Self-Adaptive Bandwidth Control for Balanced QoS and Energy Aware Optimization in Wireless Sensor Network

Zong Yi Liu

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *04/07/2017* **par :**

Zongyi LIU

Self-Adaptive Bandwidth Control for Balanced QoS and Energy Aware Optimization in Wireless Sensor Network

---

JURY

Helmult HLAVACS	Professeur d'Université	Rapporteur
Dominique HOUZET	Professeur d'Université	Rapporteur
Gene COOPERMAN	Professeur d'Université	Examinateur
Giogia LONGOBARDI	Chercheur	Examinateur
Daniela DRAGOMIRESCU	Professeur d'Université	Directeur de thèse
Georges DA COSTA	Maître de conférences	Co-Directeur de thèse
Thierry MONTEIL	Maître de conférences	Co-Directeur de thèse

École doctorale et spécialité :
MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Daniela DRAGOMIRESCU, *Georges DA COSTA* et *Thierry MONTEIL*
Rapporteurs :
Helmult HLAVACS et *Dominique HOUZET*

Abstract

In the Wireless Multimedia Sensor Networks (WMSNs) field, highly saturated flow increases the probability of collision and congestion in data transmission which dramatically degrade the performance of Quality of Service (QoS). Multi-channels deployment technique is often applied to parallel transmission for QoS guarantee. However, how to make trade-off between QoS requirement and energy efficiency is a challenges to energy-constrained WMSNs.

Theoretical analysis of MAC layer and PHY layer structure based on IEEE 802.15.4 standard, aim to study on the cross-layer analytical model in order to provide stronger understanding on the relationship between sensor network parameters and performance, pave the way for new enhancements in succedent multi-channel optimization research. Find effective performance indicator and design efficient performance collection or estimation approach based on the corresponding metrics, which could be used as the parameter input of multi-channel assignment mechanism.

Comprehensive dynamically control system is designed for multi-channel assignment task based on light weight and high efficient computation intelligence techniques. We present a fuzzy-based dynamic bandwidth multi-channel assignment mechanism (MCDB_FLS). Cross-layer proactive available bandwidth is estimated as parameters for multi-channel deployment admission control. Furthermore, fuzzy logic-based bandwidth threshold model provides dynamic optimization on system admission control. Simulations show the MCDB_FLS performs better than benchmark on the metrics of QoS and energy efficiency, achieves the trade-off between energy efficiency and QoS improvement.

Finally, we introduce the integration of incremental machine learning approach into multi-channel assignment mechanism with Deep Q Network reinforcement learning method (DQMC). Besides, fully action weight initialization is implemented based on multi-class supervised learning classifier with stacking ensemble approach. The results shows that deep reinforcement learning model successfully achieves a better multi-channel allocation strategy after a certain periods of self-learning process. Initial weight fusion mechanism allows us to reduce the energy expenditure due to initial environment exploration in the

early stage of learning process.

Résumé

Dans le domaine des réseaux de capteurs multimédias sans fil (WMSN), le flux fortement saturé augmente la probabilité de collision et de congestion dans la transmission de données, ce qui dégrade considérablement la performance de la qualité de service (QoS). La technique de déploiement multicanaux est souvent appliquée à la transmission en parallèle pour garantir la QoS. Cependant, comment faire le compromis entre l'exigence QoS et l'efficacité énergétique est un défi pour WMSN énergie-limité.

L'analyse théorique de la couche MAC et de la structure de la couche PHY basée sur la norme IEEE 802.15.4, vise à étudier le modèle analytique cross-layer afin de mieux comprendre la relation entre les paramètres du réseau de capteurs et la performance, ouvrant ainsi la voie à de nouvelles améliorations. Recherche d'optimisation multi-canaux.

Trouver un indicateur de performance efficace et concevoir une méthode de collecte ou d'estimation de performance efficace basée sur les métriques correspondantes, qui pourraient être utilisées comme entrée de paramètre du mécanisme d'affectation multicanaux.

Concevoir un système de contrôle dynamique complet sur la tâche d'attribution multi-canal basée sur des techniques d'intelligence de calcul léger et efficace. Dynamiquement système de contrôle de bande passante devrait faire le compromis entre la performance du réseau de qualité de service et l'efficacité énergétique.

Intégrer l'approche incrémentielle d'apprentissage automatique dans le mécanisme d'attribution dynamique de bande passante multi-canal, essayez de mettre en œuvre le système de contrôle auto-adaptatif pour apprendre le modèle de l'environnement différent dans le WMSN multi-tâche. Les résultats montrent que le modèle d'apprentissage par renforcement profond réussit à obtenir une meilleure stratégie d'allocation multicanaux après quelques périodes de processus d'auto-apprentissage. Le mécanisme de fusion pondéral initial nous permet de réduire les dépenses énergétiques en raison de l'exploration initiale de l'environnement au début du processus d'apprentissage.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisors Prof. Daniela Dragomirescu, Georges Da Costa and Thierry Monteil for their excellent guidance, assistance and patience. They provided me the creative freedom to build necessary skills and pursue research goals. These great experiences will be my precious treasure in the future life.

I would also like to express my appreciation to the other members of the Jury: Prof. Helmult Hlavacs, Dominique Houzet, Gene Cooperman and Giogia Longobardi, for their reviewing and suggestion in my thesis.

This research is funded by China Scholarship Council (CSC). I'd like to express my appreciation for the full scholarship support.

I would also like to thank all my colleague, friends from IRIT. François Thiebolt, who helped me on infrastructure and hardware setup. Hongyang Sun, Violaine Villebonnet, Leandro Fontoura Cupertino, Inés De Courchelle, Léo Grange, Wei-Tsun Sun, Chaopeng Guo, Gustavo Rostirolla , Malik Irain, Ophélie Fraisier, It's my pleasure to work with you and I have benefited a lot from your conversation and encourage. To my parents and fiancee, I would like to take the opportunity to express my feeling of gratitude for your continual support and love these years.

Finally, I'd like to thank douban.fm for being with me every coding night. The elegant UI design and great recommendation algorithm bring me a perfect peaceful atmosphere.

I dedicate my dessertation to all whom I love.

Contents

1 Introduction **1**
1.1 General introduction . . . 1
1.2 Problem statement . . . 5
1.3 Research goals . . . 7
1.4 Dssertation outline . . . 8

2 Survey of Releated Works **11**
2.1 Multi-channels techniques in WSNs . . . 11
2.1.1 Conventional multi-channel approach . . . 11
2.1.2 Cognitive radio multi-channel approach . . . 14
2.1.3 Spectrum bonding-based approach . . . 15
2.2 Artificial intelligence applications in WSNs . . . 16
2.2.1 Supervised learning-based approach . . . 17
2.2.2 Unsupervised learning-based approach . . . 22
2.2.3 Fuzzy logic-based approach . . . 25
2.2.4 Reinforcement learning-based approach . . . 26

3 Multivariate Stack Model for Cross-layer Analysis of IEEE 802.15.4 Networks **29**
3.1 Introduction . . . 30
3.1.1 Contention-Based Medium Access . . . 31
3.1.2 Probability Models for WSNs . . . 33
3.1.3 Related Works . . . 34
3.2 Stack Cross-layer IEEE 802.15.4 Model . . . 36
3.2.1 Physical Channel Propagation Model . . . 37

3.2.2 MAC layer CSMA/CA Markov Chain Model 39
3.3 Performance Analysis 43
3.3.1 Simulation Scenario 43
3.3.2 Performance Metrics 44
3.3.3 Simulation Results 45
3.4 Conclusion 51

4 Fuzzy-Based Dynamic Bandwidth Multi-Channel Assignment Mechanism for Wireless Multimedia Sensor Network 53
4.1 Introduction 54
4.2 Multi-Channel Dynamic Bandwidth Assignment 55
4.2.1 Residual Bandwidth Estimation Module 57
4.2.2 Reinforcement Learning Module for Multi-Channel Allocation 62
4.2.3 Fuzzy Logic Admission Control Module 64
4.3 Performance Simulation 68
4.3.1 Simulation schemes 68
4.3.2 Performance metrics 70
4.3.3 Simulation result 71
4.4 Conclusion 76

5 Incremental Learning with Deep Q-Network for Enhanced Multi-Channel Allocation Strategy 79
5.1 Introduction 79
5.2 Multi-channel Allocation Approach based on Deep Q-Network (DQMC) 80
5.3 Experiment of Deep Q-Network Multi-channel Allocation (DQMC) and Performance Analysis 87
5.3.1 Environment hyperparameters 87
5.3.2 Experiment and result analysis 89
5.4 Fully Initialization DQMC with Stacking Ensemble Multi-class Classifiers 94
5.4.1 Initial weight fusion of Q-function value 95
5.4.2 Multi-class classifier training procedure 96
5.4.3 Experiment and result analysis 101
5.5 Conclusion 103

6 Conclusion and Perspective 105
6.1 Conclusions 105
6.2 Contributions 108
6.3 Perspectives 110

List of Figures

1.1 Application scenarios of wireless sensor network. 2
1.2 Reference architecture of wireless multimedia sensor network. . . 4

2.1 Concepts of Channel Bonding, Channel Aggregation, and Channel Width Adaptation. 15
2.2 Supervised learning workflow diagram. 19
2.3 Architecture of two hidden layers neural network. 20
2.4 Visualization of eigenvectors in Principal component analysis (PCA). 23
2.5 Interaction visualization of reinforcement learning. 27

3.1 Workflow of carrier sense multiple access with collision avoidance algorithm. 32
3.2 stack cross-layer model structure 37
3.3 Markov chain transition states diagram for CSMA/CA algorithm in IEEE 802.15.4 39
3.4 Typical transceiver structure 41
3.5 Effective energy consumption in CSMA/CA mechanism 42
3.6 Transmission failure probability with information loads 46
3.7 Transmission failure probability with distance 46
3.8 stack cross-layer model structure 47
3.9 stack cross-layer model structure 47
3.10 Average thoughtput with information loads 48
3.11 Average delay with information loads 48
3.12 Reliability with information loads 49
3.13 Energy consumption in distances 50
3.14 Effective energy consumption with information loads 50
3.15 The average and standard deviation of critical parameters . . . 51

4.1 Estimation-based adaption of available bandwidth ratio 56
4.2 Multi-channel Dynamic Bandwidth Assignment Structure 57
4.3 IEEE 802.15.4 frame exchange sequence 59
4.4 Structure of fuzzy logic system 65

4.5 Typical procedure of rule in fuzzy logic system 67
4.6 Discrete gradient of fuzzy logic output ω^{low} 68
4.7 Instant network reliability in sequence 71
4.8 Instant network delay in sequence 72
4.9 P_{tf} and P_{bf} of different sensor nodes 73
4.10 End-to-end delay of different sensor nodes 74
4.11 Instant effective energy consumption of different schemes 75
4.12 Instant effective energy consumption of different schemes 75
4.13 Average effective energy consumption of different sensor nodes . 76

5.1 Reinforcement learning of multi-channel allocation with deep Q-network (DQMC) 81
5.2 Structure of neural network layers 84
5.3 Instant (points) and average (line) metrics of DQMC in training session, ε-greedy policy with $\varepsilon = 0.1$ for 20000 steps: (a) reward of each action step (maximum possible reward is 10), (b) number of action steps every episode, (c) loss value in training, (d) mean Q-value. 89
5.4 The average and standard deviation of critical parameters . . . 90
5.5 Rewards per action step under different policies in DQMC. . . . 91
5.6 Rewards per action step with different gradient descent algorithms in DQMC. 92
5.7 Cumulative rewards based on full tasks with different hyperparameters (memory size, policy, gradient descend method). 93
5.8 Average energy consumption per episode. 94
5.9 DQMC with initial weight fusion using multi-class classifier . . . 95
5.10 Pairplot representing feature distribution of multi-class labeled training dataset. 97
5.11 Illustration of stacking ensemble machine learning algorithm. . . 98
5.12 Classification performance with changing number of features. . . 99
5.13 Classification performance with changing size of training dataset. 99
5.14 Visualization of multiclass classification by single machine learning model and stacking ensemble model. 100
5.15 Rewards per episode in 20000 action steps 102
5.16 Action steps of each episode in 20000 action steps 102
5.17 Average energy consumption per episode. 103

List of Tables

3.1 Symbol error probability expression 38
3.2 Simulation Parameters 44

4.1 Simulation Parameters 70

5.1 Hyperparameters setting of DQMC experiment. 88
5.2 Mean Accuracy of Multilabel Classification on 10-CV training Dataset. 101

Chapter 1

Introduction

1.1 General introduction

This chapter briefly introduces the background of Wireless Snsor Network, motivation of the research, and objectives of the study. Furthermore, the fundamental objectives and main contributions are introduced respectively. Outline includes general framework of this thesis at the end of this chapter.

In the recent decades, along with the proliferation in Micro-Electro-Mechanical Systems (MEMS) technology which has motivated the development of smart sensors [1, 2], wireless communication field witnessed explosive development and attracted broad attention in the world. Wireless access networks provide various connectivity to the networks and offer different features according to the requirements of clients and applications. Wireless Access Technologies include Mesh Networks (WMNs) [3, 4], Wireless Sensor Networks (WSNs) [5, 6, 7], Mobile ad hoc Networks (MANETs) [8, 9], etc.

A typical WSN is a collection of wireless nodes with multifunctional sensor collaborate together to monitor assigned area to accomplish a sensing task for dynamically changing environment. Tracking (e.g. enemy tracking, habitat tracking, etc.) and monitoring (e.g. environmental monitoring, industrial processes automation, etc.) are two major application fields in sensor network. Figure 1.1 [10] presents a comprehensive display of application scenarios in wireless sensor networks.

Besides sensor wireless networks have the following unique characteristics and constraints, as it is stated in [11]:

- Dense Node Deployment. Several orders of magnitude in sensor nodes number construct varied density of node deployment.
- Unreliable. Sensor nodes are prone to damages or failures.

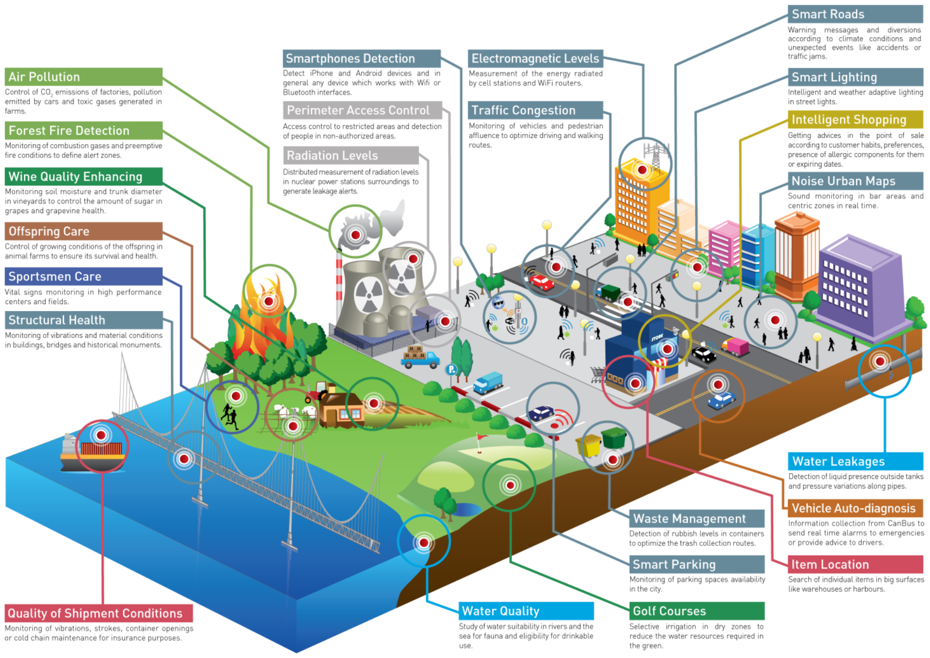

Figure 1.1: Application scenarios of wireless sensor network.

- Self-Configurable. Sensor nodes have ability to perform self-configuration in order to connect into wireless network.
- Application Specific. Special wireless network is usually designed and deployed for different applications.
- Variably Topology. Network topology changes due to additional node, damaged node, energy expenditure, or channel fading.
- Power Limited Sensor Nodes. Sensor nodes are supplied by battery with limited power. It is difficult or even impossible to change or recharge their batteries in some situations.
- Resource Limited Node. Sensor nodes have to perform tasks with limited hardware resource which has energy, computation, and storage constraints.
- No Identification. Global identification scheme is not possible for a sensor network application due to its high overhead on hardware resources and energy.
- Traffic Pattern. In most applications of sensor network, the data sensed by sensor nodes flow from multiple source sensor nodes to a particular

sink.

- Data Redundancy. Sensed data typically have a certain level of correlation or redundancy in spatially and temporally.

Researchers and industry are trying to enhance the performance of wireless sensor networks in cost, throughput rate, energy consumption, robustness, networks throughput, quality of service and security, etc. In recent years, a lot of hardware and software enhancement has been achieved to improve the performance of wireless network. A series of logical techniques have been deployed to achieve the required network performance, such as energy aware MAC layer or cross-layer design technique, efficient sensing technique, and remarkable improvement in hardware design, etc., but these techniques have their own limitations.

Recently, cognitive techniques [12] have been applied in wireless sensor networks to solve the limited performance of conventional WSNs. The cognitive technique is the process of knowing through perception, planning, reasoning, acting, and continuously updating and upgrading with the learning history. The successfully integration of cognitive radio into wireless sensors, which could solve many challenges and limitations in current conventional WSNs, as mentioned above. Cognitive radio could achieve unutilized licensed and unlicensed spectrum band, which has the ability to utilize the available spectrum with opportunity. The incumbents or primary users (PU) have the right to use the spectrum anytime, whereas secondary users (SU) can utilize the spectrum only when the PU is not using it. CR allows unlicensed users to access multiple licensed channels opportunistically. This nature of CR gives potential advantages to WSNs by improving the communication reliability and energy efficiency in high load wireless network applications.

With the development of Wireless Multimedia Sensor Network (WMSN) which is composed by embedded cameras and microphones besides scalar sensors, real-time multimedia applications require high level quality of service(QoS) guarantee in high data delivery rate flows, as shown in Figure 1.2. Increasing interference combined with the overheads of MAC protocol limit the available bandwidth in WMSN. These overheads can result in congestion which degrades QoS of WMSNs. Multimedia applications, such as on-demand or live video streaming, audio, and still images over resource constrained WSNs, are extremely challenging because of their huge bandwidth requirements [13, 14]. Other WSN applications, such as WSNs in a hospital environment, vehicular WSNs, tracking, surveillance, etc., have vast spatial and temporal variations in data density correlated with the node density. These applications are bandwidth-hungry, delay intolerable and bursty in nature. WSNs enriched

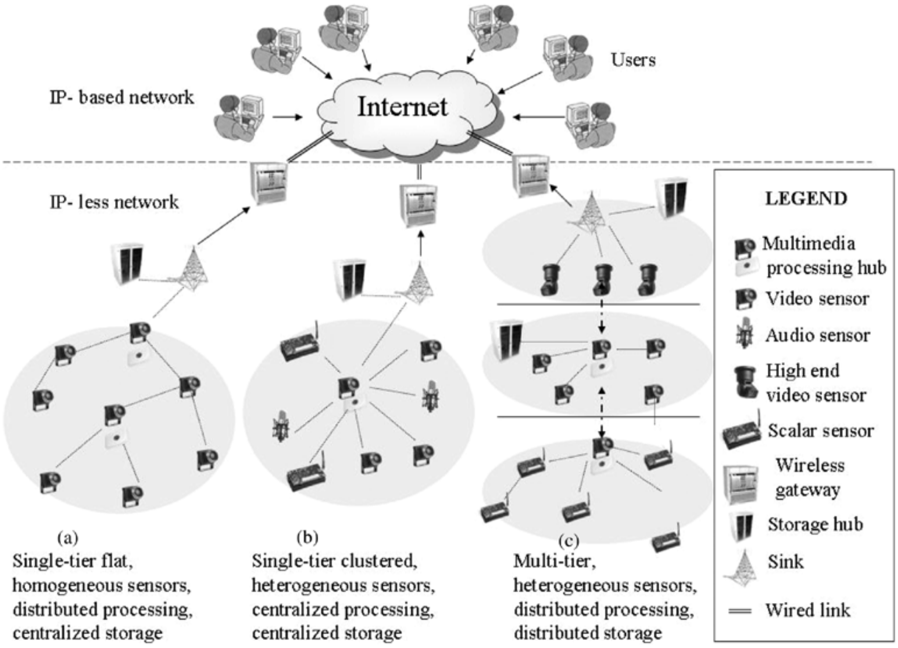

Figure 1.2: Reference architecture of wireless multimedia sensor network.

with the additional capabilities of cognitive radio has led to the evolution of cognitive radio sensor networks (CRSNs).

Cognitive Radio Sensor Networks (CRSN) enable more this kind of applications than conventional WSNs and have the potential to better solve the interference and collision issues in WSNs. A WSN usually experiences a bursty traffic once a certain event is detected; contentions and collisions cause transmission delay. The dynamic spectrum access to licensed channels in CRSN better solves this problem compared to conventional multi-channel approaches due to the limited number of orthogonal channels in the unlicensed band.

The overall attributes of wireless sensor networks has proposed a main challenge: how to make the balance between energy consumption and QoS requirements of different applications. Besides, in terms of different wireless network systems, the common challenge is try to fully exploit limited hardware resource node. This mechanism could pick out optimal patterns from uncertain environment of network, making valuable and actionable configuration to improve the performance of WSNs.

1.2 Problem statement

Conventional WSNs usually communicate on unlicensed bands, transmit small amounts of data, and have no strict restrictions on latency. Conventional WSNs are mostly suitable for low-duty cycling and monitoring applications. These applications do not have strict requirements on throughput and end-to-end delay. However, emerging WSN applications support more complex operations such as real time surveillance and target tracking and they require timely data delivery and high data rate. Once a certain event is detected, a WSN usually experiences a bursty traffic which results in contentions and collisions that limit the data throughput. The impact of interference in single channel WSNs also limits the network capacity.

By enabling transmissions over multiple channels, interference can be alleviated and collisions can be largely reduced. An efficient use of multiple channels in WSNs enables parallel transmissions over multiple channels, therefore timely communication with high data rate can be achieved. In multi-channel communication nodes may operate on different channels. An important objective is designing efficient schemes for channel assignment to ensure network connectivity and coordination between nodes, besides the performance improvement.

Besides approaches exploiting unlicensed channels, the advances in the technology of cognitive radios makes the utilization of licensed channels possible. Cognitive Radio Sensor Networks (CRSN) which employs cognitive technology into WSNs merged recently. A cognitive radio is capable of spectrum sensing, which enables it to work on both licensed and unlicensed channels. The licensed channels in the lower frequency bands have better propagation characteristics. With the same transmission power, the transmission range is larger on lower frequency. This characteristic makes cognitive radio based approaches promising for energy constrained WSNs. Extra benefit will add to the advantages brought in by the conventional multi-channel approaches.

Multi-channel approaches using ISM unlicensed bands and mechanisms for cognitive radios using licensed bands are often addressed separately. CRSNs have many advantages. However, cognitive radio hardware is more expensive than traditional transceivers. CRSN also brings challenges such as spectrum sensing and spectrum handoff. It should be noted that CRSNs are suitable for high throughput and delay-sensitive applications. It is not justified for example using cognitive radio networks for duty-cycled sensor networks intended for applications transmitting limited amount of data with no strict restrictions on the delay. The trade-offs have to be carefully analyzed before deciding whether to use cognitive radio based sensor networks.

Consequently, the challenges of multi-channel and spectrum bonding techniques [15, 16] in wireless sensor networks can be concluded as follows:

- Effective Performance Indicator:

  Received signal strength indicator (RSSI) provides information of signal quality and usually it corresponds to the distance between transmitter and receiver. However, it is possible that a node showing low RSSI can transmit at higher throughput than a node having good RSSI. As RSSI alone is not an effective indicator of network performance so other parameters are also required

- Cross layer design:

  Channel assignment is critical to the design of MAC and routing protocols. Very few works are considering all aspects. Especially in CRSNs, many proposed approaches only focus on one aspect of the performance. Cross layer designs that jointly optimize channel assignment, medium access, and routing are worth investigating.

- Energy efficiency:

  Energy efficiency is critical in WSNs since sensors are battery powered with limited energy. There is no comparison between single channel WSNs, multi-channel conventional WSNs and cognitive radio WSNs regarding to energy efficiency. Current research works lack realistic models to estimate power consumption.

- Multiple applications running simultaneously and QoS support:

  QoS is the ability to provide different priority to different applications or data flows, and guarantee a certain level of performance to a data flow. QoS should guarantee a certain bit rate, delay, jitter, and bit error rate. WSNs are mainly used for low duty cycle and monitoring applications. Recently, multi-channel and cognitive radio approaches enable various QoS demanding applications such as real-time surveillance and target tracking. It is possible to support different applications running simultaneously within the same network. Different applications may require different QoS. In multi-channel approaches, channel usage should be monitored and channels with sufficient capacity should be selected for QoS demanding applications.

- Dynamically control of multi-channel assignment

  However, cognitive radio hardware is more expensive than traditional transceivers. CRSN also brings challenges such as spectrum sensing and spectrum handoff. It should be noted that CRSNs are suitable for high throughput and delay-sensitive applications. It is not justified for example using cognitive radio networks for duty-cycled sensor networks intended for applications transmitting limited amount of data with no strict restrictions on the delay. it should be considered that multi-channel assignment should have dynamically control mechanism to make the trade-offs between Quality of service and energy efficiency.

- Self-adaptive system for multi-task wireless sensor network applications

  More flexibility and intuitiveness into the prediction process is desired by using artificial intelligence as part of system decision architecture. Specifically, the proposed model uses machine learning to perform the adaptive target classification in order to assign smart decision-making step into self-adaptive system. The added benefit of using machine learning is that the process becomes automated and adaptable to changes in the underlying mapping functions, assuming that the classifiers are periodically retrained using new empirical samples for different environment of multi-task wireless sensor network applications.

Based on the challenge in multi-channel and channel bonding techniques listed above, our work try to research some of these aspects. The cross layer performance of WNSs will be investigated under different metrics, which is prerequisite objective for the consequent research. The definition of energy efficiency and effective performance indicator are also critical parts, which allow us to make faithfully description of WSNs performance and effective optimization algorithm. Specially for the problem of resource allocation strategy, our works focus on the design of dynamically control of multi-channel assignment mechanism. Cross-layer proactive available bandwidth is estimated as parameters for channel deployment admission control. Several possible artificial intelligence techniques will be discussed in detail, aim to provide more wisely system configuration control and effective decision-making architecture. These main components of fundamental methodology and paradigm constitute the motivations and objective of this dissertation.

1.3 Research goals

These objectives in this work can be summarized as follows:

- Theoretical analysis of MAC layer and PHY layer structure based on IEEE 802.15.4 standard, aim to study on the cross-layer analytical model in order to provide stronger understanding on the relationship between sensor network parameters and performance, pave the way for new enhancements in succedent multi-channel optimization research.

- Find effective performance indicator and design efficient performance collection or estimation approach based on the corresponding metrics, which could be used as the parameter input of multi-channel assignment mechanism.

- Design comprehensive dynamically control system on multi-channel assignment task based on light weight and high efficient computation intelligence techniques. Dynamically bandwidth control system should make the trade-off between network performance of Quality of Service and energy efficiency.

- Integrate incremental machine learning approach into dynamically bandwidth multi-channel assignment mechanism, try to learn pattern from wireless network environment, which allow nodes to make smart actions on channel allocation control and achieve multi-channel communication with energy efficiency. Design self-adaptive control system using deep Q-network reinforcement learning approach which generate output multi-channel allocation commands based on the instant observation of network environments.

- Optimize the self-learning algorithm in deep Q-network reinforcement learning, try to design a rapid learning algorithm to deal with the fluctuation of performance during the initial learning stage due to the uncertain factors in WSNs.

1.4 Dssertation outline

The remaining portion of the dissertation is organized as follows. **Chapter 2** has two main sections. The first section presents the fundamental concepts of multi-channels assignment and spectrum bonding techniques in the domains of conventional wireless sensor networks (WSN) and evolutional cognitive radio sensor networks (CRSN). General state of the art which is close-related to proposed research is summarized with the discussion of advantage and weakness for each contribution. The second section presents relevant artificial intelligent algorithms which is suitable for variant applications of wireless sensor network , related literature is introduced in supervised learning, unsupervised learning, fuzzy logic system, and reinforcement learning, respectively.

Chapter 3 introduces foundmental research about performance analytical model about CSMA/CA mechanism in IEEE 802.15.4 standard. Enhanced stacking cross-layer analytical model is proposed based on the comprehensive combinations and interaction between PHY layer propagation model and MAC layer Markov chain model. Dynamic interaction between sub-layer models achieves adaptive performance estimation under the variation of systematic hyperparameters distribution. We analysis the cross-layer performance degradation from the Quality of Service (QoS) metrics and effective energy consumption metric based on CSMA/CA procedure under variable parameter sets, respectively.

Chapter 4 proposes a multi-channel assignment mechanism for WMSNs based on dynamic bandwidth control. Non-overlapping channels are dynamically allocated according to the instantaneous performance of QoS. Bandwidth is a crucial resource in WMSNs which has tight relevance with performance of QoS and energy consumption in existing studies. First, the foundmental research about residual available bandwidth estimation method will be studied, then our passive available bandwidth estimator is presented. Second, we propose a multi-channel assignment mechanism for WMSNs based on dynamic bandwidth control. Passive estimated available bandwidth is selected as metric to evaluate the global performance and considered as trigger parameters in our multi-channel deployment algorithm. For the system admission control, light weight fuzzy logic system is integrated into bandwidth threshold estimation model.

Chapter 5 is organized in four main sections. The first section presents the disadvantages of reinforcement learning algorithm based on look-up table method, discuss the motivation of deep Q-network reinforcement learning approach. In second section, reinforcement learning framework is presented to be integrated into incremental multi-channel assignment mechanism. The multi-channel allocation approach based on deep Q-network is represented in detail. Third section presents the experiments of deep Q-network multi-channel allocation approach, system performance in different metrics are presented in different iterations of Q-learning procedure under variable learning parameters. Fourth section proposes fully initialization deep Q-network multi-channel allocation model with stacking ensemble multi-class classifiers. Experiment is also discussed in this section in order to analysis the performance of enhanced learning model.

Finally, the general conclusion of this thesis is presented and research perspective is proposed.

Chapter 2

Survey of Releated Works

In this thesis, we pay more attention to the dynamically control multi-channel allocation techniques. Our works are inspired by some of these existing papers, which could be divided into two main aspects. In the first section, the existing multi-channels techniques of WSNs are introduced, which bring the methodology with several kinds of approaches. Second section presents related works of artificial intelligence techniques used in WSN applications. Data could be analyzed with precisely identification in historical datasets or new network events. The collaboration of artificial intelligence approaches and different levels of application tasks in WSNs allow nodes to propose valuable and reliable system control and application services from unexpected and uncertain network environment.

2.1 Multi-channels techniques in WSNs

Many researches have been proposed to exploit multiple channels utilization on unlicensed frequencies. The multi-channels techniques in WSNs is introduced with three subsets: conventional multi-channel approach, cognitive radio multi-channel approach and spectrum bonding-based approach.

2.1.1 Conventional multi-channel approach

MMSN [17] is one of the first multi-channel MAC protocols designed for WSNs. No specific topology is required for this approach. It aims to assign different frequencies among 2-hop neighbors for interference free data reception. In this approach four frequency assignment schemes are proposed: exclusive frequency assignment, even-selection, eavesdropping, and implicit-consensus. In the exclusive frequency assignment nodes exchange their IDs among 2-

hop neighbors through beacon messages. Frequency decisions are made in a distributed manner in increasing order of their IDs. The node with the smallest ID chooses the lowest frequency among the available ones and then beacons the choice to its 2-hop neighbors. Other nodes wait for the decisions of all the neighbors with smaller IDs and then choose the smallest frequency among those not chosen by its 2-hop neighbors. If there are not enough frequencies, then the even-selection scheme is used. When a node finds out that all the available frequencies have been chosen by neighbors, then it randomly chooses one of the least selected frequencies.

The eavesdropping scheme can be used to reduce the overhead since it does not require neighborhood information. In the eavesdropping scheme, each node selects a random backoff interval and eavesdrops its 1-hop neighbors' frequency decisions during this period. Nodes randomly choose one of the least used frequencies and broadcast their decision after a random backoff interval. The last channel assignment scheme is implicit-consensus, where all nodes share the same pseudo-random number generator. In the implicit-consensus scheme, the 2-hop neighbors' IDs are also collected. Each node calculates a random number for itself and a random number for all its 2-hop neighbors. A node chooses the current frequency only if its current random number is higher than those of its 2-hop neighbors.

Tree based multi-channel protocol (TMCP) is proposed by [18]. TMCP is designed for data collection and uses a greedy channel allocation algorithm. The whole network is partitioned into disjoint subtrees operating on different orthogonal channels rooted at the sink. The number of subtrees is equal to the number of available orthogonal channels.

The tree construction in TMCP is combined with the channel assignment. First, a fat tree rooted at sink is computed using the Breadth-First-Search algorithm. In the fat tree, nodes can have multiple parents operating on different channels and with the same minimum hop count to the sink. Channels are allocated in increasing order of the level, from the top to the bottom of the fat tree. At each level, the node with the fewest number of parents is the first to choose an optimal channel. A node will always join the subtree with the minimum interference as result of its joining. After joining a tree, the node chooses the parent with the least interference value. Using this process, multiple subtrees working on different channels are formed, thus eliminating the inter-tree interference.

