



**OPTIMIZING ENERGY CONSUMPTION:
FROM WIRELESS SENSOR NETWORKS TO
LARGE "SMART" BUILDINGS**

BINBIN LI

Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

**BOSTON
UNIVERSITY**

BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

Dissertation

**OPTIMIZING ENERGY CONSUMPTION: FROM
WIRELESS SENSOR NETWORKS TO LARGE “SMART”
BUILDINGS**

by

BINBIN LI

B.Eng., Tsinghua University, 2004
M.Eng., Tsinghua University, 2006
M.Sc., Boston University, 2010

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2011

Approved by

First Reader

Ioannis Ch. Paschalidis, Ph.D.
Professor, Department of Electrical and Computer Engineering,
and Division of Systems Engineering

Second Reader

Christos G. Cassandras, Ph.D.
Professor, Department of Electrical and Computer Engineering,
and Division of Systems Engineering

Third Reader

David A. Castañón, Ph.D.
Professor, Department of Electrical and Computer Engineering,
and Division of Systems Engineering

Fourth Reader

Michael C. Caramanis, Ph.D.
Professor, Department of Mechanical Engineering,
and Division of Systems Engineering

Truth is much too complicated to allow anything but approximations.

John von Neumann

To Mom, Dad, and The One.

Acknowledgments

I want to express my deepest gratitude to my advisor, Prof. Ioannis (Yannis) Ch. Paschalids, for his invaluable guidance and constant support. His passion and enthusiasm towards research motivated me, his wisdom and insights inspired me, and his patience and encouragement carried me on through difficult times.

I would like to thank Prof. Christos G. Cassandras, Prof. David A. Casta  n and Prof. Michael C. Caramanis for serving in my Ph.D. committee and as my thesis readers. I benefited a lot from learning in their knowledgeable lectures to discussing during our enlightening meetings. Their valuable comments were extremely helpful for evaluating the thesis research. Part of Chapter 4 in this thesis was shaped through collaboration with Prof. Michael C. Caramanis, and he deserves special thanks.

During my five years of Ph.D. study at Boston University (BU) and living in the city with “*The Spirit of America*”, I received generous help from a number of fellow students and close friends. I would like to thank Dr. Dong Guo for not only insightful advices on my research but also taking care of me like a brother. I would like to thank all past and current members in Yannis’ group: Dr. Keyong Li, Dr. Yang Shen, Dr. Seong-Cheol Kang, Dr. Xiangdong Song, Ruomin Wu, Yin Chen, Yingwei Lin, Fuzhuo Huang, Wuyang Dai, Ronald Taylor Locke, Hanieh Mirzaei, Reza Moazzez Estanjini, and Mohamad Moghadasi for our helpful discussions. I would like to thank

my friends from Center for Information and Systems Engineering (CISE) and Department of Electrical and Computer Engineering at BU: Jing Qian, Manqi Zhao, Ke Chen, Prof. Jianfeng Mao, Dr. Xu Ning, Minyin Zhong, Chen Yao, Yiduo Zhou, Jiefu Zheng, and Na Sun for their support and good cheer. I would like to thank Dali Wang, Yu Zou, Sha Li, Linxia Ren, and Yu Zhou for their help and always having full confidence on me. I would like to thank Bo Cao for being the best apartment-mate I can ever have. I would like to thank Chikako Nakagawa Cheng for helping me with the wording and phrasing of my thesis.

I am grateful to staff members Mrs. Elizabeth Flagg and Mrs. Ruth Mason for making CISE such an ideal and enjoyable place for graduate students from all over the world and solving every problem that came across my path.

Finally, I wish to express my love to my mother Falian Yuan and my father Benjie Li without whose blessings I simply cannot be what I am. I am forever indebted to them, and this thesis is my humble tribute to their love.

**OPTIMIZING ENERGY CONSUMPTION: FROM
WIRELESS SENSOR NETWORKS TO LARGE “SMART”
BUILDINGS**

(Order No.)

BINBIN LI

Boston University, College of Engineering, 2011

Major Professor: Ioannis Ch. Paschalidis, Ph.D.,
Professor, Department of Electrical and Com-
puter Engineering, and Division of Systems En-
gineering

ABSTRACT

This dissertation addresses two problems of optimizing energy consumption in *Wireless Sensor Networks* (WSNETs) and “smart” buildings.

For the first topic, we study energy-efficient implementations of averaging/consensus algorithms in WSNETs. For static topologies we notice that a *Bidirectional Spanning Tree* (BST) is preferable in terms of convergence time. We formulate the combinatorial optimization problem of selecting such a minimal energy tree as a mixed integer linear programming problem. We then devise a semi-definite relaxation and a series of graph-based algorithms that yield energy-efficient BSTs, and establish associated bounds on the optimal cost. For dynamic topologies we consider a load-balancing algorithm which has preferable convergence time. We formulate the

problem of selecting a minimal energy interconnected network over which we can run the algorithm as a *Dynamic Programming* (DP) problem. We first consider the scenario of a large enough time horizon and show that the problem is equivalent to constructing a Minimum Spanning Tree. We then consider the scenario of a limited time horizon and employ a “rollout” heuristic that generates near-optimal solutions.

For the second topic, we develop a market-based mechanism that enables a building *Smart Microgrid Operator* (SMO) to offer regulation service reserves and meet the associated obligation of fast response to commands issued by the wholesale market *Independent System Operator* (ISO). The proposed market-based mechanism allows the SMO to control the behavior of internal loads through price signals and to provide feedback to the ISO. A regulation service reserve quantity is transacted between the SMO and the ISO for a relatively long period of time. During this period the ISO follows shorter time scale stochastic dynamics to repeatedly request from the SMO to decrease/increase its consumption. We model the operational task of selecting an optimal short time scale dynamic pricing policy as a DP that maximizes average SMO and ISO utility. We then formulate a non-linear programming static problem that provides an upper bound on the optimal utility. We study an asymptotic regime in which this upper bound is shown to be tight and the static policy provides an efficient approximation of the dynamic pricing policy.

Contents

1	Introduction	1
1.1	Wireless Sensor Networks	2
1.1.1	An Overview	2
1.1.2	Energy Conservation in WSNets	7
1.2	Consensus and Averaging	12
1.2.1	The Agreement Algorithm	13
1.2.2	Static Topologies and Convergence Results	15
1.2.3	Dynamic Topologies and Convergence Results	18
1.3	Smart Grid, Smart Buildings and Regulation Service	21
1.3.1	Smart Grid	22
1.3.2	Smart Buildings	24
1.3.3	Power Markets: Energy and Reserve Commodities	25
1.3.4	Demand Response	28
1.4	Outline and Contribution of This Dissertation	30
1.5	Notational Conventions	33
2	Energy Optimized Topologies for Averaging in Static WSNets	34
2.1	Problem Formulation	35

2.1.1	Network Model	35
2.1.2	Mixed Integer Linear Programming Model	36
2.2	An Algorithm Based on a Semi-Definite Relaxation	39
2.3	Graph-Based Algorithms	48
2.3.1	Augmented Graph Construction	48
2.3.2	Minimum Cost Flow Problem	52
2.3.3	Minimum Weight Spanning Tree Problem	57
2.3.4	Distributed Approaches	58
2.4	Numerical Experiments	61
3	Energy Optimized Topologies for Averaging in Dynamic WSNets	69
3.1	Problem Formulation	70
3.1.1	Network Model	70
3.1.2	Dynamic Programming Model	73
3.2	Large Enough Horizon Length	77
3.2.1	An MST-Based Policy	79
3.2.2	Numerical Experiments	83
3.3	Limited Horizon Length	84
3.3.1	An Algorithm Based on Rollout	84
3.3.2	Numerical Experiments	88
4	A Market-Based Mechanism for Providing Demand-Side Regulation Service Reserves	93

4.1	Problem Formulation	95
4.2	Dynamic Programming Formulation	101
4.3	Static Pricing Policy	103
4.4	Optimal Performance Upper Bound	104
4.5	Asymptotic Behavior	106
4.5.1	Many Small Loads	106
4.5.2	Energy Neutrality	112
4.5.3	Optimal Selection of R and R_h	116
4.6	Numerical Experiments	118
4.6.1	Tracking of ISO Requests	118
4.6.2	Long Term Energy Neutrality	120
4.6.3	Optimal Selection of R and R_h	126
5	Conclusion	128
5.1	Topology Control in WSNets	128
5.2	Regulation Service Reserves by Smart Buildings	130
	References	132
	Curriculum Vitae	143

List of Tables

2.1	The Steiner Problem in Network	51
2.2	Numerical Experiments on Sparse Networks.	62
2.3	Numerical Experiments on Dense Networks.	65
3.1	MST-Based Algorithm vs. DP Algorithm.	83
3.2	The Effectiveness of the Rollout Algorithm.	89
3.3	The Sub-Optimality of the Rollout Algorithm.	90
3.4	The Efficiency of the Rollout Algorithm in Sparse Networks.	91
4.1	The Arrival Rates of Internal Classes and the RS Class.	120
4.2	Penalty Function Slope Settings.	127

List of Figures

2·1	An example of augmented graph construction in a 3-real-node network.	50
2·2	An example of constructing MILP feasible solution from MCF optimal solution.	56
2·3	An example of constructing MCF feasible solution from MILP optimal solution.	57
2·4	Numerical experiments on sparse networks - effectiveness.	63
2·5	Numerical experiments on sparse networks - efficiency.	64
2·6	Numerical experiments on dense networks - effectiveness.	66
2·7	Numerical experiments on dense networks - efficiency.	67
4·1	The system model.	99
4·2	Energy consumption by internal classes and active RS requests. . . .	121
4·3	Number of active internal loads and active RS requests.	122
4·4	Energy consumption by internal classes and active RS requests. . . .	124
4·5	Number of active internal loads and active RS requests.	125
4·6	Optimal selection of R and R_h for different RS class maximal arrival rates.	127

List of Abbreviations

ADC	Analog to Digital Converter
BST	Bidirectional Spanning Tree
DP	Dynamic Programming
HVAC	Heating, Ventilating, and Air Conditioning
ISO	Independent System Operator
kbps	kilo-bits per second
LA	Load Aggregator
MAC	Medium Access Control
MCF	Minimum Cost Flow
MDP	Markov Decision Problem
MILP	Mixed Integer Linear Programming
MST	Minimum Spanning Tree
NLP	Non-Linear Programming
OSI	Open Systems Interconnection
QoS	Quality of Service
RS	Regulation Service
RSDPA	Recursive Semi-Definite Programming Algorithm
RSO	Regional System Operator
RSSI	Received Signal Strength Indication
s.t.	subject to
SDP	Semi-Definite Programming
SMO	Smart Microgrid operator
TDMA	Time Division Multiple Access
WBASNs	Wireless Body Area Sensor Networks
WSNETs	Wireless Sensor Networks

Chapter 1

Introduction

Energy, in particular electricity, is vital to ensuring a healthy economy and population. The 2001 California power shortages and the 2003 Northeast-Midwest power blackout, along with persistent high oil and natural gas prices in past decades, have highlighted the importance of energy efficiency and conservation.

This dissertation addresses the optimization of energy consumption in two research areas, *Wireless Sensor NETWORKs* (WSNETs) and large “smart” buildings.

This chapter gives an overview of WSNETs and discusses issues concerning electricity consumption in large buildings. We first highlight energy conservation issues in WSNETs and then focus on their closely related applications. In particular, we consider consensus and averaging problems. A review of some distributed averaging algorithm-s with appealing convergence results and a proposal of constructing energy efficient topologies for distributed averaging in WSNETs follow.

In the second part of this dissertation, we briefly introduce smart buildings connected to a smart grid, where WSNETs have significant application potential. After reviewing the advanced retail power markets with special interest in the real-time markets, we propose the problem of modulating electricity consumption in smart

buildings as means to offer regulation service to the power grid.

1.1 Wireless Sensor Networks

1.1.1 An Overview

A WSNET consists of a large number of spatially distributed small-size, low-cost and low-power autonomous devices, which we call “sensor nodes” or “motes”. Sensor nodes are mainly powered by batteries and equipped with sensing, data processing and wireless communication units. Despite their limited capability, individual nodes can coordinate to monitor physical or environmental conditions, process sensed data, exchange information and perform specific actions as a network.

Since their emergence in the late 1990s, WSNETs have attracted extensive attention and wide interests in both academia and industry [Akyildiz et al., 2002; Puccinelli and Haenggi, 2005]. Applications of WSNETs are mostly found in, but not limited to the following areas:

- *Environmental and Wildlife Monitoring:* The University of California at Berkeley and the College of the Atlantic deployed a network of motes on the Great Duck Island off the coast of Maine to monitor ecological habitats [Mainwaring et al., 2002]. In Princeton’s Zebranet, a dynamic sensor network is created by attaching special collars to the necks of zebras to monitor their moves and behaviors [Juang et al., 2002]. Harvard University, the University of New Hampshire, and the University of North Carolina deployed a WSNET at Volcán Tungurahua

to collect infrasonic signals during eruptions [Werner-allen et al., 2005].

- *Medical Research and Health Care:* Harvard University and Boston University Medical School’s CodeBlue project uses *Wireless Body Area Sensor Networks* (WBASNs) in rapid disaster response scenarios [Malan et al., 2004]. A flexible and efficient WBASN solution for a wide range of applications is developed in [Farella et al., 2008] and focuses on posture and activity recognition applications through practical and on-the-field testing. We note that research in WBASN is rapidly growing and has the potential to significantly impact the delivery of health care [Latré et al., 2011].
- *Military Applications:* DARPA’s self-healing minefield uses a self-organizing WSNET to respond to attacks and redistribute the mines in order to heal branches and complicate the progress of enemy troops [DARPA-ATO, 2002]. In Vanderbilt University’s PinPtr [Maroti et al., 2004], an acoustic sensor network is developed for sniper localization by detecting the muzzle blast and acoustic shock wave. The network uses the arrival times of acoustic events at different sensor nodes to estimate the sniper’s position.
- *Industrial Automation and Smart Buildings:* A two-layer WSNET is developed to monitor industrial plants [Yamaji et al., 2008]. Data measured by the lower layer are transmitted to the upper layer through an ad-hoc mesh network and redundant gateways, so that reliability can be maintained even under severe

conditions. A WSNET is deployed in a Boston University building which utilizes pairwise Received Signal Strength Indications (RSSI) of sensor nodes for indoor localization [Paschalidis and Guo, 2009]. The developed testbed has been extended to forklift tracking and scheduling in warehouses [Paschalidis et al., 2009b]. An intelligent energy-conservation system by a WSNET can monitor the usage of electric appliances in a building and help determine whether there are electric appliances that can be turned off for energy conservation [Yeh et al., 2009].

Although individual applications may feature specific characteristics and requirements, successfully designed WSNETs usually share some common features.

WNNETs have little or no infrastructure. The number of sensor nodes in a WSNET could be several orders of magnitude higher than that in ad hoc networks, and dense deployments are required to ensure coverage and connectivity. The sensor node position need not be engineered or pre-determined. This facilitates WSNETs to be randomly deployed in human inaccessible terrains. Sensor nodes operate on nonrenewable power resources and rely on a short-range, multi-hop communication paradigm to send/receive messages wirelessly. Despite the general assumption that sensor nodes are stationary, their relatively frequent breakdowns, volatile nature of wireless channel, and harsh and uncertain environmental conditions result in a dynamic network topology.

Compared with traditional wired networks, WSNETs reduce wire layout cost while

supporting mobility, enhancing network scalability and adding deployment convenience. However, WSNETs also present many challenges to researchers and engineers that cannot be adequately addressed by existing techniques. Prominent issues with WSNETs include [Akyildiz et al., 2002; Puccinelli and Haenggi, 2005; Yick et al., 2008]:

- *Network Lifetime*: Lifetime is critical to WSNETs and is primarily determined by sensor nodes' energy consumption. Non-rechargeable batteries with a limited energy budget usually power sensor nodes. Energy depletion causes sensor nodes to “die” and disconnect from the network. Some sensor nodes failure may significantly affect network coverage and connectivity, influencing the useful lifetime of the whole network. Therefore, energy conservation is a primary concern in WSNETs.
- *Quality of Service (QoS)*: Sensor nodes periodically alternate between sleep and active states following a very small duty cycle (the fraction of time nodes are active during their lifetime) and transmit/receive at very low rates during non-idle phases to minimize energy consumption. This greatly reduces the QoS of WSNETs. For example packet latency can range from a few seconds to minutes while the network throughput may vary from negligible to tens of kilo-bits per second (kbps), depending on the applications and channel properties. Successful WSNETs applications must appropriately trade off QoS with energy efficiency. [Paschalidis and Lai, 2008] considers balancing energy consumption versus rout-

ing latency optimally. [Paschalidis et al., 2009a] studies optimized scheduled Multiple Access Control (MAC) to maximize weighted WSNET throughput.

- *Robustness and Fault-tolerance:* Sensor nodes are prone to failure due to battery depletion, along with physical damage, especially when they operate in harsh environments. Human maintenance is usually impossible. WSNETs must therefore be fault-tolerant, namely, the failure of individual nodes should not endanger the overall operation of the network.
- *Self-configuration and Distributedness:* WSNETs are by nature autonomous and distributed systems, which require sensor nodes to self-organize into a network and subsequently control and self-manage to adapt to dynamic conditions. Global knowledge regarding the whole network is generally unavailable at single nodes due to the large network size and limited communicating and processing abilities. It is therefore preferred that sensor network protocols and algorithms work in a distributed manner by only utilizing local information.
- *Security and Privacy:* WSNETs are usually left unattended in open fields after deployment. This provides opportunities for an adversary to compromise sensor nodes and alter data integrity. Messages that are broadcasted over wireless channels can be easily eavesdropped, jammed or distorted. This poses new challenges to security monitoring, detection and response to malicious attacks in WSNETs. [Paschalidis and Chen, 2010] leverages large deviation techniques

of Markov models for anomaly detection with WSNETs.

1.1.2 Energy Conservation in WSNETs

A generic sensor nodes is comprised of four basic components:

1. A sensing unit consisting of one or more sensors and Analog to Digital Converters (ADCs), which monitor the physical phenomenon and perform analogously to electrical signal conversions;
2. A computing unit containing a microprocessor or microcontroller and external storage, which processes data and controls other components of the node;
3. A communicating unit consisting of a short-range transceiver, which connects the node to a network and exchanges information with neighboring nodes;
4. A power unit, which provides energy for all components of the node and essentially determines the node's lifetime.

As discussed in the previous section, the key issue in WSNETs is energy conservation, which directly influences the network lifetime. It has been shown that the communicating unit has the highest energy consumption [Shnayder et al., 2004]. Numerous research works are dedicated to this area [Anastasi et al., 2009; Dietrich and Dressler, 2009]. Generally, energy conservation with respect to communication in WSNETs can be considered from two perspectives: the node level and the network level, referred to as *power management* and *topology control* respectively. Power

management guides a single sensor node to switch between sleep and wake-up modes during pairwise communication sessions in order to save energy. Topology control considers a set of sensor nodes that form an energy efficient topology for information exchange and data routing. From the perspective of the Open Systems Interconnection (OSI) 7-layer model [Zimmermann, 1980], the main focus of power management is the physical layer and the data link layer, while that of topology control emphasizes the network layer and the application layer. We next review the related literature.

Power Management of WSNs

Power management of WSNs can be further classified into two categories. The first category runs on top of a MAC protocol. The second category is directly integrated with the MAC protocols.

Numerous sleep/wakeup schemes have been proposed and implemented as independent protocols transparent to MAC protocols. Among those of interest include STEM [Schurgers et al., 2002], PTW [Yang and Vaidya, 2004], TinyDB [Madden et al., 2005], TASK [Buonadonna et al., 2005], TAG [Madden et al., 2002], Tinkle [Hohlt and Brewer, 2006], AWP [Zheng et al., 2003], and RAW [Paruchuri et al., 2004].

STEM and PTW work in an on-demand fashion, employing a low-rate signal radio to wake up a high-rate data radio whenever communication with another node is needed. However, it incurs the additional cost of a second radio, and may lead to mismatch coverage between the two radios. Some other schemes employ a scheduled

rendez-vous pattern, where all (TinyDB, TASK) or a subset (TAG, Twinkle) of nodes periodically wake up at the same time and stay active for a while to send and receive messages. These schemes enable broadcast in WSNs, but require strict clock synchronization and cause a large number of collisions due to simultaneous transmission during active time slots. Lastly, AWP and RAW adopt asynchronous schemes where each node wakes up independently of the others while still being able to communicate with neighboring nodes. Drawbacks of these schemes include implausibility of broadcast and additional requirements such as large node density and pre-designed scheduling functions.

For power management integrated with MAC protocols, existing schemes rely on specific sleep/wakeup patterns to optimize medium access and save energy. Some well-known protocols, including TRAMA [Rajendran et al., 2003], FLAMA [Rajendran et al., 2005], and LMAC [van Hoesel and Havinga, 2004], are based on Time Division Multiple Access (TDMA), in which time is divided into frames and further slots. Each node only turns on the radio during slot(s) that is/are assigned to it in each frame and sends/receives messages. Although these protocols are inherently energy efficient, they lack flexibility and scalability to accommodate dynamics in WSNs. For example, the slot assignment scheme may fail when facing a frequently changing topology. Moreover, TDMA-based protocols require strict clock synchronization, which further narrows their applications in WSNs. In contrast, contention-based protocols, such as B-MAC [Polastre and Culler, 2004], S-MAC [Ye

et al., 2004], T-MAC [Dam and Langendoen, 2003] and D-MAC [Lu et al., 2004], are robust, scalable, and widely used in WSNETs. These contention-based protocols integrate similar sleep/wakeup schemes (scheduled rendez-vous schemes or asynchronous schemes) with channel access functionalities for energy efficiency, but still have higher energy consumption than TDMA-based schemes because of collision and retransmission. Finally, hybrid protocols, for example Z-MAC [Rhee et al., 2008], are proposed and combine TDMA-based and contention-based protocols' advantages. However, hybrid protocols are only applicable for small scale networks because of their complex implementation.

Topology Control of WSNETs

Topology control of WSNETs is objective/application dependent. It emphasizes the design of energy aware routing protocols and/or energy efficient topologies to simultaneously fulfill assigned tasks and maximize network life.

A significant amount of work has focused on finding an energy efficient path from an origin node to a specific destination node, potentially adapting to dynamic topologies. [Gomez et al., 2002] considers minimizing a link cost as a function of transmission power. [Rodoplu and Meng, 1999] presents a distributed algorithm that leverages location information to build a topology with minimum energy that communicates with a given master node. [Aslam et al., 2003] devises a max-min formulation to balance minimum transmission power routing with maximum minimum residual energy. [Wu and Cassandras, 2005] applies an optimal control method to single origin-destination

routing problem while avoiding nodes with low residual energy. [Ning and Cassandras, 2009] investigates the optimization problem in energy allocation to maximize network lifetime. From a different perspective, [Chang and Tassiulas, 2004] addresses the problem of maximizing network lifetime for a set of destinations and formulate the problem into a linear programming framework. [Nama and Mandayam, 2005] considers the problem of optimal node distribution via the joint optimization problem of optimal energy distribution and flow over information field. [Paschalidis and Wu, 2008] considers a robust maximum lifetime routing problem in the presence of uncertainty and develop a series of alternative formulations which allow trading off between the predicted lifetime and the probability it is achieved.

Energy efficient topology design sets the nodes' transmission range optimally to impose some desired structure to the communication graph required by the application. This topic is investigated in both homogeneous and non-homogenous cases, which differ in whether all nodes have equal transmission range or not [Li and Yang, 2006]. For homogenous WSNs, [Narayanaswamy et al., 2002] proposes a distributed protocol which aims at determining the minimum required common transmission range to ensure network connectivity. [Santi, 2005] analyzes the trade-off between the network's transmission range and the size of the largest connected components. For non-homogenous WSNs, [Kirov et al., 2000] studies the issue of assigning transmission range to construct a strongly connected graph, and concludes it is NP-hard. A variant of the problem with symmetric communication constraints

is discussed in [Blough et al., 2002] and is proven to remain NP-hard.

The problems considered in the first part of this dissertation are closely related to topology control in non-homogenous WSNETs. We are interested in constructing specific structured communication graphs in WSNETs **with minimum energy** and **in a distributed manner** for a class of important and interesting applications, namely consensus and averaging. We note that the problem of energy conservation in the context of a consensus/averaging algorithm has not been considered in the related literature.

1.2 Consensus and Averaging

In many applications such as environmental monitoring and military surveillance, a common function of WSNETs is the computation of an average value over the values sensed by individual nodes. With respect to the design considerations in WSNETs, distributed algorithms are more preferable for multiple reasons. A centralized process according to which a single node (a gateway) collects all measurements and computes the average requires too much data to be transported over a potentially large number of hops and is therefore energy wasteful and impractical. Moreover, such a process has to be repeated frequently often as the measurements, corresponding to an underlying physical system, change over time. Alternatively, a distributed computation of the average has the advantage of being fault-tolerant and self-organizing [Boyd et al., 2006; Dimakis et al., 2006].

Distributed averaging has been studied extensively as a problem of distributed *consensus* in multi-agent decision and control. Consensus in a general setting, including dynamic connectivity and communication delays, has been considered in [DeGroot, 1974; Tsitsiklis, 1984; Jadbabaie et al., 2003; Olfati-Saber and Murray, 2004; Blondel et al., 2005]. When the connectivity of the network does not change over time (i.e., *static networks*) [Olshevsky and Tsitsiklis, 2009] shows that a so-called *Bidirectional Spanning Tree* (BST) (a spanning tree in which all links are bidirectional) achieves a consensus running time that has the same polynomial rate of growth as a lower bound established for all topologies. When the connectivity can change (i.e., *dynamic networks*) the network must become connected infinitely often, e.g., once every B time units for some large enough B , in order to achieve consensus. [Olshevsky and Tsitsiklis, 2009] shows that an asynchronous load-balancing algorithm from [Bertsekas and Tsitsiklis, 1989] can guarantee a convergence time which is polynomial in the number of nodes and B . These promising results, which we review in the sequel, are the starting point of the work in the first part of the dissertation.

1.2.1 The Agreement Algorithm

We start by describing the agreement algorithm run by a set $\mathcal{N} = \{1, 2, \dots, N\}$ of nodes. The values stored at the nodes at time t are denoted by the vector $\mathbf{x}(t) = (x_1(t), \dots, x_N(t))$. The nodes update their values as

$$x_i(t+1) = \sum_{j=1}^N a_{ij}(t)x_j(t), \quad i = 1, \dots, N, \quad (1.1)$$

or, in matrix form $\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t)$, where $\mathbf{A}(t) = (a_{ij}(t))_{i,j=1}^N$ is a nonnegative stochastic matrix. In words, every node forms a convex combination of its value with the values of the nodes it receives messages from, assuming that $a_{ij}(t) > 0$ if and only if node j sends messages to i at time t . As is common with consensus algorithms we make the following assumption.

Assumption 1.1

There exists a positive constant α such that:

- (a) $a_{ii}(t) \geq \alpha$, for all i, t .
- (b) $a_{ij}(t) \in \{0\} \cup [\alpha, 1]$, for all i, j .
- (c) $\sum_{j=1}^N a_{ij}(t) = 1$, for all i .

The communication pattern between the nodes at time t can also be represented by a directed graph $\mathcal{G}(t) = (\mathcal{N}, \mathcal{E}(t))$, where $\mathcal{E}(t)$ is the arc set and $(j, i) \in \mathcal{E}(t)$ if $a_{ij}(t) > 0$.

Our next two assumptions require that following an arbitrary time t , and for any i, j , there is a sequence of communications through which node i will influence (directly or indirectly) the value held by node j .

Assumption 1.2

For every $t \geq 0$, the graph $\mathcal{G} = (\mathcal{N}, \cup_{s \geq t} \mathcal{E}(s))$ is strongly connected.

Assumption 1.3 (Bounded Interconnectivity Times)

There is some B such that for all k , the graph $(\mathcal{N}, \mathcal{E}(kB) \cup \mathcal{E}(kB+1) \cup \dots \cup \mathcal{E}((k+1)B))$ is strongly connected.

For the agreement algorithm, the following result is from [Tsitsiklis, 1984] and [Tsitsiklis et al., 1986].

Theorem 1.1 *Under Assumptions 1.1 and 1.3, the agreement algorithm guarantees asymptotic consensus, i.e., there exists some c , which depends on $\mathbf{x}(0)$ and the sequence of graphs $G(\cdot)$, such that $\lim_{t \rightarrow \infty} x_i(t) = c$, for all i .*

1.2.2 Static Topologies and Convergence Results

Consider now the static connectivity case [DeGroot, 1974], i.e., $\mathbf{A}(t) = \mathbf{A}$ for all t . Notice \mathbf{A} can be thought as the transition probability matrix of an (irreducible and aperiodic) Markov chain, thus, \mathbf{A}^t converges to a matrix whose rows are equal to $\pi = (\pi_1, \dots, \pi_n)$ – steady-state probability vector. Interesting special cases of the consensus algorithm include: (i) the so called *bidirectional* or *symmetric* case where communication between nodes is always bidirectional, i.e., $(i, j) \in \mathcal{E}(t)$ if and only if $(j, i) \in \mathcal{E}(t)$, and (ii) the *equal-neighbor* case where the a_{ij} ’s are equal for all $j \in \{i\} \cup \mathcal{N}_i(t)$ where $\mathcal{N}_i(t) = \{j \neq i \mid (j, i) \in \mathcal{E}(t)\}$.

We are interested in using the agreement algorithm to average the initial values $x_i(0)$, $i = 1, \dots, n$, of the nodes. To that end, we can scale the initial values by replacing $x_i(0)$ with $x_i(0)/(N\pi_i)$ in which case the consensus algorithm converges to the average $\lim_{t \rightarrow \infty} x_i(t) = \sum_{i=1}^N \pi_i x_i(0)/(N\pi_i) = (1/N) \sum_{i=1}^N x_i(0)$. We will refer to this as the *scaled averaging algorithm* and it requires a centralized computation of the steady-state probabilities which needs global connectivity information (the matrix \mathbf{A}). An approach that uses only local information and no centralized

computation is provided in [Olshevsky and Tsitsiklis, 2009] for the bidirectional, equal-neighbor case and employs two parallel runs of the consensus algorithm. The corresponding algorithm, given below, starts with initial values $\mathbf{x}(0)$ and maintains values $\mathbf{y}(t) = (y_1(t), \dots, y_N(t))$, $\mathbf{z}(t) = (z_1(t), \dots, z_N(t))$, and $\mathbf{x}(t)$ at each time t .

Algorithm 1.1 (Two Parallel Passes of the Agreement Algorithm)

- Initialize: $y_i(0) = 1/|\mathcal{N}_i(t)|$ and $z_i(0) = x_i(0)/|\mathcal{N}_i(t)|$ for all i .
- The nodes update $\mathbf{y}(t), \mathbf{z}(t)$ using the consensus algorithm as: $\mathbf{y}(t+1) = \mathbf{A}\mathbf{y}(t)$ and $\mathbf{z}(t+1) = \mathbf{A}\mathbf{z}(t)$.
- Each node i sets: $x_i(t) = z_i(t)/y_i(t)$.

Noting that in the bidirectional equal-neighbor case the transition probabilities are given by $\pi_i = |\mathcal{N}_i(t)|/(\sum_{i=1}^N |\mathcal{N}_i(t)|)$, it can be easily verified that $\lim_{t \rightarrow \infty} x_i(t) = (1/N) \sum_{i=1}^N x_i(0)$ as desired.

