# REPORT DOCUMENTATION PAGE



Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

| 1. REPORT DATE (DD-MM-YYYY) | 2. REPORT TYPE | 3. DATES COVERED (From - To) |
|---|---|---|
| 26-03-2014 | Ph.D. Dissertation | - |

**4. TITLE AND SUBTITLE**
Optimization-based approach to cross-layer resource management in Wireless networked control systems

**5a. CONTRACT NUMBER**
W911NF-10-1-0005

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
106011

**6. AUTHORS**
Jia Bai

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAMES AND ADDRESSES**
Vanderbilt University
PMB # 407749
2301 Vanderbilt Place
Nashville, TN 37240 -7749

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)**
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

**10. SPONSOR/MONITOR'S ACRONYM(S)**
ARO

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
56919-NS-DPS.39

**12. DISTRIBUTION AVAILIBILITY STATEMENT**
Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

**14. ABSTRACT**
Wireless Networked Control Systems (NCS) are increasingly deployed to monitor and control Cyber-Physical Systems (CPS), as wireless network provides great convenience in terms of fully mobile operation, rapid deployment and flexible installation. To support mission critical operations of CPS, NCS need to achieve and maintain a desirable level of performance. In the wireless network however, resources are constrained by limited bandwidth and power; dynamic user behaviors and resource interference also aggravate network uncertainties and introduce random packet loss and time varying delay. These resource constraints and network dynamics pose

**15. SUBJECT TERMS**
cross-layer resource management, sampling rate adaptation, networked control system

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 15. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Yuan Xue |
| UU | UU | UU | UU | | 19b. TELEPHONE NUMBER 615-322-2926 |

**Report Title**

Optimization-based approach to cross-layer resource management in Wireless networked control systems

**ABSTRACT**

Wireless Networked Control Systems (NCS) are increasingly deployed to monitor and control Cyber-Physical Systems (CPS), as wireless network provides great convenience in terms of fully mobile operation, rapid deployment and flexible installation. To support mission critical operations of CPS, NCS need to achieve and maintain a desirable level of performance. In the wireless network however, resources are constrained by limited bandwidth and power; dynamic user behaviors and resource interference also aggravate network uncertainties and introduce random packet loss and time-varying delay. These resource constraints and network dynamics pose significant challenges and require a fresh treatment to the design of wireless NCS.

In this dissertation, we investigate the problem of resource management in wireless networks to support NCS with stringent Quality of Service (QoS) requirements. The capability of adaptive resource management is crucial for NCS to fully exploit the available resource and achieve optimal performance. The interaction between networking systems and control systems is the key to adaptive resource management. It allows informed operation decisions within individual systems to collaboratively achieve a global management objective. In particular, we present a cross-layer approach to support interactions between wireless networks and networked control systems. The cross-layer design aligns with network layered architecture, thus is feasible for broader adoption in real-world deployment. We consider two information exchange directions in our design. When the control systems deliver their performance requirements to the wireless network, the network adjusts its operation parameters to facilitate the performance optimization of the control systems; When the wireless network passes congestion signals to the control systems, the control systems dynamically adapt their sampling rates to preserve optimal performance. We further explore the cross-layer interactions between the two systems and among layers within the networking protocol stack, which combine the design of control system sampling rate adaptation and the network scheduling. To arbitrate the resource sharing among multiple control systems, we present a new fairness model for wireless network based on the game theoretical framework, and evaluate the impact of resource sharing regions approximated by different neighborhood models.

# OPTIMIZATION-BASED APPROACH TO CROSS-LAYER RESOURCE MANAGEMENT IN WIRELESS NETWORKED CONTROL SYSTEMS

By

Jia Bai

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

May 2013

Nashville, Tennessee

Approved:

Professor Yuan Xue

Professor Xenofon D. Koutsoukos

Professor Gabor Karsai

Professor Sherif Abdelwahed

Professor William H. Robinson

To my darling daughter, Sophie Wang, endless inspiration

## ACKNOWLEDGEMENTS

This work would be impossible without the scientific insight of my advisor, Dr. Yuan Xue. I am greatly appreciative for her great patience and tremendous encouragement. Thanks you for guiding me to be an independent researcher. I would also like to thank Dr. Sherif Abdelwashed and Dr. Xenofon D. Koutsoukos for helping me improve both my research ability and technical expertise. I am grateful to Dr. Gabor Karsai and Dr. William H. Robinson for serving on my dissertation committee. I appreciate their time, advice and support.

I would also like to thank all the members of VANETS group. It has been my pleasure to study and collaborate with all of you. Moreover, I wish to thank Dr. Douglas H. Fisher for the continuous encouragements, and Emeka Eyisi for his diligent research attitude, with which we have great time collaborating and developing innovative research results.

Finally, I want to express my deepest gratitude to my family. I am indebted to my parents for their support and belief in me. Special thanks to my husband, Furui Wang, who makes all my work worthwhile. This work is dedicated to them. In addition, I would like to thank all my friends at Vandy. You have made my experience memorable.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

The integration of physical systems through computing and networking has become a trend, known as Cyber-Physical Systems (CPS). Many real-world CPS such as automotive vehicles and distributed robotics, are monitored and controlled by Networked Control Systems (NCS), where information among sensors, controllers and actuators is exchanged via a communication network. NCS are increasingly deployed over wireless networks, as they provide great convenience in terms of fully mobile operation, rapid deployment and flexible installation. Study of the wireless NCS has formed a multidisciplinary research topic and attracted great interest from both academia and industry [37], finding applications in unmanned robotic vehicles, automated highways and factories, smart homes and appliances, remote telemedicine, etc [32].

## Problem Description and Research Goal

This dissertation investigates the problem of resource management in wireless networks that supports NCS with stringent Quality of Service (QoS) requirements. We address two design objectives, stability and performance, in two design spaces. Stability of the control system is ensured using a passivity-based architecture [56]

at the control system, and the performance of the NCS is optimized via the cross-layer interaction of the network. In particular, we examine a representative class of wireless NCS, where multiple physical plant and digital controller pairs communicating over a multi-hop wireless network. The plants follow the reference trajectories provided by the controllers. We consider the NCS performance in terms of tracking error minimization in the control system.

The capability of adaptive resource management is crucial for NCS to fully exploit the available resource for optimal performance, while avoiding network congestion. Specifically, the following wireless communication characteristics significantly affect the stability and performance of the control systems.

- The wireless communication is unreliable. Signals propagating through the air makes wireless transmission vulnerable to noise and interference. If the transmitter, receiver, or surrounding objects are moving, the shadowing effect, multipath fading, and interference from other devices will make the channel conditions fluctuate unpredictably, causing random packet loss and time-varying delay.

- The wireless resource is constrained. The fast growing of wireless usage leads to increasing demand for the shared resources, e.g., limited bandwidth and power. Moreover, the usage of wireless spectrum is strictly administered, placing restrictions on the available resource. On the other hand, different control systems may require different amount of resource depending on their operation needs.

Building a wireless networking system that supports mission-critical NCS requires fresh design approaches to address these difficulties. This Ph.D. research work studies dynamic allocation of the network resource for optimizing the NCS performance. The research goal is to design wireless networks that deliver the best support for the optimal NCS operation.

## Research Approach and Dissertation Contributions

Three major directions have been explored in literature to address the challenges in building wireless NCS. One direction, independent of the network design, studies the strategies of control systems. The goal is to maintain stability and provide certain level of performance assurance of control systems despite of underlying network uncertainties [37, 101]. Another direction focuses on reliable and timely network packet delivery, independent of the control systems. Yet without cooperation between the network and the control systems, it is hard for the above two approaches to achieve the overall optimal system performance [79, 12]. The third direction is to integrate and perform a joint design of the wireless networked control system. While theoretically this approach can achieve optimal system performance, it requires tight coupling of communication and control, and inevitably involves too many interactions across different components, which is unrealistic for deployment [11, 71, 87].

In this Ph.D. research work, we perform cross-layer resource management in order to optimize the performance of NCS. While the NCS can naturally be decomposed into two essential components: wireless network and networked controller, more vitally is the interaction between the components. We optimize the performance of the wireless networked control systems via decomposition and interaction. That is, the network layer and the control layer perform their individual tasks and coordinate with each other via cross-layer signals.

The decomposition and interaction facilitate efficient layer abstraction and encapsulation by defining a clean interface. Each component will retain its implementation details, with only the vital information exposed to the other component, e.g., system requirements and performance status. This brings the following advantages:

- Flexible maintenance. Internal states of one component can update locally without changing or even notifying the other component. Components work more independently as long as the interaction is intact.

- Clear Focus. With the shared information specified explicitly, NCS designers only need to focus on these parameters without touching the major part of the components. Also the interactions are relatively independent and convenient to modify.

Two key questions about the decomposition and interaction remain unanswered though:

1. How to define a clean interface so that only essential parameters are shared across the components.

2. How to process the received interaction signals so that the components' stability is maintained and their performance is optimized.

Our main research task is to define the interface between the control system and the network layer, with the complexity of the interactions regulated. In this work, we first investigate one-way interaction between the control system and the wireless network. From the control system to the network, the performance information of the control system is delivered, and the network adjusts its operation parameters to best support the performance optimization of the control system. From the wireless network to the control system, the congestion signal of the wireless network is passed, and the control system dynamically adapts its sampling time to preserve the optimal performance. Next, we explore the cross-layer interactions between the two systems and among the layers within the networking protocol stack, which integrate the control system sampling rate adaptation with network scheduling. The work is described in more details as follows:

### Dynamic Tuning of the Wireless Networks

We first study how the network loss and delay may affect the NCS performance in chapter III. We observe that both factors have negative impact on the NCS performance, and they can be balanced through the Medium Access Control (MAC) retransmission limit of the network. Thus we adjust the operation

parameters of the wireless network based on the NCS tracking error passed from the control system. In particular, we present a MAC controller that dynamically tunes the retransmission limit to track the optimal trade-off between packet loss and transmission delay and thus optimizes the control system performance. Simulation results show that our approach significantly improves the performance of the NCS.

Cross-layer Sampling Rate Adaptation of the Control Systems

In Chapter V, we present optimization formulations for minimizing the tracking errors introduced due to (1) discretization and (2) packet delay and loss.

In the first step, we consider the problem of dynamic rate adaptation for NCS so that they can fully utilize the scarce wireless resource to minimize the discretization induced tracking errors. The ability of using different sampling rates in control systems provides the flexibility for adapting their resource needs based on the dynamic resource availability in the wireless networking environment. From the control perspective, the more a controller knows about the system, the better the controller will perform [71]. This can be done by exchanging messages more frequently. However this increases the communication burden on the network and may lead to congestion. The congestion results in longer delays and more packet losses, which will degrade the control performance. We employ a price-based approach, where prices are generated to reflect the congestion level of the contention

regions in the wireless network and used as the basis for rate adaptation. A distributed solution for sampling rate adaptation is then developed using utility-based optimal resource allocation, where a utility function quantifies the relationship between the sampling rate and the capacity of disturbance rejection of the control system (i.e., minimizing the discretization-induced tracking errors).

In the second step, we minimize both tracking errors by joint design of control system sampling rate adaptation and network scheduling. The optimization problem maximizes the utility function in the first step. The optimization constraints are from the wireless network capacity and the end-to-end delay requirements. The solution leads to a joint design of sampling rate adaptation and network scheduling, which can be naturally deployed over existing layered networking systems. Based on a passivity-based control framework, we show that the proposed cross-layer design can achieve both stability and performance optimality.

We conduct simulation studies in Networked Control System Wind Tunnel (NCSWT) [99], an integrated simulation environment consisting of Matlab/Simulink and ns-2. The results demonstrate that our algorithm is able to provide agile and stable sampling rate adaptation and achieve optimal wireless NCS performance.

To arbitrate the resource sharing among multiple control systems, we further define new end-to-end fairness models based on a game theoretical framework for wireless networks with non-convex capacity region, and evaluate the impact of the resource sharing regions approximated by different neighborhood information on the fairness performance.

End-to-End Fairness of Non-convex Wireless Networks

In Chapter VI, we present a new fairness model for IEEE 802.11 wireless network where the capacity region is non-convex. Fair resource allocation for end-to-end flows is an important yet challenging problem in multi-hop wireless networks. Recent research on fair resource allocation is mainly based on the assumption of convex resource regions, which has been theoretically proven untrue. To characterize the desired fairness property, we adopt an axiomatic approach based on the game theoretic framework. The new fairness model grounds on the Nash Extension Solution (NES), which is shown to be consistent with the concept of proportional fairness under the convex cases, while approximating it under the non-convex cases. We further present an efficiency enhanced version of Nash extension solution to push the NES to the strong Pareto frontier. A time-decomposed price-based rate allocation algorithm is then presented and its stability is proven.

End-to-End Fairness of Neighborhood-Information Impacted Wireless Network

In Chapter VII, we evaluate the different heuristic rate allocation solutions that simplify the scope of resource sharing regions using different neighborhood models. A baseline fairness model (*i.e.*, proportional fairness) is established using the price-based resource allocation framework. In this framework, price represents the cost of the resource usage incurred by unit flow. The rate of a flow is directly linked to its price, which is the aggregated price of the links it traverses. The link price is the sum of the prices of all the resource sharing regions which it belongs

to. Obviously, when the resource sharing regions are approximated by different neighborhood models, link prices will manifest as different values. Six different neighborhood models are constructed and their deviations from the baseline proportional fairness model are assessed.

In addition, we present in Chapter II an overview of existing related literature, which includes control system design, wireless network design, and joint design for the NCS, emphasizing the problem of resource allocation. We conclude the dissertation in Chapter VIII.

# CHAPTER II

# BACKGROUND AND RELATED WORKS

A brief history on the use of communication networks as parts of a control system has been summarized in [139, 37]. We will give a literature review emphasizing the recent developments of resource allocation in wireless networks supporting the design of NCS.

## NCS Overview

### Control with Network Effects

Introducing networks into control loops provides great convenience and flexibility for control operations, while it also complicates the analysis and design of the NCS. The behavior of NCS largely depends on the network effects of the underlying networks [140, 71]. These effects including time-varying delay, random packet dropouts, undetermined channel capacity, and etc., all can lead to network uncertainties. Specifically, the following network effects are mainly addressed in literature.

