Li, Yun

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/xmlui/handle/123456789/12851

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk
Joint Scheduling and Duty Cycle Control Framework for Hierarchical Machine-to-Machine Communication Networks

by

Yun Li

A thesis submitted to the University of London for the degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

2015
TO MY FAMILY
Abstract

This thesis presents a novel distributed optimisation framework for machine-to-machine (M2M) communication networks with dynamic traffic generation, heterogeneous applications and different device capabilities. The aim of the framework is to effectively manage the massive access of energy constrained M2M devices while satisfying different application requirements. The proposed framework has three control blocks which run at cluster heads and M2M gateways:

i) The distributed duty cycle control that adapts to dynamic network traffics for IEEE 802.15.4 MAC layer protocol with stop-and-wait automatic repeat request (ARQ) and Go-Back-N ARQ schemes.

ii) The cluster head control that applies dynamic programming (DP) and approximate dynamic programming (ADP) techniques to maximise single cluster utility while balancing the tradeoff between system performance and algorithm complexity.

iii) The gateway control that applies network utility optimisation (NUM) and mixed integer programming (MIP) techniques to maximise the aggregated long-term network utility while satisfying different application requirements among clusters.

Both theoretical and practical concerns are addressed by the proposed control framework. Simulation results show that the proposed framework effectively improve the overall network performance in terms of network throughput, energy efficiency, end-to-end delay and packet drop ratio.
Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Michael Chai, for the continuous support during my Ph.D study and research. He guided me through my research and writing of this thesis with his patience, motivation, encouragement, enthusiasm, and immense knowledge.

I am also grateful to my independent accessor, Dr. Yue Chen, for her insightful suggestions and valuable discussions, and also for her consideration and sharing. I would also like to express my special thanks to Dr. Jonathan Loo, who has spent lots of time on discussing research ideas and writing skills with me.

I will also thank the Chinese Scholarship Council (CSC). Without the support of CSC, I could not have had this precious opportunity to study at this great university.

My gratitude goes to all the colleagues and friends, who have shared the time with me that made the past four years so memorable. Moreover, warm thanks are given to all the helpful staff at Queen Mary. It is you who make Queen Mary such a great place to do research.

At last, I would like to express my gratitude to my parents, in particular, for their love and support in all the important moments of my life. Without their love, I would not have made it through.
Declaration

I, Yun Li, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party’s copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Yun Li

Date: January 27, 2016
Table of Contents

Abstract i

Acknowledgments ii

Table of Contents iv

List of Figures viii

List of Tables xi

List of Abbreviations xii

Notations xv

1 Introduction 1

1.1 Research Motivation 1

1.2 Research Scope 4

1.3 Contributions 5

1.4 Thesis Structure 7

1.5 Publications List 9

2 Background and Methodology 12

2.1 Machine-to-Machine Communications 12

2.2 Hierarchical M2M Communication Networks 14
List of Figures

Figure 1.1 Hierarchical M2M networks .................................. 4
Figure 2.1 IEEE 802.15.4 CSMA-CA algorithm ..................... 19
Figure 2.2 Superframe structure of IEEE 802.15.4 .................... 22
Figure 2.3 Classification of duty cycle based MAC protocols ...... 26
Figure 3.1 ETSI compatible hierarchical M2M system model ...... 34
Figure 3.2 MDP model of gateway ...................................... 38
Figure 3.3 Change of queue length of devices ......................... 39
Figure 3.4 Packet drop due to reach the buffer size ................ 39
Figure 4.1 Utility illustration for different applications .......... 45
Figure 4.2 Transmission process of designed control framework .... 53
Figure 4.3 Flow chart of designed simulation platform ............... 56
Figure 4.4 Dual-slope path loss model ............................... 61
Figure 4.5 Flow chart of network initialisation module .............. 62
Figure 4.6 One realisation of network deployment ................... 64
Figure 4.7 Received power of devices ................................. 65
Figure 5.1 Duty cycle setting with different ARQ schemes ........ 89
Figure 5.2 Optimal transmission threshold by applying DP ......... 91
Figure 5.3 Average energy consumption of DP based duty cycle control 92
Figure 5.4 Average end-to-end delay of DP based duty cycle control 93
Figure 7.6  Performance of cluster with low priority and low traffic load:
   a) throughput, b) utility, c) energy efficiency, d) end-to-end delay,
   e) packet drop ratio, and f) profit. ........................................ 147
Figure 7.7  Network performance with heterogeneous applications. . 150
Figure 7.8  CDF of end-to-end delay with heterogeneous applications. 151
Figure 7.9  Performance of cluster with inelastic application and low
   traffic load (cluster head 1). .................................................. 152
Figure 7.10 Performance of cluster with inelastic application and high
   traffic load (cluster head 3). .................................................. 153
Figure 7.11 Performance of cluster with elastic application and low traffic
   load (cluster head 5) .............................................................. 154
List of Tables

Table 1-A Access Requirements of M2M Applications ........................................... 3
Table 2-A Parameters in IEEE 802.15.4 CSMA/CA algorithm ............................ 20
Table 3-A Channel Model Parameters of Indoor-indoor Environments .......... 41
Table 3-B Channel Model Parameters of Outdoor-outdoor Environments ....... 42
Table 3-C Channel Model Parameters of Outdoor-indoor Environments ........... 42
Table 4-A Device Parameters of Example Hierarchical M2M Network .......... 63
Table 4-B IEEE 802.15.4 MAC Layer Parameters ............................................. 63
Table 5-A Duty Cycle Control Simulation Parameters ................................. 90
Table 6-A Simulation Parameters of Cluster Head Controls ......................... 109
Table 6-B Energy Consumption and Buffer Parameters ................................. 115
Table 7-A Gateway Control Simulation Parameters ...................................... 138
Table 7-B Gateway Control Energy and Buffer Parameters ......................... 139
Table 7-C Gateway Control Network Setting ................................................. 140
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AN</td>
<td>Access Network</td>
</tr>
<tr>
<td>ADP</td>
<td>Approximate Dynamic Programming</td>
</tr>
<tr>
<td>ARQ</td>
<td>Automatic Repeat Request</td>
</tr>
<tr>
<td>BE</td>
<td>Backoff Exponent</td>
</tr>
<tr>
<td>BI</td>
<td>Beacon Interval</td>
</tr>
<tr>
<td>BO</td>
<td>Beacon Order</td>
</tr>
<tr>
<td>BOAA</td>
<td>Beacon Order Adaptation Algorithm</td>
</tr>
<tr>
<td>BP</td>
<td>Backoff Period</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CAP</td>
<td>Contention Access Period</td>
</tr>
<tr>
<td>CCA</td>
<td>Clear Channel Access</td>
</tr>
<tr>
<td>CFP</td>
<td>Contention-Free Period</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster Head</td>
</tr>
<tr>
<td>CN</td>
<td>Core Network</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear-To-Send</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DCA</td>
<td>Duty Cycle Algorithm</td>
</tr>
<tr>
<td>DCLA</td>
<td>Duty Cycle Learning Algorithm</td>
</tr>
<tr>
<td>DM</td>
<td>Device Domain</td>
</tr>
<tr>
<td>DP</td>
<td>Dynamic Programming</td>
</tr>
<tr>
<td>DSMAC</td>
<td>Dynamic Sensor MAC</td>
</tr>
<tr>
<td>DW-MAC</td>
<td>Demand Wakeup MAC</td>
</tr>
<tr>
<td>EE</td>
<td>Energy Efficiency</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FFD</td>
<td>Full Functional Device</td>
</tr>
<tr>
<td>FPA</td>
<td>Fast Path Algorithm</td>
</tr>
<tr>
<td>GTS</td>
<td>Guaranteed Time Slot</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>LTE</td>
<td>Long-Term Evolution</td>
</tr>
<tr>
<td>LTE-A</td>
<td>Long-Term Evolution Advanced</td>
</tr>
<tr>
<td>LR-WPAN</td>
<td>Low-Rate Wireless Personal Area Network</td>
</tr>
<tr>
<td>M2M</td>
<td>Machine-to-Machine</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed Integer Programming</td>
</tr>
<tr>
<td>MDP</td>
<td>Markov Decision Process</td>
</tr>
<tr>
<td>ND</td>
<td>Network Domain</td>
</tr>
<tr>
<td>NLOS</td>
<td>None Line of Sight</td>
</tr>
<tr>
<td>NUM</td>
<td>Network Utility Maximisation</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>PF</td>
<td>Proportional Fairness</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical</td>
</tr>
<tr>
<td>PI</td>
<td>Policy Iteration</td>
</tr>
<tr>
<td>P-UPF</td>
<td>Proposed-Utility Proportional Fairness</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RA</td>
<td>Rollout Algorithm</td>
</tr>
<tr>
<td>RAN</td>
<td>radio access network</td>
</tr>
<tr>
<td>RACH</td>
<td>Random Access Channel</td>
</tr>
<tr>
<td>RL</td>
<td>Reinforcement Learning</td>
</tr>
<tr>
<td>RTS</td>
<td>Request-To-Send</td>
</tr>
<tr>
<td>SD</td>
<td>Superframe Duration</td>
</tr>
<tr>
<td>SO</td>
<td>Superframe Order</td>
</tr>
<tr>
<td>SMAC</td>
<td>Sensor MAC</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TDDCA</td>
<td>Traffic-adaptive Distance-based Duty Cycle Assignment</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TMAC</td>
<td>Timeout MAC</td>
</tr>
<tr>
<td>TLDTCA</td>
<td>Tree-Like Distributed Topology Control Algorithm</td>
</tr>
<tr>
<td>UFC</td>
<td>Utility Fairness Control</td>
</tr>
<tr>
<td>UPF</td>
<td>Utility Proportional Fairness</td>
</tr>
<tr>
<td>U-MAC</td>
<td>Utilisation based MAC</td>
</tr>
<tr>
<td>VI</td>
<td>Value Iteration</td>
</tr>
<tr>
<td>WSNs</td>
<td>Wireless Sensor Networks</td>
</tr>
</tbody>
</table>
Notations

\( n \) \quad M2M gateway in the network

\( \mathcal{N} \) \quad the set of M2M gateways in the network

\( N \) \quad the total number of M2M gateways in the network

\( \mathcal{I}_n \) \quad the set of child cluster heads of gateway \( n \in [1, 2, \cdots, \mathcal{N}] \)

\( i \) \quad the cluster head connected with the gateway \( n \)

\( I \) \quad the total number of cluster heads connected with gateway \( n \)

\( \mathcal{L}_{i,n} \) \quad the link set between cluster heads \( i \in \mathcal{I}_n \) and gateway \( n \)

\( l_{i,n} \) \quad the link between cluster head \( i \) and gateway \( n \)

\( \mathcal{C}_i \) \quad the child nodes set of cluster head \( i \in \mathcal{I}_n \)

\( j \) \quad the child device of cluster head \( i \), \( j \in \mathcal{C}_i \)

\( (i, n) \) \quad the device pair of cluster head \( i \) and gateway \( n \)

\( (i, j) \) \quad the device pair of cluster head \( i \) and its child devices \( j \)

\( \mathcal{L}_{i,j} \) \quad the link set between cluster head \( i \) and its child devices \( j \in \mathcal{C}_i \)

\( l_{i,j} \) \quad the link between cluster head \( i \) and its child device \( j \)

\( g_{i,j}^t \) \quad the traffic batch of cluster head \( i \) and its child device \( j \) at time period \( t \)

\( \mathcal{H} \) \quad the number of hops of a selected path

\( M \) \quad the total number of devices in the network
\( T \)  
total control time periods

\( t \)  
discrete time period of control system

\( S \)  
the finite discrete states set of cluster heads

\( s_t \)  
the system state at time period \( t \)

\( A_i \)  
the finite control decision set of cluster head \( i \)

\( a_t \)  
the decision action at time period \( t \)

\( w_t \)  
disturbance random parameter at time period \( t \)

\( P(s, s') \)  
the probability that the system transit from state \( s \) to \( s' \)

\( J(s, s') \)  
the immediate cost of the system transit from state \( s \) to \( s' \)

\( \gamma \)  
the discount factor of the system

\( \pi \)  
the control policy of the system

\( \mu_t \)  
The control maps the state \( s_t \) into the decision action \( a_t \) at time period \( t \)

\( Q(s_t, a_t) \)  
the Q-factor at time period \( t \) with state \( s_t \) and action \( a_t \)

\( L_{c,dB} \)  
the channel propagation loss in dB

\( L_{0,dB} \)  
the reference base channel propagation loss in dB

\( X_{s,dB} \)  
the zero mean Gaussian random in dB

\( X_f \)  
the unit-mean gamma-distributed random in dB

\( d \)  
the distance between the sender and receiver

\( d_1 \)  
the reference distance in propagation model

\( m \)  
the Nakagami fading parameter

\( P_{\text{rec},i}^t \)  
the received power of cluster head \( i \)

\( P_{\text{tran},i}^t \)  
the conducted power of cluster head \( i \)

\( G_{\text{tran},i}^t \)  
the transmit antenna gain of cluster head \( i \)

\( G_{\text{rec},i}^t \)  
the receive antenna gain of cluster head \( i \)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_{i,j}^t$</td>
<td>successful transmission probability between cluster head $i$ and its child device $j$</td>
</tr>
<tr>
<td>$C_{i,n}^t$</td>
<td>Link capacity between cluster head $i$ and gateway $n$ at time period $t$</td>
</tr>
<tr>
<td>$C_{n,b}^t$</td>
<td>Link capacity between gateway $n$ and the base station $b$ at time period $t$</td>
</tr>
<tr>
<td>$H_{i,n}$</td>
<td>power gain between gateway $n$ and cluster head $i$</td>
</tr>
<tr>
<td>$B$</td>
<td>bandwidth</td>
</tr>
<tr>
<td>$N_0$</td>
<td>noise spectral density</td>
</tr>
<tr>
<td>$q_{i}^t$</td>
<td>queue length of cluster head $i$ at time period $t$</td>
</tr>
<tr>
<td>$q_{i}^{max}$</td>
<td>maximum queue length of cluster head $i$ at time period $t$</td>
</tr>
<tr>
<td>$r_i^t$</td>
<td>number of packets received by cluster head $i$ at time period $t$</td>
</tr>
<tr>
<td>$f_i^t$</td>
<td>number of packets transmitted by cluster head $i$ at time period $t$</td>
</tr>
<tr>
<td>$g_i^t$</td>
<td>number of packets generated by cluster head $i$ at time period $t$</td>
</tr>
<tr>
<td>$c_f$</td>
<td>cost function coefficients of transmitting</td>
</tr>
<tr>
<td>$c_r$</td>
<td>cost function coefficients of receiving</td>
</tr>
<tr>
<td>$c_l$</td>
<td>coefficients of idle listening</td>
</tr>
<tr>
<td>$c_d$</td>
<td>cost function coefficients of delay</td>
</tr>
<tr>
<td>$\theta_{i}^t$</td>
<td>QoS indicator of cluster head $i$ at time period $t$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>weighting factor of energy efficiency in cost function</td>
</tr>
<tr>
<td>$\beta$</td>
<td>weightings factor of end-to-end delay in cost function</td>
</tr>
<tr>
<td>$E_{t}(f_i^t)$</td>
<td>transmitting energy consumption cost of cluster head $i$ with $f_i^t$ forwarded packets at time period $t$</td>
</tr>
<tr>
<td>$E_{r}(r_i^t)$</td>
<td>receiving energy consumption cost of cluster head $i$ with $r_i^t$ received packets at time period $t$</td>
</tr>
</tbody>
</table>
\( E_i(r^t_i) \) idle listening energy consumption cost of cluster head \( i \) with \( r^t_i \) received packets at time period \( t \)

\( D(r^t_i) \) end-to-end delay cost of cluster head \( i \) with \( r^t_i \) received packets at time period \( t \)

\( J(r^t_i) \) joint-cost of cluster head \( i \) with \( r^t_i \) received packets at time period \( t \)

\( J^t_i \) cost-to-go function of cluster \( i \) until time period \( t \)

\( U(f^t_i, r^t_i, \theta^t_i) \) system utility function in terms of number of forward packets \( f^t_i \), received packets \( r^t_i \) and the QoS indicator \( \theta^t_i \)

\( U(f^t_i, r^t_i, \theta^t_i) \) pseudo system utility function

\( U_n(f^t_i, \theta^t_i) \) gateway utility function in terms of number of forward packets \( f^t_i \) and the QoS indicator \( \theta^t_i \)

\( U_i(r^t_i) \) cluster utility function in terms of received packets \( r^t_i \)

\( V^t_i \) value function of cluster head \( i \) at time period \( t \)

\( R^t_i \) economic return of cluster head \( i \) at time period \( t \)

\( p^t_i \) bid price of cluster head \( i \) at time period \( t \)

\( p^t_{i,n} \) charge price of gateway \( n \) to cluster head \( i \) at time period \( t \)

\( SO(r^t_i) \) superframe order (SO) of cluster head \( i \) with \( r^t_i \) received packets at time period \( t \)

\( PD \) total packet transmission duration

\( SD \) superframe duration

\( D_{bcn} \) beacon transmission duration

\( P_s \) successful packet transmission duration with Stop-and-Wait ARQ

\( D_p \) transmission duration per packet with Go-Back-N ARQ
$\delta$  waiting time for ACK packet at IEEE 802.15.4 MAC layer

$P_{CCA}$  transmission time of two CCAs in IEEE 802.15.4 MAC layer

$P_L$  per packet transmission time

$P_{ACK}$  ACK packet transmission time
Chapter 1

Introduction

Machine to machine (M2M) communications refer to smart devices or machines automatically communicate with each other through a network with little or no human intervention. It contributes to the integration of advanced Internet of things (IoT) applications and services in various areas, such as smart cities, health care, transport systems, public safety, industrial and agricultural automations [CCC+10, ZYX+11, YZG+13, CCZS08, GC13, LLKC12, XYWV12]. To support IoT applications via wireless technologies, a hierarchical M2M communication architecture has been specified by European Telecommunication Standards Institute (ETSI) M2M Technical Committee [ETS].

1.1 Research Motivation

With the hierarchical M2M communication architecture proposed by ETSI in place, there are still numerous issues remaining for supporting M2M communications. As has been pointed by different standard bodies, channel access of massive amount devices has become one of the potential bottleneck of the M2M communications [3GPP11a, 3GPP11c, Soc12]. The channel access challenges of M2M
communication networks include:

i) **Capacity Limited Devices**: The devices involved in M2M networks have limited capacities in terms of computation and memory. These capacity limited devices are usually very cheap, so that they ensure the affordable deployment of M2M networks. However, with these computational and memory limited devices, the channel access algorithms for M2M communications need to have low computation and storage complexities.

ii) **Energy Efficiency**: Energy efficiency (EE) is one of the most important design considerations due to the fact that most M2M devices are expected to be battery powered. EE is an important design criteria as it is also highly related with the operational costs and profit margins for network operators.

iii) **Network Throughput**: Maximising the network throughput is another essential design criteria. Wireless bandwidth is a scarce resource for M2M communications, due to massive amount of devices, limited spectrum resources, duty-cycled operation and the low data rate of low-power wireless radios. Additionally, serious degradation of throughput may occur due to the network congestion and packet drop when devices generate heavy traffics.

iv) **Adaptivity**: Flexible adaptation is required in many practical scenarios where both channel conditions and data arrival processes are changing over time. In addition, to increase the efficiency of data transmission, the adaptation should be achieved with little or no control information exchange.

v) **Feasibility**: The feasibility of channel access is another important issue that needs to be addressed in real-life scenarios. In practice, it is hard or impossible for devices to have reliable statistical network information, such
as real-time traffics, detailed device capabilities, various application requirements and instant channel conditions. Thus, the access control algorithm needs to work effectively with little or no priori network information.

vi) **Diverse Quality of Service (QoS):** How to effectively multiplex the massive access with enormously diverse QoS requirements turns out to be one of the most challenging tasks. In [ITA14], the diverse QoS requirements of M2M applications are summarised as shown in Table 1-A.

In summary, the critical channel access challenges for M2M communications lies in facilitating channel access management to large number of capacity limited devices while supporting the diverse QoS requirements.

### Table 1-A: Access Requirements of M2M Applications.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Access requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time health services</td>
<td>Strict delay requirements</td>
</tr>
<tr>
<td>Non-real-time services</td>
<td>Periodic access, moderate delay requirements</td>
</tr>
<tr>
<td>Smart grid</td>
<td>Periodic access, moderate delay requirements</td>
</tr>
<tr>
<td>Security/surveillance</td>
<td>Periodic and/or event-driven access, moderate high delay requirements</td>
</tr>
<tr>
<td>Intelligent transport system</td>
<td>Periodic access, strict/moderate delay requirements</td>
</tr>
<tr>
<td>Industrial supply and provisioning</td>
<td>Periodic access, high delay requirements</td>
</tr>
<tr>
<td>Tracking inventory and security</td>
<td>Periodic and/or event-driven access, high delay requirements</td>
</tr>
<tr>
<td>Raw material supply and distribution supervision</td>
<td>Periodic access, high delay requirements</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction

1.2 Research Scope

This thesis focuses on access control management of ETSI compatible hierarchical M2M communication networks, as shown in Fig. 1.1. The aim of this thesis is to optimise the overall network performance, while satisfying different application requirements of hierarchical M2M communications, while exploiting the potential energy efficiency of duty cycle mechanism. Various network dynamics have been taken into consideration, such as network traffics, device capabilities, channel conditions and diverse application requirements.

To do so, a joint scheduling and duty cycle control framework for the ETSI compatible hierarchical M2M communication networks is proposed. The proposed
control framework consists of three control blocks, namely duty cycle control, cluster head control and gateway control.

i) The duty cycle control aims to address the trade-off between energy efficiency and delay for IEEE 802.15.4 based networks.

ii) The cluster head control aims to optimise the cluster empirical network performance and economic return while addressing the trade-off between optimality and algorithm complexity.

iii) The gateway control aims to conduct transmission schedule which optimises the long-term network utility while satisfying different QoS requirements among clusters.

To cope with the requirements of low computational complexity, non-prior network information and adjacent resource slots allocation for M2M communications, different control algorithms are developed for the proposed cluster head control and gateway control.

1.3 Contributions

- A practical and comprehensive Markov decision process (MDP) based network formulation is proposed for the ETSI compatible hierarchical M2M networks. The proposed network formulation takes various dynamics into account, such as time-varying link conditions, stochastic network traffics, various application requirements, and different device capabilities.

---

1The dominate short-range technology IEEE 802.15.4 is considered, as it has been widely adopted for various M2M applications, such as environment monitoring, scientific observation, emergency detection, field surveillance, and structure monitoring [TPS+05, WAL+07, LL09, HVY+09, XRC+04]
• A joint scheduling and duty cycle control framework for M2M communications is proposed. The proposed framework consists of three control blocks: a duty cycle control for IEEE 802.15.4 based networks, a cluster head control and a gateway control. The proposed control framework aims at maximising the aggregated overall network utility and satisfying different application requirements, while exploiting the potential energy efficiency by applying duty cycle mechanism.

• The duty cycle controls for IEEE 802.15.4 based networks are designed based on local traffics. The duty cycle controls aim at jointly optimising the energy efficiency and end-to-end delay for IEEE 802.15.4 MAC protocol with both stop-and-wait automatic repeat request (ARQ) and Go-back-N ARQ schemes. The optimal duty cycle controls are derived by applying dynamic programming (DP). Simulation results shown that the proposed duty cycle controls achieve about 50% reduction in both energy consumption and end-to-end delay as compared to that of the current IEEE 802.15.4 standard.

• The cluster head control aims at optimising the single cluster utility. The optimal cluster head control is achieved by applying DP algorithm. A policy iteration (PI) based cluster control algorithm is proposed, taking into account the computational limitation of M2M devices. Simulation results shown that the proposed optimal cluster head control effectively improves energy efficiency, reduces end-to-end delay and packet drop ratio under various network traffics.

• A low complexity rollout algorithm (RA) based cluster head control is proposed to strike a reasonable balance between algorithm complexity and optimality. Simulation results shown that RA based cluster head control ef-
effectively improves the cluster head utility under various network traffics. In addition, RA based cluster head control achieves an exponential reduction of computational complexity compared with the DP optimal control.

- A reinforcement learning (RL) based cluster head control is proposed for practical M2M networks with non-priori network information, various network dynamics, and time-varying traffics. Simulation results shown that the proposed RL based cluster head control achieves the balance between optimality and stability, compared to the optimal solutions and the existing solutions.

- The gateway control optimises the long-term network utility, while satisfying different QoS requirements among clusters. The gateway control problem is formulated as a utility maximisation problem and the optimal Lagrangian solution is derived by using duality. Simulation results shown that the proposed Lagrangian based gateway control algorithm is capable of dealing with elastic applications and improves energy efficiency.

- A mixed integer programming (MIP) based gateway control is proposed to ensure the adjacent allocation of resource slots to each cluster. Simulation results shown that the MIP based control is capable of dealing with inelastic applications compare with the Lagrangian based control.

1.4 Thesis Structure

This thesis is organised as follow.

Chapter 2 provides a comprehensive introduction of current channel access
management approaches for M2M communication networks. Special attention is given for the IEEE 802.15.4 based hierarchical M2M networks. Then, the background of IEEE 802.15.4 MAC protocol is introduced, followed by the state-of-the-art of the duty cycle control for MAC protocols. In the end, the introduction of relevant methodologies used in this thesis are presented.

**Chapter 3** presents the theoretical model of the ETSI compatible IEEE 802.15.4 based hierarchical M2M networks. The formulation of the system model, network model, traffic model and channel model are provided in detail.

**Chapter 4** shows the designed joint scheduling and duty cycle control framework and simulation platform for the proposed hierarchical M2M networks. The theoretical analysis and derivation of the optimisation problem on maximise overall network utility are presented. In the end, the simulation platform validation is provided.

**Chapter 5** solves the duty cycle control problem for IEEE 802.15.4 MAC protocol with Stop-and-Wait and Go-back-N ARQ schemes. The aim of the duty cycle control is to minimise a joint-cost of energy efficiency and end-to-end delay. The theoretical duty cycle control optimisation problem is derived through dynamic programming (DP). The performance of the proposed control algorithms are evaluated via simulations.