To characterize the rate of convergence and the convergence time, let $\lambda_1 (= 1) \geq \lambda_2 \geq \dots \geq \lambda_N$ be the eigenvalues of \mathbf{A} and define $\mathbf{x}^* = \lim_{t \rightarrow \infty} \mathbf{A}^t \mathbf{x}(0)$, $\mathcal{X} = \{\mathbf{x}^* | \forall \mathbf{x}(0)\}$. The *convergence rate* of the consensus algorithm on a graph with connectivity matrix \mathbf{A} is defined

$$\rho = \sup_{\mathbf{x}(0) \notin \mathcal{X}} \lim_{t \rightarrow \infty} \left(\frac{\|\mathbf{x}(t) - \mathbf{x}^*\|}{\|\mathbf{x}(0) - \mathbf{x}^*\|} \right)^{1/t} = \max\{|\lambda_2|, |\lambda_N|\}, \quad (1.2)$$

and the *convergence time* $T_N(\epsilon)$ is defined as

$$T_N(\epsilon) = \min \left\{ \tau \left| \frac{\|\mathbf{x}(t) - \mathbf{x}^*\|_\infty}{\|\mathbf{x}(0) - \mathbf{x}^*\|_\infty} \leq \epsilon, \forall t \geq \tau, \forall \mathbf{x}(0) \notin \mathcal{X} \right. \right\}. \quad (1.3)$$

[Olshevsky and Tsitsiklis, 2009] shows that there exist graphs for which the convergence time $T_N(\epsilon)$ is $\Omega(N^2 \log(1/\epsilon))$. Moreover, in the *bidirectional equal-neighbor* case one can achieve the same polynomial growth of the convergence time if enough arcs are deleted so that the consensus algorithm runs on a so called *bidirectional spanning tree*.

Definition 1.1 (Bidirectional Spanning Tree)

A directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is said to be a *bidirectional spanning tree* if

- (a) it is symmetric, i.e., if arc $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$;
- (b) it contains all self-arcs (i, i) ;
- (c) if we ignore the orientation of the arcs and delete self-arcs and any duplicate arcs between any two nodes then we obtain a spanning tree.

For such bidirectional spanning trees, [Olshevsky and Tsitsiklis, 2009] shows that $\rho \leq 1 - 1/(3N^2)$ and $T_N(\epsilon) = O(N^2 \log(N/\epsilon))$. It follows that Algorithm 1.1 has $O(N^2 \log(N/\epsilon))$ convergence time when run on a bidirectional spanning tree. In fact, if the nodes know the total number of nodes N , Algorithm 1.1 is not needed and the scaled averaging algorithm can reach the average using local information only. To that end, note that in a bidirectional spanning tree we have N self-arcs and $2(N - 1)$ other arcs yielding $\sum_{i=1}^N d_i = 2(N - 1) + N$. It follows that $\pi_i = d_i / \sum_{i=1}^N d_i = d_i / (3N - 2)$. Thus, each node can compute $N\pi_i$ using local information only and use it to initialize the scaled averaging algorithm.

1.2.3 Dynamic Topologies and Convergence Results

Next consider the case where communications are still bidirectional but the topology changes dynamically ([Mehyar et al., 2005; Moallemi and Roy, 2005]). [Olshevsky and Tsitsiklis, 2009] shows that the convergence time of the general agreement algorithm is not polynomially bounded, even though it is an open question whether this is the same case when restricting to symmetric graphs. When focusing on the equal-neighbor version of the agreement algorithm [Cao et al., 2005] gives an example that has exponential convergence time. In particular, if B is proportional to n , the convergence time increases faster than an exponential in N .

For the symmetric dynamic network satisfying Assumption 1.3, [Olshevsky and Tsitsiklis, 2009] proposes a variation of an old *load-balancing* algorithm from [Bertsekas and Tsitsiklis, 1989] to tackle the problem. We describe the steps that each node carries out at each time t in Algorithm 1.2. We denote the node executing the steps below as node i . Some steps refer to the neighbors of node i , which means nodes other than i that are its neighbors at the time the corresponding step is being executed.

Algorithm 1.2 (Load-Balancing Algorithm for Averaging)

For a node i in the network, if $\mathcal{N}_i(t)$ is empty, node i does nothing at time t ; else, node i carries out the following steps.

- Node i broadcasts its current value x_i to all of its neighboring nodes (every k with $k \in N_i(t)$).
- Node i finds a neighboring node j with the smallest value: $x_j = \min\{x_k : k \in$

$\mathcal{N}_i(t)\}$. If $x_i \leq x_j$, then node i does nothing further at this step. If $x_i > x_j$, then node i makes an offer of $(x_i - x_j)/2$ to j .

- If Node i does not receive any offers, it does nothing further at this step. Otherwise, it sends an acceptance to the sender of the largest offer and a rejection to all the other senders. It updates the value of x_i by adding the value of the accepted offer.
- If an acceptance arrives for the offer made by node i , node i updates x_i by subtracting the value of the offer.

To characterize the rate of convergence and the convergence time, [Olshevsky and Tsitsiklis, 2009] introduces the following “Lyapunov” function to quantify the distance of the state $\mathbf{x}(t)$ from the desired limit:

$$V(t) = \|\mathbf{x}(t) - \frac{1}{N} \sum_{i=1}^N x_i(0) \mathbf{1}\|_2^2. \quad (1.4)$$

It is shown that $V(t)$ is a monotonically non-increasing function of t when performing Algorithm 1.2. Given a sequence of graphs $G(t)$ on N nodes, and an initial vector $x(0)$, the convergence time $T_{G(\cdot)}(x(0), \epsilon)$ is defined as:

$$T_{G(\cdot)}(x(0), \epsilon) = \min\{t | V(\tau) \leq \epsilon V(0), \forall \tau \geq t\}. \quad (1.5)$$

The (worst case) convergence time, $T_N(B, \epsilon)$, is defined as the maximum value of $T_{G(\cdot)}(x(0), \epsilon)$, over all initial conditions $x(0)$, and all graph sequences $G(\cdot)$ on \mathcal{N} that satisfy Assumption 1.3 for that particular B . [Olshevsky and Tsitsiklis, 2009] shows that there exists a constant $c > 0$ such that for every N and $\epsilon > 0$, $V((k+1)B) \leq (1 - 1/(2N^3))V(kB)$, i.e., $T_N(B, \epsilon) \leq cBN^3 \log \frac{1}{\epsilon}$.

Motivated by these results, we define our research scope as **Energy Optimized**

Topologies for Distributed Averaging in WSNETs. We consider building energy efficient WSNET topologies over which we can implement these distributed averaging algorithms. We analyze scenarios of both static and dynamic WSNETs, which are distinguished by whether the communication pattern between nodes is deterministic or stochastic.

For the static scenario, we assume that pairwise communication can be successfully constructed if two incident nodes work on sufficiently large power level for support. This essentially leads to a time-invariant topology and helps implement the equal-neighbor agreement algorithm for distributed averaging. This setting allows broadcasting while a node can simultaneously send its current value to all identified neighboring nodes using a proper power level. Our work considers the optimization problem of determining each node's power level to support a bidirectional spanning tree with minimum total energy. This problem coincides with topology control in non-homogeneous WSNETs which is NP-hard. A mathematical formulation as well as methods that can generate satisfying solutions for large scale instances are desired. Moreover, to accommodate applications in WSNETs, algorithms that can be implemented in a distributed manner are preferred or even required.

For the dynamic scenario, we assume that communication between two nodes with sufficient power to talk to each other is intermittent (affected by the “state” of the channel). Hence, any packet generated by one of the nodes is received by the other node with a certain “success” probability. The load-balancing algorithm of [Olshevsky

and Tsitsiklis, 2009] is suitable for distributed averaging in this circumstance. Yet, the condition of bounded interconnectivity times must be satisfied to ensure the convergence of load-balancing algorithm. Our work considers the problem of specifying at any given point in time a pair of nodes that should attempt to communicate so as to guarantee bounded interconnectivity time while maintaining energy-efficiency. This is essentially a *Markov Decision Problem* (MDP) (provided sufficient state information) that can be cast into a *Dynamic Programming* (DP) framework. Due to the curse of dimensionality, exploiting the problem structure is a prerequisite to deriving efficient algorithms. Meanwhile, such algorithms are expected to work in a distributed manner for applications in WSNETs.

We end this section by noting that both the equal-neighbor agreement algorithm and the load-balancing algorithm for distributed averaging have been recently proposed and analyzed, and to the best of our knowledge no other work has been dedicated to design a topology for the implementation of these algorithms in WSNETs.

1.3 Smart Grid, Smart Buildings and Regulation Service

One particular class of WSNET applications with great potential lies in the emerging field of *smart grid* and *smart buildings* [Weber, 2009]. Sensors and WSNETs can be deployed at multiple locations along the power grid, from generation, transmission, distribution, to consumption [Shargal and Houseman, 2009]. Spread over the power grid, WSNETs can monitor the health of devices (generators, conductors,

transformers etc.) and detect power outages and quality disturbances. WSNETs also provide control centers with accurate condition of the grid for instant maintenance in the case of disruptions. An important operational objective of power system control centers is to maintain system stability conditions that include the requirement to balance generation and consumption in almost real time. Sensors and WSNETs provide real-time monitoring of energy consumption, detect power fluctuations and outages, and enable intelligent control for energy saving and efficiency [Siderius and Dijkstra, 2006]. Combined with other information and communication technologies, the advent of sensors and WSNETs brings smart grid and demand participation in power markets through smart buildings, which are briefly introduced below.

1.3.1 Smart Grid

Traditional grids are challenged in integrating distributed energy resources and micro-grids because of their high centralization and reliance on central power stations to function [European Commission, 2006]. The grids only support one-way power flow and communication from suppliers to consumers, providing neither information about energy consumption across the grid nor pricing incentives to balance energy consumption over time [Atkinson and Castro, 2008]. Furthermore, the grids rely on additional peak load power plants to accommodate unexpected demand increases that are highly expensive [The Climate Group and GeSI, 2009].

The smart grid emerges as a revolutionary solution to overcome these difficulties. It utilizes digital technologies to save energy, reduce costs, and increase reliability and

transparency. Its definition remains vague because it is a relatively new concept and composed of various components. A representative definition is given below:

Definition 1.2 (Smart Grid [The Climate Group and GeSI, 2009])

A “smart grid” is a set of software and hardware tools that enable generators to route power more efficiently, reducing the need for excess capacity and allowing two-way, real time information exchange with their customers for real time demand side management. It improves efficiency, energy monitoring and data capture across the power generation and transmission and distribution network.

According to this definition, the smart grid is characterized by [Weber, 2009]:

- more efficient energy routing, which leads to optimized energy usage, reduction of the need for excess capacity and increase in power quality and security;
- better monitoring and control of energy and grid components;
- two-way flow of electricity and real-time information, which enables incorporation of green energy sources, demand-side management and real-time market transactions;
- highly automated, responsive and self-healing energy network with seamless interfaces between all components of the grid.

In general, a smart grid is composed of five main components [Weber, 2009]:

1. new and advanced grid components, such as advanced conductors, power electronics, and improved electric storage components;

2. smart devices and smart metering, include a variety of sensors and smart meters;
3. integrated communication technologies, for example, wireless communication and wired ethernet;
4. computer programs for decision support and human interfaces;
5. advanced control systems.

We note that WSNets play a vital role in connecting smart devices through integrated communication technologies, and can support advanced control systems.

1.3.2 Smart Buildings

Smart buildings, or intelligent buildings, are closely linked to smart grids. Smart buildings employ a set of technologies to enhance energy-efficiency, user comfort and security of the buildings. Similar to smart grid, a variety of definitions exist in literature [Wong et al., 2005]. A representative definition is given below:

Definition 1.3 (Smart/Intelligent Buildings [Robathan, 1989])

An Intelligent Building is one that provides a productive cost-efficient environment through the optimization of four basic elements: systems, structures, services, management and the inter-relationship between them.

Based on [Sharples et al., 1999], smart buildings can be distinguished into first-, second-, and third-generation. First-generation smart buildings consist of numerous self-regulating sub-systems that operate independently of the others. Second-generation smart buildings can be controlled remotely and can facilitate some cen-

tral scheduling because they are formed by sub-systems that are network-connected. Third-generation smart buildings are capable of *learning* the status of the building, and adapt their control correspondingly.

Smart buildings are anticipated to act as intelligent components on the electric grid by modulating their consumption to accommodate the integration of distributed generation. Under *well-designed* systems, smart buildings can participate in power markets by exploiting detailed and close to real-time information about the state of the building and the external power grid. Such participation will be mutually beneficial for both buildings and the power grid. It allows buildings to reduce their net energy costs and meanwhile helps the power grid to manage peak loads, identify and reduce waste, and facilitate the integration of distributed renewable generation.

1.3.3 Power Markets: Energy and Reserve Commodities

As stated in [Savvides et al., 2011], wholesale power markets were introduced in the US in the mid 1990s [Joskow, 2006]. These markets operate over a balancing area's transmission system and clear simultaneously energy and several types of reserve requirements. Retail/distribution power markets are emerging [Joskow, 2008; Li et al., 2008] under the presence of competing retailers with granted access to the distribution network. However, the advent and development of the smart grid, especially in the past five years [SAIC, 2006; The House of Representatives, 2007; NIST, 2010; Tabors et al., 2010], leads to the expectation that advanced retail power markets will be established in the near future. Power markets are associated with various time scales

spanning from months to days to minutes. Among those of interest to WSNET applications are the *day ahead*, *adjustment* and *real-time* markets [NYISO, 2000; NYISO, 2001; Kranz et al., 2003; Ott, 2003; Bryson, 2007; Ott, 2008; Bryson, 2008; PJM, 2008]. Participation in these markets has been limited to date to centralized generators, large industrial consumers and wholesale markets. Extending participation to the demand side of the building or neighborhood level is starting to take place [PJM, 2005]. This development puts strict information requirements on buildings, expecting close to real-time information about usage patterns and load characteristics. We describe next the basic characteristics of the three power markets of interest.

The *day-ahead* market closes at noon of each day. It determines energy supply and demand, and supply of capacity reserves for each of the next 24 hours. Each market participant offers price/quantity pairs for energy and reserves. The market administrator, known as Independent or Regional System Operator (ISO/RSO), clears the market by maximizing consumers' and producers' surplus subject to energy balance, transmission, and reserve capacity requirements. The ISO schedules sufficient generation and reserves to meet demand and reserve requirements. The ISO also obtains clearing prices for energy and reserves for the purpose of charging consumers and paying generators.

Adjustment markets are similar to day-ahead markets except that they have a less than 24-hour horizon and close after the day-ahead market. A typical adjustment market is the single-period hour-ahead adjustment market.

Real-time (or close to real-time) economic dispatch markets clear energy and capacity reserves at each node or bus of the transmission system but cover a single period of 5 to 15 minutes.

The aforementioned capacity reserves are offered to the power market and are scheduled so as to meet reserve requirements at minimum cost. A scheduled reserve is equivalent to a promise to stand by and provide capacity up to a maximum quantity with an agreed upon dynamic response. Current practice has established primary, secondary, and tertiary reserves. These correspond to the provider's obligation to activate the standby capacity upon the market operator's (control center's) request with a maximum of 30 seconds, 5 minutes and 15 minutes delay, respectively. Primary and secondary reserves, commonly referred to in the United States as frequency control, and *regulation service* (RS) reserves respectively, must offer a band of up or down capacity. Upwards or downwards adjustments of power consumption are needed to balance generation and consumption of power in real time by responding to random/unpredictable fluctuations in demand, and with the increasing introduction of volatile and intermittent renewable generation, fluctuations in generation as well. In fact, whereas primary and secondary reserves secured in today's power markets equal to 0.1% and 1% of peak load respectively, they are expected to double or quadruple with the significant integration of renewables in the production mix [Makarov et al., 2009; Smith et al., 2007]. In fact, power markets that have not adjusted the reserve requirements that they secure to the presence of wind generation are forced to re-

ject (i.e., not harvest fully) a considerable portion of wind generation [Dominguez et al., 2007; Energy Reliability Council of Texas (ERCOT), 2010a; Energy Reliability Council of Texas (ERCOT), 2010b; Reliability and Operations Subcommittee (ROC), 2007]

1.3.4 Demand Response

We assume that with the advent of the smart grid, distributed loads connected at the distribution network (e.g., loads in a building or a neighborhood) can be pooled by a designated *Load Aggregator* (LA) and participate in power markets on a par basis with centralized generators. In particular we focus on RS reserves and assume that a LA will be able to buy energy on an hourly basis at the corresponding clearing price and sell RS reserves for which it will be credited at the system RS clearing market price [Ott, 2003; PJM, 2005; Ott, 2008]. These transactions take place either in the day-ahead multi-period markets or the adjustment markets described above. An ISO who procures R_h KW of RS is entitled to consider it as a stand by increment or decrement (i.e., positive or negative change) of consumption that it can utilize at will in total or in part. The ISO procures RS from many providers (generator or LAs) to satisfy the requirements needed during a given hour. The ISO may send commands to each RS provider to request that the RS provider modulates its consumption either up or down by an amount that does not exceed the total reserves, R_h , procured from the LA during an hour. These requests may arrive at inter-arrival times of 5 seconds or longer. To observe RS reserve contractual obligations, the RS provider

must deliver the requested increase or decrease in its load with a ramp rate of $R_h/5$ KW per minute. The ISO typically re-dispatches the power system in the 5 minute intervals corresponding to the real-time markets described above. At each 5 minute system dispatch the ISO schedules slower response tertiary reserves so as to reset the utilized RS reserves to their set points. As a result, although not guaranteed, the RS reserve provider's tracking of ISO commands is for all practical purposes energy neutral over the long time scale of an hour and beyond. To meet the aforementioned contractual requirements for participation into the RS reserves market, we assume that the LA has access to a smart microgrid managed by a Smart Microgrid Operator (SMO). The SMO must be capable of controlling loads through the collaboration of a cyber-physical event scheduling layer [Savvides et al., 2011], a smart microgrid with sub-metering and control sensors and actuators, and a higher *decision support and communication* layer that interacts with users of energy services in order to adapt their demand behavior to ISO requests for RS reserve usage. The lower SMO layer consists of sensing and actuation components that collect building state information and actuate so as to safely implement goals determined at the higher decision support layer and authorized by building occupants.

We define our research scope as a **Market-Based Mechanism for Providing Demand Side Regulation Service Reserves**. We focus expressly on providing the higher decision support layer with a virtual market that operates on the building side of the meter for the purpose of eliciting a collaborative response of building

occupants. More specifically, our objective is to derive an optimal SMO pricing or incentive policy towards building occupants so that they consent to the sale of RS reserves to the ISO and collaborate in meeting ISO RS utilization requirements. To the best of our knowledge, little relevant work has been published, and we are the first to propose such a market based policy for demand control aiming at the provision of RS reserves. Methodologically, related techniques have been used in pricing Internet services [Paschalidis and Tsitsiklis, 2000; Paschalidis and Liu, 2002].

1.4 Outline and Contribution of This Dissertation

The rest of this dissertation is organized as follows.

The next two chapters consider energy optimized topologies for distributed averaging in WSNs. Our contribution concerns both static and dynamic networks that are discussed in Chapters 2 and 3, respectively.

For static networks we formulate the combinatorial optimization problem of selecting power levels of the nodes to support a bidirectional spanning tree. The problem has been shown to be NP-complete. We provide a Mixed Integer Linear Programming (MILP) formulation which is tractable only for relatively small instances. We develop a Semi-Definite Programming (SDP) relaxation and a rounding scheme that can generate MILP feasible solutions. Further, we develop a series of graph-based algorithms that yield energy efficient bidirectional spanning trees, and lower and upper bounds on the optimal energy cost. Two of the proposed algorithms can be run in a

distributed manner, a feature that is particularly appealing for large WSNETs. A set of numerical results demonstrates that our graph-based algorithms obtain solutions that are quite close to optimal.

For dynamic networks, we consider the optimization problem of, at any given time point, selecting a pair of nodes that makes a communication trial to reach interconnectivity for every block of B time points. We formulate this problem as a sequential decision problem and cast it into a DP framework. In particular, we provide a finite horizon formulation with a horizon of length B and a large terminal cost that corresponds to the case when connectivity is not achieved at the end of the horizon. We first consider the regime where B is sufficiently large and the terminal cost is never paid. We establish some structural properties of the optimal policy and show that it corresponds to the construction of a Minimum Spanning Tree (MST). The MST problem can be solved in a distributed manner using an algorithm from [Gallager et al., 1983]. We then aim to address the case in which the horizon B is finite and a terminal cost cannot be ignored. The corresponding DP problem is difficult to solve. We resort to heuristics, in particular, we use a rollout algorithm [Bertsekas et al., 1997] that leverages the MST solution.

Then, we progress to the second topic of this dissertation. In Chapter 4, we develop a market-based mechanism to regulate electricity demand within a smart building and enable the SMO to offer RS reserves to the ISO. Different types of loads within the building are segregated into classes depending on their characteristics. Requests from

the SMO to increase/decrease the building's consumption is also treated as a special class. Each class is seen as requesting discrete chunks of electricity according to a stochastic arrival process for a certain stochastic period of time. For each accepted request the SMO charges a certain price, which may be different for each class. Classes respond to these prices by modulating the rate at which they request electricity. Based on a social welfare model originated from [Paschalidis and Tsitsiklis, 2000], we formulate the problem of selecting an optimal short time scale dynamic pricing policy as a stochastic dynamic program that maximizes average SMO and ISO utility over the long time scale horizon. After formulating an associated Non-Linear Programming (NLP) static pricing policy problem that provides an upper bound to the optimal dynamic policy performance, we show that the bound is asymptotically tight as the number of SMO occupants approaches infinity. This asymptotic result provides an efficient approximation of the dynamic pricing policy. Of equal importance, it allows us to optimize the long time scale decision of determining the optimal regulation service reserve quantity. We eventually verify and validate the proposed approach through a series of Monte Carlo simulations of controlled system time trajectories.

Lastly, we summarize the work of this dissertation and propose several future research directions in Chapter 5.

1.5 Notational Conventions

Throughout the paper all vectors are assumed to be column vectors. We use lower case boldface letters to denote vectors and for economy of space we write $\mathbf{x} = (x_1, \dots, x_N)$ for the column vector \mathbf{x} . \mathbf{x}' denotes the transpose of \mathbf{x} , $\|\mathbf{x}\|$ its Euclidean norm, and $\mathbf{0}$, $\mathbf{1}$ the vector of all zeros and ones, respectively. We use upper case boldface letters to denote matrices and $\mathbf{A} = (a_{ij})_{i,j=1}^N$ indicates the matrix \mathbf{A} with i, j element a_{ij} . We write \mathbf{I} for the identity matrix and $\mathbf{0}$ for the matrix of all zeros. We use $\text{diag}(\mathbf{x})$ to denote a diagonal matrix with the elements of the vector \mathbf{x} in the main diagonal and zeros elsewhere. Similarly, $\text{diag}(\mathbf{A}_1, \dots, \mathbf{A}_n)$ denotes the block diagonal matrix with matrices $\mathbf{A}_1, \dots, \mathbf{A}_n$ in the main diagonal. Finally, for any set \mathcal{X} , $|\mathcal{X}|$ denotes its cardinality.

Chapter 2

Energy Optimized Topologies for Averaging in Static WSNets

Motivated by the appealing convergence result of the equal-neighbor agreement algorithm on a bidirectional spanning tree, we investigate how to construct such a graph with minimum energy cost. We focus on static WSNets, assuming that a bidirectional communication link is guaranteed if two incident nodes work on sufficient power level to reach each other. Due to wireless communication's broadcast nature, it may be energy-beneficial to the whole network if some nodes work as *cluster heads* at relatively larger power levels to cover wider ranges while other nodes only need to talk to the nearest cluster head. We first devise a Mixed Integer Linear Programming (MILP) formulation for the optimization problem of selecting power levels of nodes to support a bidirectional spanning tree. However, the MILP is not solvable for large problem instances. Alternatively, we turn to some schemes to weaken the binary constraints, the Semi-Definite Programming (SDP) relaxation in particular, because it is quite tight and can be easily solved by some interior point methods. However, both the MILP and its SDP relaxation require global information. To accommodate WSNET applications, we develop a mechanism to transform the original network to

an augmented graph, which is suitable for a series of our proposed graph-based algorithms. The algorithms can generate near-optimal solutions and work in the desired distributed manner.

The rest of this chapter is organized as follows. In Section 2.1, we introduce the network model and formulate a MILP problem. In Section 2.2, by some relaxation technique, we convert the MILP to a SDP problem and aim at finding a sufficiently good solution based on the obtained solution and heuristic rounding schemes. In Section 2.3, we adopt the graph model containing both real and artificial nodes to characterize the WSNET. Based on the proposed graph model, we design a one-to-one correspondence strategy to obtain feasible solutions based on results from graph algorithms. We present simulation results for the proposed methods in Section 2.4.

Note that work reported in this chapter has been published in [Paschalidis and Li, 2009] and [Paschalidis and Li, 2011a].

2.1 Problem Formulation

2.1.1 Network Model

Consider a set of nodes $\mathcal{N} = \{1, 2, \dots, N\}$. Each node i has some power limit P_i^{max} ($i = 1, \dots, N$) which determines its potential neighboring nodes. We let P_{ij} denote the minimum transmit power needed by node i to reach node j . Let \mathcal{A} denote the set of *potential bidirectional* links between the nodes, that is, $\mathcal{A} = \{(i, j) \mid i < j, P_i^{max} \geq P_{ij}, P_j^{max} \geq P_{ji}\}$. As defined, \mathcal{A} contains undirected links; replacing each link (i, j)

in \mathcal{A} with the two corresponding directed links (i, j) and (j, i) we form an arc set \mathcal{A}_c . Consistent with Assumption 1.2 we assume that the graph $(\mathcal{N}, \mathcal{A}_c)$ is strongly connected.

Transmissions by a node i to node j can be heard without any additional energy expense by any other node k with $P_{ik} \leq P_{ij}$. The objective is to minimize the total transmit energy consumption over all nodes in the network while maintaining a bidirectional spanning tree. To that end, maintaining a connection between nodes i and j requires that i uses a transmit power no less than P_{ij} and j uses a transmit power no less than P_{ji} . We note that nodes consume energy when they receive as well, but this is no significantly different from the energy used while listening. In the model we assume that nodes are listening all the time so there is no need to include the corresponding energy consumption in our minimization. To reduce the latter energy, one could allow the nodes to “sleep”; see related work in [Paschalidis et al., 2007].

We note that the model we introduced can accommodate any physical layer model that determines the transmit power needed to reach node j from node i . As an example, we can take $P_{ij} = d_{ij}^\alpha$, where d_{ij} is the Euclidean distance between nodes i and j , and α is the channel loss exponent.

2.1.2 Mixed Integer Linear Programming Model

Our formulation is similar to [Das et al., 2004a] which proposed an MILP for minimum power multicasting in wireless networks with sectorized antennas.

Let $P_i \in [0, P_i^{max}]$ be the transmit power of node i . Let F_{ij} (respectively, F_{ji}), $(i, j) \in \mathcal{A}$, represent the flow from node i to node j (respectively, from node j to i). We adopt indicator variables $X_{ij} \in \{0, 1\}$, $(i, j) \in \mathcal{A}$, to show whether there is flow on link $(i, j) \in \mathcal{A}$. For those $(i, j) \notin \mathcal{A}$ we make the convention $X_{ij} = 0$; while for those $(i, j) \in \mathcal{A}$, if $F_{ij} > 0$ or $F_{ji} > 0$ then $X_{ij} = 1$.

The bidirectional topology optimization problem can be cast as a single-origin multiple-destination uncapacitated flow problem with integer constraints indicating the selection of bidirectional links. Any node can be selected as the origin of the flow; we select node 1. The corresponding flow problem involves routing $N - 1$ units of supply from node 1 (which has no demand) to all other nodes each of which has one unit of demand and zero units of supply. The MILP formulation is presented next:

$$C^{MILP} = \min \sum_{i=1}^N Y_i \quad (2.1)$$

$$\text{s.t.} \quad \sum_{\{j|(1,j) \in \mathcal{E}\}} F_{1j} = N - 1, \quad (2.2)$$

$$\sum_{\{j|(1,j) \in \mathcal{E}\}} F_{j1} = 0, \quad (2.3)$$

$$\sum_{\{j|(i,j) \in \mathcal{E} \text{ or } (j,i) \in \mathcal{E}\}} F_{ji} - \sum_{\{j|(i,j) \in \mathcal{E} \text{ or } (j,i) \in \mathcal{E}\}} F_{ij} = 1, \quad \forall i \in \mathcal{N} \setminus \{1\}, \quad (2.4)$$

$$(N - 1) \cdot X_{ij} - F_{ij} \geq 0, \quad \forall (i, j) \in \mathcal{E}, \quad (2.5)$$

$$(N - 1) \cdot X_{ij} - F_{ji} \geq 0, \quad \forall (i, j) \in \mathcal{E}, \quad (2.6)$$

$$Y_i - X_{ij} P_{ij} \geq 0, \quad \forall (i, j) \in \mathcal{E}, \quad (2.7)$$

$$Y_j - X_{ij} P_{ji} \geq 0, \quad \forall (i, j) \in \mathcal{E}, \quad (2.8)$$

$$\sum_{(i,j) \in \mathcal{E}} X_{ij} = N - 1, \quad (2.9)$$

$$X_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{E}, \quad (2.10)$$

$$F_{ij}, F_{ji} \geq 0, \quad \forall (i, j) \in \mathcal{E}, \quad (2.11)$$

$$Y_i \geq 0, \quad \forall i. \quad (2.12)$$

Some explanations are in order. The objective function (2.1) is equal to the total energy consumption by all nodes. Constraints (2.2)–(2.4) maintain flow conservation. Constraints (2.5)–(2.6) reflect our conventions for the binary variables X_{ij} setting

them to one if there is flow on the link $(i, j) \in \mathcal{A}$ in either direction. Constraint (2.9) guarantees that no redundant links are selected and the resulting graph is a spanning tree. Finally, constraints (2.7)–(2.8) ensure that each selected link from \mathcal{A} will be bidirectional and the appropriate level of transmit power will be accounted for at both nodes incident to the link. The number of integer variables X_{ij} is $|\mathcal{A}|$ and the number of continuous variables (F_{ij} 's and P_i 's) is equal to $2|\mathcal{A}| + N$.

The problem considered has been shown to be NP-complete in [Clementi et al., 1999]. [Montemanni and Gambardella, 2005] proposes a technique to reduce some redundant integer variables which may increase the speed of solving the problem. However, it is still intractable to solve large scale instances with MILP solvers. This has led to heuristics. For instance, [Das et al., 2004a] uses an incremental cost mechanism to select communication links to form a spanning tree. [Das et al., 2004b] adopts an ant colony optimization approach. [Wieselthier et al., 2000] proposes a procedure to incrementally increase transmit powers until a spanning tree is formed. A similar approach is employed in [Yuan et al., 2008] for minimum energy broadcasting with symmetric communication links.

2.2 An Algorithm Based on a Semi-Definite Relaxation

In this section we develop an SDP relaxation of the MILP and use it to bound the objective as well as obtain a feasible solution. SDP minimizes a linear function subject to a linear matrix inequality. SDP problems are convex and can be solved

in polynomial-time by interior point methods using available solvers (e.g., [Fujisawa et al., 2005]).