- *End-to-end delay*. The end-to-end delay can be captured by the plant-to-controller delay and controller-to-plant delay. An early introductory tutorial [17] addresses that analysis and design of control systems when the communication delays are varying in a random fashion are complex. To overcome these problems, the delays can be modeled as either constant, jittered, time varying with known probability distribution or independently random [85, 86, 67]. [76] reviews the effects of constant and varying time delays on the performance of closed-loop control systems for different control models.

- *Packet dropouts*. Packets can get lost during transmission due to congestion, collision or unreliable hardware. The loss probability can be defined as the reliability of the network paths. Usually network protocols have retransmission scheme, but will drop the packets after retrying for a limited number of times. It has though been addressed that discarding old, untransmitted message and transmiting the new one if available is advantageous for the control system to always get fresh data [140].

- *Network throughput*. The throughput defines how many bits per time unit can be transferred along a given path. When the paths of different control systems share network channels, each system can only get certain portion of the constrained bandwidth.

Some other network effects include delay or loss jitter, quantization error, and packet ordering. [106, 139] provide literature reviews categorized according to the different network effects.

### Category of Methodologies

The research of NCS has been widely explored in existing literature [140, 106, 43, 101, 41, 37] along the following three major directions.

- *Control over network*. Develop of the control strategies that explicitly takes the network uncertainties into account over given networks [27].
- *Control of network*. Study of the different layers in the networks to minimize the effects of the network uncertainties on designed control systems, e. g., congestion regulation, route selection, MAC scheduling [5, 77], and etc.
- *Control with network*. Joint design combining the efforts from both the control systems and the networks.

Our research focuses on the last two categories. That is, how wireless networks can work more efficiently with the control systems via the design of wireless networking systems and the design of the joint operation. Several solutions to packet loss and end-to-end delay in the wireless NCS have adopted similar approaches to those in wired networks. However, the design is usually more complicated with wireless networks. Note that the framework in [89] presents a new

paradigm different from the classical NCS model, which obtains the controller's behavior from the aggregate computation of different nodes in the network.

**Control over Network**

[41, 101, 37] provide overviews mainly on the direction of control over network. It has been emphasized that the choice of network depends upon the desired application. The applications of the control systems can be classified based on their real-time requirements.

- Hard deadline, where all given deadlines must be met under any circumstance, otherwise the system will fail. Hard guarantees are impractical on platforms such as wireless networks since the networks' load and resource capacities are very difficult to predict and manage.

- Soft deadline, where missing a deadline will degrade the system performance. Guaranteed delivery and best effort are two subsets of this requirement. The former does not have any particular constraints on the actual delay of a packet, while the latter attempts to deliver packets as fast and errorless as possible, without successful delivery guarantees. Many modern applications require some form of performance assurances.

To control over networks, different state estimation methods are used to track the network effects. [41] provides a survey on problem of state estimation over

imperfect communication channels. [48] uses a two-state Markov model to estimate the frame loss over time of a moving window. It detects the network disturbances and predicts the system operation in terms of stability and disturbance rejection. [73] models the wireless delay dynamics with a Kalman filter, which produces an optimal delay estimate with a stochastic model.

Many control-based approaches have been applied to NCS to provide stability guarantee and performance enhancement, such as state augmentation, queuing and probability theory, nonlinear control and perturbation theory. Following are two commonly used methods.

### Compensation Method

Compensation methods are used for the end-to-end delay and packet loss. Different mathematical, heuristic, and statistical-based approaches are presented for compensation in NCS [79]. Advanced techniques include queuing/buffering [74], optimal stochastic control [86], robust control [137], linear matrix inequality (LMI) [61], gain scheduler middleware (GSM) [116], predictive gain scheduler [52], predictive control [68], and etc. Detailed descriptions can be found in [37]. Recent work [59] focuses on compensating for exponentially bounded long dropout bursts in the network by reconfiguring the controller or the network to guarantee stability.

Reduced communication

The idea of reduced communication control is to under-utilize the network resource. Results in [19, 43] demonstrate that when it is blocked either from sudden bursts of traffic or from long periods of poor channel conditions, under-utilizing the bandwidth of the channel during periods can ensure sufficient bandwidth, prevent queue buildup, and optimize the control and network performances simultaneously.

Significant research efforts have been devoted to the problem of determining the minimum bit rate that is needed to stabilize a linear system through feedback over a channel with finite capacity [123, 28, 39, 81, 113, 13]. Kim et. al. [51] defines a Maximum Allowable Delay Bound (MADB) guaranteeing stability in terms of LMI. [75] proposes a less conservative MADB. Other literatures derive bounds on transmission period to guarantee NCS stability. Branicky [141] obtains the bound on the time-varying transmission period. Walsh et. al. [119] introduces the notion of Maximum Allowable Transfer Interval (MATI). Nešić et. al. [82] presents less conservative MATI bounds that can be applied to a general class of NCS with static and dynamic protocols. Several other subsequent improvements of MATI bounds are reported in [15] and [112]. [40] and [111] further relax the MATI bounds by considering stochastic stability properties instead of deterministic ones. [38] provides tradeoffs between MATI and the maximally allowable delay. Deadbands [88] are also used to reduce traffic in NCS. A node with a deadband compares the previous value it sent to the network to the most recent one. If

their absolute difference is within the deadband, no update is sent to the network. The optimal size of the deadband is determined by the tradeoff between traffic and control performance. [78] employs model-based NCS that explicitly uses the knowledge about the plant dynamics and only places critical information about the plant on the network to reduce traffic load.

**Control of Network**

Kumar [57] provides the first critical analysis on the use of wireless control. It explores a wide spectrum of issues of the wireless network effects on control performance, from the physical layer up to the networking layer. In a network system, the Internet protocol stack consists of five layers from top to bottom: the application layer, the transport layer, the network layer, the link layer, and the physical layer [58]. For the NCS, control systems reside in the application layer. The control of network is thus mainly about design of the other four layers.

- Transport layer. The transport layer provides end-to-end communication services for the application layer. There are two transport protocols, TCP and UDP. TCP provides connection-oriented service with guaranteed delivery and flow control, while UDP provides connectionless service with no reliability, no flow control, and no congestion control. The NCS prefers UDP over TCP, considering its distinct advantages of low overhead, small latency and flexibility over TCP [106].

- Network layer. The network layer is responsible for routing packets from one network host to another.

- Link layer. The link layer is primarily about how channel is shared among multiple transmitters, with the Medium Access Control (MAC) protocol defining the access scheduling and collision arbitration policies. The MAC can be either of random access or with scheduling [107]. In the standard protocols, random accessed network often uses Carrier Sense Multiple Access (CSMA), whereas scheduling commonly employs Token Passing (TP) and Time Division Multiple Access (TDMA). CSMA is used in DeviceNet and Ethernet, with possible packet collision and unbounded worst-case transmission time. TP protocol appears in token bus (IEEE Standard 802.4), token ring (IEEE Standard 802.5) and Fiber Distributed Data Interface (FDDI) MAC architectures; TDMA is used in FireWire. TP and TDMA eliminate contention and with bounded and constant packet transmission delays [140].

- Physical layer defines point-to-point communication transmitting individual bits from one node to the next. Its services include modulation and coding.

The standard layer design usually cannot meet the real-time requirements of the control applications. Thus, some NCS research work assumes simplified network without layering, some focuses on the design of one layer, and other work highlights the cross-layer design [69, 70, 71]. The cross-layer design has been applied to network applications, such as video over wireless [135], sensor networks

with energy constraints [24], and in wireless ad hoc networks [104]. In the NCS, it optimizes the end-to-end control performance by designing the network protocol stack layer by layer.

To analyze the effects of networks on the NCS performance, several work studies the trade-offs between network communication parameters. [140] illustrates the relationship between the sampling period and allowable time delay by a stability region plot. [69] examines the trade-offs between the data rate, error correction coding and the maximum number of retransmissions of the link layer design in Linear Quadratic (LQ) gaussian control. It then determines the optimal selection of these parameters to achieve the best control performance. Later work [70] examines the effects of different wireless network MAC layer protocols on the NCS performance, including centralized TDMA, polling and decentralized random access without ACK, random access with ACK, and CSMA/CA of 802.11. The 802.11b MAC layer is studied in details in [19, 18]. They investigate the properties of the wireless hard real-time NCS under heavy contention from control-only and mixed-traffic data, as well as their performance under channel error and different bandwidth. [138] investigates the trade-off between the sampling rate and the data accuracy, with given fixed average throughput. It demonstrates that the controller prefers more frequent communication with the plant even if the information is crude to obtaining a piece of more precise information with a long delay.

**Control with Network**

The joint optimization considers attributes of the control systems and features of the wireless networks simultaneously. The join design is of two-fold: the controller needs to be robust and adaptive to communication faults such as random delays and packet losses, while the network should be designed with the goal of optimizing control performance [71]. One important concept of joint design [11] is that the co-design approach begins by expressing both the control and network constraints. The separation of concerns allows the control community to focus on the control design without worrying about the details of how the network is implemented. At the same time, it allows the network community to focus on development of scheduling theory and computational models to provide guaranteed services without thorough understanding of how the control system needs to be constructed [96].

[26, 25, 80] design controllers and protocol simultaneously, producing larger bound on MATI. [4] uses play-back buffers to remove the uncertainty in the delay. The increased loop delay of the buffers is then compensated by model predictive controllers. [84] adjusts the data retransmissions attempts based on the QoS factors monitored by counting the lost packets. It also periodically tunes the controller's minimum gain to decrease the control system's sensitivity. The gains are computed via the combination of LMI-theory and a point-and-shoot algorithm in an ad hoc procedure. [44] investigates the data transmission rate, power consumption, and congestion levels in the wireless network. Co-design is pursued by

formulating communication and control systems as a state-space model that help meet the desired signal-to-interference ratio and stability.

**Resource Allocation**

Among all the works of NCS, resource allocation has drawn great attention of both control and network engineers. Traditionally, NCS operates at a fixed sampling rate based on its average or worst-case resource requirements. However, with the rapid growth of network traffic, control systems need to compete for finite network resources, such as network bandwidth. Without proper coordination, congestion is a common consequence. [121] has suggested that resource allocation can be an effective method to improve performance of control loops under such conditions. In addition, [138] shows that each application may have different requirements for timing and sampling periods according to the control strategy used and the noise in the data. [87] also addresses that the amount of data to be transferred through the network may vary for different control systems, which requires the need to allocate the network resources according to the necessity of a control loop.

The problem of resource allocation is usually formulated into an optimization problem with an objective function subject to constraints. The objective function is the sum of the utility functions $U_f(\cdot)$, which relates control performance with

system characterization like sampling rate, or control characterization like controlled system error. For example, $U_f(r)$ expresses the degree to which a particular system $f$ can benefit from sampling rate $r$ [2]. In this case, the utility function is monotonically increasing, reflecting that higher sampling rates lead to better control performance. It is also strictly concave, reflecting the law of diminishing returns as the rate increases. In the NCS literature, quadratic and exponential utility functions are commonly used. The constraints incorporate the resource limits and other NCS requirements. [102] is the initial work that uses an utility function to capture the relationship between the sampling rate and control performance. It leads to an offline solution that deals with fixed computing resources.

An allocation scheme can be either static or dynamic.

- Static. The pattern of allocation is determined in advance and fixed during system operation. It ensures average control performance at the expenses of permanently occupying certain amount of resource. But it may not be efficient as the pre-assigned resources can be under or over utilized.

- Dynamic. The access to the shared resources is determined at runtime, based on the dynamic changes of the information. It is flexible and adaptable to dynamic changes, so that under-utilized resource can be made available to other applications to provide new functionality or otherwise improve

performance; over-utilize of the resource can be regulated. However, the dynamic allocation may take more computation time. [116] argues that computation time of a properly designed bandwidth management system might not be substantial when compared to other delay in NCS.

[98, 11, 127] study the problem of static resource allocation. [98] considers the case of several control systems sharing one communication channel, and only one controller can use it at each communication instant. Its objective function is formulated to minimize the difference between the sampled-data closed-loop performance and that of the continuous control. Exhaustive search is then employed to find the optimal communication sequence. [11] associates each NCS with a performance measure as a function of transmission period. Its optimization problem is to minimize/maximize the performance function with RM schedulability constraints and NCS stability constraints. [127] presents a linear system transmitting several signals over communication channels with bit rate limitations during each sampling period. With the fixed linear system, it minimizes the quantization error of a white-noise model with the constraint of resource limitations.

Recent work has identified the need for dynamic resource management of NCS. [2] addresses that an economical resource allocation of NCS needs to exhibit the following features:

- It ensures stability of all control systems, if feasible.
- It attains the maximum aggregate performance of all control systems.

- It is efficient and controls congestion to minimize delays and losses, and is fair by fulfilling performance objectives of different control loops.
- It provides a fully distributed, an asynchronous, and a scalable solution. Each node executes independently using local information with no central managing entity. It scales up as the number of controlled systems and/or the size of the network increase.
- It is dynamic and flexible. It dynamically reallocates network resources as different control systems acquire and release the network.

The dynamic resource allocation can be used to accommodate network changes. For instance, NCS on the move, such as the automated highway systems, need to take the time-varying channel into account. The amount of resource required could also depend on different factors according to the nature of the control system and the operating environment local to each agent in the system. Dynamic resource allocation can be achieved by sampling rate adaptation in the control system, or scheduling in the network. Other approaches include market auction [117], where each control loop in the NCS competes for the bandwidth using market based technology with reward systems to control the auction price.

### Sampling rate adaptation

Sampling rate adaptation implicitly regulates traffic injected into the network. Several papers [122, 69, 70] have shown that faster sampling extends the range

of stability with respect to packet loss, but higher sampling rates in turn produce larger rates of data loss. The advantages [21] of sampling rate adaptation include no requirement for plant models as well as being controller- and protocol-independent, making it convenient to integrate with current commercial-off-the-shelf networking technology.

