qs) + Sats_{z-1}^t(ps, qs)}{2} \tag{5.1}$$

In the above, considering agent qs's service up to z transactions in the tth time epoch, $Sats_z^t(ps, qs)$ means the amount of satisfaction agent ps owns regarding agent qs. The following presents the detailed update function about the satisfaction.

$$Sats_z^t(ps, qs) = \alpha \times Sats_{cur} + (1 - \alpha) \times Sats_{z-1}^t(ps, qs) \tag{5.2}$$

α is a weight. $Sats_0^t(ps, qs) = Sats_{last}^{t-1}(ps, qs)$. Namely, the initial satisfaction value is $Sats_0^0(ps, qs) = 0$. And satisfaction value at the beginning of tth time epoch, is the same as the satisfaction value last computed in the $(t-1)$th time epoch. Moreover, the satisfaction value about the most recent transaction is $Sats_{cur}$, which is updated with a feedback system based on the following function.

$$Sats_{cur} = \begin{cases} 0, & \textit{if transaction is fully unsatisfactory;} \\ 1, & \textit{if transaction is fully satisfactory;} \\ \in (0,1), & \textit{otherwise.} \end{cases} \tag{5.3}$$

Computation of Trust Values of Data Centers

Given that up to z transactions in the tth time epoch, the historical trust which agent pd (i.e., a data center in cloud) owns regarding agent qd (i.e., another data center in cloud) is $vd_z^t(pd, qd)$. The forgetting factor (discounting older experiences) is $\varrho d (0 \leq \varrho d \leq 1)$ and $vd_0^0(pd, qd) = 0$.

$$vd_z^t(pd, qd) = \frac{\varrho d \times vd_{z-1}^t(pd, qd) + Satd_{z-1}^t(pd, qd)}{2} \tag{5.4}$$

Where considering agent qd's service up to z transactions in the tth time epoch, $Satd_z^t(pd, qd)$ means the amount of satisfaction agent pd owns regarding agent qd. The detailed satisfaction update function is presented as below.

$$Satd_z^t(pd, qd) = \beta \times Satd_{cur} + (1 - \beta) \times Satd_{z-1}^t(pd, qd) \tag{5.5}$$

Similarly, β is also a weight. $Satd_0^t(pd, qd) = Satd_{last}^{t-1}(pd, qd)$ and $Satd_0^0(pd, qd) = 0$. $Satd_{cur}$ means the satisfaction value, regarding the most recent transaction. It is updated with the same feedback system, as utilized in the previous computation of trust values of sensor nodes.

Table 5.1: Features of proposed TASCs (i.e., ITASC, CTASC and MTASC)

TASCs	Features
ITASC	Sensor nodes' trust values and trust value threshold are determined by WSN; Data center's trust values and trust value threshold are determined by cloud.
CTASC	Sensor nodes' trust values are determined by WSN; Sensor nodes' trust value threshold is determined by collaboration of WSN and cloud; Data centers' trust values are determined by cloud; Data centers' trust value threshold is determined by collaboration of WSN, cloud and users.
MTASC	Sensor nodes' trust values and trust value threshold are determined by WSN; Data center's trust values and trust value threshold are determined by cloud; There are trust values, regarding WSNs and clouds as well as users; There are mutual trust value thresholds, among WSNs and clouds as well as users.

5.3.2 ITASC, CTASC and MTASC

Characteristics about the proposed three types of TASC (i.e., ITASC, CTASC and MTASC) are summarized in Table 5.1 and the detailed information about ITASC, CTASC and MTASC are shown as follows.

ITASC

In ITASC, the sensor nodes' trust values and data centers' trust values are determined by the WSN and the cloud, independently. The trust value thresholds of sensor nodes and data centers are also chosen by the WSN and the cloud, independently. The detailed process is shown as follows.

1. The WSN obtains the trust value of each sensor node, the cloud achieves the trust value of each data center, with the trust value calculation methods discussed in Section 5.3.1 respectively.

2. In each time epoch, according to *a)* whether the transmission path can be formed in WSN, the trust

value threshold of sensor nodes is determined by the WSN. In addition, considering *b)* whether the task can be fulfilled in cloud, the trust value threshold of data centers is chosen by the cloud. After the trust value thresholds of sensor nodes and data centers are selected, the trusted sensor nodes and trusted data centers are used in the WSN and the cloud, respectively.

3. From the WSN to the cloud, the sensory data is gathered and transmitted. From the cloud to the users, the sensory data is stored, processed and further on demand delivered.

CTASC

For CTASC, the Step 1) and Step 3) of CTASC are the same as that of ITASC, except Step 2) of CTASC. Namely, regarding the trust value threshold selection in Step 2) of CTASC, WSN not only considers *a)* whether the transmission path can be formed in WSN, but it also incorporates *b)* the previous interactions with cloud resulting from various sensor nodes' trust value thresholds in the history. Similarly, the cloud takes into account both *c)* whether the task can be fulfilled in cloud and *d)* the previous interactions with WSN and users leading by different data centers' trust value thresholds in the past.

In other words, comparing CTASC with ITASC, the trust values of sensor nodes and data centers are both chosen by the WSN and the cloud independently. However, the trust value threshold of sensor nodes is determined by the collaboration of WSN and cloud in CTASC. The trust value threshold of data centers is determined by the collaboration of WSN, cloud and users in CTASC. Collaborating WSN and cloud as well as users during the trust value threshold selection procedure, is to choose more appropriate trust value thresholds, considering the previous interactions among the WSN, the cloud and the users triggered by different trust value thresholds in the history.

MTASC

In ITASC and CTASC, it is only assumed that *i)* there are trust values of the sensor nodes in the WSN and trust values of the data centers in the cloud; *ii)* there is trust value threshold about the sensor nodes in the WSN and there is trust value threshold about the data centers in the cloud. In MTASC, apart from *i)* and *ii)*, it is supposed that *iii)* there are trust values regarding the WSN (V_{WSN}), the cloud (V_{Cloud}) and the user (V_{User}); *iv)* there are mutual trust value thresholds between WSNs and clouds as well as users.

Specifically, in MTASC, sensor nodes' trust values and trust value threshold are determined by WSN. Data center's trust values and trust value threshold are determined by cloud. The trust values of the WSN (V_{WSN}), the cloud (V_{Cloud}) and the user (V_{User}), can be achieved with the trust and reputation management system (e.g., [83]). The mutual trust value thresholds among WSNs and clouds as well as users, are determined by them mutually. For example, the trust value threshold d_3 for WSN to choose cloud is determined by WSN. The trust value threshold d_4 for cloud to select WSN is chosen by the cloud. Similarly, the threshold d_5 for cloud to trust user is cloud's decision and the threshold d_6 for user to trust cloud is user's decision.

The detailed steps of MTASC are presented as below.

1. Comparing V_{WSN}s, V_{Cloud}s, V_{User}s with d_3s, d_4s, d_5s, d_6s (e.g., V_{WSN} needs to surpass d_4; V_{Cloud} needs to surpass d_3 and d_6; V_{User} needs to surpass d_5), each WSN chooses the cloud(s) it trusts. Similarly, each cloud selects the WSN(s) and the user(s) it trusts. Each user chooses the trusted cloud(s). With this process, the mutual trust between WSNs and clouds as well as users, are established. Namely, the WSNs and clouds as well as users trust each other mutually.

2. Step 1) of ITASC

3. Step 2) of ITASC

4. Step 3) of ITASC

About V_{WSN}, V_{Cloud}, V_{User}, they actually mean the confidence that WSN, cloud and user have shown to each other facing uncertainty in future transactions. Regarding the mutual trust value thresholds, d_3 and d_6 together determine whether the cloud is qualified to deal with the sensory data from WSN as well as handle the data requests from user. d_4 and d_5, determine whether the WSN and the user are trustworthy, respectively. By utilizing V_{WSN}s, V_{Cloud}s, V_{User}s and d_3s, d_4s, d_5s, d_6s, WSNs and clouds as well as users will start mutual transactions with more confidence.