We start with some background and notation.

Definition 2.1 (Frobenius Inner Product)

For symmetric matrices \mathbf{M} and \mathbf{Z} in $\mathbb{R}^{m \times m}$ their Frobenius inner product is given by

$$\mathbf{M} \bullet \mathbf{Z} \triangleq \sum_{i=1}^m \sum_{j=1}^m M_{ij} Z_{ij} = \text{Tr}(\mathbf{MZ}), \quad (2.13)$$

where $\text{Tr}(\cdot)$ denotes the trace of a matrix.

Definition 2.2 (SDP Formulation [Vandenberghe and Boyd, 1996])

Given symmetric matrices $\mathbf{M}_i \in \mathbb{R}^{m \times m}$, $i = 0, \dots, n$, an SDP (in its dual form) is a problem of the following form with decision variables the elements of the symmetric matrix \mathbf{Z} :

$$\begin{aligned} \max \quad & \mathbf{M}_0 \bullet \mathbf{Z} \\ \text{s.t.} \quad & \mathbf{M}_i \bullet \mathbf{Z} = c_i, \quad i = 1, \dots, n, \\ & \mathbf{Z} \succeq 0, \end{aligned} \quad (2.14)$$

where “ $\succeq 0$ ” denotes positive semi-definiteness.

The integrality constraint (2.10) of the MILP can be rewritten as

$$X_{ij}^2 - X_{ij} = 0, \quad \forall (i, j) \in \mathcal{A}. \quad (2.15)$$

The inequalities (2.5), (2.6), (2.7), and (2.8) can be easily transformed into equal-

ities by adding slack variables

$$(N - 1) \cdot X_{ij} - F_{ij} - S_{ij} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.16)$$

$$(N - 1) \cdot X_{ij} - F_{ji} - S_{ji} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.17)$$

$$P_i - X_{ij}P_{ij} - T_{ij} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.18)$$

$$P_j - X_{ij}P_{ji} - T_{ji} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.19)$$

$$S_{ij}, S_{ji}, T_{ij}, T_{ji} \geq 0, \quad \forall (i, j) \in \mathcal{A}. \quad (2.20)$$

We will add the following (redundant in the MILP) constraint to make the relaxation tighter. Specifically, the constraint guarantees that every node is connected to some other node:

$$\sum_{\{j|(j,i) \in \mathcal{A}\}} X_{ji} + \sum_{\{j|(i,j) \in \mathcal{A}\}} X_{ij} - Q_i = 1, \quad \forall i \in \mathcal{N}, \quad (2.21)$$

$$Q_i \geq 0, \quad \forall i \in \mathcal{N}. \quad (2.22)$$

Let also adopt a special variable $V = 1$ or equivalently,

$$V^2 = 1, \quad V \geq 0. \quad (2.23)$$

Let now $\mathbf{x} = (X_{ij}; (i, j) \in \mathcal{A})$, $\tilde{\mathbf{x}} = (\mathbf{x}, V)$, $\mathbf{p} = (P_1, \dots, P_N)$, $\mathbf{f} = (F_{ij}, F_{ji}; (i, j) \in \mathcal{A})$, $\mathbf{s} = (S_{ij}, S_{ji}; (i, j) \in \mathcal{A})$, $\mathbf{t} = (T_{ij}, T_{ji}; (i, j) \in \mathcal{A})$, and $\mathbf{q} = (Q_1, \dots, Q_N)$. Let also $k(X_{ij})$ (respectively, $k(F_{ij})$, $k(T_{ij})$) denote the position of X_{ij} (respectively, F_{ij} , T_{ij}) in \mathbf{x} (respectively, \mathbf{f} , \mathbf{t}). Assume that X_{ij} 's (respectively, F_{ij} 's, T_{ij} 's) are stacked in \mathbf{x} (respectively, \mathbf{f} , \mathbf{t}) in the ascending order of subscripts, with the first digit i

dominating the second digit j . Then,

$$k(X_{ij}) = \begin{cases} \frac{1}{2}(i-1)(2N-i) + (j-i), & \text{if } i < j, \\ \frac{1}{2}(j-1)(2N-j) + (i-j), & \text{if } i > j, \end{cases} \quad (2.24)$$

$$k(F_{ij}) = k(S_{ij}) = k(T_{ij}) = \begin{cases} (i-1) \cdot (N-1) + j, & \text{if } j < i, \\ (i-1) \cdot (N-1) + j - 1, & \text{if } j > i, \end{cases} \quad (2.25)$$

Define $\mathbf{Z} = \text{diag}(\mathbf{X}, \text{diag}(\mathbf{p}), \text{diag}(\mathbf{f}), \text{diag}(\mathbf{s}), \text{diag}(\mathbf{t}), \text{diag}(\mathbf{q}))$, where

$$\begin{aligned} \mathbf{X} &= \tilde{\mathbf{x}}\tilde{\mathbf{x}}' \\ &= \begin{bmatrix} X_{12}^2 & X_{12}X_{13} & \cdots & X_{12}X_{ij} & \cdots & X_{12}X_{N-1N} & X_{12}V \\ X_{13}X_{12} & X_{13}^2 & \cdots & X_{13}X_{ij} & \cdots & X_{13}X_{N-1N} & X_{13}V \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ X_{ij}X_{12} & X_{ij}X_{13} & \cdots & X_{ij}^2 & \cdots & X_{ij}X_{N-1N} & X_{ij}V \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ X_{N-1N}X_{12} & X_{N-1N}X_{13} & \cdots & X_{N-1N}X_{ij} & \cdots & X_{N-1N}^2 & X_{N-1N}V \\ VX_{12} & VX_{13} & \cdots & VX_{ij} & \cdots & VX_{N-1N} & V^2 \end{bmatrix}. \end{aligned} \quad (2.26)$$

The explicit expression in (2.26) is for the purpose of demonstration and it holds only if $(\mathcal{N}, \mathcal{A}_c)$ is a complete graph, i.e., \mathcal{A} contains all possible links.

Next, we consider the constraints of the MILP one by one and write them in an SDP form. We also give out explicit expressions in the case of a complete network.

Eq. (2.2) can be written as

$$diag(\mathbf{0}, \mathbf{0}, \mathbf{B}_1, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = N - 1, \quad (2.27)$$

where \mathbf{B}_1 is a matrix with $(\mathbf{B}_1)_{k(F_{1j}), k(F_{1j})} = 1$ for $(1, j) \in \mathcal{A}$ and all other entries zero.

Eq. (2.3) can be written as

$$diag(\mathbf{0}, \mathbf{0}, \mathbf{B}_2, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad (2.28)$$

where \mathbf{B}_2 is a matrix with $(\mathbf{B}_2)_{k(F_{j1}), k(F_{j1})} = 1$ for $(1, j) \in \mathcal{A}$ and all other entries zero.

Eq. (2.4) can be written as

$$diag(\mathbf{0}, \mathbf{0}, \mathbf{B}_3^i, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 1, \quad \forall i \in \mathcal{N} \setminus \{1\}, \quad (2.29)$$

where \mathbf{B}_3^i is a matrix with $(\mathbf{B}_3^i)_{k(F_{ji}), k(F_{ji})} = 1$ for all j such that (j, i) or $(i, j) \in \mathcal{A}$, $(\mathbf{B}_3^i)_{k(F_{ij}), k(F_{ij})} = -1$ for all j such that (j, i) or $(i, j) \in \mathcal{A}$ and all other entries zero.

Eq. (2.15) can be written as

$$diag(\mathbf{B}_4^{ij}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.30)$$

where \mathbf{B}_4^{ij} is a matrix with $(\mathbf{B}_4^{ij})_{k(X_{ij}), |\mathcal{A}|+1} = (\mathbf{B}_4^{ij})_{|\mathcal{A}|+1, k(X_{ij})} = -1/2$,

$(\mathbf{B}_4^{ij})_{k(X_{ij}), k(X_{ij})} = 1$ and all other entries are equal to zero.

Similarly, $V^2 = 1$ (cf. (2.23)) can be written as

$$\text{diag}(\mathbf{B}_5, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 1, \quad (2.31)$$

where $(\mathbf{B}_5)_{|\mathcal{A}|+1, |\mathcal{A}|+1} = 1$ and all other entries are set to zero.

Eq. (2.16), (2.17) can be written as

$$\text{diag}(\mathbf{B}_6^{ij}, \mathbf{0}, \mathbf{B}_7^{ij}, \mathbf{B}_7^{ij}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.32)$$

$$\text{diag}(\mathbf{B}_6^{ij}, \mathbf{0}, \mathbf{B}_7^{ji}, \mathbf{B}_7^{ji}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.33)$$

where $(\mathbf{B}_6^{ij})_{k(X_{ij}), |\mathcal{A}|+1} = (\mathbf{B}_6^{ij})_{|\mathcal{A}|+1, k(X_{ij})} = \frac{1}{2}(N-1)$, $(\mathbf{B}_7^{ij})_{k(F_{ij}), k(F_{ij})} = -1$, and all other entries are equal to zero.

Eq. (2.18), (2.19) can be written as

$$\text{diag}(\mathbf{B}_8^{ij}, \mathbf{B}_9^i, \mathbf{0}, \mathbf{0}, \mathbf{B}_{10}^{ij}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.34)$$

$$\text{diag}(\mathbf{B}_8^{ji}, \mathbf{B}_9^j, \mathbf{0}, \mathbf{0}, \mathbf{B}_{10}^{ji}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A}, \quad (2.35)$$

where $(\mathbf{B}_8^{ij})_{k(X_{ij}), |\mathcal{A}|+1} = (\mathbf{B}_8^{ij})_{|\mathcal{A}|+1, k(X_{ij})} = -P_{ij}/2$, $(\mathbf{B}_9^i)_{i, i} = 1$, $(\mathbf{B}_{10}^{ij})_{k(T_{ij}), k(T_{ij})} = -1$, and all other entries are equal to zero.

Eq. (2.9) can be written as

$$\text{diag}(\mathbf{B}_{11}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = N-1, \quad (2.36)$$

where $(\mathbf{B}_{11})_{k(X_{ij}), |\mathcal{A}|+1} = (\mathbf{B}_{11})_{|\mathcal{A}|+1, k(X_{ij})} = 1/2$ for all $(i, j) \in \mathcal{A}$.

Eq. (2.21) can be written as

$$\text{diag}(\mathbf{B}_{12}^i, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{B}_{13}^i) \bullet \mathbf{Z} = 1, \quad \forall i \in \mathcal{N}, \quad (2.37)$$

where $(\mathbf{B}_{12}^i)_{k(X_{ji}), |\mathcal{A}|+1} = (\mathbf{B}_{12}^i)_{|\mathcal{A}|+1, k(X_{ji})} = 1/2$ for all $(j, i) \in \mathcal{A}$, $(\mathbf{B}_{12}^i)_{k(X_{ij}), |\mathcal{A}|+1} = (\mathbf{B}_{12}^i)_{|\mathcal{A}|+1, k(X_{ij})} = 1/2$, for all $(i, j) \in \mathcal{A}$, $(\mathbf{B}_{13}^i)_{i,i} = -1$, and all other entries zero.

Combining all the above equations we obtain the SDP relaxation as follows.

$$C^{SDP} = \min \quad \text{diag}(\mathbf{0}, \mathbf{I}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} \quad (2.38)$$

$$\text{s.t.} \quad \text{diag}(\mathbf{0}, \mathbf{0}, \mathbf{B}_1, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = N - 1,$$

$$\text{diag}(\mathbf{0}, \mathbf{0}, \mathbf{B}_2, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0,$$

$$\text{diag}(\mathbf{0}, \mathbf{0}, \mathbf{B}_3^i, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 1, \quad \forall i \in \mathcal{N} \setminus \{1\},$$

$$\text{diag}(\mathbf{B}_4^{ij}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A},$$

$$\text{diag}(\mathbf{B}_5, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 1,$$

$$\text{diag}(\mathbf{B}_6^{ij}, \mathbf{0}, \mathbf{B}_7^{ij}, \mathbf{B}_7^{ij}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A},$$

$$\text{diag}(\mathbf{B}_6^{ij}, \mathbf{0}, \mathbf{B}_7^{ji}, \mathbf{B}_7^{ji}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A},$$

$$\text{diag}(\mathbf{B}_8^{ij}, \mathbf{B}_9^i, \mathbf{0}, \mathbf{0}, \mathbf{B}_{10}^{ij}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A},$$

$$\text{diag}(\mathbf{B}_8^{ji}, \mathbf{B}_9^j, \mathbf{0}, \mathbf{0}, \mathbf{B}_{10}^{ji}, \mathbf{0}) \bullet \mathbf{Z} = 0, \quad \forall (i, j) \in \mathcal{A},$$

$$\text{diag}(\mathbf{B}_{11}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \bullet \mathbf{Z} = N - 1,$$

$$\text{diag}(\mathbf{B}_{12}^i, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{B}_{13}^i) \bullet \mathbf{Z} = 1, \quad \forall i \in \mathcal{N},$$

$$\mathbf{Z} \succeq \mathbf{0}.$$

The following result is immediate since we are relaxing the MILP.

Proposition 2.1 *It holds $C^{SDP} \leq C^{MILP}$.*

In general, the solution produced from the SDP does not provide us with integer X_{ij} 's. Thus, we will “project” the SDP solution to obtain effective MILP feasible solutions. To that end, notice that a matrix \mathbf{X} that is optimal for the SDP is positive semi-definite and has the form

$$\mathbf{X} = \begin{bmatrix} \mathbf{R}_{\mathbf{x}} & \mathbf{m}_{\mathbf{x}} \\ \mathbf{m}'_{\mathbf{x}} & 1 \end{bmatrix}. \quad (2.39)$$

It follows that $\Sigma_{\mathbf{x}} = \mathbf{R}_{\mathbf{x}} - \mathbf{m}_{\mathbf{x}}\mathbf{m}'_{\mathbf{x}} \succeq 0$ can be interpreted as a covariance matrix. This motivates us to think of the SDP as providing a probability distribution for \mathbf{x} which we can sample (and project) to obtain MILP feasible solutions. On a notational remark, we will use $\mathbf{x}(X_{ij})$ and $\mathbf{m}_{\mathbf{x}}(X_{ij})$ to denote the value of \mathbf{x} and $\mathbf{m}_{\mathbf{x}}$, respectively, corresponding to X_{ij} . We derive the following Recursive Semi-Definite Programming Algorithm (RSDPA).

Algorithm 2.1 (RSDPA)

- Step 1: Solve the SDP of (2.38) and obtain an optimal solution for \mathbf{X} in the form (2.39). Let $\mathbf{R}_{\mathbf{x}}$, $\mathbf{m}_{\mathbf{x}}$, and $\Sigma_{\mathbf{x}}$ as defined earlier. Set $\mathbf{x}^r = \mathbf{0}$.
- Step 2: (Rounding): $\forall (i, j) \in \mathcal{A}$ such that $\mathbf{x}^r(X_{ij}) = 0$ set $\mathbf{x}^r(X_{ij}) = 1$ if $\mathbf{m}_{\mathbf{x}}(X_{ij}) > \alpha$. Check whether the undirected graph $(\mathcal{N}, \{(i, j) \mid \mathbf{x}^r(X_{ij}) = 1\})$ is connected. If it is, go to Step 7.
- Step 3: (Sampling) Draw $\mathbf{x}^s = \mathbf{m}_{\mathbf{x}} + (\Sigma_{\mathbf{x}})^{1/2}\mathbf{g}$ where \mathbf{g} is a standard Gaussian

random vector $N(\mathbf{0}, \mathbf{I})$. For all $(i, j) \in \mathcal{A}$ such that $\mathbf{x}^r(X_{ij}) = 0$ set $\mathbf{x}^r(X_{ij}) = 1$ if $\mathbf{x}^s(X_{ij}) > \alpha$.

- Step 4: Check again whether the graph induced by \mathbf{x}^r is connected. If it is, go to Step 7. If the maximal iteration count has been reached go to Step 6.
- Step 5: In the SDP of (2.38) fix all X_{ij} with $\mathbf{x}^r(X_{ij}) = 1$ and resolve to obtain an optimal solution for \mathbf{X} in the form (2.39) with components \mathbf{R}_x , \mathbf{m}_x , and associated Σ_x . Go to Step 2.
- Step 6 (Connectivity): Reduce $\alpha := \beta\alpha$. Redraw $\mathbf{x}^s = \mathbf{m}_x + (\Sigma_x)^{1/2}\mathbf{g}$ where $\mathbf{g} \sim N(\mathbf{0}, \mathbf{I})$. For all $(i, j) \in \mathcal{A}$ such that $\mathbf{x}^r(X_{ij}) = 0$ set $\mathbf{x}^r(X_{ij}) = 1$ if $\mathbf{x}^s(X_{ij}) > \alpha$. If the updated \mathbf{x}^r corresponds to a connected graph, go to Step 7; otherwise, repeat Step 6.
- Step 7 (Link Trimming): Sort all $(i, j) \in \mathcal{A}$ such that $\mathbf{x}^r(X_{ij}) = 1$ in descending order of the power needed to support the link, i.e., $P_{ij} + P_{ji}$. Examine each link (i, j) in this list in descending order. If (i, j) can be removed without disconnecting the network then set $\mathbf{x}^r(X_{ij}) = 0$.
- Step 8: Output \mathbf{x}^r .

The RSDPA yields a feasible solution \mathbf{x}^r of the MILP. It uses the following parameters: a rounding threshold α ($0 < \alpha \leq 1$), a shrinking factor β ($0 < \beta < 1$), and a maximum number of iterations. These parameters can be tuned to improve performance.

Note that the algorithm produces an integer solution \mathbf{x}^r that specifies which links in \mathcal{A} are selected. Due to Step 7, \mathbf{x}^r induces a bidirectional spanning tree with $N - 1$ bidirectional links. To evaluate the cost of this tree, for each node $i \in \mathcal{N}$ set $P_i^r = \max\{P_{ij} \mid \mathbf{x}^r(X_{ij}) = 1 \text{ or } \mathbf{x}^r(X_{ji}) = 1\}$. The total energy cost of \mathbf{x}^r is:

$$C^{RSDP} = \sum_{i=1}^N P_i^r, \quad (2.40)$$

and $C^{MILP} \leq C^{RSDP}$.

We note that C^{RSDP} is the energy cost of building the bidirectional spanning tree as well as applying one iteration of the agreement algorithm 1.1.

2.3 Graph-Based Algorithms

Next we devise a series of graph-based algorithms. By construction, all these algorithms provide upper bounds to the optimal MILP cost and in some cases a lower bound as well. The graph we will consider is an augmented graph that “transfers” the energy costs from the nodes to links so that suitable network flow algorithms can be employed.

2.3.1 Augmented Graph Construction

Consider the set of “real” nodes \mathcal{N} . For each $i \in \mathcal{N}$, we define $\mathfrak{N}(i) = \{j | j \in \mathcal{N}, (i, j) \text{ or } (j, i) \in \mathcal{A}\}$ as the potential neighboring nodes of i . For each real node i we add a set of $|\mathfrak{N}(i)|$ artificial nodes, each one corresponding to a different potential neighbor of i . Each artificial node corresponds to a different power level that node i may work on. The detailed augmented graph construction is as follows. We will denote this graph by $\mathcal{G}_a = (\mathcal{N} \cup \mathcal{V}, \mathcal{E}_a)$ where \mathcal{V} denotes the set of artificial nodes.

Algorithm 2.2 (Construction of Augmented Graph)

- **Step 1:** For each ordered pair (i, j) of nodes $i, j \in \mathcal{N}$ such that $j \neq i$ and $P_{ij} \leq P_i^{max}$ create an artificial node v associated with node i and assign to v the tag $\mathbf{T}_v = (i, j, P_{ij})$. Add an undirected link from i to v with cost $\mathbf{T}_v(3) = P_{ij}$.

- **Step 2:** Consider each pair of artificial nodes v_1 and v_2 with tags $\mathbf{T}_{v_1} = (i_1, j_1, P_{i_1 j_1})$ and $\mathbf{T}_{v_2} = (i_2, j_2, P_{i_2 j_2})$ such that $v_1 \neq v_2$ and $i_1 \neq i_2$. If $P_{i_1 j_1} = \mathbf{T}_{v_1}(3) \geq P_{i_1 i_2}$ and $P_{i_2 j_2} = \mathbf{T}_{v_2}(3) \geq P_{i_2 i_1}$ then add an undirected link (v_1, v_2) with cost $w(\mathbf{T}_{v_1}(3) + \mathbf{T}_{v_2}(3))$, where w is a very small positive constant.

The tag of an artificial node v indicates that v can act as a relay for data sent from i to j , which requires power of at least P_{ij} at node i . The augmented graph contains links between a node and its artificial nodes and also links between artificial nodes corresponding to potential bidirectional links between real nodes. We assign a small cost to these latter links to avoid degenerate solutions in the algorithms we employ. From the graph \mathcal{G}_a we construct a directed graph $\tilde{\mathcal{G}}_a = (\mathcal{N} \cup \mathcal{V}, \mathcal{A}_a)$ such that for every undirected link $(i, j) \in \mathcal{E}_a$, \mathcal{A}_a contains both directed links (i, j) and (j, i) , each with cost the same as the cost of the undirected link (i, j) .

An example of augmented graph construction in a network consisting of 3 real nodes is shown in Fig. 2.1. Every node can work at two different power levels - a higher level, at which it can send messages to both two neighboring nodes, and a lower level, at which it can only communicate with the closer neighboring node. The real lines connect every real node with its artificial nodes and the weights of them correspond to the associated power levels. Take Node 1 for example; it can either work at a power level of 5 and reach both Node 2 and 3, or transmit at a power level of 3 and only talk to Node 3. The dotted lines connect all possible communication pairs and each one is assigned a small positive weight to avoid zero cost loops. In Fig. 2.1, $w = 0.01$. For example, suppose Node 1, 2 and 3 work at power levels of 3,

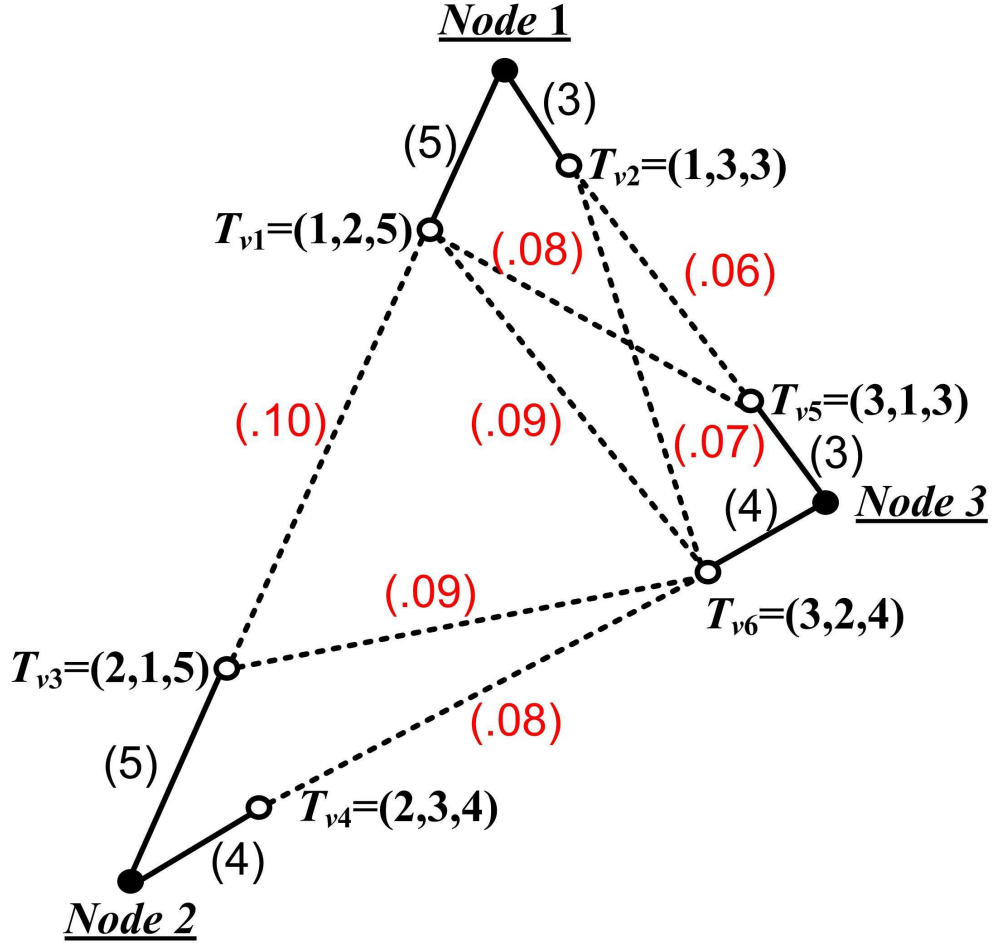


Figure 2.1: An example of augmented graph construction in a 3-real-node network.

4, and 4, respectively. The bidirectional communication pairs that can be supported are $(1 \leftrightarrow 3)$ and $(2 \leftrightarrow 3)$. We draw dotted lines to connect T_{v2} with T_{v6} , and T_{v4} with T_{v6} , and set the weights as 0.07 and 0.08. Following the same procedure, other dotted lines can be added and all possible communication patterns among real nodes are presented in the augmented graph.

Table 2.1: The Steiner Problem in Network

-
- GIVEN: An undirected network $\mathcal{G} = (\mathcal{N} \cup \mathcal{V}, \mathcal{E}, c)$ where $c : \mathcal{E} \rightarrow R$ is an edge weight function, and a non-empty set \mathcal{N} of terminals.
 - FIND: A subnetwork $T_{\mathcal{G}}(\mathcal{N})$ of \mathcal{G} such that:
 - there is a path between every pair of terminals,
 - total weight $|T_{\mathcal{G}}(\mathcal{N})| = \sum_{e_l \in T_{\mathcal{G}}(\mathcal{N})} c(e_l)$ is minimized.
-

It should be pointed out the original MILP problem (2.1) is equivalent to the *Steiner tree problem* [Huang et al., 1992] in the augmented graph.

Generally, the Steiner problem [Huang et al., 1992] in a network can be formulated as in Tab. 2.1.

The vertices in \mathcal{V} are called non-terminals. Non-terminals that end up in $T_{\mathcal{G}}(\mathcal{N})$ are called Steiner vertices. If \mathcal{G} is connected and all edges in \mathcal{G} have positive weight, $|T_{\mathcal{G}}(\mathcal{N})|$ must be a tree. The problem is therefore often referred in the literature as the *Steiner tree problem*.

In the augmented graph, real nodes are viewed as terminals and artificial nodes correspond to non-terminals. The solution for the original MILP problem is equivalent to the optimal Steiner tree with the same number of real nodes and artificial nodes. In general, the Steiner tree problem is NP-complete.

2.3.2 Minimum Cost Flow Problem

In this subsection, we will apply a minimum cost flow (MCF) algorithm to the (directed) augmented graph $\tilde{\mathcal{G}}_a$. Then, based on the optimal solution, we will construct a bidirectional spanning tree for the original graph \mathcal{G} and derive bounds on the optimal value of the MILP.

We take one real node in \mathcal{N} , say node s , as the source node having a supply of $N - 1$ units. Every other real node in $\mathcal{N} \setminus \{s\}$ has a demand of 1 unit. All the artificial nodes in \mathcal{V} have zero supply and demand. With these demands and supplies we solve the uncapacitated MCF problem on graph $\tilde{\mathcal{G}}_a$ (see [Bertsekas and Tsitsiklis, 1997]). This is the linear programming (LP) problem of minimizing the total cost of shipping flows from the nodes with supplies to the nodes with demands, where arc costs indicate cost per unit of flow. Such an LP can be solved by standard LP methods or special purpose methods (e.g., network simplex) that exploit its special structure. Let C_s^{MCF} denote its optimal value and $\mathbf{f}_s^{MCF} = (F_{ij}; \forall (i, j) \in \mathcal{A}_a)$ denote an optimal solution where F_{ij} is the optimal flow on arc $(i, j) \in \mathcal{A}_a$. Clearly, we can solve N different MCF problems depending on the selection of the source node $s = 1, \dots, N$.

Given the solution of any of the above flow problems, we can construct a bidirectional spanning tree for the original graph \mathcal{G} . The procedure is described as follows. It builds an undirected graph $\mathcal{G}_{BST,s}^{MCF} = (\mathcal{N}, \mathcal{E}_{BST,s}^{MCF})$ which is a bidirectional spanning tree of \mathcal{G} .

Algorithm 2.3**(Construction a Bidirectional Spanning Tree from the MCF Solution)**

- **Step 1:** Solve the MCF problem with source s . Initialize $\mathcal{E}_{BST,s}^{MCF} = \emptyset$. For any pair (v, u) , $v, u \in \mathcal{V}$, such that $\mathbf{f}_s^{MCF}(v, u) > 0$ or $\mathbf{f}_s^{MCF}(u, v) > 0$ add the link $(\min\{i, j\}, \max\{i, j\})$ to $\mathcal{E}_{BST,s}^{MCF}$ where i and j are the real nodes corresponding to v and u , respectively.
- **Step 2:** Sort all $(i, j) \in \mathcal{E}_{BST,s}^{MCF}$ in descending order of the power needed to support the link, i.e., $P_{ij} + P_{ji}$. Examine each link (i, j) in this list in descending order. If (i, j) can be removed without disconnecting the network then remove (i, j) from $\mathcal{E}_{BST,s}^{MCF}$.
- **Step 3:** Set the power at i as $P_i = \max\{P_{ij} \mid (i, j) \text{ or } (j, i) \in \mathcal{E}_{BST,s}^{MCF}\}$. Let $C_{BST,s}^{MCF} = \sum_{i \in \mathcal{N}} P_i$.

In Step 1 we identify all links between artificial nodes with a positive flow. Note that a real node i may have an artificial node v with a link (u, v) (or (v, u)) selected in Step 1 but with no flow in the (i, v) or the (v, i) link. This may happen if v serves as a relay to node u . Step 2 is similar to Step 7 of the RSDPA and eliminates redundant links to end up with a spanning tree ($N - 1$ links). Note that the solution to the MCF problem guarantees that we will end up with a connected network since it satisfies flow conservation and ships flow from the source s to every real node in $\mathcal{N} \setminus \{s\}$. The MCF problem also provides us with bounds on the optimal objective value C^{MILP} of the MILP.

Theorem 2.2 *As $w \rightarrow 0$ it holds*

$$\frac{N}{2(N-1)} \min_s C_s^{MCF} \leq \frac{1}{2(N-1)} \sum_{s=1}^N C_s^{MCF} \leq C^{MILP} \leq \min_s C_{BST,s}^{MCF}. \quad (2.41)$$

Proof: The far right inequality is immediate since the algorithm produces a bidirectional spanning tree, thus forming a feasible solution to the MILP. The far left inequality is also obvious.