Motivated by the goal of maintaining low levels of data loss while keeping to the highest possible rate of sampling and control [43], positive results have been obtained by dynamically adjusting sampling rates to reflect varying network conditions. [33] proposes an heuristic algorithm to adapt bandwidth allocation of control systems over a CAN bus based on two factors, network load and stability threshold. [48] varies the sampling period of each controller based on the state estimation of network conditions, system stability/performance requirements, and computation/bandwidth limits of the hardware. In [2], control systems vary their sampling periods based on the congestion level of Wide Area Networks (WANs) fed back from the network. It allocates bandwidth to avoid network congestion of WANs and preserve high performance level of NCSs. A convex optimization is lower bounded by the minimum rates that guarantee system stability, and upper bounded by the total network capacity. It is then solved by dual Lagrange multipliers in a fully distributed manner. [21] adapts the sampling interval based on the measurement of round-trip delay and assures stability in the mean square sense using discrete-time Markov Jump Linear System (MJLS) theory. The MJLS is based on an ‘a priori’, static sampling policy, with network dynamics described

as linear time-invariant systems switching between a finite combination of sampling interval and delay. Bao [7] proposes a rate allocation technique to minimize the distortion introduced by quantization over a noisy channel. An optimization problem is constructed with the objective of minimizing the LQ cost by means of Mean Squared Error (MSE). It is constrained by the total rate, and solved using Lagrange duality. The MSE is derived for the instantaneous distortion, and the system state is not affected by the rate allocation. [100] presents a methodology for determining optimal sampling rates for feedback loops focusing on WirelessHart networks.

Some sampling rate adaptation schemes are also based on the control system dynamics. [118] presents a resource management approach in NCS that allows control loops to locally consume network's available bandwidth according to the dynamics of the controlled process while attempting to optimize overall control performance. [62] always allocates larger sampling rates constrained by the available network bandwidth to signals with higher priority. A sampling rate goodness measurement function and a system cost function are defined to assign the controlled plants experiencing faulty conditions with higher priorities. Thus the signal sampling and reconstruction error due to limited network bandwidth will be minimized. [87] calculates the sampling time according to the internal status of the control systems as well as their environmental condition. It first ensures the NCS stability with minimum bandwidth requirement and then improves the NCS performance by allocating the remaining bandwidth considering the normalized value of curvature and speed of each vehicle.

## Network Scheduling

The availability of time on a shared network for communication is an important factor limiting the performance of a NCS. [63] performs an experimental study of communication network characteristics. It shows that the transmission time of a message in the most used networks can be neglected and the delays in NCS are mainly due to the contention between messages sent by different nodes. The most efficient way of delay reduction is thus through the design of appropriate message scheduling strategies.

In the real-time systems, scheduling techniques, e. g., Rate Monotonic (RM) and deadline monotonic, are used in processors. Network scheduling is initially derived from these techniques, and later takes different forms to be compatible with network properties. One main difference between processor scheduling and network scheduling lies in that task executions can be preemptive while network transmissions are typically not. The network scheduling algorithm is a set of rules that, at any time, determine the order in which messages are transmitted [11]. When a set of control systems share network resources, with no coordination, concurrent transmissions can occur and backoff schemes are important to avoid collisions or bandwidth violations. However, backoff will result in transmission delays or even packet drops if the waiting queues are full. Good scheduling algorithms should minimize such system performance loss. Term *schedulable* is often used for control systems with hard deadlines. A scheduling algorithm is said to be schedulable if a set of NCS transmissions can all be completed before their

deadlines. The scheduling algorithm can be static like token rings and polling, or dynamic.

The priority function is a convenient way to formally define the scheduling policies for controllers so existing standard control techniques can be implemented. It distinguishes different types of data with classified priorities. Network resources are then allocated according to the highest priority first paradigm. Different priority assignment will result in distinct resource allocations [125]. Accordingly, the assignment of nodes' priorities can be used to distribute network resources among control loops. Therefore, the problem of network scheduling becomes how to dynamically assign priorities to the control systems, with the goal to achieve optimal resource allocation and maximize the overall control performance. [34] highlights a class of online scheduling policies targeted at scheduling frames in the MAC layer on top of the CAN priority bus.

The scheduling can be based on the information related to the states of the controlled systems or to the message deadlines. [119, 134, 125] determine the controllers' priorities based on the control errors. Walsh et al. [119] study two scheduling methods: token-ring-type scheduling and dynamical Try-Once-Discard (TOD) for MIMO continuous NCS. In TOD, the plant with the greatest weighted error from its last report value to the controller will transmit. The error is defined as the difference between the signal out of the network and that into the network. It is set to 0 at the transmission time if a plant transmits over the network. [134] presents a Large Error First (LEF) scheduling paradigm, which adjusts node priorities based on the state errors of the controlled plants. The control application

demand is expressed using a dynamic cost function. [125] presents a maximum urgency first scheduler, in which the urgency of each control loop is computed based on the setpoints and current system outputs in a central master node. Further, signals with low priorities may be discarded to give network resources to more urgent control loops. The scheduling of [124, 91] are based on message deadlines. [124] proposes a distributed dynamic message scheduling method based on Deadline of Message (DM) to satisfy timeliness of messages and improve the system's flexibility on CAN. Identifier of message is dynamically changed: the longer the elapsed time is since the preceding message is transmitted or the shorter DM is, the higher the priority level of message will be. [91] develops a new wireless MAC protocol with static-priority scheduling in a wireless industrial network. The goal is to schedule all messages such that all transmissions are accomplished before their relative deadlines without any collision of data bits. [133, 105] develop new protocols that assign priority to controller or measurement type data, and then propose algorithms for dynamical scheduling: constant penalty, estimated error order and lag first-order schemes in [133], ID initialization in [105].

### Joint Design

The co-design of sampling rate adaptation and network scheduling is another topic in literature. [11] computes the smallest transmission period with RM schedulability and NCS stability. If a set of NCSs cannot be scheduled with the given time constraint, some packets of the faster sampling NCSs will be dropped

to guarantee stability. [90] allows for heuristic sampling period adjustment to allocate bandwidth to different types of messages and exchanges the transmission orders of data. [126] develops an integrated feedback scheduler. It encompasses a cascaded feedback scheduling module for sampling period adjustment and a direct feedback scheduling module for priority modification. The sampling period is adjusted according to an optimization problem. It minimizes the aggregated absolute instantaneous control error, and constrained by the total utilization obtained via the control error of deadline miss ratio. The controller priorities will be switched according to individual's absolute instantaneous control error.

[127, 71, 65, 20] study the joint optimization of control systems' sampling rate adaptation with other network control schemes. Xiao [127] jointly solves the controller synthesis and rate allocation over communication channels with bit rate limitations. It iteratively fixes one set of variables, and then uses dual decomposition method to optimize over the rest of variables. It applies the conventional uniform quantization method with a white-noise model and focuses on the trade-off between the transmission power and the bandwidth allocated to channels. Liu [71] highlights the network cross-layer design problem, and optimizes the control performance by investigating the interaction of the physical layer design, the MAC protocol selection, and the controller sampling period. A LQ cost function is used as the optimization object. And a suboptimal iterative method jointly solves the controller synthesis and communication rate allocation. Liberatore [65] proposes an algorithm to integrate a play-back buffer with sampling time adaptation and contingency control. Colandairaj [20] uses data rate scaling to improve the frame

error ratio of the IEEE 802.11b wireless channel under poor channel conditions. The resulted variation in channel bandwidth is compensated by sampling interval adaptation [21].

[76, 42, 31, 142, 120] work on the co-design of network scheduling policies and other control techniques. Assuming bounded network delay, [76] shows that the co-design of adaptive controller and LEF scheduling allows for the optimization of the overall Quality of Control (QoC). The adaptive controller adapts the control decisions online according to the dynamics of both the application and executing platform through message scheduling. The scheduling is considered as a bandwidth allocation problem with the optimization goal of maximizing the QoC. It takes into account system states, and subject to the communication bandwidth constraint. [42] studies scheduling and control co-design based on robust $H_\infty$ fault-tolerant control. Parametric expression of controller is given based on feasible solution of LMI. Two classes of scheduling: task scheduling and network scheduling are considered. The network scheduling employs EDF, and for the task scheduling, the plant integral absolute error index is adopted to assign task priority. [31] uses the model predictive control to find the optimal control sequence and optimal network allocation sequence that minimize a quadratic cost function, based on the prediction of the future system evolution over a horizon of $N$ sampling periods. It first computes the off-line scheduling with branch and bound algorithm over limited bandwidth deterministic networks, and then addresses a heuristic pointer placement online scheduling as a compromise. [142]

proposes a predictive control co-designed with preemptive RM and dynamic feedback scheduling. [120] proposes a predictive control and static RM scheduling co-design. The predictive controller generates the predictive control sequences using sensor data and previous control information, and a communication constraint compensator is designed at the actuator side to actively dynamically compensate for the communication constraint in the forward path.

Some other co-design of resource allocation includes [127]. [127] optimizes the stationary performance of a linear system by jointly allocating the number of bits transmitted during each sampling period in the communication network and tuning parameters of the linear systems. It sequentially fixes one set of variables, and then uses dual decomposition method to optimize over the others.

# CHAPTER III

# DYNAMIC TUNING RETRANSMISSION LIMIT OF IEEE 802.11 MAC PROTOCOL FOR NETWORKED CONTROL SYSTEMS

## Introduction

The integration of physical systems through computing and networking has become a trend now known as Cyber-Physical Systems (CPS). Many CPS such as automotive vehicles and distributed robotics, are monitored and controlled by Networked Control Systems (NCS) which exchange information among sensors, controllers and actuators over a communication network. Wireless network is gaining increasing popularity with NCS, as it provides great convenience in terms of deployment and mobility support. Yet building NCS over wireless networks is an extremely challenging task. The wireless communication characteristics, such as random packet loss, time-varying delay and limited channel capacity, significantly affect the stability and the performance of the control systems.

Two major approaches have been investigated in the existing literature to address the challenges in building wireless networked control systems. One approach, independently of the network protocol design, investigates the design of the control layer (*e.g.,* controller). The goal is to achieve a desired control system performance despite of the underlying network difficulties. For example, several works [12, 56] have been done to ensure the stability of the NCS in presence of packet losses and time-varying delay. Other works have focused on improving

the performance of the NCS [114, 76, 115, 97, 85]. Yet without any support from the network, it is quite hard for the control layer to achieve stability and optimal performance simultaneously. For the works that ensure the NCS stability [12, 56], the issue of performance degradation is not addressed. For the works that improve the NCS performance [114, 76, 115, 97, 85], it is not clear whether they can achieve stability in wireless environment. The other approach is to perform a co-design of the control layer and the communication layer (*e.g.,* network protocols) [71, 83, 19, 128, 84]. While this approach can achieve both stability and optimal performance of NCS, its design inevitably involves too many interactions between the control and the communication layers, which prevents efficient layer abstraction and encapsulation and also hinders broader adoption.

To address the above open issues, we present a novel approach to the design of wireless networked control system. This approach decomposes the design concerns into two factors and addresses them separately in two design spaces – stability of the system is ensured through controller design at the control layer; performance of the system is optimized through adjusting network protocol parameters at the communication layer. At the control layer, we leverage our previous work on using a passivity-based architecture in designing NCS that is robust to network delay and packet loss [56, 54]. In this chapter, we focus on studying the impact of MAC layer packet retransmission on the performance of the passive controller and investigating the optimal design of retransmission strategies.

In IEEE 802.11 MAC protocol, a frame will get retransmitted up to a certain limit, if it is lost due to random channel errors. It is obvious that allowing a higher

retransmission limit increases the chance of successful packet transmissions at the cost of longer packet transmission delays; while a lower limit will result in a larger packet loss probability with smaller delays for delivered packets. Since both packet loss and transmission delay have negative impact on the performance of the controller, the key questions we would like to answer are, what is the packet retransmission limit that optimizes the controller performance and how to achieve it. We consider a passive controller which produces a trajectory for the plant (a robotic arm in our system) to track and define the performance of this NCS as its absolute tracking error. We observe that the relationship between the NCS performance and the MAC retransmission limit can be characterized by convex functions depending on the channel error probability. Using this convex property, we design a heuristic control algorithm that dynamically adjusts the MAC retransmission limit to track the optimal retransmission limit under time-varying channel errors. Simulation results show that the MAC controller can converge quickly to a proper retransmission limit which optimizes the performance of the control system.

The main contributions of this chapter are as follows. First, we present a novel approach to NCS design. By ensuring the stability of NCS at the controller layer and optimizing its performance at the communication layer, this approach is able to achieve both design goals of NCS while maintaining a clean cross-layer interaction. Second, we present a control algorithm that dynamically adjusts the MAC retransmission limit to track the best trade-off between packet loss and delay that optimizes NCS performance.

The rest of the chapter is organized as follows. The system models are described in Section III. Section III shows the observation of the effect of loss and delay to the performance of the NCS. The MAC controller is designed in Section III. The experiment evaluation results are presented in Section III. summarizes this chapter.

## System Models

We consider a networked control system consisting of a controller and a plant communicating through a UDP connection over an IEEE 802.11-based wireless network. The controller controls the plant, which is a robotic arm, to follow.

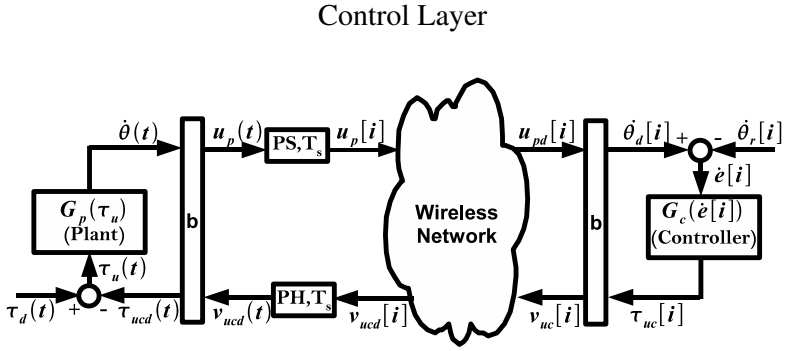


Figure III.1: Passivity Based Control Architecture Over Wireless Networks

Fig. V.3 shows the structure of the system in the control layer. The figure depicts a passive control architecture for the digital control of a continuous plant, over a wireless Local Area Network (LAN). In [56], the architecture is shown to

be *passive* by design, which means it ensures stability of the NCS in the presence of network uncertainties such as time varying delays and packet losses. A control system is considered stable if its output will stay bounded for any bounded input, and the system performance is considered as how fast and accurate the plant can track the control signal within the bound. Using this passive architecture allows us to focus solely on system performance. We provide a brief description of this architecture, and refer the reader to [56] for a detailed description and proofs pertaining to the passive control architecture.