HMC-MAC [19] is a multi-channel MAC protocol which reduces the interference and collisions by channel allocation and network segmentation. The network has a Network Coordinator (NC) used for central control and data collection. In HMC-MAC the time is divided into cycles, where each cycle starts with a beacon exchange period, followed by a data transmission period and an inactive period. Each node sends a beacon in a unique time slot during the beacon period, as determined by the NC. A 3-hop neighborhood needs to be discovered during the beacon period. A tree topology is built starting from the root and each branch is indexed. Nodes are organized into groups according to their depth (hops to sink) and the branch index it belongs to. Nodes at even depth and belonging to a branch with odd branch index, as well as nodes at odd depth and belonging to a branch with even branch index form the Group 1. The other nodes form the Group 2. Nodes alternate between transmission and reception mode. When Group 1 is in transmission mode, Group2 is in reception mode, and vice versa. The NC (i.e. the sink) is equipped with multiple interfaces and set in reception mode all the time.

A total of 16 channels are available for channel assignment and each node chooses dynamically its own channel. The node with the smallest network address has the highest priority in choosing a channel. Each node tries to assign a free channel that is being used by its 3-hops neighbors and sends its channel selection through a beacon. If it is not able to find a free channel among the channels that are already used by its 3-hops neighbors, then it tries to find a free channel among its 2-hops neighbors, and then among its 1-hop neighbors. If all channels are used, then the node randomly chooses a channel among those which are least used by its 1-hop neighbors.

[20] proposes a game theoretic framework for channel selection in multi-channel WSNs. The proposed framework aims to reduce the amount of overhearing by reducing the number of neighbors operating on the same channel based on a tree topology. The multi-channel allocation game is formulated as a coalition formation game. Neighboring nodes are assigned different receiving channels in the game. Non-leaf sensor nodes are considered a player set and a set of orthogonal channels form coalitions. The payoff that a player receives by joining a coalition is defined as $1/|Ck|$, where $|Ck|$ is the number of neighbors in the same coalition including the player. A balanced coalition structure is achieved when all nodes have a minimum number of neighbors in the coalition where the node belongs. Initially all sensor nodes communicate on the same channel and the communication topology is a tree rooted at the sink. A player will join another coalition if its payoff can be improved. The required knowledge is the number of neighbors assigned to each receiving channel. The coalition game ends when a balanced coalition structure is achieved. The proposed channel assignment method reduces overhearing through the multi-channel allocation game. However, this approach adds a relatively large overhead since each iteration involves a large number of actions and multiple iterations are needed to reach the equilibrium.

MC-LMAC [21] is distributed node-based multi-channel MAC protocol based on the single channel MAC protocol LMAC [22]. In this approach, all nodes are initially communicating on the same predefined base channel. A node only switches channels when the current channel is overcrowded. Initially, when all the nodes communicate on the base channel, each node selects its transmitting time slot based on the 2-hop neighbor information similar to LMAC. Each node stores the slot selection of its 2-hop neighborhood in a vector whose length is equal to the number of timeslots in a frame. Each node transmits the time slot selected and its 1-hop neighbors slot selection through a control message. On receiving a packet, a node executes the logic OR operation to update the information about the occupied slots of its neighborhood and its vector. If the node has not yet selected a transmitting slot, then it selects one from the free slots. When node receives information from its neighbors, the logic OR operation is executed and it takes the free slot as its transmitting time slot. If a node seeking a time slot finds all of them occupied, then the node becomes slotless. Such a node will try to switch channels for data transmission. In MC-LMAC, the slots occupied by the 2-hop away neighbors can be reused on a different channel. A node broadcasts the slots occupied by its 2-hop away neighbors as free slots to the neighbors. A slotless node monitors the advertised free slots and selects one node to negotiate a slot and a frequency pair. After this, both nodes are switching to the negotiation channel in the specified time slot.

2.1.2 Cognitive radio multi-channel approach

[23] proposes a cluster-based residual energy aware channel assignment scheme for multi-channel CRSN. Residual energy for each node is estimated by an R-coefficient. Channel assignment is based on the R-coefficient and it aims to balance the residual energy of each sensor. In this approach, the whole network is composed of different clusters where each cluster has a cluster head (CH). A common control channel is defined for nodes within the same cluster. Nodes exchange information on the common control channel and the CH is responsible for assigning data transmission channels for the nodes in its cluster. A frame is divided into k+1 time slots, where the first slot is reserved for channel assignment and the remaining k slots are used for data transmission. Since this work addresses the issue of channel assignment within a cluster, we still consider it as an approach for single node channel assignment.

Another single node channel assignment approach is proposed in [24]. It incorporates CR technology into WSNs and proposes an energy aware channel selection scheme. This work focuses on the channel decision between a pair of nodes in order to minimize the energy consumption. Initially nodes communi-

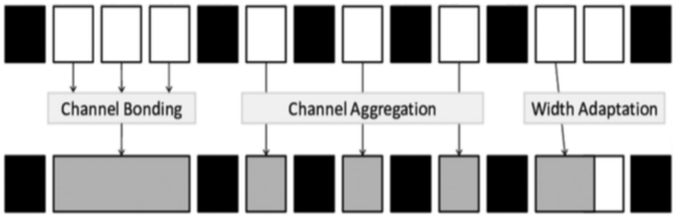

Figure 2.1: Concepts of Channel Bonding, Channel Aggregation, and Channel Width Adaptation.

cate on a common control channel. When a node S1 has to send packets to S2, S1 first transmits the number of packets to S2. S2 estimates the energy cost of each available channel, selects an optimal one, and informs S1 of the selected channel. After that both nodes tune their radios to the selected channel for data transmission. Channel occupancy is modeled using a simple semiMarkov model which is similar to the work in [23] where a channel can be either in the state idle or busy so that the average channel occupancy can be calculated. The purpose is to determine whether to stop the sensing procedure and to choose one of the channels from the already sensed channel set Sk, or to continue sensing other channels until it finds a better one.

2.1.3 Spectrum bonding-based approach

In the context of cognitive radio (CR) based networks, dynamic spectrum access (DSA) has turned out to be a promising approach for communication in those bands where radio spectrum is already overcrowded [25]. This approach has already shown a positive impact on the power consumption levels, network life time and the interference faced by its member nodes. DSA can be implemented easily in CRNs as CR nodes can change their operating parameters in order to dynamically tune into a free channel. The benefits of CB in CRNs include larger bandwidth, less complexity and higher channel capacity for the equal amount of transmission power [26].

Experimental measurements show that spectrum sharing (under defined power restrictions) can offer significant frequency reuse opportunities [27] and go smoothly without creating any harmful interference or congestion problems hence contiguous channels can be combined using CB to improve spectrum utilization.

CB is a technique to combine set of contiguous non-overlapping channels to make a bond of large bandwidth. CB and channel aggregation (CA) are two different techniques which have different requirements and prerequisites. CB needs to be applied on contiguous channels as shown in Figure 2.1 whereas channel aggregation does not need contiguous channels. CB is more beneficent if we want to increase the system capacity. Shannon's channel capacity formula gives channel capacity which is directly proportional to bandwidth.

2.2 Artificial intelligence applications in WSNs

From the first introduction of learning program sketch in 1950's [28], machine learning is proposed as the subset of artificial intelligence techniques, is used to build enhance computer models which autonomously learning processes from data and information. With the development in last decades, machine learning methods are widely used in different kinds of tasks including classification, regression and recommendation, etc. Practical machine learning application have been exploited in various area of cancer prognosis, fault detection, individual recommendation, stock prediction and system self-control, etc.

In the dynamic environment of wireless sensor networks, nodes monitor and gather varied sensed data which following specified system decision scheme. For the rapidly changing environment, the most challenging task is to develop an efficient algorithm that can meet the requirement of different wireless sensor network application scenarios simultaneously. Recent years, to address this problem, many researchs [29] aim to integrate machine learning techniques into wireless sensor network for the maximum resource utilization of diverse network environment and self optimization in network life extension. The main reason that machine learning is important technique for WSN applications can be interpreted as follow:

- The environment of sensor networks usually dynamically change and fluctuate. In this kind of uncertain circumstance, traditional system models with static and pre-programmed rules is difficult to keep good performance. Machine learning techniques could be adopted to make adaptive system decision for dynamic environment.

- In some specific environments, complicated system behaviors are difficult to be described with single mathematical models. High dimensionality of hyperparameter make such problems difficult to be solved at hand. The preprocessing analysis such as Principal component analysis (PCA) [30, 31] or Linear Discriminant Analysis (LDA) could identify patterns to reduce dataset dimensions with minimal loss of information. The high-variance features are extracted onto smaller subspace which can be used in predictive models and exploratory sensor data analysis and aggregation task.

- For exploratory WSN applications, the target of monitoring environment might be dangerous or unaccessible. sensor nodes are deployed without initialized model optimization due to the lack of collected data and information for unknown sensor environment. Several machine learning algorithms allow sensor nodes to build robust system decisions that could tolerate noisy environment variable and acquire knowledge from unexpected environment during monitoring tasks.

- In the area of Internet of things (IoT) [32] and cyber physical systems (CPS), high level platforms based on WSN techniques need more smart system-decision and real-time reliable performance with the requirement of quality of experience. For analyzing data immediately from the precisely identification in previously collected knowledge or new client behaviors that system never actually met, different levels of application tasks are performed depending on the collaboration with machine learning to find and optimize patterns in order to propose reliable and flexible application services.

2.2.1 Supervised learning-based approach

The primary function of machine learning is to automate the learning process from data samples. The fundamental approach typically used in machine learning is the inductive form of learning [33]. Induction refers to the classical type of inference where a generalization is obtained from a set of observations. The generalized model proposes the dependencies or describes an approximation function between the inputs and output. The future values can be predicted using the approximation function which is the general intent of inducing a generalized learning model, and this process follows the classical inference mechanism of deduction. Prediction of model is valuable in many scenarios. Some situations are considered when the measurement system is expensive to calculate the output relative to the input, or when the task is to use input values to make proactive adjustments and control the output values of system. The advantage of machine learning is that it can extract general trends or patterns in the observed data. This ability is attractive because in many cases, the system environment is complex and cannot be analyzed and described. In this case, machine learning provides an alternative way to describe system behavior through an approximation function which is from the learning of observation. An example of a complex environment is the wireless channel. It is a complex system involving multiple dynamics, especially when any antenna is mobile. Therefore, analytical descriptions of wireless environments are often not feasible [34]. However, through machine learning, a set of empirical samples consisting of input and output variables can be collected and then provided to a learning algorithm. Then, using the training data, the model can systematically find the appropriate model or approximate potential dependencies between variables in statistical or probabilistic sense. It should be noted that for dynamic systems such as wireless channels, learning is a continuous process and the approximated function should evolve with the system. In addition, while most of the machine learning processes can be automated, manual intervention is still needed which include domain knowledge, modeling assumptions, or modeling parameters optimization for the best performance [35].

The typical workflow of supervised learning is introduced in Figure 2.2. Initially, we have original dataset with training example. Training dataset should be representative of real-world use of function, but with unknown target function $f : X \to Y$. Original data should be processed with prerequisite data clearing and feature engineering. The performance of learned function depends strongly on how the input object is represented. The preprocessing generate input distribution $P(x) : x_1, x_2, \ldots, x_n$ which is transformed into input feature vectors with target value: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Feature engineering process should be cautious, too high dimensional features increase the difficulty for machine learning algorithm to gather prediction strategies from sparse and dissimilar data features efficiently. Insufficient data features don't contain enough information or additional noise which will lead to underfitting or bias of prediction model. Prediction model established on high correlation features will learn too much information from similar features. Excessive weight is assigned by learning algorithm result in excessively complicated model which has the risk of overfitting. Then learning algorithm A [36] is determined based on different machine learning tasks and characters of training examples. In supervised learning, the performance of hypothesis prediction $g(x)$ is evaluated in each learning iteration on special test subset of training examples. Parameters in hypothesis set H could be adjusted according to the validation performance from test subset, try to optimize performance of prediction model.

Support Vector Machines

Support vector machines (SVMs) is a set of supervised learning methods to find a decision boundary that maximum separate distance between different classes. SVM algorithm builds model that assigns new examples different categories as a non-probabilistic binary linear classifier. In high dimensional space, hyperplane or a set of hyperplanes is built to achieve separation with largest functional margin between classes of training data. Mapping original training data space into much higher dimensional space make separation easier, but also brings large computational cost [37]. In order to reduce the expenditure, higher dimensional

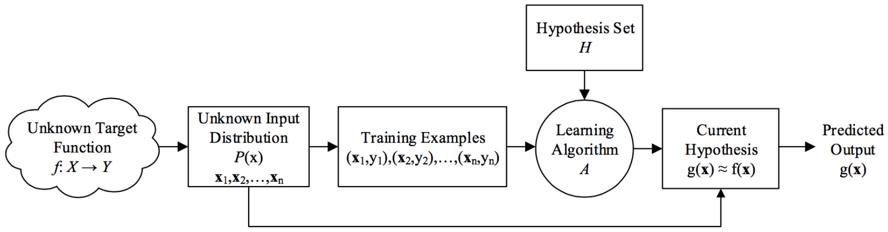

Figure 2.2: Supervised learning workflow diagram.

hyperplane could be find more efficiently through a similarity function. Dot products could be replaced with a $kernelfunction$ where high performing kernels are processed without huge and potentially infinite dimensional feature vector. Gaussian radial basis function (RBF) kernel [37] and Polynomial kernel [38] are two kernel functions which are commonly used in classification tasks of SVMs.

MAPPLE [39] introduces a supervised learning method to learn link quality estimation of wireless network. Online protocol is applied to collect link quality measurements by broadcasting $probemessages$ with different rates to generate the expectation of network traffic conditions. Supervised learning framework $SupportVectorRegression$ (SVR) is built to process offline the collected data to make the prediction model for link quality estimation. Training data set consists of 8 attribute selected features with labels of measured link quality, are non-linearly mapped into higher dimensional feature space. k-fold cross-validation (CV) splits training data set into k subsets in order to avoid overfitting during training model. The best parameter set of error penalty parameter C, Gaussian RBF kernel parameter σ, and loss function parameter ϵ, is optimized by the evaluation process of $GridSearch$ method.

Decision Tree

Authors in [40] present the Self-Adapting MAC layer (SAML) that switch MAC protocols dynamically from a Reconfigurable MAC Architecture (RMA) to gain the desired characteristics to meet the requirement of application. MAC selection engine is selected based on decision tree supervised model, which is most suitable protocol for given application QoS requirement, current traffic pattern and ambient interference levels. The features of training data consist of application specified REL order, measurement value of mean and variance of Inter-packet Interval (IPI) and Received Signal Strength (RSS) in different sliding window size, respectively.

Neural Networks

From the initial finding [41] in biological neural systems, synaptic connections are built between biological neurons and logic gates with binary outputs. Neurons can be understood as the subunits of a neural network in biological brain, accumulated input signal exceeds certain threshold that will generate passed output signal. These prerequisite works inspired basic concept of artificial neuron network: cascading chains of decision units used to learn the values of weight w then multiplied with input features in order to recognize nonlinear and complex function [42]. The basic step of perceptron neuron network can be described as follow:

- m-dimensional input values $x_i, i = 1, 2, \ldots, m$ and bias are combined with corresponding initial weight vector w;
- Network input function makes aggregation processing to compute the summation of input single for activation function.
- Output value is calculated from multiple inputs, signal is generated if value exceed threshold of activation functions $g(z)$. Typical activation functions include unit step, linear, sigmoid, tanh, Rectified Linear Unit (ReLU), etc.
- Cost function is defined based on different learning tasks. The optimization algorithm (e.g., Gradient descent) generates feedback to update weight vector w in order to minimize cost function.

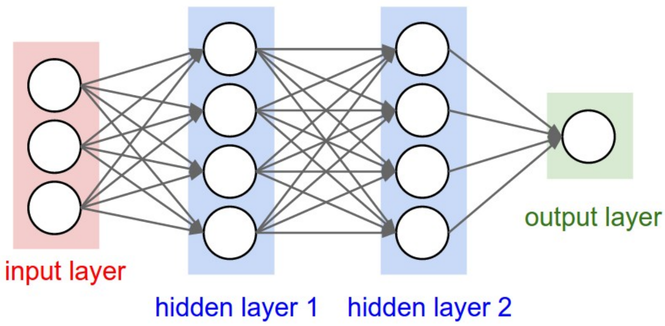

Figure 2.3: Architecture of two hidden layers neural network.

The work flow of two hidden layers neural network is illustrated by Figure 2.3. From the online approximator structures, neural networks have been effectively applied for the identification and control of system-dependent nonlinear structures in wireless sensor networks. [43, 44].

The learning methods of neural network which map the input layer and output layer could be classified into supervised learning, unsuperivesd learning and reinforcement learning. For the implementation of neural network with supervised learning-based approach, [45] proposes a two-layer neural network regression model to approximate dynamics conditions of healthy sensor nodes and estimate the possible failure function with a fault output threshold value based on the output estimation error in current instant instance. A modified NN model [46] is presented based on the recurrent neural network[47] (RNN) for the identification and fault detection of sensor nodes. The interconnections and interaction between neighbor sensor nodes construct the methodology of building estimation model with backpropagation-based neural network structure. Input space is selected with the combination of previous output samples of modeling sensor node and the current and previous output samples of neighboring sensor nodes.

In unsupervised learning applications, neural networks aims to research patterns with only a set of unlabeled training examples. Neural networks try to learn better representations of the input space, which is used in the features extraction, meta-features engineering of observed data [48, 49, 50]. Fuzzy Hopfield neural network (FHNN) [51] is presented to solve transmission problem of TDMA broadcasting scheduling in wireless sensor networks. Each time slot and node are considered as data sample and cluster respectively, time slots are distributed into the nodes of network with interference constraints. fuzzy c-means clustering is interpreted into unsupervised two dimensional fuzzy Hopfield neural network in order to find minimize frame length with transmission scheduling and avoid potential transmission collisions. [52] integrates Hopfield neural network and data transformation techniques together to solve travelling salesman problem (TSP). Data statistical techniques such as Z-score transformation [53] and base-10 logarithmic transformation are integrated into neural networks in order to achieve optimal tours with less total distances.

In reinforcement learning algorithm, agent try to communicate with environment and learn to find optimized *policy* which could select the best action with highest long terms rewards from environment. For the $Q(s, a)$ function of Q-learning, state-action space is too large to store in look-up table, neural network is constructed in Q-learning [54, 55] to find maximum Q values $max(s', a')$ for every possible action in the new state.

Logistic regression

[56] proposes data-driven link quality estimator which consist of three steps: data collection, offline modeling, and online prediction. For the step of data collection, four attributes of input data features are considered: Packet Recep-

tion Rate (PRR), Received Signal Strength Indicator (RSSI), Signal-to-Noise Ratio (SNR), and Link Quality Indicator (LQI). PRR is combined each feature of PHY parameters as ($PKT = [PRR, RSSI/SNR/LQI]$) by window mean estimator with exponentially weighted moving average. The offline modeling step includes model training and selection which three classification algorithms are considered: Naive Bayes classifier, Logistic Regression, and Neural Networks. Logistic Regression results in the fastest training speed and best binary classification performance on test data set. In order to avoid the correlation of PRR and physical parameters changing with time due to the variations of hardware specification, aggregated data from different nodes are collected into one data set as the input vectors of single training model.

K-nearest neighbor

K-nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

For the classification task of wireless sensing event on resource constraints nodes, [57] offer a resource efficiency event classification approach with k-nearest neighbor search on condensed kd-tree algorithm. Dynamic neighborhood mechanism is used to evaluate the likelihood of a node belonging to a class, magnitude of each class is calculated by taking the summation of neighboring nodes within the class. In the condensed kd-tree, each new node is merged with an existing node if they are in the same $condensingradius$ in order to reduce the size of tree in the acceptable level of performance degradation.

2.2.2 Unsupervised learning-based approach

Unsupervised learning-based algorithm is used in data aggregation, data extraction, data selection, sensor nodes clustering and adaptive system control in wireless sensor network tasks [58]. For unsupervised learning approach, training sets don't provide precise labels, models are trained to find the structure and relationship from different feature characteristic of inputs. The typical unsupervised learning model aims to find pattern to divide data set into different groups that keep the similarity of features in each subspace.

Principal component analysis (PCA) could extract latent variables that indicates dominant variance from redundant features, which is an important approach for dimensionality reduction and data cleaning of projective clustering problems [59]. From covariance matrix on normalized features, eigenvectors and eigenvalues are calculated to represent direction of component and amount of variance explained by the component in order to find top eigenvectors for low-dimensional subspace. The visualization of eigenvectors are plotted on 3-dimensional sample data space, as shown in Figure 2.4.

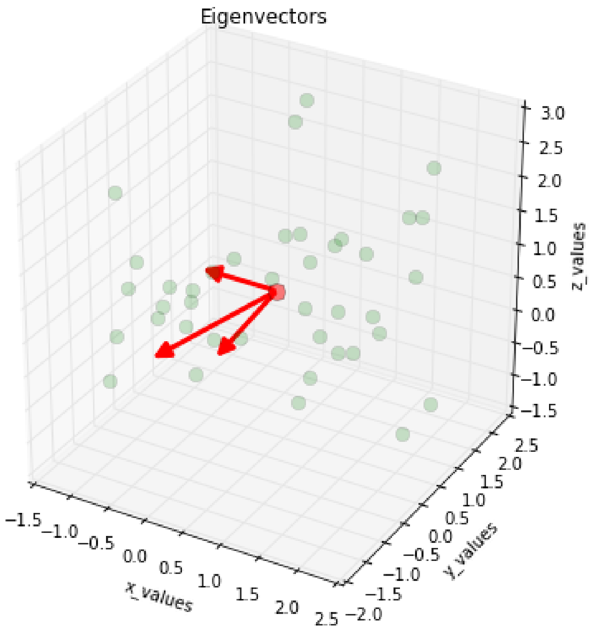

Figure 2.4: Visualization of eigenvectors in Principal component analysis (PCA).

In [60], based on excited distributed approaches of principal component analysis to reduce communication cost, authors propose consensus-based algorithms CB-EM-DPCA to achieve global performance with only local communications between neighbors. expectation maximization (EM) algorithm is used to estimate latent principal subspace with maximum probability of dominating the data observation.

[61] proposes distribute adaptive covariance matrix eigenvector estimation algorithm (DACMEE) to estimate eigenvectors corresponding to the Q largest or smallest eigenvalues of the network features covariance matrix. Instead of the communication cost for transmitting raw sensor data observations and computational overhead of EVD in large dimensional matrix, DACMEE only transmits fused Q-dimensional observations with compression matrix.

[62] introduces a feature extraction and feature selection framework for performance characterization in multi-hop wireless sensor networks by compressing original feature vectors and eliminating redundant network attributes. For the collection of network measurements, the performance of link $i \rightarrow j$ is

translated into packet reception ratio PRR_{ij} and classified into category with feature binning. Different available network metrics from cross layers generate set of multi features and form normalized feature vector f_{ij} for each link by z-score. Mix of supervised and unsupervised learning approachs are collaborated for feature selection process. Principal component analysis (PCA) is used to eliminate redundant features and generate top-ranked feature m^*, which could decrease the feature matrix size. Besides, k-nearest neighbor (k-NN) is applied to cluster the features that indicate high entropy between feature m^* and its kth nearest neighbor. High redundancy features are eliminated after several iteration of k-NN process. Accordingly, feature extraction and selection process identify patterns to integrate network performance into compressed feature vector and reduce the dimensions of network feature vector with minimal loss of network attribute informations.

K-means [63] is another unsupervised learning algorithm which is usually used in data clustering tasks. The main algorithm of K-means is try to find centroid positions for each cluster of data in euclidian space. K-means clustering partition the m observations $(x^{(1)}, \dots, x^{(m)})$ into K sets, which aims to minimize the sum of distance functions of each point in the cluster ti the K center. The optimized object function of K-means clustering algorithm can be expressed as follow:

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2 \tag{2.1}$$

After randomly initialization of K cluster centroid $(\mu_1, \dots, \mu_K)$ for m observations, K-means algorithm can be basically divided into two step:

- Cluster assignment: Assign cluster index $(1, \dots, K)$ to $c^{(i)}$ with cluster centroid closest to $x^{(i)}$.
- Move centroid: Calculate the average value of points assigned to cluster $(1, \dots, K)$, then update centroid position $(\mu_1, \dots, \mu_k)$.

These two steps of K-means are repeated until the convergence of objective function. The prototype of centralized and distributed K-means clustering algorithm are implemented in [64]. In the initialization of clustering, k group of centroids are selected at random locations. Euclidian distance between each node and all centroids are calculated, nodes are assigned to the group which has nearest distance with centroid, respectively. Recalculate centroid position and distance change with previous iteration, then repeat the previous clustering step until convergence. Based on the character of network hard partition by using k-means clustering algorithm, [65] proposes Fuzzy C-Means algorithms that sensor nodes are grouped into clusters with the degree of each cluster rather than the hard partition clustering mode. Sensor nodes near the boundary of cluster has probability to be selected as the member of another neighbor cluster which has similar degree.

2.2.3 Fuzzy logic-based approach

Fuzzy logic system (FLS) is a machine intelligence technique based on multivariate truth values. Fuzzy logic agent works by mapping crisp values into fuzzy linguistic variable, numeric data limited to input variable range is mapped into another output value of single variable. For the applications in WSNs, fuzzy control system has natural advantage that no completed mathematical model or formulation is required for the dynamic systems of network which is difficult to be formulated precisely with equation [66], the relationship between system output and control inputs is built with fuzzy logic function instead of completed numeric model.

Low energy adaptive clustering hierarchy (LEACH) [67] is the early well-known hierarchical routing protocol. The CH is elected in rotation basis on the basis of probabilistic value to balance the resource overhead on each sensor node. Only selected CHs have permission to communicate with base station (BS). Based on the typical LEACH protocol, [68] proposes improved protocol LEACH-FL with fuzzy logic approach that consider battery level, distance and node density into consideration. LEACH protocol only select cluster heads based on the probability model without considering the network density and location of nodes. In LEACH-FL, high battery level and node density increase the probability that node is selected to be CH, increasing distance between node and BS result in the output that node has decreasing probability to be selected as CH. In the Energy-aware distributed dynamic clustering protocol (ECPF) [69], residual energy of nodes is the primary parameter for tentative CHs election. Then, fuzzy logic is employed to evaluate the cost of nodes for determining the final CH from the tentative CHs. Every node select the connection with CH from neighbors which has the least fuzzy cost value. [70] consider one super-CH (SCH) among CHs which can send data to BS with the utilization of bandwidth efficiency, SCH is selected from CHs base on fuzzy rules then collects aggregated data from CHs and send data to BS. Remaining battery power, mobility and centrality are considered as three fuzzy input variables for the selection of tentative Super Cluster Head.

SUIT [71] provides fuzzy logic-based congestion detection estimation and congestion mitigation technique by decreaseing frame quality to an acceptable level. In the buffer management design of transport layer, three metrics (current hop count, average delay and frame index) are considered to calculate the overall score for packet prioritization processing. Current hop count value is integrated into packet header. The packet which has high transmission hop count value until it reach sink has higher drop-out probability than the packet has low hop count value from source to sink. Based on the range of minimum transmitting hop value to maximum hop value defined by network topology, possible hop count value is divided into four subspace with interval points $h1$, $h2$, and $h3$, The score of weight W_h is calculated from different subspace. For the congestion detection, The metrics used in FLS control input set contains ratio of incoming to outgoing packets of sliding window, number of contenders and buffer occupancy of next-hop node. $LOW(L)$, $MEDIUM(M)$, $HIGH(H)$ are defined as three linguistic variables which indicate the congestion level of system. In fuzzy inference system, the weight of fuzzy classes for three linguistic variables are set respectively where rule evaluation method (REM) calculates the output of rule with it and generate consequence fuzzy class c. Instead of using centroid calculation for defuzzification processing which has relative computational cost for sensor node device, isosceles triangle is defined to calculate membership function of each fuzzy class in order to decrease the computational complexcity level.

In [72], a fuzzy logic-based routing algorithm is proposed to realize energy optimized routing decision with multiparameter. Social welfare function is used to predict the unbalance of residual energy in neighbors after determining next hop nodes. fuzzy logic system consists of three input fuzzy variables: degree of closeness of node to the shortest path (DCSP) and degree of closeness of node to Sink (DCS) represent the observation of energy efficiency for selecting the next hop node, degree of energy balance (DEB) indicates the energy balance of routing decision. The output fuzzy variable space indicates the precise election chance of next hop nodes which is separated into seven internal with different custom output functions.

2.2.4 Reinforcement learning-based approach

In recently decade, reinforcement learning-based techniques are used in self-control system decision tasks for WSNs applications [29], such as smart MAC protocol [73, 74, 75], probability routing [76, 77], data aggregation [78, 79] and event detection [80, 81, 82], etc. Reinforcement learning allows sensor to play the role of agent, learning procedures are performed by the interaction with environment, as shown in Figure 2.5[83]. In each iteration, agent perceives network environment state s_t and performs action a_t by different action selection algorithms. Agent updates its beneficial value from the immediate rewards $r(a_t, s_t)$ of each action after every iteration. The final object of agent is self-learning to take the best system-decision that maximize its long-range time expected rewards from special wireless network tasks.

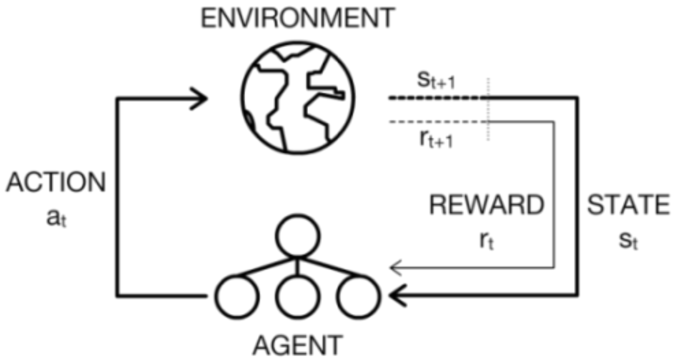

Figure 2.5: Interaction visualization of reinforcement learning.

Q-Probabilistic Routing (Q-PR) [77] proposes dynamic geographic opportunistic-based routing algorithm that makes intelligent routing decisions from custom rewards from previous routing experiments and local interaction with neighbour nodes. The custom cost metrics are estimated by the expected number of retransmissions (ETX) to destination divided by the progress in the hop, which build the mapping between performance of energy consumption, latency, link quality, transmit distance and routing policy. The decision of forwarding message from observation vector is followed by Bayesian decision model, makes the trade-off between transmission energy and message importance.

[84] studies reinforcement learning application for self-organizing wake-up scheduling on three different network topologies. Each node stores Q-value for each slot within its data frame which indicates the system feedback benifit from the node staying awake previous slots. The initialization Q-values are selected from uniform random distribution between 0 and 1 for each slot of agents. In every iteration, sensor node will select the awake time from the consecutive time slots that have the most beneficial Q-values. Smooth exploratory strategy is deployed to make sure that each random initiated slot could access to exploration and update Q-value based on the existed greedy policy. If the summation of Q-values in the slots is much larger than others slots, the reinforcement learning policy is considered to be the state of convergence.

In [85], authors presents non-deterministic Q-learning approach to manage the system decision of sensor networks with extracted cross layer information. In the proposed method cross-layer collaborative communication (CL-CC), sensor nodes combine the extracted information from MAC layer interactions with the spatial and temporal correlation of observations and optimize the schedule of Active and Sleep states. Non-deterministic Q-learning method is used to optimize energy consumption in the way of self-learning, the objective of Q-learning is to learn the best policy π which aims to maximize the expected sum of rewards from performing selected acitons. The immediate reward r_t is calculated from the estimated conditional entropy $H(X_t|X_{1:t-1})$ divided by the total energy consumption by sensor node S_i. However, author didn't consider that classical Q-learning is designed based on neural network model which also increase computational overhead and additional energy consumption for sensor nodes.

ALOHA-Q [86] combines frame based Slotted ALOHA and Q-Learning algorithm to avoid collisions and retransmissions. The receiver make the decision which slots to listen based on the extracted information from transmitters. Transmitters carry the information that indicate number of future frames that transmitters will use in current slot i which is calculated from Q-function. Ping packet is proposed to provide correct transmission information to receiver in order to avoid unnecessary energy consumption caused by idle listening and over hearing under low traffic mode. [87] optimizes the exploration and exploitation approach to improve ALOHA-Q for dynamic environment condition. In the ϵ-*greedy* policy, constant value of ϵ is not suitable for different conditions of network environment. Aloha-Q-EPS method dynamically control the probability of exploration which is calculated from current Q-value to provide adequate level of exploration. After convergence, ϵ-*greedy* policy follows the strategy with the value of $Q_{convergence}$ which is decided by network environment conditions.

RMCA [88] proposed regret matching based channel assignment algorithm to exploit parallel channel transmission. Three metrics Packet Delivery Ratio (PDR), Valid Receiving Ratio (VRR) and Average packet Transfer Delay (ATD) are considered as the indication of network interference. Each node in network is considered to perform modified regret matching procedure for solving the channel assignment problem from non-overlapping channels set $C = \{1, 2, \ldots, c\}$. The object of regret matching algorithm is making the trade off between ATD minimization and VRR maximization. Based on the probability $p_i^t(x)$ that sensor node i selects channel x in iteration t, estimated average regret R is computed for self-optimization on channel assignment strategy of next stage. After the achievement of stable performance in channel assignment, sensor nodes stop regret matching based optimization procedure.