To establish the middle inequality we will start from an optimal solution of the MILP and construct a feasible solution for the MCF problem. Let $\{X_{ij}\}$, $\{F_{ij}\}$ and $\{P_i\}$ form an optimal solution of the MILP. For each real node $i \in \mathcal{N}$ add a virtual node v_i with tag $\mathbf{T}_{v_i} = (i, j, Y_i)$, where $j = \arg \max_j \{P_{ij} | X_{ij} = 1 \text{ or } X_{ji} = 1\}$. Then, introduce the directed arcs (i, v_i) and (v_i, i) with arc weights P_i . For any two real nodes $i_1, i_2 \in \mathcal{N}$, if $X_{i_1 i_2} = 1$, then introduce the directed arcs (v_{i_1}, v_{i_2}) and (v_{i_2}, v_{i_1}) with arc weights $w(\mathbf{T}_{v_{i_1}}(3) + \mathbf{T}_{v_{i_2}}(3))$. Note that the graph we have constructed is a bidirectional spanning tree because it has been constructed based on the solution of the MILP.

Fix now some node $s \in \mathcal{N}$ and designate it as a source node. Set the flows on the graph constructed above in order to ship $N - 1$ units of supply from s to each other node which should receive 1 unit of flow. The resulting flow vector is a feasible solution of the MCF problem. Notice that the flow on any arc (v_{i_1}, v_{i_2}) between artificial nodes can not be greater than $N - 1$. Denote the largest weight of such an arc as $w \max_{i_1, i_2} (\mathbf{T}_{v_{i_1}}(3) + \mathbf{T}_{v_{i_2}}(3)) = w \mathbf{T}_{\max}$. Then, the flow cost on (v_{i_1}, v_{i_2}) must be no more than $w(N - 1) \mathbf{T}_{\max}$. Since there are $(N - 1)$ arcs between artificial nodes with positive flows, the flow cost on these arcs can not be larger than $w(N - 1)^2 \mathbf{T}_{\max}$.

The feasibility of the constructed feasible solution for the MCF problem yields

$$\begin{aligned} C_s^{MCF} &\leq \sum_{i=1, i \neq s}^N P_i + P_s(N-1) + w(N-1)^2 \mathbf{T}_{\max} \\ &= C^{MILP} + (N-2) \cdot P_s + w(N-1)^2 \mathbf{T}_{\max}. \end{aligned}$$

Adding the above inequalities for $s = 1, \dots, N$ we have

$$\sum_{s=1}^N C_s^{MCF} \leq 2(N-1)C^{MILP} + wN(N-1)^2 \mathbf{T}_{\max}.$$

Taking $w \rightarrow 0$ we arrive at the desired result. ■

To demonstrate the constructive methods introduced in the proof, we take some examples and show how to derive a MILP feasible solution from the MCF optimal solution, and vice versa.

As shown in Fig. 2-2, assume that Node 1 has a supply of 2 units and Node 2 and 3 each has a demand of 1 unit. Weights are labeled along links. In the left figure, we solve the MCF problem, mark selected links by directed arrows (blue in color print), and denote optimal flows on these selected links in brackets. The optimal MCF solution requires Node 1, 2 and 3 to work at power levels of 3, 4, and 4, respectively, and a corresponding MILP feasible solution is obtained as in the right figure.

The construction of the MCF feasible solution based on the MILP optimal solution is depicted in Fig. 2-3. Assume the optimal MILP solution maintains a tree as shown in the left figure. Notice that each node works at a certain power level, which corresponds to a *unique* artificial node in the augmented graph. For each real node,

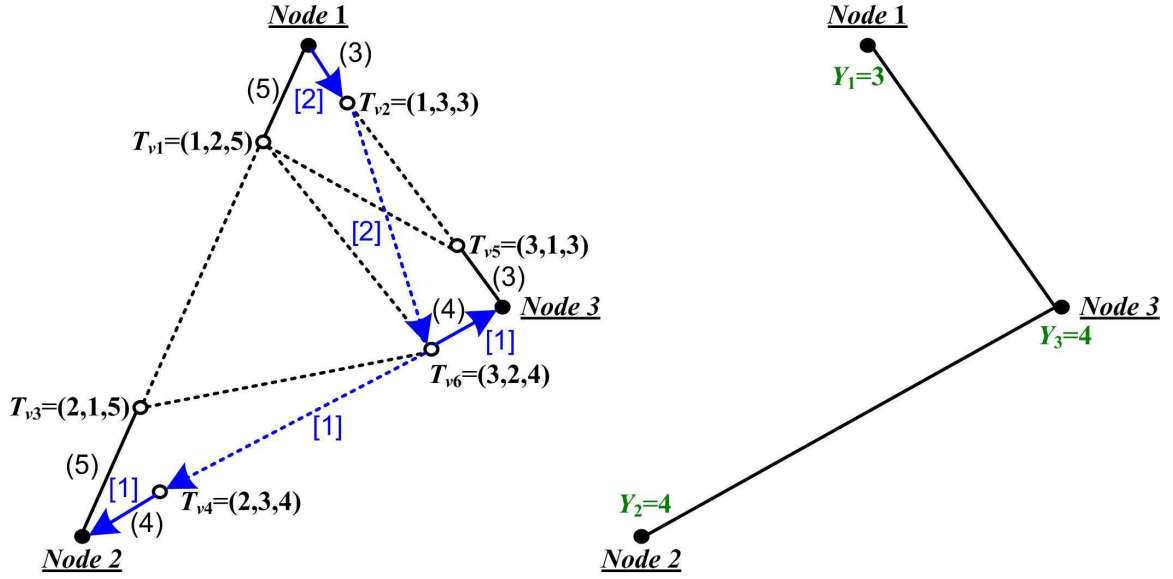


Figure 2.2: An example of constructing MILP feasible solution from MCF optimal solution.

only keep the unique artificial node and remove all other artificial nodes and incident links. The weight of the link between each real node and its unique artificial node is essentially the power level the real node works at. Map links to the augmented graph and maintain the same topology among artificial nodes. For example, in the left figure, there is a link between real node n_1 and real node n_8 , and we draw a link between artificial node v_1 and artificial node v_8 in the right figure. Designate the flows on the outgoing link from the source node, say real node n_1 , to its artificial node as $N - 1$, which is 7 in Fig. 2.3, and designate the flow on incoming link from each artificial node to its corresponding real node as 1. As $w \rightarrow 0$, the weights on links among artificial nodes are negligible. Due to that the network in the right figure is connected, and one can arbitrarily dispatch flows among artificial nodes as long as

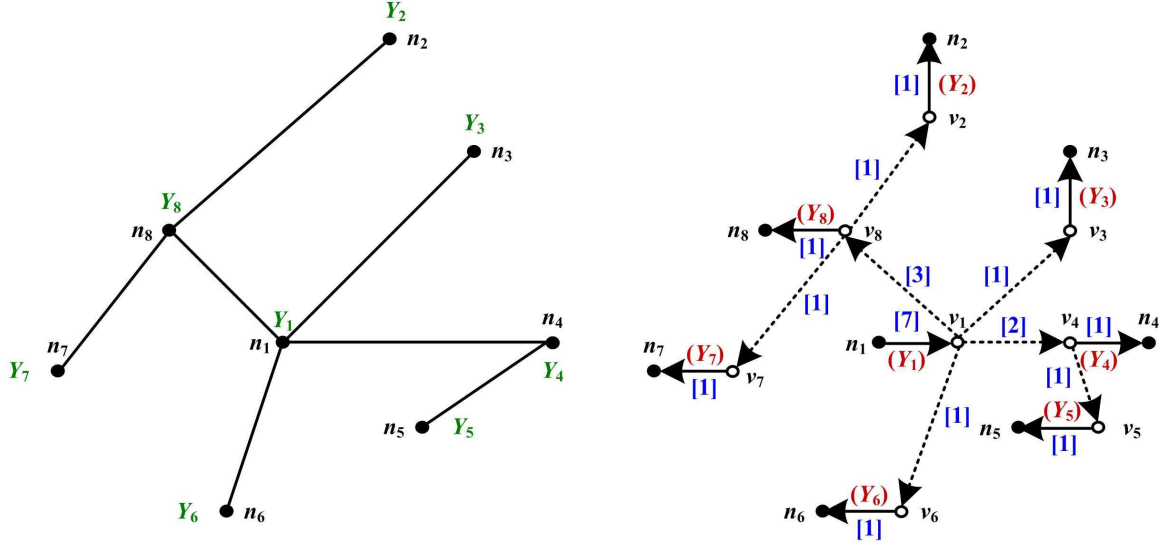


Figure 2.3: An example of constructing MCF feasible solution from MILP optimal solution.

flow conservations are satisfied. Therefore, a feasible solution for the MCF problem is obtained.

2.3.3 Minimum Weight Spanning Tree Problem

Similarly, based on a minimum weight spanning tree (MST) on the undirected augmented graph $\mathcal{G}_a = (\mathcal{N} \cup \mathcal{V}, \mathcal{E}_a)$, we can construct a bidirectional spanning tree for the original graph \mathcal{G} , see Algorithm 2.4.

Algorithm 2.4

(Constructing a Bidirectional Spanning Tree by Solving an MST Problem on the Augmented Graph \mathcal{G}_a)

- Step 1: Construct an MST of \mathcal{G}_a .
- Step 2: Remove all artificial nodes which are leaf nodes in the tree and their incident arcs until no artificial node which is a leaf node remains.

- Step 3: The resulting graph contains a path between any two real nodes, potentially through artificial nodes. Remove all artificial nodes but maintain the exact same path between any two real nodes (going through the same set of real nodes as before).
- Step 4: The resulting undirected graph is a spanning tree $\mathcal{G}^{MST} = (\mathcal{N}, \mathcal{E}^{MST})$. Set the power level of node i as $P_i = \max\{P_{ij} \mid (i, j) \in \mathcal{E}^{MST}\}$. Let $C^{MST} = \sum_{i \in \mathcal{N}} P_i$.

Since \mathcal{G}^{MST} provides a feasible bidirectional spanning tree for the original graph \mathcal{G} , $C^{MILP} \leq C^{RSDP}$.

2.3.4 Distributed Approaches

The approaches we detailed in the previous sections yields a bidirectional spanning tree and bounds on the optimal performance. It is though a centralized approach as both the standard MCF solution methods and Step 2 of Algorithm 2.3 require centralized computations. Step 1 of Algorithm 2.4 also needs global information if traditional MST algorithms, such as Kruskal's algorithm [Kruskal, 1956] and Prim's algorithm [Prim, 1957], are applied. In this subsection we discuss how to construct a bidirectional spanning tree in a distributed manner.

Minimum Cost Flow Based Distributed Algorithm

The first observation is that the MCF problem we are solving is a single-source-multiple-destination problem. It follows that optimal flows are shipped along shortest paths from the source to each destination. As a result, an alternative way for solving the problem is to compute these shortest paths. To that end, we can use the

distributed asynchronous Bellman-Ford algorithm [Bertsekas and Gallager, 1992] on the graph $\tilde{\mathcal{G}}_a$. Note that based on our connectivity assumption and the way $\tilde{\mathcal{G}}_a$ is constructed, every link is bidirectional, the graph is strongly connected, and every link has a positive cost which implies that all cycles have a positive cost. Under these conditions, the asynchronous Bellman-Ford algorithm converges. Notice that in the worst case, the distributed asynchronous Bellman-Ford algorithm may require an excessive number of iterations to terminate [Bertsekas and Gallager, 1992]. Alternatively, some other distributed algorithms [Frederickson, 1985] based on Dijkstra's algorithm can be employed to find the single source shortest paths.

The second observation deals with the distributed implementation of Step 2 in Algorithm 2.3. Examine that algorithm and note that after Step 1, the undirected graph $\mathcal{G}_{BST,s}^{MCF} = (\mathcal{N}, \mathcal{E}_{BST,s}^{MCF})$ is connected because it is constructed from the MCF solution which ships flow to every node from the source node s . (One can easily construct examples, where $\mathcal{G}_{BST,s}^{MCF}$ contains more than $N - 1$ links.) We can now assign to each link $(i, j) \in \mathcal{E}_{BST,s}^{MCF}$ a weight equal to $P_{ij} + P_{ji}$ which is the power needed to support the link. Then we can run a distributed algorithm proposed in [Gallager et al., 1983] to construct a MST. The resulting spanning tree is a bidirectional spanning tree whose overall energy cost can be evaluated as in Step 3 of Algorithm 2.3.

Notice that for the distributed graph-based algorithms, one incurs some communication overhead for finding the bidirectional spanning tree. We next assess the associated energy cost.

In the distributed approach we presented above, the total numbers of messages exchanged by the shortest path algorithm is $O((|\mathcal{N}| + |\mathcal{V}|)^{\frac{5}{3}})$ [Frederickson, 1985]. Computing the minimum spanning tree requires $O(|\mathcal{E}_{BST,s}^{MCF}| + |\mathcal{N}| \log |\mathcal{N}|)$ messages ([Gallager et al., 1983]). In the case of a planar and non-dense network, we have $|\mathcal{V}| = O(|\mathcal{N}|)$ and $|\mathcal{E}_{BST,s}^{MCF}| = O(|\mathcal{N}|)$ and the total number of messages needed is $O(|\mathcal{N}|^{\frac{5}{3}})$. The corresponding energy cost is $O(|\mathcal{N}|^{\frac{5}{3}}) \cdot P_{\max} = O(|\mathcal{N}|^{\frac{5}{3}})$, where P_{\max} denotes the maximal energy cost for one message to be transmitted.

Once the sub-optimal bidirectional spanning tree is found, the energy cost of building the topology and applying one iteration of the agreement algorithm 1.1 is $C_{BST,s}^{MCF}$.

Minimum Weight Spanning Tree Based Distributed Algorithm

A similar distributed approach can be derived by applying the algorithm in [Gallager et al., 1983] to obtain the MST on \mathcal{G}_a in Step 1 of Algorithm 2.4.

Since only the distributed minimum spanning tree algorithm is employed in Algorithm 2.4, the total energy cost of finding and constructing the bidirectional spanning tree topology is $O(|\mathcal{E}_a| + (|\mathcal{N}| + |\mathcal{V}|) \log(|\mathcal{N}| + |\mathcal{V}|)) = O(|\mathcal{N}| \log |\mathcal{N}|)$, in the case of $|\mathcal{E}_a| = O(|\mathcal{N}|)$ and $|\mathcal{V}| = O(|\mathcal{N}|)$. Similarly, after finding the sub-optimal bidirectional spanning tree, the energy cost of building the topology and applying one iteration of the agreement algorithm 1.1 is C^{MCT} .

2.4 Numerical Experiments

We generate networks by uniformly scattering N nodes on a 10×10 square. We assume that the minimum power required by a node to reach another is d^2 , where d is their distance. We set the maximum power of each node to $10^2\lambda$, where λ is a parameter we can tune. Small values of λ induce a “sparse” network where a large λ induces a “dense” network and implies many potential bidirectional links. In the results we present we take $\lambda = 0.2$ for sparse networks and $\lambda = 0.4$ for dense networks.

Tables 2.2 and 2.3 report our results for different instances (N ranging from 10 to 50) corresponding to sparse and dense networks, respectively. Fig. 2.4, 2.5 (sparse networks), and Fig. 2.6, 2.7 (dense networks), provide direct visual comparison of the results.

In the tables, MILP denotes the optimal value of the MILP in (2.1) which was solved using the CPLEX solver. The larger instances were not possible to solve in a reasonable amount of time. RT denotes the running time for each algorithm we compare. All algorithms were run on a computer with a Ubuntu-8.04-OS, 2GB of memory, and an Intel-XEON-2.00GHz CPU. SDP denotes the optimal value of the SDP relaxation given in (2.38) which is solved using the SDPA solver. The “Ratio” rows compute the ratio of the entries in the immediately preceding row over the MILP optimal cost. RSDP UB denotes the cost of the bidirectional spanning tree obtained from Algorithm 2.1. In that algorithm we set the maximum number of iterations to 4, and use $\alpha = 0.5$, and $\beta = 0.9$. MCF LB and UB denote the values obtained from

Table 2.2: Numerical Experiments on Sparse Networks.

Number of Nodes	10	20	30	40	50
MILP	49.32	62.77	81.19	NA	NA
MILP Running Time (secs)	<1	66	59792		
SDP	23.85	18.66	24.75	25.69	28.42
Ratio	0.48	0.30	0.30		
RSDP UB	63.98	82.59	116.08	126.27	95.91
Ratio	1.30	1.32	1.43		
RSDP Running Time (secs)	7	178	765	5519	24454
MCF LB	25.99	24.99	38.57	49.52	48.39
Ratio	0.53	0.40	0.48		
MCF UB	49.32	64.05	87.44	86.79	92.08
Ratio	1.00	1.02	1.08		
MCF Running Time (secs)	<1	2	6	32	120
MST UB	51.24	67.26	84.96	87.15	89.77
Ratio	1.04	1.07	1.05		
MST Running Time (secs)	<1	<1	1	11	154

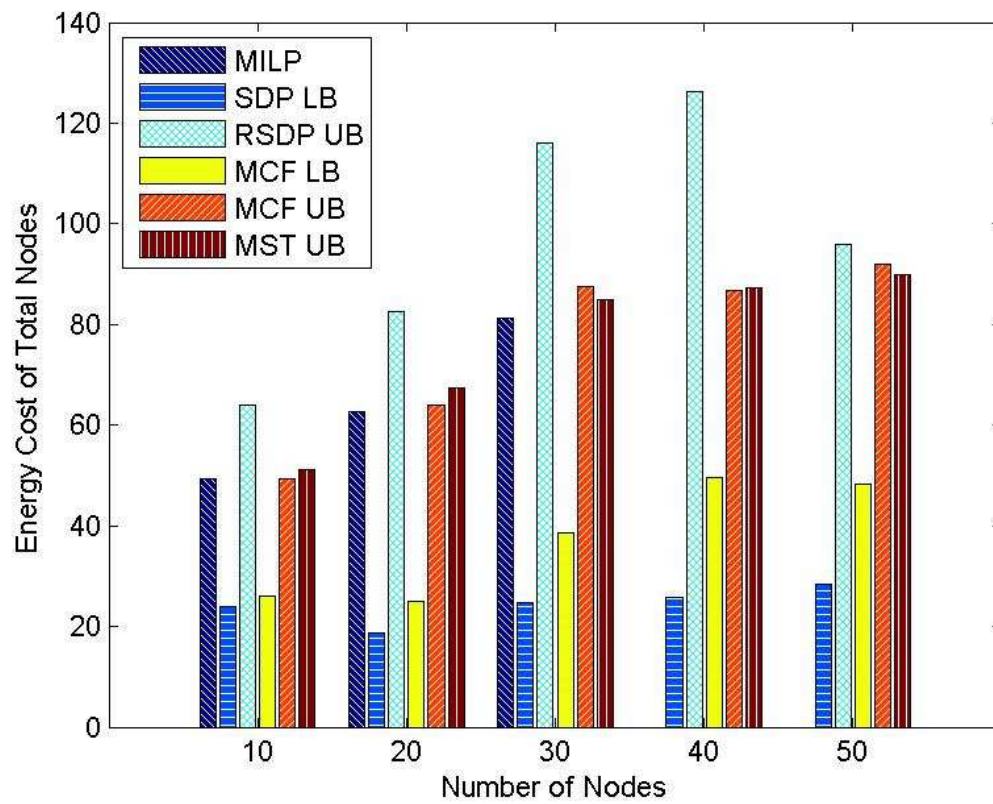


Figure 2-4: Numerical experiments on sparse networks - effectiveness.

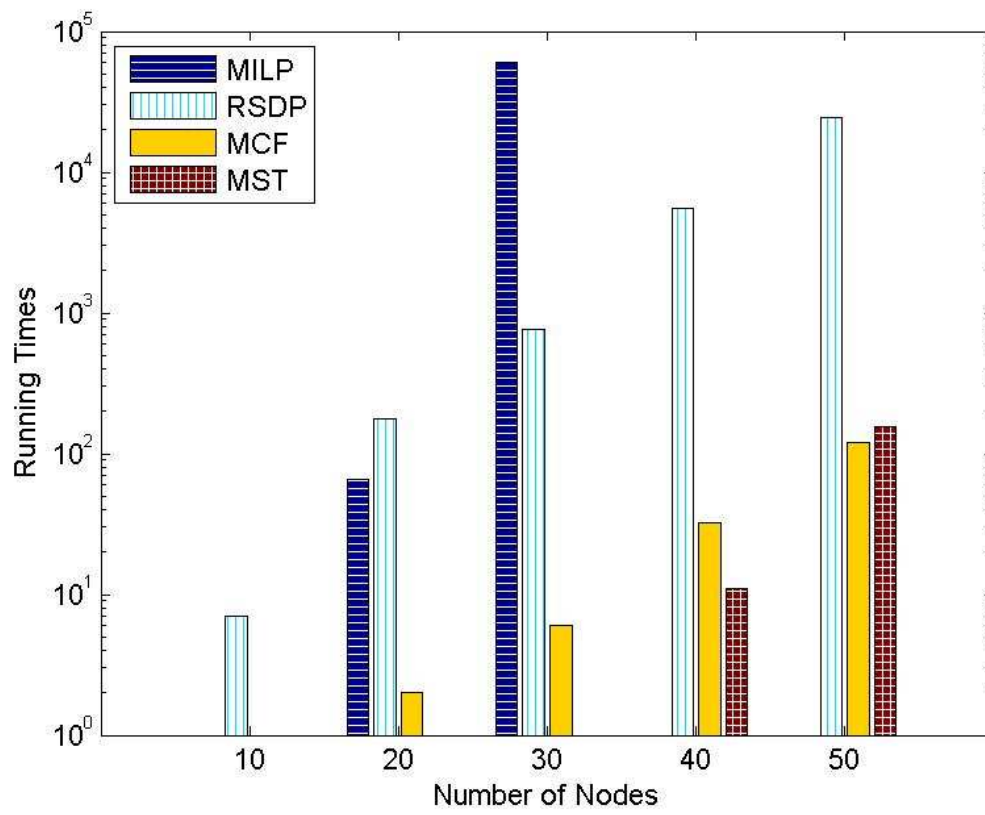


Figure 2.5: Numerical experiments on sparse networks - efficiency.

Table 2.3: Numerical Experiments on Dense Networks.

Number of Nodes	10	20	30	40	50
MILP	77.11	82.71	68.38	NA	NA
MILP Running Time (secs)	1	41	66416		
SDP	42.06	31.40	16.68	19.50	18.07
Ratio	0.55	0.38	0.24		
RSDP UB	81.05	135.24	112.62	162.41	120.79
Ratio	1.05	1.64	1.65		
RSDP Running Time (secs)	10	405	5138	19048	72822
MCF LB	45.86	46.12	39.81	39.73	38.52
Ratio	0.59	0.56	0.58		
MCF UB	83.16	82.89	72.63	85.95	71.92
Ratio	1.08	1.00	1.06		
MCF Running Time (secs)	1	3	36	110	314
MST UB	83.16	84.56	71.42	87.76	72.24
Ratio	1.08	1.02	1.04		
MST Running Time (secs)	1	1	35	208	998

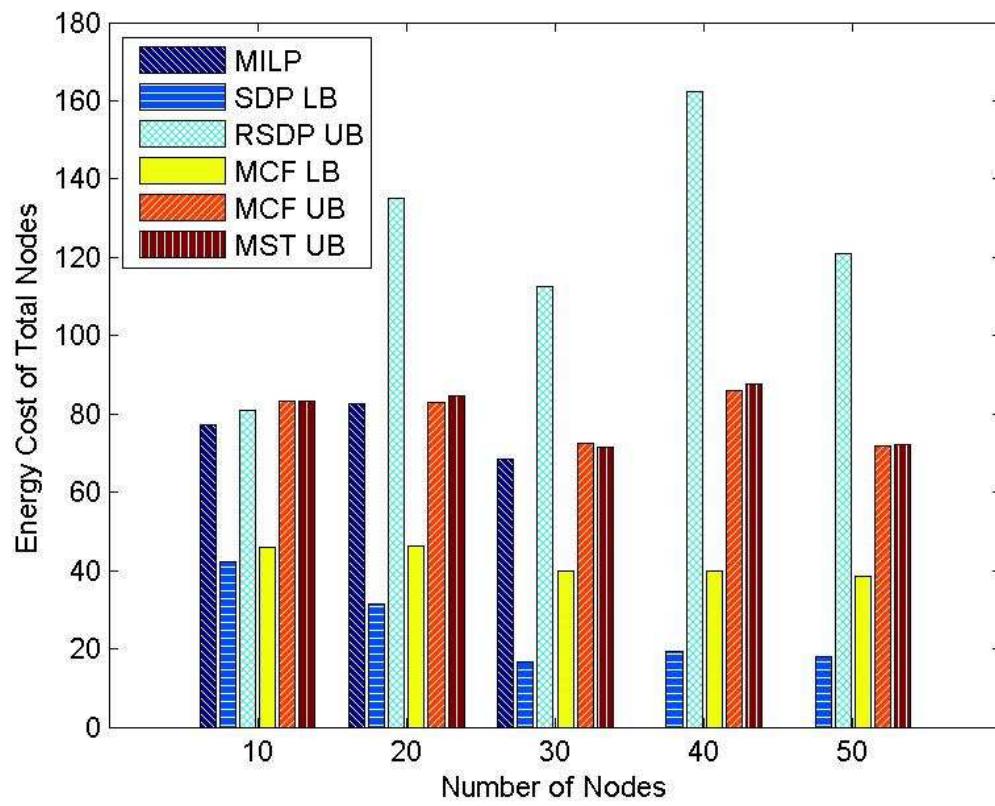


Figure 2·6: Numerical experiments on dense networks - effectiveness.

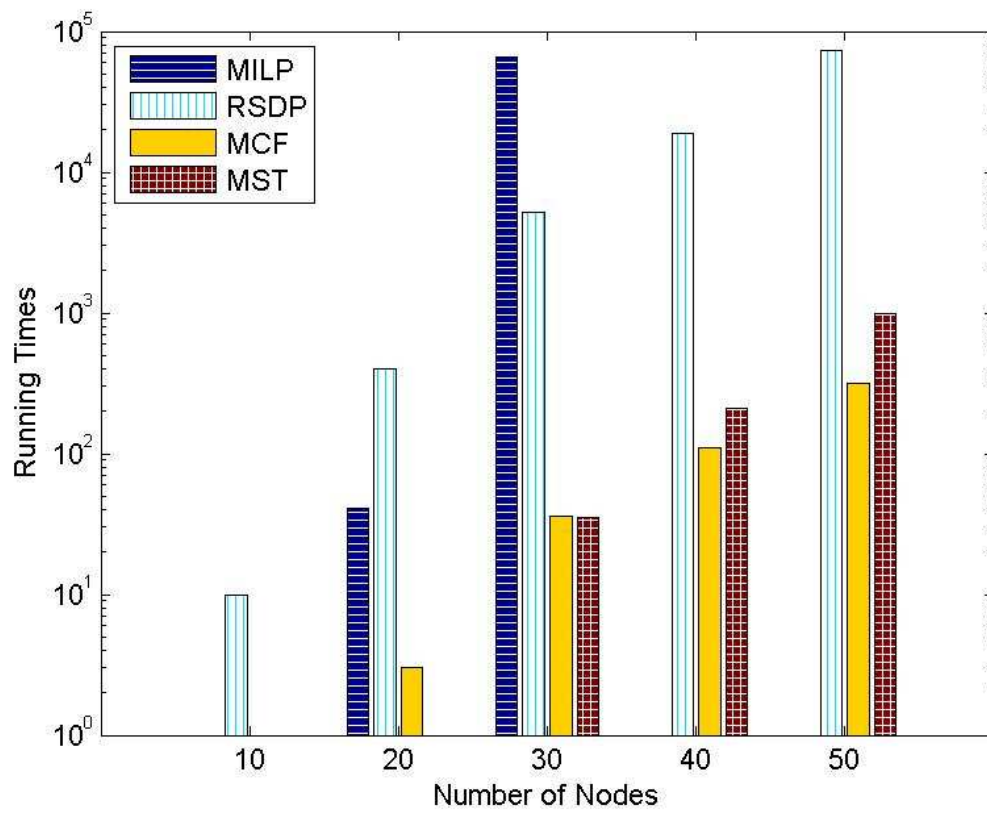


Figure 2.7: Numerical experiments on dense networks - efficiency.

Theorem 2.2 and the Algorithm of Fig. 2.3, respectively. Finally, MST UB denotes the cost of the spanning tree obtained from Algorithm 2.4.

The results indicate that both the minimum-cost-flow-based algorithm (Algorithm 2.3) and the minimum-weight-spanning-tree-based algorithm (Algorithm 2.4) can produce solutions that are very close to the optimal for all those instances where an optimal MILP solution is obtainable. For the larger instances they dominate the SDP-based solution. Moreover, the lower bounds in 2.2 based on the min-cost-flow-based algorithm are tighter than the lower bounds from the SDP relaxation, though the results vary, depending on the network configurations. It is important to note that the bidirectional spanning trees produced by the approaches in Sec. 2.3.4 can be obtained in a distributed manner which is critical for large-scale wireless sensor networks.

Chapter 3

Energy Optimized Topologies for Averaging in Dynamic WSNets

In this chapter, we extend the network model considered in Chapter 2 by incorporating dynamic factors. In particular, we assume that randomness affects pairwise communication between two nodes which can only be established with some probability. Considering technical and practical issues, we assume that we can successfully construct any link present in the network within a potentially large but finite number of attempts. To facilitate implementation of distributed averaging algorithm in such dynamic WSNets, we consider the problem of selecting two nodes that attempt to communicate for load balancing purposes at any given time point. However, this selection should be performed with two concerns. First, network interconnectivity must be reached for every block of B time points to assure the convergence of the load-balancing algorithm. Secondly, energy consumption should be optimized during the whole process. This is a primary issue in WSNets. We formulate the sequential decision problem into a finite horizon DP problem and analyze two scenarios of large enough and limited horizon length. For the first case, we exploit the problem structure and show that the DP optimal solution is equivalent to a Minimum Spanning

Tree (MST) in the graph with appropriately assigned weights. Hence, the problem can be solved in a distributed way which accommodates WSNETs applications. For the second case, due to problem's complexity, we devise some heuristic algorithm to obtain a sub-optimal solution in a short computation time.

The rest of this chapter is organized as follows. In Section 3.1, we introduce the network model, formulate the sequential decision problem as a DP, and establish a monotonicity property. In Section 3.2, we show that for a large horizon length the DP problem can be solved via an MST algorithm. In Section 3.3, we consider the case with pre-determined horizon length and introduce our heuristic. Simulation results demonstrate the effectiveness and efficiency of the proposed algorithms.

Note that work reported in this chapter has been published in [Paschalidis and Li, 2011a] and [Paschalidis and Li, 2011b].