In Fig. V.3, $G_p(\tau_u)$ is the plant system to be controlled. The plant is a continuous linear time-invariant system and the composite dynamics of the plant is by design, strictly output passive. The *plant system* takes the torque control command $\tau_u(t)$ as input, and outputs velocity $\dot{\Theta}(t)$. $G_c(\dot{e}[i])$ denotes the digital controller which controls the plant to behave in a desired manner. The *digital controller* is a discrete-time linear time-invariant system and is also designed to be strictly output passive. The controller takes as input, the error velocity $\dot{e}[i]$ between the reference and the plant output, and outputs torque command $\tau_{uc}[i]$.

The block $b$ transforms the power variables (*i.e.*, the direct input and output of plant and controller) into wave variables for communication over a wireless network. These wave variables preserve the passivity of the system. On the plant side, the wave variable $v_{ucd}(t)$ and the velocity measurement $\dot{\theta}(t)$ are considered inputs to the wave transform block and the wave variable $u_p(t)$ and delayed torque command $\tau_{ucd}(t)$ are considered outputs of the wave transform block. On the controller side, the wave variable $u_{pd}[i]$ and the control torque $\tau_{uc}[i]$ are considered

inputs to the wave transform block and the wave variable $v_{uc}[i]$ and delayed velocity measurement, $\dot{\theta}_d[i]$ are considered outputs of the wave transform block.

The $[PS, T_s]$ and $[PH, T_s]$ blocks represent the passive sampler and passive hold respectively. The passive sampler, at a sampling time $T_s$, interconnects the plant to the digital controller. It converts the continuous wave variable $u_p(t)$ to an appropriately scaled discrete wave variable $u_p[i]$. The passive hold, on the other hand, converts the discrete time wave variable $v_{ucd}[i]$ to an appropriately scaled wave variable $v_{ucd}(t)$ which is held for $T_s$ seconds.

### Communication Layer

The controller and the plant are implemented on two separated nodes which send their commands and measurements (precisely wave variables) using UDP protocol. The UDP packet rate naturally corresponds to the sampling rate of the controller.

The two nodes communicate with each other directly over wireless channel using the IEEE 802.11 MAC protocol. Here we consider a wireless channel with random errors. In IEEE 802.11, if a frame is corrupted due to channel errors, it will be retransmitted. When the number of retransmission reaches a certain limit, the frame will be dropped. According to [1], the value of retransmission limit depends on the size of the frame. For frames with sizes larger than RTSThreshold, LongRetryLimit of 4 times will be used; for frames smaller than RTSThreshold, ShortRetryLimit of 7 times will be used. To simplify the system model, we disable

the RTS/CTS mechanism by setting RTSThreshold to a very large value in the IEEE 802.11 MAC protocol.

It is obvious that the MAC retransmission strategy may affect the NCS performance. Given a packet loss probability, allowing a larger retransmission limit increases the chance of successful transmission of a particular packet. However, it can also result in a longer delay in the packet transmission, which can be harmful especially if the system is delay-sensitive. On the other hand, if a small retransmission limit is used, the packets may experience a higher drop rate, which can also degrade the system performance especially if the system is loss-sensitive. Yet, to identify the optimal MAC retransmission strategy, we need to investigate how much the delay and the loss will be factored into the NCS performance.

## Observations

To understand how the network loss and delay may affect the performance of the NCS and how retransmission strategy should be designed to minimize such effect, we perform a set of experiments using ns-2 simulator.

### Methodology

We implement the passive control architecture on top of IEEE 802.11 wireless network in ns-2 simulator. In our experiment, the sampling rates of the plant and the controller are both 20 samples/sec, which is also the UDP packet rate. The

packet size is 210 bytes. The wireless network has a capacity of 1Mbps. Each simulation runs for 100 seconds.

The velocity of the plant system $G_p(\tau_u)$ tracks a sinusoidal reference input $\dot{\theta}_r[i] = sin(\omega i)$ with $\omega = \frac{2\pi}{10}$. The performance of the system is evaluated using the instantaneous tracking error $J[i] = |\dot{\theta}[i] - \dot{\theta}_r[i]|$, where $\dot{\theta}[i]$ is the plant's output and $\dot{\theta}_r[i]$ is the reference input the plant is supposed to track. $J[i]$ demonstrates the tracking ability of the system.

In what follows, we first inspect how the network loss affects the plant output when network delay is negligible, then test the effect of network delay to the NCS in a loss free condition. We finally investigate the effect of the MAC retransmission limit on the NCS, which will establish the basis of our control algorithm for retransmission limit.

### Effect of Packet Loss

In this experiment, we disable the retransmission mechanism of IEEE 802.11 MAC so that each packet will only be transmitted once. In this case, the packet error directly translates to a packet loss.

Fig. III.2 demonstrates the system performance with different packet error probabilities. Fig. III.2(a) shows the average tracking error $\bar{J}$ over all sampling points with standard deviation $J_d$. Fig. III.2(b) shows the maximum tracking error $J_m$ experienced out of all samples corresponding to different error probabilities. With the increase of the error probabilities, $\bar{J}$, $J_d$ and $J_m$ all increase. When the

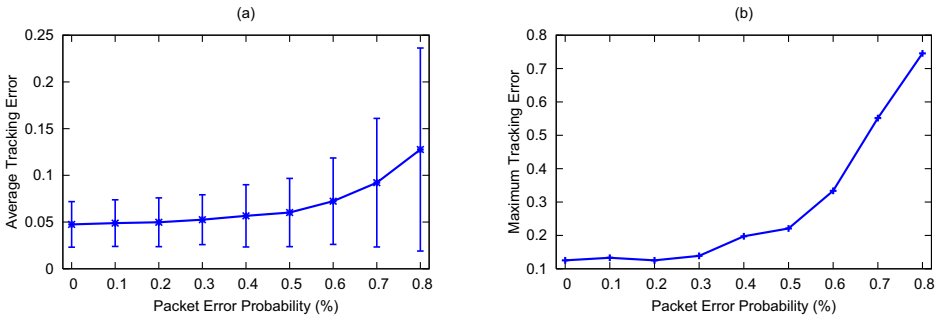


Figure III.2: Impact Of Loss Rate On System Performance

error probability is small, only a few packets are dropped. If the samples are exchanged frequently enough, the plant and the controller can still keep track of each other's status. However, when too many packets are dropped, the plant cannot interpret the control command correctly, while the controller no longer has the right velocity information of the plant. For example, when the packet error probability is 70%, the errors suddenly become very large. Further experiments using different passive controllers on different signals and with different sampling rates all show the same trend. Yet the exact mathematical relation between error probability and tracking error varies depending on these system parameters (*e.g.*, signal, sampling rate).

### Effect of Network Delay

In this experiment, the controller and the plant work in a loss free network. A varying amount of delay $D$ is introduced before the packet transmission at the MAC layer. The value of $D$ can be regarded as the time a packet spends in channel

contention. Thus it is an indicator of the intensity of background traffic in the wireless network. Fig. III.3 shows the NCS performance in terms of average and maximum tracking error under different values of $D$.

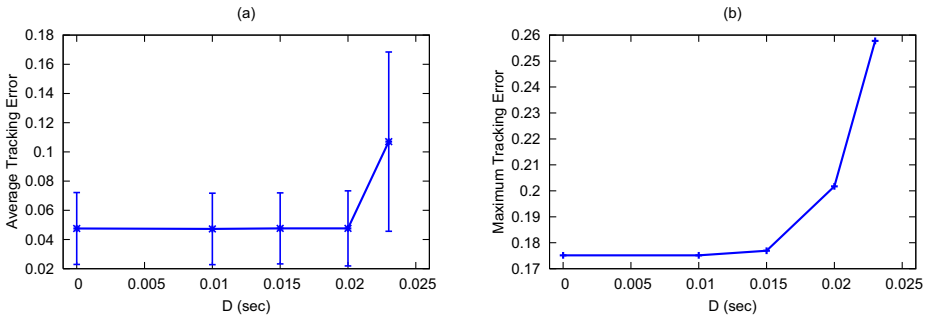


Figure III.3: Impact Of Transmission Delay On System Performance

We observe that when $D$ is small, the system performance does not change much. When exceeding certain value, the performance degrades significantly. Consider that the controller is a discrete time system, when the delay is smaller than one discrete time step, the controller can still receive the signals for the next sampling period in time, so the performance does not deteriorate. However, when a larger delay is experienced, signals cannot reach the other end within one sampling period. In this case, When the controller receives a signal from the plant, the state of the continuous plant may have already changed considerably. But the controller will still produce a control signal for the plant using the received plant state information. This control signal will also experience transmission delay before it arrives at the plant. When these two delays are combined, the plant will

deviate more from the expected trajectory. It is important to note that increasing $D$ also reduces the departure rate of the packets and may cause queueing delay (*e.g.*, when $D = 0.023s$[1]), so the actual delay experienced by the system is much larger. Due to this reason, this delay threshold ($0.02$ in this experiment) highly depends on the sampling rate, the signal of the system and the channel capacity. Further experiments on different signals and with different sampling rates validate the same observation.

### Effect of MAC retransmission

In IEEE 802.11-based wireless networks, packet loss will be recovered through retransmission up to a limit. As a result, a packet may experience higher delay before getting successfully transmitted. In the first two experiments, we have demonstrated that increasing either the network loss or delay will harm the performance of the control system. To achieve the optimal NCS performance, the retransmission strategy needs to be carefully designed to provide the best trade-off between the packet delay and the loss. Here, we exam the impact of MAC retransmission strategy by varying the retransmission limit and measure the system performance.

Fig. III.4 shows the average and maximum tracking errors of the system under different retransmission limits, with varied packet error probabilities and delay

[1]To have a controlled environment, where only the impact of delay is assessed, the queue length and the value of $D$ are carefully chosen in this experiment, making sure no queuing loss is incurred.

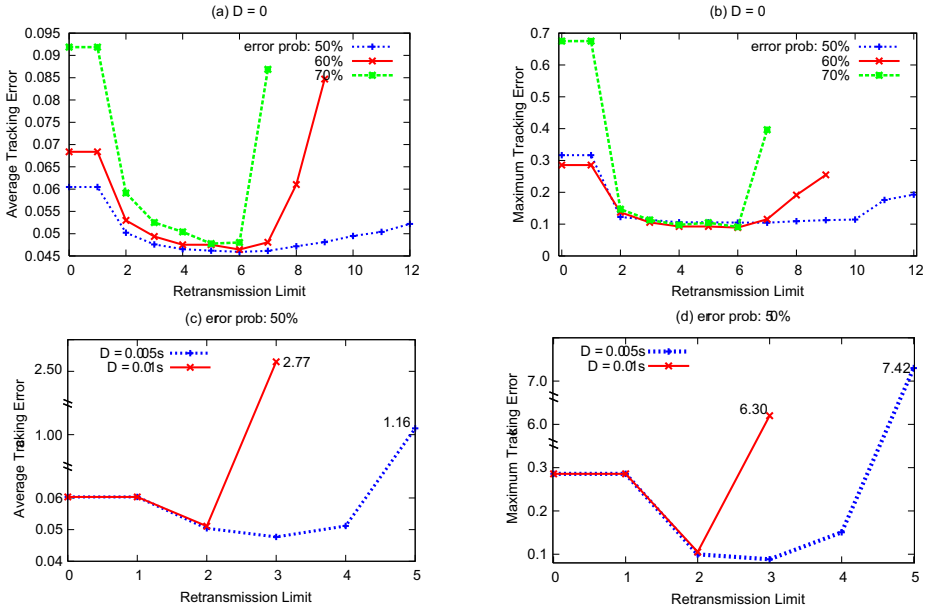


Figure III.4: Impact Of Retransmission Limit On System Performance

parameters $D$. We observe that the relationship between the tracking error and the retransmission limit follows a convex curve in all experiments. When the retransmission limit is small, high packet loss rate leads to relatively large tracking errors. When the limit is too high (*e.g.*, 7 defined as ShortRetryLimit in IEEE 802.11), the tracking error raises due to large delay. Experiments with different signals and sampling rates confirm the convex relation between the retransmission limit and the tracking error. This observation implies that there exists a unique optimal value for retransmission limit. Yet this optimal value varies depending on the channel error probability, background traffic, signal property, etc. We summarize our observations below:

- The NCS performance based on passive controller is negatively affected by network factors, including packet losses and transmission delays.
- The MAC-layer packet retransmission limit and the NCS performance follows a convex relation, which shows the existence of a unique value for retransmission limit that optimizes the NCS performance. This optimal value depends on many control system properties, such as sampling rate, signal type, as well as network factors, such as channel error probability, background traffic, etc.
- The fixed retransmission limit (4 as LongRetryLimit, 7 as ShortRetryLimit) used in IEEE 802.11 is not optimal for the NCS performance, considering the dynamics in wireless network with bursty traffic and fluctuating channel conditions. To achieve the optimal NCS performance, the retransmission limit needs to be dynamically adjusted based on the system property.

## MAC Control Design

In this section, we present a MAC-layer controller that dynamically adjusts the retransmission limit under different network conditions. Most NCS systems define a maximum performance error they can tolerate. We use $\tilde{J}$ to represent this threshold. Our MAC controller can achieve the following two goals: (1) keep the NCS performance within this error threshold; (2) minimized the NCS performance error (when the error threshold is not achievable, or set to a very small value).

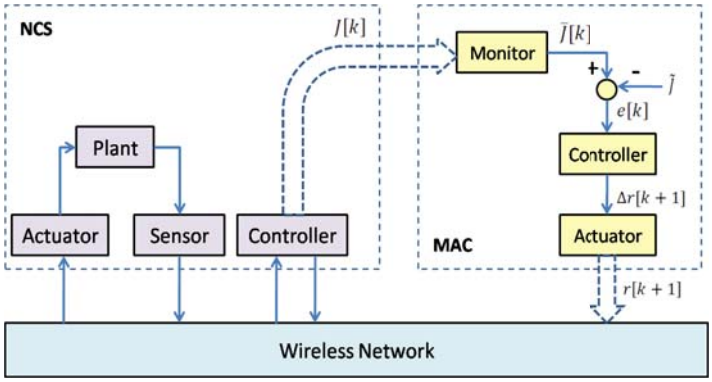


Figure III.5: MAC Controller Architecture

Fig. III.5 is an overview of the MAC controller architecture, which consists of a monitor, a controller and an actuator. The monitor resides on the same node as the NCS controller. It interfaces with the NCS and measures the average track error $\bar{J}[k]$ of the last $m$ samples in the current MAC sampling period $k$[2]. The monitor will then derive the difference $e[k]$ between $\bar{J}[k]$ and $\tilde{J}$ as $e[k] = \bar{J}[k] - \tilde{J}$ and pass it to the controller. Let $r[k+1]$ be the retransmission limit that will be used in the MAC layer at time $k+1$ and $\Delta r[k+1]$ be the adjustment of $r[k+1]$. The controller will compute $\Delta r[k+1]$ based on $e[k]$, and send it to the actuator. The actuator interfaces with the wireless network and tunes the retransmission limit to $r[k+1] = r[k] + \Delta r[k+1]$. For the node on which the NCS controller resides, the actuator will directly pass the new retransmission limit to the MAC layer. For the node on which the plant resides, the MAC actuator sends the new retransmission limit information on a separate packet or piggyback on a data packet.