Chapter 3

Multivariate Stack Model for Cross-layer Analysis of IEEE 802.15.4 Networks

In order to design efficient cross-layer optimization mechanism and self-adaptive control cognitive system, it's critical to deeply analyze the wireless sensor network protocol. Because of the specifications in low-cost, low-power IEEE 802.15.4 wireless sensor networks, comprehensive analytical model is important for evaluating the performance under varying wireless channel constraints. The systematic properties of single physical layer and medium access control (MAC) layer protocol have been studied through the techniques based on mathematical models or experiment-based approachs. However, It is insufficient to evaluate network performance on the basis of existing single layer model or cross-layer model with stationary parameters, especially for the multivariable parameters-based wireless network environment.

In this chapter, we propose an enhanced stack cross-layer analytical model based on the comprehensive combination and interaction between PHY layer propagation model and MAC layer Markov chain model. Dynamic interaction between sub-layer models achieve adaptive performance estimation with hyper-parameters sets. Cross-layer performance degradation is analyzed under the varying inputs of multi-parameters vectors, several Quality of Service (QoS) metrics and effective energy consumption metric are proposed and evaluated, respectively. From the simulation results compared with benchmark models, the stack cross-layer model offers the most comprehensive performance analysis with different cross-layer parameters sets which include distance, transmit power, noise power, and information loads, etc.

3.1 Introduction

In this chapter, we pay the attention to the performance analytical model for IEEE 802.15.4 wireless sensor networks. IEEE Std 802.15.4 defines the physical layer (PHY) and medium access control (MAC) sublayer specifications [89] for low-data-rate wireless communication with limited battery consumption devices.

Due to the different applications and environment constraints in wireless sensor network, performance analytical model is an important study for the evaluation and estimation of Quality of Service (QoS) and energy consumption. Several researchs use the empirical-based approach [90, 91, 92, 93] to investigate the behavior and performance of networks. Some experiments indicate different performance results of wireless network from analytical simulation model, which shows us clearly the reason behind the phenomenon of network performance. Apart from empirical-based approach, several studies [94, 95, 96, 97, 98] focus on the analytical model based on the Markov chain model. Analytical studies aim to develop generalized mechanisms with multivariable functions which is able to track the network performance with key indicators (such as throughput, reliability, delay, etc.). In the existing methods of cross-layer analytical model [97, 95, 98], performance metrics of each sub-layer are calculated independently. Joint model [98] consider additional influence of physical channel constraints, which is combined with Markov MAC layer model in order to reproduce synthetic performance analysis.

However, the joint model in [98] only calculate a constant value of packet reception rate in PHY layer model where interpreted into Markov chain model of CSMA/CA mechanism. Due to the uncertainty features in WSNs, different application tasks and varying environment parameters generate fluctuating performance requirement. A performance analysis should be considered under the sets of multivariate hyperparameter. Consequently, we built a dynamic stack cross-layer analytical model based on IEEE 802.15.4 CSMA/CA mechanism which combined PHY layer propagation model and MAC layer Markov chain model in a sufficient way. The interactional PHY and MAC layer models share multi-dimensional systematic parameters from multivariate input vectors (include information load, transmit distance, transmit power, SNR, etc.). Cross-layer overhead increases along with the changing values of parameters input space, which cause the packet transmission error on physical channel. Dynamic PHY channel constraints impact on CSMA/CA mechanism will result in further global performance degradation. Stack cross-layer model aims to fully integrates PHY layer channel constraints and MAC layer CSMA overhead into a combined analytical model, which is available to predict the variance of QoS performance with multi-dimensional environment parameters. Besides, energy consumption evaluation metric should be elaborated based on instantaneous QoS performance in network. Effective energy consumption represents the expected energy expenditure for each successfully transmitted bit, which indicates the overall energy conversion efficiency.

3.1.1 Contention-Based Medium Access

Wireless sensor network is deployed to perform sensing tasks and communication without intervention of human for a long time. MAC layer protocol is designed to appropriately manage and control the node's behavior to communicate with other sensor node avoiding collision and congestion. Several reservation-based protocols are designed based on code division multiple access (CDMA), time division multiple access (TDMA), or frequency division multiple access (FDMA) techniques to pursue less network collision. Besides, many MAC layer protocols for WSNs are implemented based on contention-based medium access techniques. Carrier sense multiple access with collision avoidance (CSMA/CA) is one of the most popular contention-based mechanisms which have flexible ability for varying network size and topology. Slots are allocated on nodes based on the contention mechanism, each node should listen to the channel to get the states that channel is busy or idle before transmit packet. If the idle state of channel is detected based on the requirement of channel clear assessment, the node will acquire permission to send packet. On the contrary, the node should wait a random time to avoid contention channel then continue listening until the channel is available. A node will discard the packet if it reaches the threshold value of limited maximum attemps which is predifined in CSMA mechanism. The basic workflow of CSMA/CA mechanism using slotted or unslotted mode is illustrated by Figure 3.1.

IEEE 802.15.4 CSMA/CA mechanism is used for each data frame transmission or MAC control packet in contention access period (CAP), which is divided into two types of channel access mechanism:

- Slotted CSMA/CA algorithm, the back-off period boundaries for each node is aligned with beacon transmission in the CAP of superframe. PHY layer begin all the transmission task on the boundary of the back-off period.
- Unslotted CSMA/CA algorithm is used on no-beacon mode networks. The period of back-off procedure of each node has no correlation in time with back-off period of other nodes.

The algorithm of CSMA/CA mechanism is implemented based on the basic time unit Back-off Period (BP), which is usually defined as $aUnitBackoffPeriod =$

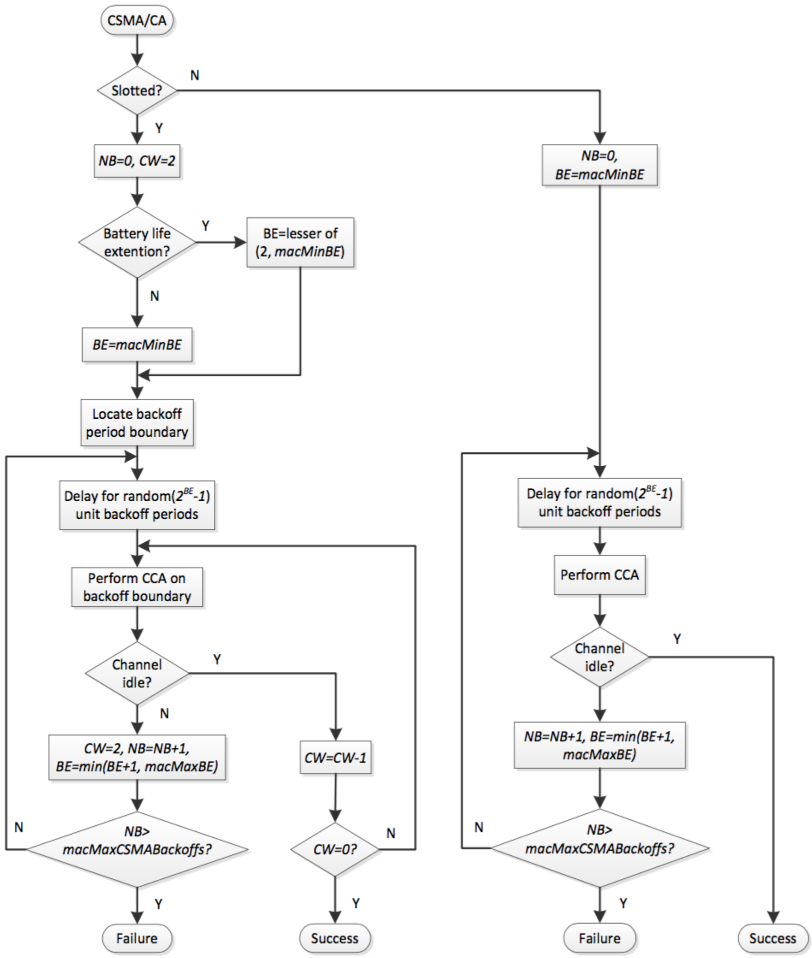

Figure 3.1: Workflow of carrier sense multiple access with collision avoidance algorithm.

20*symbols*. NB is initialized to 0 and it records the number of backoff procedures for the current transmission attemp. Backoff exponent (BE) defines the number of backoff periods each node should wait for the next assessment in the channel or clear channel assessment. Random number of backoff periods is selected in the range of $(0 : 2^{BE} - 1)$. The contention window size (CW) is defined as the number of consecutive backoff periods a channel should be detected as clear before transmission is permitted. Before each transmission attemp, CW should be initialized as constant value (Default $CW = 2$), which means the node has to conduct 2 carrier sense mechanisms named clear channel assessment (CCA) at the backoff period boundary to avoid potential collisions of acknowledgement frames. Sensor node should sense wireless medium as the status of idle during both of the two CCAs. Otherwise, the node will cancel the packet transmission and perform backoff period again. macCSMABackoffs is defined as the maximum number of backoff attemps, CSMA/CA will clarify a channel assess failure if NB is greater than macCSMABackoffs, which is usually predifined from 0 to 5.

3.1.2 Probability Models for WSNs

In order to mimic the behavior of wireless sensor network and corresponding change of performance, probabilistic models are applied into performance analytical model which usually involves integrating a complex, multi-dimensional probability distribution. Fot instance, It should be a complex integration of different factors to calculate the expectation of channel congestion state. This is difficult to calculate due to the high dimensionality of model distribution and is hard to find closed-form expression for the integral available using calculus. Markov Chain Monte Carlo (MCMC) is usually used as the fundamental method that allow us to approximate complex pattern in WSNs system using stochastic sample routines.

A Markov chain [99] is a stochastic process that operates sequentially (e.g. temporally), transitioning from one state to another within an allowed set of *states*, $S = \{s_1, s_2, ..., s_r\}$. The process starts in one of these states and moves successively from one state to another with *step*. If Markov chain in currently in state s_i, it has probability of p_{ij} to move to state s_j. Markov processes are stochastic processes that have the property that the next value of the process depends on the current value, but it is conditionally independent of the previous values of the stochastic process. This probability does not depend on which states the chain was in before the current state, which we call it finite state-space Markov chain. The probabilities p_{ij} are called transition probabilities, is defined by Equation 3.1. The process can remain in the current state, and this occurs with probability p_{ii}. Initial probability distribution is deployed with a particular state as the starting state, the chain will run for a long time $t \to \infty$ until reach an equilibrium which is the chain's stationary distribution.

$$p_{ij} = p(X^{(t+1)} = j | x^{(t)} = i) \tag{3.1}$$

Transition matrix P is used to calculate transition probabilities, probability p_{ij} is given as the i^{th} row and j^{th} column. The total transition probability from state i to other states or remain in current state must be 1, as shown in Equation 3.2.

$$\sum_{j=1}^{r} p_{ij} = 1 \tag{3.2}$$

$$p_{ij}^{(2)} = \sum_{k=1}^{r} p_{ik} p_{kj} \tag{3.3}$$

For considering two steps of transition probabilities from i to j is denoted as $p_{ij}^{(2)}$. In transition matrix operation (Equation 3.3), the i^{th} row of P dot with the j^{th} column of P for r states Markov chain. The computation in Markov chain allows analytical model to describe the basic functionalities of IEEE 802.15.4 CSMA/CA and study the behavior of each node under idle and saturated traffic channel conditions.

3.1.3 Related Works

Some related researches try to reproduce IEEE 802.15.4 standard performance and achieve further optimization mechanism. On the one hand, Markov chain model is frequently used in the systematic performance analysis on CSMA/CA MAC layer protocol. [94] firstly proposes Markov-based analytical model which mimic the performance of slotted CSMA/CA mechanism. The generalized analysis allows to measure reliability, delay and energy consumption by a Markov chain, depending on the collision probability in unsaturated traffic network. [96] analyses the performance of CSMA/CA mechanism in non-ACK mode by modified Markov chain model. In [100], authors provide analytical model through event chains computation approach. It only considers chains with a probability to occur greater than pre-defined threshold to reduce complexity. [101] introduces an analytical MAC layer model which tune parameters $macMinBE$ and $macMaxCSMABackoffs$ in order to improve the trade-off between sampling frequency and application requirement. Markov model with k states is proposed in [95] to account for varying changing conditions under LR-WPAN lossy channel, error rate and frame sequences correlation are derived between transmitter and receiver.

On the other hand, several studies focus on PHY layer model in order to reproduce the performance influence of physical channel constraints. [102] quantifies the impact of PHY channel constraints and hardware variance on unreliable and asymmetric links and generates expectation and variance of packet reception rate through given transmission region boundary. PHY layer transmission model [103] is built based on the degradation of AWGN channel and block Rayleigh fading channel. Expected energy cost for each successfully received bit is considered as metrics for energy consumption analysis. Numerical parameters about transmission power, transmit hoping distance with different modulation scheme are optimized in order to find the energy consumption minimization of packet transmission.

Besides, several works make the combination of sub-layer models with the aim to faithfully mimic the cross-layer functionalities. [97] proposes a cross-layer model based on the integrated MAC and PHY layer models, which consider the impact of multi-path shadow fading channels on the network performance. A joint layer model is presented in [98], which make the combination of two relevant models from PHY and MAC layer. Transmission error on PHY layer is first calculated then additional estimation is integrated into Markov MAC layer model. However, the PHY layer model only consider transmission error as a static estimation value from independent computation. How to develop cross-layer dynamic tuning model with multivariable distribution function is the object of improvement.

Consequently, it is important to analyze the system performance wireless sensor network in a comprehensive way, which allows us to fully understand the network structure and precisely estimate the network performance before trying to develop pertinent optimization and improvement algorithms for specific problems. The challenge is:

- **Cross layer interactions** : In the existing performance evaluation model, many proposed approaches only focus on one aspect of the different layers. The interaction between sub-layer cannot be neglected. This could influence the network performance additionally. For instance, the transmission error due to PHY channel constraints will aggravate the performance degradation in MAC layer because of the CCA and retransmission attemps in CSMA/CA.

- **Energy consumption estimation** : For a basic evaluation method of the energy consumption in network, overall energy expenditure is considered based on elementary energy consumption, which is calculated from different states of network mechanism or experimental results in each hardware component. It is important to investigate advanced energy overhead metrics that indicate the effective conversion of overall energy consumption. In this way it is possible to evaluate the performance level both for QoS and energy efficiency.

- **Multivariable parameters** : In the real environment of wireless sensor network, multiple dimensional factors and features will impact on the network performance dynamically. The performance variances distribution should be depicted under multivariable parameters input space. This could generate directly analytical results with more accurate network information.

3.2 Stack Cross-layer IEEE 802.15.4 Model

Our research is based on [98] which proposed analytical joint model over PHY and MAC layer. It combined two relevant models together, by considering with the impact of PHY layer error on Markov chain model based on MAC layer, in order to elaborate the performance description. However, the PHY layer model did not consider environment parameters as variables, output of PHY layer packet transmission error probability p_e is computed as a static value to be joined into MAC layer model along with the different information load. Thus, we propose a stack cross-layer IEEE 802.15.4 analytical model, that aims to integrate channel parameter variables into submodel of each layer to obtain a more precise performance estimatio. The estimation of each sublayer model and fully interaction between combined sublayer models allow us to analyze more widely performance in different states of sensor network environment.

Initially, the assumptions of network environment are made for proposed stack cross-layer model:

- Each node perform synchronization at backoff period without sleep interval-based mechanism.
- One hop star topology network that consist of certain number of contending nodes.
- The probability to start sensing the channel and packet sending probability is assumed to be independent.

The structure of stack cross-layer model workflow is illustrated in Figure 3.2. Only information load Φ is considered as variable input in existing joint model. On the basis of that, multivariate vectors are considered as input space with multi-dimensional parameters, such as propagation distance d, transmit power P_t and noise power N_0, etc. The multi-interactions between submodels generate the combined performance analysis model to obtain dynamic performance response in the case of network environment with multivariable parameters. Transmission error on PHY layer is calculated dynamically then integrated into additional estimation of performance degradation in MAC layer Markov chain model. Simulation stochastic process follows M/M/1/K queueing model [98] as convergence control mechanism. Queue process run follows Poisson arrival process for each receives frame until Markov chain process reaches an equilibrium which converge the stationary state probability.

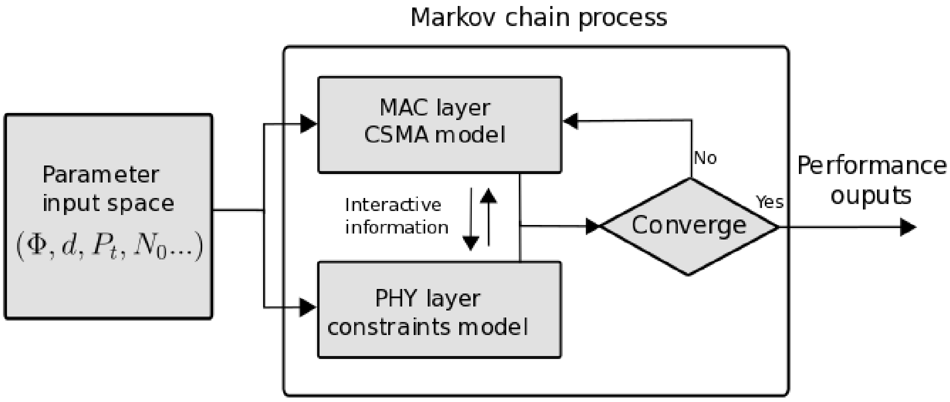

Figure 3.2: stack cross-layer model structure

3.2.1 Physical Channel Propagation Model

Firstly, we analyse the transmission failure probability due to PHY layer channel constraints. The physical layer transmission model is developed based on the approaches which are derived from [102, 103, 104]. Propagation distance, transmit power, fixed circuit energy consumption and symbol/packet error are included as variables of PHY layer model. Additive White Gaussian Noise (AWGN) is considered as the physical channel environment which signal is transmitted with power spectral density of $N_2/2$. Degradation of performance due to the impact of physical channel error is estimated in the meta-model of sublayer model. We assume the wireless network under one-hop signal propagation mode, relational expression between transmitted power P_{tx} and received power P_{rx} is defined in formula 3.4, 3.5. G_T and G_R denote antenna gain, L_s indicates the free-space path loss factor that influences the signal-to-noise ratio (SNR) in symbol.

$$P_{rx} = \frac{P_{tx} G_T G_R}{L_s} \tag{3.4}$$

$$L_s = (\frac{4\pi d}{\Phi})^2 \tag{3.5}$$

In order to mimic the elementary performance degradation from modulation typical sensor node radio, we analyze internal characters from sensor node and external channel environment. These impact the symbol transmission

Table 3.1: Symbol error probability expression

Modulation	Error probability P_b
FSK	$\exp\left(-\frac{k}{2}\left(\frac{\varepsilon_b}{N_0} - 2ln2\right)\right)$
BPSK	$Q\left(\sqrt{\frac{2\varepsilon_b}{N_0}}\right)$
DPSK	$\frac{1}{2}\exp\left(-\frac{\varepsilon_b}{N_0}\right)$
QPSK	$1 - \left[1 - Q\left(\sqrt{\frac{2\varepsilon_b}{N_0}}\right)\right]^2$
16-QAM	$3Q\left(\sqrt{\frac{4}{5}\frac{\varepsilon_{avg}}{N_0}}\right)\left[1 - \frac{3}{4}Q\left(\sqrt{\frac{4}{5}\frac{\varepsilon_{avg}}{N_0}}\right)\right]$

error of different modulations. Internal limitations of sensor node magnify along with MAC layer overhead, PHY channel transmission failure give rise to additionally performance deterioration of MAC layer. For the typical symbol error probability based on different modulations, the summary formula [105] is represented in Table 3.1. In here, we consider symbol error probability of BPSK modulation $p_{e,s|BPSK}$ which is represented as formula 3.6:

$$p_{e,s|BPSK} = Q\sqrt{\frac{P_{rx}}{BN_0}} \tag{3.6}$$

$$= Q(\sqrt{\frac{P_{tx}G_T G_R \Phi^2}{(4\pi)^2 d^2 BN_0}}) \tag{3.7}$$

Where symbol error probability of BPSK modulation is computed approximately with Q-function. Substituting the expression of transmission loss into symbol error probability, the formula can be expanded eventually in equation 3.7. It can be deduced that transmission power p_t provides positive correlation with $p_{e,s}$. Transmission distance d, bit repetition rate Φ and noise power N_0 with bandwidth B are in negative correlation with $p_{e,s}$. Additionally, packet transmission failure probability can be represented as follows:

$$n_s = \frac{L_P - L_H}{log_2(M)} \tag{3.8}$$

$$p_{e,f} = 1 - (1 - p_{e,r}(\gamma < \gamma_t))(1 - p_{e,s})^{n_s} \tag{3.9}$$

where $p_{e,r}$ indicates average transmission error probability due to system outage over Rayleigh fading channels, the number of symbols per packet is defined as n_s

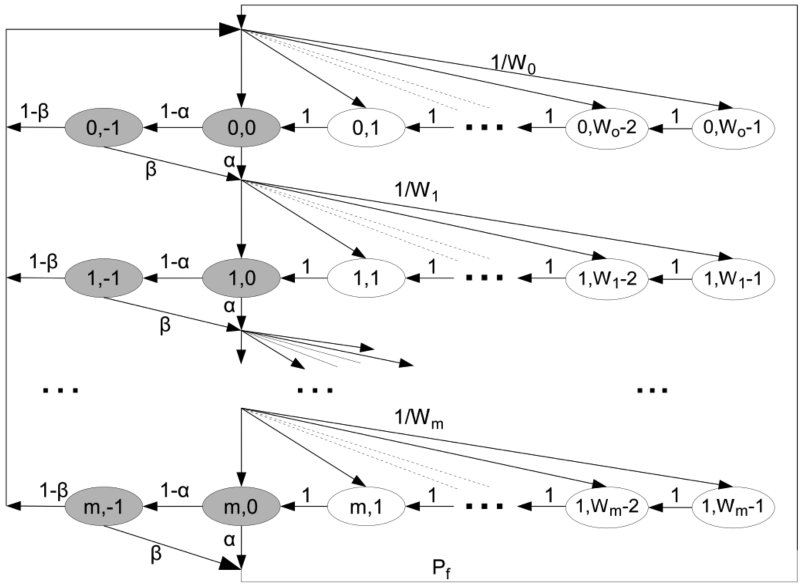

Figure 3.3: Markov chain transition states diagram for CSMA/CA algorithm in IEEE 802.15.4

in Equation 3.8 in the condition of packet with L_P bit payload size. Therefore, packet transmission error probability is computed which undergo different physical channel constraints with varying parametric variables set. This can allow us to integrate PHY layer performance degradation adaptively into MAC layer model in order to obtain stack model with multivariate parameters.

3.2.2 MAC layer CSMA/CA Markov Chain Model

For MAC layer analytical model, we analysis CSMA/CA procedure with Markov chain model which is derived from [94]. The detail of Markov chain transition states diagram for CSMA/CA algorithm is illustrated in Figure 3.3.

$$\tau = (1 - P_{idle}) \left(\frac{1 - x^{N_{caf}+1}}{1 - x} \right) \left(\frac{1 - y^{N_{rtx}+1}}{1 - y} \right) b_{0,0,0} \tag{3.10}$$

$$\alpha = (1 - (1 - \tau)^{N-1})(1 - \alpha)(1 - \beta) \left(L + L_{ACK} \frac{N\tau(1 - \tau)^{N-1}}{1 - (1 - \tau)^{N}} \right) \tag{3.11}$$

$$\beta = \frac{1 - (1 - \tau)^{N-1} + N\tau(1 - \tau)^{N-1}}{2 - (1 - \tau)^{N} + N\tau(1 - \tau)^{N-1}} \tag{3.12}$$

Reliability

Three elemental variables in CSMA/CA procedure are defined by τ, α and β, which is expressed in Equation 3.10 - 3.12. They denote three states probability in channel clear assessment (CCA) procedure, which indicate the probability when node is not in idle state in a random slot, the first CCA attempt failure probability and the second CCA attempt failure probability respectively. Following the expressions in [94], the system error can be estimated as follows:

$$x = \alpha + (1 - \alpha)\beta \tag{3.13}$$

$$y = (1 - (1 - p_{c,f})(1 - p_{e,f}))(1 - x^{N_{caf}+1}) \tag{3.14}$$

$$p_{caf} = \frac{x^{N_{caf}+1}(1 - y^{N_{rtx}+1})}{1 - y} \tag{3.15}$$

$$p_{rtx} = y^{N_{rtx}+1} \tag{3.16}$$

Where, x, y represent probability of CCA attempts failure and transmission failure in the range of maximum CCA attempts limitation N_{caf} respectively. p_{caf}, p_{rtx} denote the probability of packet discard due to maximum CCA attempts limitation and maximum retransmissions failure attempts respectively.

Delay

System delay is considered as the expected time expenditure to transmit a packet successfully. Expected number of retransmissions attempts $\mathbb{E}[n_{rtx}]$ is defined in Equation 3.17 to evaluate the influence of variable parameters on time overhead for receiving packet. $P(X = n)$ represents probability corresponding to each number of retransmission attempts, N_{rtx} is maximum retransmission attempts limitation. After expanding expression, $\mathbb{E}[n_{rtx}]$ could be described with $p_{c,f}$ and $p_{e,f}$. Furthermore, the total expectation value of time overhead $\mathbb{E}[\tau_d]$ is expressed in Equation 3.18. $\mathbb{E}[\tau_0]$ represents the time consumption

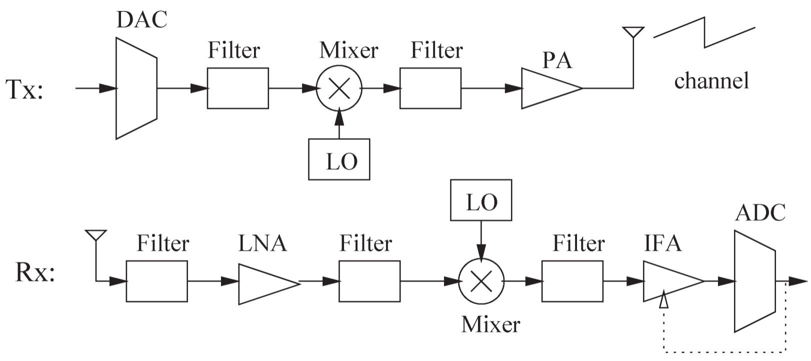

Figure 3.4: Typical transceiver structure

of a successfully transmission, wasted time overhead can be divided into the sum of each expected time intervals multiplied by the corresponding number of retransmission attempts.

$$\begin{aligned}\mathbb{E}[n_{rtx}] &= \sum_{n=1}^{N_{rtx}-1} \tau_n P(X=n) \\ &= \frac{(1-(1-p_{c,f})(1-p_{e,f}))(1-(1-(1-p_{c,f})(1-p_{e,f}))^{N_{rtx}})}{(1-p_{c,f})(1-p_{e,f})}\end{aligned} \tag{3.17}$$

$$\begin{aligned}\mathbb{E}[\tau_d] =& \mathbb{E}[\tau_0] + \mathbb{E}[n_{rtx}](T_{DATA} + T_{ACK}^{timeout} + 2T_{IPS}) + \\ & \frac{\mathbb{E}[n_{rtx}]}{N_{rtx}-1} \sum_{n=1}^{N_{rtx}-1} \mathbb{E}[T_{backoff}(n)]\end{aligned} \tag{3.18}$$

Energy consumption

Cross-layer energy consumption analytical model is proposed based on the combination of PHY layer and MAC layer energy analytical model. Overall energy consumption is separated into effective energy consumption and wasted energy expenditure, as illustrated in Figure 3.5. In the basic procedure of CSMA/CA mechanism, Clear Channel Assessment is excuted in contention window size, backoff mechanism reset CCA procedure in the case of congestion. Frequent channel congestions increase the dependent energy usage on backoff mechanism for each successful packet transmission. Therefore, packet discard due to maximum backoff failure numbers or maximum frame retries attempts limitation result in the wasted energy without successful data transmission.

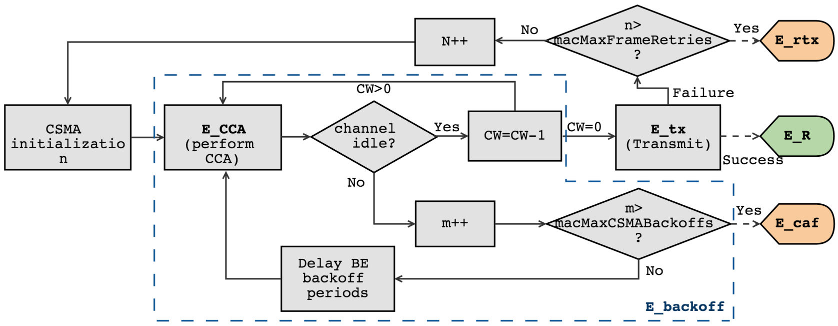

Figure 3.5: Effective energy consumption in CSMA/CA mechanism

For the base power consumption of circuit component, we accept the expression of power consumption of transmitter and receiver from [104]. The typical transceiver structure with linear modulation is shown in 3.4. The major elements of energy consumption are digital-analog converter (DAC/ADC), low pass filter (LPF), low noise amplifier (LNA), bandpass filter (BPF), mixer, frequency synthesizer and power amplifier (PA). All the components of power consumption are considered as fixed values. The final fixed circuit power consumption is estimated by the summation of elementary values, which is integrated into the calculation of effective energy consumption. The effective energy consumption can be estimated with expectation of energy consumption for receiving every bit information successfully, which is accepted as metrics to demonstrate the efficiency of energy conversion in given network environment parameters. As presented in Equation 3.19-3.23, each part of expected energy consumption value could be computed separately:

$$\mathbb{E}[E_{backoff}] = \sum_{i=0}^{n_{caf}} (\alpha - (1-\alpha)\beta)^i \left(2 + \frac{\alpha + 2(1-\alpha)\beta}{\alpha + (1-\alpha)\beta} i\right) \cdot (E_{CCA} + E_{c,fixed}) \tag{3.19}$$

$$\mathbb{E}[E_{caf}] = \sum_{j=1}^{n_{caf}} y^j j (E_{backoff} + E_{tx} + E_{c,fixed}) + (n_{rtx}+1) \frac{\alpha + 2(1-\alpha)\beta}{\alpha + (1-\alpha)\beta} (E_{CCA} + E_{c,fixed}) \tag{3.20}$$

$$\mathbb{E}[E_{rtx}] = (n_{rtx}+1)(E_{backoff} + E_{tx} + E_{c,fixed}) \tag{3.21}$$

$$\mathbb{E}[E_R] = \sum_{j=0}^{n_{caf}} y^j (j+1)(E_{backoff} + E_{tx} + E_{c,fixed}) \tag{3.22}$$

$$E^* = \frac{R^* \cdot E_R + p_{caf} E_{caf} + p_{rtx} E_{rtx} + p_{idle} E_{idle}}{R^* \cdot \Phi(L_p - L_H)} \tag{3.23}$$

Where overall energy consumption is separated into several contributions. $\mathbb{E}[E_R]$ represents the energy expenditure in terms of reliable transmission. $\mathbb{E}[E_{backoff}]$ indicates the elementary energy that sensor node performs backoff procedure in the CSMA/CA backoff mechanism. $\mathbb{E}[E_{caf}]$, $\mathbb{E}[E_{rtx}]$ are calculated for the expected wasted energy in failed CSMA/CA procedure due to channel access failure and maximum number retransmission attempts, respectively. $E_{c,fixed}$ is fixed circuit energy of transmitter and receiver [104]. Consequently, the overall efficiency of energy consumption is derived from Equation 3.23. The summation of expected energy overhead E^* is computed depend on the prior probability of each state then divided by reliable throughput in bit.

3.3 Performance Analysis

3.3.1 Simulation Scenario

We analyze simulation results of stack cross-layer analytical model through the comparison with benchmark models in different simulation scenarios. The simulation concludes two aspects of performance evaluation. On the one hand, the relationship between system performance degradation and input variables is evaluated. Information loads Φ and propagation distance d are selected as two scenarios in the experiment. On the other hand, four metrics (Thoughtput, Delay, Reliability and Effective energy consumption) are chosen to characterize node performance with increasing information loads. Besides, we evaluate the effective energy consumption by stack model which is under multi-dimensional

Table 3.2: Simulation Parameters

Parameter	**Value**
Node numbers	20
Channel carrier	2450MHz
MacMinBE	3
MacMaxBE	5
MacMaxCSMABackoffs	4
MacMaxFrameRetries	3
MinBackoffExponent	3
MaxBackoffExponent	5
L_H	6 $bytes$
L_P	127 $bytes$
E_{tx}	30 mW
E_{rx}	40 mW
E_{CCA}	40 mW
E_{idle}	0.8 mW
$\overline{E}_{c,fixed}$	2.86 $\mu J/symbol$

parameters space. Joint model [98] and typical MAC layer model [94] are evaluated as benchmark performance. The elementary power states [106] and CSMA/CA parameters are listed in Table 3.2.

3.3.2 Performance Metrics

the local probability of collision should be the same for all nodes. Under saturated traffic, all nodes will reach a steady state where the probability of collision reaches a constant and equal value for all sensor nodes. For the performance evaluation of multivariate stack model, the main performance metrics are simulated:

- **Throughput :** Total amount of data flow through communication.
- **End-to-end delay :** This metric indicates the average elapsed queuing time to receive a frame which is an important metric in respect of quality of service.
- **Transmission error probability :** The probability of transmission attempt failure due to congestion and channel error which indicates the accuracy of sensed data transmission.

- **Maximum retransmission failure probability :** The probability of CSMA/CA retransmission attemps failure due to retransmission attempt limitation which indicates the overall performance of transmission.

- **Packet delivery ratio :** The probability of successful message transmission that indicates the data accuracy and dependability of communication.

- **Effective Energy consumption :** E^* energy model in previous subsection is used to indicate the performance of efficiency in energy usage. Instant energy consumption in simulation sequence and average energy consumption in terms of different number of sensor nodes are evaluated separately.