**Chapter 6** focuses on the practical cluster head control for hierarchical M2M networks with the aim of maximising single cluster utility. Utility function is designed for each single cluster head. The utility function consists of both empirical network performance component and economic component. To meet different practical challenges, three control algorithms are proposed. Comprehensive discussion of each cluster head control algorithm are given based on both theoretical analysis
Chapter 1. Introduction

and simulation results.

Chapter 7 focuses on the gateway control with the aim of maximising network utility and satisfying different QoS requirements among clusters. Different utility functions are designed for different applications. The optimal gateway control is derived by applying network utility maximisation (NUM) and MIP approaches. In the end, the performance of the proposed gateway control algorithms are evaluated via simulations.

Chapter 8 presents the conclusions of this thesis. The ideas for future work based on the research carried out in this thesis are also presented.

1.5 Publications List

Journal Publications


Conference Publications


10. Y. Li, K. K. Chai, Y. Chen and J. Loo, “Rollout Algorithm Based Duty Cycle Control with Joint Optimisation of Delay and Energy Efficiency for Beacon-enabled IEEE 802.15.4 Networks”, in Proc. of 20th European Wireless (EW), Barcelona, Spain, May. 2014.


Chapter 2

Background and Methodology

This chapter presents the introduction of current research efforts on channel access management for hierarchical M2M networks. Taking the energy efficiency as key design criteria, special focus is given to the channel access management solutions for IEEE 802.15.4 based hierarchical M2M networks and different duty cycle control mechanisms. In the end, the basics of the relevant methodologies used in this thesis are provided.

2.1 Machine-to-Machine Communications

A machine-to-machine (M2M) communication system is a large-scale network with diverse applications and massive number of interconnected devices, such as sensors, actors, vending machines and vehicles. The devices are embedded in a remote asset and capture data such as temperature, location, consumption, heart rate, stress levels, light, movement, altitude and speed. These data will be transmitted wirelessly to central servers where the applications translate the data into meaningful information that can be analysed and acted upon.

M2M networks represent a future IoT which is expected to be widely utilised
in many fields, such as smart cities, health care, transport systems, public safety, industrial and agricultural automations [CCC+10, ZYX+11, YZG+13, CCZS08, GC13, LLKCI2, XYWV12]. In addition, M2M communications have been listed as one of the horizontal topics of the Europe METIS 2020 project [MET]. More recently, following the study of M2M support in LTE-A networks [3GP11a, 3GP11b, 3GP11c, YZWG14, TLS+13], M2M communications have been identified as one of the key drivers to guide the design of 5G network [CFH14, BJL+14].

In the literature, two M2M communication architectures have been proposed [MMLN12]. One is the infrastructure-based approach which directly utilises cellular networks, such as LTE/LTE-A systems. The other is the infrastructure-less based approach, such as the ETSI proposed hierarchical M2M architecture [ETS].

Enabling M2M communications over LTE/LTE-A systems offers many benefits, due to its ubiquitous coverage, global connectivity with a number of providers, and well developed charging and security solutions [MMLN12]. A comprehensive survey of M2M communications with LTE/LTE-A systems is provided in [GC15], in terms of architectures, service requirements, challenges, and applications. There is a wide range of research for LTE/LTE-A based M2M communications [SJL+13, LC11, NK11, KKB+12, GLA12, WZJ+13, OHL15, WC15, PCLW14, SS14, THH11, HH12, LC12, LCL11, OKM12, ATN+14, DHVV14, LA11, LC11, LMWCFC12, SYCX15, LL15].

ETSI M2M technical committee has proposed an attempt to support M2M communications with a hierarchical architecture. The following part of this section will focus on the infrastructure-less based approach and give the state-of-the-art of channel access management for the ETSI compatible hierarchical M2M communications.
2.2 Hierarchical M2M Communication Networks

The hierarchical architecture consists of M2M Network Domain (ND) and M2M Device Domain (DD). Instead of direct cellular access, a large number of non-cellular M2M devices use short-range, low-cost, low-energy consumption radio interfaces, such as IEEE 802.15.4 or IEEE 802.11 to connect to the base station (BS) via M2M gateways. The networks with the non-cellular M2M devices are called capillary networks [ZHW+12]. The M2M gateways are acting as traffic aggregation and protocol translation points for the capillary networks to LTE/LTE-A networks [GLA12].

The M2M ND is the access and transport network that provides the interconnection of an M2M device or a gateway to application servers. The ND also includes the network management functions and the M2M device management functions. The ND is mainly composed of the core network (CN) and the access network (AN). The CN essentially provides: IP connectivity, interconnection with other networks, roaming with other CNs, service and network control functions. The AN represents the link, e.g. radio access network (RAN), to allow an M2M device or an M2M gateway to access CN services. Beside CN and AN, the ETSI proposed M2M architecture defines network management functions located in the CN or the AN, and M2M management functions located at M2M application level. The M2M DD involves the devices that support one or more M2M applications, by connecting them to application servers through the ND. The M2M DD, also referred as capillary networks in some literatures.

More specifically, the following devices are defined in the M2M DD (capillary networks):
i) **M2M Devices**: devices that can support one or more M2M applications. They are categorised into two classes: 1) LTE devices: they have LTE/LTE-A interface and can connect to the network domain by directly accessing the LTE-A network. 2) Non-LTE-A devices: they do not have LTE-A interface, but form capillary network(s) using short-range network access technologies, such as IEEE 802.15.4 or IEEE 802.11x. They can connect to the network domain through a M2M gateway, and run M2M applications locally.

ii) **Cluster Heads (CHs)**: They can be considered as more powerful M2M devices with some additional capabilities. Like regular M2M devices, they are also part of capillary networks and the communication from a regular M2M device may be directed through and managed by a CH. The functionalities of a CH may include data aggregation, and device management, etc.

iii) **M2M Gateway**: provides the inter connection between the core network, such as LTE-A network and the capillary networks. It provides various functionalities, such as protocol translation, resource management, device management and data aggregation, etc. In some cases, the gateway may also act as an application server which provides M2M services locally in the capillary network. It is expected that the M2M gateway will normally connect to the LTE/LTE-A network with a direct cellular link [GLA12].

The main advantages of hierarchical M2M approach over the LTE/LTE-A based approach lie on i) the low cost and low energy consumption M2M devices which do not require direct cellular interfaces; ii) its ability to aggregate, off-loading and shape M2M traffic for future transport to the LTE/LTE-A networks [Azi12].

With many solutions in place for M2M communications over LTE/LTE-A systems, however, these solutions cannot be applied to the hierarchical M2M networks.
due to the different network architectures, operation radios and device limitations. On the other hand, there are only very limited work focus on the channel access management for hierarchical M2M communications.

A data compression strategy is proposed in [MAH13] with an attempt to reduce the traffic congestion. The proposed strategy operates at the gateways and aims at effectively exploiting the temporal and spatial correlation in sensors observations.

In [LCCZ13], topology control is discussed for M2M networks with two different types of devices. A tree-like distributed topology control algorithm (TLDTCA) is proposed taking the good characteristics of the hierarchical topology.

In [PKH14], a novel medium access control (MAC) protocol for hierarchical M2M networks, more specifically, the backoff time decision rule is proposed and evaluated by simulations. Analytical channel access success probability at cluster heads is derived and validated by simulations.

In [LYC+14], a scalable hybrid MAC protocol, which consists of a contention period and a transmission period, is designed for heterogeneous M2M networks. An optimisation problem is formulated to maximise the channel utility by finding the key MAC parameters.

Besides the aforementioned channel access management specially designed for hierarchical M2M communications, there is still a lack of systematic framework to optimise the overall network performance which fits the dynamic and practical M2M scenarios. With the aid of gateways, each M2M device is able to attach to the existing LTE/LTE-A cellular infrastructure. Thus, subsequent challenge lies in the massive access management for the capillary M2M network.

This thesis focuses on the M2M DD and aims to bridge the gap between theoreti-
ical and practical work in hierarchical M2M networks. Both quantitative insights gained from mathematical theories and practical principles for real dynamic networks are taken into consideration.

2.3 IEEE 802.15.4 based M2M Device Domain

According to EXALTED [EXA], a hierarchical M2M communication project based on ETSI, the benefits of enabling capillary M2M communications over short-range communication technologies in M2M DD are:

i) The energy consumption of devices can be reduced. The devices lifetime is extended by transmitting at lower transmission power.

ii) The interference between devices can be reduced. This will enable the coexistence of a greater number of simultaneous operating networks.

iii) The cost of M2M devices can be reduced. The M2M devices are usually very cheap as they require less functionalities and capabilities.

iv) The traffic for the LTE-A system can be reduced. Thus, the load balance between wide-range communications (LTE-A networks) and short-range communications (capillary networks) is achieved.

IEEE 802.15.4 provides ultra low complexity, ultra low cost, ultra low power consumption, and low data rate wireless connectivity among inexpensive devices [Soc11]. IEEE 802.15.4 is regarded as the dominant short-range technology to implement in the hierarchical M2M networks. IEEE 802.15.4 has been widely adopted for various IoT applications, such as environment monitoring, scientific observation, emergency detection, field surveillance, and structure monitoring [TPS+05, WALJ+07].
Thus, this thesis adopts the IEEE 802.15.4 as the short range wireless technology in the DD of the considered hierarchical M2M networks.

2.3.1 IEEE 802.15.4 Background

The IEEE 802.15.4 standard defines the physical layer (PHY) and MAC layer of the Open Systems Interconnection (OSI) model. The PHY defines frequency, power, modulation, and other wireless conditions of the link. The MAC defines the format of the data handling.

The IEEE 802.15.4 MAC protocol has two operation modes: beacon-enabled mode and non-beacon-enabled mode. The non-beacon-enabled mode uses unslotted carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. The beacon-enabled mode uses a slotted CSMA/CA mechanism and it provides duty cycle based power management mechanism. This thesis focuses on the beacon-enabled mode.

2.3.1.1 Medium Access Control for beacon-enabled IEEE 802.15.4

For beacon-enable IEEE 802.15.4, each time period is composed of three parts: a beacon, a contention access period (CAP), and the optional contention-free period (CFP). Each CAP is divided into 16 equal slots. The slotted CSMA/CA algorithm in the CAP is illustrated in Fig. 2.1 and the relevant parameters are listed in Table 2-A.

The IEEE 802.15.4 slotted CSMA/CA algorithm works as follows. All devices are synchronised via beacon transmission. Transmissions can begin only at the boundaries of time units, called backoff slots, which is denoted by aUnitBackoff-
Figure 2.1: IEEE 802.15.4 CSMA-CA algorithm.
Table 2-A: Parameters in IEEE 802.15.4 CSMA/CA algorithm.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>the number of times required to back off while attempting the current transmission</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CW</td>
<td>contention window length</td>
<td>initialise to $CW_0$</td>
<td></td>
</tr>
<tr>
<td>$CW_0$</td>
<td>initial value of CW</td>
<td>950 MHz band, 1; otherwise, 2.</td>
<td></td>
</tr>
<tr>
<td>$macMinBE$</td>
<td>minimum value of BE</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$macMaxBE$</td>
<td>maximum value of BE</td>
<td>3 - 8</td>
<td>5</td>
</tr>
<tr>
<td>$macMaxCSMABackoff$</td>
<td>The maximum number of backoffs attempt before declaring a channel access failure</td>
<td>0 - 5</td>
<td>4</td>
</tr>
</tbody>
</table>

Period = 20 symbols. A device with a packet ready for transmission first back off for a random number of backoff slots before sensing the channel. The number of backoff slots is chosen uniformly between $[0, macMinBE]$. $macMinBE$ is the minimum value of the backoff exponent (BE). The default value of BE is 3. The random backoff and the following channel sensing aim to reduce the probability of collisions among contending devices and ensure the channel is clear for a contention window (CW) duration. The CW duration is of two backoff slots in IEEE 802.15.4. This means that the device applies two channel clear assessments (CCAs) before transmitting. If the channel is found busy, BE is incremented by one, and a new backoff stage starts before channel sensing. This process is repeated until either BE equals $aMaxBE$ (the maximum value of the BE whose default value is five) or until a certain maximum number of permitted random backoff stages is reached.

In the former case, the value of BE is frozen at $aMaxBE$ in the backoff periods. In the later case, an access failure is declared to the upper layer. The maximum number of permitted random backoff stages is determined by the parameter $macMaxCSMABackoff$, which has a default value of 4.
Furthermore, IEEE 802.15.4 supports optional retransmission scheme based on acknowledgements in beacon-enabled mode \[\text{ACF09}\]. When retransmissions are enabled, the receiver device must send a positive acknowledgement right after receiving a data frame. Acknowledgment and beacon frames are sent without using CSMA/CA mechanism.

### 2.3.1.2 Duty Cycle Control of IEEE 802.15.4

One of the most attractive features of IEEE 802.15.4 is that it utilises the duty cycle mechanism to save energy. A device can optionally bound its channel access time using a superframe structure. A superframe is bounded by the transmission of a beacon frame. A superframe has an active portion and an inactive portion. The devices enter sleep mode during the inactive portion to save energy. The duration between two consecutive beacons is called beacon interval \((BI)\), while the duration of an active period is called superframe duration \((SD)\), where

\[
BI = a_{\text{BaseSuperFrameDuration}} \times 2^{BO}, \quad (2.1)
\]

\[
SD = a_{\text{BaseSuperFrameDuration}} \times 2^{SO}, \quad (2.2)
\]

and beacon order \((BO)\) and superframe order \((SO)\) are two integers ranging from 0 to 14 \((0 \leq SO \leq BO \leq 14)\), and \(a_{\text{BaseSuperFrameDuration}} = 15.36\text{ms}\) at 2.4 GHz with 250 kbps bandwidth. The duty cycle is defined as the ratio of the active portion over each BI, thus

\[
Duty\ Cycle = SD/BI = 2^{SO-BO}. \quad (2.3)
\]

The superframe structure in multi-hop scenarios is shown in Fig. 2.2. For each
cluster head, the $BI$ is divided into two superframes, named incoming superframe and outgoing superframe. The cluster head receives the beacon from its parent gateway in the incoming superframe, and transmits its beacon in the outgoing superframe. As there are two $SDs$ in each $BI$, according to (2.1) and (2.2), $SO \leq BO - 1$ for all cluster heads.

![Figure 2.2: Superframe structure of IEEE 802.15.4.](image)

Depending on the value of the parameter $BO$, the duration of each $BI$ in 2.4 GHz varies from 15.36ms to 251.7s [CWGD10]. The duty cycle of different devices are equal in the current standard [Soc11]. Due to the fixed duty cycle, it is not possible for devices to dynamically change its own duty cycle to the time varying or spatially no-uniform traffic load among the network. In addition, the fixed duty cycle may result in high latency, especially for multi-hop networks, where devices on the routes may have different schedules [BDWL10] and the sleep delay will be accumulated hop by hop [SRS12].

### 2.3.1.3 Existing Adaptive Duty Cycle Control for IEEE 802.15.4

Determining the duty cycle is a crucial problem in beacon-enabled IEEE 802.15.4. However, such problem is complicated because simple and accurate models for
duty cycle control on reliability, delay and power consumption are not available. Moreover, it hard or impossible to compute the optimal duty cycle due to the scarce computational capacity of the devices and the lack of prior information of the network in real-life scenarios.

Some work have been done on duty cycle adaptation for IEEE 802.15.4 MAC protocol. A comprehensive survey of duty cycle control for beacon-enabled IEEE 802.15.4 MAC protocol can be found in [KGM14]. Some of the main duty cycle control approaches are listed below.

The earliest work on IEEE 802.15.4 duty cycle adaptation is known as the Beacon order adaptation algorithm (BOAA) [NPK]. In this work, the duty cycle adaptation is triggered by the change of traffic load in the network. The network traffic load is estimated from the number of packets received by the end devices. The proposed algorithm has the memory advantage as a buffer matrix is used to store the received packets. However, this buffer matrix made the algorithm unscalable for large networks which leads to the increase of end devices in the network.

Later, a duty cycle algorithm (DCA) is proposed in [JLHK07]. The duty cycle selection is based on additional information such as queue occupation and end-to-end delay. The reserved frame control field presented in the MAC frame header is modified, thus no extra overhead is incurred. The queue occupation is computed as the average of a queue indicator embedded in all frames during the active periods. However, this result is under the estimation that the number of transmission requested by devices decrease along with the decrease of active durations.

Authors in [LHJ+07] proposed the extension of the CAP based on a busy tone sent by devices. The busy tone is only sent if the device has failed to transmit all its
data frames. In [LJ08], the CAP extension is done based on the type of the traffic that is waiting in the queue. Specifically, the CAP is extended if there is some real-time data in the transmit queue at the end of the CAP. These extensions are however not compliant with the standard as they added the extension immediately after the CAP thus modifying the superframe structure. In addition, they are not proportional envisaged to the amount of traffic waiting to be transmitted. Finally, they are not as flexible as duty cycle adaptations because they still work on fixed active and beacon interval durations.

A duty cycle learning algorithm (DCLA) [AP12] adapts the duty cycle during run time without the need of human intervention. DCLA aims to minimise power consumption while balancing probability of successful data delivery and delay constraint of the applications. DCLA running on coordinator devices, it collects network statistics during each active duration to estimate the incoming traffic. Then, at each beacon interval the reinforcement learning (RL) is used to learn the optimal duty cycle. The algorithm is formulated as a multi-armed bandit problem where the agent’s objective is to minimise idle listening and buffer overflows. DCLA adjusts its policy for selecting the duty cycle according to the feedback information provided by the network. Both the MAC parameters (BO, SO) are taken into consideration to find a compromise among beacon overhead and queuing delay. Basically, RL depends on repetitive interactions. The selected duty cycle is updated iteratively till the optimal one, which achieves the targeted performance.

In this way, DCLA achieves a fully adaptive system that can self-correct its parameters based on the traffic conditions, without the need for any manual configurations to specific requirements of different applications. This gives DCLA the credit of reducing time and cost of installation, operation and management. How-
ever, the learning processes presented slow convergence for non-stationary scenarios due to the selection of a large state-action space.

2.3.2 Delay Aware Duty Cycle Based MAC Protocols

Besides the duty cycle controls specifically designed for IEEE 802.15.4, there are some other duty cycle control mechanisms for general duty cycle based MAC protocols. In this part, the work with the aim of designing energy efficient and delay aware duty cycle based MAC protocols is given.

Sensor MAC (SMAC) [YHE02] is the fundamental work of energy efficient MAC protocol, which first introduces the periodic duty cycle concept to reduce idle listening power consumption. The key issue to improve the performance of SMAC lies on finding the adaptive optimal tradeoff between the sleep delay and energy efficiency [BDWL10].

The existing duty cycle controls based on SMAC are classified according to their specific approach and the aimed problem, as shown in Fig. 2.3. The detailed discussion on these controls are given in the following sessions.

2.3.2.1 Adaptive Active Periods

The motivation of solutions with this approach is to alleviate the high latency problem when the traffic load is high, while reducing the idle listening when the traffic load is low.

Timeout MAC (TMAC) [DL03] is the classic protocol to solve the fixed active
Chapter 2. Background and Methodology

Adaptive duty cycle based synchronised MAC protocols

Adaptive active periods
Path-aware duty cycle schedule
Delay-aware duty cycle control
Adaptive sleep periods

Figure 2.3: Classification of duty cycle based MAC protocols

period problem. With TMAC, the device can switch into sleep mode when it does not expect further transmission by predicting channel activity during an active period. A utilisation based MAC (U-MAC) \[^{YTW+05}\] works similar with TMAC. UMAC controls the length of active periods based on a utilisation function, which is the ratio of the actual transmission and receptions performed by the device.

The main problems of this approach are: Firstly, by having device ending their active periods prematurely, TMAC and UMAC partially break the synchronisation among the devices, which leads to the early sleep problem \[^{BDWL10}\]. The early sleep problem happens when a third-hop device, which supposes to be the next relay of an ongoing transmission, prematurely goes to sleep. Secondly, the
uniformed duty cycle for all devices is not flexible when each device generates
different amount of traffic with different quality of service (QoS) requirements.

2.3.2.2 Adaptive Sleep Periods

Sleep delay is the delay caused by waiting to be transmitted when devices are in
sleep mode. Sleep delay is the main cause of delay due to the long sleeping periods
in low duty cycle MAC protocols. To reduce the sleep delay, some of the research
try to adapt the length of sleep periods according to the traffic conditions in the
network.

The idea of achieving a good trade-off between energy efficiency and end-to-
end delay through adaptive sleep intervals was first explored by dynamic sensor
MAC protocol (DSMAC) [LQW04]. DSMAC is able to dynamically change the
sleeping intervals. The key idea of DSMAC is to maintain the synchronisation,
thus the initial active periods never get changed. Devices reduce the length of
sleep periods by inserting extra active periods in the middle of the sleep periods
when less latency is required or when it observes an increasing traffic load. Thus
the duty cycle adaptation of DSMAC can only be double or half of the initial
setting.

Similar with DSMAC, the fast path algorithm (FPA) proposed in [LYH05]
provides fast data forwarding paths by adding additional active periods along the
paths from the sender to the destination. A device uses its hop distance from the
sender to estimate when its upstream neighbour will send a packet to it. Then,
the device wakes up at the estimated time to receive and potentially forward the
packet to its next-hop neighbour, so that devices along the path can wake up at
the right time to avoid schedule misses.
The concern with this approach is its implementation issue. To establish the schedule the device must have some additional information, such as the average latency and the transmission path information, which increases the overhead of the transmission.

### 2.3.2.3 Path-aware Duty Cycle Schedule

The key idea of this approach is to sequentially schedule the active periods for the devices along the transmission path. Thus, with a properly schedule, the packets can be sent out immediately hop by hop once received without queuing.

RMAC [DSJ07] is a novel scheme to cope with the end-to-end delay accumulation. Each operation cycle contains a Synch, Data, and a Sleep period. To reduce end-to-end latency, RMAC forwards a control frame called PION over multi-hops during a Data period, and schedules the upcoming data packet delivery along that route. Each intermediate device along the data packet delivery route sleeps and wakes up at the scheduled time to forward data (only devices involved in the communication will wake-up, while others keep in sleep mode). However, RMAC has a long latency when packet transmission errors occur, and two hidden sources that have succeeded in scheduling through PIONs will cause collisions at the beginning of the next sleep period. In this case, RMAC simply resumes the packet transmission in the next time period without any immediate packet retransmission.

Demand-wakeup MAC (DW-MAC) [SDGJ08] relies on RMAC’s approach and tries to reduce the delay caused by collisions and transmission errors. DW-MAC assumes that a separate protocol is used to synchronise the clocks in sensor devices during the Sync period, which ensures the required precision. The DW-MAC’s scheduling mechanism ensures that data transmissions do not collide at
their intended receivers. The proposed protocol works under random topology and supports both unicast and broadcast traffics. DW-MAC sets up a one-to-one mapping between a Data period $T_{Data}$ and the following Sleep period $T_{Sleep}$. In such a way that the time, when the scheduling frame is transmitted, determines the corresponding offset of the DATA packet exchange.

### 2.3.2.4 Delay-aware Duty Cycle Control

With this approach, the end-to-end delay requirement of the whole transmission is broken down into several single-hop delay requirements.

In [WWXY10], DutyCon is proposed to guarantee end-to-end delay by assigning a local delay requirement to each single hop along the transmission path. However, as a feedback controller is designed, this approach requires significant amount of signalling from the neighbour devices to compute the delay. To reduce the signalling among neighbour devices, a distributed duty cycle controller is proposed in [BY13] aiming at controlling the local queue length of the device to be the same as the predetermined threshold. The distributed duty cycle control is achieved by adjusting the sleep duration of each device based on its local queue length independently. However, this approach needs specific syntonisation scheme, and the evolution of the proposed control requires carefully setting of the initial duty cycle and control parameters.


2.4 Relevant Control and Scheduling Methodologies

In this section, the introduction of relevant methodologies used to solve the massive access and various QoS requirements problem for IEEE 802.15.4 based hierarchical M2M networks are given. The mentioned methodologies are stochastic control, dynamic control and approximate dynamic control.

Stochastic control is a subfield of control theory that deals with the existence of uncertainty that drives the evolution of the system. Optimal control is one of the most useful systematic methods to provide solutions to control problems.

Stochastic optimal control aims to design the desired control variables over time so that the controlled task is accomplished with minimum cost or maximum reward under the existence of uncertainty. The stochastic optimal control has found its applications in areas such as industrial control systems, inventory management, dynamic resource allocation, production planning, queuing networks, finance and so on [Ast70, Aok76, BSLG74, KS98, SZ94, YZ99].

In M2M communication networks, a massive amount of M2M devices are random distributed in large areas. These M2M devices generate different traffics and communicate with each other through stochastic wireless channels, and the transmitted signals are subjected to random fading. Thus, modelling and optimisation of the emerging M2M communication networks are naturally resort to stochastic optimal control.
Chapter 2. Background and Methodology

The general form of the discrete-time stochastic system is described as,

\[ s_{t+1} = f_t(s_t, a_t, w_t), \quad t = 0, 1, \cdots, T - 1 \] (2.4)

where

- \( t \) is the discrete time periods of the control;
- \( s \) is the state of the system, which summarises past information that is relevant for future optimisation;
- \( a \) is the control, which is the decision to be selected at time \( t \) from a given valid control set;
- \( w \) is the random parameter, which is also called ”disturbance” or ”noise”.

Dynamic programming (DP) offers a unified approach to deal with complex multistage stochastic problems by breaking them down into simpler subproblems \[\text{[Ber05]}\].

*Principle of optimality* and *Bellman equation* are two conditions for problems can be solved by DP. The *principle of optimality* means that the tail policy is the optimal solution to the tail subproblem, in other words, the optimisation of the future does not depend on what we did in the past.

The DP optimal algorithm works as follows: first, the cost-to-go function \( J_t(s) \) is computed from time period \( t \) back to time period 0; Then, the policy from time period 0 to time period \( t \) is selected with the minimum \( J_t(s) \) and the control is decided based on the selected policy.