5.4 Analysis of TASC and SCWTA

Regarding this section, a general comparison of TASC and SCWTA is performed first, followed with the definition about the throughput and response time of a SC. Then a path concept in SC is introduced. With that, the detailed analysis is performed about TASC and SCWTA, in terms of the throughput and response time.

5.4.1 General Comparison

1. TASC: Owning trust values surpassing a certain threshold, the trusted sensor nodes perform the sensory data gathering and transmission in WSN. Meanwhile, having trust values surpassing a certain threshold, the trusted data centers implement the sensory data storage, processing and

on demand delivery in cloud. Therefore, the probability that the sensory data could be achieved successfully from the WSN's source node to the SC's user is higher, with less data loss. The needed time for the achievement of sensory data from the WSN's source node to the SC's user is also low.

2. SCWTA: The sensor nodes and data centers are used, ignoring their trust values. As a result, the gathering and transmission of sensory data in WSN may be conducted by some sensor nodes which have very low trust values. In addition, the storage, processing and further on demand delivery of sensory data in cloud may be implemented by some data centers which have very low trust values. With such manner, it is low, in terms of the chances that the sensory data is achieved successfully by the SC's user from the WSN's source node. Also, it is low, regarding the proportion of the sensory data achieved successfully by the SC's user from the WSN's source node. In addition, it is high, about the corresponding time required for the sensory data to be achieved by the SC's user from the WSN's source node.

5.4.2 Throughput and Response Time of SC

1. Response time (RT_{SC}) of a SC is defined, as the end to end time needed to receive the sensory data by the SC's user from the WSN's source node. This includes the time needed by the sensor nodes (for performing sensory data collection and transmission) in WSN and the time required by the data centers (for performing sensory data storage, processing and delivery) in the cloud.

2. Throughput (TH_{SC}) of a SC is defined, as the end to end successfully received sensory data by the SC's user from the WSN's source node, divided by the end to end response time.

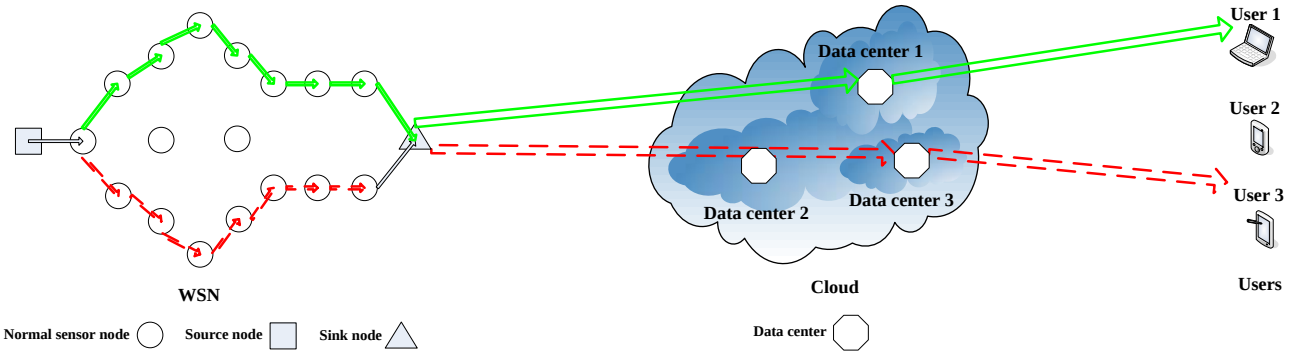


Figure 5.4: An instance about paths in SC

5.4.3 Paths in SC

Fig. 5.4 presents an instance of the paths (named as SC paths) from the WSN's source node to the SC's user in a SC. Further, it can be observed that for different SC paths, the sensor nodes' trust values and the data centers' trust values in the SC paths should be different generally. In addition, the response time and throughput utilizing different SC paths to offer sensory data to the SC's user, should be different in general.

Intuitively, it is assumed that if the sensor nodes' trust values and the data centers' trust values are higher in a SC path, then the corresponding response time using the SC path is lower and the corresponding throughput utilizing the SC path is higher. In other words, it is supposed that there is a decreasing function, which associates the RT_{SC} with the trust values of sensor nodes and data centers in the SC path. There is also an increasing function that relates the TH_{SC} to the sensor nodes' trust values and data centers' trust values in the SC path.

Note: 1) Since the selection of SC path is with respect to the routing strategy which is a well discussed another topic, the SC path selection is not discussed in this chapter. Instead, the focus is analyzing RT_{SC} and TH_{SC}. 2) The computations of the sensor nodes' trust values and the data centers's trust values, are already analyzed in Section 5.3.1. Thus in each time epoch, each sensor node's trust value and each data center's trust value, can be obtained for performing the next analysis.

5.4.4 Detailed Analysis

For SCWTA, in each time epoch, assume that in the selected SC path, there are N sensor nodes with trust values x_n ($n \leq N$) and M cloud data centers with trust values y_m ($m \leq M$). About TASC, in the chosen SC path, assume that there are C sensor nodes and D data centers. Trust values of them are a_c ($c \leq C$) and b_d ($d \leq D$), respectively.

Regarding SCWTA and TASC, generally there are different sensor nodes and data centers in the SC paths. In particular, x_n and y_m are random in SCWTA, while a_c and b_d surpass certain trust value thresholds in TASC. In addition, assume that the decreasing function that associates each sensor node's response time with each sensor node's trust value is g_1 in the SC path. Similarly, the decreasing function that relates each data center's response time to each data center's trust value is g_2 in the SC path.

With the above assumptions, the expected RT_{SC} and expected TH_{SC} are analyzed. Particularly, the focus is analyzing the expected RT_{SC}, since the expected TH_{SC} is achieved with the following equation, in which r is the data rate and ι is a parameter indicating the proportion after data loss due to various factors (e.g., interference, etc.) affecting sensory data gathering, storage, processing and transmission.

$$\mathbb{E}\{TH_{SC}\} = \frac{r \cdot \iota}{\mathbb{E}\{RT_{SC}\}} \tag{5.6}$$

In the following, for simplicity, e_1 is denoted as the RT_{SC} of SCWTA and e_2 is denoted as the RT_{SC} of TASC, followed with the analysis about $\mathbb{E}\{e_1\}$ and $\mathbb{E}\{e_2\}$ with respect to four detailed distribution-function combinations.

$e_1 = \sum_{n=1}^{N} g_1(x_n) + \sum_{m=1}^{M} g_2(y_m)$, $e_2 = \sum_{c=1}^{C} g_1(a_c) + \sum_{d=1}^{D} g_2(b_d)$. $X = [x_1, \ldots, x_N]$, $Y = [y_1, \ldots, y_M]$. $A = [a_1, \ldots, a_C]$, $B = [b_1, \ldots, b_D]$. $X, Y \in [0, 1]$, $A \in [Th_1, 1]$, $B \in [Th_2, 1]$. X, Y, A, B are independent. Besides, X and A have similar distributions while Y and B have similar distributions. The functions $g_1(x)$ and $g_2(y)$ are decreasing w.r.t. x and y, respectively. Thus, the expectations of e_1 and e_2 can be obtained as

$$\mathbb{E}\{e_1\} = N \cdot \mathbb{E}\{g_1(x)\} + M \cdot \mathbb{E}\{g_2(y)\}, \tag{5.7}$$

$$\mathbb{E}\{e_2\} = C \cdot \mathbb{E}\{g_1(a)\} + D \cdot \mathbb{E}\{g_2(b)\} \tag{5.8}$$

where $x, y \in [0, 1]$, $a \in [Th_1, 1]$, $b \in [Th_2, 1]$. Besides, x and a have similar distributions while y and b have similar distributions.

Denote $f_X(x)$, $f_Y(y)$, $f_A(a)$ and $f_B(b)$ as the probability density functions (PDFs) of x, y, a and b, respectively. Moreover, $f_A(a)$ and $f_B(b)$ are normalized as

$$f_A(a) = \frac{1}{\int_{Th_1}^{1} f_X(x)\mathrm{d}x} f_X(a), a \in [Th_1, 1], \tag{5.9}$$

and

$$f_B(b) = \frac{1}{\int_{Th_2}^{1} f_Y(y)\mathrm{d}y} f_Y(b), b \in [Th_2, 1], \tag{5.10}$$

respectively.