3.3.3 Simulation Results

Analysis of packet error probability with multivariable parameters

Firstly, Figure 3.6 compares performance degradation on transmission failure under three transmission distances. Joint model generates significant impact on transmission failure probability, which is compared to single MAC layer model, especially in the light information loads. Constant output of PHY channel error have risk of overestimation for the condition of multivariate parameters. From the results in different distances, it is obvious that expanded information loads Φ results in frequent channel collision $p_{c,f}$ and maximum CSMA backoff failure p_{caf} which is under PHY channel constraints $p_{e,f}$. It further causes a higher probability of failed packet transmission. Transmission failure probability also presents growth trend with increasing distance parameter simultaneously in each parts of simulation scenario Φ, which verifies the output in Figure 3.7.

Furthermore, we characterize packet error probability over different information loads . In Figure 3.7, transmission failure probability of stack model is simulated in the range of transmission distance parameter from 0 to 100 m in channel environment with different noise power. The information load Φ is given at 2 frame/s. According to the description of joint cross-layer model, dashed line represents the static estimated output of transmission failure probability in 20 m distance range. In stack model, transmission failure is counted dynamically through the combination of PHY and MAC layer model. Under the case that noise power $N_0 = 10$ dB, transmission failure probability increases significantly during the distance range 20 to 60 m. Thus, theoretical transmission distance can be predicted from given parameters. It can be observed that higher channel noise level increases the probability of packet transmission error in given distance.

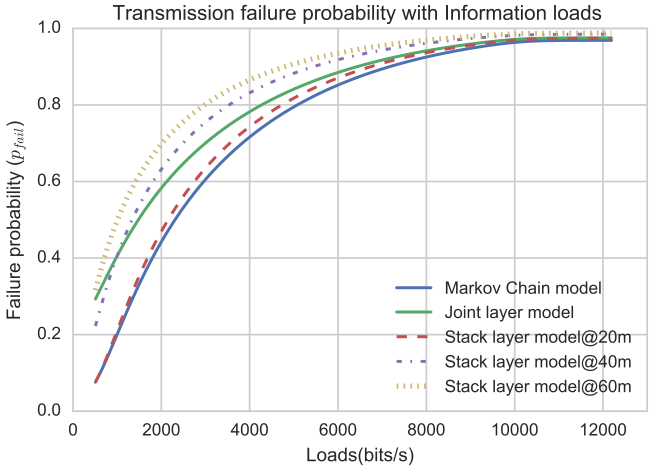

Figure 3.6: Transmission failure probability with information loads

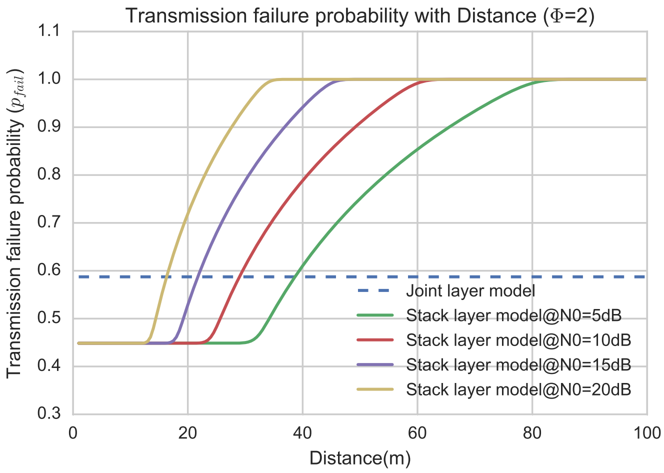

Figure 3.7: Transmission failure probability with distance

Analysis of retransmission mechanism with multivariable parameters

Similar analysis can be interpreted to the performance variance of maximum retransmissions failure probability, as shown in Figure 3.8. Figure 3.9 demonstrates packet discard probability due to the maximum retransmission attempts failure of CSMA/CA procedure. Prior probability of combined transmission failure boosts probability distribution variance of maximum packet retries limitation in different distances and noise levels.

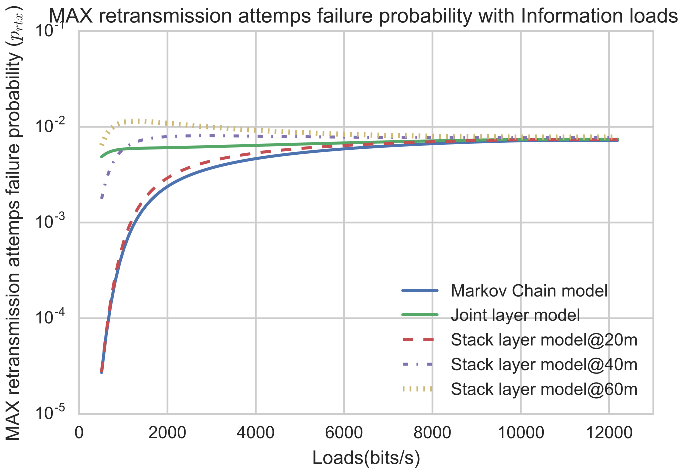

Figure 3.8: stack cross-layer model structure

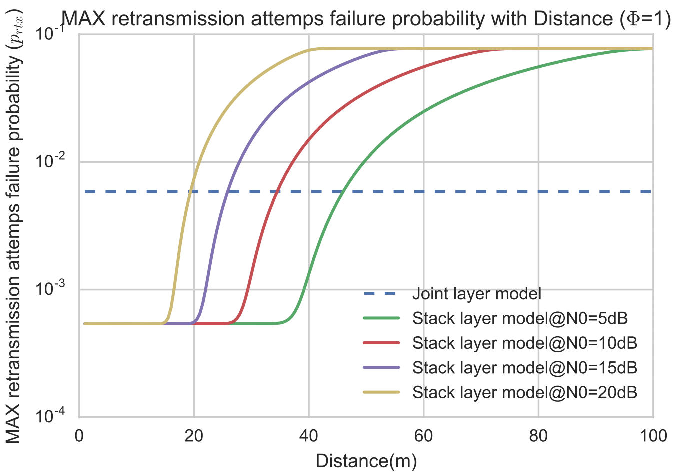

Figure 3.9: stack cross-layer model structure

Analysis of QoS performance

In this subsection, we compare the global performances of stack model, joint model and single MAC layer model which are appraised under pressure testing scenario. Data information load Φ increases from 400 bits/s to 12000 bits/s with given parameters sets.

As shown in Figure 3.10, with low data loads, adaptive PHY layer model provides inconspicuous influence on node average thoughtput. Under the case of stable condition, stack model keeps cautious estimation on PHY channel error which compared with joint model. As offered information loads increase to the

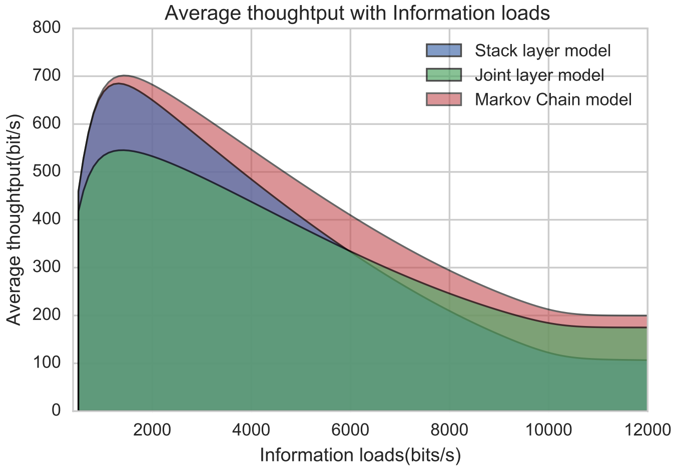

Figure 3.10: Average thoughtput with information loads

saturate range, the provided average thoughtput shrinks significantly in stack layer model. This is caused by frequent channel collision, transmission failure and optimized PHY channel error which are computed based on combined layers model as shown in Figure 3.6-3.9.

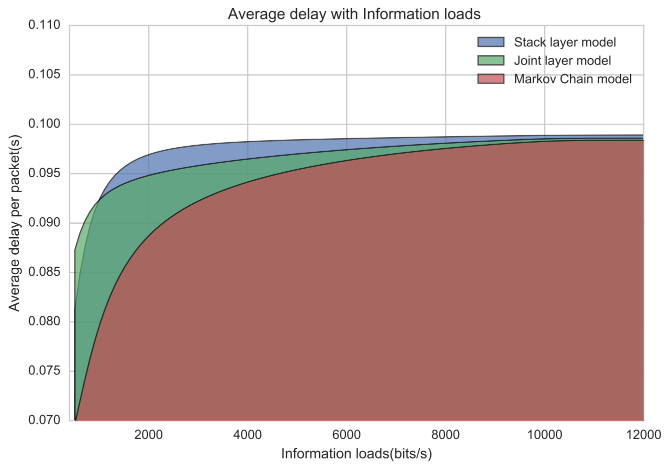

Figure 3.11: Average delay with information loads

Figure 3.11 depicts the evolution of time overhead due to CSMA procedure for transmitting each packet successfully. In the heavy network loads of stack model, the expected time overhead increases dramatically. Expanded number of retransmissions attempts cause extra wasted time along with the probability of channel collision $p_{c,f}$ and PHY channel error $p_{e,f}$, respectively. Expected frame retries number n_{rtx} is applied as a coefficient of time expenditure due to frame control message overhead (T_{IPS}, T_{ACK}, $T_{ACK}^{timeout}$, etc.) for additional delay estimation. Furthermore, time overhead of continual backoff procedure also give rise to additional latency in the range of maximum retransmissions attempts N_{rtx}. High data loads have significant impacts on the overall system reliability, as shown in Figure 3.12. Stack model obtains more decline trend of reliability performance compared with joint layer model.

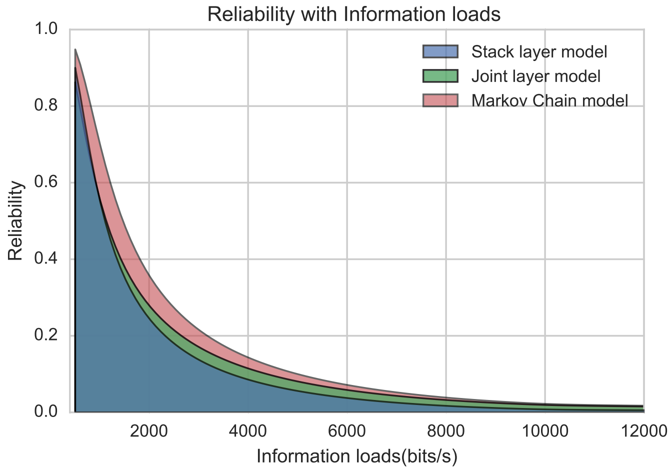

Figure 3.12: Reliability with information loads

Analysis of effective energy consumption

For the evaluation of efficient energy consumption, we also rebuild energy consumption estimation module for single MAC layer model [94] and joint model [98] respectively. Additional fixed circuitry energy cost [103] is considered as elementary value $\overline{E}_{c,fixed} = 2.86\mu J/symbol$. As illustrated in Figure 3.14, at the range of light offered loads in stack model, effective energy consumption increases linearly with input value Φ. PHY layer model achieves indistinctively outcome of effective energy consumption for reliable transmission compared to the results of signal layer MAC model. The result of joint model in early range can be explicated as its overestimation on systematic performance degradation as the result of the constant PHY channel error estimation. In the situation that Φ increases to a saturate load level, efficient energy consumption E^* increases dramatically due to the extra wasted energy expenditure under cross-layer constraints model. Evolution indicates that network sacrifice efficiency of overall energy conversion to make up the lack of QoS.

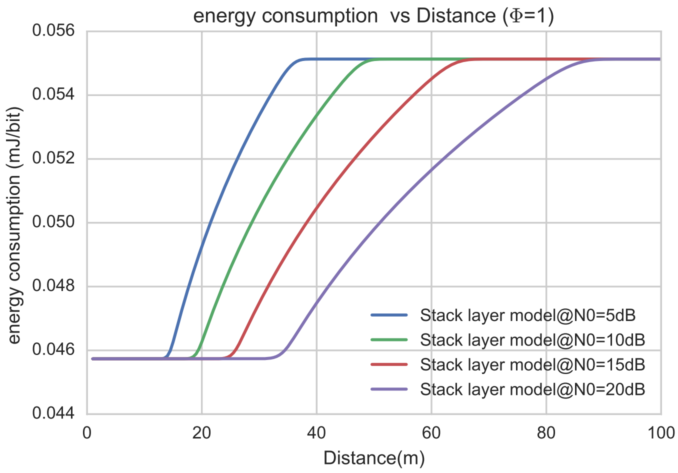

Figure 3.13: Energy consumption in distances

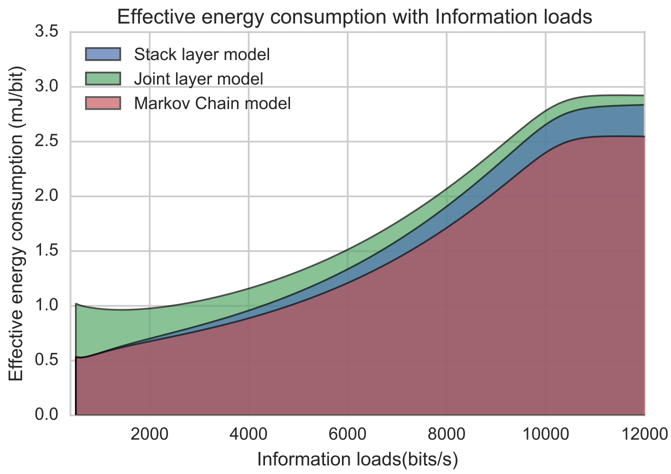

Figure 3.14: Effective energy consumption with information loads

Finally, three dimensional surface figure helps us to mimic the multivariate functionalities of stack cross-layer model. Figure 3.15 presents the sampling observation of effective energy consumption output, which influenced by the input vectors of multivariate parameters (distance d, offered loads Φ and noise level N_0). In the condition of $N_0 = 10$, effective energy consumption E^* generates linear increasing trend along with the growth of data flow Φ or propagation distance d. It's noticeable that the exceptional results of energy expenditure locate in the region of long distance and high data rate. The phenomenon could be interpreted by the comprehensive affecting factors under

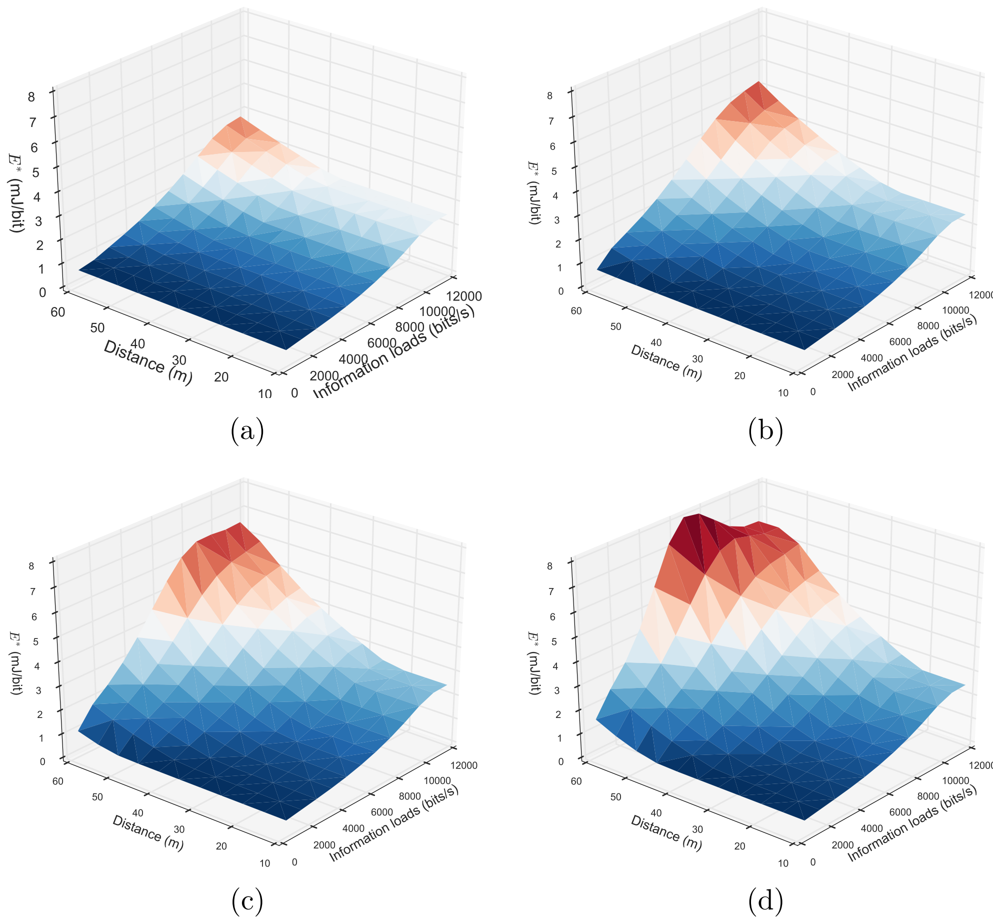

Figure 3.15: 3D plots of effective energy consumption E^* with multivariate parameter sets (Φ, d, N_0): (a) $N_0 = 10$, (b) $N_0 = 15$, (c) $N_0 = 18$, (d) $N_0 = 20$.

cross-layer conditions. From the evolution of E^* with growing channel noise level N_0, region in long distances and saturated data flow show more significant increment. Simulation results are in a good agreement with the interpretations in previous section.

3.4 Conclusion

In this chapter, based on the joint layer model [98] with static PHY layer error calculation, we proposed a stack cross-layer model for comprehensive performance analysis of IEEE 802.15.4 network. This approach is published in [107]. Adaptive physical channel propagation model is integrated into Markov chain MAC layer model, which is evaluated with multivariate parameter inputs. Network performance are fully assessed under the dynamic interaction from single sublayer models and multi-dimensional parameters environment, respectively. The simulation results in different scenarios verify that multivariate stack model achieves more comprehensive systemic analysis on QoS performance and effective energy consumption, especially allow us to reproduce faithful performance tracking under multi-dimensional parameters. On the one hand, dynamically interactions in stack model boost the global performance variance in the event of network deterioration. On the other hand, under the situation of light application loads or network with high reliable hyperparameters, stack model offers conservative prediction output that avoid overestimating the degradation of network performance.

Chapter 4

Fuzzy-Based Dynamic Bandwidth Multi-Channel Assignment Mechanism for Wireless Multimedia Sensor Network

In previous chapter, we study the stack cross-layer analytical model based on the comprehensive combination and interaction between PHY layer propagation model and MAC layer Markov chain model. The essential research allow us to analysis the relationship between system performance and changing network environment with multivariate parameters. Especially in the application field of Wireless Multimedia Sensor Networks (WMSNs), Saturated data flow increases the probability of congestion and collision in transmission which dramatically degrade the performance of Quality of Service (QoS). Multi-channels deployment technique is often applied to parallel transmission for QoS guarantee. However, how to make trade-off between QoS requirement and energy efficiency is challenge for energy-constrained WMSNs. Smart system decision plays an important role in dynamic resource deployment of network. On the basis of requisite study in previous chapter, we present a fuzzy-based dynamic bandwidth multi-channel assignment mechanism (MCDB_FLS). Cross-layer proactive available bandwidth is estimated as parameters for multi-channel deployment admission control. non-overlapping channels is dynamically deployed according to the channel allocated admission algorithm for the different application load. Reinforcement learning-based approach is proposed for more wisely decision-making in multi-channel allocation mission. Furthermore, fuzzy logic-based bandwidth threshold model provides dynamic optimization on system admission control. The object of MCDB_FLS is try to make smart system control on multi-channel allocation, which could achieves the trade-off between energy efficiency and QoS improvement in multimedia application with saturate information loads.

4.1 Introduction

In traditional wireless sensor network(WSN) applications, energy efficiency may be considered to be the most important concern whereas bandwidth utilization and throughput maximization are of secondary importance. In the 2.4 GHz band, the standard allows 16 non-overlapping channel of 5 MHz each, in theory, each channel is capable of transmitting in 250 Kbps. However, with the development of Wireless Multimedia Sensor Networks (WMSNs) which is composed by embedded cameras and microphones besides scalar sensors, real-time multimedia applications require high level Quality of Service(QoS) guarantee in high data rate. Multimedia tasks usually experience high traffic which give rise to interference and collisions in conventional WSNs with limited single channel bit rates.

For the performance requirement of multimedia tasks, multiple channels techniques [108, 15] are studied and deployed. WSNs perform parallel transmissions over multiple channels which could alleviate congestions and improve network capacity. In the conventional multi-channel assignment approaches, sensor mote with multiple channels radio module [109] can be programmed to operate on different unlicensed channels. Multiple channels are assigned based on different protocols which can be classified by fixed channel-based assignment [18, 110, 111], dynamic channel-based assignment [112, 113], and semi-dynamic channel-based assignment [17, 114, 115]. Additional, with the capability of dynamic spectrum access in cognitive radio-based sensor networks (CRSNs), channel bonding techniques [16] combine contiguous non-overlapping channels to provide higher spectrum utilization. Experimental research [116, 27] indicate that channel bonding techniques achieve larger channel capacity, which effectively enhance overall throughput and alleviate congestion even within the extra negotiation overheads of multi-hop sensor networks. However, due to the features of sensor network applications, efficient multi-channel schemes should provide intelligent approaches to allocate available resources based on the requirement of applications. The parts of challenges are shown as follow:

- Find effective performance indicator and network performance estimation approach, which could be used as the system control input of multi-channel assignment mechanism.

- Multi-channel approaches consume more energy than conventional WSNs on energy constrained devices. It's necessary to design comprehensive control mechanism for the trade-offs between QoS guarantee and energy efficiency.

- In multiple application schedules, sensor nodes may perform different tasks simultaneously or make multi-channel controls without experience. Static multi-channel approaches hardly guarantee good performance all the time. Self-adaptive control system is required to learn appropriate pattern from different environments, specifically for multi-task WMSNs.

In this chapter, we propose a fuzzy-based multi-channel assignment mechanism for WMSNs based on dynamic bandwidth control. Non-overlapping channels are dynamically allocated according to the instantaneous performance of QoS. Bandwidth is a crucial resource in WMSNs which has tight relevance with performance of QoS and energy consumption in existing studies. Limited throughput along with cross-layer overhead and interference give rise to congestion and collision, which increase end-to-end latency and packet error rate correspondingly. The degradation of performance accompanies Medium Access Control (MAC) protocol process further impact on the available bandwidth. Thus, we estimate residual available bandwidth as the global performance indicator and integrate it into the proposed multi-channel assignment algorithm.

For the purpose of adaptive decision-making, we propose reinforcement learning-based approach for multiple application tasks. During the self-learning iterations, sensor nodes collect system performance and learn to make best channel assignment decision for adapting with the surrounding environment. For the system admission control, light weight fuzzy logic system is integrated into bandwidth threshold tuning model. Fuzzy logic-based algorithm build mapping function from extracted performance metrics to suitable triggering threshold for multi-channel assignment mechanism in an efficient way.

The remainder of this chapter is organized as follows. In Section 4.2, dynamic bandwidth multi-channel assignment mechanism is proposed and discussed in detail. Section 4.3 illustrates performace evaluation and analysis. Finally this paper is concluded in Section 4.4.

4.2 Multi-Channel Dynamic Bandwidth Assignment

In this section, we present a QoS aware and energy efficient cross-layer structure based on multi-channel bandwidth adaptive control, as illustrated in Figure

4.2. We assumes that a network is well-connected, and nodes within two hops distance constitute the interference range of each node. Whenever a sensor node received data flow, the residual bandwidth estimation module collects the current t-th state information from MAC layer and physical layer. Cross-layer data are gathered and preprocessed which can be exploited to estimate available bandwidth for next period of communication task. Available bandwidth ratio ω is considered as the metric for system control of multi-channel allocation mechanism.

Simultaneously, application layer acquire the available bandwidth estimation value and monitor the performance of MAC layer. Based on the variable QoS requirement from application object, two tunable thresholds ω^{high} and ω^{low} are defined in order to trigger the function of multi channels allocation for next period state. As shown in Figure 4.1, on the basis of estimated value ω from residual bandwidth estimation module, the range between ω^{high} and ω^{low} indicates that sensor node currently stay in steady state. If ω exceed the threshold values, Bandwidth control mechanism is triggered.

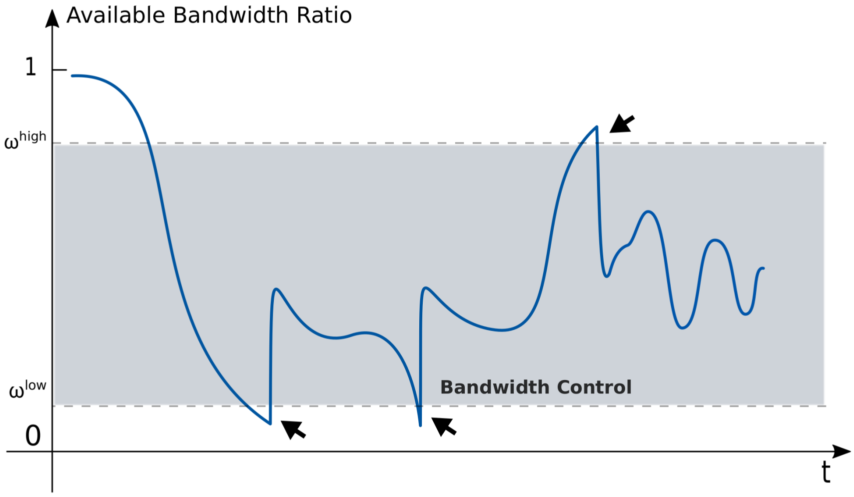

Figure 4.1: Estimation-based adaption of available bandwidth ratio

In order to dynamically perform multi-channel admission control, Fuzzy logic system is designed to tune bandwidth thresholds according to the changing of network environment. Cross layer performance statistic is selected as input space of fuzzy logic admission control module, generate crisp output values of ω^{high} and ω^{low}. Tabular-based reinforcement learning module try to make better system decision of allocated channel numbers if it receive self-learning procedure single from fuzzy logic admission control module. After each iteration that reinforcement learning module is triggered to select action based on strategy policy, updating Q-function values will boost self-learning procedure of RL

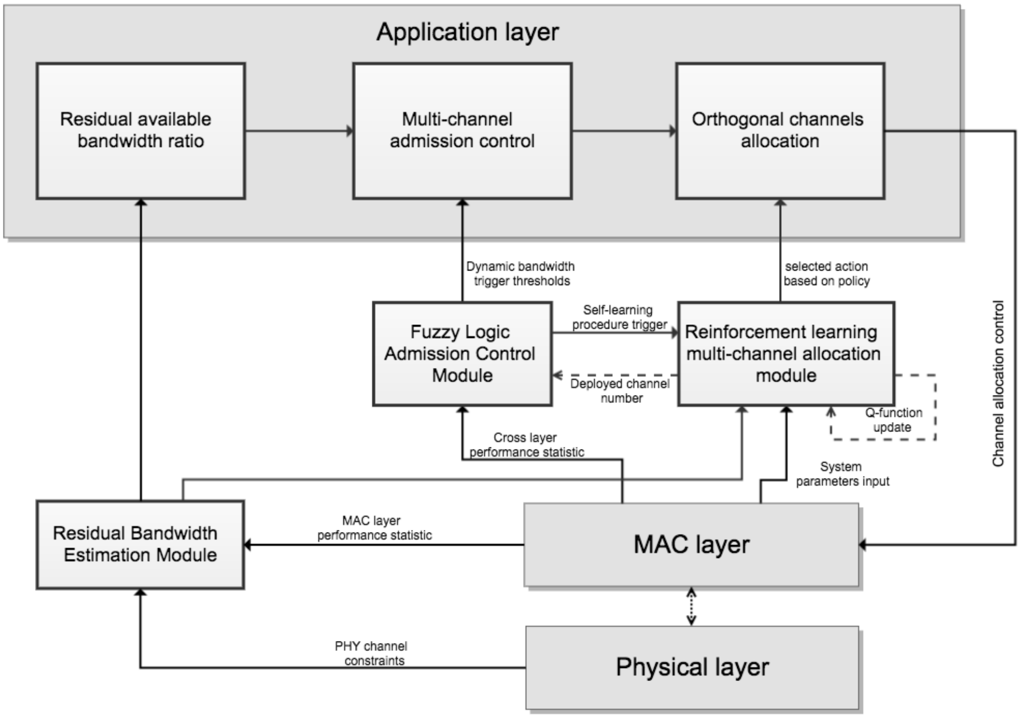

Figure 4.2: Multi-channel Dynamic Bandwidth Assignment Structure

module. Orthogonal channels are allocated and channel numbers are optimized for next $t+1$-th period state.

4.2.1 Residual Bandwidth Estimation Module

For controlling available bandwidth-based multi-channel deployment system, high-precision residual bandwidth prediction is requisite foundation. The state of the art in available bandwidth evaluation is divided into passive estimation methods and active estimation methods. The passive estimation approach is based on the local information retrieved from cross-layer of node and exchanged with neighbour nodes, consequently evaluate the real-time available bandwidth estimated information. In this article, bandwidth estimation module rely on passive approach is proposed as the solution of available bandwidth estimation for dynamic deployment system. Residual available bandwidth ratio compared to current maximum channel rate is considered as the metric in order to evaluate the global performance of node. Our bandwidth estimation module get inspiration from preview passive estimation method Available Bandwidth Estimation (ABE) [117] for IEEE 802.11 wireless network. This method considers the bandwidth consumption due to non-synchronization between neighbour nodes, practical-based collision estimation and random backoff mechanism. Equation 4.1 represent the available bandwidth calculated by ABE module.

$$\omega_{ABE} = (1 - K) \times (1 - P_c) \times \frac{T_I^e}{\Delta} \times \frac{T_I^r}{\Delta} \times C_{max} \tag{4.1}$$

Where Δ is united monitoring period of each node, T_I^e and T_I^r are the idle time of wireless medium during united monitoring period Δ at emitter and receiver respectively. P_c is the evaluated collision probability depends on different payload size which cause the degradation of available bandwidth, collision probability of packet with given payload size is calculated through the measured collision probability of periodically emitted *hello* packet with Lagrange interpolating polynomial. K represents the ratio of time consumption due to backoff mechanism in period interval Δ. The expectation value of backoff time is calculated in equation 4.2, in the condition of maximum contention windows size equals to C.

$$\begin{aligned} T_{\overline{backoff}} &= \sum_{k=0}^{C} P(X=k) \cdot \frac{min(CW_{max}; 2^k \cdot CW_{min}) - 1}{2} \\ &= \frac{1 - p_c - 2^M p_c^{M+1}}{2 - 4p_c} CW_{min} - \frac{1}{2} \\ e.t. \quad & CW_{max} = 2^M \cdot CW_{min} \quad M \leq C \end{aligned} \tag{4.2}$$

Packet Control Messages Overhead

The CSMA/CA mechanism of IEEE 802.15.4 standard can be divided into reliable mode and unreliable mode specific on MAC layer protocol. In this article, we consider reliable mode as the default MAC layer protocol mode because of the high QoS requirements in real-time multimedia applications. The IEEE 802.15.4 frame exchange sequence with reliable mode can be depicted in Figure 4.3. After sending data frame each time, node will wait for a constant time interval for the feedback ACK frame. if no ACK frame is detected, a new backoff delay procedure will be triggered, then node executes the retransmission attempts within the range of maximum retransmission limitation which is predefined in CSMA/CA protocol. Thus, acknowledgement packet is a non-ignorable overhead that impact to the available bandwidth.