Approximate dynamic programming (ADP) is a method for modelling and solving DP problems that are large, complex, and stochastic. ADP often presents as a
method for overcoming the classic *curse of dimensionality* of DP by finding good suboptimal solutions [Pow11].

The basic idea of ADP is on replacing the cost-to-go function $J(s)$ by an approximation $\tilde{J}(s)$. Thus, this method applies at time period $t$ and state $s_t$ a control $\mu_t(s_t)$ that minimise the cost-to-go function over $u_t \in U_t(s_t)$,

$$\mathbb{E}\left\{g_t(s_t, u_t, w_t) + \tilde{J}_{t+1}(f_t(s_t, u_t, w_t))\right\}. \quad (2.5)$$

The corresponding suboptimal policy $\bar{\pi} = \{\bar{\mu}_1, \bar{\mu}_2, \cdots, \bar{\mu}_T\}$ is determined by the approximate cost-to-go functions $\tilde{J}_1, \tilde{J}_2, \cdots, \tilde{J}_T$. The approximation of the cost-to-go function is calculated either by functional form or by an algorithm to calculate their values at each state.

There are several alternative approaches for selecting or calculating the approximate cost-to-go function $\tilde{J}(s)$, rollout algorithm and reinforcement learning (RL) are two widely used online approximation methods [Pow11].

### 2.5 Summary

This chapter provides the introduction of current research efforts on massive access control for hierarchical M2M networks. With the focus on IEEE 802.15.4 based capillary M2M DD networks, the state-of-the-art review on duty cycle controls with the aim of joint improving energy efficiency and delay are presented. In the end, the relevant methodologies applied in this thesis are provided.
Chapter 3

Hierarchical Machine-to-Machine Communication Networks

In this chapter, the mathematical formulation of an IEEE 802.15.4 based capillary M2M system is presented. Then, the network model, traffic model and channel model of the formulated capillary M2M system are given.

3.1 System Model

The LTE/LTE-A base station (BS), M2M gateways, cluster heads and M2M devices are four types of devices in an ETSI compatible M2M network [ETS EXA]. A set of statically-deployed devices with uplink transmission to BS is considered in this thesis.

The network operates at discrete time domain where time is divided into time periods $t = 0, 1, \ldots, T$. Each device is equipped with a single omnidirectional antenna. As shown in Fig. 3.1 the set of M2M gateways is denoted as $\mathcal{N}$. The child cluster heads of gateway $n$ forms the cluster head set $\mathcal{I}_n$ and the link set between gateway $n$ and cluster head $i$ is denoted as $\mathcal{L}_{i,n}$. The immediate child
devices of cluster head $i \in \mathcal{I}_n$ form the child devices set $\mathcal{C}_i$ and the link set between cluster head $i$ and its child device $j \in \mathcal{C}_i$ is denoted as $\mathcal{L}_{i,j}$. The link $l \in \mathcal{L}_{i,j}$ or $l \in \mathcal{L}_{i,n}$ can also be represented by device pair $(i,j)$ and device pair $(i,n)$, respectively.

For a given capillary M2M network, different clusters may run different applications. According to [ZHW+12], four classes of applications are considered in this thesis,

i) Elastic applications: these applications are rather tolerant of delays, the environment monitoring is a typical example of such application.

ii) Hard real-time applications: these applications need their data to be served within a given delay constraint. Vehicle and asset tracking are typical M2M service application fall into this class.

iii) Delay adaptive applications: these applications are delay sensitive but can be made rather tolerant of occasional delay bound violation and dropped
packets. A typical example is remote monitoring in e-Health services.

iv) Rate-adaptive applications: these applications adjust their transmission rates according to available radio resources while maintaining moderate delays. Video transmission is one example application in this class.

3.2 Network Model

3.2.1 Markov Decision Process

Markov decision process (MDP) is a discrete time stochastic control process. MDP provides a mathematical approach for modelling decision making in situations where outcomes are partly random and partly under the control of a decision maker or controller.

A controller interacts with the system by taking actions based on its observations at each discrete decision period. Associated with each action in each state, there is a cost to the controller. Typically, an action will lead the system from the current state to some other states with certain probabilities. The goal of the controller is to minimise the total cost over finite or infinite time horizon by making sequential decisions based on its current observations.

An general MDP based system is described by a tuple \(< S, A, P, J, \gamma >\), where

- \( S \) is a finite set of states;
- \( A \) is a finite set of actions (alternatively, \( A_s^t \) is the finite set of actions available at state \( s_t \));
Chapter 3. Hierarchical Machine-to-Machine Communication Networks

- \( P_a(s, s') \) is the probability that action \( a \) in state \( s_t \) at time \( t \) will lead to state \( s' \) at time \( t + 1 \), thus \( P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a) \);

- \( J_a(s, s') \) is the immediate cost (or expected immediate cost) of transition to state \( s' \) from state \( s \);

- \( \gamma \in [0, 1] \) is the discount factor, which represents the importance between future costs and the present cost.

The core problem of a MDP system is to find a policy \( \pi \) for the controller. A policy \( \pi \) specifies the action rules \( \mu_t \) the controller will follow at state \( s_t \),

\[
\pi = \mu_1, \mu_2, \ldots, \mu_T,
\]

where \( \mu_t \) maps states \( s_t \) into control \( a_t = \mu_t(s_t) \), such that \( \mu_t(s_t) \in a(s_t) \) for all \( s_t \in S \).

Note that MDP is an extension of Markov chains. Once a MDP is combined with a particular policy, which fixes the action for each state, the resulting combination of the MDP and the policy behaves as a Markov chain.

MDPs are useful for studying a wide range of optimisation problems which can be solved by applying DP and RL.

### 3.2.2 Markov Decision Process based Network Formulation

The access control of the formulated hierarchical capillary M2M network is formulated as a finite-state MDP. For each cluster head \( i \in \mathcal{I}_n \), the cluster is described
as a tuple $< I, S, A, P, U, \gamma >$ where

- $I$ is the total number of cluster heads connected to gateway $n$;

- $S$ is a finite set of discrete states of cluster heads. For the cluster head $i$ at time period $t$, its state $s_i^t \in S$ is represented by its queue length $q_i^t$;

- $A_i$ is the finite set of control decision available to the cluster head $i$, and $A = A_1 \times A_2 \times \cdots A_I$;

- $U$ is the utility function of cluster heads. Thus, for the cluster head $i$, $s_i^t \times a_i^t \mapsto U_i^t$.

- $P$ is the transition function. $S \times A \times S \mapsto [0,1]$, and $P(s_i^t, a_i^t, s_i^{t+1}) = 1$ represents the transmission link between the M2M device and its cluster head meets the sensitivity requirements of the cluster head $i$. Note that $\forall s_i^t \in S, \forall a_i^t \in A, \sum_{s_i^{t+1} \in S} P(s_i^t, a_i^t, s_i^{t+1}) = 1$.

- $\gamma \in [0,1]$ is the discount factor, which represents the difference between future utility and current utility.

For each cluster head, the MDP is repeated over $T$ time periods, as shown in Fig. 3.2. More specifically, at each time period $t$, each cluster head $i (0 \leq i \leq I)$ will senses its own queue length $q_i^t \rightarrow s \in S$, then the cluster head $i$ decides the number of packets it will receive $r_i^t \rightarrow a \in A_i$. As a result, the queue length of cluster head $i$ transits to $q_i^{t+1} \rightarrow s' \in S$ according to transition probability $P_{ss'}(a) \in P$ and thereby generates a utility $U_i^t = U(s, a)$ passing to the cluster head $i$.

As shown in Fig. 3.2, the access control between each cluster head $i$ and its child M2M devices follows the formulated MDP $< I, S, A, P, U, \gamma >$, where $S$ is the queue length of the cluster, $A$ is the cluster head control decision and $U$ is the
3.3 Traffic Model

The queue length of a device is defined as the number of packets in the transmission queue. Assume all generated and received packets are available at the beginning of each time period $t$. Let $q_i^t$ denote the queue length of the cluster head $i$ at the beginning of superframe at time period $t$. The change of queue length of cluster head $i$ is given as

$$q_i^{t+1} = \min \left( \left[ q_i^t + r_i^t - f_i^t + g_i^t \right]_, q_i^{\max} \right),$$  \hspace{1cm} (3.2)

where $0 \leq t \leq T - 1$, $\left[ \cdot \right]_+ = \max(0, \cdot)$, $g_i^t$ is the number of packets being generated by device $i$ at time period $t$; $f_i^t$ is the number of packets transmitted by cluster head $i$ at time period $t$; and $r_i^t$ is the number of packets received by cluster head $i$.
at time period $t$, as shown in Fig. 3.3. In addition, let $q_{i}^{\text{max}}$ denote the maximum queue length of cluster head $i$, which is also the buffer size of cluster head $i$. The new arrived packets will be dropped if the queue length reaches its maximum $q_{i}^{\text{max}}$, as shown in Fig. 3.4.

![Figure 3.3: Change of queue length of devices.](image1)

![Figure 3.4: Packet drop due to reach the buffer size.](image2)

The batch Poisson process is applied as the traffic arrival model for all devices. This is different from the assumption in most of the existing work that there is only one packet being transmitted in each time period. In this batch Poisson process model, the number of arrived packets, a batch, follows the Poisson process. The number of packets of the link $l_{i,j}$ batch is denoted by $g_{i,j}^{t}$. $g_{i,j}^{t}$ is identically and independently Poisson distributed.
Since the traffic arrival of each M2M device follows a batch Poisson process, the traffic arrival of each cluster follows a compound Poisson distribution. Thus, the arrived traffic of each cluster $g_t^i = \sum_{j \in C_i} g_{t,j}^i$.

This traffic model is adopted in the utility optimisation problem formulation and access management control in the following chapters. In addition, it is also implemented in the simulations to evaluate the performance of all proposed controls.

### 3.4 Channel Model

To fit the practical scenarios, this thesis adopts an empirical dual-slope propagation model of path loss with distance, Nakagami frequency-flat small-scale fading, and lognormal shadowing [NIST1]. Based on this dual-slope propagation model, the overall channel propagation loss is expressed as

\[ L_{c,\text{dB}} = L_{0,\text{dB}} + X_{s,\text{dB}} + X_{f,\text{dB}} \]

\[ = L_{0,\text{dB}} + \begin{cases} 
10n_0 \log(d) & d \leq d_1 \\
10n_0 \log(d_1) + 10n_1 \log\left(\frac{d}{d_1}\right) & d > d_1
\end{cases} \tag{3.3} \]

where $L_{0,\text{dB}}$ is the reference path loss where the distance between the sender and receiver is 1m; $X_{s,\text{dB}}$ is a zero mean Gaussian random variable with standard deviation $\sigma_s$. $d$ is the distance between the sender and receiver and $d_1$ is the reference distance; $X_{f,\text{dB}} = 10 \log(X_f)$ and $X_f$ is a unit-mean gamma-distributed random variable with variance $1/m$ ($m$ is the Nakagami fading parameter), and all logarithms are base 10.
Chapter 3. Hierarchical Machine-to-Machine Communication Networks 41

As 2.4GHz is globally available for the IEEE 802.15.4, Table 3-A, Table 3-B and Table 3-C list the channel model parameters at 2.4GHz for the modelled capillary M2M network. These parameters are set based on the data provided in [NIST].

Table 3-A shows the parameters of the channel model for indoor-indoor environments. Each environment is further differentiated into line-of-sight (LOS) and non-line-of-sight (NLOS) conditions. Note that in some environments the applied dual-slop model simply collapses to the single-slope model by setting $n_1 = n_0$ and choosing $d = d_1$. And the path loss exponents are stated relative to the free space path loss. Table 3-B shows the channel model parameters for outdoor-outdoor urban-canyon environments. Table 3-C shows the channel model parameters for outdoor-indoor environments. Because of the nature of the outdoor-indoor environment, all parameters are for NLOS conditions.

<table>
<thead>
<tr>
<th>Environment</th>
<th>$L_0$</th>
<th>$n_0$</th>
<th>$n_1$</th>
<th>$d_1$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>residential</td>
<td>16.3</td>
<td>2.2</td>
<td>2.2</td>
<td>1</td>
<td>2.4</td>
</tr>
<tr>
<td>office</td>
<td>22.8</td>
<td>1.2</td>
<td>1.2</td>
<td>1</td>
<td>1.7</td>
</tr>
<tr>
<td>industrial</td>
<td>22.4</td>
<td>1.1</td>
<td>1.1</td>
<td>1</td>
<td>2.1</td>
</tr>
<tr>
<td>cinder block</td>
<td>24.2</td>
<td>1.5</td>
<td>1.5</td>
<td>1</td>
<td>2.8</td>
</tr>
<tr>
<td>NLOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>residential</td>
<td>12.5</td>
<td>2.2</td>
<td>5.6</td>
<td>11</td>
<td>3.0</td>
</tr>
<tr>
<td>office</td>
<td>26.8</td>
<td>2.2</td>
<td>6.7</td>
<td>10</td>
<td>3.7</td>
</tr>
<tr>
<td>industrial</td>
<td>29.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1</td>
<td>6.3</td>
</tr>
<tr>
<td>cinder block</td>
<td>9.1</td>
<td>4.9</td>
<td>4.9</td>
<td>1</td>
<td>6.7</td>
</tr>
</tbody>
</table>

The received power $P_{\text{rec},i}^{t}$ of cluster head $i$ is a function of the transmitted power, antenna gains, and channel attenuation. It is modelled (in decibels referenced to
Table 3-B: Channel Model Parameters of Outdoor-outdoor Environments

<table>
<thead>
<tr>
<th>Environment</th>
<th>( L_0 )</th>
<th>( n_0 )</th>
<th>( n_1 )</th>
<th>( d_1 )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOS urban-canyon</td>
<td>6.9</td>
<td>1.7</td>
<td>1.7</td>
<td>1</td>
<td>2.4</td>
</tr>
<tr>
<td>NLOS urban-canyon</td>
<td>21.3</td>
<td>1.6</td>
<td>1.6</td>
<td>1</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 3-C: Channel Model Parameters of Outdoor-indoor Environments

<table>
<thead>
<tr>
<th>Environment</th>
<th>( L_0 )</th>
<th>( n_0 )</th>
<th>( n_1 )</th>
<th>( d_1 )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NLOS) office</td>
<td>0.2</td>
<td>2.0</td>
<td>4.2</td>
<td>70</td>
<td>3.3</td>
</tr>
<tr>
<td>high-rise</td>
<td>8.8</td>
<td>2.2</td>
<td>2.2</td>
<td>1</td>
<td>5.6</td>
</tr>
<tr>
<td>convention center</td>
<td>4.2</td>
<td>0.6</td>
<td>3.7</td>
<td>100</td>
<td>4.6</td>
</tr>
<tr>
<td>mine tunnel</td>
<td>5.7</td>
<td>0.7</td>
<td>18.3</td>
<td>70</td>
<td>5.8</td>
</tr>
</tbody>
</table>

1mw) as

\[
P_{rec,i}^t = P_{trans,i}^t + G_{trans,i}^t + G_{rec,i}^t - L_{c,i}^t,
\]  

(3.4)

where \( P_{trans,i}^t \)(dBm) is the conducted power to the transmit antenna (dBm), \( G_{trans,i}^t \)(dBi) and \( G_{rec,i}^t \)(dBi) are the transmit and receive antenna gains (dBi), respectively, \( L_{c,i}^t \) is the loss due to channel propagation.

It is assumed that the fading and shadowing are constant during each time period. The condition for the successful transmission is that the received signal power \( P_{rec,i}^t \) is above the sensitivity threshold \( P_{sens,i}^t \)(dBm) of the device. The
successful transmission probability $\rho_{i,j}^t$ is denoted as,

$$
\rho_{i,j}^t = \begin{cases} 
1 & P_{rec,i}^t \geq P_{sens,i}^t, \\
0 & P_{rec,i}^t < P_{sens,i}^t.
\end{cases}
$$

(3.5)

The transition function $P_{i,j}$ gives the successful transmission probability that action $a$ in states $s$ at time period $t$ will lead to state $s'$ at next time period $t+1$:

$$
P_{i,j}(s,a,s') = P_{ss'}(s^{t+1} = s'|a^t = a, s^t = s, \rho_{i,j}^t = 1).
$$

(3.6)

As defined in the standard, the transmission rate of IEEE 802.15.4 is 250kbps at 2.4GHz. Due to the duty cycle mechanism, the amount of data could be transmitted within each $BI$ should be the transmission rate times the active period length. Thus, we denote the equivalent actual link capacity as

$$
C_{i,n}^t = 250kbps \times \text{duty cycle}.
$$

(3.7)

### 3.5 Summary

In this chapter, the system model, network model, traffic model and channel model of IEEE 802.15.4 based hierarchical capillary M2M network are presented. The system of the capillary M2M communication takes elastic applications, hard real-time applications, delay adaptive applications and rate-adaptive applications into consideration. The network formulation is based on discrete time MDP. The traffic model of the M2M network is formulated into a batch Poisson process and an empirical dual-slope propagation model is adopted as the channel model.
Chapter 4

Joint Scheduling and Duty Cycle Control Framework

In this chapter, the utility maximisation problem of the modelled IEEE 802.15.4 based hierarchical capillary M2M networks is formulated. The formulated network utility optimisation problem takes link constraints, application types and device capacities into consideration. Then, a joint scheduling and duty cycle control framework for M2M communication networks is presented. The design of simulation platform and validation are presented at the end of this chapter.

4.1 Network Utility Optimisation for Hierarchical M2M Communications

For the modelled hierarchical M2M communication networks with multi-class applications in Chapter 3, the network utility optimisation problem aims not only to maximise the overall network utility, but also meet different QoS requirements among clusters.

In order to optimise the network utility, the utility function for each applica-
tion needs to be designed carefully. The widely adopted utility function design approach uses a logarithmic function for the elastic source (elastic applications), and a sigmoid function for the inelastic source (hard-real time applications, delay adaptive applications and rate adaptive applications) [She95, JSKP10, ZHW+12]. Fig. 4.1 illustrates the utility function of each application class considered in this thesis.

![Utility Illustration for Different Applications](image)

**Figure 4.1: Utility illustration for different applications.**

Different from the formulation in the existing work, the problem formulation of the network utility optimisation in this thesis takes multi-class applications into consideration by introducing the application factor $\theta_i^t$ of each cluster head $i$. The application factor $\theta_i^t$ indicates, i) the application class of each cluster, and ii) the priority of clusters within the same application class.

Application factor $\theta_i^t$ is considered as one of the parameters in the proposed network optimisation problem. Then, the joint scheduling and duty cycle control
optimisation problem for the modelled hierarchical M2M networks with multi-class applications is formulated as,

\[ P_1 : \min \sum_{i=0}^{T} \sum_{i \in \mathcal{I}_n} U_i(f^t_i, r^t_i, \theta^t_i) \]  
\[ s.t. \sum_{i \in \mathcal{I}_n} f^t_i \leq C^t_{n,b} \]  
\[ 0 \leq f^t_i \leq C^t_{i,n}, \quad i \in \mathcal{I}_n \]  
\[ r^t_i \leq \min(C^t_{i,n}, q^{\text{max}}_i), \quad i \in \mathcal{I}_n. \]

The objective of \( P_1 \) is to maximise the network long-term aggregated utility, taking into account the application indicator \( \theta^t_i \), number of packets each cluster head received \( r^t_i \), and number of packets each cluster head forward \( f^t_i \) of cluster \( i \) at time period \( t \). The constraints (4.1b) to (4.1d) are the link capacity constraints stating that the total transmitted packets of each link should be no more than its link capacity or its maximum queue length.

It is worth noting that the utility functions for the inelastic applications (hard-real time applications, delay adaptive applications and rate adaptive applications) are not concave functions. This makes the problem \( P1 \) hard to be solved by classic network utility maximisation (NUM) directly.

The NUM problem was firstly formulated as an optimisation problem in [KMT98]. The objective of NUM in [KMT98, LL99] was to maximise the aggregate utility in wired networks with delay-tolerant applications for all users under the link capacity constraints. In [LMS05, Kur12], the NUM framework was extended to wireless networks with real-time applications. Using duality, the distributed iterative algorithm for the optimal rate allocation was achieved.
The overall objective of NUM is to maximise the utility $U(s)$. The problem can be achieved by solving the following primal optimisation problem,

$$
\mathcal{P} : \max \sum_{s=1}^{S} U(s) \quad (4.2)
$$

$$
s.t. \quad \sum_{s \in S_l} s \leq c_l, l = 1, \cdots, L.
$$

At convergence, the bandwidth proportional fairness is achieved for a logarithmic utility function. The rate allocation is fair with respect to utility percentage. A resource allocation $x^* = [x^*_1, x^*_2, \cdots, x^*_S]^T$ is proportionally fair, if it is feasible for any other feasible allocation $x$,

$$
\sum_{s \in S} \frac{x_s - x^*_s}{x^*_s} \leq 0. \quad (4.3)
$$

A common theme underlying NUM protocols is that they are targeted managing the elastic traffics. This means that the traffic will present a non-zero utility as long as non-zero bandwidth is allocated to it.

### 4.2 Problem Analysis on Network Utility Optimisation for Hierarchical M2M Communications

As has pointed out by [JSKP10], the NUM has made great advances in dealing with resource allocation. However, serious limitations still exist as shown below,

i) NUM aims at managing elastics traffics, the utility of which can be modelled
ii) For the scenario where applications have different QoS requirements, a serious conflict exists between the utility maximisation and the utility fairness. In particular, the elastic traffics will always get allocated due to the rapid increase of utility with little resource allocated to them, while the inelastic traffics might receive less resource than its minimum requirement.

Thus the conventional criteria proportional fairness may not be suitable to address the resource allocation for networks with different QoS requirements \cite{JSKP10}. In \cite{CZ99}, the concept of *utility max-min fair* is suggested to support resource allocation for networks with different applications.

An allocation $x^* = [x^*_1, x^*_2, \cdots, x^*_I]^T$ is *utility max-min fair*, if it is feasible and for any other feasible allocation $x$, without decreasing the utility $U_{i'}(x^*_{i'})$ for some other user $i'$ with a lower utility $U_{i'}(x^*_{i'}) \leq U(x^*_i)$. Max-min fair allocation is recovered with

$$U_i(x_i) = x_i, \ i = 1, \ldots, I. \quad (4.4)$$

More recently, the *utility-fair* based resource allocation strategy is proposed for networks with both elastic and inelastic traffics \cite{WPL06}. The proposed strategy also guarantees the utility fairness among different traffics.

A bandwidth allocation $x^* = [x^*_1, x^*_2, \cdots, x^*_I]^T$ is *utility proportionally fair*, if it is feasible and for any other feasible allocation $x$,

$$\sum_{x \in I} \frac{x_i - x^*_i}{U_i(x^*_i)} \leq 0. \quad (4.5)$$
To achieve the utility-fair resource allocation, each source device $i$ with utility function $U_i(x_i)$ has a pseudo utility” $\mathcal{U}_i(x_i)$ associated with it. $\mathcal{U}_i(x_i)$ is defined as

$$\mathcal{U}_i(x) = \int_{c_i}^{C_i} \frac{1}{U_i(x)} \, dx, \quad c_i \leq s \leq C_i. \quad (4.6)$$

where $(c_i, C_i)$ is the capacity region of $x_i \in \mathcal{I}$.

Due to the non-negative and strictly increasing properties of $U_i$, it is clear that $\mathcal{U}_i$ is a strictly increasing concave function regardless the concavity of the original utility function $U_i$. Thus, with this “pseudo utility”, the original optimisation problem with both elastic and inelastic traffics is mapped into a convex NUM optimisation problem. And the solution of the “pseudo utility” problem achieves utility proportionally fair [JSKP10].

The “pseudo utility” is adopted in this thesis to solve the original joint scheduling and duty cycle control problem $\mathcal{P}_1$. Replacing the utility function of $\mathcal{P}_1$ with “pseudo utility” leads to a convex optimisation problem $\mathcal{P}_2$,

$$\mathcal{P}_2 : \quad \min_{t=0}^{T} \sum_{i \in \mathcal{I}_n} \mathcal{U}(f_i^t, r_i^t, \theta_i^t) \quad \text{(4.7a)}$$

s.t. 

$$\sum_{i \in \mathcal{I}_n} f_i^t \leq C_n, \quad (4.7b)$$

$$0 \leq f_i^t \leq C_i, \quad i \in \mathcal{I}_n \quad (4.7c)$$

$$r_i^t \leq \min(C_i, q_i^\text{max}), \quad i \in \mathcal{I}_n, j \in \mathcal{C}_i. \quad (4.7d)$$
Furthermore, $P_2$ can be decomposed into the following cluster head optimisation problem $P_3$ and gateway optimisation problem $P_4$:

\[ P_3 : \max \sum_{t=0}^{T} \sum_{i \in I_n} U_i(r_i^t) - p_{t,n} \]  
\[ s.t. \quad p_{t,n} \geq 0, \]  
\[ r_i^t \leq \min(C_{i,j}, q_{i}^{\text{max}}) j \in C_i, \]  

where $p_{t,n} = p_{t} f_{t}^i$ is the charged price by gateway $n$ for cluster head $i$ with the forwarded flow $f_{t}^i$. The constraint (4.8c) is the link capacity constraint. And

\[ P_4 : \max \sum_{t=0}^{T} \sum_{i \in I_n} p_{t,n} U_n(f_{t}^i, p_{t}^i, \theta_{t}^i) \]  
\[ s.t. \quad \sum_{i \in I_n} f_{t}^i \leq C_{n,b} \]  
\[ 0 \leq f_{t}^i \leq C_{i,n}, \]

where $p_{t}^i$ is the bid price per unit/packet of cluster head $i$ for the transmission at time period $t$. The constraints (4.9b) and (4.9c) are link capacity constraints.

Based on the problem formulation in [KMT98], an example bid price $p_{t}^i$ updating algorithm proposed in [LL99] is,

\[ p_{t+1}^i = \left[ p_{t}^i + \gamma (\sum_{i \in I_n} f_{t}^i - C_{n,b}) \right]^+. \]  

This equation indicates that if the aggregated number of packets exceeds the maximum queue length $q_{i}^{\text{max}}$, the bid price will be increased; otherwise it will be
decreased.