3.1 Problem Formulation

3.1.1 Network Model

We retain the network model in Section 2.1.1 with the following modification. Notice that any two nodes i, j with $(i, j) \in \mathcal{A}$ can communicate with each other. However, any attempt to communicate may not be successful due to physical environmental dynamics and interference from other nodes. To model this uncertainty we introduce a *truncated* geometric distribution with parameters (p_k, M_k) for each (bidirectional) link $k = (i, j) \in \mathcal{A}$. We denote by Y_k the number of trials until a success, which

follows

$$P(Y_k = y) = \begin{cases} \frac{p_k(1-p_k)^{y-1}}{1-(1-p_k)^{M_k}}, & y = 1, 2, \dots, M_k; \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

Notice that the coefficient $1/(1 - (1 - p_k)^{M_k})$ is to ensure that the probabilities add up to one. Suppose one would like build a link between two nodes. One can use proper scheduling schemes to combat interference from neighboring links (see for instance work on the CSMA protocol [Jiang and Walrand, 2009] and other schemes that completely avoid interference [Paschalidis et al., 2009a]). We assume that this ensures that it takes no more than M_k trials for a successful transmission. The expected number of trials needed is:

$$\begin{aligned} E[Y_k] &= \sum_{y=1}^{M_k} P(Y_k = y) \cdot y \\ &= \sum_{y=1}^{M_i} \frac{p_i(1-p_i)^{y-1}}{1-(1-p_i)^{M_i}} \cdot y \\ &= \frac{-p_i}{1-(1-p_i)^{M_i}} \cdot \frac{d}{dp_i} \left(\sum_{y=1}^{M_i} (1-p_i)^y \right) \\ &= \frac{-p_i}{1-(1-p_i)^{M_i}} \cdot \frac{d}{dp_i} \left(\frac{1-p_i-(1-p_i)^{M_i+1}}{p_i} \right) \\ &= \frac{-p_i}{1-(1-p_i)^{M_i}} \cdot \frac{(1-p_i)^{M_i+1} + (M_i+1)p_i(1-p_i)^{M_i} - 1}{p_i^2} \\ &= \frac{1-(M_k+1)(1-p_k)^{M_k} + M_k(1-p_k)^{M_k+1}}{p_k(1-(1-p_k)^{M_k})}. \end{aligned} \quad (3.2)$$

For every trial on link k , let $c_k \geq 0$ be the energy cost for each node, which depends on the power level needed to support the link. Comparing this with the model we considered in Chapter 2, c_k can be seen as the sum of energy costs incurred by the

two nodes incident to link k . The expected cost to successfully construct this link is:

$$\begin{aligned} E[C_k] &= c_k E[Y_k] \\ &= c_k \frac{1 - (M_k + 1)(1 - p_k)^{M_k} + M_k(1 - p_k)^{M_k+1}}{p_k(1 - (1 - p_k)^{M_k})}. \end{aligned} \quad (3.3)$$

Once messages have been successfully exchanged on link k then there are no subsequent transmissions (hence no energy cost) until the next time messages need to be exchanged on this link (as may be needed by Algorithm 1.2).

We note that given m_k failed trials on link k , the total number of trials until a success still follows a truncated geometric distribution (memoryless property):

$$\begin{aligned} (Y_k = y + m_k | Y_k > m_k) &= \frac{P(Y_k = y + m_k)}{P(Y_k > m_k)} \\ &= \frac{p_i(1 - p_i)^{y+m_i-1} / (1 - (1 - p_i)^{M_i})}{1 - \sum_{y=1}^{m_i} p_i(1 - p_i)^{y-1} / (1 - (1 - p_i)^{M_i})} \\ &= \frac{p_i(1 - p_i)^{y+m_i-1} / (1 - (1 - p_i)^{M_i})}{1 - \frac{p_i}{1 - (1 - p_i)^{M_i}} \cdot \frac{1 - (1 - p_i)^{m_i}}{1 - (1 - p_i)}} \\ &= \begin{cases} \frac{p_k(1 - p_k)^{y-1}}{1 - (1 - p_k)^{M_k - m_k}}, & y = 1, 2, \dots, M_k - m_k; \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (3.4)$$

This is equivalent to a geometric distribution with parameter $(p_k, M_k - m_k)$. Analogously, $E[Y_k | Y_k > m_k]$ and $E[C_k | Y_k > m_k]$ can be calculated analytically. The following Lemma is immediate.

Lemma 3.1.1 $P[Y_k = m_k + 1 | Y_k > m_k]$ is a monotonically increasing function of m_k .

Remark: Two issues should be clarified here. First, we assume that any potential link in the network can be constructed after some potentially large but fixed number of trials, i.e., we assume M_k to be finite. This is to ensure that Assumption 1.3 holds for the consensus algorithm to converge. Otherwise, any policy could end up with a non-connected graph with some positive probability. Second, the only other property needed for results in this section to hold is monotonicity (cf. Lemma 3.1.1). Essentially, every failed trial provides some more information about the link between two nodes, such as channel property, antenna angle etc., thus, facilitating the link construction in the next trial. For simplicity of the exposition, we adopt the truncated geometric distribution to characterize the link construction process. Generally, any model resulting in Lemma 3.1.1 is applicable.

3.1.2 Dynamic Programming Model

As we have seen in Section 1.2.1, to guarantee convergence of the load balancing algorithm we need to enforce Assumption 1.3. The convergence time was shown to be polynomial in B and N . In this section we formulate the problem of efficiently enforcing Assumption 1.3, either by minimizing the time until the graph becomes strongly connected or by minimizing the energy cost of doing so.

As before, let $\mathcal{G}(t) = (\mathcal{N}, \mathcal{E}(t))$, $\mathcal{E}(t) \subseteq \mathcal{A}$, denote the communication pattern between the nodes at time t and let $\mathcal{E}_c(t) = \mathcal{E}(1) \cup \mathcal{E}(2) \cup \dots \cup \mathcal{E}(t)$ and $\mathcal{G}_c(t) = (\mathcal{N}, \mathcal{E}_c(t))$. We treat these graphs as undirected and we say that $\mathcal{G}_c(t)$ is strongly connected if the directed graph formed by replacing each link (i, j) with the two

directed links (i, j) and (j, i) is strongly connected. We split time into blocks of length B . In each block, say $\{1, \dots, B\}$, we seek to successfully connect enough links so that $\mathcal{G}_c(B)$ is strongly connected. We can repeat this process in every such block, so we only focus on the link selection decisions concerning a single block.

For simplicity of the analysis, we assume that at every discrete time instant we select a single link and attempt to establish communications between its incident nodes. Such an attempt may succeed or fail according to the model we presented earlier. We note that more than one links can attempt communications at the same time as long as there is no interference between them. To account for this we would need to assume an interference model and resolve the underlying scheduling problem (see e.g., [Paschalidis et al., 2007; Paschalidis et al., 2009a]). In this paper however, we are only focusing on the dynamic link selection and we forgo these physical layer issues.

Before we present the DP formulation let us define the “state” of the network at time t which will consist of two parts. One part is the graph $\mathcal{G}_c(t)$, which includes all successfully constructed links up to time t . The other part contains information on links that have been attempted but not constructed (those that have not been tried can be viewed as have been tried 0 times). We denote the state of these links by $\mathcal{L}(t) = \{k(m_k)\}_{k=1, k \notin \mathcal{G}_c(t)}^{|\mathcal{A}|}$, where $k(m_k)$ indicates that we have made m_k attempts to construct link $k \in \mathcal{A}$. We define the state at time t as $\mathcal{S}(t) = (\mathcal{G}_c(t), \mathcal{L}(t))$.

Recall that for each link k the number of trials until a success follows a truncated

geometric distribution with parameters (p_k, M_k) . As we have seen, given m_k failures the p.m.f. of the residual number of trials until a success also follows a truncated geometric distribution with parameters $(p_k, M_k - m_k)$. The success probability of the next trial on link k is $\frac{p_k}{1 - (1 - p_k)^{M_k - m_k}}$. Given the information at time t and under the assumption that at t we only select a single link, say k , to attempt to construct, $\mathcal{S}(t)$ is Markovian and evolves as follows:

$$\begin{aligned} \mathcal{S}(t+1) &= (\mathcal{G}_c(t+1), \mathcal{L}(t+1)) \\ &= \begin{cases} ((\mathcal{N}, \mathcal{E}_c(t) \cup \{k\}), \mathcal{L}(t) \setminus k(m_k)), & w.p. \quad \frac{p_k}{1 - (1 - p_k)^{M_k - m_k}}, \\ (\mathcal{G}_c(t), \mathcal{L}(t) \setminus k(m_k) \cup k(m_k + 1)), & w.p. \quad 1 - \frac{p_k}{1 - (1 - p_k)^{M_k - m_k}}. \end{cases} \end{aligned} \quad (3.5)$$

If we wish to minimize the total cost of constructing a strongly connected graph $\mathcal{G}_c(B)$ we end up with the following finite-horizon DP iteration:

$$J_t(\mathcal{S}(t)) = \min_{k \in \{\mathcal{A}, \emptyset\}, k \notin \mathcal{E}_c(t)} E[c_k + J_{t+1}(\mathcal{S}(t+1))], \quad (3.6)$$

where $J_t(\mathcal{S}(t))$ is the optimal cost-to-go (or value) function at state $\mathcal{S}(t)$. We make the convention $c_\emptyset = 0$. The boundary conditions are $J_t(\mathcal{S}(t)) = 0$ if $\mathcal{G}_c(t)$ is strongly connected. We note that as written, (3.6) allows no link to be selected at any given time t . This amounts to “idling,” incurs a zero immediate cost, and can be selected when connectivity of $\mathcal{G}_c(t)$ has already been achieved. The horizon (block length) in

(3.6) is equal to B and we impose a terminal cost at the end of the horizon equal to

$$J_B(\mathcal{S}(B)) = \begin{cases} W, & \text{if } \mathcal{G}_c(B) \text{ is not strongly connected,} \\ 0, & \text{otherwise,} \end{cases} \quad (3.7)$$

where $W \gg 1$ is a large enough penalty.

We next establish a useful monotonicity property of the value function $J_t(\mathcal{S}(t))$.

Lemma 3.1.2 *It holds that $\forall t, J_t(\mathcal{S}^\alpha(t)) \geq J_t(\mathcal{S}^\beta(t))$ and $\mathcal{S}^\alpha(t) = (\mathcal{G}_c^\alpha(t), \mathcal{L}^\alpha(t))$, $\mathcal{S}^\beta(t) = (\mathcal{G}_c^\beta(t), \mathcal{L}^\beta(t))$ such that $\mathcal{G}_c^\alpha(t) \subseteq \mathcal{G}_c^\beta(t)$ and $\mathcal{L}^\alpha(t)$ coincides with $\mathcal{L}^\beta(t)$ for all links $k \notin \mathcal{E}_c^\beta(t)$.*

Proof: We will establish the result using a coupling argument. Fix some time t and consider two systems α and β corresponding to states $\mathcal{S}^\alpha(t)$ and $\mathcal{S}^\beta(t)$, respectively. Notice that the randomness is due to the link construction process only. By defining the two systems on a common probability space, we can assume that for each common link, they encounter same sequences of random numbers uniformly distributed on $[0, 1]$, which in turn decide the success or failure of constructing the link. For some particular sample path let $\pi^\alpha(t)$ be the optimal sequence of links selected by System α starting at t . A policy for β is to mimic α , that is, attempt to construct the exact same links at exactly the same time if the link selected by α does not belong to $\mathcal{G}_c^\beta(t)$. Otherwise, that is if the link selected by α is already in $\mathcal{G}_c^\beta(t)$, System β can idle (at no cost). Due to the coupling, both Systems face the exact same success and failure events on the common links they attempt to construct. Since System β idles on the links it has already constructed, its cost will be no more than that of

System α . Taking expectations and noting that System β used a suboptimal policy we establish that $J_t(\mathcal{S}^\alpha(t)) \geq J_t(\mathcal{S}^\beta(t))$. ■

An immediate corollary is that $\mathcal{G}_c(B)$ should be a tree; otherwise we can simply construct only links in a spanning tree of $\mathcal{G}_c(B)$ which will reduce the overall cost. The DP formulation is insightful but it does not lead to practical (efficient) algorithms. Next we develop such algorithms.

3.2 Large Enough Horizon Length

We start with the simpler case where B is large enough, more specific $B \geq \sum_{k \in \mathcal{A}} M_k$, so that network interconnectivity is guaranteed to be reached before B for any feasible policy. Hence, the terminal cost is always 0. The next Proposition establishes another monotonicity property of $J_t(\cdot)$. Because the horizon is long enough and we pay no terminal cost with probability one (w.p.1.), we do not need to keep track of time; we will simplify the notation for the states and the cost-to-go function by writing $\mathcal{S} = (\mathcal{G}_c, \mathcal{L})$ and $J(\mathcal{S})$, respectively.

Proposition 3.2.1 *Suppose we are at some state $\mathcal{S}^\alpha = (\mathcal{G}_c^\alpha, \mathcal{L}^\alpha)$ and link $k \notin \mathcal{E}_c^\alpha$. Assume that there is a positive probability that link k participates in the connected graph at the end of the horizon, i.e., $k \in \mathcal{E}_c^\alpha$ at time B . Consider some other state $\mathcal{S}^\beta = (\mathcal{G}_c^\beta, \mathcal{L}^\beta)$ such that $\mathcal{G}_c^\alpha = \mathcal{G}_c^\beta$ and $\mathcal{L}^\beta = \mathcal{L}^\alpha \setminus k(m_k) \cup k(m_k + 1)$. Then, $J(\mathcal{S}^\alpha) > J(\mathcal{S}^\beta)$.*

Proof: We will use a coupling argument as in the proof of Lemma 3.1.2. Consider two systems α and β corresponding to states \mathcal{S}^α and \mathcal{S}^β , respectively. By defining the two systems on a common probability space, we can assume that for each common

link they encounter the same sequences of random numbers uniformly distributed on $[0, 1]$, which in turn decide the success or failure of constructing the link. Fix some particular sample path Ω and let $\pi^\alpha(\mathcal{S}^\alpha)$ the optimal sequence of links selected by System α starting at state \mathcal{S}^α . System β mimics the System α decisions.

Given Ω , we distinguish two cases. In Case 1, System α does not select link k until the end of the horizon. Then, System β which mimics System α selects the exact same links and both systems find selected links at the exact same state. Given the coupling we introduced, both systems accumulate the same cost until the end of the horizon.

In Case 2, System α does select link k until the end of the horizon. System β selects link k at exactly the same times. Every time that both Systems select link k System β finds k in a state with one more failure. Note that for the corresponding success probabilities it holds $P^\alpha[\text{Success}|m_k] = \frac{p_k}{1-(1-p_k)^{M_k-m_k}} < \frac{p_k}{1-(1-p_k)^{M_k-m_k-1}} = P^\beta[\text{Success}|m_k+1]$. Success is decided by drawing a random number s uniformly from $[0, 1]$ and declaring success in connecting link k in System α if $s \leq P^\alpha[\text{Success}|m_k]$. Given the coupling we introduced, success in System β is determined by the exact same random number. It follows that for all realizations of s that link k gets connected in System α it also gets connected in System β . For these realizations where link k gets connected in both Systems at the same time it follows that both systems accumulate the same cost until the end of the horizon.

There are however realizations Ω in which at some attempt, say at time t' , to connect link k it only gets connected in System β . For these realizations, using the same argument as in Lemma 3.1.2, the total cost until the end of the horizon in System β is no more than the cost in System α .

Given the assumption we made in the statement of the proposition, note that there are realizations with positive probability in which System α will need to select link k after time t' . For these sample paths System β can idle (at no cost) while System α attempts to connect link k . Since no penalty will be paid by either System idling does not affect the total cost. Thus, the overall cost of System β is strictly less than that of System α .

We have shown that for all sample paths Ω the cost of System β is no more than that of System α and that there are sample paths Ω with positive probability for which the cost of System β is strictly less than that of System α . Taking expectations over all sample paths we establish that $J(\mathcal{S}^\alpha) > J(\mathcal{S}^\beta)$. ■

3.2.1 An MST-Based Policy

We next evaluate a policy which is much easier to compute than the optimal DP policy. Specifically, we consider the policy which when this policy selects a certain link k it continues to try that link until the link becomes connected.

Starting from an empty graph $\mathcal{G}_c(0)$, suppose that we select link k . We will concentrate on the trials required for connecting k . Let us denote $\tilde{J}(k(m_k))$ as the cost-to-go function of the particular policy we described after m_k failed trials on k .

Let $\tilde{J}(k(M_k))$ denote the cost-to-go after link k has been successfully connected. The probability of constructing link k at the m_k -th trial is $p_k^{(m_k)} = \frac{p_k}{1-(1-p_k)^{M_k-m_k+1}}$, where $0 < m_k \leq M_k$. We have

$$\begin{aligned}
& \tilde{J}(\mathcal{S}(0)) \\
&= c_k + p_k^{(1)} \tilde{J}(k(M_k)) + (1 - p_k^{(1)}) \tilde{J}(k(1)) \\
&= c_k + p_k^{(1)} \tilde{J}(k(M_k)) + (1 - p_k^{(1)}) [c_k + p_k^{(2)} \tilde{J}(k(M_k)) + (1 - p_k^{(2)}) \tilde{J}(k(2))] \\
&= c_k + c_k(1 - p_k^{(1)}) + (p_k^{(1)} + (1 - p_k^{(1)})p_k^{(2)}) \tilde{J}(k(M_k)) + (1 - p_k^{(1)})(1 - p_k^{(2)}) \tilde{J}(k(2)) \\
&= \dots \\
&= c_k (P[Y_k \geq 1] + P[Y_k \geq 2] + \dots + P[Y_k \geq M_k]) \\
&\quad + (P[Y_k = 1] + \dots + P[Y_k = M_k]) \tilde{J}(k(M_k)) \\
&= c_k E[Y_k] + \tilde{J}(k(M_k)).
\end{aligned}$$

We can now proceed in the same manner and attempt to connect a second link. We repeat this process until a spanning tree of \mathcal{A} is formed (so that $\mathcal{G}_c(t)$ becomes strongly connected for some t). It follows that if \mathcal{T} denotes a spanning tree of \mathcal{A} then

$$\tilde{J}(\mathcal{S}(0)) = \min_{\mathcal{T}} \sum_{k \in \mathcal{T}} c_k E[Y_k]. \quad (3.8)$$

The analyzed policy leads to the following algorithm. We note that the MST problem can be solved in a distributed manner by using an algorithm in [Gallager et al., 1983].

Algorithm 3.1 (A Minimum Weight Spanning Tree Construction Algorithm)

For every link k present in \mathcal{A} , assign $c_k E[Y_k]$ as its weight and compute the Minimum weight Spanning Tree (MST).

We will now establish that this MST-based algorithm is optimal.

Proposition 3.1 *Algorithm 3.1 is optimal.*

Proof: Consider the optimal policy, say π , obtained by solving the DP problem in (3.6). The optimal policy constructs a connected graph $\mathcal{G}_c(B)$ at the end of the horizon and, given the assumption in effect in this section, pays no terminal cost. Pick a spanning tree \mathcal{T}_c of $\mathcal{G}_c(B)$ (e.g., uniformly over all spanning trees of $\mathcal{G}_c(B)$) and let l_1, \dots, l_{N-1} denote the links it contains. For any sample path Ω the cost accumulated by π may contain unsuccessful trials at various links, hence, this cost is no less than the cost paid to construct \mathcal{T}_c . The latter cost is $\sum_{i=1}^{N-1} c_{l_i} y_{l_i}$ where y_{l_i} is the number of trials contained in Ω in order to construct link l_i . Taking expectations over all Ω that result in the same \mathcal{T}_c we obtain a cost of $\sum_{i=1}^{N-1} c_{l_i} E[Y_{l_i}]$. Clearly, this cost is no less than $\tilde{J}(\mathcal{S}(0))$.

Finally, consider an arbitrary sample path of π and let $p_{\mathcal{T}}$ the probability that this sample path leads to a spanning tree \mathcal{T} of \mathcal{A} . This probability corresponds to both the sample path and the manner in which a spanning tree is selected from $\mathcal{G}_c(B)$. Taking expectation over all sample paths we obtain an optimal expected cost equal to $\sum_{\mathcal{T}} p_{\mathcal{T}} \sum_{l_i \in \mathcal{T}} c_{l_i} E[Y_{l_i}]$ which is also no less than $\tilde{J}(\mathcal{S}(0))$. This establishes the optimality of the MST-based algorithm. ■

As in the static case, we investigate the energy cost associated with the MST computation. By employing the distributed algorithm of [Gallager et al., 1983] and following the same analysis as in the static case, a total of *successful* $O(|\mathcal{A}| + |\mathcal{N}| \log |\mathcal{N}|)$ messages have to be exchanged. Considering the worst case where maximal number of trials is required for each message, the total number of such trials is $\Theta((|\mathcal{A}| + |\mathcal{N}| \log |\mathcal{N}|) \cdot M_{\max})$, where $M_{\max} = \max_{k \in \mathcal{A}} M_k$. Therefore, the total energy cost is $\Theta((|\mathcal{A}| + |\mathcal{N}| \log |\mathcal{N}|) \cdot M_{\max} \cdot c_{\max}) = \Theta(|\mathcal{A}| + |\mathcal{N}| \log |\mathcal{N}|)$, where $c_{\max} = \max_{k \in \mathcal{A}} c_k$ is the maximal energy cost for one trial at any link in the network. Note that the energy cost considered here takes into account the cost of building the topology and applying the load balancing algorithm for every block of B time units. Once the MST is found, the *expected* energy cost of building the topology and applying one iteration of the load balancing algorithm 1.2 is $\tilde{J}(\mathcal{S}(0))$.

If we interpret the cost c_k as the energy cost for a trial on link k , then the MST-based algorithm minimizes the *expected energy cost to reach connectivity*. We next consider a number of alternative options on the selection of these costs.

1. *Minimum expected interconnectivity time.* In this case we set $c_k = 1$ for all links $k \in \mathcal{A}$. Essentially we seek to minimize the expected time until we construct a connected graph (a spanning tree of \mathcal{A}).
2. *Mixed cases.* As a way to take into account both the energy cost and time we introduce a fixed “set-up” cost c_0 for each trial on any link. Specifically, we set $c_k := c_0 + c_k$ thus penalizing many trials (hence a long time to reach

connectivity) even if these trials do not cost a lot in terms of energy.

3.2.2 Numerical Experiments

We generate networks by uniformly scattering N nodes on a 10×10 square. We assume that the minimum power needed by a node to reach another node is d^2 , where d is their distance. We employ the sigmoid function to relate p_k , the success probability for trial on link k , with d_k , the distance between the two nodes incident to k , i.e., $p_k = 2/(e^{d_k^2/50} + 1)$. We assume that the maximum power of each node is large enough to cover the whole region. The maximum number of trials M_k for each link k is an integer uniformly drawn from $[1, 5]$.

Table 3.1: MST-Based Algorithm vs. DP Algorithm.

MST-Based Algorithm			DP Algorithm	
N	Running Time (secs)	Result	Running Time (secs)	Result
3	<1	47.99	<1	47.99
4	<1	55.93	10.01	55.93
5	<1	127.84	2.45×10^3	127.84
6	<1	149.42	6.21×10^4	NA

We compare our MST-based algorithm for each cost selection we discussed with the corresponding DP algorithm. All algorithms are run on a computer with Ubuntu-8.04-OS, 2GB of memory, and Intel-XEON-2.00GHz CPU. As expected, our algorithms output the same results as the DP algorithm. However, in terms of running

time, our algorithms are much more efficient than DP, which becomes incredibly slow when $n \geq 5$ and runs out of memory almost every time. In contrast, our algorithms usually take less than 1 second for most instances. Table 3.1 shows typical numerical experiments, where NA means that the DP algorithm runs out of memory.

3.3 Limited Horizon Length

We now turn our attention to the more challenging case where a terminal penalty is incurred when interconnectivity can not be reached within a *small* block of length B , i.e., $|\mathcal{N}| - 1 \leq B < \sum_{k \in \mathcal{A}} M_k$ such that some policy may fail within B time units. Since solving the DP problem is not practical even for reasonably-sized instances we seek suboptimal solutions. To that end, we will leverage the so-called rollout algorithms introduced in [Bertsekas et al., 1997].

3.3.1 An Algorithm Based on Rollout

Rollout algorithms offer approximate solutions to discrete optimization problems using procedures that are capable of magnifying the effectiveness of any given heuristic algorithm through sequential application. In particular, in [Bertsekas et al., 1997], the problem is embedded within a DP framework, and several types of rollout algorithms are introduced, which are related to notions of policy iteration. It has been proved that under certain conditions the rollout algorithm is guaranteed to improve the performance of the original heuristic algorithm.

The key idea of rollout algorithms is to employ one or more suboptimal policies

and use their value function in a policy improvement step. The policy obtained through this step is then applied. To make matters concrete, consider the decisions induced by the DP iteration in (3.6), namely, at time t we select link

$$l = \arg \min_{k \in \{\mathcal{A}, \emptyset\}, k \notin \mathcal{E}_c(t)} E[c_k + H_{t+1}(\mathcal{S}(t+1))], \quad (3.9)$$

with the only difference being that instead of using the optimal policy to evaluate the cost-to-go at the next state we use a heuristic/suboptimal policy \mathcal{H} whose cost-to-go is denoted by $H_{t+1}(\mathcal{S}(t+1))$.

As policy \mathcal{H} we can employ one of the MST-based heuristics we developed in Sec. 3.2: Algorithm 3.1 which minimizes expected cost and its special case with $c_k = 1$ for all k which minimizes the time to reach interconnectivity. More specifically, starting from state $\mathcal{S}(t+1) = (\mathcal{G}_c(t+1), \mathcal{L}(t+1))$ we apply each one of the MST-based heuristics to compute the expected cost of adding links to $\mathcal{G}_c(t+1)$ in order to form a connected subgraph of \mathcal{A} . To that end, we can simply (i) concatenate each connected component of $\mathcal{G}_c(t+1)$ into one aggregate node g , (ii) form the graph \mathcal{G}_{t+1} whose node set includes g and all remaining nodes in \mathcal{N} that were not included in g and whose edge set includes all links in \mathcal{A} that are not included in $\mathcal{E}_c(t+1)$, and then (iii) form an MST of \mathcal{G}_{t+1} . Note that if $\mathcal{G}_c(t+1)$ contains no cycles then the resulting graph will be a spanning tree of \mathcal{A} . Let $\tilde{J}_{\text{cost}}(\mathcal{S}(t+1))$ be the total cost for constructing the MST we just described starting from state $\mathcal{S}(t+1)$ when we use Algorithm 3.1 (cf. (3.8)). Let also $\tilde{J}_{\text{time}}(\mathcal{S}(t+1))$ be the corresponding cost when we

use Algorithm 3.1 with c_k 's set to one. These costs account for the cost to connect the necessary links but do not account for any potential penalty cost that has to be paid if no connectivity is achieved before the end of horizon. We next describe how such expected penalty costs can be computed.

Suppose that at state $\mathcal{S}(t+1)$ the computed MST selects links $1, \dots, K$, where each one of these links has already been tried m_1, \dots, m_K times, respectively. The time Y_k needed to connect link $k = 1, \dots, K$ has a truncated geometric distribution with parameters $(p_k, M_k - m_k)$. Its z -transform is given by

$$E[z^{Y_k}] = \frac{p_k}{[1 - p_k - (1 - p_k)^{M_k - m_k + 1}]} \sum_{y=1}^{M_k - m_k} (1 - p_k)^y z^y. \quad (3.10)$$

The z -transform of the total time needed to construct the selected MST is

$$\prod_{k=1}^K E[z^{Y_k}] = \prod_{k=1}^K \frac{p_k}{[1 - p_k - (1 - p_k)^{M_k - m_k + 1}]} \cdot \prod_{k=1}^K \sum_{y=1}^{M_k - m_k} (1 - p_k)^y z^y. \quad (3.11)$$

The coefficient of z^y , for $y = K, \dots, \sum_{k=1}^K (M_k - m_k)$ in (3.11) is equal to the probability that it will take y steps to construct the MST. Given that we are at time $t+1$, we can compute the probability, say p_{t+1}^F , that a penalty cost will be paid by summing up all coefficients of z^y for all $y > B - (t+1)$. Thus, the expected penalty cost is equal to $W p_{t+1}^F$.

Now, at any time $t+1$ and state $\mathcal{S}(t+1)$ with a connected $\mathcal{G}_c(t+1)$, policy \mathcal{H} does not need to select any links, hence $H_{t+1}(\mathcal{S}(t+1)) = 0$. Finally, at time B , the policy \mathcal{H} has no time to act and for any state $\mathcal{S}(B)$ the cost-to-go is W if $\mathcal{G}_c(B)$ is

not connected and 0 otherwise.

Collecting all of the above and letting $H_{t+1}^{\text{cost}}(\cdot)$ denote the cost-to-go of the MST-based policy which uses Algorithm 3.1, we have

$$H_{t+1}^{\text{cost}}(\mathcal{S}(t+1)) = \begin{cases} 0, & \text{if } \mathcal{G}_c(t+1) \text{ is connected,} \\ W, & \text{if } t+1 = B \text{ and } \mathcal{G}_c(B) \text{ is not connected,} \\ \tilde{J}_{\text{cost}}(\mathcal{S}(t+1)) + Wp_{t+1}^F, & \text{otherwise.} \end{cases} \quad (3.12)$$

Similarly, we can obtain the cost-to-go $H_{t+1}^{\text{time}}(\cdot)$ of the MST-base policy that uses Algorithm 3.1 with c_k 's set to one. The following algorithm uses both MST-based policies we discussed to obtain an improved policy.

Algorithm 3.2 (A Heuristic Rollout Algorithm)

1. Given the current state $\mathcal{S}(t)$ and for each link $k \in \mathcal{A}$ such that $k \notin \mathcal{E}_c(t)$ consider performing one more trial on k . Compute the two possible next states $\mathcal{S}(t+1)$ as in (3.5) corresponding to a success or failure on link k . For each possible next state $\mathcal{S}(t+1)$:
 - (a) Apply Algorithm 3.1 and compute $H_{t+1}^{\text{cost}}(\mathcal{S}(t+1))$.
 - (b) Apply Algorithm 3.1 with the c_k 's set to one and compute $H_{t+1}^{\text{time}}(\mathcal{S}(t+1))$.
2. Select link l such that

$$l = \arg \min_{k \in \{\mathcal{A}, \emptyset\}, k \notin \mathcal{E}_c(t)} c_k + E[\min\{H_{t+1}^{\text{cost}}(\mathcal{S}(t+1)), H_{t+1}^{\text{time}}(\mathcal{S}(t+1))\}], \quad (3.13)$$

where the expectation is taken with respect to the two possible outcomes for the next state.

As it has been pointed out in [Bertsekas et al., 1997] Rollout algorithms may not necessarily terminate and can cycle. While the principle of optimality precludes cycling when one applies an optimal policy sequentially, with a suboptimal policy it is possible that a sequence of actions can lead to the exact same state and form a cycle. This is not possible in our setting because we are dealing with a finite number of states and no state gets repeated. To see this note that every action (selection of a link) increases the number of times that link has been tried (which is reflected in $\mathcal{L}(t)$). We summarize this argument in the following proposition.

Proposition 3.2 *The rollout algorithm introduced in Algorithm 3.2 is terminating.*

We note that Algorithm 3.2 is not a distributed algorithm since the approximated cost-to-go function and link selection are based on global network information. For each block of B time units, the total energy cost is associated with connecting the sequence of selected links, which in turn is determined by the network status, the value of B and the terminal penalty cost W .

3.3.2 Numerical Experiments

We generate networks by uniformly scattering N nodes on a $a \times a$ square, where $a = 20$. We assume that the minimum power needed by a node to reach another node is d^2 , where d is their distance. The maximum power available at each node is λa^2 . We employ the sigmoid function to relate p_k , the success probability of trial on link k , with d_k , the distance of nodes incident to k , i.e., $p_k = 2/(e^{d_k^2/10} + 1)$. To simplify

the calculations we perform, we set $p_i = 0.1$ if $p_i < 0.1$. The maximum number of trials M_k for link k is an integer drawn uniformly from the interval $[1, 10]$. We set the penalty cost as $W = 1000 \times \max_{i,j \in \{1,2,\dots,n\}} d_{(ij)}^2$.

Table 3.2: The Effectiveness of the Rollout Algorithm.