For successfully transmitting a packet, n is assumed as the number of retransmission attempts. In the $n-1$ previous unsuccessful delivery intervals, node

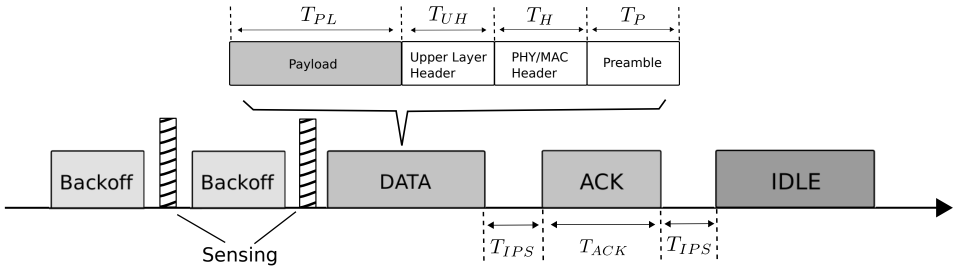

Figure 4.3: IEEE 802.15.4 frame exchange sequence

consumed constant time $T_{ACK}^{timeout}$ in order to verify the acknowledgement packet loss. In the last nth successful transmission period, as shown in Figure 4.3, T_{ACK} is the time consumption that transmitter receives the acknowledgement packet. The time period of T_{ACK} can be expressed as [118]

$$T_{ACK} = \frac{L_H}{B \cdot R_c} + T_p \tag{4.3}$$

Where L_H is the length of PHY and MAC layers header in bits, B is channel rate, R_c represents channel code rate and T_p indicates the time duration for transmitting preamble of each frame. Thus, for transmitting a packet successfully, the time consumption $\tilde{\tau}_{ACK}$ due to ACKs mechanism in CSMA/CA protocol can be approximately calculated in equation 4.4. $\tilde{n}$ denotes the expectation number of retransmission attempts which will be defined in the following subsection, T_{IPS} represents the time interval of inter packet space.

$$\tilde{\tau}_{ACK} = T_{ACK} + \tilde{n} T_{ACK}^{timeout} + 2(\tilde{n} + 1) T_{IPS} \tag{4.4}$$

MAC Layer Overhead

In the previous subsection, we discussed the bandwidth overhead due to random exponential backoff mechanism $T_{\overline{backoff}}$ in the limited contention windows for each transmission attempt and the expectation time consumption of acknowledgement control messages T_{ACK} and $T_{ACK}^{timeout}$ for every successful transmission. Considering the retransmissions functionality of CSMA/CA specific to collision and packet loss, the retransmissions of frame guarantee the reliability of packet delivery at the expense of efficient channel rate. If every packet is transmitted successfully for the first time, the predictable duration for active time interval τ_0 can be written as:

$$\tau_0 = T_{\overline{backoff}} + T_{DATA} + T_{ACK} + 2T_{IPS} \tag{4.5}$$

In order to predict the degradation of available bandwidth due to retransmissions mechanism, the expectation number of retransmissions attempts $\widetilde{n}_{rt}$ can be derived by equation 4.6. n and N indicates the number of retransmission attempts and the maximum retransmission limitation respectively. $P(X=n)$ represents the probability corresponding to each number of retransmission attemps, collision probability p_c can be derived from measure-based method with *hello* packets [117] or unconditional estimation method by Poisson process [119].

$$\begin{aligned}
\widetilde{n}_{rt} &= \sum_{n=1}^{N-1} nP(X=n) + Np_c^N \qquad (4.6)\\
&= p_c(1-p_c) + 2p_c^2(1-p_c) + \ldots + (N-1)p_c^{N-1}(1-p_c) + Np_c^N \\
&= \frac{p_c(1-p_c^N)}{1-p_c}
\end{aligned}$$

Therefore, actual predictable time overhead $\widetilde{\tau}_{rt}$ due to failed transmission attemps can be divided into the summation of different expected time intervals multiply by probability based on corresponding number of retransmission attemps. Then, we integrate estimated time overhead $T_{\overline{backoff}}$ from equation 4.2 and time consumption for transmitting control message from previous subsection into retransmission mechanism. $\widetilde{n}_{rt}$ is substituted into expansion formula that total expected time overhead of failed transmission can be expressed as equation 4.7. Coefficient $\dfrac{\widetilde{n}_{rt}}{N-1}$ is used to calculate the expected time consumption of backoff mechanism. The collisions of packet transmission trigger retransmissions mechanism, increase extra time consumption for every successful packet transmission, time expectation $\widetilde{\tau}_{rt}$ can be considered as parameter to evaluate the degradation of available bandwidth.

$$\begin{aligned}
\widetilde{\tau}_{rt} &= \sum_{n=1}^{N-1} \tau_n P(X=n) \qquad (4.7)\\
&= \widetilde{n}_{rt}(T_{DATA} + T_{ACK}^{timeout} + 2T_{IPS}) + \frac{\widetilde{n}_{rt}}{N-1}\sum_{n=1}^{N-1} T_{\overline{backoff}}(n)
\end{aligned}$$

Physical Channel Constraints

In the previous subsections, we quantified the impact of MAC layer overhead on extra time consumption for each successfully packet transmission. Furthermore, physical channel constraint between emitter and receiver is non-ignorable factor, we take physical channel theoretical model into consideration. Degradation of available bandwidth due to the impact of physical channel error is estimated and integrated into available bandwidth estimation model. We assume the wireless network under one-hop signal propagation through AWGN channels, the relational expression between transmitted energy ε_{tx} and received energy ε_{rx} is from formula 4.8, 4.9. G_T and G_R denote antenna gain, L_s indicates the free-space path loss factor that influence SNR in symbol.

$$\varepsilon_{rx} = \frac{\varepsilon_{tx} G_T G_R}{L_s} \tag{4.8}$$

$$L_s = (\frac{4\pi d}{\lambda})^2 \tag{4.9}$$

Based on the error formulas table in [105], we can estimate the approximate error probability per symbol. Considering BPSK modulation, error probability per symbol is expressed by equation 4.10, we can build the relationship between error probability and physical channel characteristics.

$$\begin{aligned} p_{s,BPSK} &= Q\sqrt{\frac{\varepsilon_{rx}}{BN_0}} \\ &= Q\sqrt{\frac{\varepsilon_{tx} G_T G_R \lambda^2}{(4\pi)^2 d^2 BN_0}} \end{aligned} \tag{4.10}$$

Then, physical channel constraints can be converted into the impact on available bandwidth. We integrate $p_{s,BPSK}$ into equation 4.6 to update the expectation number for retransmissions attemps $\widetilde{n}_{rt,BPSK}$ as shown in follows:

$$\widetilde{n}_{rt,BPSK} = \frac{(1-(1-p_c)(1-(1-p_{s,BPSK})^{n_s}))(1-(1-(1-p_c)(1-(1-p_{s,BPSK})^{n_s}))^N)}{(1-p_c)(1-(1-p_{s,BPSK})^{n_s})} \tag{4.11}$$

where p_c and $p_{s,BPSK}$ are assumed as the probabilities of two independent events, n_s is the number of symbols in a packet with certain payload information. Subsequently, $\widetilde{n}_{rt,BPSK}$ is substituted into equation 4.7, $\widetilde{\tau}_{rt,BPSK}$ is obtained as the overall wasting time consumption for every successful packet transmission based on joint layers model. In order to build the relationship between failed transmission time consumption and bandwidth wastage, we propose bandwidth degradation coefficient $\theta_{rt,BPSK}$ which is defined as the ideal active time interval τ_0 divided by the actual predictable time consumption $\widetilde{\tau}_{rt,BPSK} + \tau_0$ as shown in equation 4.12. Consequently, we reconsider the available bandwidth estimation value ω_{ABE} that can be expressed with equation 4.13.

$$\theta_{rt,BPSK} = \frac{\tau_0}{\widetilde{\tau}_{rt,BPSK} + \tau_0} \tag{4.12}$$

$$\omega_{ABE} = \theta_{rt,BPSK} \times \frac{T_I^e}{\Delta} \times \frac{T_I^r}{\Delta} \times C_{max} \tag{4.13}$$

4.2.2 Reinforcement Learning Module for Multi-Channel Allocation

The trade off between energy efficiency and QoS is an important issue in WMSNs, however, single system decision-making is difficult to guarantee the overall performance based on QoS and energy management of WMSNs. In order to perform channels deployment properly, sensor nodes make systemic decision for optimized action by using reinforcement learning approach. In reinforcement learning, sensor nodes separate perceived environment and possible action into tuple type that is represented as state-action pairs (s_t, a_t). In the interaction, agents select actions then environment generate feedback to update state and give rise to rewards which is agent's objective to maximize the expected sum. s_t denotes the state of t-1th iteration in the set of possible states $\mathcal{S}$. $a_t \in \mathcal{A}(s_t)$ represents the actions in the available action set of state s_t. Reinforcement learning specifies the Q-function value, $Q_k(s_t, a_t)$, which accepts states and actions then update the value of state-action pair for next iteration. The tabular Q-function transition rule can be expressed as

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \frac{1}{visit_k(s_t, a_t)}[r(a_t \mid s_t) - Q_k(s_t, a_t)] \tag{4.14}$$

where $Q_k(s_t, a_t)$ is the Q-function value of state-action pair (s_t, a_t) for k-th transitions. $visit_k(s_t, a_t)$ indicates the number of times that (s_t, a_t) pair has been visited and performed in the past k-th transitions. $r(a_t \mid s_t)$ denotes the immediate reward for performing particular action a_t which is selected by optimal policy. The policy build a mapping function from states to actions, $\pi : \mathcal{S} \to \mathcal{A}$, aimes to maximise long timespan rewards. Considering the energy constraintion of WSN, we use more computationally efficient, tabular-based softmax action selection policy. The policy that softmax method chooses action a_t from the available action set $\mathcal{A}(s_t)$ can be expressed as

$$\pi^{\star}(a_t \mid s_t) \leftarrow \left\{ \frac{e^{Q_k(s_t,a)\big/\tau}}{\sum\limits_{i \in \mathcal{A}(s_t)} e^{Q_k(s_t,i)\big/\tau}}, a \in \mathcal{A}(s_t) \right\} \tag{4.15}$$

where softmax function calculates probability distribution of each action, parameter τ scales the probability distribution which control the trade-off between exploration and exploitation of action selection policy.

In t-th iteration active state, if estimated available bandwidth value in previous state, $\widehat{\omega}_t$, is in the steady range of preset threshold ratio value, node will skip the action selection step in current iteration. On the contrary, decision-making based on action selection of channel deployment will be triggered before communication event. The state for current iteration s_t is determined by number of active physical channels n_c and extracted information set from cross layer which was used for available bandwidth estimation in previous state s_{t-1}. From the extracted information set, mapping functions transform continuous features to discrete category dataset, only corresponding state-action pairs in lookup table are considered. The size of each available actions set $\mathcal{A}(s_t)$ is $n_{cmax} - 1$, which n_{cmax} indicates the maximum number of deployable orthogonal channels. Based on the Q-function values $Q_k(s_t, a)$ in available action set, Sensor node use $\pi^{\star}(a_t)$ to determine the best action a_t for channels deployment of current active state.

The immediate reward $r(a_t \mid s_t)$, as the main indication on Q-function transition, is defined as the gain of data throughput efficiency. During each available bandwidth estimation stage, reinforcement learning model reuse the extracted information from current observation internal and calculate immediate reward which can be expressed as

$$r(a_t \mid s_t) = \frac{G'_{RT}(a_t \mid \Delta t)}{G'_{E}(a_t \mid \Delta t)} \tag{4.16}$$

where G'_{RT} and G'_{E} represent gain of action a_t in reliable throughput and overall energy consumption respectively, compared to the average performance in the last Δt observation interval. In consequence, improvement of data transmission efficiency increase the total reward, Q-function will update through this feedback mechanism. The details of multi-channel allocation algorithm is illustrated in Algorithm 1.

Algorithm 1: Channel Allocation Algorithm

Input: $n_{c,t-1}$: Active channel numbers of t-1 th state; $\widehat{\omega}_t$: Available bandwidth estimation of t th state; (s_{t-1}, a_{t-1}): state-action pairs in previous state;

Output: $n_{c,t}$: Active channel numbers of t th state.

Init ω^{low}, ω^{high}
if $a_{t-1} \neq Nan$ **then**
  update $Q_k(s_{t-1}, a_{t-1})$
end
if $\widehat{\omega}_t < \omega^{low} C_{max}$ *or* $\widehat{\omega}_t > \omega^{high} C_{max}$ **then** `// Saturated State`
  if $\widehat{\omega}_t < \omega^{low} C_{max}$ *and* $n_{c,t-1} = n_{cmax}$ **then** `// Boundary State`
    $a_t \leftarrow Nan$
    $n_{c,t} \leftarrow n_{cmax}$
  else if $\widehat{\omega}_t > \omega^{high} C_{max}$ *and* $n_{c,t-1} = n_{cmin}$ **then**
    $a_t \leftarrow Nan$
    $n_{c,t} \leftarrow n_{cmin}$
  else `// Transition state`
    $(s_t,\, a_t) \leftarrow \pi^{\star}(a_t \mid s_t),\, a \in \mathcal{A}(s_t)$
    $n_{c,t} \leftarrow (s_t,\, a_t)$
  end
else `// Steady State`
  $a_t \leftarrow Nan$
  $n_{c,t} \leftarrow n_{c,t-1}$
end
$(s_{t-1},\, a_{t-1}) \leftarrow (s_t,\, a_t)$
return $n_{c,t}$
update C_{max}, $n_{c,t-1}$

4.2.3 Fuzzy Logic Admission Control Module

From algorithm 1, ω^{low} and ω^{high} are two trigger thresholds which is presetted with numeric metrics in order to be compared with residual available bandwidth $\widehat{\omega}_t$. Whereas, trigger threshold value with static value is inadequate for the bandwidth dynamic control mechanism because of the variability in WSNs. For the integrity of dynamic control system and the improvement of precise decision-making based on vague information, we use fuzzy logic system (FLS) to fully exploit dynamic threshold metrics for the admission control of channel assignment mechanism introduced in previous section. In fuzzy logic system, crisp numeral value is expressed into linguistic variable, several numeric data limited to input variable range is mapped into another output value of single variable. Fuzzy logic has some advantages which make it suitable to be applied in WSNs:

- Linguistic variable is expressed with natural language which is based on ordinary human communication, appropriate to describe WSNs events in more intelligent and significant way than conventional crisp logic.
- Fuzzy logic could tolerate imprecise data and uncertainty in WSNs. In conventional control techniques of WSNs, the relationship between control input space and system output space is difficult to be formulated explicitly and precisely with mathematical equations.[120]
- Fuzzy logic is intuitive approach without complex numeric model. Comparing with other artificial intelligence and machine learning techniques, fuzzy logic system is a light-weight and feasible solution which matching the intention of energy-aware WSNs.

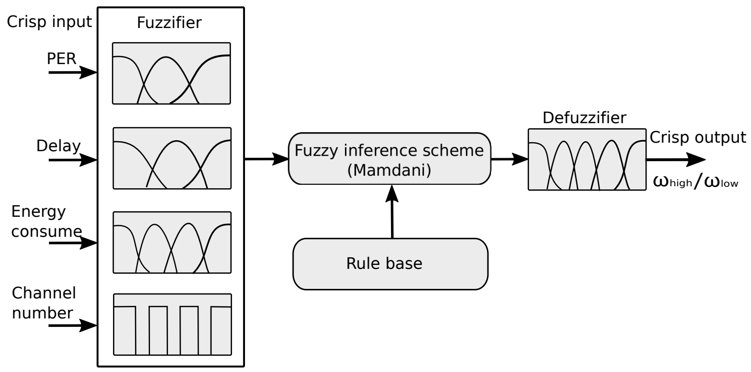

Figure 4.4: Structure of fuzzy logic system

Fuzzy logic system consists of four components: Fuzzifier, Fuzzy inference scheme, Rule base, Defuzzifier. A typical fuzzy logic system is deployed in our cross-layer architecture, as shown in figure 4.4. In the first place, several metrics of WSNs are set as the crisp input variables of fuzzy logic system, denoted by equation 4.17.

$$N = (\chi_1,\ \chi_2,\ \chi_3,\ \chi_4) \tag{4.17}$$

In crisp input set N, four numeric metrics are considered for fuzzy logic system: packet error rate (PER), latency, energy consumption and numbers of active channels. For the fuzzification step, every crisp input value is converted into multi degrees of membership functions. All the input value limit to the input vatiable range that apply to the corresponding membership functions in fuzzy set L, as shown in follow:

$$L : N \rightarrow (\chi_1^{'},\ \chi_2^{'},\ \chi_3^{'},\ \chi_4^{'}) \tag{4.18}$$

$$\chi_i^{'} = \left[\mu_{\chi_i}^{MF_1}(\chi_i),\ \mu_{\chi_i}^{MF_2}(\chi_i),\ \mu_{\chi_i}^{MF_3}(\chi_i),\ \cdots,\ \mu_{\chi_i}^{MF_j}(\chi_i)\right] \tag{4.19}$$

where, each crisp variable in N is converted into vague concept with different fuzzy sets, $\chi_i^{'}$ represents calculated fuzzy value for each input variable respectively, $\mu_{\chi_i}^{MF_j}$ denotes each degree of membership functions that describes vague value of χ_i with appropriate linguistic value in specific subinterval. Common curves of membership function include triangular, trapezoidal and Gaussian curve. In our system, Gaussian curve is selected for the membership functions of PER, delay and energy consumption because of its smooth distribution. On the contrary, rectangular waves curve with flat top and unsmooth feature, is selected as specifying membership function with non-fuzzy set which is suitable for the crisp integer of active channel numbers.

After fuzzification step, fuzzy inference scheme plays the rule of system decision making. Combine with the IF-THEN rule base, fuzzy inference process formulates logical operation and constitutes the mapping from fuzzy sets to fuzzy output variable. A typical IF-THEN fuzzy rule consists of several antecedent statements with Boolean logic relationship and consequence statement, can be express as

$$\begin{aligned} R^{(l)} : & IF\ \chi_1\ is\ f_{\chi_1}^l\ and(or)\ \chi_2\ is\ f_{\chi_2}^l\ and(or)\ \cdots\ and(or)\ \chi_4\ is\ f_{\chi_4}^l, \\ & THEN\ output\ is\ f_{\omega}^l. \end{aligned} \tag{4.20}$$

Here $R^{(l)}$ denotes one of the fuzzy rules which is contained in fuzzy rule base. $f_{\chi_i}^l$ represents different fuzzy linguistic characteristics (ex. low, middle, high) that correspond to membership function in specific range of crisp input variable, respectively. As the output of each rule, consequence value f_{ω}^l is linguistic fuzzy set represented by membership function for implication process.

$$\mu_l = \mu_{f_{\chi_1}^l}(\chi_1) * \mu_{f_{\chi_2}^l}(\chi_2) * \mu_{f_{\chi_3}^l}(\chi_3) * \mu_{f_{\chi_4}^l}(\chi_4) * \mu_{f_{\omega}^l}(\omega_l) \tag{4.21}$$

According to the membership function of fuzzy linguistic characteristics which is defined in the rule $R^{(l)}$, corresponding antecedent and consequent numeric value are applied for implication process, as shown in equation 4.21. $\mu_{f_{\omega}^l}(\omega_l)$ is the membership function of consequence, $\mu_{f_{\chi_i}^l}(\chi_i)$ denotes the single membership function value corresponding to the fuzzy linguistic characteristic $f_{\chi_i}^l$ in rule $R^{(l)}$, on the basis of crisp input value PER, delay, energy consume and active channel numbers respectively. Considering the operation overhead on the sensor node, a simplify minimum function is accepted for implication process in our system. The membership function of output fuzzy set is reshaped via minimum function. Figure 4.5 represents the typical procedure of rule in fuzzy logic system.

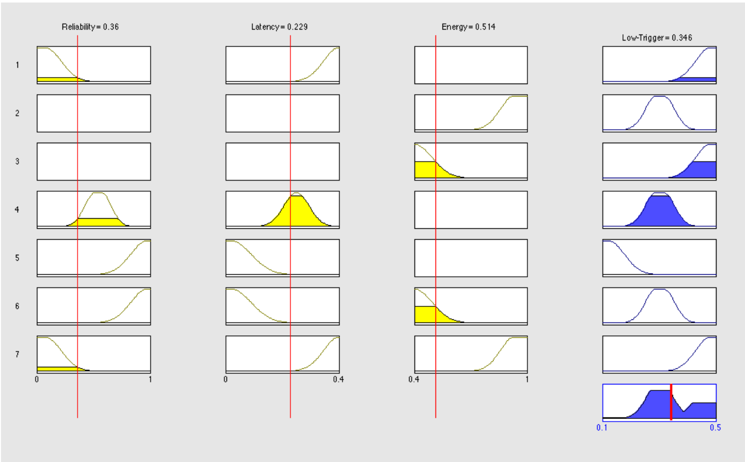

Figure 4.5: Typical procedure of rule in fuzzy logic system

All fuzzy rules and implication processes are executed in parallel. For the defuzzification step, all outputs of fuzzy rule base are aggregated into one crisp output variable $\widehat{\omega}$, that can be approximated by equation 4.22:

$$\widehat{\omega} = \left[\sum_{l=1}^{K} \omega_l \mu_l\right] \Big/ \left[\sum_{l=1}^{K} \mu_l\right] \tag{4.22}$$

where, K represents dimension of fuzzy rule base, μ_l is the output value of fuzzy inference scheme in each rule. As a popular defuzzy method, centroid defuzzifier is applied in here that centre point of aggregated membership function is calculated as the final result of crisp output variable. Figure 4.6 illustrates a discrete gradient example of output values ω^{low}, under the impact of rule set $R^{(l)}$, high reliability and low latency results decreasing trend on low trigger value ω^{low}.

As a consequence, two independent fuzzy logic systems schedule the mapping from input information to output threshold variables for ω^{low} and ω^{high} separately. In the t-th active state s_t, before the action of multi channel allocation process, relevant information is extracted from the physical layer and MAC layer for FLS. Taking uncertainty feature of WSNs into consideration, the moving average observation of last $\tau = 5$ state periods is calculated for the estimation of tendency in reliability, latency and energy overhead. If the

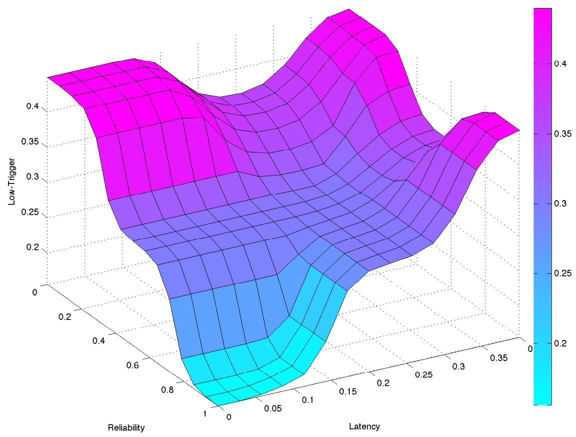

Figure 4.6: Discrete gradient of fuzzy logic output ω^{low}

crisp value is within the range of input space interval N for $FLS(\omega^{low})$ or $FLS(\omega^{high})$, FLS action is executed for the evaluation of $\widehat{\omega}_{t+1}^{low}$ or $\widehat{\omega}_{t+1}^{high}$ in next state s_{t+1}. On the other hand, if input value is out of scope, $\widehat{\omega}_{t}^{low}$ and $\widehat{\omega}_{t}^{high}$ remain unchanged. On condition that only single channel is active currently, there is no need to update $\widehat{\omega}_{t}^{low}$, only $FLS(\omega^{low})$ will be executed. If fully available channels are activated, only $FLS(\omega^{high})$ will be executed. The regulation of fuzzy logic system control is illustrated in Algorithm 2.

4.3 Performance Simulation

The WSN is considered 5 to 20 sensor nodes in different simulation scheme, every node is defined as variable channels from default single channel to multi channels which is up to maximum 4 orthogonal channels. The size of application frame payload is 121 *bytes*, the size of overhead frame is set in 6 *bytes*. For energy consumption based on default single channel in WSN, measurement value [121] is used for estimating the power consumption in each state of simulation, as shown in table 4.1.

4.3.1 Simulation schemes

Single channel with default parameter SC_DP is evaluated as benchmark scheme performance. For MCDB algorithm test, multi-channel dynamic bandwidth with constant threshold MCDB_CP evaluate performance of channel assignment algorithm with the tunable threshold parameter ω^{low} and ω^{high}.

Algorithm 2: Fuzzy Logic Control

Input: $\overline{PER}_t$: Average packet error rate in last τ observations; $\overline{D}_t$: Average time delay in last τ observations; $\overline{E}_t$: Average energy consumption in last τ observations. $n_{c,t}$: Current active channel numbers; $\widehat{\omega}_t^{low}$: Current low threshold of available bandwidth ratio; $\widehat{\omega}_t^{high}$: Current high threshold of available bandwidth ratio.

Output: $\widehat{\omega}_{t+1}^{low}$: Low threshold of available bandwidth ratio for s_{t+1}; $\widehat{\omega}_{t+1}^{high}$: High threshold of available bandwidth ratio for s_{t+1}.

$N \leftarrow (\overline{PER}_t,\ \overline{D}_t,\ \overline{E}_t,\ n_{c,t})$
if *Input values are in the range of* N **then**
  if $n_{c,t} > n_{cmin}$ *and* $n_{c,t} < n_{cmax}$ **then**
    $\left\{FLS(\omega^{low}), FLS(\omega^{high})\right\} \leftarrow N$
    $\widehat{\omega}_{t+1}^{low} = FLS(\omega^{low})$
    $\widehat{\omega}_{t+1}^{high} = FLS(\omega^{high})$
  else if $n_{c,t} = n_{cmax}$ **then**
    $FLS(\omega^{high}) \leftarrow N$
    $\widehat{\omega}_{t+1}^{high} = FLS(\omega^{high})$
    $\widehat{\omega}_{t+1}^{low} = \widehat{\omega}_t^{low}$
  else if $n_{c,t} = n_{cmin}$ **then**
    $FLS(\omega^{low}) \leftarrow N$
    $\widehat{\omega}_{t+1}^{low} = FLS(\omega^{low})$
    $\widehat{\omega}_{t+1}^{high} = \widehat{\omega}_t^{high}$
  end
else
  $\widehat{\omega}_{t+1}^{low} = \widehat{\omega}_t^{low}; \quad \widehat{\omega}_{t+1}^{high} = \widehat{\omega}_t^{high}$
end
return $\widehat{\omega}_{t+1}^{low}, \quad \widehat{\omega}_{t+1}^{high}$
$s_t \leftarrow s_{t+1}$

Three pairs of trigger threshold are evaluated in MCDB_CP for the analysis of performance dissimilarity: MCDB_CP1 with parameter ω^{low}=0.1, ω^{high}=0.7; MCDB_CP2 with parameter ω^{low}=0.2, ω^{high}=0.7; MCDB_CP3 with parameter ω^{low}=0.3, ω^{high}=0.8. Furthermore, MCDB_FLS presents the proposed model with reinforcement learning module and fuzzy logic system, which is compared with other benchmark solution on the performance of energy efficiency. For the fundamental simulation scenario in performance analysis, application load for each node evolves in 80 simulation state periods. The basic test scenario can be divided into two segments. Application load increases and reaches the

Table 4.1: Simulation Parameters

Parameter	**Value**	**Parameter**	**Value**
Node numbers	5-20	$L_{overhead}$	6 $bytes$
macMinBE	3	$L_{application}$	121 $bytes$
macMaxBE	5	E_{tx}	30 mW
macMaxCSMABackoffs	4	E_{rx}	40 mW
macMaxFrameRetries	3	E_{CCA}	40 mW
MinChannelNumbers	1	E_{idle}	0.8 mW
MaxChannelNumbers	4	n_{cmax}	4

peak at $8kbps$ each node, which allows us to mimic the saturated application load in terms of stress test of actual networks, then application load decreases with Symmetrical trend until idle network condition. Instant performance of MCDB on QoS and energy efficiency are evaluated based on test sequence. Additionally, average performance metrics based on test sequence are calculated to investigate the impact of different sensor nodes on QoS degradation and energy expenditure.

4.3.2 Performance metrics

For the performance evaluation of MCDB mechanism, four performance metrics are simulated:

- P_{tf} The probability of transmission attempt failure due to congestion and channel error which indicates the accuracy of sensed data transmission.
- P_{bf} The probability of CSMA/CA backoff process failure due to CCA procedure or retransmission attempt limitation which indicates the overall degradation of performance on MAC layer.
- **Reliability** The probability of successfully messages transmission that indicates the data accuracy.
- **End-to-end delay** This metric indicates the average elapsed queuing time to receive a frame which is an important metric in respect of quality of service.
- **Energy consumption** E^* energy model in previous subsection is used to indicate the performance of efficiency in energy usage. Instant energy consumption in simulation sequence and average energy consumption in terms of different number of sensor nodes are evaluated separately.

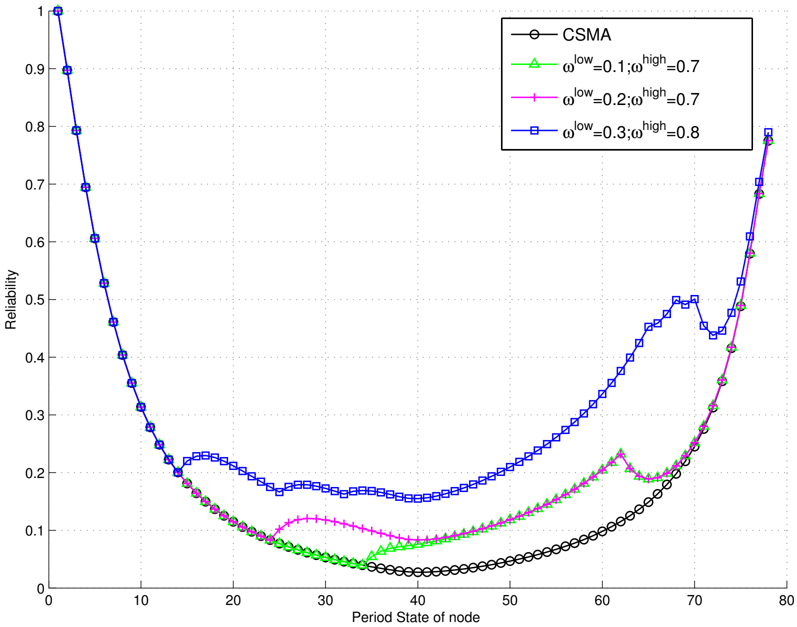

Figure 4.7: Instant network reliability in sequence

4.3.3 Simulation result

Analysis of QoS

In the first part, two performance metrics *Reliability* and *Latency* is considered in order to indicate the instant data accuracy and average elapsed time in network. 10 sensor nodes was considered in single-hop star network.

The performance as a function of reliability is shown in Figure 4.7, the results confirm the previous presumption. Along with the increase of application load, the system global reliability of SC_DP has a conspicuous degradation due to the increasing possibility of retransmissions, back-off mechanism failure, collision and channel error. Increasing flow accompanied the additional bandwidth overhead due to MAY layer procedure cause the result of limited available bandwidth.

Depicted from Figure 4.7, the performance of MCDB_CP indicate a improvement in the range of saturate application flow condition. If the estimation ratio between proactive available bandwidth and current maximum bandwidth is in excess of the threshold value ω^{low} or ω^{high}, the channel allocated algorithm will be triggered and executed. Different threshold parameters has different susceptibility of trigger, MCDB_CP3 performs the best in reliability compared

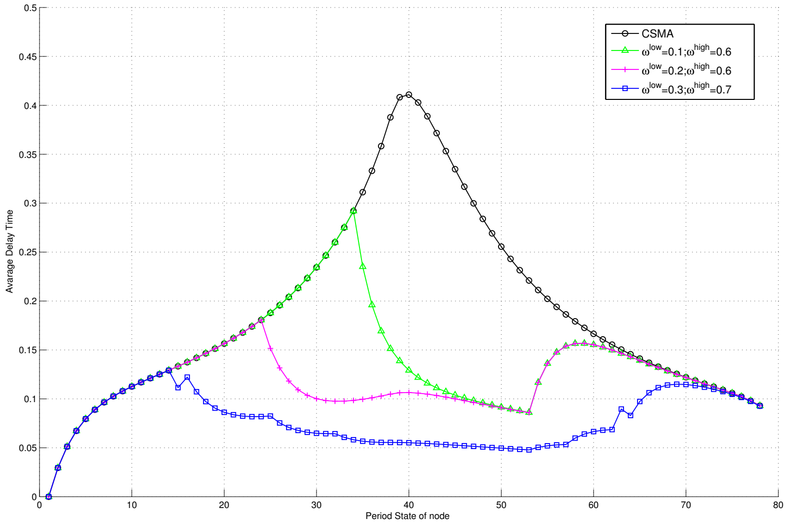

Figure 4.8: Instant network delay in sequence

to MCDB_CP1 and MCDB_CP2.

Figure 4.8 indicates the performance of average latency in different schemes. Without the consideration of transport delay on PHY layer, only elapsed delay due to CSMA-CA procedure is calculated in this figure. Saturate application flow causes congestion on MAC layer processing, the retransmissions will trouble the delay time, higher possibility of CCA failure also increase the additional elapsed time overhead on clear channel assessment. Obviously, MCDB_CP3 achieves the lowest latency value irrespective of the fluctuating application load.

In terms of data accuracy with different network size, Figure 4.9 shows the error probability p_{bf} and p_{tf} of MCDB respectively. In each baseline stress test scenario, saturate application load increased the probability of backoff process attempt failure and channel interference. Additionally, P_{bf} and P_{tf} caused extra bandwidth overhead due to MAY layer procedure. It also should be indicated that high density sensor nodes signifies less data accuracy and more congestion under the baseline stress test scenario. With the MCDB approach, multi-channel allocated algorithm was triggered in the range of saturate packet rate segment which estimated ratio of available bandwidth exceed the range of ω_{low} and ω_{high}. Different pairs of threshold parameters result different susceptibility of trigger. Obviously, MCDB_CP2 overperforms other schemes because of the better threshold parameters.

Figure 4.10 shows the end-to-end delay of MCDB and benchmark schemes.

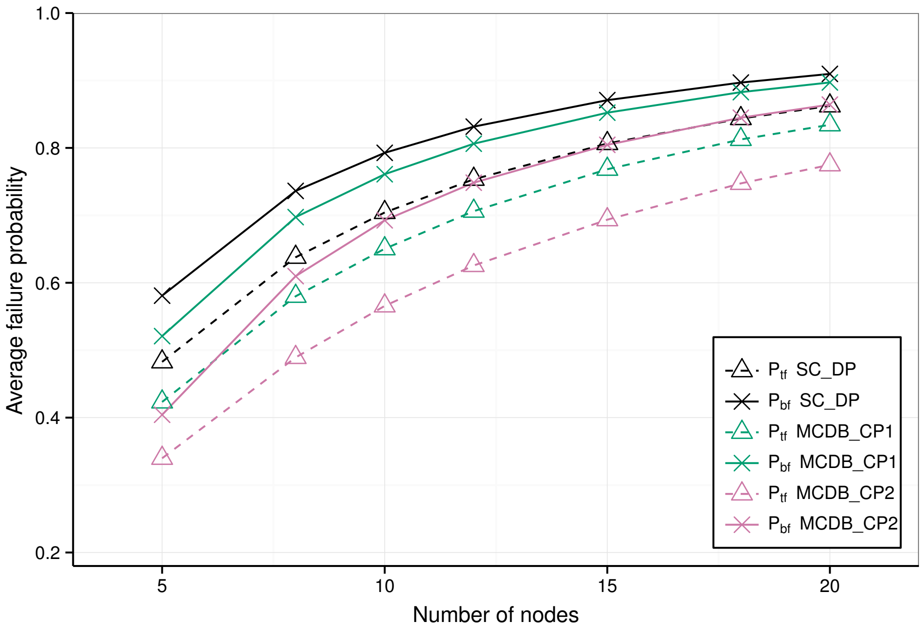

Figure 4.9: P_{tf} and P_{bf} of different sensor nodes

Saturated packet rate causes congestion on MAC layer processing, the retransmissions will trouble the delay time, higher possibility of CCA failure also increase the additional elapsed time overhead on clear channel assessment. It can be observed that the average end-to-end delay of MCDB_CP3 is 36.9%, and 59% less than MCDB_CP1 and SC_DP respectively. The results obviously indicates that multi-channel transmission technique achieve better spectrum utilization and channel capacity. MCDB could get benefit from Quality of Service performance in high data flow applications, which effectively improve overall throughput and alleviate congestion. Next, based on the enhancement of Quality of Service, we analysis the performance of energy consumption in MCDB_FLS.