After the transmission, cluster head \( i \) is charged \( p^{t}_{i,n} = p^{t}_{i}f^{t}_{i} \) by gateway \( n \) for the forwarded flow \( f^{t}_{i} \). Suppose the gateway \( n \) knows its revenue vector \( P_{n} = (p^{t}_{i,n}, i \in I_{n}) \) and the application indicator \( \theta^{t}_{i} \). To guarantee the gateway is able to gain some profit, in this thesis \( p_{i,n} \) is calculated as

\[
p^{t}_{i,n} = p^{t}_{i} + 1. \tag{4.11}
\]

Due to the structures of the decomposed \( P_{3} \) and \( P_{4} \), it is possible to design the cluster head utility function and gateway utility function separately, so that the cluster head and gateway can address their application requirements and device capabilities, separately.

As the utility function \( U_{i} \) of cluster head is not required by the gateway, and it only appears in the optimisation problem \( P_{3} \) faced by cluster head \( i \), that the joint scheduling and duty cycle control problem is solvable in a distributed way.

Compared with the centralised approach where the central controller needs to assign the scheduling for the whole network, the distributed approach described above has the following advantages:

i) Scalability. For distributed controls, devices make decisions based on their local information rather than global information covering the whole network. As a result, the control overheads, energy and memory of distributed controls can be much less than those of centralised controls, especially in large-scale network.

ii) Adaptability to network dynamics. In M2M communications, the changes
in network conditions at some individual devices would only affect local network conditions rather than the whole network. As a result, distributed controls can address the dynamic network changes locally without expensive communications across the whole network.

iii) Faster response. To save energy, M2M networks normally operate at a low duty cycle mode. Compared to centralised low duty cycle networks, distributed controls significantly reduce the response time by reducing the time for transmitting the local information from each device to the central controller and the time for disseminating information from the central controller to each device.

4.3 Control Framework Design

Based on the aforementioned analysis, the network utility optimisation problem $\mathcal{P}_1$ is solved by two distributed problems $\mathcal{P}_3$ and $\mathcal{P}_4$. A joint scheduling and duty cycle control framework is proposed with three controls, named as duty cycle control, gateway control and cluster head control. The detailed solution for each control will be presented in the following Chapter 5, Chapter 6 and Chapter 7, respectively.

The overall control process of the proposed framework for hierarchical M2M networks is shown in Fig. 4.2. The proposed framework incorporates with the current IEEE 802.15.4 standard.

**IEEE 802.15.4 beacon transmission**

For beacon-enabled IEEE 802.15.4, a beacon is transmitted from the parent
Figure 4.2: Transmission process of designed control framework.
device to its child devices at the beginning of each superframe. A device needs to
listen for the beacon from its parent device before transmitting data to the parent
device. When the beacon is found, the device synchronises to the superframe
structure based on the information within the beacon.

For the proposed control framework, the beacon from the parent device will
provide duty cycle, transmission schedule and the charged price for each child
device. More specifically, for each time period $t$, the beacon from the cluster head
$i$ to its child M2M devices $j \in C_i$ will contain the duty cycle parameters $SO(r_i^t)$
and $BO(r_i^t)$, transmission schedule $r_i^t$, and the charged price $p_{i,j}^t$ for each M2M
device $j$.

**Cluster head signalling transmission**

At the end of the packet transmission duration from cluster heads to their
parent gateway, each cluster head will send a signalling to its parent gateway.
The information enclosed is its current duty cycle, the QoS requirement of its
application and its bid price $p_i^t$ for the transmission in next time period.

**Cluster head control**

The cluster head control is operated at each cluster head to decide its local
transmission schedule, control the duty cycle and operate the bid price to the
gateway depending on the local traffics. The cluster head control is based on the
optimisation problem $P3$ with the aim of maximising the single cluster utility.

For cluster head $i$, the output of the cluster head control are the cluster duty
cycle parameters $SO(r_i^t)$, the transmission schedule $r_i^t$, and the bid price $p_i^t$ offered
to its parent gateway $n$. The information about bid price $p_i^t$ is enclosed in the
signalling to the gateway at the end of time period $t - 1$. 
Gateway control

The gateway control is implemented at each gateway to adapt the amount of traffic each child cluster head will forward to it. The gateway control is based on the optimisation problem $\mathcal{P}4$, and the aim is to maximise the aggregated network utility with the consideration of different application requirements of clusters.

After the gateway $n$ receives the signalling from all its child cluster head $i \in \mathcal{I}_n$ and the bid price $p^t_i$. The gateway control will conduct the transmission schedule $f^t_i$ for all cluster head $i \in \mathcal{I}_n$ and set the outgoing superframe parameters. The gateway will also calculate the charged price $p^t_{i,n}$ for each cluster head $i \in \mathcal{I}_n$ according to the amount of traffic cluster head $i$ has forwarded. The charged price is enclosed in the beacon of the gateway at the beginning of time period $t + 1$.

4.4 Simulation Platform Design

The main modules of the simulation platform in this thesis are summarised in this section.

4.4.1 Simulation Flow Design

The simulation process is composed of a number of simulation iterations, and each simulation iteration consists of $T$ time periods. In each time period, the joint scheduling and duty cycle control for the hierarchical M2M network is processed. The simulation flow chart is given in Fig. 4.3.

Network Initialisation: Firstly, this module initialises the devices deployment, including the number and position for the base station, M2M gateways,
Figure 4.3: Flow chart of designed simulation platform
cluster heads and M2M devices. Then, it calculates the path loss, adds shadow fading and calculates the received energy level of each device. The output of this module is one network deployment with all successful links.

**System Initialisation**: This module initialises the application related parameters $\theta_i$, $U_i$, and the simulated MAC layer parameters. It also initialises the buffer capacity of each M2M device, cluster head and M2M gateway.

**Traffic Generation**: This module generates traffic of each cluster head and M2M device. It also initialised the packet transmission table, which helps to keep a record of the time periods at which the packets have been generated, transmitted, received or dropped.

**Gateway Control**: This module assigns the utility functions to clusters according to its application class. Then, it conduces the transmission schedule between cluster heads and the parent M2M gateway. The aim of the scheduling is to maximise the network throughput while satisfying different QoS requirements among clusters.

**Cluster Head Control**: This module conduces the transmission schedule within each cluster. Then, the duty cycle for IEEE 802.15.4 is adapted based on the cluster head transmission scheduling results. The aim of the cluster head scheduling is to achieve the joint optimisation of energy efficiency and delay.

### 4.4.2 Performance Metrics

During the simulation, each packet has a packet transmission table. The table keeps all the information about this packet, including the packet generation time, the packet forwarded time to cluster head, gateway and base station, and the packet
drop time. With this table, given the device energy consumption parameters, the performance metrics including energy efficiency, end-to-end delay, packet drop ratio and throughput can be obtained.

The following parts give the definitions and calculation methods of the performance metrics for the modelled hierarchical M2M networks.

4.4.2.1 Energy Efficiency

For energy-efficient communications, it is desirable to maximise the amount of data been sent with a given amount of energy. Thus, energy efficiency is defined as the ratio of transmission rate to energy consumption. The unit of energy efficiency is bits per Joule, which has been frequently used in literature for energy-efficient communications [MPSM05] [GM00] [FM04].

The duty cycle determines transmission energy consumption as it defines the medium access and active/sleep operation of devices. The overall transmission energy consumption contains:

\[ \text{transmission energy} = \text{sending energy + receiving energy} \]
\[ + \text{idle listening energy + sleeping energy}. \]

Thus, energy efficiency of duty cycle based MAC protocols is calculated as:

\[ \text{energy efficiency} = \frac{\text{total amount of data been transmitted}}{\text{transmission energy consumption}}. \]
4.4.2.2 End-to-end Delay

Delay is a very important QoS measurement since it influences the performance and stability of some applications, such as industrial control system. In this thesis, the end-to-end delay is measured as an average over a certain period of time.

The end-to-end delay is defined as the time consumed for a packet from the arrival at the source to the reception at the destination. The one-hop delay at MAC layer is defined as:

\[
\text{one-hop delay} = \text{processing delay} + \text{queuing delay} + \text{channel access delay} + \text{transmission delay} + \text{propagation delay} + \text{reception delay}.
\]

The end-to-end delay, which is the accumulation of the one-hop delay from sender to receiver, is expressed as:

\[
\text{end-to-end delay} = \sum_{h=1}^{H} \text{one-hop delay},
\]

where \( h = 1, \ldots, H \) is the number of hops of a selected path.

4.4.2.3 Network Throughput

Network throughput is measured as the average number of packets successfully transmitted over the whole network per unit time. For the modelled cluster-based hierarchical M2M network, the network throughput is the sum of the throughput
of all clusters.

\[ \text{per cluster throughput} = \text{transmitted data to the cluster head per unit time}, \]

\[ \text{network throughput} = \sum_{i=1}^{I} \text{throughput of cluster } i. \]

where \( I \) is the number of clusters of the gateway.

### 4.4.2.4 Packet Drop Ratio

Due to the limited buffer size, the packets are dropped when the queue length exceeds the buffer limitation of the device. For each device the packet drop ratio is defined as the ratio between the number of packets been dropped and the total number of packets been generated by the device. The averaged packet drop ratio in the network is given as,

\[ \text{device packet drop ratio} = \frac{\text{data dropped due to buffer limitation}}{\text{data generated by the device}}, \quad (4.13) \]

\[ \text{averaged packet drop ratio} = \frac{1}{M} \sum_{j=1}^{M} \text{packets drop ratio of device } j, \]

where \( M \) is the total number of devices in the network.

### 4.4.3 Platform Validation

In this part, the validation of the designed simulation platform is presented. The validated modules in this section are channel module and network initialisation module.
4.4.3.1 Channel Model

The validation of the channel model is done by comparing the results with that in [NIS11]. Fig. 4.4 shows the modelled dual-slop path loss model for indoor-indoor residential environment at $f_c = 5GH\text{z}$. As shown in Fig. 4.4, the trend and value in the simulated scenario align with the ones presented in [NIS11] with the same environmental settings.

![Figure 4.4: Dual-slope path loss model.](image)

4.4.3.2 Network Initialisation

The flow chart of the network initialisation module is shown in Fig. 4.5. The network initialisation module has four steps:

i) Randomly deployed M2M gateway $n$ within the simulated area;

ii) Randomly deployed cluster head $i \in \mathcal{I}_n$;
Chapter 4. Joint Scheduling and Duty Cycle Control Framework

Figure 4.5: Flow chart of network initialisation module
iii) Randomly deployed M2M device $j \in \mathcal{C}_i$ of each cluster head $i$;

iv) Check link quality for $\mathcal{L}_{i,n}$ and $\mathcal{L}_{i,j}$.

To validate the network initialisation module, one realisation of modelled hierarchical M2M network is carried out. The device energy consumption parameters are based on XBee®/RF data sheet [Int]. The XBee® 802.15.4 devices are used as M2M devices and XBee-PRO® 802.15.4 devices are used as cluster heads.

The setting of device parameters are given in Table. 4-A. The MAC layer parameters for the validated scenario are given in Table. 4-B

<table>
<thead>
<tr>
<th>Table 4-A: Device Parameters of Example Hierarchical M2M Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap Parameters:</td>
</tr>
<tr>
<td>XBee® 802.15.4 XBee-PRO® 802.15.4</td>
</tr>
<tr>
<td>Transmit Power</td>
</tr>
<tr>
<td>RF Line of Sight Range</td>
</tr>
<tr>
<td>Receiver Sensitivity (1%PER)</td>
</tr>
<tr>
<td>RF Data Rate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4-B: IEEE 802.15.4 MAC Layer Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>frequency</td>
</tr>
<tr>
<td>CCA size</td>
</tr>
<tr>
<td>unit backoff period</td>
</tr>
<tr>
<td>data rate</td>
</tr>
<tr>
<td>packet size</td>
</tr>
<tr>
<td>ACK packet size</td>
</tr>
</tbody>
</table>
Fig. 4.6 shows the deployed network, where the M2M gateway is represented by black triangle, cluster head is represented by square and M2M devices are represented by circle. The cross dots represent the failure devices (cluster head and M2M devices). The different colours represent different clusters of the gateway.

The device sensitivity requirement, which gives the minimum receive power, determines whether the link is successful or not.

Fig. 4.7 shows the averaged received power of cluster heads and M2M devices in the deployed M2M network. Comparing Fig. 4.7 and Table. 4-A, it can be seen that the successful transmission range of the simulated network over the formulated channel aligns with that of the presented device sensitivity requirements and RF transmission range according to the device data sheet [Int]. This also validates the accurate of the built channel module.
Chapter 4. Joint Scheduling and Duty Cycle Control Framework

4.5 Summary

In this chapter, the joint scheduling and duty cycle control optimisation problem for the modelled hierarchical M2M networks with multi-class applications is formulated as $\mathcal{P}_1$. However, the utility functions for inelastic applications (hard-real time applications, delay adaptive applications and rate adaptive applications) are not concave functions. This makes the problem $\mathcal{P}_1$ hard to be solved by applying classic NUM directly.

Next, the $\mathcal{P}_1$ has been transferred into the convex optimisation problem $\mathcal{P}_2$ by replacing the utility function of $\mathcal{P}_1$ with “pseudo utility”. Then, the $\mathcal{P}_2$ has been decomposed into distributed optimisation problems $\mathcal{P}_3$ and $\mathcal{P}_4$, which are named as cluster head control and gateway control, respectively.

Then, a joint scheduling and duty cycle control framework for M2M commu-
cation networks is proposed. The outcome of the control framework is a distributed optimal solution for the modelled hierarchical M2M networks with three controls: i) the duty cycle control with the aim of joint optimisation on energy efficiency and delay, ii) the cluster head control with the aim of maximising the single cluster utility, and iii) the gateway control with the aim of maximising network utility while satisfying different QoS requirements among clusters.

In the end, the design of simulation platform for the formulated hierarchical M2M networks is provided. The channel model and network deployment module of the platform are validated.
Chapter 5

Duty Cycle Control with Joint Optimisation of Energy Efficiency and Delay

This chapter focuses on the duty cycle control of the proposed control framework in Chapter 4. The theoretical optimal duty cycle control for IEEE 802.15.4 with both Stop-and-Wait and Go-Back-N ARQ are derived by applying dynamic programming (DP). The aim of the duty cycle control is to minimise the joint cost of energy consumption and end-to-end delay. Simulation results and discussion are presented at the end of this chapter.

5.1 Duty Cycle Control for IEEE 802.15.4 with Stop-and-Wait ARQ

The background of duty cycle control for IEEE 802.15.4 has been presented in the Section 2.3. Recall the duty cycle is defined as the ratio of the active portion over each time period. The duty cycle of IEEE 802.15.4 is controlled by the parameter
pair \((SO, BO)\),

\[
Duty\ Cycle = SD/BI = 2^{SO-BO}.
\] (5.1)

The duty cycle control designed for IEEE 802.15.4 works in a distributed manner. Each cluster head and gateway decides its own incoming duty cycle based on its local traffic. The outgoing superframe duty cycle of the cluster head is controlled by its parent gateway and enclosed in the received beacon. In this thesis, same \(BO\) is set to all devices to simplify the synchronisation. Thus, the duty cycle control of each device is achieved by setting the incoming duty cycle control parameter \(SO(ri)\).

The device can only transmit packets when it is in active mode. Thus, there is an direct link between the length of \(SD\) and the total number of transmitted packets \(ri\). Due to the collision of CSMA/CA, the actual transmission throughput has a co-relation with the number of contending devices. The throughput limitation coefficient \(b\) is adopted for IEEE 802.15.4 CSMA/CA according to [PKC+05].

Thus, within each \(SD\), if the beacon transmission duration is denoted as \(D_{bcn}\), the total packet transmission duration \(PD\) is given as,

\[
PD = SD - D_{bcn} = b \times (ri \times Ps),
\] (5.2)

where \(Ps\) is the the successful single packet transmission time.

In each contention access period (CAP), IEEE 802.15.4 adopts CSMA/CA for packet transmission. Before the packet transmission, devices need to perform two clear channel accesses (CCAs). If Stop-and-Wait ARQ is applied, an acknowledgement (ACK) is required for each successful received packet. Thus, the successful
single packet transmission time $P_s$ is

$$P_s = [P_{CCA} + P_L + \delta + P_{ACK}], \quad (5.3)$$

where $P_{CCA}$ is the transmission time for two CCAs, $P_L$ is the transmission time for each packet, $\delta$ and $P_{ACK}$ are waiting and transmission time of the ACK packet, respectively. Thus,

$$r_i = \left\lfloor \frac{SD - D_{bcn}}{b \times P_s} \right\rfloor. \quad (5.4)$$

Based on (5.2) - (5.4), the relationship between the incoming $SO(r_t^i)$ of the cluster head $i$ and the amount of packets the device could receive from its child devices $r_t^i$ at time period $t$ is given as,

$$SO(r_t^i) = \left\lceil \log_2\left(\frac{r_t^i \times P_s}{b} + D_{bcn}\right) \right\rceil. \quad (5.5)$$

### 5.2 Cluster Head Control with Stop-and-Wait ARQ

In this part, the cluster head control with the aim of joint-optimisation of energy efficiency and delay is formulated and solved by applying DP.

The cluster head control is formulated based on the classic inventory control problem, due to the good match between these two problems. The inventory control problem is a classical DP problem with the aim of minimising the overall cost by properly deciding the inventory level at each time period while meeting the
customer demand. This is similar to the cluster head control problem, which is to control the duty cycle with the aim of minimising energy consumption while reducing delay. The trade-off between energy consumption and delay of the duty cycle control lies: the reduced number of transmitted packets will reduce the transmission energy consumption, however this is at the cost of increasing the end-to-end delay and idle listening energy consumption.

Taking inventory control problem as the fundamental approach, the following four costs are defined for each cluster head $i$,

- $E_t(f^i_t)$ is the transmitting energy cost;
- $E_r(r^i_t)$ is the receiving energy cost;
- $E_l(r^i_t)$ is the idle listening energy cost;
- $D(r^i_t)$ is the end-to-end delay cost.

To ensure the costs of energy consumption and end-to-end delay are additive, the above costs are all defined with the unit “number of packets”. The energy consumption of ACK packet transmission exists only when the cluster head receives packets. With the Stop-and-Wait ARQ scheme, an ACK packet is required by each packet successfully received by the receiver. The definitions of the costs are given
as:

\[ E_r(r_t^i) = \begin{cases} c_r \cdot \frac{r_t^i}{q_{i_{max}}} & \text{if } r_t^i > 0, \\ 0 & \text{if } r_t^i = 0. \end{cases} \]  

(5.6)

\[ E_f(f_t^i) = c_f \cdot \frac{f_t^i}{q_{i_{max}}}, \]  

(5.7)

\[ E_{l}(r_t^i) = c_l \cdot \frac{[f_t^i - g_t^i - q_t^i - r_t^i]_{+}}{q_{i_{max}}}, \]  

(5.8)

\[ D(r_t^i) = c_d \cdot \frac{[q_t^i + r_t^i + g_t^i - f_t^i]_{+}}{q_{i_{max}}}, \]  

(5.9)

where \( c_f, c_r, c_l \) and \( c_d \) are the cost coefficients of transmitting, receiving, idle, listening and delay, respectively. Note that \( c_r < c_l \), as if \( c_r \) were greater than \( c_l \), it would never be optimal to receive packets at the last period and possibly in earlier periods.

Coefficients \( \alpha \) and \( \beta \) are further introduced to assign the weightings of energy efficiency and end-to-end delay for different application requirements. Then, the expected weighted-sum joint-cost function for cluster head \( i \) at time period \( t \) is

\[ J(r_t^i) = \mathbb{E}\left\{ \alpha \left( E_f(f_t^i) + E_r(r_t^i) + E_{l}(r_t^i) \right) + \beta D(r_t^i) \right\}. \]  

(5.10)

The objective is to find the optimal cluster head control policy \( \pi_i^* \) for each cluster head \( i \) over \( T \) time periods, which minimises the overall expected joint-cost. Hence, the joint optimisation problem is:

\[ \mathcal{P}_i : \min_{\pi_i \in \mathcal{D}} \mathbb{E}\left\{ \sum_{t=0}^{T-1} J(r_t^i) \right\} \]  

s.t. \( q_t^T = 0, \)

\[ r_t^i \leq r_{i_{max}}^i, \]
where $\mathcal{D}$ is valid policy sets of cluster head $i$, $r_{t_i}^{\text{max}}$ is the maximum number of packets cluster head $i$ could receive. $r_{t_i}$ subjected to the active duration and maximum queue length $q_{t_i}^{\text{max}}$ of cluster head $i$ at time period $t$.

5.3 Optimal Distributed Cluster Head Control with Stop-and-Wait ARQ

By applying the principle of DP, the problem $\mathcal{P}_i$ can be decomposed into a sequence of subproblems $\mathcal{S}(r_{t_i})$, where $0 \leq t \leq T$. The objective of each subproblem $\mathcal{S}(r_{t_i})$ is to minimise the sum of joint-cost functions from time period $t$ to $T$. Thus, the total cost of $\mathcal{P}_i$ is equal to that of $\mathcal{S}(r_{t_i}^0)$, which means the optimal solution of $\mathcal{S}(r_{t_i}^0)$ is the optimal solution of $\mathcal{P}_i$. Based on (5.10), the cost-to-go function $J_{t_i}$, which is the sum of joint-cost functions from time period $t$ to $T$ is given as

$$J_{t_i} = \min_{\pi_i \in \mathcal{D}} \mathbb{E}\left\{ \alpha \left( E_r(r_{t_i}) + E_t(f_{t_i}) \right) + H(r_{t_i}) + \mathbb{E}\{J_{t_i+1}\} \right\},$$ \hspace{1cm} (5.12)

where

$$H(r_{t_i}) = \mathbb{E}\left\{ \alpha E_i(r_{t_i}) + \beta D(r_{t_i}) \right\}$$ \hspace{1cm} (5.13)

$$= \mathbb{E}\{\alpha \cdot c_i \cdot \frac{[f_{t_i} - q_{t_i} - r_{t_i} - g_{t_i}]}{q_{t_i}^{\text{max}}} + \beta \cdot c_d \cdot \frac{[q_{t_i} + r_{t_i} + g_{t_i} - f_{t_i}]}{q_{t_i}^{\text{max}}} \}.$$ shows the tradeoff between the energy consumption cost due to idle listening and the end-to-end delay cost.

The objective of each subproblem $\mathcal{S}(r_{t_i})$ for cluster head $i$ is to minimise the
cost-to-go function $J_t^i$

\[
S(r_t^i) : \min_{\pi_i \in \mathcal{D}} \mathbb{E}\left\{ \sum_{k=0}^{t-1} J_t^i \right\}
\]
\[
\text{s.t. } q_t^i = 0, \quad r_t^i \leq r_{i}^{\text{max}}.
\]

To solve the subproblem $S(r_t^i)$, $m_t^i = q_t^i + r_t^i$ and $n_t^i = f_t^i - q_t^i$ are introduced for notation clarity. Then, combined with (5.10) and (5.11), $H(r_t^i)$ in (5.13) can be rewritten as

\[
H(m_t^i) = \mathbb{E}\left\{ \alpha c_i \times \max(q_{\text{max}}, [n_t^i - m_t^i]^+) + \beta c_d \times \frac{\max(q_{\text{max}}, [m_t^i - n_t^i]^+)}{q_{\text{max}}} \right\}.
\]

As the convexity preserved by taking expectation over $n_t^i$, with each fixed $n_t^i$, $H(m_t^i)$ is convex. Due to the convexity of $H(m_t^i)$, the cost-to-go function $J_t^i$ is rewritten as

\[
J_t^i(m_t^i) = \min_{\pi_i \in \mathcal{D}} \mathbb{E}\left\{ W(m_t^i) - \alpha c_r \times \frac{q_t^i}{q_{\text{max}}} \right\},
\]

where

\[
W(m_t^i) = \alpha c_r \times m_t^i + \alpha \times E_f(f_t^i) + H_i(m_t^i) + \mathbb{E}\{ J_t^i(m_t^{i+1}) \}.
\]

Then the objective of each subproblem $S(r_t^i)$ is to find the minimum value of (5.17).

Before giving the solution to each subproblem $S(r_t^i)$, the following Lemma 5.1 gives the sufficient condition for the convexity of functions $W(m_t^i)$.
Lemma 5.1. If $H(m_i^t)$ and $U(m_i^t)$ are convex functions, so is $W(m_i^t)$.

Proof. Based on the definition of convex function, $W(m_i^t)$ is convex if,

$$W\left(\frac{m_i^t + \bar{m}_i^t}{2}\right) \leq \frac{W(m_i^t) + W(\bar{m}_i^t)}{2}. \quad (5.18)$$

After eliminating, it is clear that (5.18) is satisfied if

$$H\left(\frac{m_i^{t_1} + m_i^{t_2}}{2}\right) + \mathbb{E}\{U\left(\frac{(m_i^{t_1} + 1) + (m_i^{t_2} + 1)}{2}\right)} \leq \frac{H(m_i^{t_1}) + H(m_i^{t_2})}{2} + \mathbb{E}\{U(m_i^{t_1} + 1) + U(m_i^{t_2} + 1)\}. \quad (5.19)$$

Since $H(m_i^t)$ and $U(m_i^t)$ are convex functions, we have

$$H\left(\frac{m_i^t + \bar{m}_i^t}{2}\right) \leq \frac{H(m_i^t) + H(\bar{m}_i^t)}{2}. \quad (5.20)$$

and

$$U\left(\frac{(m_i^t + 1) + (\bar{m}_i^t + 1)}{2}\right) \leq \frac{U(m_i^t + 1) + U(\bar{m}_i^t + 1)}{2}. \quad (5.21)$$

Adding (5.21) and (5.22), the inequality (5.19) is satisfied. Thus, $W(m_i^t)$ is also a convex function.

Based on Lemma 5.1, the following Theorem gives the optimal transmission policy of problem $P_i$.