In terms of the functions $f_X(x)$ and $f_Y(y)$, two cases are considered as follows.

1. P1: *Uniform distribution*, i.e., $f_X(x) = 1$, $x \in [0,1]$, and $f_Y(y) = 1$, $y \in [0,1]$.

2. P2: *Normalized exponential distribution*, i.e., $f_X(x) = \frac{\lambda_1}{1-\mathrm{e}^{-\lambda_1}}\mathrm{e}^{-\lambda_1 x}$, $x \in [0,1]$, and $f_Y(y) = \frac{\lambda_2}{1-\mathrm{e}^{-\lambda_2}}\mathrm{e}^{-\lambda_2 y}$, $y \in [0,1]$. Here, λ_1 and λ_2 are positive constant parameters.

In terms of the functions $g_1(x)$ and $g_2(y)$, two cases are also considered as follows.

1. F1: *Inverse function*, i.e., $g_1(x) = \frac{k_1}{x+\varepsilon_1}$, $x \in [0,1]$, and $g_2(y) = \frac{k_2}{y+\varepsilon_2}$, $y \in [0,1]$. Here, (k_1, k_2) are positive constant parameters and $(\varepsilon_1, \varepsilon_2)$ are nonnegative constant parameters.

2. F2: *Negative exponential function*, i.e., $g_1(x) = \mathrm{e}^{-h_1 x+\theta_1}$, $x \in [0,1]$, and $g_2(y) = \mathrm{e}^{-h_2 y+\theta_2}$, $y \in [0,1]$. Here, (h_1, h_2) are positive constant parameters and (θ_1, θ_2) are nonnegative constant parameters.

Note that $\mathbb{E}\{g_1(x)\} = \int_0^1 f_X(x)g_1(x)\mathrm{d}x$, $\mathbb{E}\{g_2(y)\} = \int_0^1 f_Y(y)g_2(y)\mathrm{d}y$, $\mathbb{E}\{g_1(a)\} = \int_{Th_1}^1 f_A(a)g_1(a)\mathrm{d}a$ and $\mathbb{E}\{g_2(b)\} = \int_{Th_2}^1 f_B(b)g_2(b)\mathrm{d}b$. Thus, based on equation (5.7), equation (5.8), equation (5.9) and equation (5.10), the following four distribution-function combinations can be achieved and analyzed.

P1-F1

$f_A(a) = \frac{1}{1-Th_1}, a \in [Th_1, 1]$, and $f_B(b) = \frac{1}{1-Th_2}, b \in [Th_2, 1]$. Thus, it can be obtained that $\mathbb{E}\{e_1\} = k_1 N \ln\left(1+\frac{1}{\varepsilon_1}\right) + k_2 M \ln\left(1+\frac{1}{\varepsilon_2}\right)$ and $\mathbb{E}\{e_2\} = \frac{k_1 C}{1-Th_1}\ln\left(\frac{1+\varepsilon_1}{Th_1+\varepsilon_1}\right) + \frac{k_2 D}{1-Th_2}\ln\left(\frac{1+\varepsilon_2}{Th_2+\varepsilon_2}\right)$.

P1-F2

Likewise, $f_A(a) = \frac{1}{1-Th_1}, a \in [Th_1, 1]$, and $f_B(b) = \frac{1}{1-Th_2}, b \in [Th_2, 1]$. Thus, it can be got that $\mathbb{E}\{e_1\} = \frac{N}{h_1}\mathrm{e}^{\theta_1}(1-\mathrm{e}^{-h_1}) + \frac{M}{h_2}\mathrm{e}^{\theta_2}(1-\mathrm{e}^{-h_2})$ and $\mathbb{E}\{e_2\} = \frac{C}{h_1(1-Th_1)}\mathrm{e}^{\theta_1}(\mathrm{e}^{-h_1 Th_1} - \mathrm{e}^{-h_1}) + \frac{D}{h_2(1-Th_2)}\mathrm{e}^{\theta_2}(\mathrm{e}^{-h_2 Th_2} - \mathrm{e}^{-h_2})$.

P2-F1

$f_A(a) = \frac{\lambda_1}{\mathrm{e}^{-\lambda_1 Th_1} - \mathrm{e}^{-\lambda_1}} \mathrm{e}^{-\lambda_1 a}$, $a \in [Th_1, 1]$, and $f_B(b) = \frac{\lambda_2}{\mathrm{e}^{-\lambda_2 Th_2} - \mathrm{e}^{-\lambda_2}} \mathrm{e}^{-\lambda_2 b}$, $b \in [Th_2, 1]$. Define $Q(lo, up, \lambda, \varepsilon) = \int_{lo}^{up} \frac{\lambda}{s+\varepsilon} \mathrm{e}^{-\lambda s} \mathrm{d}s$, which can be calculated with numerical methods once $(lo, up, \lambda, \varepsilon)$ is given. Thus, it can be achieved that $\mathbb{E}\{e_1\} = \frac{k_1 N}{1-\mathrm{e}^{-\lambda_1}} Q(0, 1, \lambda_1, \varepsilon_1) + \frac{k_2 M}{1-\mathrm{e}^{-\lambda_2}} Q(0, 1, \lambda_2, \varepsilon_2)$ and $\mathbb{E}\{e_2\} = \frac{k_1 C}{\mathrm{e}^{-\lambda_1 Th_1} - \mathrm{e}^{-\lambda_1}} Q(Th_1, 1, \lambda_1, \varepsilon_1) + \frac{k_2 D}{\mathrm{e}^{-\lambda_2 Th_2} - \mathrm{e}^{-\lambda_2}} Q(Th_2, 1, \lambda_2, \varepsilon_2)$.

P2-F2

Similarly, $f_A(a) = \frac{\lambda_1}{\mathrm{e}^{-\lambda_1 Th_1} - \mathrm{e}^{-\lambda_1}} \mathrm{e}^{-\lambda_1 a}$, $a \in [Th_1, 1]$, and $f_B(b) = \frac{\lambda_2}{\mathrm{e}^{-\lambda_2 Th_2} - \mathrm{e}^{-\lambda_2}} \mathrm{e}^{-\lambda_2 b}$, $b \in [Th_2, 1]$. Thus, it can be got that $\mathbb{E}\{e_1\} = \frac{\lambda_1 N}{(\lambda_1+h_1)(1-\mathrm{e}^{-\lambda_1})} \mathrm{e}^{\theta_1} \left[1 - \mathrm{e}^{-(\lambda_1+h_1)}\right] + \frac{\lambda_2 M}{(\lambda_2+h_2)(1-\mathrm{e}^{-\lambda_2})} \mathrm{e}^{\theta_2} \left[1 - \mathrm{e}^{-(\lambda_2+h_2)}\right]$ and $\mathbb{E}\{e_2\} = \frac{\lambda_1 C}{(\lambda_1+h_1)(\mathrm{e}^{-\lambda_1 Th_1} - \mathrm{e}^{-\lambda_1})} \mathrm{e}^{\theta_1} \left[\mathrm{e}^{-(\lambda_1+h_1)Th_1} - \mathrm{e}^{-(\lambda_1+h_1)}\right] + \frac{\lambda_2 D}{(\lambda_2+h_2)(\mathrm{e}^{-\lambda_2 Th_2} - \mathrm{e}^{-\lambda_2})} \mathrm{e}^{\theta_2} \left[\mathrm{e}^{-(\lambda_2+h_2)Th_2} - \mathrm{e}^{-(\lambda_2+h_2)}\right]$.

Note: In this chapter, affecting the chances that the sensory data is received successfully and the proportion of the sensory data received successfully by the SC's user from the WSN's source node, various factors are considered. In other words, it is assumed that the end to end successfully received sensory data (i.e., $r \cdot \iota$) by the SC's user from the WSN's source node, are the same for TASC and SCWTA, in the following section that analyzes numerical results.