Energy Consumption

For the performance evaluation of energy consumption, we test MCDB_FLS which is fully train through self-learning iterations and integrated with fuzzy logic system. Performance results are compared with MCDB_CP1 and MCDB_CP3 in different parameter sets, respectively. It can be observed from Figure 4.11, 4.12 that the degradative performance of system reliability causes additional energy expenditure and low conversion efficiency from total energy consumption to actual network throughput. In MCDB_CP, nodes try to minimize the power

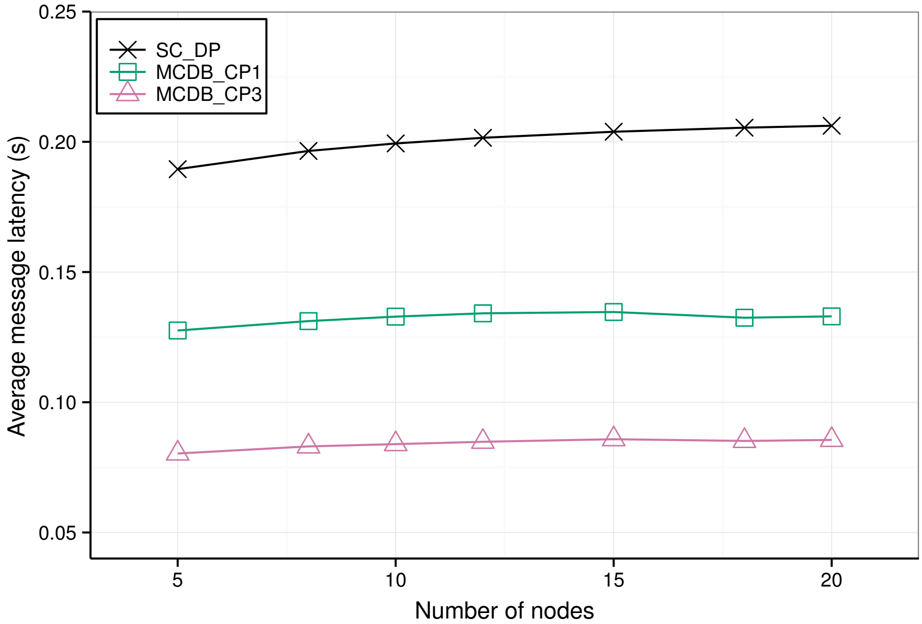

Figure 4.10: End-to-end delay of different sensor nodes

consumption in saturated throughput segments. However, MCDB_CP does not converge to benchmark scheme in several segments of sequences no matter what bandwidth threshold parameter sets MCDB_CP predefined. Stationary scheme for channel assignment admission control hardly make a trade-off between QoS and energy convergence. It can be noted that MCDB_FLS provides better results in instant E^* compared with benchmark results. The bandwidth threshold values for multi-channel allocation admission control are assigned dynamically with fuzzy logic-based estimation approach. Energy expenditure of MCDB_FLS successfully converges to default performance of benchmark scheme for all segments of test sequences.

Figure 4.13 depicts the impact of the size of network on the performance of energy consumption for each node. MCDB_CP2 consumes more energy compared with benchmark result when there are less than 10 nodes. Because of the small size of network, collision between sensor nodes is limited in highly loaded case, performance improvement is inconspicuous compare to the excess energy consumption due to multi-channel parallel transmission. On the other hand, when the network is getting congested, MCDB_CP1 has performance degradation on average E^* due to the conservative parameter setting. As we can see from result, static multi-channel algorithm could not guarantee good performance both on QoS and Energy consumption for different environments. After sufficient iterations of self-learning, nodes learn appropriate pattern from

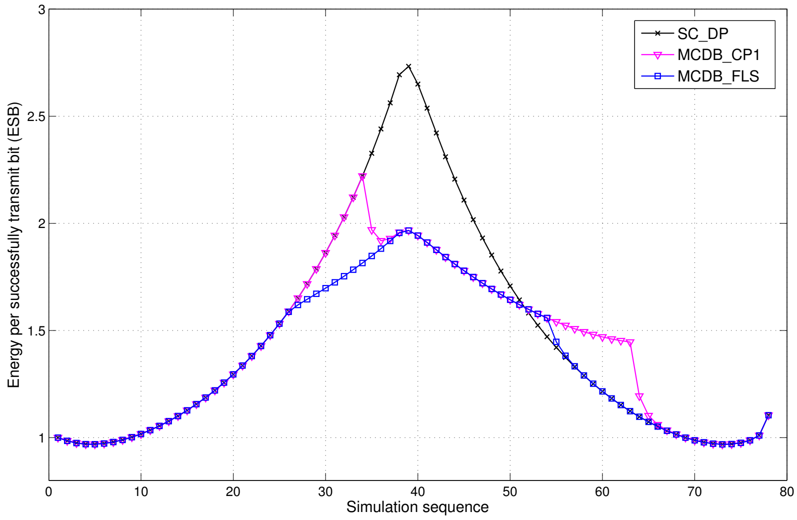

Figure 4.11: Instant effective energy consumption of different schemes

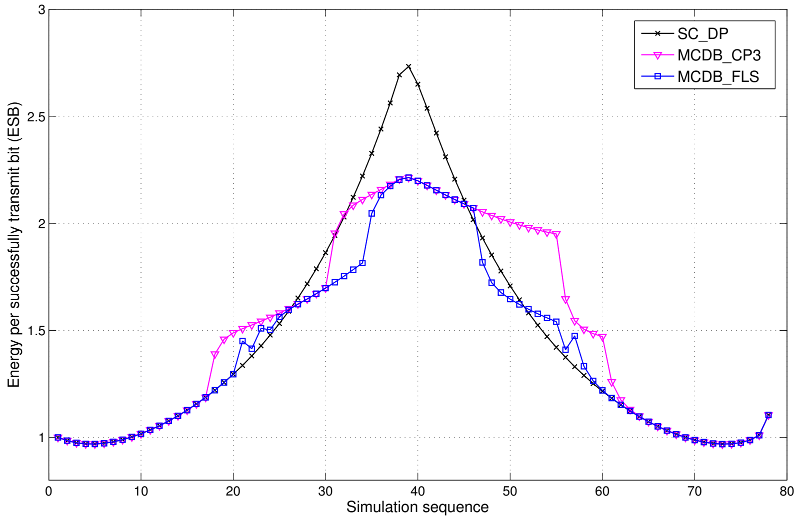

Figure 4.12: Instant effective energy consumption of different schemes

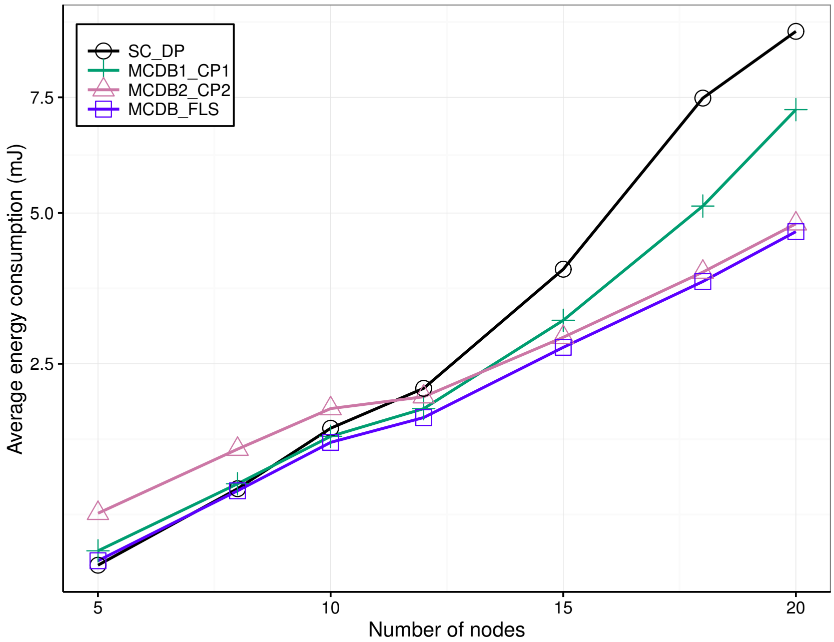

Figure 4.13: Average effective energy consumption of different sensor nodes

environment and allow us to select best action which could acquire best rewards $r(a_t \mid s_t)$ we predifined initially. Furthermore, MCDB_FLS benefit from the dynamically bandwidth threshold control $\widehat{\omega}^{low}$, $\widehat{\omega}^{high}$ in combined fuzzy logic system. MCDB_FLS with fully optimization provides the best E^* performance results under different network size and successfully make a trade-off between the QoS improvement and energy efficiency. Self-adaptive system decision allow sensor node to learn appropriate pattern from varying network environment, deploy multi-channel resource in a smart way .

4.4 Conclusion

In this chapter, we proposed a dynamic multi-channel assignment mechanism MCDB based on the requirement of efficient transmission over WMSNs. Parts of this work has published in [122]. Passive available bandwidth estimation model is used in MCDB as the global performace indication of sensor nodes. The impact of control message, back-off and channel constraint on additional bandwidth overhead is estimated. Reinforcement learning-based method allow us to deploy multi-channel resource in a smart way. Furthermore, fuzzy logic-based bandwidth trigger threshold control module is proposed in order to find the optimal values for channel allocation function. Simulation result verifies that MCDB_FLS successfully improve the performance to make trade-off between energy efficiency and dynamic QoS requirement.

Chapter 5

Incremental Learning with Deep Q-Network for Enhanced Multi-Channel Allocation Strategy

5.1 Introduction

In the previous chapter, we proposed a reinforcement learning method on the basis of look-up table for dynamical multi-channel assignment mechanism. The iterative update of Q function $Q(s_t, a_t)$ follows the rule called Bellman equation, as expressed in Equation 5.1. The states and actions set performs transition procedure from one state to another by observation sequence, make up a Markov decision process. The Markov decision process only perform based on Markov assumption, which the probability of next state s_{t+1} relies only on current state action pair (s_t, a_t) instead of preceding state action memory.

In practice environment, the basic reinforcement learning has a certain disadvantage along with the increasing complexity of wireless sensor network environment. This is caused by separate estimated action-value function for each sequence [123]. Application-based wireless network may generate high correlated environment observations in time sequence, which has inherently risky to generate bias or diverge of learning strategy.

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha \cdot \left(r(a_{t+1} \mid s_{t+1}) + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right) \quad (5.1)$$

In real network environment, large number of features increases the dimensions of hyperparameter set. For tabular-based reinforcement learning, the combination of large actionable states, terminal states and possible actions will generate higher number of possible Q-values. In this case, state-action space is much too large to store in a table. It cause the problem to store and deal with large scale dataset from table structure, specially for wireless embedded device with limited computational resources.

Besides, if the observations of wireless network and proposed possible actions are continuous values in terms of elaborate control tasks, it's difficult for tabular-based reinforcement learning to handle continuous features with computational efficiency. In order to perform behavior and system control directly from the raw environment observation and predifined target value, we propose a multi-channel deployment control mechanism by deep Q-network reinforcement learning approach (DQMC). Through the effective design of DQMC, the deep-RL model evaluated appropriate action and motivated system to output multi-channel allocation commands based on the instant observation and performance of network environments. The target of DQMC is try to learn pattern from wireless network environment, which allow nodes to make smart actions on channel allocation control and achieve multi-channel communication with energy efficiency.

5.2 Multi-channel Allocation Approach based on Deep Q-Network (DQMC)

In the multi-channel environment of wireless network, several research has proved that network performance has tight relationship with the influence of changing external environment. In terms of multi-channel deployment tasks in wireless multimedia applications, the complicated environments make the anonymous relationship between observations and control targets. Thus, we introduce the deep Q-network reinforcement learning method, training neural network-based structure to learn appropriate pattern and mapping from continuous raw network features to discrete output actions which is defined as number of deployed channel. The main structure of deep Q-network multi-channel allocation (DQMC) is illustrated in Figure 5.1.

For the wireless multimedia network environment, we consider four raw observations as the features for the development of deep Q-network reinforcement learning: instant information flow Φ, propagation distance d, number of nodes in contention window N_n and approximate average white noise N_0 from measured signal noise ratio. Raw observations are retrieved and collected by OpenAI Gym [124] after feature extraction and data clearing process. OpenAI

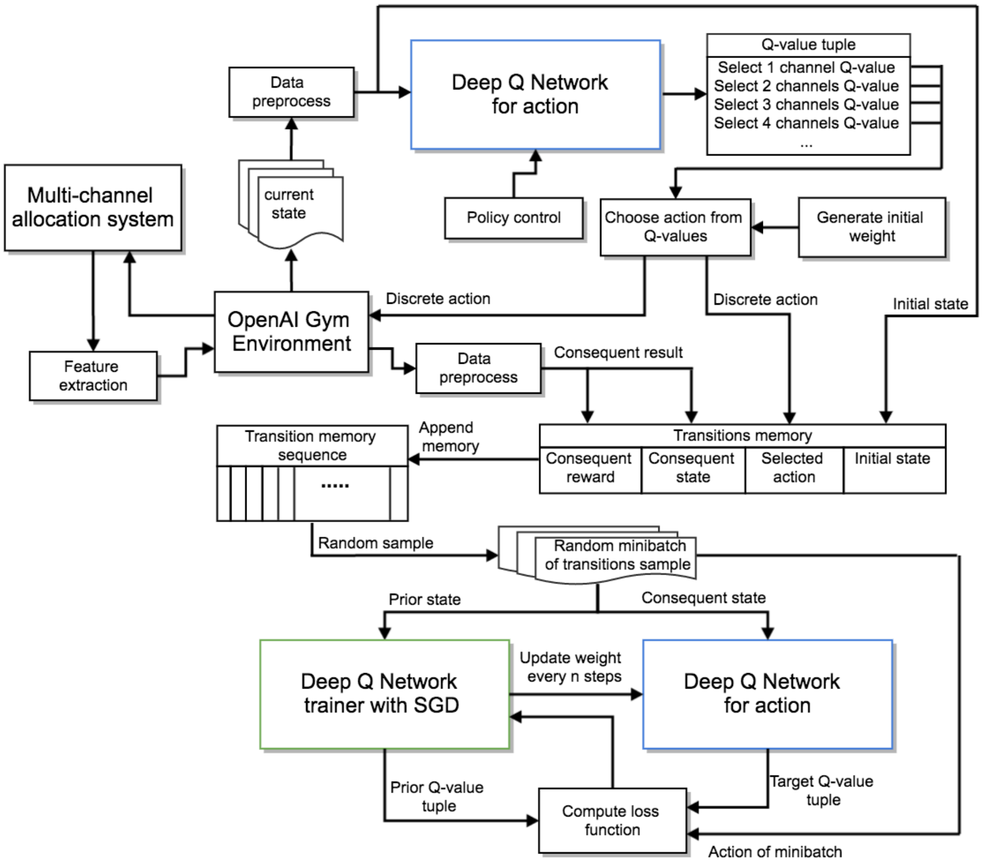

Figure 5.1: Reinforcement learning of multi-channel allocation with deep Q-network (DQMC)

Gym is a toolkit which include basic framework of reinforcement learning environment, is compatible with neural network library, suach as TensorFlow [125] or Theano [126]. The gym environment contains a basic agent-environment loop, which is formalized as a partially observable Markov decision process (POMDP) [127]. We extend function of gym to build a bridge between wireless network observations and reinforcement learning algorithm. In each iteration, gym performs step function which should returns several values: observations of environment, amount of reward achieved by previous action, boolean flag to indicate the reset trigger of environment because of completed episode and additional informations for special learning algorithms.

In order to appropriately train learning model by collected features in iteration t, the raw data are preprocessed by feature engineering procedure, such as normalization, discretization and one-hot-encoding. Processed features are formalized as current state s_t which could be trained by the deep Q-network. Policy control offer different reinforcement learning policies to adjust strategy for action selection. Deep Q-network output available discrete actions space which formed as Q-value tuple with each possible action. In this chapter, we consider actions as the number of deployed multi-channel for next iteration of data transmission. The node acts with maximum 4 channels bonding configuration in each iteration. DQMC choose an action a_t from Q-value tuple with initial weight, then send a_t back to gym which configure node to deploy multi-channels with action a_t. Consequently, we can acquire posterior results from environment including consequent reward r_{t+1} and consequent state s_{t+1}, which is combined with prior action-state pair into a transition memory tuple. The collected transition memories will be trained by neural network of deep Q-network trainer within the procedure called experience replay. DQMC could acquire knowledge and learn from the memories by itself, get benefit from both explored and exploited experiences, then update the neural network parameters by every n action steps to achieve better action strategy.

Deep Q-Learning

For the deep reinforcement learning algorithm in DQMC, we accept deep Q-network (DQN) [123], [128] and double deep Q-network (DDQN) [129]. Deep neural networks is trained to represent the components in traditional reinforcement learning problems, which could deal with continuous state-action space with deep Q-function. The optimal Q-values $Q^*(s,a)$ follow the Bellman equation as defined in Equation 5.2:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s^{'},a^{'})|s,a\right] \tag{5.2}$$

The object in DQN is to learn the parameter vector w which contains all the weights of Q-Network that could update current Q-value towards to the target optimal value function $Q^*(s,a)$, as shown in Equation 5.3.

$$Q(s,a,w) \approx Q^*(s,a) \tag{5.3}$$

Thus, with the structure of $Q(s,a,w)$ in DQN, which can generate all the discrete action Q-values $Q(s,a_1,w),...,Q(s,a_n,w)$ for a specific state at once with optimal computational efficiency.

Using the same Q-values for both train and evaluated action, which has the risk of overestimation on action selection, resulting overfitting learning model. Based on the original DQN, two action-value functions are trained separately. Each experience will pick one of the two Q-value functions randomly for updating. The target optimal Q-value can be written as:

$$Q^*(s,a,w,w^{'}) = \mathbb{E}_{s'}\left[r + \gamma \max_{a^{'}} Q(s^{'}, \operatorname*{argmax}_{a^{'}} Q(s^{'}, a^{'}, w), w^{'})|s,a\right] \quad (5.4)$$

For each iteration of Q-value updating, one set of Q-network weight w is used to determine the policy and select actions. Another set of Q-network weight $w^{'}$ is used to fairly evaluate and determine its Q-value with the action which is made by first set of Q-network weight w. The weight set for Q-value evaluation could be updating symmetrically by switching the two weight sets under a fixed probability, which allow learning model has ability to substantially reduce the overestimation, perform faster and reliably training.

The neural network of DQMC is designed by the sequential model of Keras library [130], workflow is defined as shown in Figure 5.2. For the initial input state of s_t, we put a flatten input layer to spread the stacked frame into a flat frame with shape (None,40), which could be connected with hidden layer. The first hidden layer connects input layer with 164 units and applies ReLU activation function. The initial weights of hidden layer is assigned by LeCun uniform initializer [131], which draws samples from a uniform distribution within $[-\sqrt{3/n_{in}}, \sqrt{3/n_{in}}]$, where n_{in} is the number of input units in the weight tensor. The second hidden layer contains 150 units and applies ReLU activation function. The final output layer is fully-connected linear activation layer with a single real-valued output for each valid action.

Explore Policy

In reinforcement learning, the long term object of an agent is to deal with optimized action from all possible states of environment. Unlike those collected training dataset with labels in supervised learning field, agent must acquire experience from environment through each action. A good policy is necessary that agent could obtain experiences for self-learning procedure efficiently and make the tradeoff between exploration and exploitation. Ideally, exploration procedure is an indispensable part of learning procedure to explore environment until the convergence state, which agent acquire enough information to make optimized action. So policy π is designed as mapping sequences to actions or distributions over actions. It's important to adjust the amount of an agent's time should be spent exploiting its existing known-good policy and the time should be spent to encourage agent to explore possible better actions.

The simple and effective solution for the above problem is $\varepsilon - greedy$ explore policy. An agent has probability ε to choose a random action as the procedure

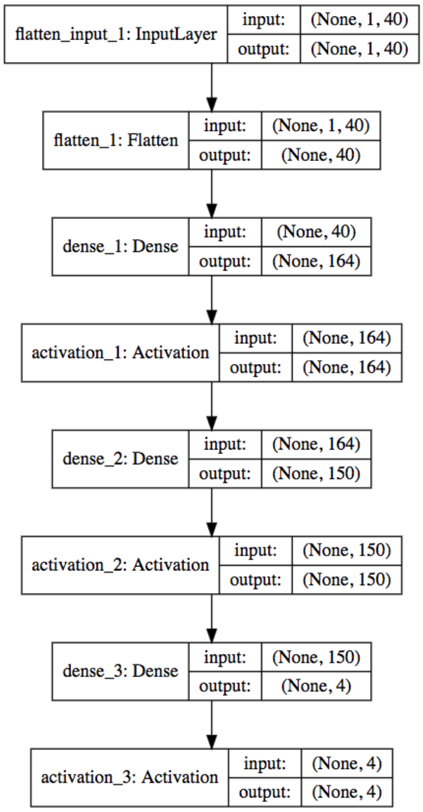

Figure 5.2: Structure of neural network layers

of exploration. Additionally, linear $\varepsilon - greedy$ exploration deploy changing values of ε. In the beginning of learning, agent makes almost random actions with the ε close to 1 in order to explore the state space widely, then linearly decrease ε to a fixed exploration rate.

Besides, Boltzmann policy is also considered as a different strategy with random action-based policy. Boltzmann policy retrieves and exploits all the information from estimated Q-values in neural network, which select action based on weighted probabilities instead of probability take random actions. The probability output of Boltzmann policy is computed as follow:

$$P_t(a) = \frac{exp(q_t(a)/\tau)}{\sum_{i=1}^{n} exp(q_t(i)/\tau)} \tag{5.5}$$

Where action probability $P_t(a)$ is calculated with softmax equation, τ is anneal parameter which controls the spread of softmax distribution. In stead of equally assigning probability for non-optimal actions in ε-based policy, Boltzmann policy compute the weight of each action by relative value. Boltzmann policy allows agent to explore potential action from sub-optimal actions which decrease the probability of wasting actions from random explorations.

Experience replay

Reinforcement learning has potential risk to be unstable when using neural network to approximately represent Q function value. The environment observations in time sequence may cause high correlations, small updates of Q function by correlated data generate the bias of policy and data distribution and correlation between Q function value and target values $r + \gamma \max_{a'} Q^*(s', a')$.

In order to solve the problem which is special obvious in wireless network tasks, experience replay train neural network from randomized sampling-based experience records. As shown in Figure 5.1, in each iteration of learning process, discrete action of multi-channel deployment is selected by neural network. OpenAI Gym accept action a_t and give order to the multi-channel evaluation environment, then give back the posterior result which include consequent state and reward based on selected state-action pair $\langle s_t \rangle$. New experience is acquired as a state-action pair and transitional reward and state in tuple, which is appended into memory sequence D:

$$D_{t+1} \longleftarrow D_t \cup \{\langle s_t, a_t, r_{t+1}, s_{t+1} \rangle\} \tag{5.6}$$

For the memory sequence D of experience replay, past state-action tuples are stored with limited length. In each learning episode, a set of transition samples called mini-batch $U(D)$ are randomly chosen from memory sequence to train the Q-Network. This method solves the problem of high correlations in time sequence observations because it is randomly choosing different state-action records from experience memory, rather than choosing them in order. So high correlation is removed because the observation order of the state-action pairs is shuffled in training process of random mini-batch. Beside, batched replay memories also increase computation efficiency of neural network training.

In the sample random minibatch of transitions $\langle s_t, a_t, r_{t+1}, s_{t+1} \rangle$, if the posterior state s_{t+1} is terminal state of episode, target is equal to r_{t+1}. Otherwise, target Q-value is calculated as $r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a', w^-)$ from Deep Q-etwork for action. Another neural network trains for the approximate Q-value function $Q(s_t, a_t, w)$. The optimal object is to minimize mean square error between the true action Q function and approximate Q function by stochastic gradient descent. Loss function $\mathbb{L}$ is expressed by Equation 5.7.

$$\mathbb{L}(w) = \mathbb{E}_{U(D)} \left[\left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a', w^-) - Q(s_t, a_t, w) \right)^2 \right] \tag{5.7}$$

Where w^- represents the weight vector of network that is used to compute target action-value function in iteration t. Updating w^- by each iteration will cause non-stationary target in gradient descent optimization. Consequently, w^- is only assigned with w for every certain episodes, w^- keeps fixed value during n steps in order to train Q-network efficiently.

Gradient Descent Optimization

In a typical linear neural network, error surface appears a quadratic bowl with a squared error. However multi-layer non-linear nets brings more complicated error surface. Learning algorithm aims to achieve quickly travel in the directions with small and consistent gradients, but small distance in the case of large gradients direction. In real optimization problems, the composition of different subfunctions based on different subsample of dataset cause complex objective functions. Stochastic gradient descent (SGD) is often selected as effective optimization method to process gradient steps on stochastic individual subfunction. Learning process with mini-batches usually performs better because of the computationally efficient in weights updating. In the begin of basic algorithm in mini-batch gradient descent algorithm, initial learning rate is selected properly according to the changing of error. If error reduce consistently and slowly, learning rate should increase; If error get worse or fluctuate, small learning rate should be selected. Turn down the learning rate in the end of each mini-batch in order to eliminate unstable of final weights caused by the variations between different mini-batches. Stop learning rate in the case of error stops decreasing. Optimization methods RMSprop [132] and Adam [133] are considered in this chapter:

RMSprop In RMSProp optimization method, the gradient is divided by a running average of its recent magnitude. As shown in Equation 5.8, RMSprop keep a moving average of the squared gradient for each weight based on the previous value of meansquare v_{t-1}. Exponential decay rate β_2 is assigned close to 1. From Equation 5.9, the weight change stepsize $\Delta\theta_t$ can be expressed with the current gradient value divided by $\sqrt{v_t}$, which decay with stepsize α less than 1. With a larger value of v_{t-1}, the learning stepsize $\Delta\theta_t$ will be closer to zero, weight change is adjusted in terms of the moving average of previous weight change and current gradient.

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \tag{5.8}$$

$$\Delta\theta_t = -\alpha g_t {v_t}^{-1/2} \tag{5.9}$$

Adam In the algorithm Adam, the stochastic optimization only rely on first-order gradients with little memory requirement. Gradient g_t is estimated from stochastic objective function at t from subsequent timesteps $1, ..., T$. Equation 5.10, 5.11 estimate biased first moment and biased second raw moment of the gradient under the control of exponential decay rate β_1 and β_2, respectively. Thus, the moving averages of mean and uncentered variance of gradient is computed. Then, initialization bias correction is performed, result in the bias-corrected estimates $\widehat{m}_t$ and $\widehat{v}_t$. The initial bias $(1 - \beta_2^t)$ is removed, which is derived from initializing exponential moving average with vector of zeros. Finally, the adaptive learning rates is controlled only by the first and second moment of gradients, as expressed in Equation 5.14.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{5.10}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{5.11}$$

$$\widehat{m}_t = m_t / (1 - \beta_1^t) \tag{5.12}$$

$$\widehat{v}_t = v_t / (1 - \beta_2^t) \tag{5.13}$$

$$\Delta\theta_t = -\alpha \widehat{m}_t \widehat{v}_t^{\,-1/2} \tag{5.14}$$

For the deep Q-network of DQMC mechanism, backpropagation performs gradient descent procedure with RMSprop and Adam to train its Q neural network. In the uncertain network environment, Q neural network has ability to deal with non-stationary objectives with little memory requirement and sparse features networks.

5.3 Experiment of Deep Q-Network Multi-channel Allocation (DQMC) and Performance Analysis

5.3.1 Environment hyperparameters

For the evaluation emulator of multi-channel allocation model, we accepted similar parameters with previous chapter, MAC layer parameters is assigned as shown in Table 5.1.

We consider information loads Φ from 0.1 to 20 frames/s, minimum 5 to maximum 20 nodes, average white noise is in the range of 5-20 dB, transmission distance from 5 to 40 meters. Observation space and action space are both assigned as discrete values with length 4. We accept the same training schedule in each experimental session, setting maximum number of action steps for

Table 5.1: Hyperparameters setting of DQMC experiment.

Parameter Description	**Value**
Max channel numbers	4
Min value of the backoff exponent	3
Max value of the backoff exponent	5
Max number of backoffs attempt	4
Max number of frame transmission retries	3
Information loads (Φ)	0.1-20 frame/s
Number of nodes in contention (N_n)	5-20
Transmission distance (d)	5-40 m
Average white noise (N_0)	5-20 dB
Observation space (S)	Discrete(4)
Action space (A)	Discrete(4)
Learning algorithm	DQN, DDQN
Max number of steps to train (n_steps)	20000
Max steps of episode (n_episode_steps)	10
Max reward of each episode	10
Min reward of each episode	0
Target model update rate (ε_w)	1e-2
Batch size (U_D)	32
Experience replay memory size	1000, 2000, 5000
Exploration policy ($\pi(a, s)$)	EpsGreedy, Boltzmann, LinearEps
Optimization algorithm	RMSprop, Adam

training as 20000, maximum action steps is up to 10 for each episode. Selecting right actions in the range of 10 action steps will acquire 10 rewards in each episode. Consequently current episode is terminated, gym will reset environment and update sequence to next episode. otherwise there is no reward earning, action step will repeat until the reward acquirement or the maximum action steps per episode. Instead of using hard update to assign target weight vector w^- for every n-th step as introduced Equation 5.7, we accept soft update policy [134] to smoothly update target model:

$$w_{t+1}^{-} = \varepsilon_w w_t^{-} + (1 - \varepsilon_w) w_t \tag{5.15}$$

Where ε_w indicates the learning rate of deep Q-network from each episode, which was set with a fixed value 1e-3 in this experiment. Fixed batch size U_D is assigned with 32, that is combined with experience replay in memory size 1000, 2000 and 5000, respectively.

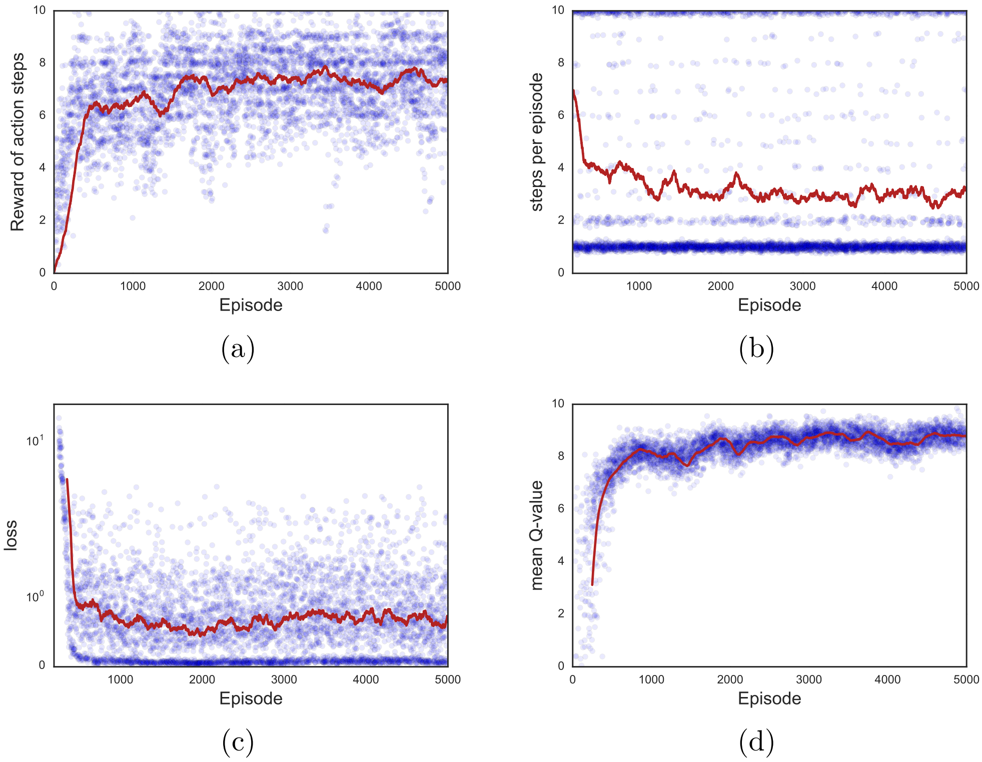

Figure 5.3: Instant (points) and average (line) metrics of DQMC in training session, ε-greedy policy with $\varepsilon = 0.1$ for 20000 steps: (a) reward of each action step (maximum possible reward is 10), (b) number of action steps every episode, (c) loss value in training, (d) mean Q-value.

5.3.2 Experiment and result analysis

Performance analysis in one training session Firsly, We extract information from DQMC training logging of one single experimental session, which is represented by Figure 5.3.