Theorem 5.1. If $W(m_i^t)$ is convex, and the minimising scalars of $W(m_i^t)$ denoted as

$$m_i^{t*} = T_i = \arg \min_{m_i^t \in \mathbb{R}} W(m_i^t). \quad (5.22)$$

where $\mathbb{R}$ is the set of all valid values of $m_i^t$. 
Based on (5.22) and \( r^*_i = m_t - q^t_i \), the minimum cost-to-go function value is attained at \( r^*_i = T_i - q^t_i \) if \( q^t_i < T_i \), and at \( r^*_i = 0 \) otherwise. Thus, the optimal transition policy of \( \mathcal{P}_i \) is

\[
\begin{align*}
    r^*_i &= \begin{cases} 
        T_i - q^t_i & \text{if } q^t_i < T_i, \\
        0 & \text{if } q^t_i \geq T_i.
    \end{cases}
\end{align*}
\]

(5.23)

Proof. Based on the above Lemma 5.1, (5.12) to (5.18), the convexity of \( W(m^t_i) \) can be proved if function \( U(m^t_i) \) is a convex function, and \( \lim_{m^t_i \to \infty} W(m^t_i) = \infty \).

For \( t = T \), function \( U(m^t_i) \) is a zero function, so it is convex.

As \( c_r < c_l \), and the derivative of \( H(m^t_i) \) tends to \( -c_l \) as \( m^t_i \to -\infty \). Thus, \( H(m^t_i) \) is convex. Based on Lemma 1, given the convexity of \( U(m^t_i) \), \( W(m^t_i) \) is also convex. In addition, \( W(m^t_i) \) has a derivative that becomes negative as \( m^t_i \to -\infty \) and becomes positive as \( m^t_i \to \infty \), thus

\[
\lim_{|m^t_i| \to \infty} W(m^t_i) = \infty.
\]

Since \( W(m^{t-1}_i) \) is minimised by \( T_i \), the convexity of \( U(m^{t-1}_i) \) is obvious. Furthermore, we have

\[
\lim_{|m^{t-1}_i| \to \infty} U(m^{t-1}_i) = \infty.
\]

As shown above, the optimal policy at time \( T - 1 \) is given by

\[
\begin{align*}
    r^{T-1}_i &= \begin{cases} 
        T_i - q^{T-1}_i & \text{if } q^{T-1}_i < T_i, \\
        0 & \text{if } q^{T-1}_i \geq T_i.
    \end{cases}
\end{align*}
\]

(5.24)
Then the cost-to-go function at the time period $T-1$, is derived as

$$U(r_{i}^{T-1}) = \begin{cases} 
\alpha \cdot c_r(T_i - q_{t_{i}}^{T-1}) + \alpha \cdot E_f(f_{i}^{T-1}) + H(r_{i}^{T-1}) & \text{if } q_{t_{i}}^{T-1} < T_i, \\
\alpha \cdot E_f(f_{i}^{T-1}) + H(r_{i}^{T-1}) & \text{if } q_{t_{i}}^{T-1} \geq T_i.
\end{cases}$$

(5.25)

For $t = T - 1, \cdots, 0$, the above arguments can be repeated: if

i) $U(m_{t+1}^{T})$ is convex;

ii) $\lim_{|m_{t}^{T-1}| \to \infty} U(m_{t}^{T-1}) = \infty$; and

iii) $\lim_{|m_{t}^{T-1}| \to \infty} W(m_{t}^{T-1}) = \infty$.

Recursively, the cost-to-go functions can be derived as

$$U(r_{i}^{t}) = \begin{cases} 
\alpha \cdot c_r(T_i - q_{t_{i}}^{t}) + \alpha \cdot E_f(f_{i}^{t}) + H(r_{i}^{t}) + \mathbb{E}\{U(r_{i}^{t+1})\} & \text{if } q_{t_{i}}^{t} < T_i, \\
\alpha \cdot E_f(f_{i}^{t}) + H(r_{i}^{t}) + \mathbb{E}\{U(k+1)\} & \text{if } q_{t_{i}}^{t} \geq T_i.
\end{cases}$$

(5.26)

and

i) $U(m_{t}^{t})$ is convex;

ii) $\lim_{|m_{t-1}^{t-1}| \to \infty} U(m_{t-1}^{t-1}) = \infty$; and

iii) $\lim_{|m_{t-1}^{t-1}| \to \infty} W(m_{t-1}^{t-1}) = \infty$.

Thus, $W(m_{t}^{t})$ is convex over $t$ time periods, which means the minimal scalars $T_i$ exist. Thus, the proof of **Theorem 5.1** is completed.

According to the optimal transmission policy $\pi^*$, by substituting the optimal value of $r_{i}^{t}$ into (5.5), the optimal duty cycle control for IEEE 802.15.4 with Stop-
and-Wait ARQ is derived as

\[
SO(r^*_t) = \begin{cases} 
\lceil \log_2(\frac{r^*_t \times P_s}{b} + D_{bcn}) \rceil & \text{if } q_t(k) < T_t, \\
\lceil \log_2(D_{bcn}) \rceil & \text{if } q_t(k) \geq T_t.
\end{cases}
\] (5.27)

Thus, the optimal duty cycle control is a multi-period policy: for each time period \(t\), before the cluster head \(i\) makes the decision on the duty cycle, it will check the current queue length \(q_t^i\). If the queue length \(q_t^i\) is smaller than the threshold \(T_t^i\) of the optimal transmission policy, the \(SO\) of current time period is set based on the optimal number of packets it should receive \(r^*_t\); otherwise, \(SO\) is set to its minimum value \(\lceil \log_2(D_{bcn}) \rceil\).

### 5.4 Duty Cycle Control for IEEE 802.15.4 with Go-back-N ARQ

The energy consumption and end-to-end delay can be further reduced by applying Go-Back-N ARQ transmission scheme. Go-Back-N ARQ uses cumulative ACK scheme, where the receiver only sends one ACK signifying that the receiver has received all the transmitted packets in a certain time period. By doing so, the energy consumption of ACK transmission is reduced, and the end-to-end delay caused by waiting for ACK transmission is also reduced.

The formulation of the packet transmission duration in each \(BI\) is similar to that of the duty cycle control with stop-and-wait ARQ. However, with Go-back-N ARQ, only one ACK packet is required for each successful transmission (may include multiple packets). Thus, for each cluster head, the total packet transmission
duration is given as

\[ PD = SD - D_{bcn} = \sum_{j=1}^{J} [D_j] + [\delta + D_{ACK}] , \quad (5.28) \]

where \( D_j \) is the packet transmission duration of child device \( j \in C_i \), \( \delta \) and \( D_{ACK} \) are waiting time and transmission duration of the ACK packet, respectively. Then the number of packets that can be received by \( i \) is

\[ r^i_t = \sum_{j=1}^{J} b \cdot \lfloor D_j / D_p \rfloor , \quad (5.29) \]

where \( D_p \) is transmission duration per packet and \( b \) is the throughput limitation coefficient, which shows impact of the backoff and contention during CSMA/CA transmission. Then, the \( SO(r^i_t) \) is presented as

\[ SO(r^i_t) = \lceil \log_2 \left( \left\lfloor \frac{r^i_t D_p}{b} \right\rfloor + [\delta + D_{ACK}] + [D_{bcn}] \right) \rceil . \quad (5.30) \]

### 5.5 Cluster Head Control with Go-back-N ARQ

As the energy consumption of ACK packets transmission exists only when the cluster head receives packets. For Go-back-N ARQ, one ACK is required for each successful transmission between the sender and receiver. Thus a fixed ACK transmission energy cost \( A \) is introduced as part of the energy consumption. \( A \) has a linear correlation with the number of the child devices of the cluster head \( i \). Similar to the costs definition for the stop-and-wait ARQ scheme, for Go-back-N ARQ
scheme, the definition of each cost is given as:

\[
E_r(r^t_i) = \begin{cases} 
A + c_r \cdot \frac{r^t_i}{q_{i}^{\max}} & \text{if } r^t_i > 0, \\
0 & \text{if } r^t_i = 0.
\end{cases}
\]  
(5.31)

\[
E_f(f^t_i) = c_f \cdot \frac{f^t_i}{q_{i}^{\max}},
\]  
(5.32)

\[
E_l(r^t_i) = c_l \cdot \left[ \frac{f^t_i - g^t_i - q^t_i - r^t_i}{q_{i}^{\max}} \right]^+, 
\]  
(5.33)

\[
D(r^t_i) = c_d \cdot \left[ \frac{g^t_i + r^t_i + g^t_i - f^t_i}{q_{i}^{\max}} \right]^+,
\]  
(5.34)

where \( A = c_f \cdot M_i, \) \( M_i \) is the total number of child devices of cluster head \( i; c_f, c_r, c_l \) and \( c_d \) are the cost coefficients of transmitting, receiving, idle listening and delay, respectively. Note that \( c_r < c_l \), as if \( c_r \) were greater than \( c_l \), it would never be optimal to receive new packet at the last period and possibly in earlier periods.

The expected weighted-sum form of the joint-cost function for cluster head \( i \) at time period \( t \) is

\[
J(r^t_i) = E \left\{ \alpha \left( E_f(f^t_i) + E_r(r^t_i) + E_l(r^t_i) \right) + \beta D(r^t_i) \right\},
\]  
(5.35)

where \( \alpha \) and \( \beta \) are the weighting factors for energy efficiency and delay.

The objective is to find the optimal cluster head control policy \( \pi^*_i \) for each cluster head \( i \) over \( T \) time periods, which minimises the overall expected joint-cost. Hence,
the joint optimisation problem for the duty cycle control with Go-back-N ARQ is:

\[
P_i : \min_{\pi_i \in D} \mathbb{E} \left\{ \sum_{t=0}^{T-1} J(r_t^i) \right\} (5.36)
\]

s.t. \( q_t^i = 0 \), \( r_t^i \leq r_{i}^{max} \),

where \( D \) is valid policy sets of device \( i \), \( r_{i}^{max} \) is the maximum number of packets cluster head \( i \) could receive which is subjected to the active duration at time period \( t \) and maximum queue length \( q_{i}^{max} \).

### 5.6 Optimal Distributed Cluster Head Control with Go-back-N ARQ

Similar to the duty cycle control problem with Stop-and-Wait ARQ, the cost-to-go function \( J_t^i \), which is the added sum of joint-cost functions from time period \( t \) to \( T \) is given as

\[
J_t^i = \min_{\pi_i \in D} \mathbb{E} \left\{ \alpha \left( E_r(r_t^i) + E_t(f_t^i) \right) + H(r_t^i) + \mathbb{E} \{ J_{t+1}^i \} \right\}, \quad (5.37)
\]

where

\[
H(r_t^i) = \mathbb{E} \left\{ \alpha E_t(r_t^i) + \beta D(r_t^i) \right\} (5.38)
\]

\[
= \mathbb{E} \left\{ \alpha \cdot c_t \cdot \frac{f_t^i - q_t^i - r_t^i - g_t^i}{q_{i}^{max}} + \beta \cdot c_d \cdot \frac{q_t^i + r_t^i + g_t^i - f_t^i}{q_{i}^{max}} \right\}.
\]
shows the tradeoff between idle listening energy consumption cost and the end-to-end delay cost.

Different from the case with stop-and-wait ARQ, it is not trivial to find the optimal solution of the problem with Go-Back-N scheme, as function $j_t^i$ is not a convex function due to the $[.]^+$ operation of limited buffer size. However, it has been proved by Scarf that an optimal multi-period $(s,S)$ solution exists, if $j_t^i$ is $A-$convex function [Sca60].

**Definition 1.** The real-valued function $f$ is an $A-$convex function, if $A \geq 0$, for all $z \geq 0, b > 0$, $f$ satisfies the $A-$convexity property

$$A + f(z + y) \geq f(y) + z \left( \frac{f(y) - f(y - b)}{b} \right).$$ (5.39)

Next step is to find out the sufficient condition to the $A-$convexity of the cost-to-go function $J_t^i$. To reduce the number of notations in the equations, denote $m_t^i = q_t^i + r_t^i$ and $n_t^i = f_t^i - g_t^i$. If $\delta(0) = 0, \delta(r_t^i) = 1$ for $r_t^i > 0$, based on (5.33) - (5.36), we have

$$J_t^i = \min_{\pi_i \in D} \mathbb{E} \left\{ A\delta(r_t^i) + W(m_t^i) \right\} - \frac{\alpha_c \cdot g_t^i}{q_{max}^t},$$ (5.40)

where

$$W(m_t^i) = \alpha E_f(f_t^i) + \alpha E_r(r_t^i) + \alpha c \cdot \frac{[n_t^i - m_t^i]^+}{q_{max}^t} + \beta_c \cdot \frac{[m_t^i - n_t^i]^+}{q_{max}^t} + J([m_t^i - n_t^i]^+).$$ (5.41)

**Lemma 5.2.** According to (5.40), if $W(m_t^i)$ is an $A-$convex function, so is $J_t^i$.

**Proof.** Based on the definition of $A-$convex, it is necessary to show the following
equation holds for all $z \geq 0, b > 0$,

$$A + J^t_i(r^t_i + z) \geq J^t_i(r^t_i) + z \left( \frac{J^t_i(r^t_i) - J^t_i(r^t_i - b)}{b} \right). \quad (5.42)$$

As $A > 0$ and $W(m^t_i)$ is $A -$ convex, denote

$$T^t_i = m^t_i^* = \arg \min_{m^t_i \in \mathbb{R}} W(m^t_i). \quad (5.43)$$

Based on (5.40), the cost-to-go at time period $t$ is

$$J^t_i = \begin{cases} 
A + W(T^t_i) - \frac{\alpha c_r q^t_i}{q^t_i \max} & q^t_i < t^t_i, \\
W(m^t_i) - \frac{\alpha c_r q^t_i}{q^t_i \max} & q^t_i \geq t^t_i. 
\end{cases} \quad (5.44)$$

Three cases are discussed to distinguish the $A -$ convexity of $J^t_i(q^t_i)$:

**Case 1**: $q^t_i \geq t^t_i$. If $q^t_i - b \geq t^t_i$, then function $J^t_i$ is the sum of a $A -$ convex function and it is also a linear function. Hence, $J^t_i$ is $A -$ convex and (5.42) holds.

If $q^t_i - b < q^t_i$, (5.42) can be written as

$$A + W(m^t_i + z) \geq W(m^t_i) + z \left( \frac{W(m^t_i) - W(t^t_i)}{b} \right). \quad (5.45)$$

i) If $J^t_i(q^t_i) \geq J^t_i(t^t_i)$, then by $A -$ convexity of $W(m^t_i)$,

$$A + W(q^t_i + z) \geq W(m^t_i) + z \left( \frac{W(q^t_i) - W(t^t_i)}{q^t_i - t^t_i} \right) \geq W(m^t_i) + z \left( \frac{W(m^t_i) - W(t^t_i)}{b} \right). \quad (5.46)$$
ii) If \( J^i_t(q^i_t) < J^i_t(t^i_t) \), then

\[
A + W(q^i_t + z) \geq A + W(T^i_t) = W(t^i_t) > W(m^i_t)
\]
\[
\geq W(m^i_t) + z \left( \frac{W(m^i_t) - W(t^i_t)}{b} \right).
\]

(5.47)

So for this case, (5.45) and hence (5.42) hold.

**Case 2:** \( m^i_t \leq m^i_t + z \leq t^i_t \). In this region, the function \( J^i_t \) is linear hence (5.42) holds.

**Case 3:** \( m^i_t \leq t^i_t \leq m^i_t + z \). For this case, write (5.42) as \( A + J^i_t(q^i_t + z) \geq J^i_t(t^i_t) \) which holds by the definition of \( t^i_t \). Thus the \( A - convexity \) of \( J^i_t \) is proved given the \( A - convexity \) of \( W(m^i_t) \).

To show the property between \( J^i_t([m^i_t - n^i_t]^+] \) and \( W(m^i_t) \), rewrite (5.41) as

\[
W(m^i_t) = \alpha \left( E_f(j^i_t) + E_r(r^i_t) \right) + \frac{\beta c_d \cdot [m^i_t - n^i_t]^+}{q^{max}_i p_i} + R(m^i_t),
\]

(5.48)

where

\[
R(m^i_t) = \alpha c_i \cdot \frac{[n^i_t - m^i_t]^+}{q^{max}_i} + J^i_t([m^i_t - n^i_t]^+).
\]

(5.49)

The \( A - convexity \) of \( W(m^i_t) \) holds if the \( A - convexity \) of \( W(m^{i+1}_t) \) implies \( A - convexity \) of \( R(m^i_t) \).

**Lemma 5.3.** According to (5.48), if \( W(m^{i+1}_t) \) is an \( A - convex \) function, so is \( R(m^i_t) \).

**Proof.** Four cases are discussed to show the \( A - convexity \) of \( R(m^i_t) \):
Case 1: \(0 \leq m_{i+1}^{t} - b < m_{i+1}^{t} \leq m_{i+1}^{t+1} + z\), \(A - \text{convexity}\) of \(R(m_{i}^{t})\) follows that of \(W(m_{i}^{t+1})\).

Case 2: \(m_{i+1}^{t} - b < m_{i+1}^{t} \leq m_{i+1}^{t+1} + z \leq 0\): in this region, \(R(m_{i}^{t})\) is linear and hence \(A - \text{convex}\).

Case 3: \(m_{i+1}^{t} - b < m_{i+1}^{t} \leq 0 \leq m_{i+1}^{t+1} + z\): for simplicity, denote \(x = m_{i+1}^{t} + z\) in this region.

i) \(0 < t_{i+1}^{t} \leq x\):

\[
A + J_{i}^{t}(x) = A - x \cdot \frac{\alpha c_{l} \cdot q_{i}^{t+1}}{q_{i}^{\text{max}}} + J_{i}^{t}(T_{i}^{t+1})
\]

\[
= J_{i}^{t}(0) - x \cdot \frac{\alpha c_{l} \cdot q_{i}^{t+1}}{q_{i}^{\text{max}}}.
\]

ii) \(t_{i+1}^{t} \leq 0 \leq x \text{ and } 0 \leq x \leq t_{i+1}^{t+1}\):

\[
A + J_{i}^{t}(x) = 2A - x \cdot \frac{\alpha c_{r} \cdot q_{i}^{t+1}}{q_{i}^{\text{max}}} + J_{i}^{t}(T_{i}^{t+1})
\]

\[
\geq J_{i}^{t}(0) - x \cdot \frac{\alpha c_{l} \cdot q_{i}^{t+1}}{q_{i}^{\text{max}}}.
\]

Thus \(A + J_{i}^{t}(x) \geq J_{i}^{t}(0) - x \cdot \frac{\alpha c_{r} \cdot q_{i}^{t+1}}{q_{i}^{\text{max}}\cdot d_{i}}\) in this case. According to the definition of \(R(m_{i}^{t})\), the \(A - \text{convexity}\) of \(R(m_{i}^{t})\) holds.

Case 4: \(m_{i+1}^{t} - b < 0 < m_{i+1}^{t} \leq m_{i+1}^{t+1} + z\). Then, \(0 < m_{i+1}^{t} < b\).

i) If \(\frac{R(m_{i+1}^{t}) - R(0)}{m_{i+1}^{t+1}} \geq \frac{R(m_{i+1}^{t}) - R(m_{i+1}^{t} - b)}{b}\), thus

\[
A + R(m_{i+1}^{t} + z) \geq R(m_{i+1}^{t} + z) + \frac{R(m_{i+1}^{t}) - R(0)}{m_{i+1}^{t+1}}
\]

\[
\geq R(m_{i+1}^{t} + z) + \frac{R(m_{i+1}^{t}) - R(m_{i+1}^{t} - b)}{b},
\]
ii) If \( \frac{R(m_{i+1}^{t+1}) - R(0)}{m_{i+1}^{t+1}} < \frac{R(m_{i+1}^{t+1}) - R(m_{i+1}^{t+1} - b)}{b} \), then we have

\[
R(m_{i+1}^{t+1}) - R(0) < \frac{m_{i+1}^{t+1}}{b} \left( R(m_{i+1}^{t+1}) - R(m_{i+1}^{t+1} - b) \right)
= \frac{m_{i+1}^{t+1}}{b} \left( R(m_{i+1}^{t+1}) - R(0) + \frac{\alpha c_l \cdot q_{i+1}^{t+1}}{q_{i+1}^{\text{max}}} l_i (m_{i+1}^{t+1} - b) \right).
\]

Since \( b > m_{i+1}^{t+1} \),

\[
R(m_{i+1}^{t+1}) - R(0) < \frac{\alpha \cdot c_l \cdot q_{i+1}^{t+1}}{q_{i+1}^{\text{max}}} m_{i+1}^{t+1}.
\]

Then we have

\[
R(m_{i+1}^{t+1}) + z \frac{R(m_{i+1}^{t+1}) - R(m_{i+1}^{t+1} - b)}{b}
= R(m_{i+1}^{t+1}) - z \frac{\alpha c_l \cdot q_{i+1}^{t+1}}{q_{i+1}^{\text{max}}}
\leq R(0) - (m_{i+1}^{t+1} + z) \frac{\alpha c_l \cdot q_{i+1}^{t+1}}{q_{i+1}^{\text{max}}} \leq A + R(m_{i+1}^{t+1} + z).
\]

Hence, \( R(m_{i+1}^{t+1}) \) is \( A \) - convex for all cases. \( \square \)

**Theorem 5.2.** If function \( J_t^i \) is \( A \)-convex, the optimal duty cycle control for IEEE 802.15.4 with Go-back-N ARQ scheme is a multi-period policy, the duty cycle control parameter \( SO(r_{i}^t) \) is set according to its current queue length \( q_{i}^t \) and its threshold \( T_{i}^t \): when \( q_{i}^t \) is smaller than the threshold \( T_{i}^t \), the optimal \( SO(r_{i}^t)^* \) is set based on (5.30), otherwise, \( SO(r_{i}^t)^* \) equals to zero:

\[
SO(r_{i}^t)^* = \begin{cases} 
\left\lceil \log_2 \left( \left\lceil \frac{r_{i}^t D_p}{b} \right\rceil + \left\lceil \delta + D_{ACK} + D_{bcn} \right\rceil \right) \right\rceil & \text{if } q_{i}^t < T_{i}^t, \\
\left\lceil \log_2 \left( \left\lfloor \frac{D_{bcn}}{b} \right\rfloor \right) \right\rceil & \text{if } q_{i}^t \geq T_{i}^t.
\end{cases}
\]

(5.53)

where \( r_{i}^t \) is the optimal number of packets the device should received at each time
Chapter 5. Duty Cycle Control with Joint Optimisation of Energy Efficiency and Delay

5.7 Optimal Dynamic Programming based Cluster Head Control

The standard DP algorithm recursively computes the optimal solution in a backward fashion. DP is applicable only when the principle of optimality holds for the problem. And DP algorithm solves all the tail sub-problems by using the Bellman equation. However, the optimal control policy cannot be tractably computed in general. This is because the size of a state space typically grows exponentially in the number of state variables, which is known as the curse of dimensionality. This phenomenon makes DP intractable for the problems with practical scale.

The general DP algorithm is presented as follow. The cost-to-go function $J$ is the sum of the additive utility $U(s_t, a_t)$ over time, where $J$ is defined as

$$J = \mathbb{E}\left\{ U(s_t) + \sum_{t=0}^{T-1} U(s_t, a_t) \right\}. \quad (5.54)$$

The policy $\pi(a_t|s_t)$ denotes the conditional probability of choosing the control $a_t$ at the given state $s_t$. The MDP problem aims at of finding a policy which maximise the expected utility,

$$\pi^* = \arg\min_{\pi} \mathbb{E}\left\{ U(s_t) + \sum_{t=0}^{T-1} U(s_t, a_t) \right\}, \quad (5.55)$$
where the distribution over trajectories under policy $\pi$ is,

$$q_\pi(\bar{x}, \bar{u}|s_0) = \pi(u_0|s_0) \prod_{t=1}^{T} \pi(a_t|s_t) P(s_{t+1}|s_t, a_t). \quad (5.56)$$

The “tail subproblem” at $s_t$ at time $t$ aims to maximise the cost-to-go function $J$ from time $t$ to time $T$

$$J = E\left\{ U(s_T) + \sum_{t}^{T-1} U(s_t, \mu_t(s_t)) \right\}, \quad (5.57)$$

where $\mu_t, \mu_{t+1}, \ldots, \mu_T$ is the tail policy for the system which states with $s_t$.

Following the same logic, the optimal DP based duty cycle control is capable to solve the cluster head control with both Stop-and-wait and Go-Back-N. This standard DP algorithm is also called value iteration (VI) algorithm in some contexts. The specific duty cycle control applies DP algorithm is given in Algorithm 1.

### 5.8 Duty Cycle Control Simulation Results

In Section 5.1 and Section 5.3 the duty cycle control for IEEE 802.15.4 with Stop-and-Wait and Go-Back-N ARQ scheme are derived, respectively. In this section, the simulation results of the duty cycle control are presented.
Algorithm 1 Dynamic Programming based Duty Cycle Control

**Require:** cluster head $i \in I_n$, total control time period $T$ and current time period $t$

1: for each $i \in I_n$ do
2:   Initialise the number of receive packets $r^0_i$ arbitrarily and cost-to-go function $J^0_i = 0$
3: for $t = T - 1 \rightarrow 0$ do
4:   for each $r^t_i \leq q^\text{max}_i$ do
5:     Calculate the cost function $J(r^t_i)$
6:   for each $r^t_i \leq q^\text{max}_i$ do
7:     the cost-to-go function $J^t_i \leftarrow \sum_{k=k}^t J(r^t_i) + J^{t+1}_i$
8: end for
9: end for
10: for $t = 0 \rightarrow T$ do
11:   **Step 1: Compute optimal transmission policy**
12:   calculate the optimal number of receive packets $r^*_i$ for cluster head $i$:
13:   $r^*_i \leftarrow \arg \min J^t_i$
14:   **Step 2: Assign the duty cycle**
15:   The duty cycle control with Stop-and-Wait should be:
16:   $SO(r^*_i) \leftarrow \left\lceil \log_2\left(\frac{r^*_i \times P_s}{b} + D_{bcn}\right) \right\rceil$ or
17:   The duty cycle control with Go-Back-N should be:
18:   $SO(r^*_i) = \left\lceil \log_2\left(\frac{r^*_i D_p}{b} + [\delta + D_{ACK} + D_{bcn}]\right) \right\rceil$.
19: end for
20: end for

5.8.1 Duty Cycle Control for IEEE 802.15.4 with Stop-and-Wait and Go-back-N ARQ

Fig. 5.1 shows the co-relation between the duty cycle parameter $SO$ and the number of received packets. The result of duty cycle control for Stop-and-Wait is shown in red line and that of the Go-back-N scheme is shown in blue line. From the Fig. 5.1 it can be seen that for same amount of transmitted packets the $SO$ of duty cycle control with Go-back-N ARQ scheme is smaller than that of the Stop-and-Wait ARQ scheme. As the duty cycle is used to represent energy efficiency, with same $BO$ setting, the smaller the $SO$ is, the higher the energy efficiency can be achieved. Thus, the duty cycle control with Go-Back-N ARQ scheme achieves
higher energy efficiency, especially when the number of transmitted packets is small.