5.5 Numerical Results

Determining the effectiveness of TASC about enhancing the QoS that the sensory data is achieved by users from SC, TASC is in contrast to SCWTA. The throughput and response time analyzed above are utilized as the evaluation metrics. Performed in NetTopo [46], the detailed simulation is presented as below.

5.5.1 Evaluation Setup

The system includes one WSN, one cloud and 10 users. One sink node, one source node and 100 normal sensors nodes are included in the WSN, with a date rate which is 1000 kbps. The WSN transmits the sensory data to the cloud including 10 data centers. Sensory data in the cloud is further on demand requested by each user. Each time epoch is 1 s.

In general, the sensor nodes' trust values and the data centers' trust values surpass certain thresholds, in TASC. The sensor nodes' trust values and the data centers' trust values are random values ranging from 0 and 1, in SCWTA. The following two scenarios show the detailed information regarding the evaluation.

- Scenario 1: For comparing the throughput and response time of TASC and SCWTA, 100 simulations with various topologies are utilized. In these topologies, the number of sensor nodes changes from 1 to 20 and the number of data centers changes from 1 to 5 in the SC path, for both TASC and SCWTA. About TASC, both sensor nodes' trust value thresholds and data centers' trust value thresholds are set to be 0.5. In terms of SCWTA, each sensor node's trust value and each data center's trust value are always random values ranging from 0 and 1.

- Scenario 2: For analyzing trust value thresholds' impacts on throughput and response time, a specific topology in which the SC path includes 10 sensor nodes and 2 data centers is utilized, for both TASC and SCWTA. For this topology, the sensor nodes' trust value thresholds and data centers' trust value thresholds are varied 7 times (from 0.0 to 0.7) in TASC. Particularly, for each time, the trust value threshold is increased by 0.1 in TASC. Meanwhile, each sensor node's trust value and each data center's trust value are still always random values, ranging from 0 and 1 in SCWTA.

In particular, in terms of the parameters about the distributions and functions illustrated in Section 5.4.4, it is set that $(\lambda_1, \lambda_2) = (1, 3)$, $(k_1, \varepsilon_1, k_2, \varepsilon_2) = (0.5, 0.5, 0.1, 0.1)$, $(h_1, \theta_1, h_2, \theta_2) = (1, 0.3, 2, 0.1)$. In addition, the TASC to SCWTA ratios (%) on the throughput and the response time resulting from the same topology (i.e., the same distribution-function combination) are utilized to compare the performance of TASC and SCWTA, since it is more fair that the analysis is based on the throughput and response time regarding the same topology.

5.5.2 Evaluation Results

Regarding the TASC to SCWTA ratios (%) on throughput and response time in Scenario 1, Fig. 5.5(a) to Fig. 5.5(d) and Fig. 5.6(a) to Fig. 5.6(d) depict the evaluation results, respectively. Particularly, from these figures, it can be obviously achieved that in different topologies, the throughput of TASC nearly always outperforms the throughput of SCWTA a lot. In the meantime, the response time of TASC almost always substantially falls behind the response time of SCWTA.

Moreover, with respect to the TASC to SCWTA ratios (%) on throughput and response time in Scenario 2, Fig. 5.7(a) to Fig. 5.7(d) and Fig. 5.8(a) to Fig. 5.8(d) describe the evaluation results, respectively. It can be obtained from those figures that in terms of different trust value thresholds, TASC still owns larger throughput than SCWTA, while TASC still owns smaller response time than SCWTA. In particular, the TASC to SCWTA ratio (%) on throughput can be increased and the TASC to SCWTA ratio (%) on response time can be decreased, by growing the trust value thresholds in general.

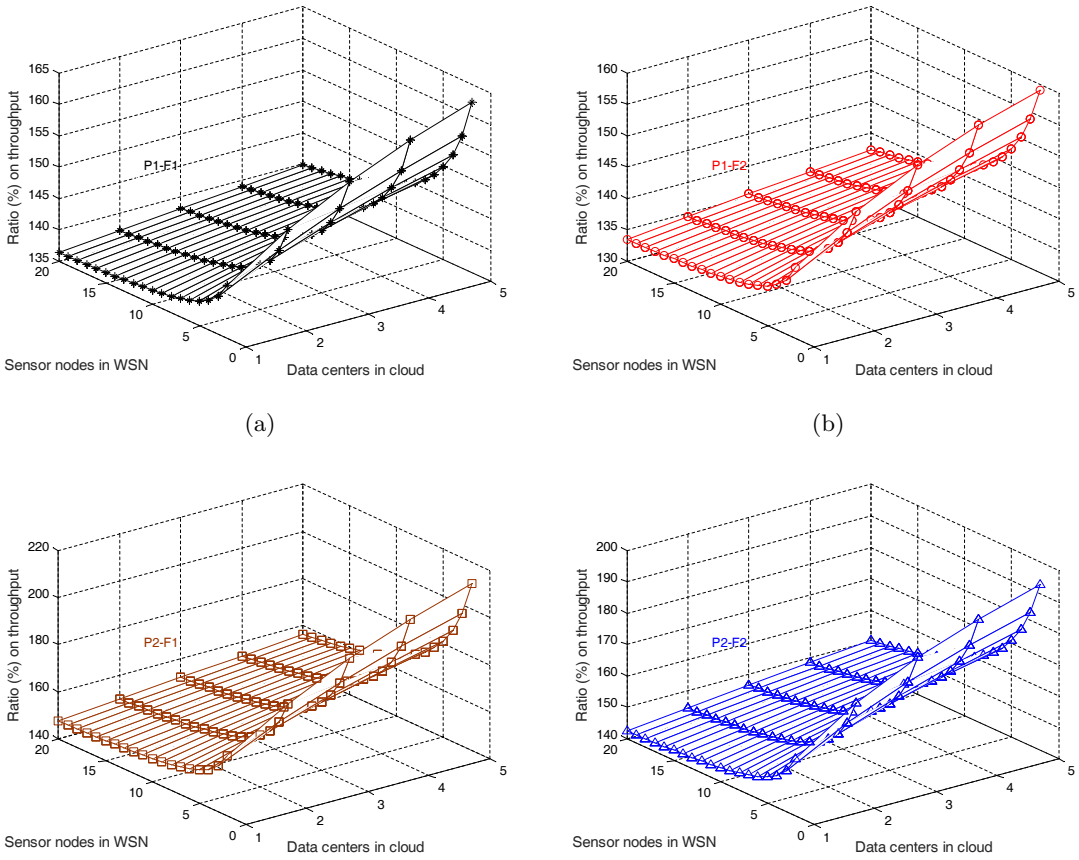


Figure 5.5: TASC to SCWTA ratio (%) on throughput in Scenario 1: P1-F1 (a), P1-F2 (b), P2-F1 (c), P2-F2 (d)