In this training session, DQMC is configured as DQN with ε-greddy policy, 1000 memory size and Adam gradient descent method for neural network training. The blue scatter points indicate instant result in the training sequence, red lines represent the moving average training curve in several logging metrics. From the result in Figure 5.3a, acquired reward for each action increase significantly in first 500 episodes, then slowly growth to a stable reward range. The distribution of instant reward shows that, DQMC presents sort of deviation on rewards accumulation. DQMC implement approximate 5000 episodes in 20000 action steps. As we can observe from Figure 5.3b, the action steps of each episode decrease from 7 to approximate 3. DQMC achieves successful action selection in the first action of each episode from most episodes of training sequence, but also has the experiments that DQMC make wrong decision until the maximum action steps of each episode. Figure 5.3c, 5.3d show the evolution of loss value in neural network training and mean Q-value achieved by DQMC respectively. Experiment results validate the optimal object of deep Q-network reinforcement learning. DQMC could improve its ability during self-learning iterations and allow nodes to make better action decision with more rewards earning.

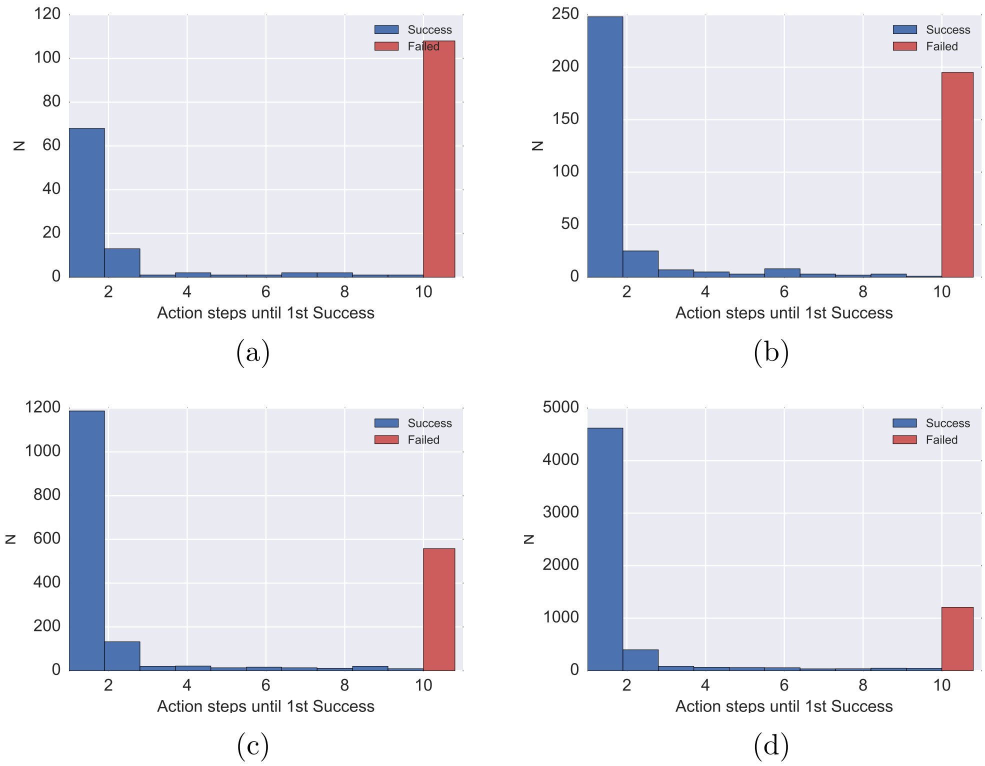

Figure 5.4: Distribution of steps until 1st successful action in each episode: (a) training steps = 200, (b) training steps = 500, (c) training steps = 2000, (d) training steps = 10000.

The distribution of action steps until the deep Q-learning model achieves its reward in each episode are shown in Figure 5.4. Blue bar plots indicate the distribution of action steps in the episodes, which successfully win the reward. Red bar plots represent the amount of failed training episodes. During the first 200 action steps of training (5.4a), DQMC explore the environment and try to make decisions without reliable experience in the initial stage. 60% of run episodes fail to solve the task within 10 action steps of each episode.

Accompanied with the process of training, experience replay could propose more valuable information, DQMC updates weight vector of neural network by several learning iterations. The knowledge acquisition from environment and self-update from experience replay allow DQMC keep gaining sustainable improvement in action selection strategy. Until 10000 action steps (Figure 5.4d), 81% of episodes successfully win the rewards and most of the successful actions is selected in the first action step of each episode.

Figure 5.5: Rewards per action step under different policies in DQMC.

Performance comparison with hyperparameters We implement training experiments on DQMC learning model using the hyperparameters and compare the performance variation between them. Each training session is repeated 10 times with same parameter set and random seeds in order to evaluate the stability of learning model. Hyperparameters of DQMC consist of core learning algorithms DQN and Double DQN with the combinations of policies and gradient descend algorithms.

Figure 5.5 shows the result of reward per action in DQMC learning process. There is no obvious performance gap between DQMC model with DQN and DDQN learning algorithm in instant result. In the initial stage before 500 episodes, as seen from the shaded area, learning models with Boltzmann policy present higher variance than ε-Greedy policy. After that, the learning curves of Boltzmann policy consistently achieve higher values than ε-Greedy policy. Learning model with Boltzmann policy implement 6000 episodes training in the limitation of 20000 action steps, which is higher than ε-Greedy policy with 5000 episodes. The result indicates that DQMC with Boltzmann policy could explore more potential actions from sub-optimal possible actions, decreasing the probability of selecting wasted actions in random exploration policy.

Figure 5.6: Rewards per action step with different gradient descent algorithms in DQMC.

Figure 5.6 shows the result of reward per action with different gradient descend algorithms. DQMC with DDQN learning method achieve slightly better performance than DQN learning method, as we can see from the average darker line over seeds and variance in shaded area. Note that the learning curve of DQMC model with Adam optimization method is always higher than the result of RMSprop optimization method. The bias correction function in Adam allow DQMC to perform more stable and smooth gradient descent procedure during backpropagation, training the Q neural network more efficiently.

Furthermore, the performance is compared between full combinations of hyperparameter set, as shown in Figure 5.7. The cumulative rewards is considered as metric based on full training tasks. Deep Q-learning policy include ε-Greedy policy, linear ε-Greedy policy and Boltzmann policy. Neural network is optimized by RMSprop and Adam algorithm. Memory sequence sizes for experience replay are assigned with 1000, 2000, 5000 respectively.

From the result of cumulative rewards, DQMC model with Double DQN learning approach show better performace than DQN approach, which is evaluated under almost all the combinations of hyperparameter. The symmetrically

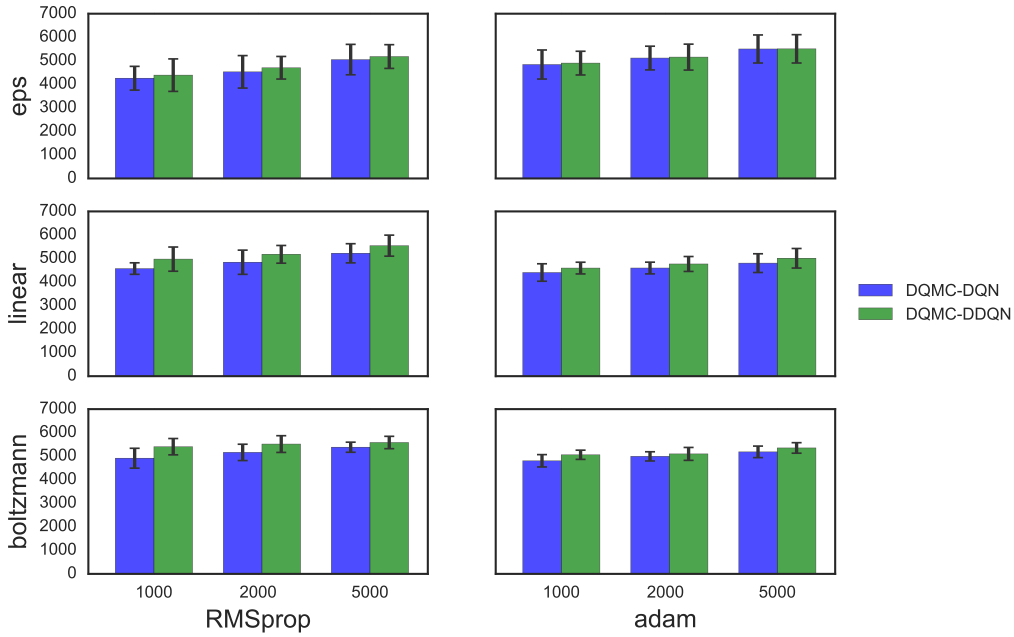

Figure 5.7: Cumulative rewards based on full tasks with different hyperparameters (memory size, policy, gradient descend method).

switch between two Q-value function weight vectors could diminish the probability of overestimation risk in DQN approach using single Q-values function for both training model and evaluated action. Two action-value functions are trained as independent Q-learning model. It's obvious that larger memory size will bring more benifit on overall rewards. More experience is stored in memory sequence that will provide more comprehensive informations for experience replay procedure. In the compared policy methods, Blotzmann policy performs better result at average overall rewards with less variances, comparing with ε-Greedy and linear ε-Greedy policy. linear ε-Greedy policy performs better than fixed ε-Greedy policy. Exploration probability ε linearly decrease in training sequence, which bring dynamically trade-off on exploration strategy and achieve promotion on overall rewards.

Analysis of effective energy consumption In the performance analysis of effective energy consumption, we evaluate multi-channel allocation models in the same event sequence with 10 random seeds. Fuzzy rule-based multi-channel method is evaluated as a performance comparison. DQMC-train indicates the deep Q-learning model is trained from initial weight in each session, DQMC-test loads pre-trained weight vector from h5f file before each session.

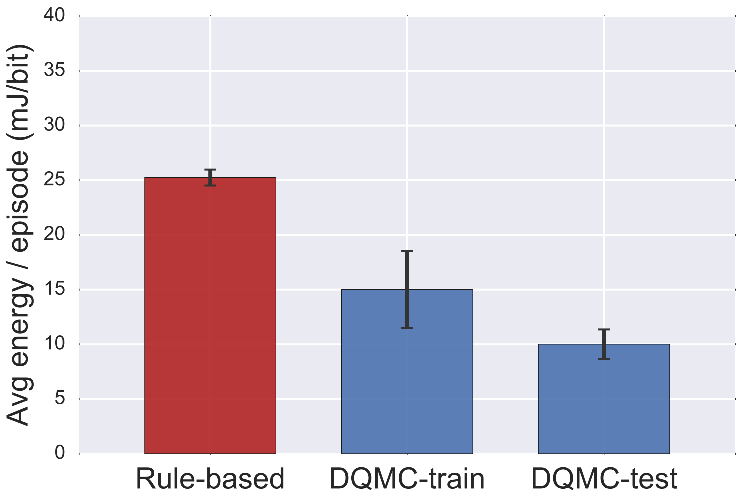

Figure 5.8: Average energy consumption per episode.

From Figure 5.8, deep Q-network learning model allows DQMC to achieve a better performance in effective energy consumption. The proposed DQMC-train model achieves a finial average energy saving enhancement of 40%. However, in the beginning of each training session, DQMC-train has to train a new learning model from initial Q-network, which will expand more energy during knowledge exploration and learning process. Different task sequences generate fluctuation of energy expenditure during learning process. If we loads the pre-trained weight vector into deep Q-network for each session, DQMC-test achieves more stable and optimal energy performance. The performance gap between DQMC-train and DQMC-test indicates the impact of initialized learning process on overall performance of energy consumption. In practical environment, the impact will be magnified which suffers from a increasing complexity environment with high-dimensional attributes.

5.4 Fully Initialization DQMC with Stacking Ensemble Multi-class Classifiers

In previous implementation of DQMC, it has been demonstrated that deep Q-network can be used to improve the accuracy and efficiency of system decision-making in multi-channel deployment tasks. However, in practical wireless network environment, more attributes should be considered in the observations of reinforcement learning structure. Fluctuating network features and reusable network for different applications all increase the complexity of reinforcement learning. As we can see from the previous experiment, the performance of deep Q-learning model was evaluated in sequences with full training tasks which start from initial state. In the early stage of learning process, DQMC is encouraged to explore informations and acquire experiences from unknown environment. The primary learning process results in extra wrong system decisions, which will generate performance degradation and wasting of resources.

Thus, the motivating factors drive us to improve the adaptability of DQMC design. We try to incorporate supervised machine learning techniques into the initial weight of Q-values in deep Q-network learning process. The initial weight fusion structure allows learning system to be more efficient in rapidly environment exploration and experience accumulation from original deep Q-learning, extensible and tolerant to changing environment and different application requirements.

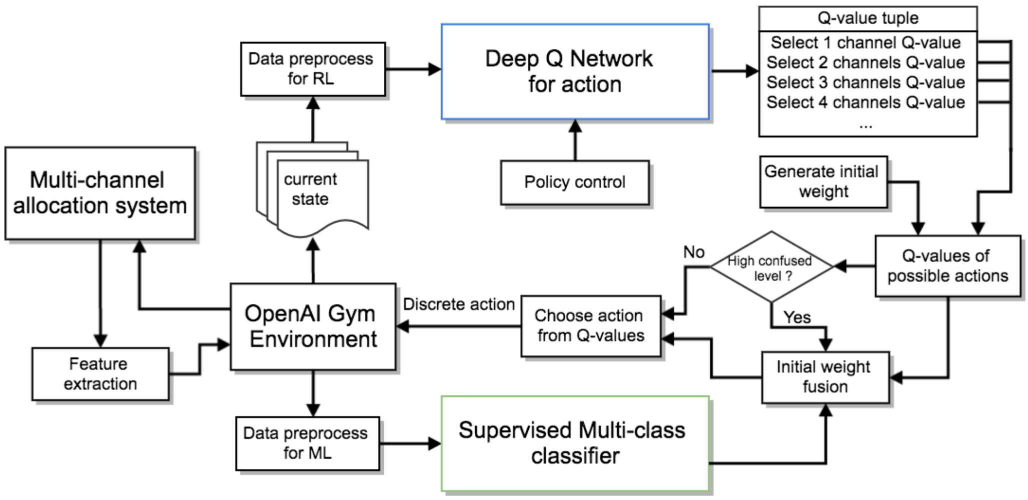

Figure 5.9: DQMC with initial weight fusion using multi-class classifier

5.4.1 Initial weight fusion of Q-function value

In the complex nature of wireless sensor network, the wireless channel tends to be classified by either probabilistic or statistical models. Empirical measurements of inputs and outputs are essential for establishing models to represent the attributes of data. A generalized illustration of supervised learning integrated into DQMC is shown in Figure 5.9. In the original DQMC model, gym environment extracts channel features iteratively as the observations of current state. We use the same feature vectors as the input of supervised machine learning model, using different data preprocessing methods for the requirement of special machine learning algorithms. Because the control objective is discrete action space, the task of machine learning model is defined as multi-class classification. The pretrained model is not used in the entire process of DQMC. Instead, DQMC only use multi-class classifier to perform an approximately classification with an output of predicted probability, which could be incorporated into the original weight of Q-value generated by deep Q-network in initial steps. Confused level indicates the reliability degree of DQMC's decision-making, which is calculated from the Q-values of action space. If confused level is lower than threshold value, it means that DQMC has already updated its Q-values for certain episodes and learned pattern from collected experience, keeping confidence to choose better action instead of random behaviors. We can directly accept the action selected by DQMC. In other words, confused level is high which indicate learning model doesn't have firm opinions because of the less-experience states. In this case, we will make the initial weight fusion between multi-class classifier and Q-values space. The predicted output probability of classifier plays the role of guideline to boost the initial learning process of DQMC. Due to the dynamics of wireless network, classification model must be periodically refreshed with new training examples. In order to avoid the overestimation problem caused by prediction error of classifier, classification model is assigned with a learning rate ε_w to control the influence of initial weight fusion process on finial performance of DQMC.

5.4.2 Multi-class classifier training procedure

training dataset preprocessing Preparing labeled training examples is the preliminary of supervised learning. We run multi-channel evaluation model with random parameters, select the multi-channel mode with best performance of effective energy consumption as the data label. 20000 multi-class labels training dataset is generated and translated into the formats for different machine learning algorithms respectively. 500 training examples are randomly choosed from full dataset to make pairplot visualization on the basis of each feature, as shown in Figure 5.10.

From the distribution of training example, it is difficult to observe the precise relationship between different pairs of features. This also could interpret the defect of rule-based method, it's hardly to design a precise and comprehensive algorithm to exploit all the underlying mapping functions between input and output space manually. After the labeling is complete and full data preprocessing is finished including remove duplicate data, outlier process,

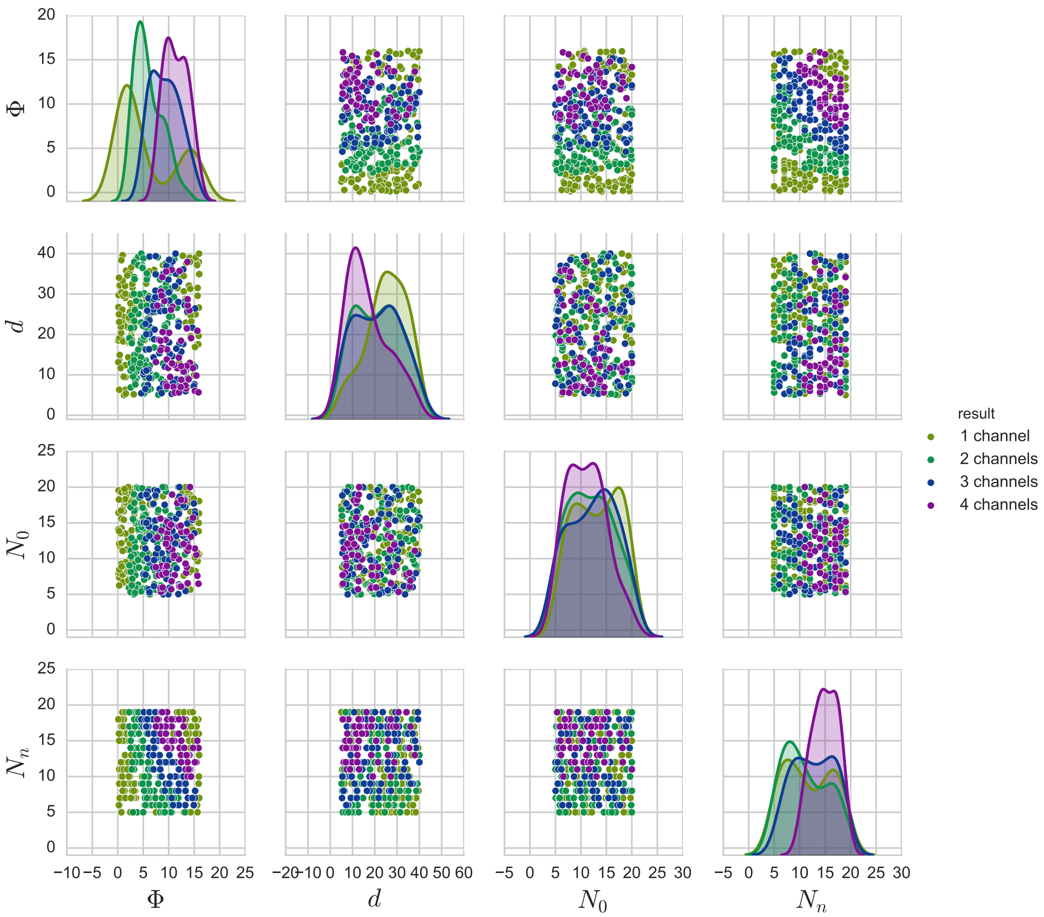

Figure 5.10: Pairplot representing feature distribution of multi-class labeled training dataset.

normalization, etc, the classifiers can be trained.

Stacking ensemble multi-class classifier The core idea of stacking ensemble model is similar to k-folds cross validation to create out-of-sample predictions. out-of-sample method avoid base models to fit all the training examples in the same way, which could cause the problem that second level model will be biased towards the best model in base models. The workflow of stacking ensemble model is illustrated in Figure 5.11. The Stacking ensemble machine learning algorithm can be summarized as follows:

- Initial training data X with shape of $(m * n)$, which contains m observations and n features. Randomly split training data into K equal-size subsets: $\{\mu_1, \mu_2 .... \mu_K\}$.

- Train every base model on $K-1$ subsets and predict on the out-of-sample fold and make the predictions on test dataset

- Each model provides predictions to generate the second level training set $\{\hat{y}^{\mu 1}, \hat{y}^{\mu 2} .... \hat{y}^{\mu M}\}$ with shape of $(m * M)$.

- Learn a classifier from the collection of $\hat{y}^{\mu}$ as the new features of dataset then obtain the finial prediction on test dataset.

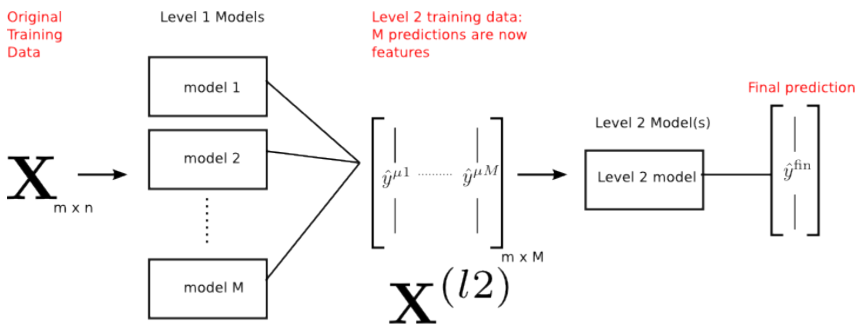

Figure 5.11: Illustration of stacking ensemble machine learning algorithm.

For the classifier selection of base models, the principle is to select models with multiple different basic theories. For example, tree-based model combine with linear-based model will achieve better performance in stacking ensemble model. This can be interpreted that any base model can not achieve high accuracy in all subsets of training dataset. A base model may have a good overall performance but with worse accuracy in several parts of dataset compared with another base model. This is why out-of-sample predictions have a higher chance of capturing distinct regions where each model performs the best to achieve a better model. However, base models with similar algorithms will increase the correlation of performance distributions in same subsets, which could not perform to the maximum potential of stacking ensemble model.

Base models are trained separately with the full procedure of features selections, observations selections, hyperparameter tuning in 10-folds cross validation. Figure 5.12 represents the an example of feature reduction decision investigation on KNN model, evaluate the influence of incorporating feature number on model. Optimal feature number is selected to make the trade off between computational efficiency and performance. Figure 5.13 indicates the learning curve of KNN model with the changing size of training dataset. It can be observed that KNN model require as least 30% size of training dataset to perform sufficiently accurate prediction.

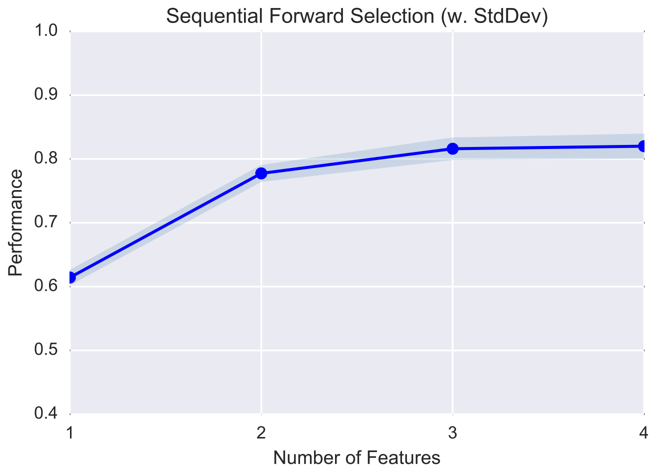

Figure 5.12: Classification performance with changing number of features.

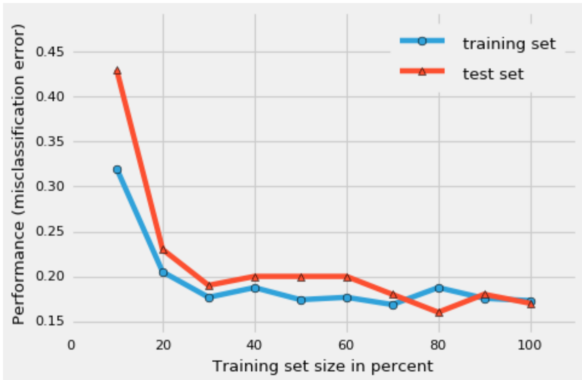

Figure 5.13: Classification performance with changing size of training dataset.

In order for the elaboration of stacking ensemble model, we implement a demonstration of plotting decision regions [135] on the multi-class classifications of base models and stacking ensemble model. As shown in Figure 5.14, 500 random test samples are evaluated based on each base learning model. Decision regions of each class are drawn based on two dimensional features of test dataset. Base models include RBF kernel SVM, random forest, KNN, ExtraTreesClassifier, Gaussain Naive Bayes, which are combined into the final stacking ensemble model. Different base models result in distinguishing performances in sub-regions of dataset because of the different algorithms of base models. Stacking ensemble model combines all the attributes of base models together, being able to discard the dross and select the essence on second level prediction.

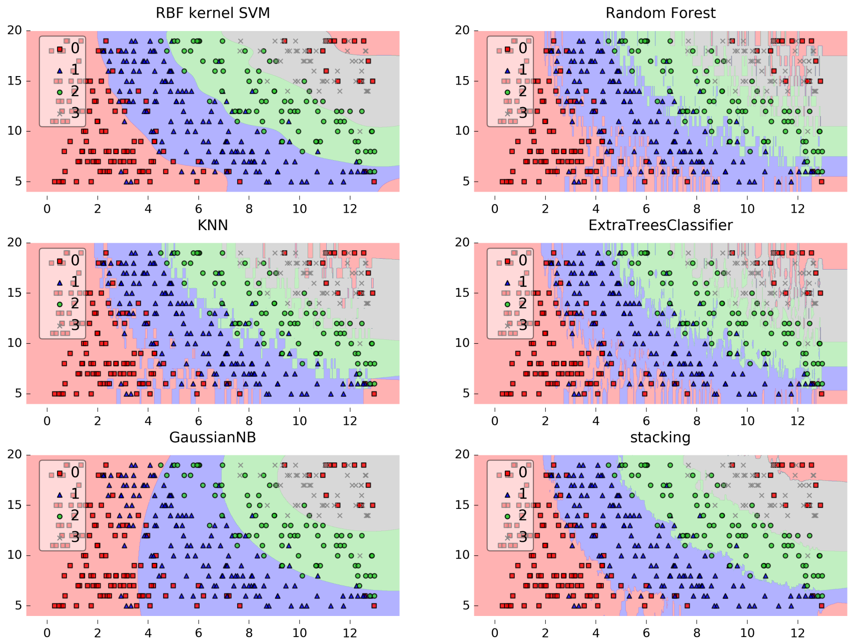

Figure 5.14: Visualization of multiclass classification by single machine learning model and stacking ensemble model.

Table 5.2: Mean Accuracy of Multilabel Classification on 10-CV training Dataset.

Classifier	**Mean accuracy**
SVC	0.802
Random Forest	0.810
KNN	0.840
ExtraTrees	0.791
GBDT	0.894
1st level ensemble	**Mean accuracy**
Soft Vote	0.914
Stacking 1	0.924
Stacking 2	0.922
2st level ensemble	**Mean accuracy**
Final Soft Vote	0.931

5.4.3 Experiment and result analysis

For the ensemble model we used in Experiment, base models [136], [137] are selected as listed in Table 5.2. Additionally, the stacking model is combined with soft vote ensemble method for higher 2st level ensemble. Soft vote model in 1st level ensemble compute the average probabilities of base models predictions using uniform weights, which is tuned according with the performance of each base model. Two stacking ensemble models are trained based on different base model parameters, feature preprocessing and random seeds. In final 2st level ensemble, another soft vote ensemble model average the three models in 1st level ensemble, eventually achieves the best prediction performance in multi-class classification task.

In the evaluation experiment of DQMC with init weight fusion (initDQMC), we set DQMC with fixed parameters: memory size is equal to 1000, using ε-greedy as default policy of deep Q-network, RMSprop is choosed as gradient descend optimization method. For the initial weight fusion mechanism, learning rate is assigned with fixed value $\varepsilon_w = 0.1$. Each experiment is repeated 10 times with random seeds.

The evaluation performance of rewards per episode is shown in Figure 5.15. It's obvious that initDQMC earns more rewards in the first 1000 episodes no matter with deep Q-network or double deep Q-network. initDQMC with DDQN performs the best result and achieve slight higher reward than other methods after 2000 episodes. Initial weight fusion allows initDQMC to rapidly explore unknown environment and acquire effective experiences under the guidance of stacking ensemble classifier.

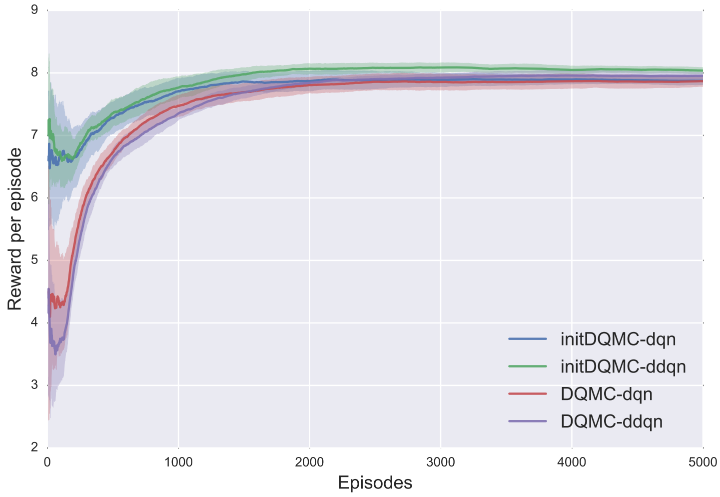

Figure 5.15: Rewards per episode in 20000 action steps

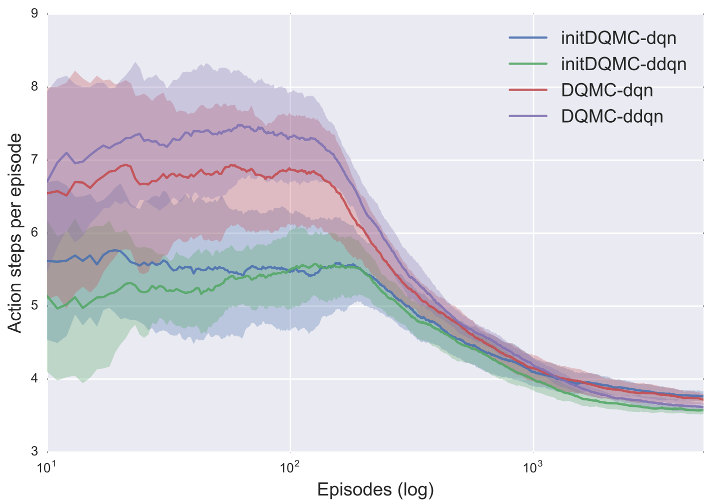

Figure 5.16: Action steps of each episode in 20000 action steps

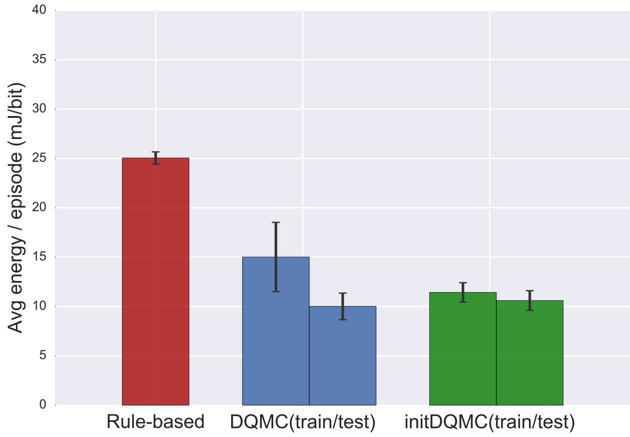

Figure 5.17: Average energy consumption per episode.

Figure 5.16 demonstrates the trace of action steps of each episode in total 20000 action steps. In the initial stage, all the learning models generate a wider margin of variation on action steps per episode, diminishing the shaking along with the program of training sequence. DQMC has to pay more actions to get reward due to the lack of experience that learning model could not make reliable system-decision on action selection. From the mean values of number of action steps, initDQMC require less steps to finish each episode compared with the result of DQMC. Improved initial weight helps deep Q-network try to assign each action with feedback of valuable experience instead of action selected from Q-values space with high confused value or random explore action.

Besides, the average performance of effective energy consumption is evaluated in Figure 5.17. For the training process, initDQMC results a better performance on the average energy consumption of each episode compared with DQMC model, which only consumes nearly 50% of energy that rule-based method generates. The effective energy expenditure of DQMC has a large margin of variation during training tasks. Instead of that, initDQMC achieves more stable and reliable learning model without fluctuate performance.

5.5 Conclusion

Based on the framework of dynamical multi-channel assignment mechanism in previous chapter, we proposed an enhanced multi-channel allocation strategy with deep Q-network reinforcement learning. Four raw observations of wireless network environment are considered as input variable of deep Q-network learning model after feature preprocessing. Three layers neural network model with ReLU activation function is trained to represent the components in traditional reinforcement learning problems, witch allow us to predict all the discrete action Q-values at once for each state. With the experiment replay mechanism, DQMC could iteratively acquire knowledge and learn pattern from the random mini-batch samples of memory sequence. The experiment results verified that DQMC successfully learn a better multi-channel allocation strategy after a certain environment exploration and self-learning episodes. The use of initial weight fusion mechanism of initDQMC alleviates the energy loss due to initial environment exploration in the early stage of learning process. Supervised classifier allows deep Q-learning model to be more efficient in rapidly environment exploration and experience accumulationg. Stacking ensemble model could enhance the accuracy of multi-class classification without any changes in the methodology of feature selection and feature engineering.