B	N	10	12	14	16
25	SC	20	20	20	20
	AS	12.3	13.4	14.2	15.0
	AE	743.1	957.1	707.6	944.5
30	SC	20	20	20	20
	AS	13.7	14.9	15.4	15.3
	AE	531.3	755.6	444.6	805.2
35	SC	20	20	20	20
	AS	14.1	15.4	17.1	16.0
	AE	528.7	636.9	409.7	383.5

First, we test the effectiveness of rollout algorithms and examine how the residual horizon length affects the preference between fast interconnectivity (choosing $H_{t+1}^{\text{time}}(\cdot)$ in (3.13)) and energy efficiency (choosing $H_{t+1}^{\text{cost}}(\cdot)$ in (3.13)). We set $\lambda = 2$, so that each node has enough power to cover the whole area (a *dense* network). For various B , we run 20 instances of the algorithm and compute the average number of steps needed and average energy cost required to achieve interconnectivity within the time block of length B . The results are shown in Tab. 3.2, where SC stands for “success counts,” denoting the number of instances reaching interconnectivity. AS stands for “average

Table 3.3: The Sub-Optimality of the Rollout Algorithm.

DP Algorithm			Rollout Algorithm		
B	$\mathcal{J}(0)$	Running Time (secs)	Rollout	Running Time (secs)	SC
3 Node Case					
5	166.69	0.23	1628.1	0.18	49
10	46.68	0.44	48.40	0.19	50
15	46.68	0.65	47.27	0.20	50
4 Node Case					
5	126.57	13.19	150.66	0.61	50
10	85.97	25.83	92.20	0.73	50
15	85.97	38.42	86.54	0.69	50
4 Node Case					
5	2792.5	341.49	6390.7	0.79	46
10	87.54	668.95	89.23	0.76	50
15	84.10	994.15	86.03	0.77	50

steps,” denoting the average number of steps needed to reach interconnectivity. AE stands for “average energy cost,” denoting the average total energy consumed.

Next we compare the performance of the rollout algorithm with DP in small enough instances so the latter is tractable. The results are in Table 3.3. For DP we report the optimal cost-to-go starting at $t = 0$ and the corresponding running time. For the rollout algorithm we report its performance (average over 50 runs), running

Table 3.4: The Efficiency of the Rollout Algorithm in Sparse Networks.

n	Link Density	Decision Time (sec)
10	68/90	0.32
20	242/380	2.97
30	474/870	11.66
40	928/1560	50.01
50	1296/2450	104.21
60	2276/3540	447.75

time, and “success counts,” denoting the number of instances reaching interconnectivity. Clearly, the rollout is much faster and for most instances its performance is close enough to DP. There are, however, two cases with small B ($B = 5$) when rollout instances will pay a penalty while the optimal policy manages to avoid it; in such cases, the rollout performance is much worse than DP.

Next, we assess the scalability of the algorithm. In this scenario, we set $\lambda = 0.3$, which is possibly closer to practical *sparse* sensor network deployments. We gradually increase the number of nodes and record the time needed to make a decision using our rollout algorithms (cf. Eq. (3.13)). The results are in Table 3.4. The first column lists the total number of nodes. In the second column, the first number is the number of total potential links while the second number is the number of links in a complete graph with the same number of nodes. The third column corresponds to the running time for each iteration of the rollout algorithm. Notice that although ρ is small in our

setting, the network contains many links, even compared to the complete network.

We conclude that our rollout algorithm works fairly well in all instances. In almost all randomly generated instances, it produces a suboptimal policy that can reach network interconnectivity within the limited horizon length. We observed that as the limited horizon length B increases, the algorithm tends to prefer the minimum cost MST-based policy (i.e., $H_{t+1}^{\text{cost}}(\cdot)$ wins in the minimization appearing in (3.13)), which is not surprising given that the risk of paying a penalty is reduced. Moreover, as Table 3.4 indicates, our rollout algorithm is scalable and can efficiently output suboptimal solutions within a reasonable running time even for large-scale instances.

Chapter 4

A Market-Based Mechanism for Providing Demand-Side Regulation Service Reserves

In this chapter, we address demand control in smart buildings that are (i) equipped with a sub-metering and actuation capable smart microgrid accessible by occupants as well as by a Smart Microgrid Operator (SMO), and (ii) connected to a cyber infrastructure enhanced smart grid that can support close-to-real-time power market transactions including participants connected at the distribution level.

We develop a market-based mechanism that enables a building SMO to offer regulation service reserves and meet the associated obligation of fast response to commands issued by the wholesale market Independent System Operator (ISO) who provides energy and purchases reserves.

As discussed in the Introduction, the main motivation for demand control is the substantial variability that renewable generation brings to electricity production. Controlling demand allows the ISO to better match supply with demand and accommodate a larger percentage of renewables in the production mix. The proposed market-based mechanism allows the SMO to modulate the Markovian behavior of occupants/loads through dynamic price signals. Loads are grouped in several classes.

Class-specific loads arrive stochastically at rates that depend on SMO dynamically controlled price signals. Each arriving load exhibits a fixed, class-specific energy consumption level and is active for a stochastic time period after which it departs. A regulation service reserves quantity is transacted between the SMO and the ISO for a relatively long period of time (e.g., one hour) that we call the long time scale (cf. Chapter 1). During this time period, the ISO follows shorter time scale (cf. Chapter 1) stochastic dynamics to repeatedly request the SMO to decrease or increase its consumption by a random fraction of the transacted regulation service reserve quantity. These ISO requests are also modeled as a stochastic process which responds to SMO price signals that assure the requests do not exceed the transacted regulation service reserve quantity. We model the operational task of selecting an optimal short time scale dynamic price policy as a stochastic dynamic program that maximizes average SMO and ISO utility over the long time scale horizon. We then formulate an associated *Non-Linear Programming* (NLP) static pricing policy problem that provides an upper bound of the optimal dynamic policy performance and we show that the bound is asymptotically tight as the number of SMO occupants grows large. This asymptotic result provides an efficient approximation of the dynamic pricing policy. Equally importantly, it also allows us to optimize the long time scale decision of determining the optimal regulation service reserve quantity. We finally demonstrate, verify and validate the proposed approach to demand-side-provided regulation service through a series of Monte Carlo simulations of controlled system time trajectories.

The rest of this chapter is organized as follows. In Section 4.1, we detail our internal market based model and formulate a related welfare maximization problem. In Section 4.2 we cast the problem into a DP framework to obtain the optimal dynamic policy. We then proceed to develop performance bounds and approximations. In Section 4.3 we develop a static policy to derive the optimal static welfare as a lower bound, and in Section 4.4 we derive an easily computable upper bound on the optimal performance. Based on this bound, we establish in Section 4.5 the asymptotic optimality of the static policy as the load class specific consumption level becomes smaller with a commensurate increase in the number of active loads, and extend the asymptotic optimal results to account for constraints modeling energy neutrality over the long time scale and the upper limit in the RS delivery requested by the ISO. We present numerical results in Section 4.6.

Note that part of work reported in this chapter is presented in [Paschalidis et al., 2011].

4.1 Problem Formulation

This section models the short time scale interaction of the SMO with microgrid occupants/loads and the ISO in conjunction with RS reserves.

The SMO can sell R_h KW of regulation service for the duration of the long time scale (e.g., one hour), provided that its microgrid's average consumption, R , exceeds R_h and its consumption capacity is at least $R + R_h$. We envision microgrid load

classes that can be potentially active during the relevant long time period to include, among others, lights, space-conditioning units or HVAC zones, computers, electrical appliances and the like. We denote the event of a load unit becoming active as an *internal arrival* (i.e., internal to the building) and associate a class specific electricity demand increment with each arrival. We similarly denote the event of a load unit becoming inactive as an *internal departure*. An actively consuming load unit derives a positive utility. With the sale of R_h KW of RS the SMO agrees to be on standby and respond to short time scale (e.g., seconds to minutes) ISO requests for an increment or decrement of the building's consumption. We denote the event of an ISO request as an *external arrival* (i.e., external to the building). The termination of an ISO request is modeled as an *external departure*. Note that the cumulative ISO increment or decrement requests can not exceed R_h or $-R_h$ respectively. As mentioned, the SMO's response does not have to be instantaneous. It must adhere, however, to a response rate of roughly $R_h/5$ KW per minute. ISO requests that are met by the SMO result in positive utility. In addition, in its periodic 5 minute system re-dispatch, the ISO typically attempts to reset its cumulative increment or decrement requests to zero in order to enable RS providers to respond to new increment or decrement requests during future inter-dispatch 5 minute periods. This renders the long time scale average deviation of building consumption from its R level equal to zero, and, hence, the sale of RS reserves has an energy neutral impact on long time scale building consumption.

The primary objective is to maximize the sum of SMO and ISO welfare associated with internal and external arrivals. Hard and soft constraints are added to model adherence to the contractual requirements and long time scale energy neutrality described above. To achieve these goals, the SMO controls the active internal loads and external requests by communicating external and internal-class-specific *prices* that may be interpreted as dynamic demand control and RS activation feedback signals.

We assume M classes of internal loads $i = 1, \dots, M$, that arrive according to a Poisson process and require r_i KW for an exponentially distributed period with rate μ_i . Let $\mu = (\mu_1, \dots, \mu_M)$. Each internal arrival of class i pays a SMO determined price u_i ; we define $\mathbf{u} = (u_1, \dots, u_M)$. We assume that the arrival rate of class i loads is a known demand function $\lambda_i(u_i)$ which depends on u_i and satisfies Assumption 4.1. We denote the number of *active* class i internal loads at time t by $n_i(t)$, $i = 1, \dots, M$, and define $\mathbf{N}(t) = (n_1(t), \dots, n_M(t))^T$.

Assumption 4.1

For every i , there exists a price $u_{i,\max}$ beyond which the demand $\lambda_i(u_i)$ becomes zero. Furthermore, the function $\lambda_i(u_i)$ is continuous and strictly decreasing in the range $u_i \in [0, u_{i,\max}]$.

ISO requests for the dynamic activation of RS reserves are modeled as a special external class. External RS activation requests occur at a rate $a(y)$ where y is a SMO set price and $a(y)$ satisfies Assumption 4.2. While they are active, external arrivals require r_e KW each. They become inactive upon their departure which follows an exponentially distributed process with rate d . Denoting the number of active

external class loads at time t by $m(t)$, we can express the request for incremented or decremented building energy consumption R as $R + R_h - m(t)r_e$. We impose the following two constraints:

$$\mathbf{N}(t)' \mathbf{r} + m(t)r_e = \sum_{i=1}^M n_i(t)r_i + m(t)r_e \leq R + R_h, \quad (4.1)$$

$$m(t)r_e \leq 2R_h. \quad (4.2)$$

Assumption 4.2

There exists a price y_{\max} beyond which the demand $a(y)$ becomes zero. Furthermore, the function $a(y)$ is continuous and strictly decreasing in the range $y \in [0, y_{\max}]$.

Inequality (4.1) ensures that at any time t the total capacity usage of all active loads does not exceed the maximal building consumption capacity. Inequality (4.2) ensures that the ISO can not request that the average building consumption R be incremented beyond $R + R_h$ or decremented below $R - R_h$. This is true because the modified energy load for dynamic provision of RS equals $R + R_h - m(t)r_e$ and $m(t)r_e$ can not exceed $2R_h$. In other words, depending on the number of external ISO requests present in the system, the SMO is being asked to consume anywhere between $R - R_h$ and $R + R_h$.

We note that due to constraints (4.1) and (4.2), it could happen that, at some state (\mathbf{N}, m) , an internal class i arrival or an external RS activation request cannot be accepted. Without loss of generality, we correspondingly set price $u_i = u_{i,\max}$ (i.e., $\lambda_i = 0$) or $y = y_{\max}$ (i.e., $a = 0$), whenever such a rejection happens. Therefore, the feasible action space is dependent on state (\mathbf{N}, m) .

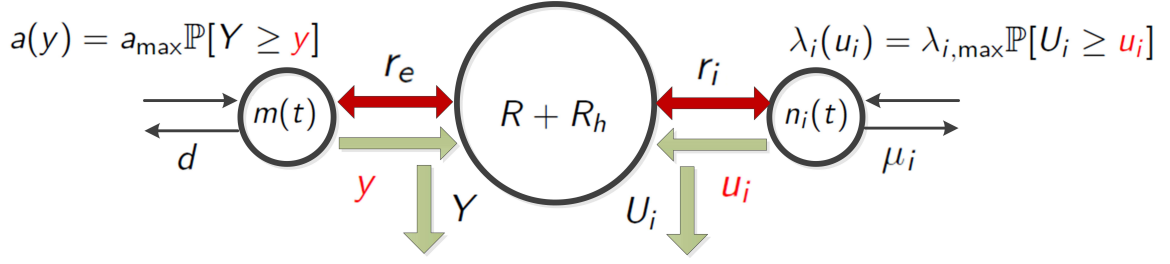


Figure 4.1: The system model.

To render the proposed constrained welfare maximization problem more meaningful we describe first the detailed model of arrivals and the underlying demand function. A simple figurative demonstration of the system is presented in Fig. 4.1. Arrival of an internal load of class i generates utility U_i , where U_i is a nonnegative random variable taking values in the range $[0, u_{i,\max}]$, and which is described by a continuous probability density function $f_i(u_i)$. Arrivals of internal class i loads are a fraction of *potential* class i arrivals generated according to a Poisson process with constant rate $\lambda_{i,\max}$. A potential arrival becomes a real arrival if and only if the random utility realization, U_i , exceeds the SMO set price u_i . This implies that internal class i arrivals occur according to a randomly modulated Poisson process with rate $\lambda_i(u_i(t)) = \lambda_{i,\max} \mathbb{P}[U_i \geq u_i(t)]$. Furthermore, the expected utility conditioned on the fact that a potential arrival has been accepted under a current price of u_i , is equal to $\mathbb{E}[U_i | U_i \geq u_i]$. We therefore conclude that the expected long-term average rate at which utility is generated by the arrival of internal loads is given by:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i=1}^M \mathbb{E} \left[\int_0^T \lambda_i(u_i(t)) \mathbb{E}[U_i | U_i \geq u_i(t)] dt \right].$$

Following a similar approach, the welfare generated from external RS class arrivals can be expressed as:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\int_0^T a(y(t)) \mathbb{E}[Y|Y \geq y(t)] dt \right],$$

where Y stands for the welfare from the admission of a potential external RS arrival, and $a(y(t)) = a_{\max} \mathbb{P}[Y \geq y(t)]$ and a_{\max} is the maximal arrival rate of the external RS class. An interesting interpretation of the long-term average utility generated by external RS class arrivals is that it represents the reservation reward level that the ISO might be willing to pay the SMO for stand-by RS reserves.

Finally, recalling that building response to active ISO RS requests implies that the modified building load must equal $R + R_h - m(t)r_e$, to avoid compliance by energy dumping we impose the following penalty:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\int_0^T P \left((R + R_h) - \left(\sum_{i=1}^M n_i(t)r_i + m(t)r_e \right) \right) dt \right],$$

where $P(\cdot)$ denotes the penalty function. We make specific assumptions on $P(x)$ later.

The optimal pricing policy can now be described as the *argmax* of:

$$\begin{aligned} \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\sum_{i=1}^M \int_0^T \lambda_i(u_i(t)) \mathbb{E}[U_i|U_i \geq u_i(t)] dt + \int_0^T a(y(t)) \mathbb{E}[Y|Y \geq y(t)] dt \right. \\ \left. - \int_0^T P \left((R + R_h) - \left(\sum_{i=1}^M n_i(t)r_i + m(t)r_e \right) \right) dt \right]. \end{aligned} \quad (4.3)$$

Due to Assumptions 4.1 and 4.2, functions $\lambda_i(u_i)$ and $a(y)$ have an inverse which we denote by $u_i(\lambda_i)$ and $y(a)$, respectively. The inverse functions are defined on $[0, \lambda_{i,\max}]$ and $[0, a_0]$, respectively, and are continuous and strictly decreasing. This allows us to use the arrival rates λ_i and a as the SMO's decision variables and write the instantaneous reward rates as $\lambda_i \mathbb{E}[U_i | U_i \geq u_i(\lambda_i)]$ and $a \mathbb{E}[Y | Y \geq y(a)]$.

4.2 Dynamic Programming Formulation

The problem introduced in Section 4.1 is in fact a finite-state, continuous-time, average reward DP problem. Note that the set $\{\mathcal{U}, \mathcal{Y}\} = \{(\mathbf{u}, y) | 0 \leq u_i \leq u_{i,\max}, \forall i; y \leq y_{\max}\}$ of possible price vectors is compact and that all states communicate assuring that there exists a policy that is associated with finite first passage time from any arbitrarily selected state (\mathbf{N}, m) , to another state (\mathbf{N}', m') . The assumption of continuous demand functions implies transition and reward rates that are continuous in the decision variables. Moreover, the reward rate and the expected holding time at each state (\mathbf{N}, m) are bounded functions of (\mathbf{u}, y) , and, so is the total transition rate out of any state. Therefore, standard DP theory results assert that an optimal stationary policy exists.

Since the process $(\mathbf{N}(t), m(t))$ is a continuous-time Markov chain and the total transition rate out of any state is bounded by $\nu = \sum_{i=1}^M (\lambda_{i,\max} + \mu_i \lceil (R + R_h)/r_i \rceil) + (a_{\max} + d \lceil (R + 2R_h)/r_e \rceil)$, we can uniformize this Markov chain and derive the following Bellman equation [Bertsekas, 2005; Bertsekas, 2007]:

$$\begin{aligned}
J^* + h(\mathbf{N}, m) = & \max_{(\mathbf{u}, y) \in \{\mathcal{U}, \mathcal{Y}\}} \left[\sum_{i \in C(\mathbf{N}, m)} \lambda_i(u_i) \mathbb{E}[U_i | U_i \geq u_i] \right. \\
& + \mathbf{1}_{D(N, m)} a(y) \mathbb{E}[Y | Y \geq y] - P\left((R + R_h) - (\mathbf{N}' \mathbf{r} + mr_e)\right) \\
& + \sum_{i \in C(\mathbf{N}, \mathbf{M})} \frac{\lambda_i(u_i)}{\nu} h(\mathbf{N} + \mathbf{e}_i, m) + \sum_{i=1}^M \frac{n_i \mu_i}{\nu} h(\mathbf{N} - \mathbf{e}_i, m) \\
& + \mathbf{1}_{D(N, m)} \frac{a(y)}{\nu} h(\mathbf{N}, m+1) + \frac{md}{\nu} h(\mathbf{N}, m-1) \\
& \left. + \left(1 - \sum_{i \in C(\mathbf{N}, \mathbf{M})} \frac{\lambda_i(u_i)}{\nu} - \sum_{i=1}^M \frac{n_i \mu_i}{\nu} - \mathbf{1}_{D(N, m)} \frac{a(y)}{\nu} - \frac{md}{\nu}\right) h(\mathbf{N}, m) \right].
\end{aligned} \tag{4.4}$$

Here, $C(\mathbf{N}, m) = \{i | (\mathbf{N}' \mathbf{r} + r_i) + mr_e \leq R + R_h\}$ is the set of internal class arrivals that can be admitted in state (\mathbf{N}, m) , and $D(\mathbf{N}, m) = \{\mathbf{N}' \mathbf{r} + (m+1)r_e \leq R + R_h \text{ and } (m+1)r_e \leq 2R_h\}$ describe the conditions under which external RS class arrivals can be admitted to the system. The above Bellman equation has a unique solution J^* and $h(\cdot)$ for an arbitrarily selected special state, say 0 at which we specify the value of the differential cost function, for example $h(\mathbf{0}) = 0$ [Bertsekas, 2005; Bertsekas, 2007]. The scalar J^* stands for the optimal expected social welfare per unit and $h(\mathbf{N}, m)$ denotes the relative reward in state (\mathbf{N}, m) . Solution of Bellman's equation yields an optimal policy that maps any state (\mathbf{N}, m) to the optimal price vector (\mathbf{u}, y) that maximizes the right-hand side of Equation (4.4). Unfortunately, the *curse of dimensionality* stipulates that Bellman's equation is only solvable for a small state space. We therefore seek a near optimal solution that is applicable to

SMO's managing relatively large buildings or neighborhoods with a large population of internal loads.

4.3 Static Pricing Policy

We consider a *static* pricing policy, namely a fixed price vector (\mathbf{u}, y) independent of the system state, for two reasons: (1) the computation effort of solving for optimal dynamic prices increases exponentially in the number of classes and active loads, and (2) good static prices can be constructed tractably and under reasonable conditions lead to reasonable behaved provision of RS. Indeed, under a static pricing policy (\mathbf{u}, y) , the system evolves as a continuous-time Markov chain and the corresponding average welfare is given by:

$$\begin{aligned} J((\mathbf{u}, y)) &= \sum_{i=1}^M \lambda_i(u_i) \mathbb{E}[U_i | U_i \geq u_i] (1 - \mathbb{P}_{\text{loss}}^i[(\mathbf{u}, y)]) \\ &\quad + a(y) \mathbb{E}[Y | Y \geq y] (1 - \mathbb{Q}_{\text{loss}}[(\mathbf{u}, y)]) \\ &\quad - \mathbb{E} \left[P \left((R + R_h) - \left(\sum_i n_i r_i + m r_e \right) \right) \right], \end{aligned} \quad (4.5)$$

where $\mathbb{P}_{\text{loss}}^i[(\mathbf{u}, y)]$ denotes the steady-state probability $\mathbb{P}[\mathbf{N}'\mathbf{r} + r_i + m r_e > R + R_h]$ that an internal class i arrival is rejected, and $\mathbb{Q}_{\text{loss}}[(\mathbf{u}, y)]$ denotes the steady-state probability $\mathbb{P}[\mathbf{N}'\mathbf{r} + (m + 1)r_e > R + R_h \text{ or } (m + 1)r_e > 2R_h]$ that an external RS class arrival is rejected. Moreover, the expected penalty cost is also given by the steady-state probability associated with the same static policy (\mathbf{u}, y) .

The optimal static welfare is defined by

$$J_s = \max_{(\mathbf{u}, y) \in \{\mathcal{U}, \mathcal{Y}\}} J((\mathbf{u}, y)), \quad (4.6)$$

and the following proposition holds.

Proposition 4.1 $J_s \leq J^*$.

Proof: Any static pricing policy (\mathbf{u}, y) is a feasible policy, therefore its average welfare provides a lower bound to the optimal expected welfare. ■

4.4 Optimal Performance Upper Bound

In this section we develop an upper bound on J^* and use it to quantify the static policy's suboptimality.

Using the inverse demand functions $u_i(\lambda_i)$, and internal class i arrival rate λ_i , the instantaneous reward rate is $F_i(\lambda_i) = \lambda_i \mathbb{E}[U_i | U_i \geq u_i(\lambda_i)]$. Similarly, $G(y) = a \mathbb{E}[Y | Y \geq y(a)]$. Assume that the functions F_i and G are *concave*. Let $J_{\mathbf{ub}}$ be the optimal value of the following NLP problem:

$$\begin{aligned}
\max \quad & \sum_i F_i(\lambda_i) + G(a) - P\left((R + R_h) - \left(\sum_i n_i r_i + m r_e\right)\right) \\
\text{s.t.} \quad & \lambda_i = \mu_i n_i, \quad \forall i, \\
& a = dm, \\
& \sum_i n_i r_i + m r_e \leq R + R_h, \\
& m r_e \leq 2R_h.
\end{aligned} \tag{4.7}$$

Remark: The non-negativity constraints $n_i \geq 0$ and $m \geq 0$ are ignored here. Notice that the departure rates μ_i and d are positive, and the arrival rates λ_i and a are also non negative by definition. Thus n_i and m are also non-negative under *well-defined* demand functions.

Proposition 4.2 *If the functions $F_i(\lambda_i)$ and $G(a)$ are concave and $P(\cdot)$ is convex, then $J^* \leq J_{ub}$.*

Proof: Consider an optimal dynamic pricing policy. Without loss of generality, we can assume that the price u_i becomes large enough and the arrival rate $\lambda_i(u_i)$ is equal to zero, whenever the state is such that an internal class i arrival cannot be accepted. Similarly for the external RS class, a equals zero for sufficiently large y . Here, we view λ_i , n_i , and a , m as random variables, and use $\mathbb{E}[\cdot]$, to indicate expectation with respect to the steady-state distribution under this particular policy. At any time we have $\sum_i n_i r_i + m r_e \leq R + R_h$, which implies that $\sum_i \mathbb{E}[n_i] r_i + \mathbb{E}[m] r_e \leq$

$R + R_h$. Similarly, we have $\mathbb{E}[m]r_e \leq 2R_h$. Furthermore, Little's law [Little, 1961] implies that $\mathbb{E}[\lambda_i] = \mu_i \mathbb{E}[n_i]$, and $\mathbb{E}[a] = d\mathbb{E}[m]$. This shows that the $\mathbb{E}[n_i]$, $\mathbb{E}[\lambda_i]$, $i = 1, \dots, M$, and $\mathbb{E}[m]$, $\mathbb{E}[a]$, form a feasible solution to problem (4.7). Using the concavity of F_i and G , the convexity of P and Jensen's inequality

$$\begin{aligned}
J_{ub} &\geq \sum_i F_i(\mathbb{E}[\lambda_i]) + G(\mathbb{E}[a]) - P\left((R + R_h) - \left(\sum_i \mathbb{E}[n_i]r_i + \mathbb{E}[m]r_e\right)\right) \\
&= \sum_i F_i(\mathbb{E}[\lambda_i]) + G(\mathbb{E}[a]) - P\left(\mathbb{E}\left[(R + R_h) - \left(\sum_i n_i r_i + m r_e\right)\right]\right) \\
&\geq \mathbb{E}\left[\sum_i F_i(\lambda_i)\right] + \mathbb{E}\left[G(a)\right] - \mathbb{E}\left[P\left((R + R_h) - \left(\sum_i n_i r_i + m r_e\right)\right)\right] \\
&= J^*.
\end{aligned} \tag{4.8}$$

where the last equality is due to the optimality of the policy under consideration. ■

The optimal solution of NLP (4.7) provides an upper bound for the optimal social welfare. Moreover, if the objective function of (4.7) is concave, the NLP is very easy to solve.

4.5 Asymptotic Behavior

In this section, we consider a number of asymptotic results and discuss how to derive the optimal policy while satisfying additional system behaviour requirements.

4.5.1 Many Small Loads

If R and R_h are large relative to the required power of a typical arrival, we expect that the law of large numbers [Ross, 1996] will dominate, attenuate statistical fluctuations,

and allow us to carry out an essential deterministic analysis. To capture a situation of this nature, we start with a base system characterized by finite capacity R and R_h and finite demand functions $\lambda_i(u_i)$. We then scale the system through a proportional increase of capacity and demand.

More specifcily, let $c \geq 1$ be a scaling factor. The scaled system has resources $R^c + R_h^c$, with $R^c + R_h^c = cR + cR_h$, and demand functions $\lambda_i^c(u_i)$, $a_j^c(y_j)$ given by $\lambda_i^c(u_i) = c\lambda_i(u_i)$ and $a^c(y) = ca(y)$. Note that the other parameters r_i , μ_i , and r_e , d are held fixed. We will use a superscript c to denote various quantities of interest for the scaled system.

In this case, consider the NLP problem (4.7). The upper bound J_{ub}^c is obtained by maximizing

$$\sum_i c\lambda_i(u_i)\mathbb{E}[U_i|U_i \geq u_i] + ca(y)\mathbb{E}[Y|Y \geq y] - P\left((cR + cR_h) - \left(\sum_i \frac{c\lambda_i(u_i)}{\mu_i}r_i + \frac{ca}{d}r_e\right)\right),$$

subject to the constraint

$$\begin{aligned} \sum_i \frac{c\lambda_i(u_i)}{\mu_i}r_i + \frac{ca(y)}{d}r_e &\leq cR + cR_h, \\ \frac{ca(y)}{d}r_e &\leq 2cR_h. \end{aligned}$$

It can seen that, if the penalty function $P(\cdot)$ is linear, then the optimal solution for (4.7), denoted by $\mathbf{u}_{ub}^* = (u_{ub,1}^*, \dots, u_{ub,M}^*)$ and y_{ub}^* , is independent of c , and $J_{ub}^c = cJ_{ub}^1$.

We impose the following assumption on the penalty function.

Assumption 4.3

$$P(x) = Kx, \tag{4.9}$$

for some $K > 0$.

We summarize the above result as follows:

Proposition 4.3 *Under Assumption 4.3, the optimal objective value of (4.7) in the scaled system increases linearly with c , i.e., $J_{ub}^c = cJ_{ub}^1$.*

We are interested in determining the gap between the two bounds derived in Section 4.3 and Section 4.4. We show that in the regime of many small users, the following result holds:

Theorem 4.4 *Assume that functions $F_i(\lambda_i)$ and $G(a)$ are concave, and Assumption 4.3 holds. Then,*

$$\lim_{c \rightarrow \infty} \frac{1}{c} J_s^c = \lim_{c \rightarrow \infty} \frac{1}{c} J^{*,c} = \lim_{c \rightarrow \infty} \frac{1}{c} J_{ub}^c. \tag{4.10}$$

Proof: It holds from Prop. 4.3 that $J_{ub}^c = cJ_{ub}^1$.

Fix some $\epsilon > 0$ and let us consider new static prices u_i^ϵ given by $u_i^\epsilon = u_{ub,i}^* + \epsilon$. Let $J^c(u^\epsilon)$ be the resulting average welfare. For every i such that $\lambda_i(u_{ub,i}^*) > 0$, we have $\lambda_i(u_i^\epsilon) < \lambda_i(u_{ub,i}^*)$. Similarly, we set $y^\epsilon = y_{ub}^* + \epsilon$, and we have $a(y^\epsilon) < a(y_{ub}^*)$. Let n_i^c (respectively, $n_{i,\infty}^c$) be the random variable which is equal to the number of active loads of class i , in steady-state, in the scaled system, under prices u_i^ϵ , with capacity

cR (respectively, with infinite capacity). Similarly define m^c and m_∞^c . We obtain

$$\begin{aligned}
& \mathbb{P} \left[\sum_i r_i n_{i,\infty}^c + r_e m_\infty^c > cR + cR_h - r_{\max} \right] \\
& \leq \mathbb{P} \left[\sum_i r_i n_{i,\infty}^c + r_e m_\infty^c > \sum_i \frac{c\lambda_i(u_{ub,i}^*)r_i}{\mu_i} + \frac{ca(y_{ub}^*)}{d} - r_{\max} \right] \\
& = \mathbb{P} \left[\sum_i r_i \frac{n_{i,\infty}^c}{c} + r_e \frac{m_\infty^c}{c} > \sum_i \frac{\lambda_i(u_{ub,i}^*)r_i}{\mu_i} + \frac{a(y_{ub}^*)}{d} - \frac{r_{\max}}{c} \right], \tag{4.11}
\end{aligned}$$

where $r_{\max} = \max(\max_i r_i, r_e)$.

Note that $n_{i,\infty}^c$ is equal to the number of customers in an $M/M/\infty$ queue with arrival rate $c\lambda_i(u_i^c)$ and service rate μ_i . As $c \rightarrow \infty$, the random variable $n_{i,\infty}^c/c$ converges in probability to $\lambda_i(u_i^c)/\mu_i$, which is less than $\lambda_i(u_{ub,i}^*)/\mu_i$. Similarly, as $c \rightarrow \infty$, the random variable m_∞^c/c converges in probability to $a(y^c)/d$. Therefore, the RHS probability of (4.11) converges to zero.