[2]As explained below, the sampling period of the MAC controller is larger than the sampling period of the NCS controller.

The MAC controller at the communication layer and the "main" NCS controller at the control layer form a time-scale-decomposed system, where the MAC controller is the slow system that evolves with a larger time scale and operates with a lower sampling rate. This allows the NCS performance to converge with the new retransmission limit.

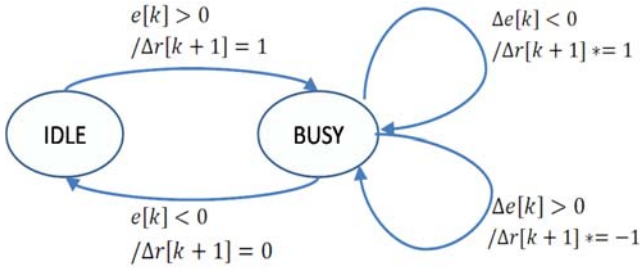


Figure III.6: MAC Controller Design

To indicate whether the NCS system is within its error threshold, the MAC controller maintains two states, IDLE and BUSY, as in Fig. III.6. Initially the controller is at the IDLE state. If the measured error is within the threshold (*i.e.*, $e[k] < 0$), the controller will remain at the IDLE state with $\Delta r[k+1]$ set to $0$, meaning no changes to the retransmission limit. When the measured error is greater than the threshold (*i.e.*, $e[k] > 0$), the controller will transit from the IDLE state to the BUSY state and set $\Delta r[k+1]$ to $1$.

At the BUSY state, the MAC controller will determine the change of retransmission limit for time slot $k+1$ based on the change of the tracking error from time $k-1$ to $k$ $\Delta e[k] = \tilde{e}[k] - \tilde{e}[k-1]$. If the tracking error becomes smaller (*i.e.*,

$\Delta e[k] < 0$), which means the previous change of the retransmission limit $\Delta r[k]$ decreases the tracking error of the NCS, the controller will keep the same change to the limit:

$$\Delta r[k+1] = \Delta r[k] \tag{III.1}$$

If $\Delta e[k] > 0$, which means $\Delta r[k]$ increases the tracking error, the controller will then change the retransmission limit towards the opposite direction:

$$\Delta r[k+1] = -\Delta r[k] \tag{III.2}$$

Whenever the tracking error falls below $\tilde{J}$, the controller will transit the state back to IDLE.

**Discussion**. We make the following important notes about the MAC controller design:

- This MAC controller will adjust the retransmission limit so that the NCS performance error is within a predefined threshold. When there are multiple values of the retransmission limit that can enable the system to perform within the error threshold, the MAC controller may bring the limit to any of these values.

- If the threshold is too small that no feasible retransimission limit value can bring the NCS system within this threshold, the MAC controller will bring the retransmission limit close to the optimal value where the NCS performance error is minimized. This is ensured by the convexity property in the

relationship between retransmission limit and the NCS performance as we have demonstrated in Section III. We can exploit this feature to achieve the goal of optimal NCS performance by manually setting the threshold to a very small value.

- In its current design, though the MAC controller can bring the retransmission limit to the optimal value, it can not stay at this point. Rather it will oscillate around it. To ensure the controller stabilizes at the optimal value, we improve its design in two ways. First, a counter is used to count the number of times the controller oscillates around one retransmission limit. If the counter exceeds a certain value, we consider the limit as the optimal value and fix the retransmission limit to it. After some time, the MAC controller will resume to the BUSY state in case the optimal retransmission limit has changed.

## Simulation Study

### Simulation Setup

We implement the MAC controller in the ns-2 simulator and use it to evaluate the NCS performance under a variety of network scenarios. The NCS used in the experiment consists of a passive controller and a plant. It has the same configuration as the one presented in Section III. The error threshold $\tilde{J}$ is set to $0.1$ in all

experiments. Two aspects of the NCS system will be examined: (1) performance of the MAC controller in terms of the convergence behavior of the retransmission limit, and (2) performance of the overall NCS in terms of the absolute tracking error of the plant output. We will compare the performance of NCS that operates with our MAC controller which dynamically adjusts the retransmission limit, with the NCS that operates over traditional IEEE 802.11-style MAC, where retransmission limit is fixed.

## Simulation Results

### Impact of Initial Value of Retransmission Limit

In this experiment, the controller and the plant operate and communicate in a network with no background traffic. The wireless channel error probability is 83%.

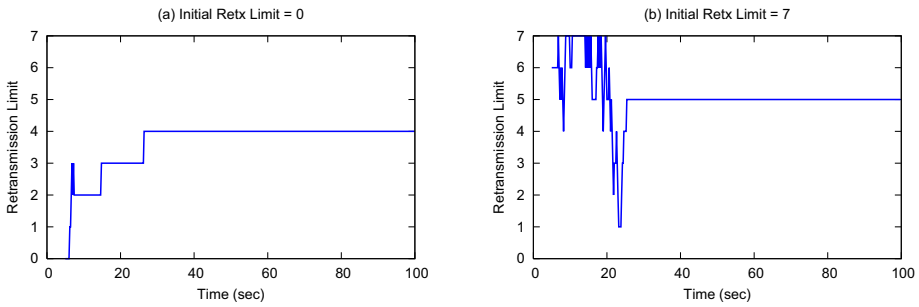


Figure III.7: Convergent Retransmission Limit With Different Initial Values

Fig. III.7 (a) and (b) display the retransmission limit adjusted by the MAC controller from initial values of $0$ and $7$ respectively. In both cases, the MAC controller converges after 25 seconds. The retransmission limit has different stable values. This is because the tracking error of the NCS performance is below the error threshold for both retransmission limits of $4$ and $5$. When the initial value is $7$, a larger delay may cause a long queue in the MAC, and the change of the retransmission limit cannot be reflected onto the NCS performance immediately. For the rest of the simulation, the initial values of the retransmission limit are all $0$.

**Impact of Background Traffic**

Background traffic in the wireless network will increase the time a packet spends on contending for the medium access, which in turn will increase the packet delay. We now study how the background traffic will affect the behavior of the MAC controller and the overall system performance. With $50\%$ packet error probability, three pairs of background traffic nodes are introduced to the network. The traffic between each pair of nodes follows a poisson distribution with rate of $200$ packets/sec and packet size of 210 bytes.

Fig. III.8 (a) and (b) show the tracking errors of using the MAC controller and using different fixed retransmission limits. Note that In Fig. III.8 (b), the retransmission limits of $6$ and $7$ are not shown as the tracking errors are in much larger scales than the others. We could see that the optimal values of retransmission

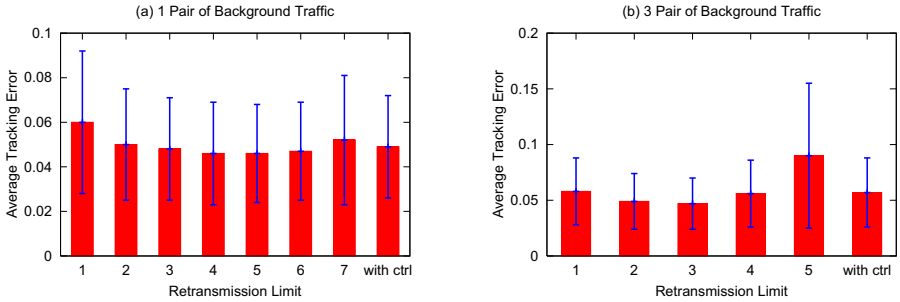


Figure III.8: Tracking Errors with Different Background Traffic

limit differ under different background traffic. The MAC controller converges to the optimal retransmission limits of 4 and 3, respectively in both scenarios. We could also see that the NCS performance under the MAC controller is close to the optimal fixed retransmission limit. The slight difference is due to the sub-optimal NCS performance during the initial transient phase when MAC controller is adjusting the retransmission limit.

### Impact of Channel Error Probability

This experiment tests the impact of packet error probability on the MAC controller and the NCS performance. With one pair of poisson background traffic as in the previous experiment, we simulate the scenarios with the packet error probability of 40% and 80%.

When the packet error probability is 40% as in Fig. III.9, several retransmission limits can provide optimal performance for the MAC controller. In this case, the MAC controller can converge to any of the retransmission limits to achieve

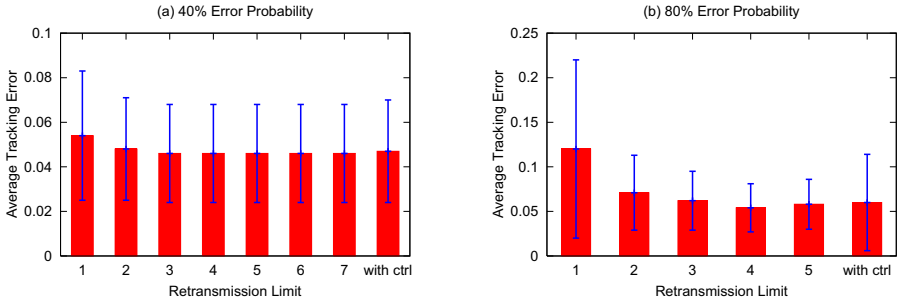


Figure III.9: Tracking Errors With Different Error Probability

the optimality of the NCS. Here it converges to 3. When the error probability increases to 80%, the MAC controller must adjust to the correct retransmission limit to best support the NCS, and here it converges to 5. The deviation of the tracking error in the MAC controller from the optimal fixed retransmission limit is due to the initial adaptation of the MAC controller.

# CHAPTER IV

# DISTRIBUTED SAMPLING RATE ADAPTATION FOR WIRELESS NETWORKED CONTROL SYSTEMS

## Introduction

Networked Control Systems (NCS) are control systems where actuators, sensors, controllers and the systems to be controlled exchange information and coordinate their operation via a communication network. NCS are increasingly deployed over wireless networks, as they provide great convenience in terms of deployment and mobility support. In wireless networks, where channel resource is constrained and available bandwidth varies due to dynamic user behaviors and external interference sources, the capability of adaptive resource management is crucial for NCS to fully exploit the available resource for better performance, and avoid network congestion.

This chapter investigates sampling rate adaptation as a mechanism of adaptive resource management for wireless NCS. In a digital control system, sampling rate specifies how often the system components exchange information, and thus determines how well the digital controller approximates its continuous equivalent. Environmental disturbances can lead to system instability and cause uncertainty in system behavior. Intuitively, the larger the sampling rate, the more frequent state updates the components of a NCS will receive. Consequently, it can have a better ability to reduce the effect of such disturbances. On the other hand, the

sampling rate is naturally linked to traffic load on the network. From the network perspective, the sampling rates should be limited to avoid congestion and packet losses, which will deteriorate the NCS performance [6].

In this chapter, we formulate the objective of NCS sampling rate adaptation as an optimal resource allocation problem, where the NCS performance is maximized subject to the wireless bandwidth constraint. The key challenge is how to quantify the relationship between the NCS performance and its sampling rate. In this chapter, we focus on the NCS robustness, *i.e.*, its capability in handling disturbances. Formally, we use the noise covariance matrix of the control system to characterize system performance with respect to its ability of disturbance rejection. We then use a *utility function* to characterize the relationship between the NCS robustness to disturbance and the sampling rate, and formally define it as the ratio of its digital controller robustness to the robustness of its continuous equivalent. We show that this utility function is a strictly concave function of the sampling rate, when random white noise is considered as the model of control system disturbance. The concavity of the utility function reflects the marginal return on the NCS performance when its sampling rate increases.

Based on this optimal rate allocation problem, a price-based algorithm is developed for distributed sampling rate adaptation. In this algorithm, a price signal is generated for each contention region of the wireless channel as a function of the traffic load in the region. The NCS then adapts its sampling rate based on its utility function so that its net profit, which is the difference between the utility and

the cost (product of price and rate), is maximized. The stability of this sampling rate adaptation algorithm is proved under our NCS utility model.

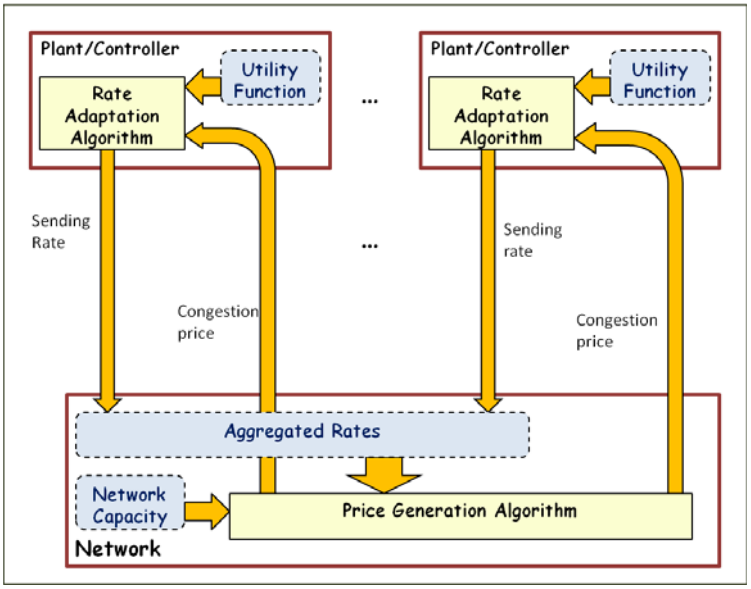


Figure IV.1: Overview of the Networked Control System

Fig. IV.1 shows an overview of our NCS system model and outlines our price-based sampling rate adaptation solution. Pairs of plant and controller components communicate via networks. When a sender sends a packet into the network, based on the aggregated rates at each contention region, a price is computed reflecting the channel congestion levels according to a price generation algorithm. The price is then piggy-backed onto the packet from the receiver and fed back to the sender. A rate adaptation algorithm plugged in the components calculates the proper rate using the information of the congestion price as well as a utility function.