Chapter 6

Conclusion and Perspective

This dissertation was dedicated to advancing the state of the art in terms of dynamically control multi-channel deployment above the cross-layer wireless multimedia sensor networks. Different approaches have been tackled, from fundamental stack cross-layer analytical model to the multi-channel mechanism perspective. The present work attemps to create a multi-channel resource allocation techniques which can be used in the higher level multi-channel wireless sensor network techniques for the balance between QoS and energy consumption, filling a gap that was missing. This chapter makes overall conclusions and summarizes the contributions of this work, along with the discussion of future work.

6.1 Conclusions

The first part of this work presented an enhanced stack cross-layer analytical model for CSMA/CA IEEE 802.15.4 networks. In order to design self-adaptive control cognitive system for wireless sensor networks specially for cross-layer efficient optimization algorithm, it's prerequisite to explore more deeply and comprehensive analysis of cross-layer network performance. The multivariate stack Model makes the comprehensive combination and interaction between PHY layer propagation model and MAC layer Markov chain model. The dynamic interaction of sub-layer models with multivariate parameters allow us to achieve adaptive performance estimation in cross-layer level.

Considering different scenarios with multi-dimensional parameters, performance metrics are evaluated as throughput, end-to-end delay, transmission error probability, maximum retransmission failure probability, packet delivery ratio and effective energy consumption. From the analysis of packet error probability p_{fail}, transmission failure probability increased significantly during the distance range 20 to 60 m, theoretical transmission distance can be predicted from given parameters. Stack model also demonstrated the packet discard probability p_{rtx} due to the maximum retransmission attempts failure of CSMA/CA procedure. Prior probability of stacking transmission failure generated variant probability distribution of p_{rtx} with multivariate parameters. For the metrics of QoS performance, adaptive PHY layer model generated inconspicuous impact on node performance in the range of low data loads. Stack model keeps cautious estimation on PHY channel error which compared with joint model and single layer Markov chain model. On the contrary, the frequent channel collision, transmission failure and optimal PHY layer propagation error increased along with the higher information loads, stack model obtained more decline trend of system reliability compared with benchmark model. Finally, the effective energy consumption E^* was demonstrated by three dimensional visualization of sampling observations. Simulation results are in a good agreement with the interpretations in multivariate stack model, verified that stack model effectively combined single sublayer models and achieved more comprehensive systemic analysis on QoS performance and effective energy consumption.

The second part of this work was focus on the dynamically bandwidth control on multi-channel assignment mechanism. From the deeper investigation in first aspect study, the evaluation shows that it's necessary to design comprehensive control mechanism for the resource allocation strategy for the trade-offs between QoS guarantee and energy saving. It is important to notice that because of the attributes of applications in wireless multimedia sensor network, efficient multi-channel deployment should provide intelligent approaches to allocate active channels based on the requirement of system.

Based on the requirement, a fuzzy-based multi-channel assignment mechanism MCDB is proposed. Residual bandwidth is a crucial resource in WMSNs which has tight relevance with QoS an energy efficiency. The instant residual bandwidth was estimated as the global system performance indicator and used as one of the feature inputs of multi-channel assignment algorithm. The impact of system factors on the estimated residual available bandwidth ratio ω_{ABE} are separated into packet control messages, MAC layer overhead and PHY channel constraints, respectively. Tabular-based reinforcement learning was used to make systemic decision for optimized action from perceived environment. Besides, fuzzy logic-based bandwidth trigger threshold control module was proposed in order to find the adaptive optimal values for channel allocation functionality.

This work investigated two aspects of metrics and scenarios to evaluate the fuzzy-based multi-channel assignment model: instant performance on QoS metrics and energy consumption based on baseline stress test scenario and different network size, respectively. The first part of simulation verified that MCDB could get benefit of QoS performance in high data flow applications compared with benchmark models, which achieved better spectrum utilization to improve overall throughput and alleviate congestion. In second part of simulation indicated that the degradative system reliability result in additional energy expenditure with lower conversion efficiency from total dissipative energy to actual network throughput. Fuzzy-based approach provided best results in instant energy consumption E^*. The dynamically allocated threshold values of channel assignment admission control make sure the convergence of energy expenditure in each segment of simulation sequences.

For the further research, basic reinforcement learning has certain disadvantage along with the increasing complexity features in practice wireless sensor network environment. The high correlated environment observations in time sequence leads learning strategy to generate bias or diverge. In order to deal with the learning risk and problem of computational cost to store large scale state-action space with table, we proposed a multi-channel deployment control mechanism by reinforcement learning with deep Q-network approach (DQMC). Sensor node could perform decision making and system control directly from informative features which are retrieved from raw environment obsevations. Deep Q-learning trained neural network-based structure to learn appropriate action pattern, mapping from continuous network environment features to discrete output actions of control targets.

This work investigated two frameworks to create deep reinforcement learning: Deep Q-network and double deep Q-network. Three kinds of explore policies are evaluated for the experiences acquirement strategy with the balance of exploration and exploitation. RMSprop and Adam are considered as the gradient descent optimization algorithm in backpropagation procedure for training neural network. For the experiment result of one training session, DQMC with 1000 memory size of experiment replay, achieved a significant increase of rewards per each action in the first 500 episodes. DQMC improved the ability during self-learning iterations which allow nodes to make better action decision with more rewards. After 10000 action steps, 81% of episodes successfully finished the tasks, most of the correct actions were selected in the first action step of each episode. In order to analysis the performance variance of learning model, comparison experiments were evaluated between DQMC learning models using different hyperparameters and core learning algorithms. It can be observed that DQMC with Boltzmann policy could explore more potential actions from sub-optimal possible action space, decreasing the probability of selecting wasted actions in random policy. For the analysis of effective energy consumption, fuzzy rule-based multi-channel approach is evaluated as a comparison with the same experiment sequence. The reported energy saving enhancement of 40% shows that DQMC model could be very effective to achieve promotion on overall rewards with energy saving target.

Another approach that may lead to improved the stability of DQMC was proposed consequently, in order to alleviate the performace gap between DQMC-train model and DQMC-test model in initialized learning process. Supervised learning model was pre-trained on collected training sample with stacking ensemble multi-class classifier. DQMC used classifier model to predict approximately estimatations with probability output, which could be incorporated into the original weight of Q-values of deep Q-network. In the demonstration of multi-class classification tasks, 2st level stacking ensemble classification achieved the best accuracy performance , which is integrated into proposed DQMC model with init weight fusion (initDQMC). The result shows that initDQMC achieved more rapid rewards accumulation in early learning stage and slight higher rewards after 2000 episodes. Initial weight fusion allows deep Q-network to rapidly explore unknown environment and collect experiences with the guidance of pre-trained classifier. During the full training process, initDQMC only consumed approximate 50% of energy consumption that rule-based method consumed, which also achieved more stable and reliable energy performance compared with DQMC learning model.

6.2 Contributions

This dissertation proposed a methodology and paradigm to create a self-adaptive bandwidth control mechanism for multi-channel deployment in wireless sensor network. Thus, some correlative contributions were achieved. We summarized some of the contributions accomplished in this work:

Multivariate stack analytical model

As the interaction between sub-layer usually neglected, existing performance evaluation model only focus on one aspect of different single layers. Locally single layer model can not faithfully mimic the system performance of wireless network. This work proposed a stack cross-layer model to fully combined the abilities of each sub-layer model. The comprehensive stacking interactions of PHY layer propagation model and MAC layer Markov chain model not only allows a performance evaluation under multivariable parameters space, but also avoid overestimation of system performance which compared with single sub-layer model or simple joint layer model. Besides, for basic evaluation method of energy consumption in analytical model, overall energy is calculated from different states of network. The proposed effective energy consumption metric allow us to investigate advanced energy overhead metrics which indicate the effective conversion of overall energy consumption. The model is distributed under the GPL license, allowing end-users to use, share, and modify the model.

Fuzzy-based multi-channel assignment mechanism

Instant residual bandwidth is proposed as the global performance indicator, fully estimate the impact of packet control message, back-off mechanism, retransmission attemps and propagation channel constraints on additional bandwidth overhead. The use of tabular-based reinforcement learning allows us to deploy multi-channel resource in a smart way. Fuzzy logic-based bandwidth trigger threshold control module achieves optimal values for channel allocation function, which successfully improve the performance to make trade-off between energy efficiency and dynamic QoS requirement.

Deep Q-network for multi-channel allocation strategy

The existing tabular-based reinforcement learning has disadvantages in WSN applications. High correlated informations in observations has potential problem to generate bias learning model. Increasing complexity of network environment bring the problem of computational cost to store and deal with large-scale data in table. In this work, we proposed an effective multi-channel allocation mechanism with deep Q-network learning model. neural network with ReLU activation function is trained to mimic the functions of components in traditional reinforcement learning method. Deep Q-network iteratively acquire knowledge from environment, learning action strategy in experiences replay using random sample mini-batch memory sequence. Experiment results verified that deep Q-network could learn an effective multi-channel allocation strategy after certain training episodes, which achieved better performance of effective energy consumption than fuzzy rule-based approach.

Initial weight fusion optimization for DQMC

Due to the strategy of explore policy in deep Q-network learning model, channel allocation model has to train a new learning model from initial Q-network, which result additional energy during knowledge exploration and learning process. The impact of initial learning process on energy efficiency performance will be magnified along with the increasing complexity features in practical wireless network environment. Consequently, we proposed an enhanced method to alleviate the wasted energy in each initialized learning process of early stage. Supervised multi-class classifier was trained to predict output probability using the same observations, which is only integrated with original initial weight of Q-values in deep Q-network. Predifined learning rate ε_w is used to avoid the overestimation problem of initial classifier, which make sure that initial weight fusion optimization effectively boost the learning speed of DQMC in early stage. Experiment results indicate that DQMC with initial weight fusion perform more stable and reliable learning process, alleviating the fluctuation of performance in early learning stage.

Deep Q-learning framework paradigm for WSNs

In this work, Deep Q-learning framework paradigm was implemented to integrated into WSNs model, which could be reproduced and extended to another WSNs research. The toolkit implemented fundamental data acquisition, data preprocessing and feature engineering. OpenAI Gym was modified and integrated into the framework to perform the basic agent-environment loop as a Markov decision process. InitDQMC was implemented with keras-based neural network and multi-class classifier with stacking ensemble learning model. This toolkit is distributed under the GPL license, allowing using and modification.

6.3 Perspectives

This work presented in this dissertation is the primary step in a significant paradigm research for multi-channel resource allocation mechanism. Some issues are discussed and further improvement is required. As a perspective, there are certainly remain opportunities to boost the research step forward in the future work.

Short-term

Full-stack analytical model The proposed stack cross-layer model achieved more comprehensive performance analysis of IEEE 802.15.4 wireless sensor network. We analyzed MAC layer and PHY layer model and combined interactive sub-layer models into a stack cross-layer analytical model. This approach not only bring more comprehensive analysis of system performance, but also investigate the variance of system performance under multiple parameters. Next step, the analytical will be extended into routing layer with multi-hop transmission architecture. Full stack cross-layer model has potential to reproduce faithful performance tracking with different type of WSN architectures and multi-dimensional parameters. The full-stack analytical model could be used in the further research of correlated optimization schemes, highly parameterized features could be configured and applied to different wireless network platforms.

Integration of different solutions In this work, the paradigm and methodology of multi-channel allocation mechanism is divided into fuzzy rule-based approach and deep Q-network reinforcement learning approach. We introduced and identified these two kind of approaches respectively, which all achieved self-adaptive control schemes in multi-channel allocation tasks. Fully investigation and comparison of these two kinds of methods should be researched in different practical environments and application tasks, the integration of different solutions allows node system to select appropriate solutions based on the variance of network environment. Besides, the main methodology of proposed multi-channel allocation schemes has extensible ability, which could be developed and applied into optimization mechanism of different application scenarios and performance enhancement of smart system decision.

Evaluating model integrated into hardware platform environment In this work, we implemented several algorithms based on the methodology of artificial intelligence techniques, simulated experiments verified that overall performance of WSNs could be improved and the balance between QoS and energy efficiency could be enhanced through suitable artificial intelligence techniques, such as reinforcement learning, fuzzy logic system, supervised machine learning methods, etc. This work bring a technological paradigm which allows the further opportunities of researchs on real wireless network platforms and development in application level. For example, the proposed model could be reproduced and evaluated on production testbed environment such as USRP cognitive radio platform [138] with GNUradio-based testbed [139]. The energy consumption of proposed algorithm also should be evaluated in real practical environment, further optimization of the balance between algorithm performance and computational cost will be another important research topic.

Long-term

Advanced learning algorithm for large-scale wireless network Along with the development of machine learning in different application fields, a lot of great learning algorithm successfully improve the performance of learning model, such as the application fields of image detection, AI gaming, robotics etc. Some methodologies behind these learning algorithm have a great potential inspiration for the intelligence development of WSNs and IoT. The Asynchronous Advantage Actor-Critic (A3C) algorithm [140] was released by Google's DeepMind group, which achieved faster, simple and robust self-learning process compared with existing deep reinforcement learning. In the global network of A3C, multiple worker agents which have their own set of network parameters. Each agent interacts with it's own copy of environment which all agents are independent workers with different experiences. Workers perform interaction and learning process in parallel, update global network with gradients. The overall structure works in the similar methodology with wireless network, which is a very interesting research direction.

Online learning algorithm for WSNs optimization Several online learning algorithms [141] have been successfully developed for predicting massive-scale ad click–through rates (CTR) problem. Online learning algorithm process model training with very new arrived data, without saving enormous quantities of data to make training samples [142]. The node could produce predictions based on online learning algorithm without storing and dealing with historical database. The attributes and advantages of online learning algorithm make it suitable to be used in WSNs field which could process real-time continuous events with limited hardware resources. It should be valuable research direction to design fast, accuracy and light weight online learning algorithm special for different levels of WSNs application tasks and system control.

Deep reinforcement learning in production IoT From the research in this work, we could obtain a new perspective about the real potential of artificial intelligence for pragmatic applications of IoT. Continuous proliferation and obsolescence of devices has more and more requirement in network devices services and resources management. Artificial intelligence techniques might be and should be the essential ingredient that will allow IoT to become a reality and yield its promise of new business models.

Deep reinforcement learning techniques that we discussed in this work points out a promising direction, which will play an important role as one of core infrastructures behind IoT environment. Underlying integration of deep reinforcement learning techniques into IoT is the broader trend for automating network infrastructure resource management with more dynamic and fault tolerant solution. Besides, more comprehensive physical infrastructure will be modeled in software as the system of information flows, which build fundamental environments for deep reinforcement learning-driven optimization.

Bibliography

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

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

[3] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005.

[4] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless mesh networks design—a survey," *IEEE Communications surveys & tutorials*, vol. 14, no. 2, pp. 299–310, 2012.

[5] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63–75, 2010.

[6] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[7] L. M. Oliveira and J. J. Rodrigues, "Wireless sensor networks: A survey on environmental monitoring.," *JCM*, vol. 6, no. 2, pp. 143–151, 2011.

[8] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32–48, 2005.

[9] S. Taneja and A. Kush, "A survey of routing protocols in mobile ad hoc networks," *International Journal of innovation, Management and technology*, vol. 1, no. 3, p. 279, 2010.

[10] libelium, *50 Sensor Applications for a Smarter World.* http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/.

[11] J. Zheng and A. Jamalipour, *Wireless sensor networks: a networking perspective*. John Wiley & Sons, 2009.

[12] G. P. Joshi, S. Y. Nam, and S. W. Kim, "Cognitive radio wireless sensor networks: applications, challenges and research trends," *Sensors*, vol. 13, no. 9, pp. 11196–11228, 2013.

[13] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE communications surveys & tutorials*, vol. 10, no. 4, 2008.

[14] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless multimedia sensor networks: Applications and testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, 2008.

[15] Y. Wu and M. Cardei, "Multi-channel and cognitive radio approaches for wireless sensor networks," *Computer Communications*, vol. 94, pp. 30–45, 2016.

[16] S. H. R. Bukhari, M. H. Rehmani, and S. Siraj, "A survey of channel bonding for wireless networks and guidelines of channel bonding for futuristic cognitive radio sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 924–948, 2016.

[17] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, T. F. Abdelzaher, *et al.*, "Mmsn: Multi-frequency media access control for wireless sensor networks.," in *Infocom*, vol. 6, pp. 1–13, 2006.

[18] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multichannel communications in wireless sensor networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 1193–1201, IEEE, 2008.

[19] R. Diab, G. Chalhoub, and M. Misson, "Enhanced multi-channel mac protocol for multi-hop wireless sensor networks," in *Wireless Days (WD), 2014 IFIP*, pp. 1–6, IEEE, 2014.

[20] S. Roy, N. Darak, and A. Nasipuri, "A game theoretic approach for channel selection in multi-channel wireless sensor networks," in *High-capacity Optical Networks and Emerging/Enabling Technologies (HONET), 2014 11th Annual*, pp. 145–149, IEEE, 2014.

[21] O. D. Incel, S. Dulman, and P. Jansen, "Multi-channel support for dense wireless sensor networking," in *European Conference on Smart Sensing and Context*, pp. 1–14, Springer, 2006.

[22] L. F. Van Hoesel and P. J. Havinga, "A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," 2004.

[23] X. Li, D. Wang, J. McNair, and J. Chen, "Residual energy aware channel assignment in cognitive radio sensor networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pp. 398–403, IEEE, 2011.

[24] L. Stabellini, "Energy-aware channel selection for cognitive wireless sensor networks," in *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pp. 892–896, IEEE, 2010.

[25] A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. Leung, "A survey of security challenges in cognitive radio networks: Solutions and future research directions," *Proceedings of the IEEE*, vol. 100, no. 12, pp. 3172–3186, 2012.

[26] K. S. Nanavati, *Channel bonding/loading for TV white spaces in IEEE 802.11 af.* PhD thesis, San Diego State University, 2012.

[27] V. Valenta, R. Maršálek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, "Survey on spectrum utilization in europe: Measurements, analyses and observations," in *Cognitive Radio Oriented Wireless Networks & Communications (CROWNCOM), 2010 Proceedings of the Fifth International Conference on*, pp. 1–5, IEEE, 2010.

[28] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach.* Springer Science & Business Media, 2013.

[29] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[30] A. Rooshenas, H. R. Rabiee, A. Movaghar, and M. Y. Naderi, "Reducing the data transmission in wireless sensor networks using the principal component analysis," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on*, pp. 133–138, IEEE, 2010.

[31] R. Masiero, G. Quer, M. Rossi, and M. Zorzi, "A bayesian analysis of compressive sensing data recovery in wireless sensor networks," in

2009 International Conference on Ultra Modern Telecommunications & Workshops, pp. 1–6, IEEE, 2009.

[32] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[33] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[34] Y. Mostofi, A. Gonzalez-Ruiz, A. Gaffarkhah, and D. Li, "Characterization and modeling of wireless channels for networked robotic and control systems-a comprehensive overview," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 4849–4854, IEEE, 2009.

[35] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[36] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[37] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with gaussian kernels to radial basis function classifiers," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2758–2765, 1997.

[38] J. H. Min and Y.-C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert systems with applications*, vol. 28, no. 4, pp. 603–614, 2005.

[39] E. F. Flushing, J. Nagi, and G. A. Di Caro, "A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, pp. 137–143, IEEE, 2012.

[40] M. Sha, R. Dor, G. Hackmann, C. Lu, T.-S. Kim, and T. Park, "Self-adapting mac layer for wireless sensor networks," in *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*, pp. 192–201, IEEE, 2013.

[41] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[42] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[43] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[44] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems.* CRC Press, 1998.

[45] Q. Wang, F. Yang, Q. Ge, and Q. Yang, "A sensor network modeling and fault detection method for large wind farms by using neural networks," in *Control & Automation (ICCA), 11th IEEE International Conference on*, pp. 308–313, IEEE, 2014.

[46] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[47] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, vol. 2, p. 3, 2010.

[48] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[49] H. He, Z. Zhu, and E. Makinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 973–982, 2009.

[50] S. Haykin and N. Network, "A comprehensive foundation," *Neural Networks*, vol. 2, no. 2004, p. 41, 2004.

[51] Y.-J. Shen and M.-S. Wang, "Broadcast scheduling in wireless sensor networks using fuzzy hopfield neural network," *Expert systems with applications*, vol. 34, no. 2, pp. 900–907, 2008.

[52] F. Jolai and A. Ghanbari, "Integrating data transformation techniques with hopfield neural networks for solving travelling salesman problem," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5331–5335, 2010.

[53] P. C. Prasad and A. Beg, "Investigating data preprocessing methods for circuit complexity models," *Expert Systems with Applications*, vol. 36, no. 1, pp. 519–526, 2009.

[54] P. Wang and T. Wang, "Adaptive routing for sensor networks using reinforcement learning," in *Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on*, pp. 219–219, IEEE, 2006.

[55] K. Shah and M. Kumar, "Distributed independent reinforcement learning (dirl) approach to resource management in wireless sensor networks," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pp. 1–9, IEEE, 2007.

[56] T. Liu and A. E. Cerpa, "Data-driven link quality prediction using link features," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 2, p. 37, 2014.

[57] H. Jiang and J. O. Hallstrom, "Fast, accurate event classification on resource-lean embedded sensors," in *European Conference on Wireless Sensor Networks*, pp. 65–80, Springer, 2011.

[58] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE communications surveys & tutorials*, vol. 13, no. 1, pp. 68–96, 2011.

[59] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1434–1453, Society for Industrial and Applied Mathematics, 2013.

[60] S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *Signal Processing Advances in Wireless Communications (SPAWC), 2010 IEEE Eleventh International Workshop on*, pp. 1–5, IEEE, 2010.

[61] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed pca," *Signal Processing*, vol. 104, pp. 120–135, 2014.

[62] A. Panousopoulou, M. Azkune, and P. Tsakalides, "Feature selection for performance characterization in multi-hop wireless sensor networks," *Ad Hoc Networks*, vol. 49, pp. 70–89, 2016.

[63] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[64] P. Sasikumar and S. Khara, "K-means clustering in wireless sensor networks," in *Computational intelligence and communication networks (CICN), 2012 fourth international conference on*, pp. 140–144, IEEE, 2012.

[65] D. Hoang, R. Kumar, and S. Panda, "Fuzzy c-means clustering protocol for wireless sensor networks," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pp. 3477–3482, IEEE, 2010.

[66] K. C. Zikidis and A. V. Vasilakos, "Asafes2: A novel, neuro-fuzzy architecture for fuzzy computing, based on functional reasoning," *Fuzzy Sets and Systems*, vol. 83, no. 1, pp. 63–84, 1996.

[67] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*, pp. 10–pp, IEEE, 2000.

[68] G. Ran, H. Zhang, and S. Gong, "Improving on leach protocol of wireless sensor networks using fuzzy logic," *Journal of Information & Computational Science*, vol. 7, no. 3, pp. 767–775, 2010.

[69] H. Taheri, P. Neamatollahi, O. M. Younis, S. Naghibzadeh, and M. H. Yaghmaee, "An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1469–1481, 2012.

[70] P. Nayak and A. Devulapalli, "A fuzzy logic-based clustering algorithm for wsn to extend the network lifetime," *IEEE sensors journal*, vol. 16, no. 1, pp. 137–144, 2016.

[71] C. Sonmez, O. D. Incel, S. Isik, M. Y. Donmez, and C. Ersoy, "Fuzzy-based congestion control for wireless multimedia sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–17, 2014.

[72] H. Jiang, Y. Sun, R. Sun, and H. Xu, "Fuzzy-logic-based energy optimized routing for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[73] Z. Liu and I. Elhanany, "Rl-mac: a reinforcement learning based mac protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 117–124, 2006.

[74] S. Galzarano, A. Liotta, and G. Fortino, "Ql-mac: a q-learning based mac for wireless sensor networks," in *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 267–275, Springer, 2013.

[75] M. Mihaylov, Y.-A. Le Borgne, K. Tuyls, and A. Nowé, "Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 9, no. 3-4, pp. 207–224, 2012.

[76] A. Arya, A. Malik, and R. Garg, "Reinforcement learning based routing protocols in wsns: A survey,"

[77] R. Arroyo-Valles, R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Q-probabilistic routing in wireless sensor networks," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 1–6, IEEE, 2007.

[78] M. Mihaylov, K. Tuyls, and A. Nowé, "Decentralized learning in wireless sensor networks," in *International Workshop on Adaptive and Learning Agents*, pp. 60–73, Springer, 2009.

[79] L. Tran-Thanh, A. Rogers, and N. R. Jennings, "Long-term information collection with energy harvesting wireless sensors: a multi-armed bandit based approach," *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 2, pp. 352–394, 2012.

[80] M. I. Khan and B. Rinner, "Resource coordination in wireless sensor networks by cooperative reinforcement learning," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pp. 895–900, IEEE, 2012.

[81] M. I. Khan and B. Rinner, "Energy-aware task scheduling in wireless sensor networks based on cooperative reinforcement learning," in *2014 IEEE International Conference on Communications Workshops (ICC)*, pp. 871–877, IEEE, 2014.

[82] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, and A. Abraham, "Cooperative game theoretic approach using fuzzy q-learning for detecting and preventing intrusions in wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 228–241, 2014.

[83] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[84] M. Mihaylov, Y.-A. Le Borgne, K. Tuyls, and A. Nowé, “Reinforcement learning for self-organizing wake-up scheduling in wireless sensor networks,” in *International Conference on Agents and Artificial Intelligence*, pp. 382–396, Springer, 2011.

[85] S. N. Das and S. Misra, “Correlation-aware cross-layer design for network management of wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 5, no. 6, pp. 263–270, 2015.

[86] Y. Chu, P. D. Mitchell, and D. Grace, “Aloha and q-learning based medium access control for wireless sensor networks,” in *2012 International Symposium on Wireless Communication Systems (ISWCS)*, pp. 511–515, IEEE, 2012.

[87] S. Kosunalp, Y. Chu, P. D. Mitchell, D. Grace, and T. Clarke, “Use of q-learning approaches for practical medium access control in wireless sensor networks,” *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 146–154, 2016.

[88] J. Chen, Q. Yu, B. Chai, Y. Sun, Y. Fan, and X. S. Shen, “Dynamic channel assignment for wireless sensor networks: A regret matching based approach,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 95–106, 2015.

[89] J. T. Adams, “An introduction to ieee std 802.15. 4,” in *Aerospace Conference, 2006 IEEE*, pp. 8–pp, IEEE, 2006.

[90] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 1–13, ACM, 2003.

[91] M. Zuniga and B. Krishnamachari, “Analyzing the transitional region in low power wireless links,” in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 517–526, IEEE, 2004.

[92] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Impact of radio irregularity on wireless sensor networks,” in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 125–138, ACM, 2004.

[93] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 78–82, ACM, 2004.

[94] P. Park, P. Di Marco, P. Soldati, C. Fischione, and K. H. Johansson, "A generalized markov chain model for effective analysis of slotted ieee 802.15. 4," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pp. 130–139, IEEE, 2009.

[95] D. Striccoli, G. Boggia, and L. A. Grieco, "A markov model for characterizing ieee 802.15. 4 mac layer in noisy environments," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5133–5142, 2015.

[96] D. Mahmood, Z. Khan, U. Qasim, M. U. Naru, S. Mukhtar, M. Akram, and N. Javaid, "Analyzing and evaluating contention access period of slotted csma/ca for ieee802. 15.4," *Procedia Computer Science*, vol. 34, pp. 204–211, 2014.

[97] P. Di Marco, C. Fischione, F. Santucci, and K. H. Johansson, "Modeling ieee 802.15. 4 networks over fading channels," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5366–5381, 2014.

[98] M.-H. Zayani, V. Gauthier, and D. Zeghlache, "A joint model for ieee 802.15. 4 physical and medium access control layers," in *2011 7th International Wireless Communications and Mobile Computing Conference*, pp. 814–819, IEEE, 2011.

[99] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability.* Springer Science & Business Media, 2012.

[100] D. De Guglielmo, F. Restuccia, G. Anastasi, M. Conti, and S. Das, "Accurate and efficient modeling of 802.15. 4 unslotted csma/ca through event chains computation," 2016.

[101] C. T. Kone, A. Hafid, and M. Boushaba, "Performance management of ieee 802.15. 4 wireless sensor network for precision agriculture," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5734–5747, 2015.

[102] M. Z. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Transactions on Sensor Networks (TOSN)*, vol. 3, no. 2, p. 7, 2007.

[103] M. Holland, T. Wang, B. Tavli, A. Seyedi, and W. Heinzelman, "Optimizing physical-layer parameters for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 4, p. 28, 2011.

[104] T. Wang, W. Heinzelman, and A. Seyedi, "Link energy minimization for wireless networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 569–585, 2012.

[105] J. G. Proakis and S. Masoud, *Digital Communications*. 2008.

[106] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. Van der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance analysis of slotted carrier sense ieee 802.15. 4 medium access layer," *IEEE Transactions on wireless communications*, vol. 7, no. 9, pp. 3359–3371, 2008.

[107] Z. Y. Liu, D. Dragomirescu, G. Da Costa, and T. Monteil, "A Stack Cross-layer Analytical Model for CSMA/CA IEEE 802.15.4 Networks," in *ICC International Conference on Internet of Things, Data and Cloud Computing, United Kingdom, 22/03/2017-23/03/2017*, (http://www.acm.org/), ACM, 2017.

[108] R. Soua and P. Minet, "Multichannel assignment protocols in wireless sensor networks: A comprehensive survey," *Pervasive and Mobile Computing*, vol. 16, pp. 2–21, 2015.

[109] T. Instruments, "Cc2520 datasheet: 2.4ghz ieee 802.15.4/zigbee rf transceiver," *SWRS068*, 2007.

[110] M. Cardei and A. Mihnea, "Channel assignment in cognitive wireless sensor networks," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 588–593, IEEE, 2014.

[111] Y. Wu and M. Cardei, "Robust topology using reconfigurable radios in wireless sensor networks," in *Mobile Ad-hoc and Sensor Networks (MSN), 2014 10th International Conference on*, pp. 38–45, IEEE, 2014.

[112] Y. Kim, H. Shin, and H. Cha, "Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks," in *Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 53–63, IEEE Computer Society, 2008.

[113] B. Jang, J. B. Lim, and M. L. Sichitiu, "An asynchronous scheduled mac protocol for wireless sensor networks," *Computer Networks*, vol. 57, no. 1, pp. 85–98, 2013.

[114] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, “Em-mac: a dynamic multichannel energy-efficient mac protocol for wireless sensor networks,” in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, p. 23, ACM, 2011.

[115] Q. Yu, J. Chen, Y. Sun, Y. Fan, and X. Shen, “Regret matching based channel assignment for wireless sensor networks,” in *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–5, IEEE, 2010.

[116] L. Choong, “Multi-channel ieee 802.15. 4 packet capture using software defined radio,” *UCLA Networked & Embedded Sensing Lab*, vol. 3, 2009.

[117] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous, “Bandwidth estimation for IEEE 802.11-based ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 10, pp. 1228–1241, 2008.

[118] T. Wang, W. Heinzelman, and A. Seyedi, “Link energy minimization for wireless networks,” *Ad Hoc Networks*, vol. 10, no. 3, pp. 569–585, 2012.

[119] N. V. Nguyen, I. Guerin-Lassous, V. Moraru, and C. Sarr, “Retransmission-based available bandwidth estimation in IEEE 802.11-based multihop wireless networks,” *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 377–384, 2011.

[120] C. Sonmez, O. D. Incel, S. Isik, M. Y. Donmez, and C. Ersoy, “Fuzzy-based Congestion Control for Wireless Multimedia Sensor Networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 63, no. 2014, pp. 1–17, 2014.

[121] S. Pollin, M. Ergen, S. Ergen, B. Bougard, F. Catthoor, A. Bahai, and P. Varaiya, “Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Acknowledged Uplink Transmissions,” *2008 IEEE Wireless Communications and Networking Conference*, vol. 7, no. 9, pp. 3359–3371, 2008.

[122] Z. Liu, D. Dragomirescu, G. Da Costa, and T. Monteil, “Dynamic multichannel allocation mechanism for wireless multimedia sensor networks,” in *Wireless Days (WD), 2016*, pp. 1–6, IEEE, 2016.

[123] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[124] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[125] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems." `http://tensorflow.org/`, 2015. Software available from tensorflow.org.

[126] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[127] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," 2011.

[128] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[129] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning.," in *AAAI*, pp. 2094–2100, 2016.

[130] F. Chollet, "keras." `https://github.com/fchollet/keras`, 2015.

[131] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.

[132] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," *University of Toronto, Tech. Rep*, 2012.

[133] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[134] M. Plappert, "keras-rl." `https://github.com/matthiasplappert/keras-rl`, 2016.

[135] R. Sebastian, "Mlxtend." http://dx.doi.org/10.5281/zenodo.49235, apr 2016.

[136] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[137] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.

[138] D. C. Tucker and G. A. Tagliarini, "Prototyping with gnu radio and the usrp-where to begin," in *Southeastcon, 2009. SOUTHEASTCON'09. IEEE*, pp. 50–54, IEEE, 2009.

[139] B. Bloessl, C. Leitner, F. Dressler, and C. Sommer, "A gnu radio-based ieee 802.15. 4 testbed," *12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013)*, pp. 37–40, 2013.

[140] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016.

[141] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, *et al.*, "Ad click prediction: a view from the trenches," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230, ACM, 2013.

[142] J. Pardo, F. Zamora-Martínez, and P. Botella-Rocamora, "Online learning algorithm for time series forecasting suitable for low cost wireless sensor networks nodes," *Sensors*, vol. 15, no. 4, pp. 9277–9304, 2015.