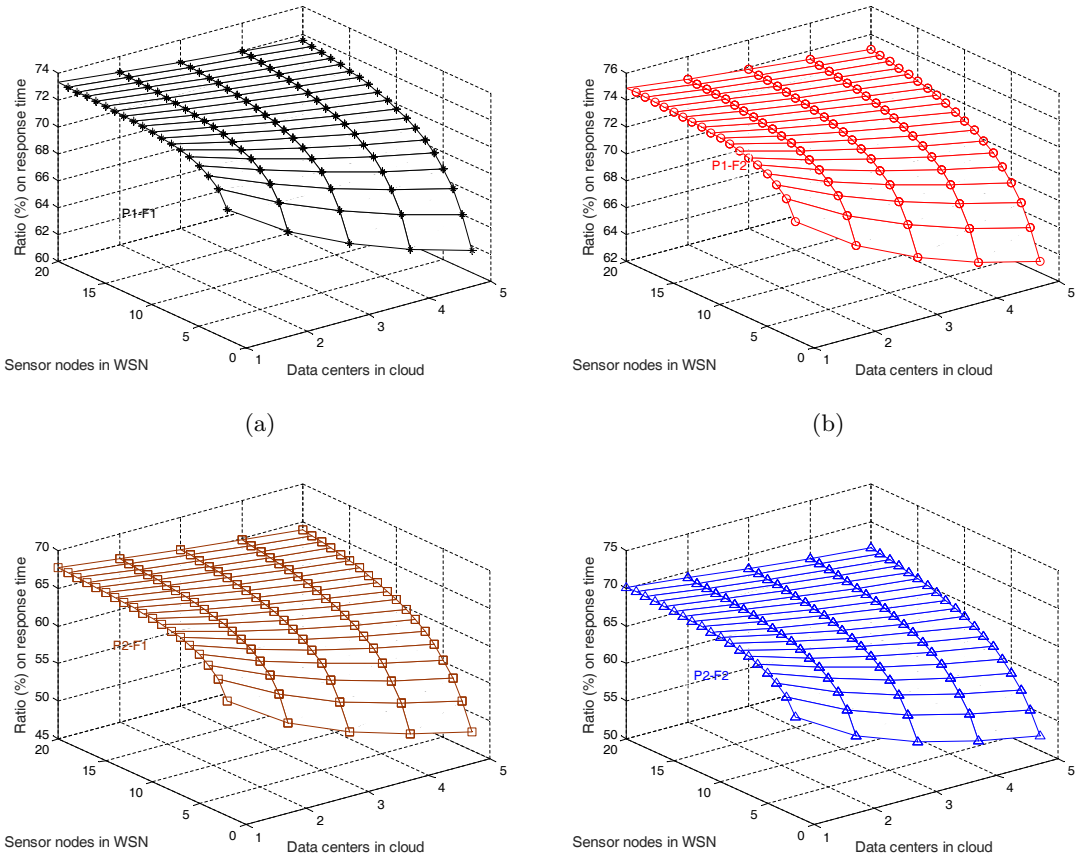


Figure 5.6: TASC to SCWTA ratio (%) on response time in Scenario 1: P1-F1 (a), P1-F2 (b), P2-F1 (c), P2-F2 (d)

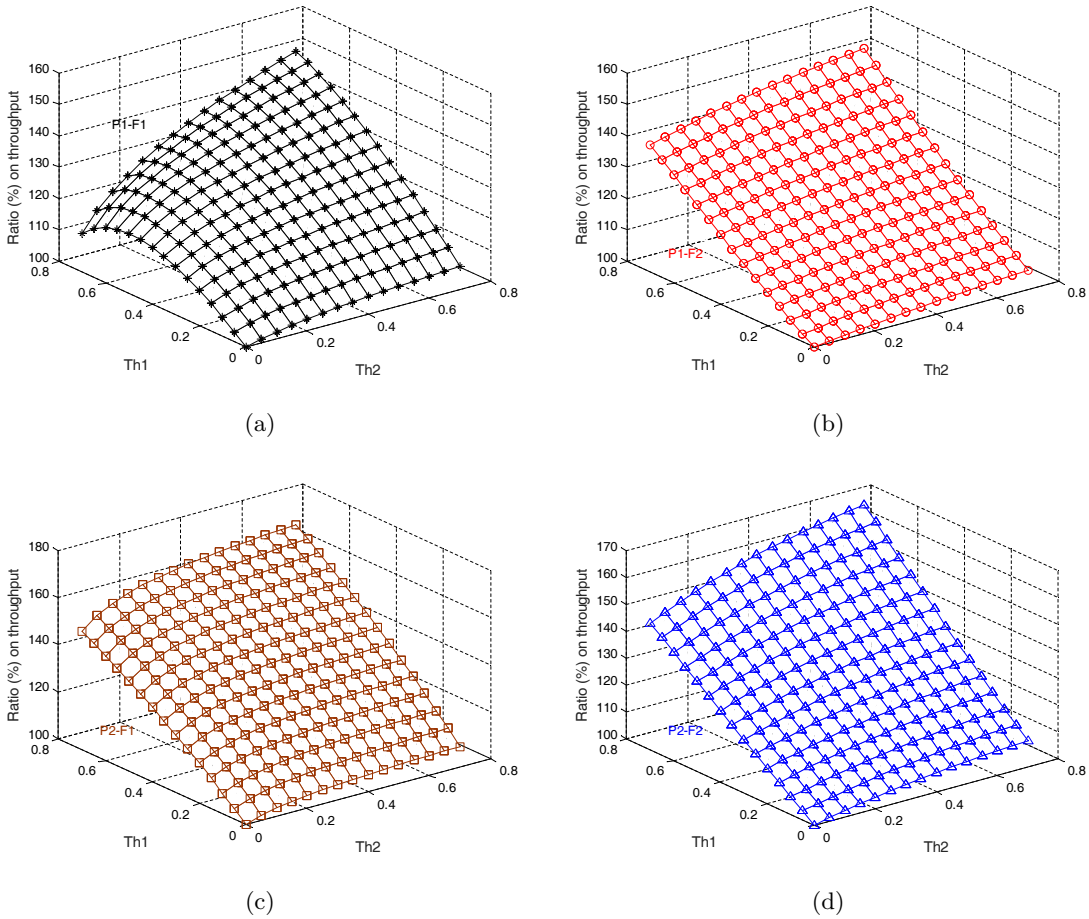


Figure 5.7: TASC to SCWTA ratio (%) on throughput in Scenario 2: P1-F1 (a), P1-F2 (b), P2-F1 (c), P2-F2 (d)

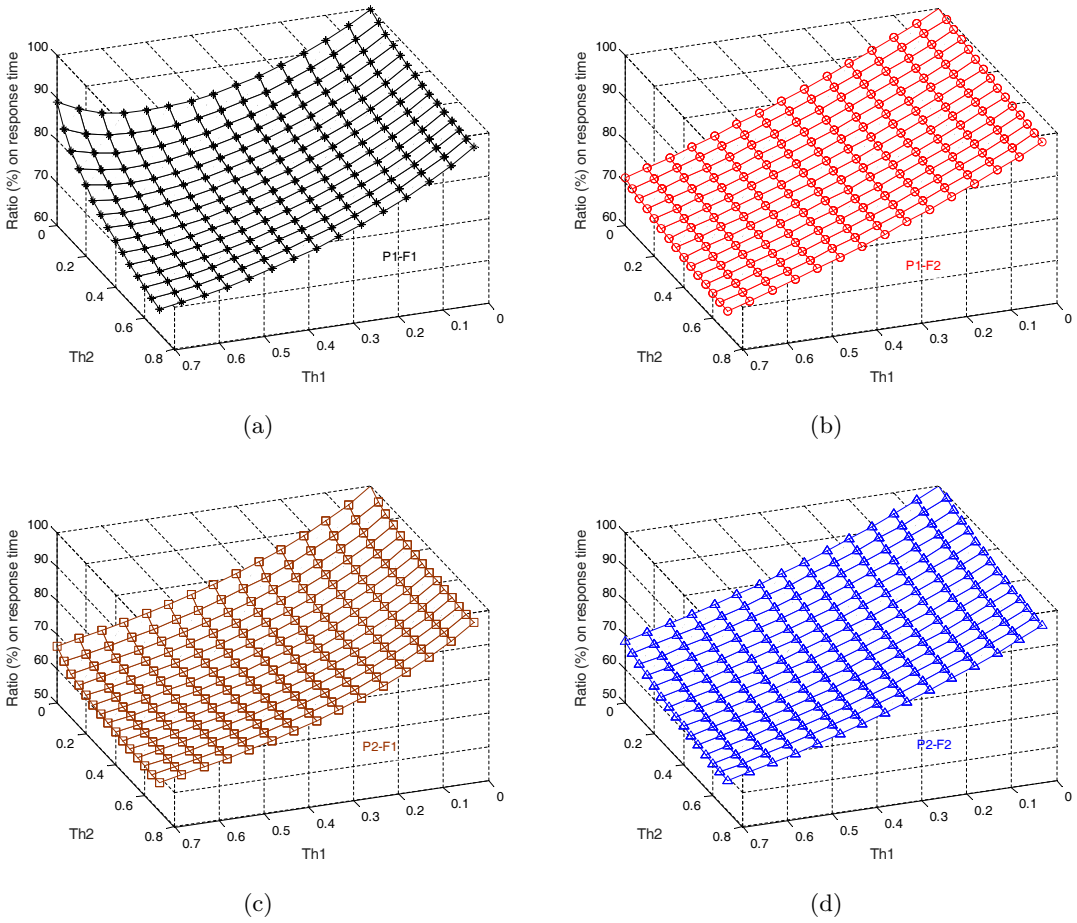


Figure 5.8: TASC to SCWTA ratio (%) on response time in Scenario 2: P1-F1 (a), P1-F2 (b), P2-F1 (c), P2-F2 (d)

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis focuses on *Sensor-Cloud*, which is a valuable and challenging topic. Particularly, *improving Sensor-Cloud*, the *important research issues that are yet to be widely investigated by other researchers* about the *energy efficiency, security, sensory data transmission and QoS* of Sensor-Cloud have been identified in this thesis. Further, regarding solving those identified research issues, our *accomplished work* has been shown.