The main contributions of this chapter are summarized as follows. First, we formally characterize the relationship between the NCS performance and its sampling rate using a utility function, which is based on its ability of disturbance rejection using the ratio and derive a utility metric as a function of its sampling rate. Our work provides a fully distributed dynamic network resource management solution for wireless NCS. It fully exploits wireless network resources and maximizes the NCS performance. Second, the NCS sampling rate adaptation algorithm is evaluated in an integrated simulation environment [99] that consists of Matlab and ns-2. Using ns-2 – a packet-level network simulator that implements all the details of the network protocol stack, allows highly accurate evaluation of network effects on the NCS performance, which is impossible by using Matlab/Simulink alone.

The remainder of this chapter is organized as follows. In Sec. IV, we present the control system model and formally define the utility function to characterize its performance. In Sec. IV, we present the wireless network model, formulate the problem of optimal rate allocation and derive the price-based rate adaptation algorithm. Finally we evaluate the algorithm using our Networked Control System Wind Tunnel (NCSWT) simulation environment in Sec. IV.

## Control System Model

In this section, we present on the model of control system and provide a formal description of the control system performance as a function of the sampling

rate. We consider the passivity-based networked control architecture in this chapter. In the sub-sections we describe the continuous-time and discrete time control systems used to determine our utility function, which is described in the last subsection. In Fig. V.3, $G_p(s)$ is the plant system to be controlled. The composite

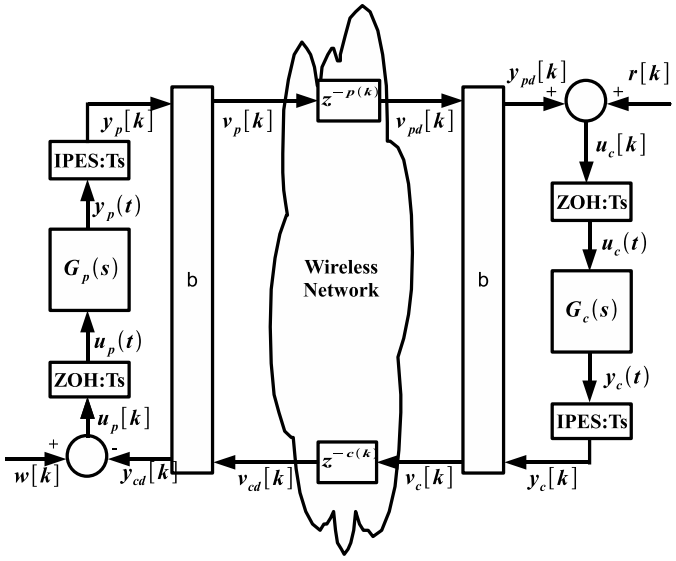


Figure IV.2: Passivity Based Control Architecture Over Wireless Networks

dynamics of the plant is by design passive, which means it ensures stability of the NCS in the presence of network uncertainties such as packet losses and time varying delays. The controller, $G_c(s)$ which is also passive, controls the plant to behave in a desired manner. The block $b$ transforms the power variables (*i.e.*, the direct inputs and outputs of plant and controller) into wave variables for communication over a wireless network. These wave variables preserve the passivity of the transmitted information over the network. The inner product equivalent sampling

(IPES) and zero order hold (ZOH) blocks on both the plant and control sides are used to implement the passive discretization of the continuous time systems. We refer readers to [56] for a detailed description and proofs pertaining to the passive control architecture.

Continuous-time Control System

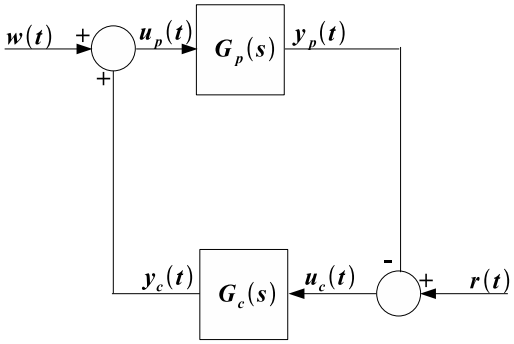


Figure IV.3: Continuous-time control system block diagram

The continuous-time control system involves a continuous-time plant interacting with a continuous-time controller as shown in Fig. IV.3. The plant $G_p(s)$ is described by the following state-space representation.

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) + B_w w(t) \quad \text{(IV.1)}$$

$$y_p(t) = C_p x_p(t) \quad \text{(IV.2)}$$

where $x_p(t) \in \Re^n$ denotes the plant states, $u_p(t) \in \Re^m$ denotes the control input, $w(t) \in \Re^m$ is the disturbance input modeled as a zero-mean white noise process, and $y_p(t) \in \Re^m$ is the plant output. $A_p$, $B_p$, and $B_w$ define the plant state matrices and $C_p$ defines the plant output matrix.

The state space of the continuous-time controller $G_c(s)$ is

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) \tag{IV.3}$$

$$y_c(t) = C_c x_c(t) + D_c u_c(t) \tag{IV.4}$$

where $x_c(t) \in \Re^n$ denotes the controller state, and $u_c \in \Re^m$ denotes the error signal, or the difference between the plant output $y_c(t) \in \Re^m$ and the reference signal input $r(t) \in \Re^m$. $A_c$ and $B_c$ define the controller state matrices, while $C_c$ and $D_c$ define the controller output matrices.

From the plant and controller state-space description, the closed loop state space form of the system can be represented as follows:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_c(t) \end{bmatrix} \\ &= \begin{bmatrix} A_p - B_p D_c C_p & B_p C_c \\ -B_c C_p & A_c \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} \\ &+ \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix} r(t) + \begin{bmatrix} B_w \\ 0 \end{bmatrix} w(t) \end{aligned} \tag{IV.5}$$

$$
\begin{aligned}
y &= \begin{bmatrix} y_p(t) \\ y_c(t) \end{bmatrix} \\
&= \begin{bmatrix} C_p & 0 \\ -D_c C_p & C_c \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} 0 \\ D_c \end{bmatrix} r(t)
\end{aligned} \tag{IV.6}
$$

The covariance matrix of the zero-mean white noise process of the continuous system can be defined by [30]

$$E[w(t)w^T(t+\tau)] = Q\delta(\tau) \tag{IV.7}$$

where $E$ denotes the expected value and $Q$ represents the power spectral density of $w$, or the continuous-time noise covariance matrix. The power spectral density can also be referred to as the "white noise intensity" or mean-square spectral density. The continuous-time state covariance matrix $P_c$ can be described by

$$P_c(t) = E[x(t)x^T(t)] \tag{IV.8}$$

Based on the knowledge of $Q$, the steady state value of the state covariance can be obtained by the equation [60]

$$A_{cl}P_c + P_c A_{cl} + B_{wcl}QB_{wcl}^T = 0 \tag{IV.9}$$

where the matrices $A_{cl}$ and $B_{wcl}$ denote the closed loop matrices, or the coefficients of $x(t)$ and $w(t)$ in Eq. (IV.5) respectively. From the resulting state covariance matrix, the root mean square of a state can then be determined. The Root-Mean-Square (RMS) of the plant state is equivalent to the standard deviation. For example in the response to white noise, assuming the system has only one plant state variable $x_{p1}$ and the result obtained for the plant state covariance is $v$. The RMS of the plant state will be equal to $\sqrt{v}$. If a plant has several states, we can use the one of the states as long as we compare it to the same discrete state obtained from the minimal realization of the discretized continuous system.

Discrete-time Control System

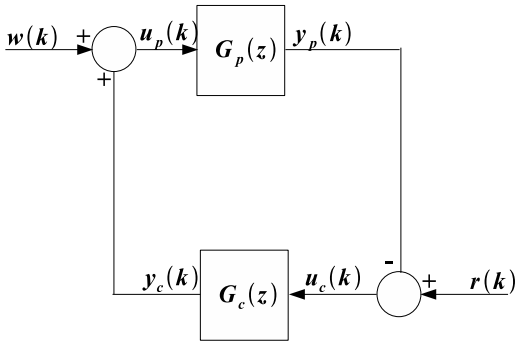


Figure IV.4: Discrete-time Control System Block Diagram

The continuous-time control system is usually implemented as a discrete-time control system via discretization. The discretization is executed with a sampling time $T_s$ using an inner-product equivalent sample and hold (IPESH) transform in

order to preserve the passivity properties of the system [54]. The resulting system is equivalent to a system with a discrete plant and discrete controller.

$G_p(z)$ represents the discrete-time equivalent of the continuous-time plant $G_p(s)$. The discrete-time state space of the plant can be given by

$$x_p(k+1) = \Phi_p x_p(k) + \Gamma_p u_p(k) + \Gamma_w w(k) \quad \text{(IV.10)}$$

$$y_p(k) = C_{pd} x_p(k) + D_{pd} u_p(k) + D_w w(k) \quad \text{(IV.11)}$$

The state space of the discrete-time controller $G_c(z)$ equivalent to the continuous-time $G_c(s)$ can be given by

$$x_c(k+1) = \Phi_c x_c(k) + \Gamma_c u_c(k) \quad \text{(IV.12)}$$

$$y_c(k) = C_{cd} x_c(k) + D_{cd} u_c(k) \quad \text{(IV.13)}$$

The overall closed-loop state equation can be determined from the discrete plant and discrete controller's state space representation. This can be described by

$$x(k+1) = \begin{bmatrix} x_p(k+1) \\ x_c(k+1) \end{bmatrix}$$

$$= \begin{bmatrix} \Phi_p - \Gamma_p D_{cd} S_f C_{pd} & \Gamma_p D_{cd} S_f D_{pd} C_{cd} \\ -\Gamma_c S_f C_{pd} & \Phi_c - \Gamma_c S_f D_{pd} C_{cd} \end{bmatrix} \begin{bmatrix} x_p(k) \\ x_c(k) \end{bmatrix}$$

$$+ \begin{bmatrix} \Gamma_p D_{cd}(1 - S_f D_{pd} D_{pc}) \\ \Gamma_c (1 - S_f D_{pd} D_{cd}) \end{bmatrix} r(k)$$

$$+ \quad \begin{bmatrix} \Gamma_w - \Gamma_p D_{cd} S_f D_w \\ -\Gamma_c S_f D_w \end{bmatrix} w(k) \tag{IV.14}$$

where $S_f = (I + (D_{cd} * D_{cd}))^{-1}$, and $I$ is the identity matrix.

Based on the knowledge of the continuous-time noise covariance matrix $Q$, the discrete-time noise covariance matrix $Q_d$ can be obtained using the Van Loan's algorithm [30] and can be defined as

$$Q_d = \int_0^{T_s} \Phi(\tau) B_{wcl} Q B_{wcl}^T \Phi^T(\tau) d\tau \tag{IV.15}$$

The steady state discrete-time state covariance matrix can then be obtained from the following equation

$$\Phi P_d \Phi^T + Q_d = P_d \tag{IV.16}$$

From the resulting state covariance matrix, the discrete RMS of the plant state can then be determined similar to the continuous-time case.

### Utility Function

When the system is discretized using certain sampling rate to implement a digital controller over a network or computer, the control system response to disturbances degrades compared to the continuous closed loop case. The level of degradation depends on the sampling rate. To characterize the impact of sampling

rate on the disturbance rejection ability of a digital controller, we consider the RMS ratio of the discrete-time system plant state and its continuous-time counterpart. Intuitively, increasing the sampling rate will decrease the RMS of the system's discrete plant state, where the continuous-time system establishes the lower bound of the RMS.

Formally, let's assume that the traffic from the controller to the plant shares the same data path as the traffic from the plant to the controller. We define the utility function of system $f$ as a function of its sampling rate $p_f$ as follows.

$$U_f(p_f) = \frac{RMS_{\texttt{continuous}}}{RMS_{\texttt{discrete}}(p_f)} \tag{IV.17}$$

If $T_s^f$ is the sampling time, $p_f = 1/T_s^f$.

Essentially, the utility function is established by comparing the performance of the discrete-time system with the continuous-time system. Using the continuous-time control system, we determine the ability of the control system to reject disturbance in the form of white noise. We then proceed to repeat a similar process when a digital implementation of the controller is used. This digital implementation depends on the chosen sampling rate. This utility function is essentially determined by the amount of degradation of the system response to white noise compared to the continuous closed loop system.

To demonstrate our utility function definition, we consider the following single-input-single output (SISO) linear-time invariant (LTI) system, without loss of generality and for simplicity. A plant system represents a single joint of a robotic arm,

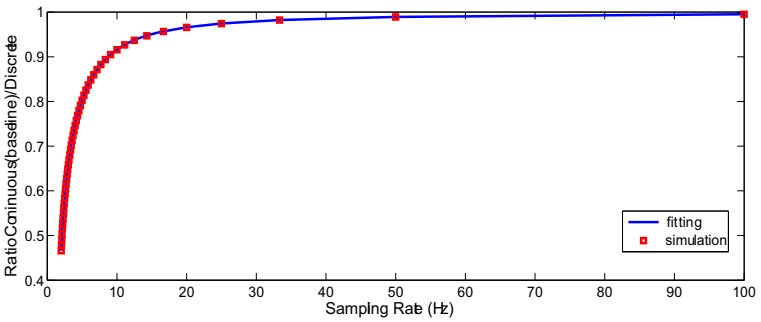


Figure IV.5: System Utility Function

with the transfer function $G_p(s) = \frac{1}{Js}$. The transfer function of the controller is given by $G_c(s) = \frac{K_p + K_d s}{s}$. With the following parameters, $J = 2.93$, $K_d = 32.1$ and $K_p = 8.2$. The utility function that characterizes the relationship between the sampling rate and this NCS system performance (as defined by the RMS ratio) is shown in Fig. V.4 as the red squares. It shows that the utility function is strictly concave, with the blue solid line perfectly fitting the utility function. The fitted function is in the form of

$$U_f(p_f) = \frac{p_1 * p_f^4 + p_2 * p_f^3 + p_3 * p_f^2 + p_4 * p_f + p_5}{p_f^4 + q_1 * p_f^3 + q_2 * p_f^2 + q_3 * p_f + q_4} \tag{IV.18}$$

where $p_1, p_2, p_3, p_4, p_5, q_1, q_2, q_3, q_4$ are the fitted parameters.