Next we check the RS capacity constraint (4.2). Similarly,

$$\begin{aligned}
\mathbb{P} \left[r_e m_\infty^c > 2cR_h - r_e \right] & \leq \mathbb{P} \left[r_e m_\infty^c > \frac{ca(y_{ub}^*)}{d} r_e - r_e \right] \\
& = \mathbb{P} \left[\frac{m_\infty^c}{c} > \frac{a(y_{ub}^*)}{d} - \frac{1}{c} \right], \tag{4.12}
\end{aligned}$$

As $c \rightarrow \infty$, by following the same argument as above, the RHS of (4.12) goes to zero.

Now compare n_i^c and m^c with $n_{i,\infty}^c$ and m_∞^c , respectively. Comparing the number of active loads in the two corresponding systems (one with capacity cR and the other with infinite capacity), and by defining the arrival processes on a common probability

space, we conclude that for all sample paths n_i^c is smaller than $n_{i,\infty}^c$. Similarly, for all sample paths, m^c is smaller than m_∞^c

Hence,

$$\begin{aligned}\mathbb{P}_{\text{loss}}^i[(\mathbf{u}^\epsilon, y^\epsilon)] &= \mathbb{P}\left[\sum_j r_j n_j^c > cR + cR_h - r_e m^c - r_i\right] \\ &\leq \mathbb{P}\left[\sum_j r_j n_j^c > cR + cR_h - r_e m^c - r_{\max}\right] \\ &\leq \mathbb{P}\left[\sum_j r_j n_{j,\infty}^c > cR + cR_h - r_e m_\infty^c - r_{\max}\right],\end{aligned}$$

and the arrival rejection probabilities $P_{\text{loss}}^i[(\mathbf{u}^\epsilon, y^\epsilon)]$ converge to zero as well.

$$\begin{aligned}\mathbb{Q}_{\text{loss}}[(\mathbf{u}^\epsilon, y^\epsilon)] &\leq \mathbb{P}\left[r_e m^c > cR + cR_h - \sum_j r_j n_j^c - r_e\right] + \mathbb{P}\left[r_e m^c > 2cR_h - r_e\right] \\ &\leq \mathbb{P}\left[r_e m_\infty^c > cR + cR_h - \sum_j r_j n_{j,\infty}^c - r_{\max}\right] + \mathbb{P}\left[r_e m_\infty^c > 2cR_h - r_e\right],\end{aligned}$$

and the loss probabilities $\mathbb{Q}_{\text{loss}}[(\mathbf{u}^\epsilon, y^\epsilon)]$ also converges to zero.

By (4.5) and (4.6), it follows that

$$\begin{aligned}
\lim_{c \rightarrow \infty} \frac{1}{c} J_s^c &\geq \lim_{c \rightarrow \infty} \frac{1}{c} J^c((\mathbf{u}^\epsilon, y^\epsilon)) \\
&= \sum_i \lambda_i(u_i^\epsilon) \mathbb{E}[U_i | U_i \geq u_i^\epsilon] + a(y^\epsilon) \mathbb{E}[Y | Y \geq y^\epsilon] \\
&\quad - \frac{1}{c} \mathbb{E} \left[P \left((cR + cR_h) - \left(\sum_i n_i^c r_i + m^c r_e \right) \right) \right] \\
&= \sum_i \lambda_i(u_i^\epsilon) \mathbb{E}[U_i | U_i \geq u_i^\epsilon] + a(y^\epsilon) \mathbb{E}[Y | Y \geq y^\epsilon] \\
&\quad - P \left(\frac{1}{c} \mathbb{E} \left[(cR + cR_h) - \left(\sum_i n_i^c r_i + m^c r_e \right) \right] \right) \\
&= \sum_i \lambda_i(u_i^\epsilon) \mathbb{E}[U_i | U_i \geq u_i^\epsilon] + a(y^\epsilon) \mathbb{E}[Y | Y \geq y^\epsilon] \\
&\quad - P \left((R + R_h) - \left(\sum_i \frac{1}{c} \mathbb{E}[n_i^c] r_i + \frac{1}{c} \mathbb{E}[m^c] r_e \right) \right) \\
&= \sum_i \lambda_i(u_i^\epsilon) \mathbb{E}[U_i | U_i \geq u_i^\epsilon] + a(y^\epsilon) \mathbb{E}[Y | Y \geq y^\epsilon] \\
&\quad - P \left((R + R_h) - \left(\sum_i \frac{\lambda_i(u_i^\epsilon)}{\mu_i} r_i + \frac{a(y^\epsilon)}{d} r_e \right) \right).
\end{aligned}$$

This is true for any positive ϵ . We now let ϵ go to zero, in which case u_i^ϵ tends to $u_{ub,i}^*$

and y^ϵ tends to y_{ub}^* . Continuity of the demand function and $P(x)$, imply

$$\begin{aligned}
\lim_{c \rightarrow \infty} \frac{1}{c} J_s^c &\geq \sum_i \lambda_i(u_{ub,i}^*) \mathbb{E}[U_i | U_i \geq u_{ub,i}^*] + a(y_{ub}^*) \mathbb{E}[Y | Y \geq y_{ub}^*] \\
&\quad - P \left((R + R_h) - \left(\sum_i \frac{\lambda_i(u_{ub,i}^*)}{\mu_i} r_i + \frac{a(y_{ub}^*)}{d} r_e \right) \right) \\
&= J_{ub}^1.
\end{aligned} \tag{4.13}$$

Meanwhile, based on Prop. 4.1 and Prop. 4.2, $J_s^c \leq J^{*,c} \leq J_{ub}^c = cJ_{ub}^1$, and the

result follows. ■

In the next two subsections, while staying in the regime of many small loads, we extend the asymptotic optimality results to conform with additional system behavior requirements.

4.5.2 Energy Neutrality

We impose energy neutrality which requires the energy consumption of long-term average cumulative active requests of the external RS class to equal R_h , i.e., $\mathbb{E}[m(t)]r_e = R_h$. We show that energy neutrality can be achieved if the SMO can appropriately influence the demand function of the RS class.

We assume linear demand:

$$\begin{aligned}\lambda_i(u_i) &= \lambda_{i,\max}\left(1 - \frac{u_i}{u_{i,\max}}\right), \\ a(y) &= a_{\max}\left(1 - \frac{y}{y_{\max}}\right).\end{aligned}\tag{4.14}$$

Suppose that the welfare U_i is uniformly distributed on $[0, u_{i,\max}]$ and Y is uniformly distributed on $[0, y_{\max}]$. Then,

$$\begin{aligned}F_i(\lambda_i) &= u_{i,\max}\left(\lambda_i - \frac{\lambda_i^2}{2\lambda_{i,\max}}\right), \\ G(a) &= y_{\max}\left(a - \frac{a^2}{2a_{\max}}\right)\end{aligned}\tag{4.15}$$

are concave in λ_i and a , respectively.

The NLP (4.7) can be now written as:

$$\begin{aligned}
\min \quad & - \sum_i u_{i,\max} \left(\lambda_i - \frac{\lambda_i^2}{2\lambda_{i,\max}} \right) - y_{\max} \left(a - \frac{a^2}{2a_{\max}} \right) \\
& + K \left((R + R_h) - \left(\sum_i \frac{\lambda_i}{\mu_i} r_i + \frac{a}{d} r_e \right) \right) \\
\text{s.t.} \quad & \sum_i \frac{\lambda_i}{\mu_i} r_i + \frac{a}{d} r_e \leq R + R_h, \\
& \frac{a}{d} r_e \leq 2R_h.
\end{aligned} \tag{4.16}$$

For ease of exposition but without loss of generality, we consider next a system involving 2 internal and 1 external RS class.

Note that the NLP problem (4.16) can be re-formulated into the following Quadratic Programming (QP) problem:

$$\begin{aligned}
\min \quad & \frac{1}{2} (\lambda_1, \lambda_2, a) \begin{bmatrix} \frac{u_{1,\max}}{\lambda_{1,\max}} & & \\ & \frac{u_{2,\max}}{\lambda_{2,\max}} & \\ & & \frac{y_{\max}}{a_{\max}} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ a \end{bmatrix} + \begin{bmatrix} -K \frac{r_1}{u_1} - u_{1,\max} \\ -K \frac{r_2}{u_2} - u_{2,\max} \\ -K \frac{r_e}{d} - a_{\max} \end{bmatrix}^T \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ a \end{bmatrix} \\
\text{s.t.} \quad & \begin{bmatrix} \frac{r_1}{\mu_1} & \frac{r_2}{\mu_2} & \frac{r_e}{d} \\ & & \frac{r_e}{d} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ a \end{bmatrix} \leq \begin{bmatrix} R + R_h \\ 2R_h \end{bmatrix}.
\end{aligned} \tag{4.17}$$

The dual of (4.17) is also a QP problem, as formulated in (4.18).

$$\begin{aligned}
\min \quad & \frac{1}{2} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}^T \begin{bmatrix} \frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} & \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} \\ \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} & \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}^T \\
& \begin{bmatrix} R + R_h - K \cdot \left(\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} \right) - \lambda_{1,\max} \frac{r_1}{\mu_1} - \lambda_{2,\max} \frac{r_2}{\mu_2} - a_{\max} \frac{r_e}{d} \\ 2R_h - K \cdot \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} - a_{\max} \frac{r_e}{d} \end{bmatrix} \\
\text{s.t.} \quad & \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \geq 0.
\end{aligned} \tag{4.18}$$

We denote the optimal solution for primal QP (4.17) by $(\lambda_1^*, \lambda_2^*, a^*)$, and the optimal solution for dual QP (4.18) by (q_1^*, q_2^*) .

Under energy neutrality, the long-term average of external RS class loads is R_h , i.e., $r_e \cdot a^*/d = R_h$. Then, $a^* = d \cdot R_h / r_e$, and the second inequality constraint of (4.17) is inactive. By complementary slackness, $q_2^* = 0$. Substituting in (4.18), q_1^* can be obtained:

$$\begin{aligned} & \text{If } R + R_h > K \cdot \frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \lambda_{1,\max} \frac{r_1}{\mu_1} + K \cdot \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \lambda_{2,\max} \frac{r_2}{\mu_2} + K \cdot \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} + a_{\max} \frac{r_e}{d}, \\ & q_1^* = 0; \\ & \text{else, } q_1^* = \left(K \cdot \frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \lambda_{1,\max} \frac{r_1}{\mu_1} + K \cdot \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \lambda_{2,\max} \frac{r_2}{\mu_2} + K \cdot \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} + \right. \\ & \left. a_{\max} \frac{r_e}{d} - (R + R_h) \right) / \left(\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} \right). \end{aligned}$$

Moreover, the optimality conditions of the Primal QP (4.17) are:

$$\begin{aligned} q_1^* &\geq 0, \\ q_2^* &= 0, \\ a^* &= \frac{d \cdot R_h^*}{r_e}, \\ u_{1,\max} \left(1 - \frac{\lambda_1^*}{\lambda_{1,\max}} \right) &= (q_1^* - K) \frac{r_1}{\mu_1}, \\ u_{2,\max} \left(1 - \frac{\lambda_2^*}{\lambda_{2,\max}} \right) &= (q_1^* - K) \frac{r_2}{\mu_2}, \\ y_{\max} \left(1 - \frac{a^*}{a_{\max}} \right) &= (q_1^* - K) \frac{r_e}{d} + q_2^* \frac{r_e}{d}. \end{aligned} \tag{4.19}$$

Notice that the left hand sides of equalities in (4.19) are non-negative. In order for these equalities to hold, it is required that $q_1^* \geq K$, which in turn enforces the

following conditions:

$$\begin{aligned}
R + R_h &\leq K \cdot \frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \lambda_{1,\max} \frac{r_1}{\mu_1} + K \cdot \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} \\
&\quad + \lambda_{2,\max} \frac{r_2}{\mu_2} + K \cdot \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2} + a_{\max} \frac{r_e}{d}, \\
q_1^* - K &= \frac{\lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} + a_{\max} \frac{r_e}{d} - (R + R_h)}{\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2}} \\
&= \frac{\mu_1}{r_1} \cdot u_{1,\max} \left(1 - \frac{\lambda_1^*}{\lambda_{1,\max}}\right) \\
&= \frac{\mu_2}{r_2} \cdot u_{2,\max} \left(1 - \frac{\lambda_2^*}{\lambda_{2,\max}}\right) \\
&= \frac{d}{r_e} \cdot y_{\max} \left(1 - \frac{1}{a_{\max}} \cdot \frac{d \cdot R_h^*}{r_e}\right).
\end{aligned}$$

Or,

$$\begin{aligned}
R + R_h &\leq \lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} + a_{\max} \frac{r_e}{d}, \\
R &\leq \lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2}, \\
y_{\max} \left(1 - \frac{1}{a_{\max}} \cdot \frac{d \cdot R_h}{r_e}\right) &= \frac{\lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} - R}{\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2}} \cdot \frac{r_e}{d}. \tag{4.20}
\end{aligned}$$

If (4.20) holds, by complementary slackness, the first inequality of (4.17) is active and there is no penalty cost in the optimal objective function, i.e., the optimal social welfare is:

$$\begin{aligned}
&-\frac{1}{2} \frac{\left(\lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} + a_{\max} \frac{r_e}{d} - (R + R_h)\right)^2}{\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2}} \\
&+ \frac{1}{2} \lambda_{1,\max} u_{1,\max} + \frac{1}{2} \lambda_{2,\max} u_{2,\max} + \frac{1}{2} a_{\max} y_{\max}. \tag{4.21}
\end{aligned}$$

We summarize the above result as follows:

Proposition 4.5 *Given (4.20), in the regime of many small loads, when applying the static pricing policy derived from (4.16), the long-term average capacity usage of the external RS class is R_h , and the optimal performance is given by (4.21).*

Remark: Notice that we assume linear demand $y(a) = y_{\max}(1 - \frac{a}{a_{\max}})$, where y_{\max} is the intercept and y_{\max}/a_{\max} is the slope. (4.20) implies that for the long-term average of $m(t)r_e$ to equal R_h , the maximal price, y_{\max} , and the reciprocal of elasticity, y_{\max}/a_{\max} , must satisfy (4.20), i.e., the optimal arrival rate and price pair (4.22) lies on the linear demand curve.

$$\begin{aligned} a^* &= \frac{d \cdot R_h}{r_e} \\ y^* &= \frac{\lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} - R}{\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2}} \cdot \frac{r_e}{d}. \end{aligned} \quad (4.22)$$

4.5.3 Optimal Selection of R and R_h

We have so far considered R and R_h as given market transactions determined at the long time scale and focused on the operational decisions of the SMO that affect short time scale behavior resulting in energy neutrality over the long time scale. We now focus on the SMO's optimal selection of the long time scale market transactions setting average load R and RS reserves R_h .

We adopt all Section 4.5.2 definitions and introduce R_0 , an estimate of the desired consumption of the building in the event that no RS is offered. Given a feasible setting (R, R_h) with $R \geq R_0$ and $R_h \geq 0$, we incur two penalty costs: (1) $\frac{1}{2}\kappa_1(R - R_0)$ with

$\kappa_1 > 0$ to penalize deviation from the desired consumption, and (2) $\frac{1}{2}\kappa_2 R_h$ with $\kappa_2 > 0$ to model the intangible (inconvenience) and tangible (control and actuation) costs of responding to offering regulation service requests.

Based on Proposition 4.5, in the regime of many small loads, for any feasible consumption and RS pair (R, R_h) , the optimal social welfare consistent with the condition of energy neutrality on the average is given by (4.21). The optimal (R, R_h) pair can be obtained by solving the following Quadratic Programming problem:

$$\begin{aligned}
\min \quad & \frac{1}{2} \frac{\left(\lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} + a_{\max} \frac{r_e}{d} - (R + R_h) \right)^2}{\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2} + \frac{a_{\max}}{y_{\max}} \cdot \frac{r_e^2}{d^2}} \\
& + \frac{1}{2} \kappa_1 (R + R_h - R_0) + \frac{1}{2} \kappa_2 R_h \\
\text{s.t.} \quad & R + R_h \leq \lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} + a_{\max} \frac{r_e}{d}, \\
& R \leq \lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2}, \\
& y_{\max} \left(1 - \frac{1}{a_{\max}} \cdot \frac{d \cdot R_h}{r_e} \right) = \frac{\lambda_{1,\max} \frac{r_1}{\mu_1} + \lambda_{2,\max} \frac{r_2}{\mu_2} - R}{\frac{\lambda_{1,\max}}{u_{1,\max}} \cdot \frac{r_1^2}{\mu_1^2} + \frac{\lambda_{2,\max}}{u_{2,\max}} \cdot \frac{r_2^2}{\mu_2^2}} \cdot \frac{r_e}{d}, \\
& R + R_h \geq R_0, \\
& R_h \geq 0.
\end{aligned} \tag{4.23}$$

The first term in the objective function models the impact on the optimal social welfare while the last two terms represent the penalty costs described above. The first three constraints impose energy neutrality over the long time scale, and the last two constraints ensure that the (R, R_h) pair is feasible.

We summarize the above result as follows:

Proposition 4.6 *The optimal average building load and regulation service reserve pair (R^*, R_h^*) can be obtained by solving (4.23).*

4.6 Numerical Experiments

In this section, we report numerical experiments that verify and validate our results.

4.6.1 Tracking of ISO Requests

Assume that the SMO can support a maximal consumption of 1200 KW with $R = 1000$ KW and $R_h = 200$ KW. This consumption is consistent with the Boston University (BU) Photonics building housing the office of the author of this dissertation. Consider two internal classes characterized by (all arrival rates are in arrivals/minute and departure rates in departures/minute): $\lambda_1(u_1) = 1600 - 80u_1$, $\lambda_2(u_2) = 800 - 80u_2$, $u_{1,\max} = 20$, $u_{2,\max} = 10$, $\lambda_{1,\max} = 1600$, $\lambda_{2,\max} = 800$, $r_1 = 2$ KW, $r_2 = 1$ KW, $\mu_1 = 1$, $\mu_2 = 2$. The RS class arrival rate is: $a(y) = 1000(1 - y/y_{\max})$ with y_{\max} to be determined, $a_{\max} = 1000$, $r_e = 1$ KW, $d = 2$. The penalty function has a slope of $K = 1000$. Assume that the social welfare U_i is uniformly distributed on $[0, u_{i,\max}]$ and Y is uniformly distributed on $[0, y_{\max}]$. With these values we can solve the NLP problem (4.16) and obtain asymptotically optimal static prices.

Consider a typical regulation service cycle consisting of three 5-minute periods. Each cycle starts with a full RS standby state, namely, with all RS active loads totalling R_h . This is the result of the ISO 5 minute dispatch which we model by

tuning the value of y_{\max} . In the following two periods within the cycle, ISO requests are modeled as random samples from a uniform distribution over $[0, 2R_h]$ which are instantiated by setting the corresponding value of y_{\max} . This random cycle is statistically energy neutral over the long time scale corresponding to an average energy consumption of RS requests equal to R_h . In this experiment, y_{\max} changes every 5 minutes and the SMO must control internal class loads to meet ISO requests within the 5 minute response requirement. By formulating and solving the NLP problem (4.16) at the beginning of every period, the SMO is able to appropriately set the prices that result in the required arrivals of internal classes. We simulate the system for the long time scale of one hour consisting of 12 periods of 5 minutes each and report the results below.

The steady-state arrival rates for the two internal classes and the RS class in these periods are shown in Tab. 4.1.

The evolution of the total consumption due to internal loads and the total load of the RS class are shown in Fig. 4.2. Note that by applying static pricing policies that are piece-wise constant over each 5-minute period, internal loads converge to the ISO request. Recalling that RS reserves are required to respond with a ramp of $R_h/5$ KW per minute, the response of internal class loads conforms well to requirements. Indeed, since $R_h = 200$ KW in this example, the rate at which $n_1(t)r_1 + n_2(t)r_2 + m(t)r_e$ move away from and then approach the 1200 KW level should be close to 40 KW per minute. Figure 4.2 demonstrates this to be the case. The SMO's decision to offer

Table 4.1: The Arrival Rates of Internal Classes and the RS Class.

Phase	Internal class 1	Internal class 1	The RS class
1	376	494	400
2	409	502	258
3	346	486	527
4	376	494	400
5	309	477	683
6	409	502	257
7	376	494	400
8	322	480	630
9	445	511	106
10	376	494	400
11	403	500	286
12	321	480	635

200 KW of RS is consistent with its capability to perform according to the associated contractual requirements. In Figure 4-3, where we plot the number of internal loads and RS requests, we note that there are on average 350 active loads of class 1 with a 2 KW consumption rate – these might be HVAC heating zone loads – and 250 active loads of class 2 with a 1 KW consumption rate. These quantities are consistent with the BU Photonics building which features several hundred heating zones.

4.6.2 Long Term Energy Neutrality

In this section, we view the dynamics of the system in a longer time horizon and investigate how to maintain energy neutrality over that horizon by controlling the arrival rates of the RS class. The whole system is modeled as in Section 4.6.1.

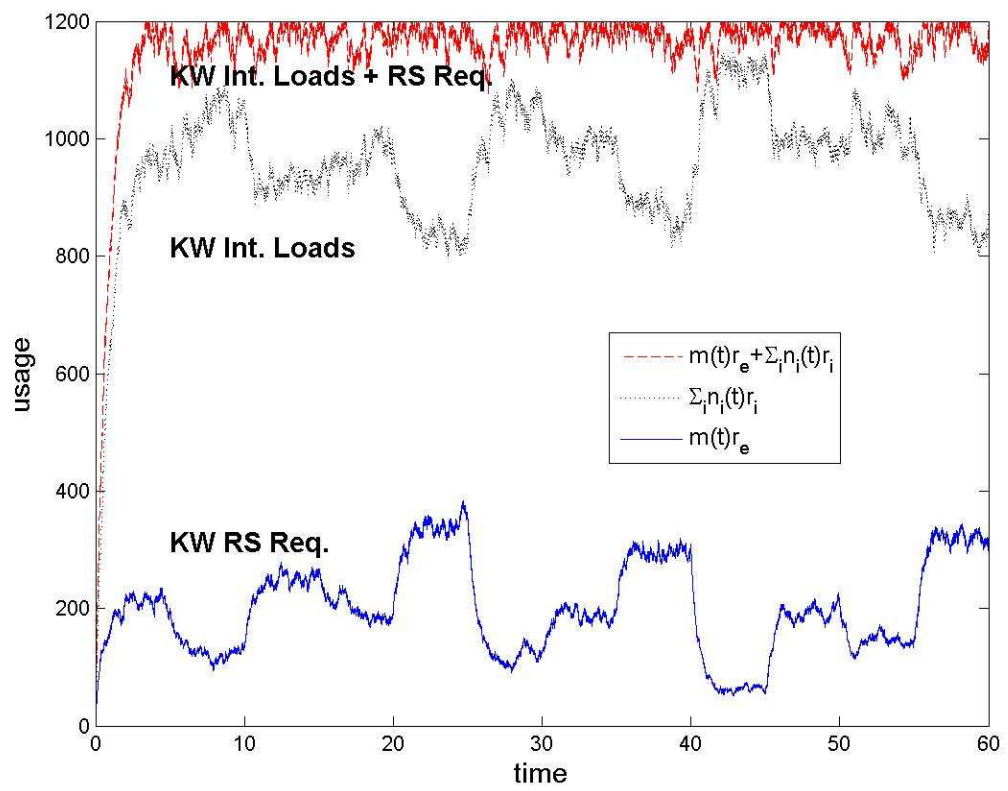


Figure 4.2: Energy consumption by internal classes and active RS requests.

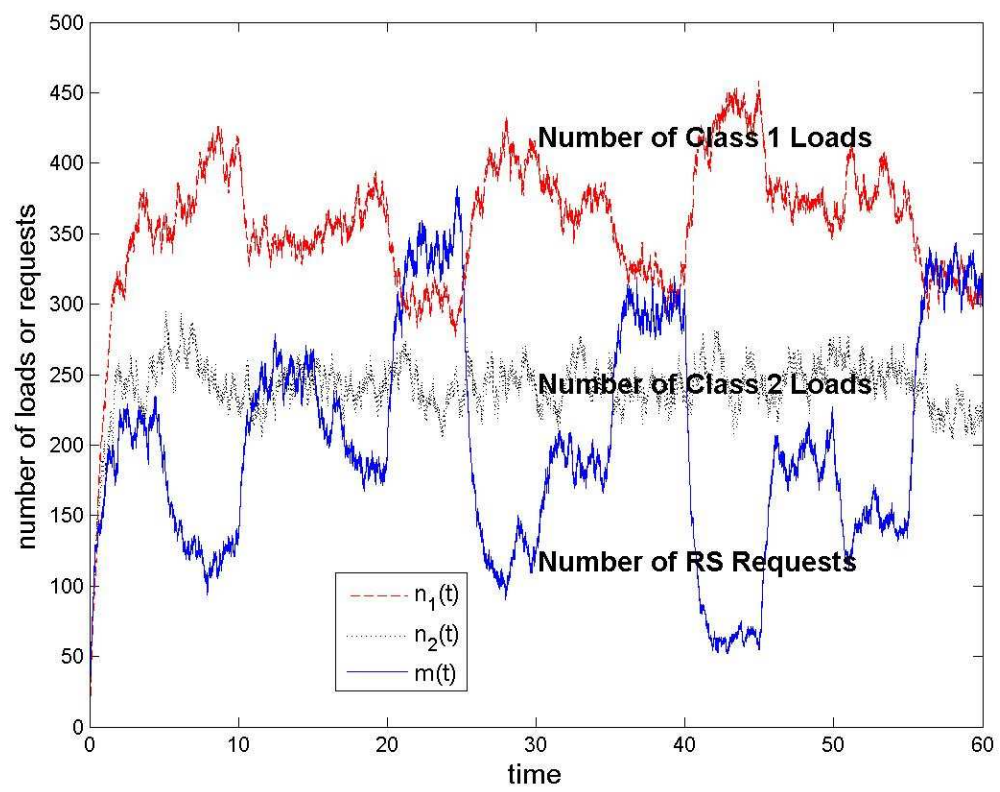


Figure 4-3: Number of active internal loads and active RS requests.

Based on Proposition 4.5, in order to achieve long-term energy neutrality, the ISO, which is in fact obligated to maintain energy neutrality, will set its utility parameters so that Eq. (4.20) holds, i.e.,

$$y_{\max}(1 - \frac{400}{a_{\max}}) = \frac{2600}{340} \cdot \frac{1}{2}.$$

Therefore, in our numerical example where $a_{\max} = 1000$, we must have $y_{\max} = 6.3725$ to maintain energy neutrality. We next formulate the NLP problem as in (4.16). The optimal arrival rates are $\lambda_1^* = 376$, $\lambda_2^* = 494$, $a^* = 400$, and the average numbers of active loads in steady-state are $n_1^* = 376$, $n_2^* = 247$, $m^* = 200$ for two internal classes and the RS class, respectively. We simulate the system for 240 minutes and show the results below.

First of all, as shown in Fig. 4-4, the average energy consumption of active RS requests is approximately 195, which is very close to R_h . Therefore, energy neutrality is statistically verified in this example. Also, the average energy consumption of active internal loads and active RS requests is about 1160, which is approximately equal to the consumption capacity $R + R_h$, i.e., there is no consumption capacity that remains unutilized. The SMO controls building loads to fully utilize consumption capacity for either meeting internal demand or providing RS reserves. Secondly, it can be concluded from Fig. 4-5 that the SMO is adequately responsive to external markets by self-adjusting active internal loads in a very short time. The average numbers of active internal loads and active RS requests present in the system are 361, 242, and

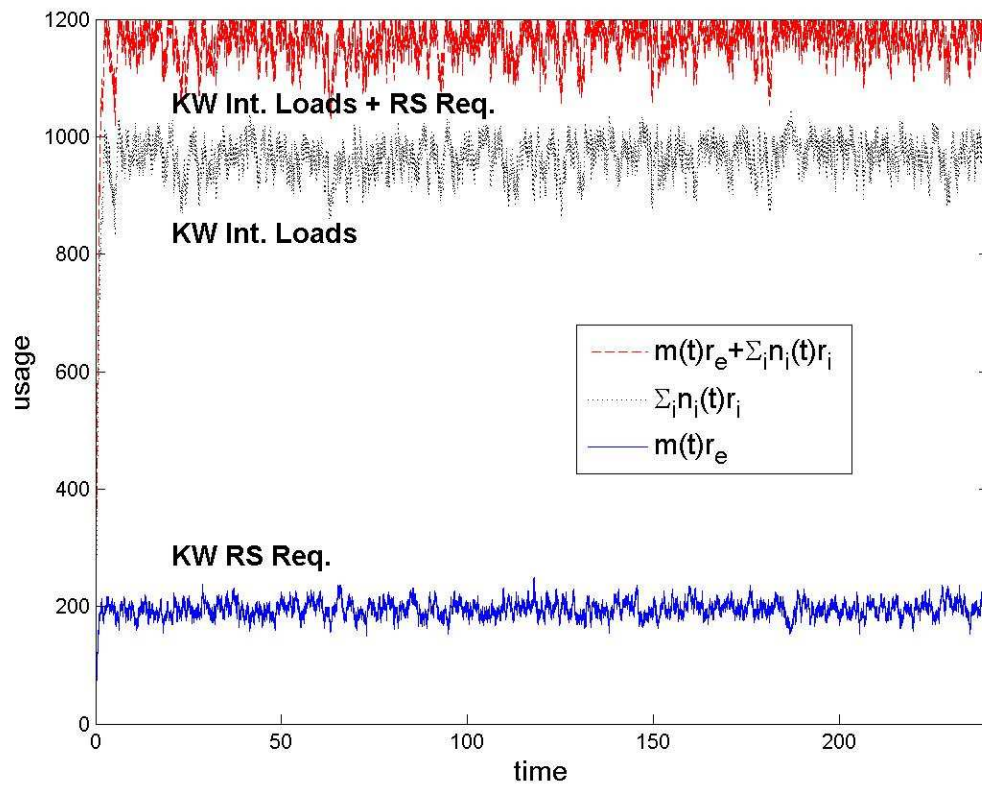


Figure 4.4: Energy consumption by internal classes and active RS requests.