- In Chapter 2, solving the *identified research issues about energy efficiency of Sensor-Cloud*, we have proposed two *CLSS* schemes (i.e., CLSS1 and CLSS2) for WSNs integrated with MCC. CLSS schemes involve both the WSN and the cloud and then dynamically change the awake or asleep status of the sensor node in the integrated WSN, based on the locations of mobile users. CLSS1 focuses on saving the most energy consumption of the integrated WSN and CLSS2 further pays attention to the scalability and robustness of the integrated WSN. For the integration of MCC and WSNs, both theoretical and simulation results have been shown and they have demonstrated that CLSS1 and CLSS2 could prolong the lifetime of the integrated WSN while still satisfying the data requests of mobile users.

- In Chapter 3, solving the *identified research issues about security of Sensor-Cloud*, we have proposed an ATRCM system for CC-WSN integration. Discussion and analysis about the authentication of CSP and SNP as well as the trust and reputation with respect to the service provided by CSP and SNP have been presented, followed with detailed design and functionality evaluation about the proposed *ATRCM* system. All these have presented that the proposed ATRCM system achieves the following three functions for CC-WSN integration: a) authenticating CSP and SNP to avoid malicious impersonation attacks; b) calculating and managing trust and reputation regarding the service of CSP and SNP; c) helping CSU choose desirable CSP and assisting CSP in selecting appropriate SNP, based on (i) the authenticity of CSP and SNP; (ii) the attribute requirement of CSU and CSP; (iii) the cost, trust and reputation of the service of CSP and SNP. In addition, our system security analysis powered by three adversary models has shown that our proposed system is secure versus main attacks on a trust and reputation management system, such as good mouthing, bad mouthing, collusion and white-washing attacks, which are the most important attacks in our case.

- In Chapter 4, solving the *identified research issues about sensory data transmission of Sensor-Cloud* to support WSN-MCC integration applications that need more useful data offered reliably from the WSN to the cloud, we have identified the critical issues that impede the usefulness of sensory data and reliability of WSN, and proposed a WSN-MCC integration scheme named *TPSS* to address some of these issues. Specifically, TPSS consists of the following two main parts: 1) TPSDT for WSN gateway to selectively transmit sensory data that are more useful to the cloud, considering the time and priority features of the data requested by the mobile user; 2) PSS algorithm for WSN to save energy consumption so that it can gather and transmit data more reliably. Both analytical and

experimental results regarding TPSS have been presented to demonstrate the effectiveness of TPSS in improving the usefulness of sensory data and reliability of WSN for WSN-MCC integration.

- In Chapter 5, solving the *identified research issues about QoS of Sensor-Cloud*, we have proposed the concept of *TASC* to enhance the throughput and response time that the sensory data is achieved by users from SC. In TASC, the sensory data gathering and transmission from WSN to cloud are performed by trusted sensors, while the sensory data storage and processing as well as on demand delivery from cloud to users are conducted by trusted data centers. Moreover, regarding the application of TASC, the methods to compute sensor nodes' trust values and data centers' trust values have been discussed. In addition, about the design of TASC, three types of TASC (i.e., ITASC, CTASC, MTASC) have been presented. In contrast to SCWTA, further extensive analysis and numerical results have also been shown to demonstrate the effectiveness of TASC about improving the throughput and response time that the sensory data is achieved by users from SC.

We hope our work can attract more research into Sensor-Cloud for making it develop faster and better.

6.2 Future Work

About the possible extensions of our work presented in this thesis, they are shown as follows.

- *Topology design for TASC*: In Chapter 5, we have proposed a TASC concept, which could improve the throughput and response time that the sensory data is achieved by users from SC. In particular, TASC utilizes the trusted sensors to perform the sensory data gathering and transmission from WSN to cloud. Moreover, TASC uses the trusted data centers to perform the sensory data storage

and processing as well as on demand delivery from cloud to users. Concerning the extension of this work, we can observe that TASC will be effective when there are sufficient sensors in WSN and sufficient data centers in cloud. Thus, it is worthwhile discussing the following point that could influence whether there are sufficient sensors in WSN and whether there are sufficient data centers in cloud: the design of the topology for TASC.

- *TASC with multiple trust value thresholds*: In Chapter 5, the trusted sensors in WSN are defined as the sensors which own trust values surpassing a threshold. In addition, the trusted data centers in cloud are defined as the data centers which own trust values surpassing a threshold. For the current TASC, although the trust value thresholds are dynamically determined in each time epoch, the trust value thresholds for all the sensors in the WSN are the same and the trust value thresholds for all the data centers in the cloud are the same. Considering the variant of this work, we can study the TASC with multiple trust value thresholds, in which the trust value thresholds for different sensors in the WSN are different and the trust value thresholds for different data centers in the cloud are different. New theoretical analysis might be needed and new numerical results might appear.

- *Social-Sensor-Cloud (SSC)*: We envision that the Sensor-Cloud could evolve into SSC, in which social networks (SNs) [84] [85] [86], WSN and cloud connect and complement each other, as shown in Fig. 6.1. In *Social-Cloud*, integrating SNs and CC, there is already much research (e.g., [57] [87] [88] [89]), in which the key idea is to share the cloud resources and services utilizing the relationships established between members of a SN. In SSC, leveraging SNs, not only will the Sensor-Cloud resources and services be shared, but also the SNs could be used to *improve Sensor-Cloud* in the following ways. i) Sharing the Sensor-Cloud resources and services to other users with SNs, could substantially reduce the resources and services requested by the Sensor-Cloud users. As a result,

the energy consumption of Sensor-Cloud could be decreased dramatically. ii) Based on the amount of resource consumption and service usages created by a variety of users in SNs, the deployment of resources could be optimized and the waste of resources could be reduced in Sensor-Cloud.