![Duty cycle setting with different ARQ schemes](image)

Figure 5.1: Duty cycle setting with different ARQ schemes.

The difference between the compared two schemes becomes less with the increase of the transmitted packets. This is due to the fact that the difference between the compared two schemes lies in the ACK packets transmission. With the increase of transmitted packets the portion of required ACK packets transmission become ignorable, thus the two controls behave similarly when the number of transmitted packets is large.

### 5.8.2 Optimal Duty Cycle Control for IEEE 802.15.4 based Hierarchical M2M Networks

In this subsection, the performance of the optimal DP transmission policy is evaluated in Matlab.
Table 5-A: Duty Cycle Control Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency</td>
<td>2.4 GHz</td>
<td>traffic model</td>
<td>Poisson</td>
</tr>
<tr>
<td>transmit power</td>
<td>57.24 mw</td>
<td>packet size</td>
<td>50 bytes</td>
</tr>
<tr>
<td>receive power</td>
<td>62 mw</td>
<td>data rate</td>
<td>250 kbps</td>
</tr>
<tr>
<td>idle listen</td>
<td>1.4 mw</td>
<td>time period length</td>
<td>3.93 s</td>
</tr>
<tr>
<td>duty cycle</td>
<td>10%</td>
<td>T</td>
<td>2, 5 &amp; 10</td>
</tr>
</tbody>
</table>

Focusing on the inherent properties of the proposed policy, a two hop cluster-tree network is considered in the simulation. The parameters on energy consumption are set based on the CC2420 data sheet [Ins] and MAC layer parameters are based on the IEEE 802.15.4 standard [Soc11]. Each simulation is repeated for 100 times. For cluster head $i$, the number of its child M2M devices $C_i$ equals to 2.

The devices generate packets following Poisson distribution. The average packet arrival rate is varied in order to study the impact of traffic load. The service rate of cluster head $i$ also follows Poisson distribution, and the mean value of packet service rate is 5 packets per active period. The maximum queue length is set to be 50 packets. The length of each time period is 3.93s ($15.36 \times 2^8$ ms) according to IEEE 802.15.4 standard. In each time period, the length of the active period is decided by duty cycle. $\alpha$ and $\beta$ are set to be 0.4 and 0.6, respectively. The simulation parameters are shown in the Table 5-A.

First, DP algorithm is used to find the minimum threshold $T_i$ of the optimal transmission policy. The simulation result is given in Fig. 5.2.

The threshold $T_i$ of DP algorithm is the maximum number of packets the cluster head $i$ can receive at each active period. Due to the convexity of the cost functions, the optimal $T_i^*$ is obtained when the joint-cost value remains the same while the
simulated threshold increased.

In Fig. 5.2, the joint-cost value is given by the optimal solution found by DP algorithm. The joint-cost value remains similar when the simulated threshold $T_i$ is bigger than 6, thus the optimal threshold $T_i^*$ of the simulated scenario is 6 packets per active period. In addition, it can be seen that the converge point of the joint-cost value is same with different settings of the total control time period $T$.

![Figure 5.2: Optimal transmission threshold by applying DP.](image)

Then, comparison between the optimal transmission policy and a random policy is given.

- **random policy**, the device receives a random number of packets in each time period based on Poisson distribution the mean value of which equals to 3;
- **DP optimal transmission policy**, the threshold $T_i$ of the optimal transmission policy is set to be 6 packets based on the result of DP algorithm as shown in Fig. 5.2.

For the compared random policy, the duty cycle is set to be fixed 10% with the active period of 0.1s, which is long enough to transmit all generated packets under the simulation setting. The evaluated performance metrics are average energy consumption per packet and average end-to-end delay under various traffic loads:

*Average energy consumption per packet:* is calculated as the total energy consumption in $T$ active periods over the total number of transmitted packets.

*Average end-to-end delay:* is the total delay caused by buffered packets over the total number of successfully transmitted packets.

![Average energy consumption of DP based duty cycle control.](image)

Figure 5.3: Average energy consumption of DP based duty cycle control.
Chapter 5. Duty Cycle Control with Joint Optimisation of Energy Efficiency and Delay

Fig. 5.3 shows the result of the average energy consumption per packet under different traffic arrival rates. The energy consumption per packet decreases as the network traffic increases for both policies. This is because higher packet arrival rates provide more available packets to be transmitted, when the active period is long enough to forward all these packets. With fixed duty cycle, the increased number of transmitted packets reduce the average consumption of each packet.

![Figure 5.3: Average energy consumption per packet under different traffic arrival rates.](image)

Fig. 5.4 shows the result of average end-to-end delay under different traffic arrival rates. The average end-to-end delay increases as the increase of packet arrival rates. Due to the increased traffic loads and the limitation of the service rate, some packets are buffered in the queue. The buffered packets have to wait to next time period for transmission, which increase the average per packet end-to-end delay.

![Figure 5.4: Average end-to-end delay of DP based duty cycle control.](image)
Taking traffic loads, network topology and application requirements into account, the optimal transmission policy achieved both lower energy consumption and end-to-end delay compared to the random policy, by adaptively scheduling the number of packets the device received at each active time period.

According to the simulation results, the optimal transmission policy is able to achieve about 50% saving in both energy consumption and end-to-end delay compared to that of the IEEE 802.15.4 random policy when the packet arrival rate is above 15 packets per active period.

5.9 Summary

In this chapter, the duty cycle control optimisation problem for IEEE 802.15.4 is formulated. The aim of the duty cycle control is to minimise the expected joint-cost of energy consumption and end-to-end delay. The duty cycle control for IEEE 802.15.4 with both Stop-and-Wait and Go-back-N ARQ scheme are calculated. Then the optimal duty cycle control is derived by applying DP algorithm. Furthermore, the proposed DP based duty cycle controls for both Stop-and-Wait and Go-back-N schemes require no initial control setting and is well compatible with the current IEEE 802.15.4 standard.

Simulation results show that the optimal transmission policy effectively reduced both energy consumption and end-to-end delay under various network traffics. Especially, when the traffic load is high, the reduction of energy consumption and end-to-end delay are more than 50% compared to the standard IEEE 802.15.4 standard.
Chapter 6

Cluster Head Control with Cluster Utility Maximisation

This chapter focuses on the cluster head control of the proposed control framework. The proposed cluster head controls aim to solve the formulated optimisation problem $\mathcal{P}3$ in Chapter 4. More specifically, three practical cluster head controls are proposed to fit into different practical hierarchical M2M scenarios. The cluster head controls aim at maximising the single cluster utility, which contains both empirical network performance component and economic component. For the empirical network performance component, the cluster throughput and the join-cost of energy efficiency and end-to-end delay have been taken into consideration.

6.1 Cluster Head Utility Design

Aforementioned in Chapter 4, the designed network optimisation problem $\mathcal{P}1$ is decomposed into the distributed cluster head control $\mathcal{P}3$ and the gateway control $\mathcal{P}4$. The aim of the cluster head control is to maximise the single cluster utility.

The first step of performing theoretical optimisation is to design a utility function which is able to reflect the identified network performance aimed to be im-
proved \cite{WPL06}. The defined cluster utility function is a mathematical measurement of the QoS performance based on the provided network resource, such as bandwidth, transmission rate and resource blocks.

In this thesis, the utility of the cluster head $i$ is designed with both empirical network performance component $V_{ti}$ and economic component $R_{ti}$. The utility function of cluster head $i$ is defined with a quasi-linear form:

$$U_i(r_{ti}) = \log(1 + V_{ti}) + R_{ti}, \quad (6.1)$$

where the empirical component $V_{ti}$ combines both empirical throughput and empirical cost of transmission, and the economical component $R_{ti}$ is the economic profit of the transmission. The economic profit is the return on revenue of packets transmission, which includes charges of receiving packets and costs of forwarding packets. The quasi-linear form of value function is chosen, since any equilibrium solutions to utility maximisation problems are independent of the initial economical setting of each cluster head.

In order to obtain the optimal theoretical solution, the formulation of the empirical component $V_{ti}$, which named as value function needs to be monotonic and concave. The value function reflects the node’s empirical network performance for receiving or forwarding certain amount of packets. For each cluster head $i \in \mathcal{I}_n$, the value function consists of two terms, the first term represents the throughput of receiving packets and the second term represents the joint-cost of energy and end-to-end delay.

$$V_{ti} = \epsilon_1 \log(1 + \frac{f_{ti}}{q_{i max}}) - \epsilon_2 J(r_{ti}) \log(1 + J(r_{ti})), \quad (6.2)$$
where $J(r^t_i)$ is the joint-cost of energy and end-to-end delay defined in the last chapter.

As designed in Chapter 4, the cluster heads charge its child nodes for forwarding packets to gain some economic benefits. If $R^t_i, \forall i \in \mathcal{I}$ is expressed as the product of the corresponding price and number of transmitted packets,

$$R^t_i = p^t_i r^t_i - p^t_i f^t_i, \forall l \in \mathcal{L}_i,$$

(6.3)

where the per packet service price $p^t_i = p_i + 1$ of cluster head $i$ on link $l_{i,j}$.

Cluster heads are assumed to be non-communicative as they are unable to observe the situation of the other cluster heads in many practical applications. Thus the utility function of each cluster head does not consider the impacts of 1) the set of the cluster heads that compete with the cluster head $i$; 2) the performance measurements on cluster head from its child device; 3) the utility function of its child devices; and 4) the price of other cluster heads. The cluster heads will make the control decision with consideration of the cluster traffic condition, its buffer capacity and the bid price it offered to the gateway.

### 6.2 Policy Iteration (PI) Algorithm based Cluster Head Control

According to the previous chapter, the optimal cluster head control can be found by running DP algorithm. To reduce the computational complexity of the optimal DP algorithm, the policy iteration (PI) algorithm based cluster head control is proposed in this chapter.
Chapter 6. Cluster Head Control with Cluster Utility Maximisation

The cluster head control has two steps, the first step is to compute the transmission policy; The second step is to assign the duty cycle based on the computed

Algorithm 2 Policy Iteration Algorithm based Cluster Head Control

Require: number of cluster head $i \in \mathcal{I}_n$, total control time periods $T$ and current control time period $t$

1: Initialise control policy $\pi$ to be evaluated arbitrarily, and
2: policy-stable $\leftarrow$ false
3: for each $i \in \mathcal{I}_n$ do
4:   Step 1: Compute transmission policy
5:   a) Policy evaluation:
6:      for $t = 0 \rightarrow T$ do
7:         Find number of receive packets $r_t^i$ according to the given control policy $\pi$
8:         $r_t^i \leftarrow \pi$
9:      Calculate the aggregated utility $U_\pi(r_t^i)$ with utility till time period $t$
10:     and the estimated utility $U_i(r_t^i+1)$ afterwards:
11:     $U_\pi(r_t^i) \leftarrow \sum_{k=0}^t U(r_t^i) + U_i(r_t^i+1)$
12:   end for
13:   b) Policy improvement:
14:      for each $\pi \in \mathcal{D}$ do
15:         $r_t^i \leftarrow \arg \min U_i(r_t^i)$
16:      Check whether the current control $r_t^i$ is aligned with the policy $\pi$
17:      if $r_t^i = u_t^i \in \pi$ then
18:         policy-stable $\leftarrow$ true
19:      end if
20:      if policy-stable = true then
21:         stop
22:      else
23:         go to line 16.
24:      end if
25:   end for
26:   Step 2: Assign the duty cycle
27:   for $t = 0 \rightarrow T$ do
28:      Calculate the number of received packets based on the policy $\pi$
29:      $r_t^{i+1} \leftarrow u_t^i \in \pi$
30:      Assign the Stop-and-Wait duty cycle control parameter $SO(r_t^i)$:
31:      $SO(r_t^i) = \lceil \log_2 \left( \frac{r_t^i \times P_s}{b} + D_{bcn} \right) \rceil$ or
32:      Assign the Go-Back-N duty cycle control parameter $SO(r_t^i)$:
33:      $SO(r_t^i) = \lceil \log_2 \left( \left\lfloor \frac{r_t^i D_p}{b} \right\rfloor + \lceil \delta + D_{ACK} \rceil + \lceil D_{bcn} \rceil \right) \rceil$.
34:   end for
35: end for
optimal transmission policy. The PI algorithm is applied in the first step.

i) Policy evaluation, a policy $\pi$ is evaluated by computing the aggregated utility function $U_\pi(r_t^i)$.

ii) Policy improvement, the PI algorithm looks for a policy $\pi'$ that has higher utility value than the previously evaluated policy $\pi$. The heuristic based policies are applied accordingly in the policy improvement step to maximise the function $U(r_t^i)$. When the same policy is found in two consecutive iterations, the algorithm has converged.

The exact embodiment of the proposed PI based cluster head control is shown in Algorithm 2.

Due to the limitation of the buffer size, the cluster head control has finite action and state spaces as well as bounded and stationary utility function. Under these conditions PI algorithm is proven to converge to the optimal policy. What’s more, it has been shown that PI converges in fewer iterations than VI in practice [Ber05].

### 6.3 Rollout Algorithm (RA) based Cluster Head Control

To further reduce the computational complexity of Algorithm 2, a low complexity suboptimal solution, named rollout algorithm (RA) based cluster head control is proposed. The utility lower bound, which is the theoretical worst case of RA based cluster head control is provided.
6.3.1 Low Complexity Suboptimal Solution

An effective way to reduce the computation required by DP is to truncate the time horizon and use a deviation based on lookahead of a small number of states. The most straightforward way is to use a one-step lookahead policy whereby at state $s_t$, the controller chooses the control $\bar{\mu}_t(s_t)$, which gives the minimum of

$$\min_{u_t \in U_t(s_t)} \mathbb{E}\left\{ g_t(s_t, u_t, w_t) + \tilde{J}_{t+1}(f_t(s_t, u_t, w_t)) \right\}. \quad (6.4)$$

The approximating function $\tilde{J}_{t+1}$ is the cost-to-go $\tilde{J}_{t+1}^{\pi}$ of some known heuristic or suboptimal policy $\pi = \{\mu_1, \mu_2, \ldots, \mu_T\}$, called base policy. The policy thus obtained is called the rollout policy based on $\pi$. Thus, the rollout policy is a one-step lookahead policy, with the optimal cost-to-go approximated by the cost-to-go of the base policy.

The process of starting from some suboptimal policy and generating another policy using the one-step-look ahead process, which is known as policy improvement, and is the basis of a primary method for solving the DP problems in policy iteration method. The rollout algorithm can be viewed as a single policy iteration. Thus, the reduction of computational complexity of RA based policy is achieved by reducing the iteration times of the optimal PI policy.

Rollout algorithms have demonstrated excellent performance on a variety of dynamic optimisation problems. Interpreted as an approximate DP algorithm, rollout algorithm estimates the utility at each time period by estimating future utility while following a heuristic control, referred to as the base policy.
6.3.2 Rollout Algorithm based Cluster Head Control

To ensure the stable of queue length, the devices should receive no more packets than it could transmit. Thus, instead of searching the optimal solution by running DP, the most straight forward approach is to set the $T_i$ equal to the mean value of $f_i$ for each cluster head $i$. Based on the Theorem 5.1 and Theorem 5.2 the heuristic base cluster head control with Stop-and-Wait ARQ is designed as

$$SO(r_i^t) = \begin{cases} 
\lceil \log_2(\frac{r_i^t P_s}{b} + D_{bcn}) \rceil & \text{if } q_i^t < f_i^t, \\
\lceil \log_2(D_{bcn}) \rceil & \text{if } q_i^t \geq f_i^t.
\end{cases}$$

(6.5)

And the heuristic base cluster head control with Go-Back-N ARQ is designed as

$$SO(r_i^t) = \begin{cases} 
\lceil \log_2(\left\lceil \frac{r_i^t P_s}{b} \right\rceil + [\delta + D_{ACK}] + [D_{bcn}]) \rceil & \text{if } q_i^t < f_i^t, \\
\lceil \log_2([D_{bcn}]) \rceil & \text{if } q_i^t \geq f_i^t.
\end{cases}$$

(6.6)

The proposed RA based cluster head control is the one that attains the maximum of the long-term utility function

$$U_i(r_i^t) = \max_{\pi_i \in S} \left[ E\{V(r_i^t) + E\{\tilde{U}(r_i^{t+1})\}\} \right],$$

(6.7)

where $\tilde{U}_i(r_i^{t+1})$ is the approximation of $U_i(r_i^{t+1})$ based on the heuristic base control. The exact RA based cluster head control is shown in Algorithm 3.

Given the approximation $\tilde{U}_i(r_i^t)$, which is calculated based on the heuristic base control, the computational saving of RA based cluster head control is evident. This is because only one single minimisation problem has to be solved at each
Algorithm 3: Rollout Algorithm Base Cluster Head Control

**Require:** number of cluster heads \( i \in \mathcal{I}_n \), total control time periods \( T \) and current control time period \( t \)

1: **for each** \( i \in \mathcal{I}_n \) **do**
2:     **for** \( t = 0 \rightarrow T \) **do**
3:         **Step 1: Compute transmission policy**
4:             **a) Base policy estimation**
5:                 **for** \( t = t + 1 \rightarrow T \) **do**
6:                     Set the based policy threshold \( T_i^{t+1} \) according to service rate \( f_i^{t+1} \)
7:                     \( T_i^{t+1} \leftarrow f_i^{t+1} \)
8:                     Calculate the number of receive packets \( r_i^{t+1} \) according to the threshold base policy
9:                     \( r_i^{t+1} = \begin{cases} T_i^{t+1} - q_i^{t+1} & \text{if } q_i^{t+1} < T_i^{t+1} \\ 0 & \text{otherwise} \end{cases} \)
10:                 **end for**
11:             The estimated aggregated utility \( \tilde{U}_i(r_i^{t+1}) \) from time period \( t + 1 \) is,
12:                 \( \tilde{U}_i(r_i^{t+1}) = \sum_{t=t+1}^{T} U_i(r_i^{t}) \)
13:             **b) Rollout algorithm based control**
14:                 **for each** \( r_i^{t} \in \tilde{S} \) **do**
15:                     Calculate the current time period utility value \( U_i(r_i^{t}) \)
16:                     The aggregated utility \( U_i(r_i^{t}) \) of cluster \( i \) is
17:                     \( U_i(r_i^{t}) = U_i(r_i^{t}) + \tilde{U}_i(r_i^{t+1}) \)
18:                 **end for**
19:                 \( r_i^{t} \leftarrow \arg \min U_i(r_i^{t}) \)
20: **Step 2: Assign the duty cycle**
21:             Assign the Stop-and-Wait duty cycle control parameter \( SO(r_i^{t}) \):
22:             \( SO(r_i^{t}) \leftarrow \left\lceil \log_2 \left( \frac{r_i^{t} \times P_s}{b} + D_{bcn} \right) \right\rceil \) or
23:             Assign the Go-Back-N duty cycle control parameter \( SO(r_i^{t}) \):
24:             \( SO(r_i^{t}) \leftarrow \left\lceil \log_2 \left( \frac{r_i^{t} D_p}{b} + \left\lceil \delta + D_{ACK} \right\rceil + \left\lceil D_{bcn} \right\rceil \right) \right\rceil \).
25: **end for**
26: **end for**

Time period. Noticed that even with readily available approximations \( \tilde{U}_i(r_i^{t+1}) \), the calculation of the minimisation over \( \pi_i \in \mathcal{S} \) may involve substantial computation. To further save the computation, a subset \( \tilde{S} \) of the promising controls is identified in the proposed RA based cluster head control. Thus, the minimisation over \( \mathcal{S} \) is
replaced by a minimisation over the subset $\tilde{S} \subset S$.

**Theorem 6.1.** Let’s denote $\hat{U}_i(r_t^i)$ as the estimate cost-to-go of RA based cluster head control, of which the control range is $\tilde{S} \subset S$. $U_i(r_t^i)$ as the expected actual aggregated utility incurred by RA based cluster head control. Then $U_i(r_t^i) \leq \hat{U}_i(r_t^i)$, which means $\hat{U}_i(r_t^i)$ is the utility lower bound of RA based cluster head control.

**Proof.** For $t = 0, 1, \cdots, T - 1$, denote

$$\hat{U}_i(r_t^i) = \max_{\pi_i \in \tilde{S}} \left[ V(r_t^i) + \mathbb{E}\{\hat{U}_i(r_{t+1}^i)\} \right].$$

(6.8)

Thus for all $q_t^i$, we have $\hat{U}(r_t^i) \leq \hat{U}(r_t^i)$, if the emphasis is given to the network performance component, let

$$\hat{U}(r_t^i) = V(r_t^i)$$

(6.9)

$$= \epsilon_1 \log(1 + \frac{f_t^i - r_t^i}{q_{t, \text{max}}^i}) - \epsilon_2 J(r_t^i) * \log \left(1 + J(r_t^i)\right).$$

Applying backward induction, we have $U(r_t^i) = \hat{U}(r_t^i) = V(r_t^i)$ for all $q_t^i$. Assuming that $U(r_{t+1}^i) \geq \hat{U}(r_{t+1}^i)$ for all $q_{t+1}^i$, we have

$$U(r_t^i) = \mathbb{E}\left\{ V(r_t^i) + U(r_{t+1}^i) \right\}$$

$$\geq \mathbb{E}\left\{ V(r_t^i) + \hat{U}(r_{t+1}^i) \right\}$$

$$\geq \mathbb{E}\left\{ V(r_t^i) + \hat{U}(r_{t+1}^i) \right\} = \hat{U}(r_t^i),$$

(6.10)

for all $q_t^i$.

The first equality above follows from the definition of the utility $U_i(r_t^i)$ of RA based cluster head control, while the first inequality follows from the induction
hypothesis, and the second inequality follows from the assumption $\hat{U}_i(r_t^i) \geq \hat{U}_i(r_t^i)$.

Then, we have $U_i(r_t^i) \geq \hat{U}_i(r_t^i) \geq \tilde{U}_i(r_t^i)$ for all $q_t^i$. Thus, the $\tilde{U}_i(r_t^i)$ is a readily obtainable utility lower bound for the utility function $U_i(r_t^i)$.

In addition, two remarks on computational complexity and control overhead of the proposed RA based cluster head control are provided as follows.

**Remark 6.1.** The proposed RA based cluster head control has lower computation complexity as compared to the PI optimal control and VI optimal control. If $S$ is the average search range of the devices, the computation complexity of DP algorithm is $O(TSN^2S)$, while that of the RA based cluster head control is only $O(NTS)$. 

**Remark 6.2.** The proposed suboptimal controls have lower synchronisation overhead as compared to controls in [WWXY10] and [BY13]. The proposed control does not need additional SYNC packet to ensure the devices are active at the same time as it employs the same BO as defined in IEEE 802.15.4 (2011) and all devices are activated at the beginning of each BI.

### 6.4 Reinforcement Learning (RL) based Cluster Head Control

In many practical scenarios it is not reasonable to assume the cluster heads have the perfect network information. The framework of RL known as the learning procedure is applied to such scenario where no perfect network information is available to compute the DP or PI based cluster head control. By applying RL algorithm, each cluster head updates its control decision according to its experience with different decisions without explicit modelling of the network.
6.4.1 Reinforcement Learning

RL is a simulation-based technique for solving the large-scale and complex problems, which is rooted in DP. RL is useful when a model is difficult or costly to derive. When the state-space of the system is large, it combines with algorithms to approximate the utility function, thereby generating a solution. It has been shown through mathematically rigorous arguments that RL can produce optimal or near-optimal solutions.

Q-learning is a model-free RL technique, which is used to find the optimal state-action policy for any finite state DP problems. It works by learning an action-utility table that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter. When such an action-utility table is learned, the optimal policy can be constructed by selecting the action with the highest utility at each state.

As one branch of ADP algorithms, Q-learning uses simulation to approximate the cost $j^\pi$ or the Q-factor $Q(s,a)$ of the current policy $\mu$. The optimal Q-factor of $(s,a)$ is defined as:

$$Q^*(s_t, a_t) = \mathbb{E}\left\{ g(s_t, a_t, w_t) + \gamma J^*(s_{t+1}) \right\}.$$  \hspace{1cm} (6.11)

The policy $\pi$ is applied at state $s_t$, and the optimal is applied thereafter. Then the Bellman’s equation is written as

$$J^*(s_t) = \min_{a \in A} Q^*(s_t, a_t), \hspace{0.5cm} \forall s \in S,$$  \hspace{1cm} (6.12)
and $Q_{t+1}$ is generated by

$$Q_{t+1}(s, a) = \mathbb{E}\left\{ g(s, a, w) + \alpha \min_{a \in A} Q_{t}(s', g) \right\},$$

(6.13) with $s' = f(s, a, w)$.

Once the optimal Q-factor $Q^*(s, a)$ are known, the model of the system is not needed, and the optimal policy can be implemented online by

$$\pi^*(s) = \min_{a \in A(s)} Q^*(s, u).$$

(6.14)

### 6.4.2 Reinforcement Learning based Cluster Head Control

The RL based cluster head control is formulated as a stochastic learning process. One challenge of the approach is that the cluster heads do not know the information of the other cluster heads due to the noncooperation among clusters. Then, the networking environment is non-stationary for all cluster heads and the convergence of learning process may not be assured. To alleviate the lack of mutual information exchange, the cluster heads form internal conjectures learn how the other cluster heads react to their present actions with only local observations form direct interactions with the network environment. Thus, learning is finished asymptotically by appropriately making use of the past experience.