## Optimal Sampling Rate Adaptation

In this section, we first review the theoretical framework for optimal resource allocation in multi-hop wireless networks [131], and then present our price-based sample rate adaption algorithm.

### Wireless network model

We consider a network that consists of a collection of wireless nodes $V$. Nodes within the transmission range of each other can communicate directly, forming a wireless link $l \in L$. While nodes that are far away communicate via relays of other nodes. Here we consider the protocol model [36] to characterize the location-dependent contention and spatial reuse of the wireless communication in this network. We further adopt the model presented in [131] where maximal cliques in the contention graph of a wireless network are used to characterize the independent resource elements in wireless networks. Here we denote a resource element as $e \in E$. Each resource element has a finite capacity $C_e$.

Such a network is shared by a set of control systems $F$. For each control system $f \in F$, its plant and controller are hosted on two different nodes in the network. We assume the traffic from the controller to the plant and the traffic backwards share the same network path. Then the control system $f$ generates a bi-directional flow between these two nodes. The flow may go through multiple hops in the network and traverse a sequence of resource elements. Let $R_{ef}$ be the amount of resource element $e$ used by system $f$.

Optimal sampling rate allocation formulation

The problem of optimal rate allocation in the sense of maximizing the aggregated utility of all control systems in the network can be formulated into the following nonlinear optimization problem:

$$\mathbf{P}: \quad \textbf{maximize} \quad \sum_{f \in F} U_f(p_f) \tag{IV.19}$$

$$\boldsymbol{R} \cdot \boldsymbol{p} \leq \boldsymbol{C} \tag{IV.20}$$

$$\boldsymbol{p} \geq 0 \tag{IV.21}$$

The constraint (VII.3) comes from the resource constraint of the shared wireless channel, where $\boldsymbol{p} = (p_f, f \in F)$ and $\boldsymbol{C} = (C_e, e \in E)$ are vectors of sampling rates and resource capacities respectively. $\boldsymbol{R} = (R_{ef})_{|E| \times |F|} \cdot m_f$. $R_{ef}$ is a matrix with element $R_{ef}$ at row $e$ and column $f$ [131], and $m_f$ is the packet size of $f$. $m_f \times p_f$ will convert the sampling rate $p_f$ in the control system to the flow rate in the network. By optimizing toward such an objective, the solution guarantees the optimal resource utilization.

Price-based algorithm

In the above formulation, the representation of the utility function is essential to further inspect the optimization problem and implement the price-based framework. We observe that the utility function of NCS defined in Sec. V is differentiable and strictly concave. Thus, the objective function of $\mathbf{P}$ in Eq. (VII.2) is

differentiable and strictly concave. Further, the feasible region of the optimization problem in inequality (VII.3) and (VII.4) is convex and compact [131]. According to the non-linear optimization theory, unique optimal solution to the resource allocation problem $\mathbf{P}$ exists.

Now we consider the dual problem $\mathbf{D}$ of $\mathbf{P}$ using its Lagrangian form:

$$\mathbf{D}: \quad \min_{\boldsymbol{\mu}\geq 0} D(\boldsymbol{\mu}) \tag{IV.22}$$

$$\begin{aligned} D(\boldsymbol{\mu}) &= \max_{p_f \geq 0} L(\boldsymbol{p}; \boldsymbol{\mu}) \\ &= \sum_{f\in F} \max(U_f(p_f) - p_f m_f \sum_{e\in E} \mu_e R_{ef}) \\ &\quad + \sum_{e\in E} \mu_e C_e \end{aligned} \tag{IV.23}$$

$\boldsymbol{\mu} = (\mu_e, e \in E)$ is a vector of Lagrange multipliers, and may be interpreted as the implied cost, or *penalty*, of a subflow accessing the resource element $e$. In other words, $\mu_e$ is the *shadow price* of resource $e$. The price of a control system $f$, $\lambda_f = \sum_{e\in E} \mu_e R_{ef}$ can be interpreted as that system $f$ needs to pay for all the resources it uses. Note that for each wireless link, its price is the aggregated price of all the resources that it belongs to.

By solving the dual problem $\mathbf{D}$, the optimal rate for system $f$ can be derived from

$$U_f'(p_f) - \lambda_f = 0 \tag{IV.24}$$

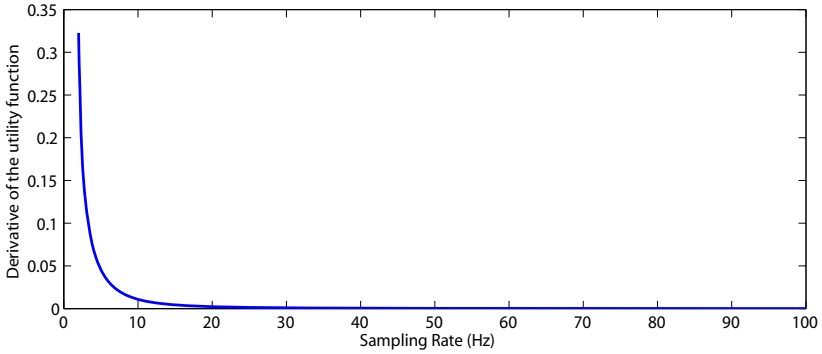


Figure IV.6: Derivative of the Utility Function

Considering that the utility function is complicated as shown in Eq. (IV.18), we use an approximated approach to solve $p_f$ from Eq. (IV.24). Following the example in Sec. V, Fig. IV.6 represents the relationship between the sampling rate and price. By switching the $x$ and $y$ axes and curve fitting the plot, the sampling rate $p_f$ can be represented as a function of price $\lambda_f$

$$p_f(\lambda_f) = a * \lambda_f^b \tag{IV.25}$$

where $a$ and $b$ are the fitted parameters. Thus the optimal rate for system $f$ can be approximated by substituting $\lambda_f$ into Eq.(IV.25).

Detailed derivation of $\boldsymbol{D}$ can be found in [131], while the adjustment of $\boldsymbol{\mu}$ can be represented as follows:

$$\frac{d}{dt}\mu_e(t) = \gamma\lceil(\sum_{e\in E} p_f(\lambda_f(t))m_f R_{ef} - C_e)\rceil^+ \tag{IV.26}$$

Alternatively, the price update algorithm can be represented in the following discrete time system.

$$\mu_e(i+1) = \lceil \mu_e(i) - \gamma(C_e - \sum_{e \in E} p_f(\lambda_f(i)) m_f R_{ef}) \rceil^+ \tag{IV.27}$$

Eq. (VII.7) reflects the law of supply and demand. If the demand for channel $e$ exceeds its supply $C_e$, the resource constraint is violated. This will cause the increase of the channel price $\mu_e$. Otherwise, $\mu_e$ is reduced. The discrete time index $i$ in Eq. (VII.7) is different from $k$ in the control system to ensure system convergence. As NCS is closed-loop, prices to controllers are appended onto the packets from the plants occupying only few bytes, and vise versa. This will not affect the overall network bandwidth much. Now we show the stability property of the price-based rate adaptation algorithm in the following theorem.

**Theorem 1**. *Let $\mathcal{V}(\mu)$ be defined as*

$$\mathcal{V}(\mu) = \sum_{f \in F} \int_0^{\sum_{e \in f} \mu_e} (U_f'(\eta))^{-1} d\eta - \sum_{e \in E} \int_0^{\mu_e} q_e(\eta) d\eta$$

*$\mathcal{V}(\mu)$ is a strictly concave function and a Lyapunov function for the system of equations (IV.24)-(VII.7). The unique value $\mu$ maximizes $\mathcal{V}(\mu)$ and is also a stable point of the system where all trajectories converge.*

*Proof.* We first will prove the strict concavity of the first term in $\mathcal{V}(\mu)$. As $U_f(\cdot)$ is strictly concave, its derivative $U_f'(\cdot)$ exists and is strictly decreasing. Considering the utility function obtained in Sec. V, its derivative is shown in Fig. IV.6

as an example. Further we observe that $U_f''(\cdot) \leq 0$ is increasing. In addition, $(U_f'(\cdot))^{-1}$, the inverse function of $U_f'(\cdot)$ has the same monotonicity as $U_f'(\cdot)$. By integrating $(U_f'(\cdot))^{-1}$, the resulted function will have the same monotonicity as $U_f(\cdot)$, which is strictly concave. Further, as the definition domain of $U_f(\cdot)$ is the codomain of the inverse function of $U_f'(\cdot)$, which is greater than $0$, the first term of $\mathcal{V}(\mu)$ is also greater than $0$.

Second, we prove the convexity of the second term in $\mathcal{V}(\mu)$. Let $q_e(\eta) = C_e\eta/(\eta+\varepsilon)$ [50]. It is a continuous and strictly increasing function of $\eta$. $q_e(\eta)$ arbitrarily closely approximates $C_e$ for a small positive $\varepsilon$. The strict concavity and positivity of the first term in $\mathcal{V}(\mu)$, as well as the assumptions on $q_e$ ensure that $\mathcal{V}(\mu)$ is strictly concave on $\mu \geq 0$ with an unique interior maximum $\mu$. It is determined by setting $\mathcal{V}'(\mu) = 0$.

$$\begin{aligned} \frac{\partial}{\partial \mu_e}\mathcal{V}(\mu) &= \sum_{e\in f}(U_f'(\sum_{h\in f}\mu_h))^{-1}R_{ef} - q_e(\mu_e) \\ &\geq \sum_{e\in f}(U_f'(\sum_{h\in f}\mu_h))^{-1}R_{ef} - C_e \end{aligned} \tag{IV.28}$$

With Eq. (IV.24), we have $(U_f'(\lambda_f))^{-1} = p_f$, so

$$\begin{aligned} \frac{d}{dt}\mathcal{V}(\mu(t)) &= \sum_{e\in E}\frac{\partial \mathcal{V}}{\partial \mu_e}\cdot\frac{d}{dt}\mu_e(t) \\ &\geq \gamma\sum_{e\in E}(\sum_{e\in f}p_f(\sum_{h\in f}\mu_h(t))R_{ef} - C_e)^2 \end{aligned}$$

This establishes that $\mathcal{V}$ is strictly increasing with $t$ unless $\mu(t) = \mu$, where the unique value $\mu$ maximizes $\mathcal{V}$. Thus function $\mathcal{V}$ is a Lyapunov function for the system (IV.24)-(VII.7), and the theorem follows. □

## Performance Evaluation

In this section, we evaluate the price-based sampling rate adaptation algorithm in the wireless networks using an integrated simulation environment named Networked Control System Wind-Tunnel (NCSWT) environment [99]. NCSWT integrates two simulators Matlab and ns-2. It is built based on the HLA standard. The tool allows us to simulate control system models in Matlab/Simlink and network models in ns-2. Using ns-2 – a packet-level network simulator that implements all the details of the network protocol stack, allows us to perform highly accurate evaluation of network effects on the NCS performance, which is impossible by using Matlab/Simulink alone.

### Simulation Setup

In our experiment, the network system consists of three pairs of plants and controllers, all of which have the same utility function as presented in Section V. The interfere range is set to 250m. The wireless network uses a single channel with the capacity of $1Mbps$. The packet size is $500$ bytes. Each simulation runs for $250$ seconds.

The velocity of the plant system tracks a sinusoidal reference input $r[k] = sin(\omega k)$ with $\omega = \frac{2\pi}{40}$ unless explicitly addressed. The power spectral density of the white noise is set to 1. The default value of $\gamma$ is set to $3 \times 10^{-8}$.

Three aspects of the system are examined:

1. performance of the control algorithm in terms of the convergence behavior of the plant output.

2. performance of the network in terms of its convergence behavior of the flow rate.

3. performance of the overall NCSWT in terms of the difference between the plant output and the reference signal.

## Simulation Results

### Comparison with Fixed-Rate Control Systems

We first inspect how our price-based dynamical sampling rate adaptation algorithm performs. It is also compared with classical systems with fixed sampling rates. The initial sampling time of plants or controllers is $0.1$ second, which is $40Kbps$ of flow rate. In the price-based rate adaptation system, the initial price is set to $5 \times 10^{-5}$, corresponding to the sampling flow rate is $80Kbps$.

Fig. IV.7 (a) and (b) show the plant outputs of three control systems. In Fig. IV.7 (b), the velocity of the plant system tracks a faster sinusoidal reference input $r[k] = sin(\omega k)$ with $\omega = \frac{2\pi}{30}$. Both outputs closely follow the reference

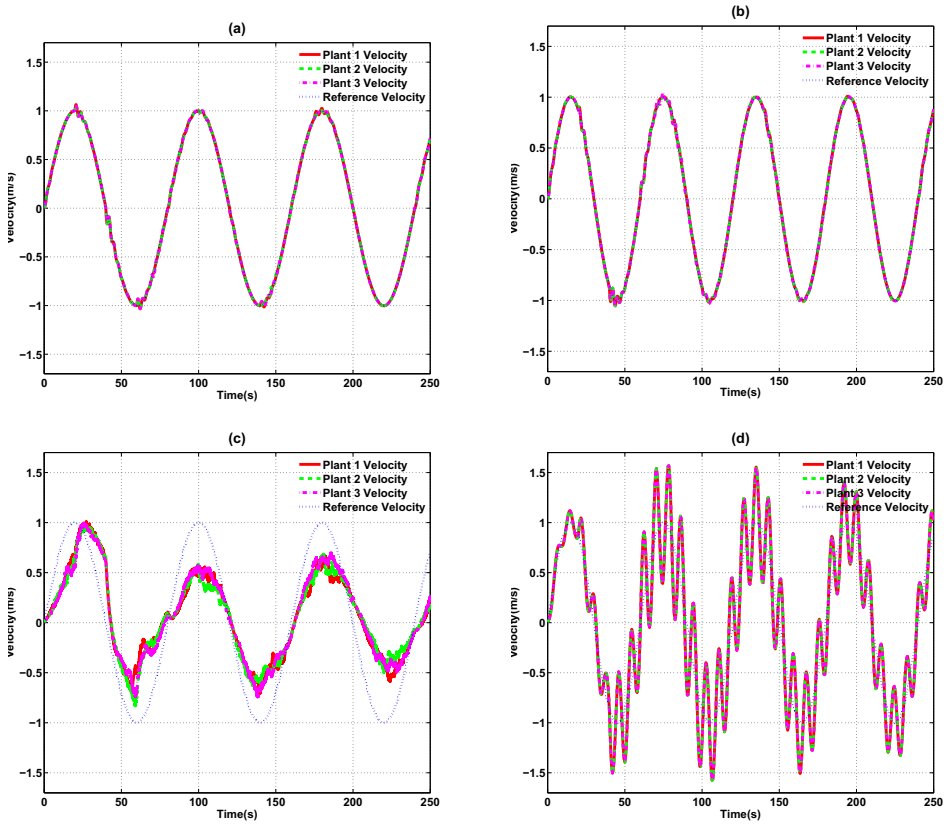


Figure IV.7: Plant Outputs with Price-Based Algorithm and Fixed Sampling Times

trajectory. The differences between the plant outputs and the reference trajectory quickly diminish every time when noise signal appears.