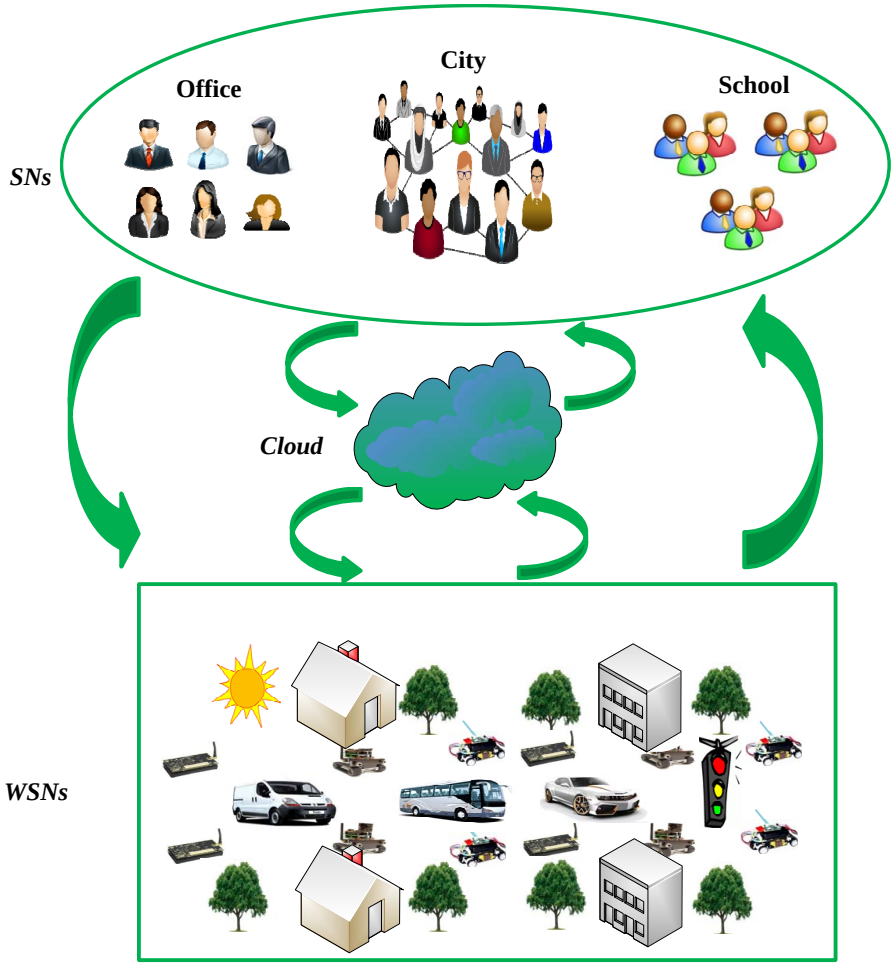


Figure 6.1: A vision of Social-Sensor-Cloud (SSC)

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Trans. Sens. Netw*, vol. 5, no. 2, pp. 1–29, Mar. 2009.

[3] 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," *Wirel. Commun. Mob. Comput.*, vol. 14, no. 1, pp. 19–36, Jan. 2014.

[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J. Ineternet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[6] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[7] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 46–53, Jun. 2013.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wirel. Commun. Mob. Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[9] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, First Quarter 2014.

[10] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: Architecture, applications, and approaches," *Int. J. Distrib. Sensor Netw.*, pp. 1–18, 2013.

[11] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor cloud: A paradigm shift from wireless sensor network," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–10, Nov. 2014.

[12] S. Madria, V. Kumar, and R. Dalvi, "Sensor cloud: A cloud of virtual sensors," *IEEE Softw.*, vol. 31, no. 2, pp. 70–77, Mar.-Apr. 2014.

[13] S. Misra, S. Bera, A. Mondal, R. Tirkey, H.-C. Chao, and S. Chattopadhyay, "Optimal gateway selection in sensor-cloud framework for health monitoring," *IET Wirel. Sens. Syst.*, vol. 4, no. 2, pp. 61–68, Jun. 2014.

[14] S. Chatterjee and S. Misra, "Target tracking using sensor-cloud: Sensor-target mapping in presence of overlapping coverage," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1435–1438, Aug. 2014.

[15] C. Zhu, X. Li, H. Ji, and V. C. M. Leung, "Towards integration of wireless sensor networks and cloud computing," in *Proc. 7th IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 2015, pp. 491–494.

[16] 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.

[17] C.-F. Lai, M. Chen, J.-S. Pan, C.-H. Youn, and H.-C. Chao, "A collaborative computing framework of cloud network and wbsn applied to fall detection and 3-d motion reconstruction," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 2, pp. 457–466, Mar. 2014.

[18] C.-R. Yi, J. Suzuki, D. H. Phan, S. Omura, and R. Hosoya, "An evolutionary game theoretic approach for configuring cloud-integrated body sensor networks," in *Proc. IEEE 13th Int. Symp. Netw. Comput. and Appl.*, 2014, pp. 277–281.

[19] L. D. P. Mendes, J. J. P. C. Rodrigues, J. Lloret, and S. Sendra, "Cross-layer dynamic admission control for cloud-based multimedia sensor networks," *IEEE Syst. J.*, vol. 8, no. 1, pp. 235–246, Mar. 2014.

[20] P. Zhang, Z. Yan, and H. Sun, "A novel architecture based on cloud computing for wireless sensor network," in *Proc. 2nd Int. Conf. Comput. Sci. Electron. Eng.*, 2013, pp. 472–475.

[21] W. Wang, K. Lee, and D. Murray, "Integrating sensors with the cloud using dynamic proxies," in *Proc. IEEE 23rd Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2012, pp. 1466–1471.

[22] Y. Li, L. Guo, C. Wu, C.-H. Lee, and Y. Guo, "Building a cloud-based platform for personal health sensor data management," in *Proc. IEEE-EMBS Int. Conf. Biomed. and Health Inform.*, 2014, pp. 223–226.

[23] D. Vouyioukas, A. Moralis, M. Sardis, D. Drakoulis, G. Labropoulos, S. Kyriazakos, and D. Dres, "Epikouros - virtualized platforms using heterogeneous sensor services in cloud computing environment," in *Proc. 4th Int. Conf. Wireless Commun., Veh. Technol., Inf. Theory and Aerospace & Electron. Syst.*, 2014, pp. 1–5.

[24] S. S. Grace and M. R. Sumalatha, "Event matching based on subscriber category in sensor cloud," in *Proc. Int. Conf. Recent Trends in Inf. Technol.*, 2014, pp. 1–5.

[25] K. Ahmed and M. Gregory, "Integrating wireless sensor networks with cloud computing," in *Proc. 7th Int. Conf. Mobile Ad-hoc Sensor Netw.*, 2011, pp. 364–366.

[26] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Secure and scalable cloud-based architecture for e-health wireless sensor networks," in *Proc. 21st Int. Conf. Comput. Commun. Netw.*, 2012, pp. 1–7.

[27] P. Stuedi, I. Mohomed, and D. Terry, "Wherestore: Location-based data storage for mobile devices interacting with the cloud," in *Proc. 1st ACM Workshop Mob. Cloud Comput. & Serv.: Soc. Netw. Beyond*, 2010, pp. 1–8.

[28] Y. Man and Y. Liu, "Towards an energy-efficient framework for location-triggered mobile application," in *Proc. Australasian Conf. Telecommun. Netw. Appl.*, 2010, pp. 3644–3647.

[29] R. Meier and V. Cahill, "On event-based middleware for location-aware mobile applications," *IEEE Trans. Softw. Eng*, vol. 36, no. 3, pp. 409–430, May-Jun. 2010.

[30] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wirel. Netw.*, vol. 11, no. 3, pp. 333–340, May 2005.

[31] C. Park, K. Lahiri, and A. Raghunathan, "Battery discharge characteristics of wireless sensor nodes: an experimental analysis," in *Proc. 2nd Annual IEEE Commun. Soc. Conf. Sens. Ad Hoc Commun. Netw.*, 2005, pp. 430–440.

[32] R. R. Rout and S. K. Ghosh, "Enhancement of lifetime using duty cycle and network coding in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 656–667, Feb. 2013.

[33] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath, "Startrack: a framework for enabling track-based applications," in *Proc. 7th Int. Conf. Mob. Syst., Appl., Serv.*, 2009, pp. 207–220.

[34] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishman, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annual Hawaii Int. Conf. Syst. Sci.*, 2000, pp. 3005–3014.

[35] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Proc. IEEE Aerospace Conf.*, 2002, pp. 1125–1130.

[36] L. Wang, Z. Yuan, L. Shu, L. Shi, and Z. Qin, "An energy-efficient ckn algorithm for duty-cycled wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, pp. 1–15, 2012.

[37] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Des. Test. Comput.*, vol. 18, no. 2, pp. 62–74, Mar./Apr. 2001.

[38] R. C. Luo and O. Chen, "Mobile sensor node deployment and asynchronous power management for wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 59, no. 5, pp. 2377–2385, May 2012.

[39] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proc. The 3rd ACM Int. Symp. Mob. Ad Hoc Netw. Comput.*, 2002, pp. 80–91.