In the cluster head control problem, the devices try to learn the optimal policy from its history of interaction with the network with Q-learning algorithm. In other words, the Q-learning does not require perfect information as it will try to learn the optimal solution directly by updating the state-action value (Q-value) while interacting with the network, thus it is also called model-free RL.
A history of cluster head \( i \) is a sequence of state-action-utility set \( \{ q_t^i, r_t^i, U_t^i \} \), which shows that the device was in state \( q_t^i \), did action \( r_t^i \), and received an immediate utility \( U_t^i \). Then the cluster head transferred into the next state \( q_{t+1}^i \).

For a given policy \( \pi \), a Q-value is the expected aggregated utility when executing action \( r_t^i \) at state \( q_t^i \) and then following the policy \( \pi \) thereafter, and it is defined as \( Q_\pi(q_t^i, r_t^i) = U_t^i + \delta \min Q_\pi(q_{t+1}^i, r_{t+1}^i) \), where \( \gamma \) is the discount factor. Then given the learning rate \( \delta \), the device will update its estimation for \( Q(q_t^i, r_t^i) \) at iteration \( t \) \((0 < t < T)\) as

\[
Q^t(q_t^i, r_t^i) = Q^{t-1}(q_t^i, r_t^i) + \delta \left\{ U_t^i + \min Q^t(q_t^i, r_t^i) - Q^{t-1}(q_t^i, r_t^i) \right\}.
\]

(6.15)

The learning rate \( \delta \in (0, 1] \) specifies how far the current estimation of \( Q(q_t^{t-1}, r_t^{t-1}) \) is adjusted toward the update target \( U_t^i + \min Q(q_t^i, r_t^i) \). The learning rate is typically time varying, decreasing with time. The expression inside the curly bracket in (6.15) is the temporal difference, which is the difference between the estimation of \( Q^*(q_t^i, r_t^i) \) at two successive time period, \( t \) and \( t + 1 \).

One of the conditions the sequence \( Q(q_t^i, r_t^i) \) provably converges to \( Q^*(q_t^i, r_t^i) \) is that the device keeps trying all actions in all states with nonzero probability. This means that the device must sometime explore, i.e., perform other actions than dictated by the current policy. Thus, finding a balance between exploitation of the current knowledge, and exploration of improving current knowledge by performing information-gathering actions is crucial to the efficiency of Q-learning algorithm.

The \( \epsilon \)-greedy, a variation on normal greedy selection, is applied in the proposed Q-learning based cluster head control, as shown in Algorithm 4. In \( \epsilon \)-greedy, the device identifies the best action according to the state-action table. At the same time, there is a small probability \( \epsilon \) that, rather than take the currently known best
action, the device will arbitrarily select an action from the remaining actions to perform the exploration.

**Algorithm 4 Q-Learning based Cluster Head Control**

**Require:** number of cluster heads \(i \in I_n\), maximum iteration times \(T\) and current control time period \(t\)

1: **for each** \(i \in I_n\) **do**
2:   Initialise the Q-factor table \(Q^0(q^0_i, r^0_i) = 0\)
3: **for** \(t = 1 \rightarrow T\) **do**
4:   **for** \(k = 0 \rightarrow 16\) (as each contention free period is divided into 16 slots) **do**
5:     **if** \(\text{rand} < \epsilon\) **then**
6:       **[Exploration step]:**
7:       Select the number of receive packets \(r^t_i\) arbitrarily
8:     **else**
9:        **[Exploration step]:**
10:       Update the Q-factor table \(Q^t(q^t_i, r^t_i)\) accordingly to:
11:       \(Q^t(q^t_i, r^t_i) = V^t_i + \delta \min Q^t(q^{t+1}_i, r^{t+1}_i)\)
12:       Calculate the number of receive packets \(r^t_i \leftarrow \arg \min Q^t(q^t_i, r^t_i)\)
13:       **end if**
14:   **end for**
15: **end for**
16: Assign the Stop-and-Wait duty cycle control parameter \(SO(r^t_i)\):
17:   \(SO(r^t_i) \leftarrow \left\lceil \log_2 \left( \frac{r^t_i \times P_s}{b} + D_{bcn} \right) \right\rceil\) or
18: Assign the Go-Back-N duty cycle control parameter \(SO(r^t_i)\):
19:   \(SO(r^t_i) \leftarrow \left\lceil \log_2 \left( \frac{r^t_i D_p}{\delta} \right) + \left[ \delta + D_{ACK} \right] + \left[ D_{bcn} \right] \right\rceil\) .
20: **end for**
21: **end for**
22: Update the Q-factor table \(Q(q^t_i, r^t_i)\) according to (6.10)

### 6.5 Simulation Results of Cluster Head Controls

In this section, the proposed PI based cluster head control, RA based cluster head control and the RL based cluster head control are evaluated in Matlab. Simulation results and analysis are provided.
6.5.1 Performance of PI Based Cluster Head Control

6.5.1.1 Simulation Setup

The MAC layer parameters used in the simulation are based on the IEEE 802.15.4 standard, as shown in Table 6-A. \( \alpha \) and \( \beta \) are set to be 0.2 and 0.4, respectively. Device energy consumption parameters of M2M gateway, cluster heads and M2M devices are based on XBee\textsuperscript{RF} Module data sheet [Int], respectively.

Table 6-A: Simulation Parameters of Cluster Head Controls

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>data rate</td>
<td>250kbps</td>
<td>frequency</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>transmit power</td>
<td>396 mw</td>
<td>packet size</td>
<td>50 bytes</td>
</tr>
<tr>
<td>receive power</td>
<td>102.3 mw</td>
<td>CCA size</td>
<td>8 symbols</td>
</tr>
<tr>
<td>idle listen power</td>
<td>102.3 mw</td>
<td>ACK packet size</td>
<td>10 symbols</td>
</tr>
<tr>
<td>sleep power</td>
<td>0.033 mw</td>
<td>unit backoff period</td>
<td>20 symbols</td>
</tr>
</tbody>
</table>

In this simulation, the performance of the proposed PI cluster head control is compared with the DP optimal control and a benchmark control.

- **PI Optimal control**: the optimal \( r_i^* \) follows the threshold structure based on the Theorem 5.1 and is computed by policy iteration (PI) based control algorithm. This method is labeled as PI optimal.

- **VI Optimal control**: this optimal solution follows the basic concept of DP. At each time period \( t \), \( r_i^* \) is the one obtains the maximum utility value \( U(r_i^*) \). This method is labeled as value iteration (VI) optimal.

- **Benchmark control**: the benchmark control aims at maximising the number of
received packets $r_t^i$ to reduce the end-to-end delay. The maximum $SO$ is bounded by the service rate $f_t^i$ of the cluster head.

6.5.1.2 Simulation Results

The performance of the proposed PI based cluster head control is evaluated in terms of the cluster utility, energy efficiency, end-to-end delay, cluster throughput, packet drop ratio and cluster profit. The definition of all the performance metrics are given in Chapter 4.4.2.

Fig. 6.1 shows the energy efficiency of the cluster head. Since $SO$ can only be integer as defined by IEEE 802.15.4, the change of active period length is radical. The idle listening energy consumption caused by the change of $SO$ leads to the non monopoly change of the energy efficiency curve of VI optimal control between 20-40 kbps.
The energy efficiency curves of benchmark control, VI optimal control and PI optimal become flat after the packet arrival rate is higher than 40 kbps. This is because the maximum number of transmitted packets is reached, which means the network is saturated. At the stable state, the energy efficiency of PI optimal control is slightly lower than that of the compared two controls.

![Figure 6.2: End-to-End delay of PI based cluster head control.](image)

Fig. 6.2 shows the averaged end-to-end delay of the successfully transmitted packets. The average end-to-end delay increased with the increase of the packet arrival rates. Among the compared results of different controls, the end-to-end delay of PI based cluster head control is higher than that of the VI optimal control, but lower than that of the benchmark control. The rapid increase of the end-to-end curve of VI optimal control between 20-30 kbps is due to the same SO with increased traffic arrival rate. This lead to more packets need to be buffered before being transmitted. Then, the end-to-end delay of VI optimal control begin to decrease between 30-40 kbps, this is due to the radical increase of SO which
extend the active period lengths, which allowed more packets to be transmitted directly without any buffering.

Fig. 6.3 shows the packet drop ratio of the cluster head. No packet drop is observed when packet arrival rate is less than 15 kbps for both PI based optimal control and the benchmark control. The packets begin to be dropped when the packet arrival rate is larger than 35 kbps for the VI based optimal control. The packet drop ratio of PI optimal control is slightly less than that of the benchmark control.

The packet drop in this simulation happened only when the queue length reached the limited buffer size, and no transmission drop is considered. Thus, the packet drop ratios are always less than 1 for all simulated controls.

Fig. 6.4 shows the average cluster throughput. The throughput of PI based cluster head control is higher than that of the benchmark control, but lower than
that of the VI optimal cluster head control as expected. The concave change of
the throughput curve of PI based cluster head control between 10-20 kbps and VI
optimal cluster head control between 20 - 40 kbps is due to the radical change of
$SO$. The throughput curves become flat when the network reached its saturation
stable state.

Fig. 6.5 shows the cluster utilities of the compared controls. It can be seen
that the utility value of the PI based cluster head control is higher than that of
the benchmark control under all simulated packet arrival rates. Compare to that
of the VI optimal control, the computational complexity reduction of PI based
cluster head control is at the cost of the degradation of utility.

The utility value curves of all three compared controls have the concave shape.
This also illustrated that the designed cluster utility function is a concave function.
For a given cluster head service rate, there is a most suitable traffic load for the
cluster, at which the cluster head will gain the highest utility.

### 6.5.2 Performance of RA based Cluster Head Control

#### 6.5.2.1 Simulation Setup

As shown in Table 6-B, the energy consumption parameters for M2M gateways, cluster heads and devices are based on Cisco 819 [Cis], XBee and XBee-Pro RF module data sheet [Int], respectively.

The performance of the proposed RA based cluster head control is compared with that of a benchmark control, the PI optimal control, and the heuristic base control of the RA based cluster head control.

*Benchmark control*: the benchmark control aims at maximising the number of received packets $r'_t$ to reduce the end-to-end delay. The maximum $SO$ is bounded

![Cluster utility of PI based cluster head control.](image-url)
Table 6-B: Energy Consumption and Buffer Parameters

<table>
<thead>
<tr>
<th>Device</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2M Devices</td>
<td>Memory</td>
<td>32 Kb</td>
</tr>
<tr>
<td></td>
<td>Device sensitivity</td>
<td>- 90 dBm</td>
</tr>
<tr>
<td></td>
<td>Transmit power</td>
<td>108.9 mw</td>
</tr>
<tr>
<td></td>
<td>Receiving power</td>
<td>92.4 mw</td>
</tr>
<tr>
<td></td>
<td>Idle listen power</td>
<td>92.4 mw</td>
</tr>
<tr>
<td></td>
<td>Sleep power</td>
<td>0.033 mw</td>
</tr>
<tr>
<td>Cluster Heads</td>
<td>Memory</td>
<td>32 Kb</td>
</tr>
<tr>
<td></td>
<td>Device sensitivity</td>
<td>- 102 dBm</td>
</tr>
<tr>
<td></td>
<td>Transmit power</td>
<td>396 mw</td>
</tr>
<tr>
<td></td>
<td>Receiving power</td>
<td>102.3 mw</td>
</tr>
<tr>
<td></td>
<td>Idle listen power</td>
<td>102.3 mw</td>
</tr>
<tr>
<td></td>
<td>Sleep power</td>
<td>0.033 mw</td>
</tr>
<tr>
<td>M2M Gateway</td>
<td>Memory</td>
<td>1 GB</td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>25 w</td>
</tr>
</tbody>
</table>

by the service rate $f^t_i$ of the cluster head.

**PI Optimal control:** based on Theorem 5.1 the optimal $r^*_i$ follows threshold structure. The optimal cluster head control in this section is computed by PI based control algorithm.

**Base control:** the heuristic base control has a fixed threshold equals to $f^t_i$. Thus, $r^t_i = f^t_i - q^t_i$. The maximum $SO$ is bounded by the service rate $f^t_i$ of the cluster head.

**RA based control:** Rollout based control will do one search at each time period to find the maximum utility at each time period, while the future utility is estimated by applying the heuristic based control. The maximum $SO$ is bounded the search range $\bar{S}$ at each time period.
6.5.2.2 Simulation Results

In this part, the performance of the proposed RA based cluster head control is presented for cluster heads with both limited buffer capacity and sufficient buffer capacity.

Fig. 6.6 shows energy efficiency performance of the compared cluster head controls. Fig. 6.6 a) is the performance of cluster head with limited buffer capacity and Fig. 6.6 b) is the performance of cluster head with sufficient buffer capacity. In both Fig. 6.6 a) and Fig. 6.6 b), the energy efficiency curves of RA base control and DP optimal control have the wave shape before the packet arrival rate is lower than 30kbps.

As has been pointed out earlier, the $SO$ is integer and the duty cycle is defined as $2^{SO-BO}$, thus the change of duty cycle is radical. For a given duty cycle, the energy efficiency curve has a convex shape with the increase of the number of transmitted packets. This convex shape illustrated the trade-off between the idle listening and transmitting energy consumption. The decrease of the energy efficiency is due to the long idle listening energy consumption and the lack of sufficient packets to transmit.

Compared with the Fig. 6.6 a) and Fig. 6.6 b), the energy efficiency of DP optimal control, RA based cluster head control and the benchmark control performed similarly for cluster heads with limited buffer capacity and sufficient buffer capacity. However, the performance of benchmark control various a lot. This shows the cluster head buffer size has a direct impact on the performance of the benchmark control, thus a co-relation between the fixed threshold selection and the buffer capacity may exist.
Fig. 6.6 shows the energy efficiency of RA based cluster head control.

Fig. 6.7 shows the end-to-end delay performance of the compared cluster head controls. For cluster heads with both limited and sufficient buffer capacity, the end-to-end delay increases with the increase of the packet arrival rates for all
compared controls. The end-to-end delay of RA based control is close to that of the DP optimal control, and on average is 4 times less than that of the benchmark control.
It can be obtained from Fig. 6.7 a) and Fig. 6.7 b) that the buffer capacity of the cluster head has a direct impact on end-to-end delay performance of the fixed threshold base policy.

Fig. 6.8 shows the cluster throughput performance. The cluster throughput increase with the increase of the packet arrival rates for all compared controls. The cluster throughputs became stable when the cluster heads reached their maximum packet transmission capabilities. The throughput of RA based cluster head control is closed to that of the DP optimal control. At the stable state, RA based cluster head control achieved 5.5 times throughput increase compared to that of the benchmark control for both cases where cluster head has limited and sufficient buffer capacity.

Comparing Fig. 6.8 a) and Fig. 6.8 b), the performance of the base policy various for cluster heads with limited buffer capacity and sufficient buffer capacity, while the other three controls performed similarly. This also shown that the DP optimal control and RA based control are able to adjust the control for devices with different capacities.

Fig. 6.9 shows the packet drop ratio performance of the compared controls. The packet drop ratio of RA based cluster head control is close to that of the DP optimal control and slightly lower than that of the base policy under all simulated packet arrival rates. For RA based cluster head control the packets begin to be dropped when the packet arrival rate is around 30 kbps, while that of the benchmark control is around 15 kbps.

Comparing Fig. 6.9 a) and Fig. 6.9 b), the packet drop ratio of the base policy shown tighter co-relation with the cluster head buffer capacity. The packet drop ratio of DP optimal control in Fig. 6.9 b) begin to increase when the packet arrival
Chapter 6. Cluster Head Control with Cluster Utility Maximisation

Figure 6.8: Throughput of RA based cluster head control.

rate is larger than 30 kbps while that in Fig. 6.9a) is around 27 kbps.

The packet drop ratio performance of RA based cluster head is similar for
cluster head with limited and sufficient buffer capacity. This indicates that the packet drop in the simulation mainly happened at the M2M devices rather than at the cluster head.
Fig. 6.10 shows the cluster utility performance of the compared controls. The RA based cluster head achieved close performance compared to that of the DP optimal control for both cases where cluster heads have limited and sufficient buffer capacity.

Figure 6.10: Cluster utility of RA based cluster head control.
capacities. For RA based cluster head control and DP optimal control, the peak utility values are about 5 times higher than that of the benchmark control.

From the peak points of the utility curves, it can be seen that compared to the benchmark control, the proposed RA based cluster head control and DP optimal control are able to deal with heavy traffic conditions while achieving higher utility value. In addition, the performance of the base policy is highly related with the buffer capacity of the cluster heads.

### 6.5.3 Performance of RL based Cluster Head Control

#### 6.5.3.1 Simulation Setup

The performance of the proposed Q-learning based control is compared with a benchmark control, DP optimal and RA based cluster head control.

*The DP optimal control:* processes DP exhausted search for entire observed time periods to find the optimal control.

*Benchmark control:* aims at minimising the end-to-end delay by maximising the number of transmitted packets $r^t_i$ at each time period. A fixed $SO = 3$ is set, which is sufficient large to transmit all generated packets.

*Optimal control:* based on the **Theorem 5.1** the optimal $r^t_i^*$ follows threshold structure. The optimal duty cycle control in this section is computed by PI based control algorithm.

*RA based control:* RA based control is the rollout algorithm based control which searches the maximised utility of current time period and estimates future utility based on a heuristic control. The maximum $SO$ is bounded the search range $\bar{S}$ at
each time period.

For generosity, the action-utility Q-factor is initialised to zero. The initial bid price of each cluster head is an unit price proportional to the amount of its child devices. The choice of the learning parameters: greedy parameter $\epsilon$, learning rate $\delta$, discount factor $\gamma$ are presented as following.

The $\delta$ is the learning rate of the action-utility Q-factor. The small $\delta$ value increases the time of the learning process. The large value introduces the oscillations of the Q elements. In this simulation, $\delta$ is chosen from the set 0.1, 0.01, 0.001. For all three parameter values, similar results are obtained. Specifically, $\delta = 0.01$ is chosen in the simulation.

The discount factor $\gamma$ determines the relative importance of the short-term utility and long-term utility. For learning with focus on maximising long-term utility, this parameter is set arbitrarily near 1. In this simulation, $\gamma = 0.95$.

In $\epsilon$-greedy method the parameter $\epsilon$ determines the probability of random action selection and must be taken from the interval $(0,1]$. This parameter impacts the possibility of environment exploration, which means searching for a better solution by changing the currently used policy. If the value of $\epsilon$ tends to zero, then the probability of random action selection also tends to zero. In this simulation, the greedy parameter is set to be 0.5.

6.5.3.2 Simulation Results

Fig. 6.11 shows the energy efficiency of the compared controls. Q-learning based control has the highest energy efficiency compared to the other controls, especially after the packet arrival rate is higher than 30 kbps. The energy efficiency curves
Chapter 6. Cluster Head Control with Cluster Utility Maximisation

Figure 6.11: Energy efficiency of RL based cluster head control.

of the benchmark control, DP optimal control and rollout based control are flat after the packet arrival rate is higher than 30 kbps.

Since the change of duty cycle is radical, the idle listening energy consumption caused by the change of $SO$ lead to the non monopoly change of the energy efficiency curves. The energy efficiency of Q-learning based control is doubled compared to that of the other compared controls at stable states.

Fig. 6.12 shows the delay performance of the compared controls. The proposed Q-learning based cluster head control has higher end-to-end delay compared to that of the DP optimal control and rollout based control, but lower than that of the benchmark control.

It is observed from Fig. 6.11 and Fig. 6.12 that the increase of energy efficiency is always came with the increase of end-to-end delay for all compared controls. This observation also illustrates the trade-off between energy efficiency and end-to-end
delay.

Fig. 6.13 shows the cluster throughput of the compared controls. The through-
put of Q-learning based control is lower than that of the DP optimal control and rollout based control, but higher than that of the benchmark control. The lower throughput of Q-learning based control is due to its needs to gradually approach to the optimal solution over time.

The performance of the overall two-hop packet drop ratio is shown in Fig. 6.14. Although the packet drop is not considered in the utility function, Q-learning based control is able to reduce the packet drop ratio. The Q-learning based control has lower packet drop ratio compared to that of the DP optimal. In addition, the packet drop ratio of Q-learning based control only started to increase when the packet arrival rate is higher than 30 kbps, while that of the optimal control started around 25 kbps.

Fig. 6.15 shows the utility of the compared controls. The utility curve of the Q-learning based control is lower than that of the DP optimal control and rollout
based control, but higher than that of the benchmark control. This is because the learning process of Q-learning based control needs to gradually approach the optimal solution over time.

The utility curves have concave shape, where the peaks of the curves show the maximum utilities the cluster head can achieve under the simulated scenarios. From the peaks of the utility curves, it can be seen that the Q-learning based control is capable of dealing with the heavy traffic situations.

The proposed Q-learning based duty cycle control achieved higher energy efficiency, similar end-to-end delay and packet drop ratio compare to the benchmark control. When compared with the DP optimal and RA based control, the proposed Q-learning based control struck a balance between optimality and algorithm stability.
6.6 Summary

This chapter focuses on the cluster utility maximisation problem for IEEE 802.15.4 based hierarchical M2M networks. The cluster utility function is designed with both empirical component and economic component. The PI based optimal cluster head control is proposed to reduce the computational complexity of the DP optimal control. Then, a suboptimal RA based cluster head control is proposed to further reduce the computational complexity of the PI based cluster head control. In the end, a Q-learning based cluster head control is developed for the situation where priori network information is not available.

Simulation results shown that the proposed PI based cluster head control has close performance to the optimal control while achieving exponential complexity reduction. The RA based cluster head control reached a balanced trade-off between the computational complexity and optimality. Compared to the optimal and the other cluster head controls, the Q-learning based cluster head control achieved the best balance between optimality and stability.
Chapter 7

Gateway Control with Network Utility Optimisation

This chapter focuses on the gateway control of the proposed control framework. The proposed gateway controls aim to solve the formulated optimisation problem $\mathcal{P}4$ in Chapter 4.

As has been discussed in Section 3.3 the applications running in the hierarchical M2M networks are classified into four classes. In this chapter, the utility functions are formulated for the four classes applications. The gateway control with network utility optimisation is proposed. To allocate resources to clusters while addressing their different application requirements, the gateway control is solved with the aim of maximising the network utility. Novel distributed gateway control algorithms for multi-application hierarchical M2M networks are proposed.

7.1 Gateway Utility Function Design

Given the nature of the devices and topology, the hierarchical M2M networks may have different types of applications [ZHW+12, JSKP10]. Based on the modelled hierarchical M2M system in Chapter 3, the application runs within each cluster may
be different. As has pointed out by [ZHW+12], the applications that run in the hierarchical M2M networks are classified into four classes: traditional elastic application, hard-real time application, delay adaptive application and rate adaptive applications.

According to the designed control framework in Chapter 4, the M2M gateway control aims at maximising the long-term network utility while satisfying different application requirements. The first task for gateway control optimisation is to design the utility functions for different applications.

The traditional logarithmic utility functions can only capture the QoS requirement for elastic applications, where a non-zero utility is presented as long as it has a non-zero resource block allocated to it. However, it is not suitable to use logarithmic utility functions for real-time traffics, such as audio and video surveillance, real-time traffic monitoring and real-time seismic activity monitoring. These real-time traffics are generally generated by inelastic applications. The key feature of inelastic applications is that they will keep present nearly zero-utility unless a minimum amount of resource blocks have been allocated.

According to [She95, JSKP10], for the inelastic applications (hard-real time applications, delay adaptive applications and rate adaptive applications), the utility function $U_n(f^i_t, \theta^i_t)$ can be presented by a sigmoid function, while for the elastic applications the utility function $U_n(f^i_t, \theta^i_t)$ can be presented by a logarithmic function. Specifically, the application indicator $\theta^i_t$ specifies the application type of cluster head. If the application of cluster head $i$ falls into the elastic applications, then the utility function of cluster head $i$ is given as,

$$U_n(f^i_t) = \log(1 + f^i_t), \quad (7.1)$$
where $f_i^t$ is offset by +1 to ensure the positiveness of the utility.

If the application of cluster head $i$ falls into the inelastic applications, the utility function of cluster head $i$ is given as,

$$U_n(f_i^t) = \begin{cases} 
0 & \text{if } f_i^t < 0 \\
\frac{1}{1+e^{-a(f_i^t-b)}} & \text{if } 0 \leq f_i^t \leq C_{i,n} \\
1 & \text{if } f_i^t > C_{i,n}
\end{cases} (7.2)$$

where the utility parameter set $(a,b)$ controls the slope of the sigmoid. Fig. 7.1 shows an example of the modelled utility functions for the four application classes.

Recall the gateway control optimisation problem $\mathcal{P}4$ which has been defined in Chapter 4. The gateway control optimisation is formulated by applying the “pseudo utility” optimisation problem to accommodate the mixed applications.

![Utility function for different application classes.](image)
running within the network, which is expressed as,

\[ P_4 : \max \sum_{t=0}^{T} \sum_{i \in I_n} U_n(f^t_i, p^t_i, \theta^t_i) \]  \hspace{1cm} (7.3a)

\[ \text{s.t.} \quad \sum_{i \in I_n} f^t_i \leq C^t_{n,b}, \]  \hspace{1cm} (7.3b)

\[ 0 \leq f^t_i \leq C^t_{i,n}. \]  \hspace{1cm} (7.3c)

where \( p^t_i \) is the bid price provided by the cluster head \( i \). \( \theta^t_i \) is the application indicator which indicates the utility function of the cluster head \( i \) is in logarithmic form as defined in (7.1) or in sigmoid form as defined in (7.2).

As discussed previously in Chapter 4, the classic NUM framework provides not only an efficient congestion control mechanism for the network, but also gives an easy way to provide a fair resource allocation among competing clusters. In addition, the utility-fair control is proposed for CSMA based wireless networks in [JSKP10] with a queue back-pressure-based algorithm.