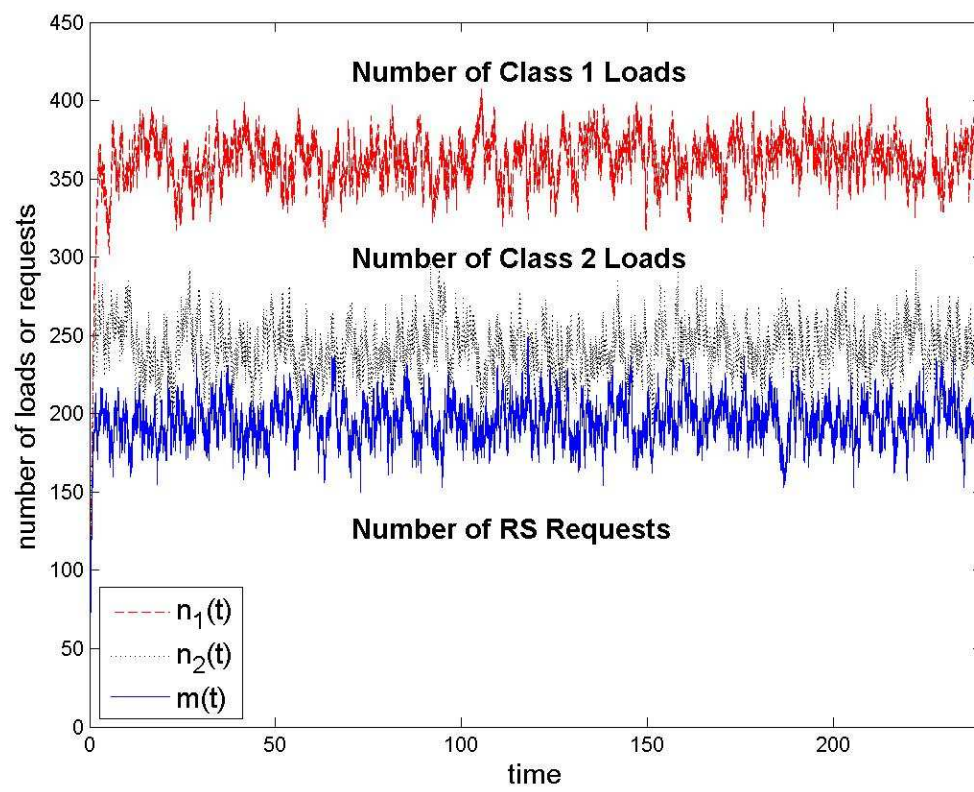


Figure 4-5: Number of active internal loads and active RS requests.

195, respectively. These numbers are consistent, i.e., they are very close to the NLP derived solution of 376, 247, and 200.

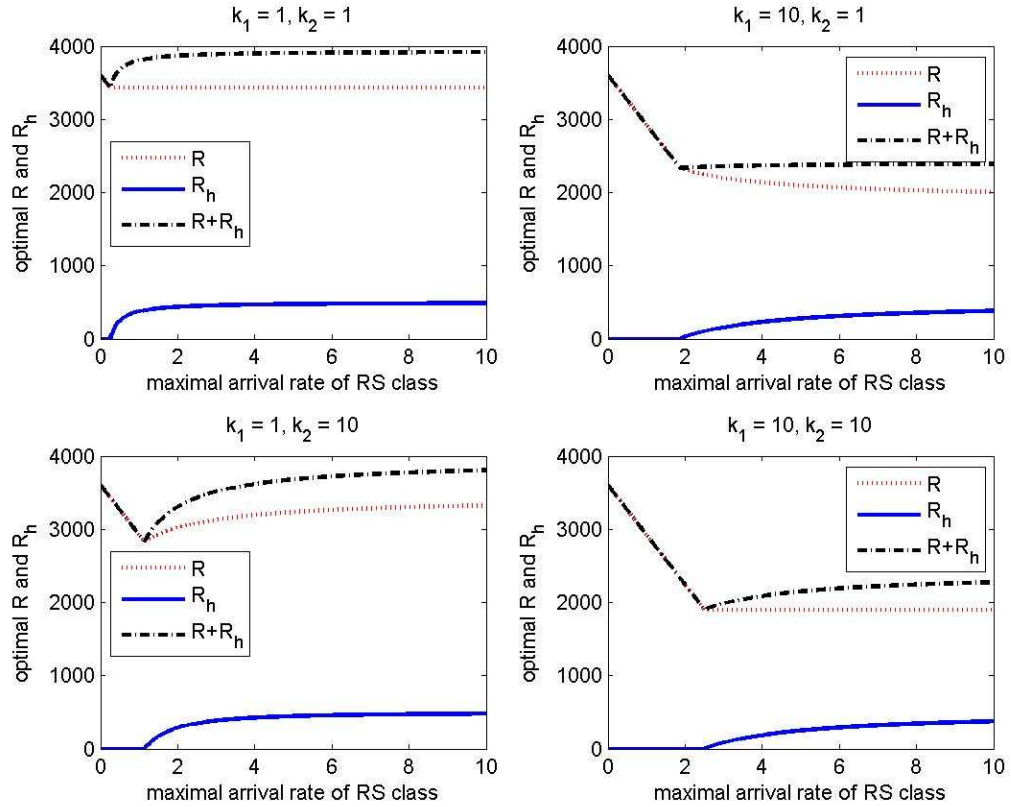
4.6.3 Optimal Selection of R and R_h

In this section, we investigate the case of optimally setting R and R_h . The considered two internal classes and the RS class are modeled as in Section 4.6.2. We assume the nominal energy consumption $R_0 = 1000$ and consider four different settings of penalty function slopes as shown in Tab. 4.2.

We recognize that each distinct pair of (a_{\max}, y_{\max}) results in a distinct optimal pair (R^*, R_h^*) when we solve the NLP problem (4.23). In the numerical experiment reported below, we set $a_{\max} = 1000$ and gradually increase y_{\max} from 0 to 10 in order to examine how different y_{\max} affects the optimal pair (R^*, R_h^*) . Simulation results are shown in Fig. 4-6. Several conclusions can be drawn. When the maximal price (correspondingly the utility) of the ISO's RS requests, y_{\max} , is relatively small, the SMO would rather not provide RS reserves. In such scenarios, the energy neutrality conditions enforce that the optimal R is a decreasing function of y_{\max} . However, when the value of y_{\max} is above a certain threshold, the SMO enjoys a utility level that compensates building occupants for the inconvenience and costs of providing the ISO with RS reserves. As a result, the optimal value of R_h increases with y_{\max} . Moreover, the threshold that determines whether the SMO provides RS reserves, is controlled by the penalty function slopes κ_1 and κ_2 . The larger κ_1 and κ_2 are, the greater the threshold is.

Table 4.2: Penalty Function Slope Settings.

Setting	κ_1	κ_2
1	1	1
2	10	1
3	1	10
4	10	10

**Figure 4-6:** Optimal selection of R and R_h for different RS class maximal arrival rates.

Chapter 5

Conclusion

This dissertation investigates the optimization problem of energy consumption in two areas: topology control in WSNETs and regulating electricity demand in large smart buildings. We now summarize the conclusions and future research directions for these two topics.

5.1 Topology Control in WSNETs

We considered the problems of constructing energy efficient topologies in WSNETs which are preferable for running consensus/averaging algorithms. We first considered the problem of constructing a minimum power bidirectional spanning tree for the case of static networks. Construction of the tree is an NP-complete problem. We provided a mixed integer programming formulation and several methods for constructing good feasible solutions. Two of our methods can be implemented in a distributed manner and are based on running well known graph algorithms (shortest path, minimum weight spanning tree) on an appropriately designed augmented graph. Computationally speaking, both methods are very efficient and yield solutions in the order of minutes for moderate instances. Furthermore, an illustrative set of numerical results

shows that the solutions we obtain are close to optimal.

Secondly, in regards to the dynamic networks case, we considered the problem of constructing an interconnected network using minimal energy. The problem was posed as a dynamic programming problem and a number of structural properties were established. We studied approximate/suboptimal algorithms that are more suitable for large-scale instances. To develop these approximate algorithms, we first considered a scenario in which a connected network needs to be built over a large enough horizon. In such regime, we established that a policy that solves a minimum spanning tree problem is optimal. In the more general scenario in which the horizon is not large enough, we developed a rollout algorithm which leverages the MST-based solutions. Numerical results imply that the proposed rollout algorithm is effective and efficient enough to handle large problems.

We note that WSNETs have great theoretical and practical potential. From a theoretical perspective, WSNETs provide an innovative platform where optimization and control theory can play an important role, and present new challenges to researchers. From an application perspective, the flexibility, fault tolerance, self-configuration, low-cost and rapid deployment characteristics of WSNETs create a wide range of innovative applications that will greatly facilitate our daily lives. We especially point out that Wireless Sensor Body Area Networks are expected to generate a lot of interests in next decade [Latré et al., 2011]. However, revolutionary breakthroughs in battery research are not in sight, energy conservation remains a primary issue in

WSNETs. Energy efficient protocols and schemes have very broad research prospect in every aspect of a WSNET.

Since [Tsitsiklis, 1984], (distributed) averaging/consensus has received considerable attention, especially in the past decade after the emergence of WSNETs. Consensus/averaging builds a bridge linking classical optimization and control theory and innovative applications in WSNETs. However, most research has been focused on static networks and investigates topologies with fast convergence rates. These topologies are generally stationary and sometimes require global information for construction, which make them not suitable for WSNET applications. We note that future work should consider consensus/averaging in dynamic environments and address the issues of distributed computation and implementation.

5.2 Regulation Service Reserves by Smart Buildings

The advent and development of the smart grid will provide great opportunities for smart buildings to participate in power markets. We derive the optimal policy for a smart building electricity operator to modulate its electricity consumption and offer valuable regulation service reserves to a wholesale power market operator. To this end, we introduced a social welfare maximization model and formulated a stochastic DP problem. Recognizing that an optimal dynamic policy is not tractable for realistic problem size instances, we explored a tractable alternative by (i) deriving bounds on the optimal performance, (ii) proposing a static policy that we show to

be asymptotically optimal, and (iii) extending the asymptotic optimality result to apply to the realistic requirements that a smart building operator must deal with. We finally provided numerical simulations to demonstrate that the proposed static policy is computationally efficient and responds quickly to accommodate fast reserve requests from external power markets.

With the smart grid and smart buildings, higher energy efficiency levels will become possible along the power grid by exploiting detailed and close to real-time information available across the grid and throughout smart building microgrids. Enabling buildings to offer capacity reserves and compete with centralized generators in the power markets promises a major contribution to infrastructure resilience and lower CO₂ emissions due to increased efficiency and accommodation of intermittent, yet clean generation. Because the smart grid and smart buildings are newly-emerging and fast-growing, little work has been done on demand control aiming at the provision of capacity reserves. We want to emphasize that optimization and control theory can have a significant impact in this fascinating area.

References

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38:393–422.
- Anastasi, G., Conti, M., Francescoand, M. D., and Passarella, A. (2009). Energy conservation in wireless sensor networks: a survey. *Ad Hoc Networks*, 7:537–568.
- Aslam, J., Li, Q., and Rus, D. (2003). Three power-aware routing algorithms for sensor networks. *Journal of Wireless Communications and Mobile Computing*, 3(2):187–208.
- Atkinson, R. D. and Castro, D. (2008). Digital quality of life: Understanding the peronal and social benefits of the IT revolution. Technical report, The Information and Technology and Innovation Foundation, Washington, DC.
- Bertsekas, D., Tsitsiklis, J., and Wu, C. (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3:245–262.
- Bertsekas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA.
- Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 3rd edition.
- Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, 3rd edition.
- Bertsekas, D. P. and Gallager, R. (1992). *Data Networks*. Prentice-Hall, 2nd edition.
- Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall.
- Blondel, V., Hendrickx, J., Olshevsky, A., and Tsitsiklis, J. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC05)*, pages 2996–3000, Seville, Spain.
- Blough, D. M., Leoncini, M., Resta, G., and Santi, P. (2002). On the symmetric range assignment problem in wireless ad hoc networks. In *Proceedings of the 2nd IFIP*

- International Conference on Theoretical Computer Science*, pages 71–82, Deventer, The Netherlands.
- Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *joint issue of the IEEE Transactions on Information Theory and IEEE/ACM Transactions on Networking*, 52(6):2508–2530.
- Bryson, M. (2007). *PJM Manual 12: Balancing Operations, Revision 16*.
- Bryson, M. (2008). *PJM Manual 11: Scheduling Operations, Revision 34*.
- Buonadonna, P., Gay, D., Hellerstein, J. M., Hong, W., and Madden, S. (2005). Task: Sensor network in a box. In *Proceedings of European Workshop on Sensor Networks (EWSN 2005)*.
- Cao, M., Spielman, D., and Morse, A. (2005). A lower bound on convergence of a distributed network consensus algorithm. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC05)*, Seville, Spain.
- Chang, J. H. and Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(4):609–619.
- Clementi, A. E. F., Penna, P., and Silvestri, R. (1999). Hardness results for the power range assignment problems in packet radio networks. In *Proceedings of the 3rd International Workshop on Randomization and Approximation in Computer Science (APPROX 1999), Lecture Notes in Computer Science*, volume 1671, pages 195–208.
- Dam, T. V. and Langendoen, K. (2003). An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA.
- DARPA-ATO (2002). Self-healing mines. <http://www.darpa.mil>.
- Das, A. K., Marks, R. J., El-Sharkawi, M., Arabshahi, P., and Gray, A. (2004a). Optimization methods for minimum power bidirectional topology construction in wireless networks with sectorized antennas. In *Proceedings of the 47th IEEE Global Communications Conference (GLOBECOM 2004)*, Dallas, Texas.
- Das, A. K., Marks, R. J., El-Sharkawi, M., Arabshahi, P., and Gray, A. (2004b). Optimization methods for minimum power multicasting in wireless networks with sectorized antennas. In *Proceedings of the IEEE Wireless Communications and Network Conference (WCNC 2004)*, Atlanta, Georgia.

- DeGroot, M. (1974). Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121.
- Dietrich, I. and Dressler, F. (2009). On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, 5:5:1–5:39.
- Dimakis, A. G., Sarwate, A. D., and Wainwright, M. J. (2006). Geographic gossip: Efficient aggregation for sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (ISPN 2006)*, Nashville, TN.
- Dominguez, T., Torre, M. D. L., Juberias, G., Prieto, E., and Cordoba., A. (2007). Wind generation issues in spain and real-time production control. In *CIGRE Conference*, Zagreb, Croatia.
- Energy Reliability Council of Texas (ERCOT) (2007–2010a). Balancing energy services market clearing prices for energy annual reports archives. <http://www.ercot.com/mktinfo/prices/mcpea>.
- Energy Reliability Council of Texas (ERCOT) (2007–2010b). Day-ahead ancillary market clearing prices for capacity archives. <http://www.ercot.com/mktinfo/prices/mcpca>.
- European Commission (2006). European smartgrids technology platform: vision and strategy for europe’s electricity networks of the future. Technical Report EUR 22040, European Commission, Brussels, Belgium.
- Farella, E., Pieracci, A., Benini, L., Rocchi, L., and Acquaviva, A. (2008). Interfacing human and computer with wireless body area sensor networks: the wimoca solution. *Multimedia Tools and Applications*, 38(3):337–363.
- Frederickson, G. N. (1985). A single source shortest path algorithm for a planar distributed network. *Lecture Notes in Computer Science*, 182:143–150.
- Fujisawa, K., Kojima, M., Nakata, K., and Yamashita, M. (2005). *SDPA-M (SemiDefinite Programming Algorithm in Matlab) User’s Manual - Version 6.2.0*.
- Gallager, R. G., Humblet, P. A., and Spira, P. M. (1983). A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77.
- Gomez, J., Campbell, A. T., Naghshineh, M., and Bisdikian, C. (2002). Power-aware routing in wireless packet networks. In *Proceedings of IEEE International Workshop on Mobile Multimedia Communications (MoMuC’99)*, pages 380–383.

- Hohlt, B. and Brewer, E. (2006). Network power scheduling for tinyos applications. In *Proceedings of 2nd IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS '06)*, pages 443–462, San Francisco, CA.
- Huang, F. K., Richards, D. S., and Winter, P. (1992). *The Steiner Tree Problem*. North-Holland.
- Jadbabaie, A., Lin, J., and Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.
- Jiang, L. and Walrand, J. (2009). Approaching throughput-optimality in a distributed csma algorithm: Collisions and stability. In *Proceedings of the 2009 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2009)*, pages 5–8.
- Joskow, P. L. (2006). Markets for power in the united states: An interim assessment. *The Energy Journal*, 38:1–36.
- Joskow, P. L. (2008). Challenges for creating a comprehensive national electricity policy. Technical Report MIT CEEP 08-019WP, Center for Energy and Environmental Policy Research, MIT.
- Juang, P., Oki, H., Wang, Y., Martonosi, M., shiuan Peh, L., and Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of the 10th international conference on Architecture support for programming languages and operating systems (ASPLOS-X)*, San Jose, CA.
- Kirousis, L. M., Kranakis, E., Krizanc, D., and Pelc, A. (2000). Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305.
- Kranz, B., Pike, R., and Hirst, E. (2003). Integrated electricity markets in New York. *The Electricity Journal*, 16(2):54–65.
- Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50.
- Latr  , B., Braem, B., Moerman, I., Blondia, C., and Demeester, P. (2011). A survey on wireless body area networks. *Wireless Networks*, 17(1):1–18.
- Li, F., Padhy, N. P., and Wang, J. (2008). Cost-benefit reflective distribution charging methodology. *IEEE Transactions on Power Systems*, 23:58–64.

- Li, M. and Yang, B. (2006). A survey on topology issues in wireless sensor network. In *Proceedings of the 2006 International Conference on Wireless Networks (ICWN 2006)*, Las Vegas, Nevada.
- Little, J. D. C. (1961). A proof of the queuing formula: $L = \lambda W$. *Operations Research*, 9:383–387.
- Lu, G., Krishnamachari, B., and Raghavendra, C. S. (2004). An adaptive energy-efficient and low-latency mac for data gathering. In *Proceedings of the 18th International parallel and Distributed Processing Symposium*, pages 26–30.
- Madden, S., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2002). Tag: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of Annual Symposium on Operation Systems Design and Implementation (OSDI)*.
- Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). Tinydb: An acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173.
- Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02)*, pages 88–97, Atlanta, Georgia, USA.
- Makarov, Y. V., Loutan, C., Ma, J., and de Mello, P. (2009). Operational impacts of wind generation on california power systems. *IEEE Transactions on Power Systems*, 24(2):1039–1050.
- Malan, D., Fulford-jones, T., Welsh, M., and Moulton, S. (2004). Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International Workshop on Wearable and Implantable Body Sensor Networks*, London, UK.
- Maroti, M., Simon, G., Ledeczi, A., and Sztipanovits, J. (2004). Shooter localization in urban terrain. *IEEE Computer*, 37:60–61.
- Mehyar, M., Spanos, D., Pongsajapan, J., Low, S. H., and Murray, R. M. (2005). Distributed averaging on asynchronous communication networks. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC05)*, pages 2996–3000, Seville, Spain.
- Moallemi, C. and Roy, B. V. (2005). Consensus propagation. In *Proceedings of the 19th Neural Information Processing Systems (NIPS 2005)*, Vancouver, Canada.
- Montemanni, R. and Gambardella, L. M. (2005). Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. *Computers and Operations Research*, 32:2891–2904.

- Nama, H. and Mandayam, N. (2005). Sensor networks over information fields: optimal energy and node distributions. In *Proceedings of IEEE Wireless Communication and Networking Conference (WCNC 2005)*, volume 3, pages 1842–1847.
- Narayanaswamy, S., Kawadia, V., Sreenivas, R. S., and Kumar, P. R. (2002). Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *Proceedings of European Wireless Conference*, pages 156–162.
- Ning, X. and Cassandras, C. G. (2009). On maximum lifetime routing in wireless sensor networks. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3757–3762.
- NIST (2010). *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0*.
- NYISO (2000). *NYISO Ancillary Services Manual 2*.
- NYISO (2001). *NYISO Day-Ahead Scheduling Manual 11*.
- Olfati-Saber, R. and Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533.
- Olshevsky, A. and Tsitsiklis, J. N. (2009). Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55.
- Ott, A. L. (2003). Experience with PJM market operation, system design, and implementation. *IEEE Transactions on Power Systems*, 18(2):528–534.
- Ott, A. L. (2008). Implementation of demand response in the PJM synchronized reserve market. In *CIGRE Paris Session and Technical Exhibition*, Paris, France.
- Paruchuri, V., Basavaraju, S., Duresi, A., Kannan, R., and Iyengar, S. S. (2004). Random asynchronous wakeup protocol for sensor networks. In *Proceedings of 1st International Conference on Broadband Networks (BROADNETS 2004)*, pages 25–29.
- Paschalidis, I. C. and Chen, Y. (2010). Statistical anomaly detection with sensor network. *ACM Transactions on Sensor Networks*, 7(2):17:1–17:23.
- Paschalidis, I. C. and Guo, D. (2009). Robust and distributed stochastic localization in sensor networks: Theory and experimental results. *ACM Transactions on Sensor Networks*, 5(4):1–22.

- Paschalidis, I. C. and Lai, W. (2008). Optimally balancing energy consumption versus latency in sensor network routing. *ACM Transactions on Sensor Networks*, 4(4):21:1–21:28.
- Paschalidis, I. C., Lai, W., and Song, X. (2009a). Optimized scheduled multiple access control for wireless sensor networks. *IEEE Transactions on Automatic Control*, 54(11):2573–2585.
- Paschalidis, I. C., Lai, W., and Starobinski, D. (2007). Asymptotically optimal transmission policies for large-scale low-power wireless sensor networks. *IEEE/ACM Transactions on Networking*, 15(1):105–118.
- Paschalidis, I. C. and Li, B. (2009). On energy optimized averaging in wireless sensor networks. In *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC-CCC09)*, pages 7137–7144, Shanghai, China.
- Paschalidis, I. C. and Li, B. (2011a). Energy optimized topologies for distributed averaging in wireless sensor networks. *IEEE Transactions on Automatical Control*. accepted for publication.
- Paschalidis, I. C. and Li, B. (2011b). On energy optimized network construction for distributed averaging in a dynamic environment. In *The 18th IFAC World Congress*, Milano, Italy. accepted for publication.
- Paschalidis, I. C., Li, B., and Caramanis, M. C. (2011). A market-based mechanism for providing demand-side regulation service reserves. In *The joint 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC11)*, Orlando, Florida. submitted for publication.
- Paschalidis, I. C., Li, K., Moazzez-Estanjini, R., Lin, Y., and Guo, D. (2009b). Intelligent forklift dispatching in warehouses using a sensor network. In *Proceedings of the 17th Mediterranean Conference on Control and Automation (MED '09)*, pages 112–114, Thessaloniki, Greece.
- Paschalidis, I. C. and Liu, Y. (2002). Pricing in multiservice loss networks: Static pricing, asymptotic optimality, and demand substitution effects. *IEEE/ACM Transactions on Networking*, 10(3):425–438.
- Paschalidis, I. C. and Tsitsiklis, J. N. (2000). Congestion-dependent pricing of network services. *IEEE/ACM Transactions on Networking*, 8(2):171–184.
- Paschalidis, I. C. and Wu, R. (2008). On robust maximum lifetime routing in wireless sensor networks. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, pages 1684–1689, Cancun, Mexico.

- PJM (2005). *White Paper on Integrating Demand and Response into the PJM Ancillary Service Markets*.
- PJM (2008). *PJM Day-Ahead Scheduling Reserve Market, version 6a*.
- Polastre, J. and Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*.
- Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401.
- Puccinelli, D. and Haenggi, M. (2005). Wireless sensor networks: applications and challenges of ubiquitous sensing. *IEEE Circuits and Systems Magazine*, 5:19–31.
- Rajendran, V., Garcia-Luna-Aceves, J., and Obraczka, K. (2005). Energy-efficient, application-aware medium access for sensor networks. In *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2005)*, Washington, D.C.
- Rajendran, V., Obraczka, K., and Garcia-Luna-Aceves, J. (2003). Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pages 181–192, Los Angeles, CA.
- Reliability and Operations Subcommittee (ROC) (2007). Effects of wind on frequency. Technical report, Energy Reliability Council of Texas (ERCOT).
- Rhee, I., Warrier, A., Aia, M., Min, J., and Sichitiu, M. L. (2008). Z-mac: A hybrid mac for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16:511–524.
- Robathan, P. (1989). *Intelligent Buildings Guide*. Intelligent Buildings Group and IBC Technical Services Ltd., London, UK.
- Rodoplu, V. and Meng, T. H. (1999). Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17:1333–1344.
- Ross, S. M. (1996). *Stochastic Processes*. John Wiley & Sons, 2nd edition.
- SAIC (2006). San Diego smart grid study final report. Technical Report 10-9-06, Science Applications International Corporation Smart Grid Team.
- Santi, P. (2005). The critical transmitting range for connectivity in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4:310–317.

- Savvides, A., Caramanis, M. C., and Paschalidis, I. C. (2011). Cyber-physical systems for next generation intelligent buildings. In *ACM/IEEE 2nd International Conference on Cyber-Physical Systems (ICCPS 2011)*, Chicago, Illinois.
- Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M. (2002). Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80.
- Shargal, M. and Houseman, D. (2009). The big picture of your coming smart grid. Smart Grid News, 5 March.
- Sharples, S., Callaghan, V., and Clarke, G. (1999). A multi-agent architecture for intelligent building sensing and control. *Sensor Review*, 19(2):135–140.
- Shnayder, V., Hempstead, M., rong Chen, B., Allen, G. W., and Welsh, M. (2004). Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pages 188–200.
- Siderius, H. and Dijkstra, A. (2006). Smart metering for households: costs and benefits for the netherlands. http://www.saena.de/media/files/Upload/smart_metering/PDF/Smart_Metering_NL.pdf.
- Smith, J. C., Milligan, M. R., DeMeo, E. A., and Parsons, B. (2007). Utility wind integration and operating impact state of the art. *IEEE Transactions on Power Systems*, 22(3):900–908.
- Tabors, R. D., Parker, G., and Caramanis, M. C. (2010). Development of the smart grid: Missing elements in the policy process. In *43rd Hawaii International Conference on System Sciences (HICSS)*, pages 1–7.
- The Climate Group and GeSI (2009). Smart 2020 : Enabling the low carbon economy in the information age. http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf.
- The House of Representatives (2007). Facilitating the transition to a smart electric grid, May 3, 2007. http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=110_house_hearings&docid=f:40588.pdf.
- Tsitsiklis, J., Bertsekas, D., and Athens, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9).
- Tsitsiklis, J. N. (1984). *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of EECS, MIT, Cambridge, MA.

- van Hoesel, L. F. W. and Havinga, P. J. M. (2004). A lightweight medium access protocol for wireless sensor networks. In *Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS 2004)*, Tokyo, Japan.
- Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Reviews*, 38:49–95.
- Weber, V. (2009). Smart sensor networks: Technologies and applications for green growth. Technical Report DSTI/ICCP/IE(2009)4/FINAL, Organization for Economic Co-operation and Development.
- Werner-allen, G., Johnson, J., Ruiz, M., Lees, J., and Welsh, M. (2005). Monitoring volcanic eruptions with a wireless sensor network. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN '05)*.
- Wieselthier, J. E., Nguyen, G. D., and Ephremides, A. (2000). On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings - IEEE INFOCOM*, pages 585–594.
- Wong, J. K. W., Li, H., and Wang, S. W. (2005). Intelligent building research: a review. *Automation in Construction*, 14:143–159.
- Wu, X. and Cassandras, C. (2005). A maximum time optimal control approach to routing in sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1137–1142.
- Yamaji, M., Ishii, Y., Shimamura, T., and Yamamoto, S. (2008). Wireless sensor network for industrial automation. In *Proceedings of the 5th International Conference on Networked Sensing Systems (INSS 2008)*, page 253, Kanazawa, Japan.
- Yang, X. and Vaidya, N. H. (2004). A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay. In *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004)*, pages 19–26, Los Alamitos, CA.
- Ye, W., Heidemann, J., and Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12:493–506.
- Yeh, L.-W., Wang, Y.-C., and Tseng, Y.-C. (2009). iPower: an energy conservation system for intelligent buildings by wireless sensor networks. *International Journal of Sensor Networks*, 5:1–10.
- Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330.

- Yuan, D., Bauer, J., and Haugland, D. (2008). Minimum-energy broadcast and multicast in wireless networks: An integer programming approach and improved heuristic algorithms. *Ad Hoc Networks*, 6(5):696–717.
- Zheng, R., Hou, J., and Sha, L. (2003). Asynchronous wakeup for ad hoc networks. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 35–45, Annapolis, MD.
- Zimmermann, H. (1980). OSI reference model – the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432.

CURRICULUM VITAE

Binbin Li

15 Saint Mary's Street, Brookline, MA 02446, USA
Phone: +1 (617) 997-6268 E-Mail: libinbin@bu.edu

EDUCATION

Doctor of Philosophy in Systems Engineering

Boston University, Boston, Massachusetts, USA 05/2011

Master of Science in Electrical Engineering

Boston University, Boston, Massachusetts, USA 05/2010

Master of Engineering in Control Science and Engineering

Tsinghua University, Beijing, P.R.China 07/2006

Bachelor of Engineering in Automation

Tsinghua University, Beijing, P.R.China 07/2004

ACADEMIC EXPERIENCES

Research Assistant

Division of Systems Engineering, Boston University 09/2006 - 04/2011

Research Student

Department of Automation, Tsinghua University 10/2003 - 06/2006

PROFESSIONAL EXPERIENCES

Summer Associate in Market Risk Management and Analysis

Goldman Sachs, New York, New York 06/2010 - 08/2010

Optimization Intern in Availability Optimization Engineering

ITA Software/Google, Cambridge, Massachusetts 05/2010 - 06/2010

PUBLICATIONS

Book Chapter

- Ling Wang and Bin-Bin Li, "Quantum-Inspired Genetic Algorithms for Flow Shop Scheduling", in *Quantum Inspired Intelligent Systems*, pages 17-56, N. Nedjah, L. D. S. Coelho and L. D. M. Mourelle, eds., Berlin: Springer, 2008.

Journal Articles

- Ioannis Ch. Paschalidis, Binbin Li, and Michael C. Caramanis, "A Market-Based Mechanism for Providing Demand-Side Regulation Service Reserves", in preparation for *IEEE Transactions on Power Systems*.

- Ioannis Ch. Paschalidis and Binbin Li, “Energy Optimized Topologies for Distributed Averaging in Wireless Sensor Networks”, accepted to *IEEE Transactions on Automatic Control*.
- Bin-Bin Li, Ling Wang, and Bo Liu, “An Effective PSO-Based Hybrid Algorithm for Multi-Objective Permutation Flow Shop Scheduling”, *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, volume 38, number 4, July 2008, pages 818-831.
- Bin-Bin Li and Ling Wang, “A Hybrid Quantum-Inspired Genetic Algorithm for Multiobjective Flow Shop Scheduling”, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, volume 37, number 3, June 2007, pages 576-591.

Conference Papers

- Ioannis Ch. Paschalidis, Binbin Li, and Michael C. Caramanis, “A Market-Based Mechanism for Providing Demand-Side Regulation Service Reserves”, submitted to *the 50th IEEE Conference on Decision and Control*, Orlando, FL, December 2011.
- Ioannis Ch. Paschalidis and Binbin Li, “On Energy Optimized Network Construction for Distributed Averaging in a Dynamic Environment”, accepted to *the 18th IFAC World Congress*, Milano, Italy, August 2011.
- Ioannis Ch. Paschalidis and Binbin Li, “On Energy Optimized Averaging in Wireless Sensor Networks”, *Invited*, in *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009, pages 7137-7144.
- Bin-Bin Li and Ling Wang, “A Hybrid Quantum-Inspired Genetic Algorithm for Multi-objective Scheduling”, *International Conference on Intelligent Computing*, Kunmin, China, August 2006, in *Lecture Notes in Computer Science*, volume 4113, pages 511-522.

HONORS AND AWARDS

Center for Information and Systems Engineering Travel Award

Boston University 09/2011

College of Engineering Dean’s Fellowship

Boston University 09/2006

Outstanding Master Thesis Award

Tsinghua University 07/2006

Cyrus Tang Foundation Fellowship

Tsinghua University 2001 - 2004

Outstanding Freshman Fellowship

Tsinghua University 2000