[40] C. Zhu, L. T. Yang, L. Shu, T. Hara, and S. Nishio, "Implementing top-k query in duty-cycled wireless sensor networks," in *Proc. 7th Int. Conf. Wirel. Commun. Mob. Comput. Conf.*, 2011, pp. 553–558.

[41] W. Wang, V. Srinivasan, and K.-C. Chua, "Extending the lifetime of wireless sensor networks through mobile relays," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, pp. 1108–1120, Oct. 2008.

[42] Y. T. Hou, Y. Shi, and H. D. Sherali, "Rate allocation and network lifetime problems for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 321–334, Apr. 2008.

[43] S. Nath and P. B. Gibbons, "Communicating via fireflies: Geographic routing on duty-cycled sensors," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, 2007, pp. 440–449.

[44] C. Zhu, L. T. Yang, L. Shu, J. J. P. C. Rodrigues, and T. Hara, "A geographic routing oriented sleep scheduling algorithm in duty-cycled sensor networks," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 5473–5477.

[45] 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 Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6346–6355, Nov. 2014.

[46] L. Shu, M. Hauswirth, H.-C. Chao, M. Chen, and Y. Zhang, "Nettopo: A framework of simulation and visualization for wireless sensor networks," *Ad Hoc Netw.*, vol. 9, no. 5, pp. 799–820, Jul. 2011.

[47] C. Pelnekar, "Planning for and implementing iso 27001," *Inf. Syst. Audit Control Assoc. J.*, vol. 4, 2011.

[48] ISO, "Iso/iec 27001:2013 information technology - security techniques - information security management systems - requirements," 2013.

[49] N. Karten, "How to establish service level agreements," *published as a Book*, 2003.

[50] P. Wieder, J. M. Butler, W. Theilmann, and R. Yahyapour, "Service level agreements for cloud computing," *published as a Book*, 2011.

[51] "Privacy level agreement outline for the sale of cloud services in the european union," *published by Cloud Security Alliance*, 2013.

[52] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A survey of trust and reputation management systems in wireless communications," *Proc. IEEE*, vol. 98, no. 10, pp. 1755–1772, Oct. 2010.

[53] J.-H. Cho, A. Swami, and I.-R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 562–583, Fourth Quarter 2011.

[54] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 279–298, Second Quarter 2012.

[55] A. Das and M. M. Islam, "Securedtrust: A dynamic trust computation model for secured communication in multiagent systems," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 2, pp. 261–274, Mar.-Apr. 2012.

[56] J. M. Pujol, R. Sanguesa, and J. Delgado, "Extracting reputation in multi agent systems by means of social network topology," in *Proc. 1st Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2002, pp. 467–474.

[57] C. Zhu, H. Wang, V. C. M. Leung, L. Shu, and L. T. Yang, "An evaluation of user importance when integrating social networks and mobile cloud computing," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 2935–2940.

[58] A. Josang and R. Ismail, "The beta reputation system," in *Proc. 15th Bled Electron. Commer. Conf.*, 2002, pp. 324–337.

[59] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. Sens. Netw.*, vol. 4, no. 3, May 2008.

[60] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust aware cognitive radio architecture," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 2, pp. 86–95, Apr. 2009.

[61] W. Viriyasitavat and A. Martin, "A survey of trust in workflows and relevant contexts," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 911–940, Third Quarter 2012.

[62] "Us government cloud computing technology roadmap volume ii release 1.0 (draft)," *National Institute of Standard and Technology*, Nov. 2011.

[63] S. Reece, A. Rogers, S. Roberts, and N. R. Jennings, "Rumours and reputation: evaluating multi-dimensional trust within a decentralised reputation system," in *Proc. 6th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2007, pp. 1063–1070.

[64] G. Wang and J. Wu, "Multi-dimensional evidence-based trust management with multi-trusted paths," *Future Generation Comput. Syst.*, vol. 27, no. 5, pp. 529–538, May 2011.

[65] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[66] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–13.

[67] Y. Sun and Y. Liu, "Security of online reputation systems: The evolution of attacks and defenses," *IEEE Signal Process. Mag.*, vol. 29, no. 2, pp. 87–97, Mar. 2012.

[68] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surv.*, vol. 42, no. 1, Dec. 2009.

[69] F. Wang, J. Liu, and M. Chen, "Calms: Cloud-assisted live media streaming for globalized demands with time/region diversities," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 199–207.

[70] B. Pan, X. Wang, C.-P. Hong, and S.-D. Kim, "Amvp-cloud: A framework of adaptive mobile video streaming and user behavior oriented video pre-fetching in the clouds," in *Proc. IEEE 12th Int. Conf. Comput. Inf. Technol.*, 2012, pp. 398–405.

[71] A. M. Riad, M. Elmogy, and A. I. Shehab, "A framework for cloud p2p vod system based on user's behavior analysis," *Int. J. Comput. Appl.*, vol. 76, no. 6, pp. 20–26, Aug. 2013.

[72] F. Hu, Y. Xiao, and Q. Hao, "Congestion-aware, loss-resilient bio-monitoring sensor networking for mobile health applications," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 450–465, May 2009.

[73] W. Fang, F. Liu, F. Yang, L. Shu, and S. Nishio, "Energy-efficient cooperative communication for data transmission in wireless sensor networks," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2185–2192, Nov. 2010.

[74] 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 Syst. J.*, vol. PP, no. 99, pp. 1–12, Jan. 2014.

[75] 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 Trans. Comput.*, vol. 64, no. 7, pp. 1844–1856, Jul. 2015.

[76] W. Fang, J. Chen, L. Shu, T. Chu, and D. Qian, "Congestion avoidance, detection and alleviation in wireless sensor networks," *J. Zhejiang Univ. Sci. C*, vol. 11, no. 1, pp. 63–73, Jan. 2010.

[77] B. Sheng, Q. Li, and W. Mao, "Optimize storage placement in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 10, pp. 1437–1450, Oct. 2010.

[78] R. Arroyo-Valles, A. G. Marques, and J. Cid-Sueiro, "Optimal selective transmission under energy constraints in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 11, pp. 1524–1538, Nov. 2009.

[79] G. Battistelli, A. Benavoli, and L. Chisci, "Data-driven strategies for selective data transmission in sensor networks," in *Proc. IEEE 51st Annu. Conf. Decis. Control*, 2012, pp. 800–805.

[80] R. Arroyo-Valles, A. G. Marques, and J. Cid-Sueiro, "Optimal selective forwarding for energy saving in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 1, pp. 164–175, Jan. 2011.

[81] C. Zhu, L. T. Yang, L. Shu, T. Q. Duong, and S. Nishio, "Secured energy-aware sleep scheduling algorithm in duty-cycled sensor networks," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 1953–1957.

[82] 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 Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1317–1328, Feb. 2015.

[83] 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 Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 118–131, Jan. 2015.

[84] Y. Jiang and J. C. Jiang, "Understanding social networks from a multiagent perspective," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2743–2759, Oct. 2014.

[85] A. M. Vegni and V. Loscri, "A survey on vehicular social networks," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2397–2419, Fourth Quarter 2015.

[86] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang, "k-clique communities detection in social networks based on formal concept analysis," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–10, Jun. 2015.

[87] K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, "Social cloud computing: A vision for socially motivated resource sharing," *IEEE Trans. Services Comput.*, vol. 5, no. 4, pp. 551–563, Fourth Quarter 2012.

[88] F. Hao, G. Min, J. Chen, F. Wang, M. Lin, C. Luo, and L. T. Yang, "An optimized computational model for multi-community-cloud social collaboration," *IEEE Trans. Services Comput.*, vol. 7, no. 3, pp. 346–358, Jul.-Sept. 2014.

[89] S. Caton, C. Haas, K. Chard, K. Bubendorfer, and O. F. Rana, "A social compute cloud: Allocating and sharing infrastructure resources via social networks," *IEEE Trans. Services Comput.*, vol. 7, no. 3, pp. 359–372, Jul.-Sept. 2014.