The objective of the gateway control optimisation problem \( P_4 \) is to optimise its overall network utility subject to different link capacity constraints, while satisfying different application requirements among clusters.

### 7.2 Lagrangian based Gateway Control

According to [JSKP10], by applying the Lagrangian approach, the control result of \( P_4 \) is utility proportional fair. By associating dual variables \( \omega^t_i \geq 0 \) to the second
constraint, a partial Lagrangian of $P_4$ is

$$
\mathcal{L}(f^t_i, p^t_i, \theta^t_i; \omega^t_i) = \sum_{i \in I_n} \mathcal{U}_n(f^t_i, p^t_i, \theta^t_i) + \sum_{k=1}^t \omega^t_i \left( \sum_{i \in I_n} f^t_i - \sum_{i \in I_n} q^t_i - \sum_{i \in I_n} r^t_i \right) \\
= \sum_{i \in I_n} \mathcal{U}_n(f^t_i, p^t_i, \theta^t_i) - \sum_{i \in I_n} \omega^t_i (q^t_i + r^t_i) + \sum_{i \in I_n} \sum_{k=1}^t \omega^t_i f^t_i. 
$$

Since the vectors $q^t_i$ and $r^t_i$ are provided with fixed values at time period $t$, $f^t_i$ is solved in the below sub-problem,

$$
\max_{f^t_i} \sum_{i \in I_n} \sum_{k=1}^t \omega^t_i f^t_i \\
\text{s.t.} \quad \sum_{i \in I_n} f^t_i \leq C_{n,b} \\
\quad f^t_i \leq q^t_i + r^t_i. 
$$

The solution of the above problem is quite straightforward: at each time period $t$ which has 16 slots, the gateway schedules the transmission to the cluster head with maximum service index $\omega^t_i = p^t_i \cdot q^t_i / q^t_{i \text{max}}$.

Substitute the solution of (7.5) into (7.4), we have

$$
\mathcal{L}(f^t_i, p^t_i, \theta^t_i; \omega^t_i) = \sum_{i \in I_n} p^t_i f^t_i + \sum_{i \in I_n} \mathcal{U}_n(f^t_i, p^t_i, \theta^t_i) - \sum_{i \in I_n} p^t_i q^t_i. 
$$

Then, the distributed gateway control algorithm is given in Algorithm 5.

The number of packets each cluster head can transmit at time period $t$ is de-
Algorithm 5 Proposed Network Utility Optimisation Gateway Control (P-UPF)

Require: cluster heads $i \in \mathcal{I}$, total control time periods $T$ and current time period $t$

1: Initialise the number of forward packets $f_t^0 = 0$
2: for $t = 0 \rightarrow T$ do
3:   for each $i \in \mathcal{I}$ do
4:     Calculate the service index $\omega_t^i = \frac{q_t^i}{q_{\text{max}}^i} \cdot p_t^i$
5:     Update the utility $U_n(f_t^i, p_t^i, \theta_t^i)$ if slot $k$ is allocated to $i$
6:     $U_n(f_t^i, p_t^i, \theta_t^i) \leftarrow U_n(f_t^i, p_t^i, \theta_t^i) + U_n(f_t^{i-1}, p_t^i, \theta_t^i)$
7:     end for
8: end for
9: Schedule the cluster head $i \in \mathcal{I}$ which has the maximum increase of utility.
10: The optimal number of transmit packets of cluster head $i$ is,
11: $f_t^* = \max_{i \in \mathcal{I}} U_n(f_t^i, p_t^i, \theta_t^i)$
12: Allocate cluster $i$ with $f_t^i$.
13: Calculate the gateway utility at time period $t$
14: $U_n = \sum_{i \in \mathcal{I}} U_n(f_t^i, p_t^i, \theta_t^i)$

terminated by

$$f_t^* = \arg \max_{i \in \mathcal{I}} (\sum_{i \in \mathcal{I}} U_n(f_t^i, p_t^i) - \sum_{i \in \mathcal{I}} p_t^i(q_t^i + r_t^i)) = U_n^{-1}(\frac{1}{p_t^i}), \quad (7.7)$$

where the bid price $p_t^i$ is updated by

$$p_t^i \leftarrow \left[ p_t^i - \alpha (\sum_{i \in \mathcal{I}} f_t^i - \sum_{i \in \mathcal{I}} q_t^i - \sum_{i \in \mathcal{I}} r_t^i) \right]^+ \quad (7.8)$$

7.3 Mixed Integer Programming based Gateway Control

According to the IEEE 802.15.4 standard, there are 16 slots in each CAP. It should be noted that standard Lagrangian-based duality technique cannot guarantee the maximum utilisation of all 16 slots, as the whole 16 slots can only be allocated to
one cluster for each time period $t$.

To increase the slot utilisation by allowing different clusters to transmit in the same time period but different slots, the optimisation problem $P_5$ is transferred into a mixed integer programming (MIP) problem. The MIP problem defines a slot allocation matrix $\text{[WOM09]}$ which ensures the adjacency of allocated slots for each cluster head.

The size of the slots allocation matrix is $K \times A$, where each row correspond to the slot index and each column corresponds to a feasible (meeting adjacency restriction) slot allocation pattern, and $A$ denotes the total number of feasible allocation patterns given by $A = 0.5 \times (K^2 + K)$. In each allocation pattern, 1 means the slot is allocated to a cluster, 0 means no cluster head is allocated to this slot. The idea of this slot allocation matrix is illustrated by (K=3):

$$M^K = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 \\
\end{bmatrix}.$$ \hfill (7.9)

A slot indicator vector $\mathbf{x} \triangleq [x_i]_{K \times 1}$, where $x_i = [x_{i,a}]_{A \times 1}$ is defined, such that each entry $x_{i,a}$ indicates whether the slot allocation pattern $a$ is allocated to the cluster head $i$ or not. In this way, the gateway control problem $P_5$ is rewritten as
where $f^t_i(x^a_i) = \min(\sum_{a=1}^A x^a_i R, q_i^{t-1} + r_i^{t-1})$, and $R$ is the maximum number of packers each time slot $k \leq 16$ could support. The constraints 7.10b) and 7.10c) gives the search space of $f^t_i$. The constraint 7.10d) ensures that at most one allocation pattern is chosen for each user. The constraint 7.10e), where $M^t_{k,a}$

**Algorithm 6** MIP based Gateway Control

**Require:** the set of cluster head $\mathcal{I}$, control time period $T$

1: Initialise $f^0_i = 0$
2: for $t = 0 \rightarrow T$ do
3:     for each slot number $k \leq 16$ do
4:         for each $i \in \mathcal{I}$, do
5:             Calculate the service index $\omega^t_i = \frac{q_i^t}{q_{max}} \cdot p^t_i$
6:             Calculate $U_n(f^t_i, p^t_i, \theta^t_i)$ if slot $k$ is allocated to $i$
7:             $U_n(f^t_i, p^t_i, \theta^t_i) \leftarrow U_n(f^t_i, p^t_i, \theta^t_i) + U_n(f^k_i, p^t_i, \theta^t_i))$
8:         end for
9:     Schedule packets transmission at time slot $k$: $f^{k*}_i = \max \Delta U^t_n(f^k_i, p^t_i, \theta^t_i)$
10:    Allocate current slot $k$ to cluster $i$,
11:    $f^t_i \leftarrow f^t_i + f^{k*}_i$
12: end for
13: Calculate the gateway utility at time period $t$
14: $U_n = \sum_{i \in \mathcal{I}} U_n(f^t_i, p^t_i, \theta^t_i)$
15: end for
denotes the $k^{th}$ row and $a^{th}$ column of the matrix $M^k$, ensures the exclusivity of allocated slots.

The optimisation problem $\mathcal{P}5$ is an MIP equivalent of the problem $\mathcal{P}4$. The MIP based gateway control algorithm is given in Algorithm 6.

### 7.4 Gateway Control Simulation Results

Table 7-A shows the MAC layer simulation parameters in the simulation. Energy consumption parameters for the M2M gateways, cluster heads and M2M devices are based on Cisco 819 [Cis], XBee and XBee-Pro RF Module data sheet [Int], as shown in Table 7-B, respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>data rate</td>
<td>250kbps</td>
<td>frequency</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>transmit power</td>
<td>36.5 mw</td>
<td>packet size</td>
<td>100 bytes</td>
</tr>
<tr>
<td>receive power</td>
<td>41.4 mw</td>
<td>CCA size</td>
<td>8 symbols</td>
</tr>
<tr>
<td>idle listen power</td>
<td>41.4 mw</td>
<td>ACK packet size</td>
<td>10 symbols</td>
</tr>
<tr>
<td>sleep power</td>
<td>0.042 mw</td>
<td>unit backoff period</td>
<td>20 symbols</td>
</tr>
<tr>
<td>learning rate</td>
<td>0.9</td>
<td>discount factor</td>
<td>0.5</td>
</tr>
</tbody>
</table>

To show the overall network performance of the proposed joint scheduling and duty cycle control framework, all the gateway control, cluster head control and the duty cycle control for IEEE 802.15.4 are implemented in the simulation.

With the purpose of demonstrating the proposed gateway control algorithms, the implemented cluster head control and the duty cycle control are chosen as $Q$-
Table 7-B: Gateway Control Energy and Buffer Parameters

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2M Devices</td>
<td>Memory</td>
<td>32 Kb</td>
</tr>
<tr>
<td></td>
<td>Device sensitivity</td>
<td>-90 dBm</td>
</tr>
<tr>
<td></td>
<td>Transmit power</td>
<td>108.9 mw</td>
</tr>
<tr>
<td></td>
<td>Receiving power</td>
<td>92.4 mw</td>
</tr>
<tr>
<td></td>
<td>Idle listen power</td>
<td>92.4 mw</td>
</tr>
<tr>
<td></td>
<td>Sleep power</td>
<td>0.033 mw</td>
</tr>
<tr>
<td>Cluster Heads</td>
<td>Memory</td>
<td>32 Kb</td>
</tr>
<tr>
<td></td>
<td>Device sensitivity</td>
<td>-102 dBm</td>
</tr>
<tr>
<td></td>
<td>Transmit power</td>
<td>396 mw</td>
</tr>
<tr>
<td></td>
<td>Receiving power</td>
<td>102.3 mw</td>
</tr>
<tr>
<td></td>
<td>Idle listen power</td>
<td>102.3 mw</td>
</tr>
<tr>
<td></td>
<td>Sleep power</td>
<td>0.033 mw</td>
</tr>
<tr>
<td>M2M Gateway</td>
<td>Memory</td>
<td>1 GB</td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>25 W</td>
</tr>
</tbody>
</table>

learning based cluster head control and duty cycle control with Go-back-N ARQ, respectively.

The gateway control simulation results are presented with two scenarios: i) the cluster heads run same class application, but different priorities. In our simulation, suppose that all applications are elastic applications; ii) the cluster heads run applications that fall into different classes. As an example, the simulation results of the M2M network with mixed elastic applications and the rate adaptive applications are presented. The simulation results of the other traffic combinations are omitted due to the space limitation.
The network setting for the simulation is given in Table 7-C. The low traffic load means that the traffic range of the cluster is between 0 to 120kbps and the high traffic load means the traffic range of the cluster is between 0 to 180kbps.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Traffic load</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Priority</td>
<td>Application</td>
</tr>
<tr>
<td>1</td>
<td>Low (0-120kbps)</td>
<td>High</td>
<td>Inelastic</td>
</tr>
<tr>
<td>2</td>
<td>Low (0-120kbps)</td>
<td>High</td>
<td>Inelastic</td>
</tr>
<tr>
<td>3</td>
<td>High (0-180kbps)</td>
<td>High</td>
<td>Inelastic</td>
</tr>
<tr>
<td>4</td>
<td>Low (0-120kbps)</td>
<td>Low</td>
<td>Elastic</td>
</tr>
<tr>
<td>5</td>
<td>Low (0-120kbps)</td>
<td>Low</td>
<td>Elastic</td>
</tr>
</tbody>
</table>

The performance of different gateway control algorithms are compared, legend as No PF control, UPF control, P-UPF control and MIP control. The details of the compared gateway control algorithms are,

- **No PF control**: no secluding algorithm is implemented for the gateway control. The number of forwarded packets to the gateway of cluster head $i$ $f_i^t$ is a random value.

- **UPF control**: the classic utility proportional fair scheduling algorithm is implemented on the gateway. No application differentiation has been taken into account for different clusters.

- **P-UPF control**: The proposed gateway control algorithm is based on the classic UPF control. The application differentiation is achieved by taking the QoS indicator $\theta_i$ of cluster head $i$ into account. The P-UPF is shown in Algorithm 5.

- **MIP control**: MIP control is the proposed gateway control algorithm in Algo-
rithm 6. The main advantage of MIP control is reducing the transmission switch by ensuring the adjacency of allocated slots for each cluster head.

7.4.1 Same Class Applications with Different Priorities

The network setting of this scenario is shown in Table 7-C as scenario 1, where all cluster heads run same class applications (elastic application) but with different priorities.

In this scenario, five clusters are randomly deployed. Among these 5 clusters, cluster 1, 2 and 3 run high QoS priority applications with a large $\theta_i$, while 4 and 5 have low QoS priority applications with a small $\theta_i$.

For the purpose of illustration, Fig 7.2 a) and Fig 7.2 b) show an snapshot of the averaged throughput and the throughput cumulative density function (CDF) of each cluster head. It shows that the proposed P-UPF and MIP based gateway controls is able to provide priority differentiation. For both P-UPF and MIP based gateway controls, the throughput of the high priority clusters (cluster 1, 2 and 3) are higher than that of the low priority clusters (cluster 4 and 5), while the classic UPF and random gateway controls share similar throughputs among 5 clusters. In addition, for the clusters with same priority, the throughput of cluster with heavy traffic load (cluster 3) is higher than that of cluster with light traffic load (cluster 1 and 2). The smooth trend of the CDF function also shown that the simulation iteration is large enough to get the representative averaged results.

Fig. 7.3 a) shows the normalised network total throughput. For all simulated gateway controls, the throughput increased with the increase of packet arrival rate, then the network got saturated when the normalised throughput reached around
Chapter 7. Gateway Control with Network Utility Optimisation

Figure 7.2: Throughput of clusters run different priority applications.

90%. The saturated normalised throughput is less than 1 due to the contention of CSMA/CA. For UPF and MIP based gateway controls, the network became saturated when the network packet arrival rate is around 300kbps, and for P-
UPF it is around 350kbps. Compared to that of the random gateway control, the network got saturated much earlier, which is around 200 kbps.
Fig. 7.3 b) shows that the network total utility. It can be seen that the utility of the random gateway control decreased dramatically after the packet arrival rate is 200kbps. The network utility of UPF, P-UPF and MIP based controls are very tight. The utility of the proposed P-UPF and MIP based controls are slightly lower than that of UPF. This is due to the aim of UPF is to maximise the network throughput, while P-UPF and MIP based gateway controls have also taking the priority of cluster heads into consideration.

The following results presented in Fig 7.4 - Fig 7.5 are the performance of example cluster 1, cluster 3 and cluster 5. The performance are presented in terms of throughput, utility, energy efficiency, end-to-end delay, packet drop ratio and profit.

The throughput of each example cluster head (cluster 1, 3 and 5) in Fig. 7.4 a) Fig. 7.5 a) and Fig. 7.6 a) shown the proposed P-UPF and MIP based controls with higher priority cluster heads are higher than that of the UPF and random controls.

For the MIP based control, the throughput of the cluster with high priority and higher traffic load is similar to that of the cluster with high priority and lower traffic load, while that of P-UPF based control has decreased. On the other hand, with MIP based control, the throughput of the cluster with lower priority is lower than that of P-UPF after network saturation. This implies that, MIP improved the throughput for clusters with higher traffic load at the cost of degrading the performance of lower priority clusters compared to P-UPF.

The energy efficiency of each example cluster head (cluster 1,3 and 5) in Fig. 7.4 c), Fig. 7.5 c) and Fig. 7.6 c) shown that the energy efficiency of UPF, P-UPF and MIP based gateway controls are higher than that of the random control under
Figure 7.4: Performance of cluster with high priority and low traffic load: a) throughput, b) utility, c) energy efficiency, d) end-to-end delay, e) packet drop ratio, and f) profit.
Figure 7.5: Performance of cluster with high priority and high traffic load: a) throughput, b) utility, c) energy efficiency, d) end-to-end delay, e) packet drop ratio, and f) profit.
Figure 7.6: Performance of cluster with low priority and low traffic load: a) throughput, b) utility, c) energy efficiency, d) end-to-end delay, e) packet drop ratio, and f) profit.
all packet arrival rates. The saturated energy efficiency of cluster with lower traffic load is around double of that of the cluster head with higher traffic load. This is due to the radical increase of IEEE 802.15.4 duty cycle is based on 2. Thus, the duty cycle of cluster head with heavy traffic load could be doubled of the cluster with light traffic load with slightly increase of throughput.

For the end-to-end delays of each example cluster head (cluster 1,3 and 5) are shown in Fig. 7.4 d) Fig. 7.5 d) and Fig. 7.6 d). It can be seen that while the end-to-end delay of random and UPF based controls are similar to all three cluster heads, that of the P-UPF and MIP based gateway controls increased with the increase of traffic load (cluster head 1 vs. 3) and decrease of the priority (cluster head 1 vs. 5). For cluster heads with same priority, the end-to-end delay due to the increase of traffic load with MIP based control is less than with P-UPF based control.

The packet drop ratios of each example cluster head (1,3 and 5) are shown in Fig 7.4 e) Fig 7.5 e) and Fig 7.6 e). For cluster with high priority, the proposed P-UPF and MIP based controls always have the lowest packet drop ratio. The packet drop ratio of UPF is slightly higher than that of the P-UPF and MIP based controls, but lower than that of the random control. For the cluster head with low priority, the UPF, P-UPF and MIP based controls have lower packet drop ratio than that of the random control.

7.4.2 Heterogeneous Applications

In this scenario, the simulation results for scenario where the clusters run different class applications. More specifically, the elastic applications and rate adaptive applications are chosen as example. The simulation setting is shown in Table. 7-C
as scenario 2.

The network total throughput and utility analysis are omitted due to the limit of space. The network performance in terms of network total throughput and total utility of all 5 clusters in this scenario is similar with those of the above scenario 1. This shown that the network has achieved similar total throughput and utility regardless of application type of each cluster head. The scheduling results of the proposed P-UPF and MIP based control various under the scenario where clusters run different applications, as shown in Fig. 7.7.

Fig. 7.7 a) and Fig. 7.7 b) shown a snapshot of the averaged throughput and averaged utility of each example cluster head when the data arrival rate is 80kbps. It is clear that the proposed P-UPF and MIP based control achieved the differentiation for clusters with different applications, while no differentiation can be found in the results of UPF and random controls. In addition, the P-UPF and MIP have different preference when doing the schedule. For the presented scenario, P-UPF is capable of dealing with the elastic application while MIP is capable of dealing with the rate adaptive application, especially in terms of the utility.

To further explore the difference of P-UPF and MIP based controls, Fig. 7.8 shows the cumulative density function (CDF) of end-to-end delay of each example cluster 1, cluster 3 and cluster 5. It can be seen that the end-to-end delay CDF of the cluster heads with inelastic application are similar. The range of the end-to-end delay are all within 5s to 12s for both P-UPF and MIP based control. However, for the cluster with elastic application the range of end-to-end delay distribution is wider, and the mean value of the MIP based control is smaller than that of the P-UPF based control.

The performance of example cluster 1, cluster 3 and cluster 5 are presented in
Fig. 7.7: Network performance with heterogeneous applications.

The average throughput performance of each example cluster is shown in Fig.
Figure 7.8: CDF of end-to-end delay with heterogeneous applications.

7.9 a), Fig. 7.10 a) and Fig. 7.11 a). The UPF based control and the random based control have similar throughput. The throughput of cluster with inelastic application is higher when applying P-UPF based control and the throughput of cluster with elastic application is higher when applying MIP based control.

The energy efficiency performance of each example cluster is shown in Fig. 7.9 c), Fig. 7.10 c) and Fig. 7.11 c). UPF, P-UPF and MIP based controls have higher energy efficiency compared to that of random based control. The energy efficiency of MIP based control is similar to that of UPF based control. The energy efficiency with P-UPF based control is higher than that of the other compared controls for cluster with elastic application. However, for cluster with inelastic application, the energy efficiency is slightly lower than that of MIP and UPF based controls when the arrival rate is larger than 60kbps. Similar to scenario 1, the energy efficiency of cluster head with light traffic load is doubled that of the cluster with heavy traffic load. This is due to the radical increment of the IEEE 802.15.4 duty cycle.

The end-to-end delay performance of each example cluster is shown in Fig.
Figure 7.9: Performance of cluster with inelastic application and low traffic load (cluster head 1)
Figure 7.10: Performance of cluster with inelastic application and high traffic load (cluster head 3)
Chapter 7. Gateway Control with Network Utility Optimisation

Figure 7.11: Performance of cluster with elastic application and low traffic load (cluster head 5)
7.9 d), Fig. 7.10 d) and Fig. 7.11 d). The end-to-end delay of UPF, MIP and random controls are similar for cluster with different application and same traffic load. However, with P-UPF based control, the end-to-end delay of cluster with elastic application is on average 2s higher than that of the other controls. And the end-to-end delay of the cluster with inelastic application is similar than that of the other controls. The end-to-end delay of cluster with heavy traffic load is slightly higher than that of cluster with light traffic load, if they are running same application.

The packet drop ratio performance of each example cluster is shown in Fig. 7.9 e), Fig. 7. 10 e) and Fig. 7.11 e). The UPF, P-UPF and MIP based controls have lower packet drop ratio compared to that of the random control. The P-UPF based control has the lowest packet drop ratio for clusters with inelastic application, while MIP based control has the lowest packet drop ratio for cluster with elastic application. For cluster with light traffic load, packets started to be dropped when the packet arrival rate is between 35kbps to 40kbps. For cluster with heavy traffic load, packets started to be dropped when the packet arrival rate is between 50kbps to 65kbps.

Due to the fact that the MIP based control takes the adjacent resource allocation constraint into account, MIP based control uses accumulated utility increase of multiple time slots as scheduling criteria while P-UPF takes the utility increase of each time period as scheduling criteria. The utility increase of elastic applications is much higher than that of inelastic application due to their different utility features. Thus, the P-UPF is capable of dealing with elastic applications while achieving higher energy efficiency. The MIP based control is capable of dealing with inelastic applications.
7.5 Summary

In this chapter, the gateway control optimisation problem is formulated and derived by applying Lagrangian-based duality techniques. The optimal gateway control P-UPF is proposed to achieve the long-term network utility maximisation and to achieve application differentiation among clusters. In addition, MIP based gateway control is proposed to ensure the adjacent resource allocation.

Through the simulation results and simulation analysis, it can be concluded that the proposed control framework outperforms the existing solutions and is capable of achieving long-term network utility maximisation with different applications under various scenarios.
Chapter 8

Conclusions and Future Works

This thesis proposed joint scheduling and duty cycle control framework for the IEEE 802.15.4 based hierarchical M2M networks. The proposed control framework aims to improve overall network performance, including energy efficiency, end-to-end delay, throughput and packet drop ratio for networks with heterogeneous applications.

i) The duty cycle control with Stop-and-Wait ARQ and Go-back-N ARQ. The aim of the duty cycle control is to address the trade-off between energy efficiency and end-to-end delay for IEEE 802.15.4 based networks. Simulation results shown that the optimal transmission policy can effectively reduce both energy consumption and end-to-end delay under various network traffics.

ii) The cluster head control aims to optimise the cluster utility which contains both empirical and economic components. The DP optimal cluster head control, PI based cluster head control, RA based cluster head control and Q-learning based cluster head control are proposed to address the trade-off between performance optimal and computational complexity. Simulation
results shown that the proposed cluster head control algorithms effectively addressed the challenges in terms of algorithm optimality, computational complexity, network dynamics and non-available network information, respectively.

iii) The gateway control aims to achieve the transmission schedule which optimise the long-term network utility while satisfying different application requirements among clusters. The proposed P-UPF based gateway control and MIP based gateway control for the M2M networks with heterogeneous applications. Simulation results and thorough simulation analysis shown that the proposed gateway controls outperformed the existing solutions, achieved long-term network utility maximisation, while providing application differentiate and utility provision under various scenarios.

The potential areas for future works include:

i) In this thesis, special attention is given to IEEE 802.15.4 based hierarchical M2M networks. Potential work can be explored on the network optimisation for hierarchical M2M networks which is supported by other short-range wireless technologies, such as low power Wi-Fi.

ii) The network optimisation in this thesis is based on cluster-tree networks. Taking the random deployment nature of the M2M networks into consideration, network optimisation for ad-hoc or mesh based networks is another open issue worth to be addressed.
References


[3GP11b] TR 37.868. 3GPP. Study on ran improvements for machine-type communications. Technical Report v.11.0.0 (release 11), September 2011.

[3GP11c] TS 22.368. 3GPP. Service requirements for machine-type communications (mtc). Technical Report v.11.3.0 (release 11), September 2011.


[ATN+14] Adnan Aijaz, Mati Tshangini, Mohammad Reza Nakhai, Xiaoli Chu, and Abdol-Hamid Aghvami. Energy efficient uplink resource allocation in lte networks with m2m/h2h co-existence under statistical qos guarantees. Communications, IEEE Transactions on,


[Cis] Cisco. Cisco 819 4g lte m2m gateway integrated service routers data sheet. Technical report.


[ETS] ETSI. Etsi tr 102 935: Machine-to-machine communications (m2m);
applicability of m2m architecture to smart grid networks; impact of smart grids on m2m platform. Technical report.


[HVY+09] Tian He, Pascal Vicaire, Ting Yan, Qing Cao, Gang Zhou, Lin


METIS. Mobile and wireless communications enablers for the twenty-twenty information society, eu 7th framework programme project. Technical report.


Kan Zheng, Fanglong Hu, Wenbo Wang, Wei Xiang, and Mischa Dohler. Radio resource allocation in lte-advanced cellular networks