We then implement the classical rate allocation scheme with a fixed sampling flow rate different from the converged optimal sampling flow rate. Fig. IV.7 (c) and (d) show the plant outputs using fixed sampling times of $0.01s$ and $0.5s$, corresponding to the rates of $400Kbps$ and $8Kbps$ for each flow. The outputs hardly track the reference trajectory. In Fig. IV.7 (c), the aggregated flow rate, which is $400 \times 6 = 2400Kbps$, is much larger than the channel capacity, and is also much larger than the converged optimal flow rate of $80Kbps$. Many packets are dropped due to congestion in the network, and long-time delay is also introduced due to severe channel contention. With white noise disturbance in the system, the plants suffer large tracking error and long convergence time. In the passive system, it is exhibited as a decreased amplitude. In Fig. IV.7 (d), the flow rate is too small, so the controller cannot be notified in time about the occurrence of the white noise disturbance. Thus the outputs of the plants experience big oscillations and cannot converge.

**Impact of Parameters**

We now study the stability of the system and the impact of different parameters by investigating the instantaneous behavior of the plant output and the flow rate.

The impact of initial value of the price is evaluated. Fig. V.9 shows the convergence of plant outputs and flow rates when different initial prices of $10^{-5}$ and

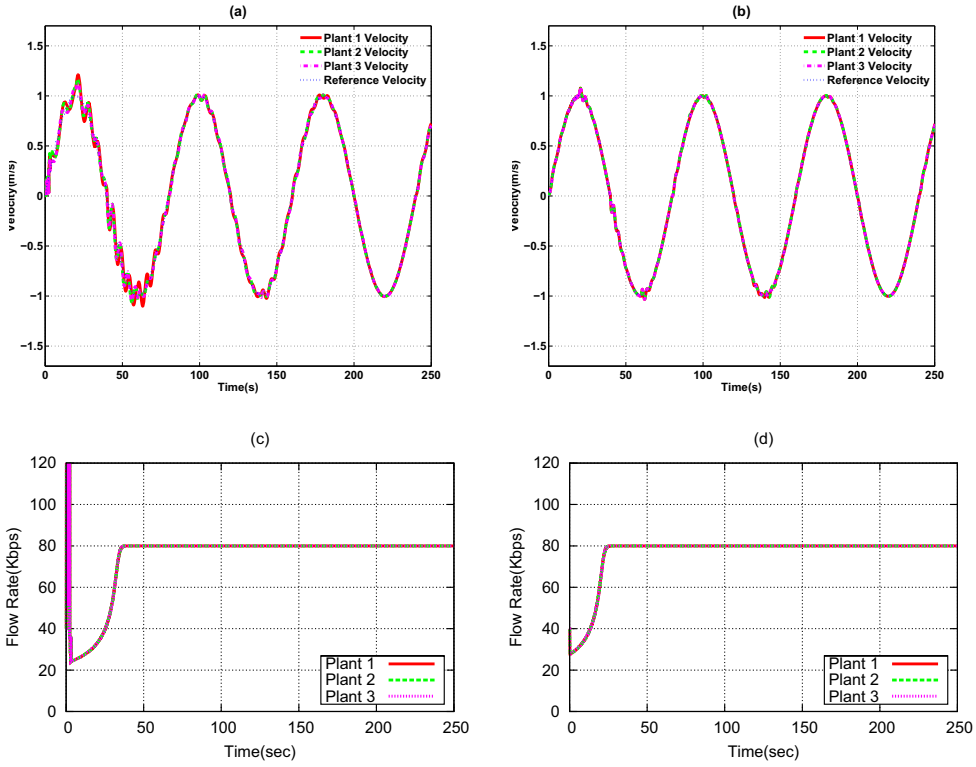


Figure IV.8: Flow Rate Convergence with Different Initial Prices

$10^{-3}$ are set. Independent of the initial prices, the flow rates converge within $40$ seconds. In Fig. V.9 (a) and (c), a smaller price leading to a larger flow rate, network channel gets overflowed, and the price immediately rises. Consequently flow rates are dropped, and later increase gradually and converge. The plant outputs then experience some oscillation due to the sudden rate drop. In Fig. V.9 (b) and (d), with a larger price, both the flow rate and plant output converge much faster. This experiment shows that a larger initial price value is better to preserve a faster system stabilization.

The impact of $\gamma$ is then studied. Fig. IV.9 demonstrates the plant outputs and the flow rate variation of the system with $\gamma$ of $10^{-8}$ on the left and $10^{-10}$ on the right. In Fig. IV.9 (a) and (c), with a larger $\gamma$, the flow rate converges much faster. However, the fast change of the flow rate introduces instability to the control systems, which may cause larger and longer oscillations to the plant outputs. Conversely in Fig. IV.9 (b) and (d), a smaller $\gamma$ leads to a slower convergence speed. The control system may keep experiencing small oscillations during the converging period.

**Impact of Dynamic NCS Join**

It is important that an algorithm is able to dynamically reallocate network resources properly and responsively.

Fig. V.10 (a) and (b) show the plant outputs of the three control systems and Fig. V.10 (c) and (d) show the variation of the flow rates with time at the bottom.

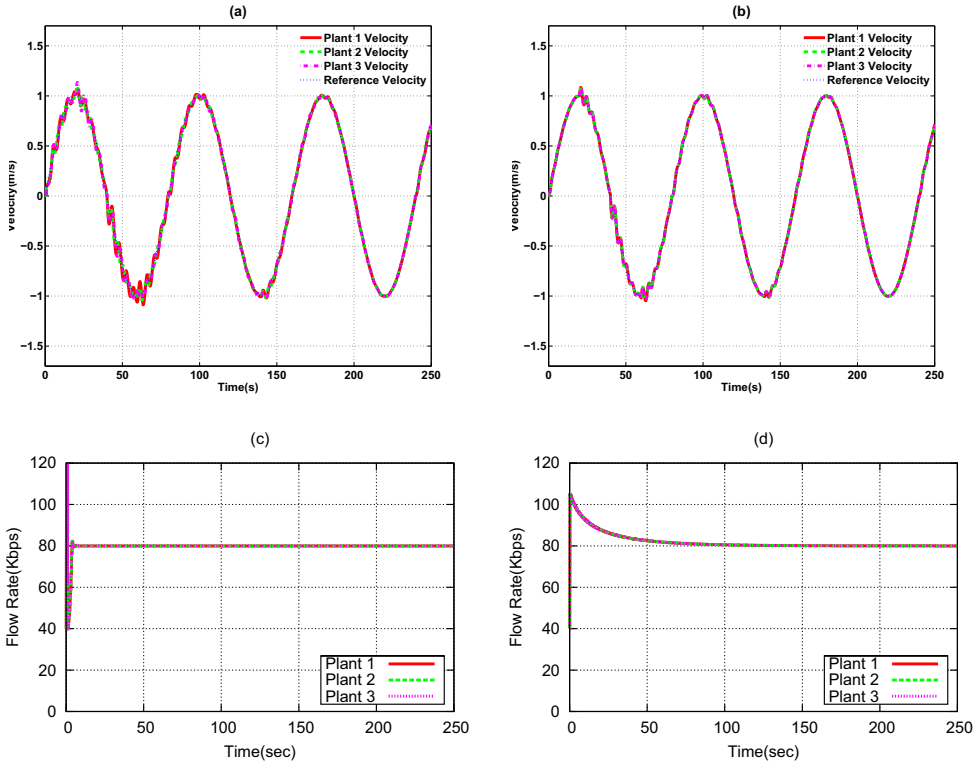


Figure IV.9: Plant Output and Flow Rate Convergence with Different $\gamma$

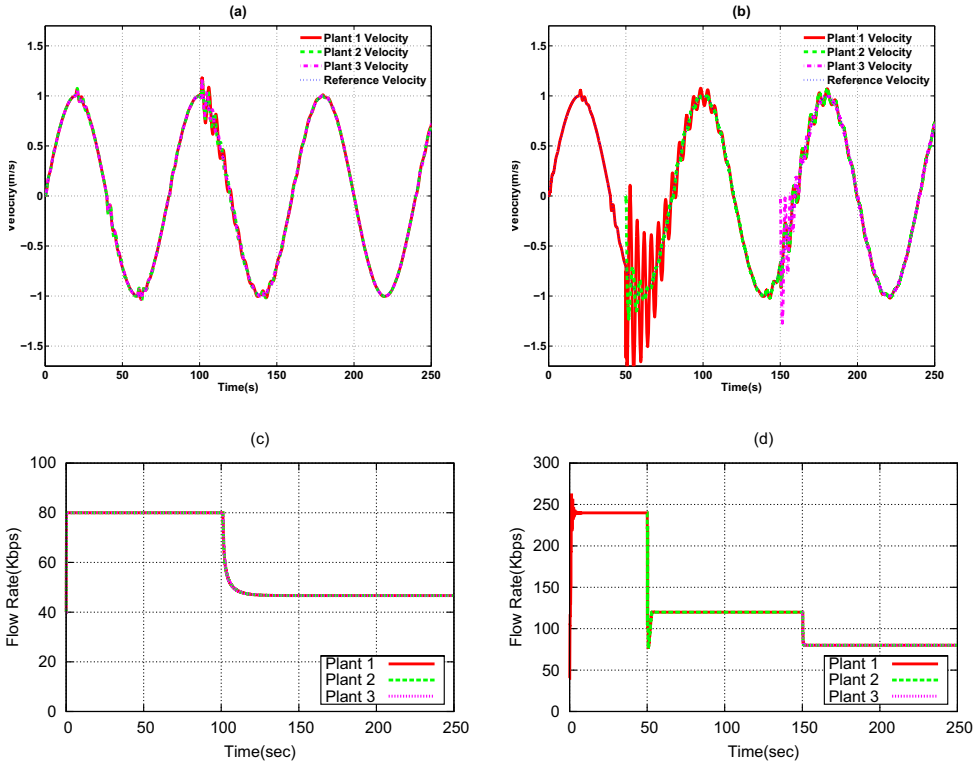


Figure IV.10: Plant Output and Flow Rate with Dynamic Join

In the first experiment, three control systems start at the beginning, Two background Constant Bit Rate (CBR) traffic flows join the channel at $100$ second. The two background traffic both uses fixed rate of $80Kbps$. At the beginning as in Fig. V.10 (a) and (c), the rates quickly converge to $80Kbps$ with the plant outputs stabilize to the sinusoidal reference. When the background traffic joins in, the rates re-converge to around $50Kbps$. The plant outputs experience some oscillation before convergence. This is because that using the old converged rate, the network demand exceeds the channel capacity. During the time of resource reallocation, packets can be dropped from the network queues, and the communication delay becomes relatively large until the law of demand and supply is satisfied again.

In the second experiment, one pair of plant and control system starts first, with the second pair starting at 50 seconds, and the third pair starts at 150 seconds. At the beginning as in Fig. V.10 (b) and (d), the flow rates quickly converge around $240Kbps$ and the plant outputs stabilize at the sinusoidal reference. When the second pair joins in, the flow rates re-converge to around $120Kbps$. The first pair experiences much larger oscillation and takes much longer time to re-converge than the newly joined pair. This is because the first pair has a higher flow rate than the initial flow rate of the new pair, and affected more by the temporary congestion. Similarly, the systems closely follow the reference trajectory after short period of time.

### Multihop Communication

Direct connection in wireless networks requires two nodes within the transmission range of each other. When they are out of range, relay nodes are required to provide relay to route packets.

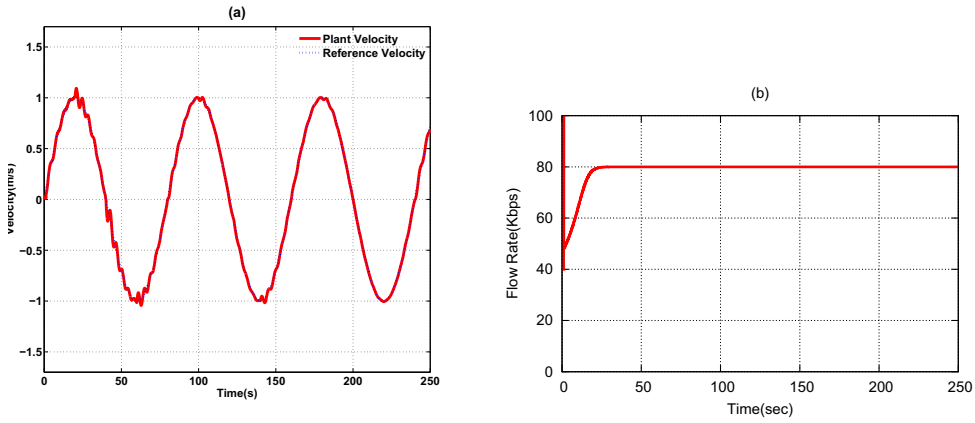


Figure IV.11: Plant Output and Flow Rate in Multihop Scenario

A chain topology is used to test such a multi-hop wireless scenario. Five nodes are formed in a chain structure with a fixed distance of 250m between neighboring nodes.The plant and controller are located at the edges of the network. Fig. IV.11 shows the simulation results. At the beginning of the simulation, the plant output has more oscillation than the case of single hop, as the packets experience higher delay. When the plant output and the flow rate has converged, the plant output performs similar to the single-hop plant systems.

# CHAPTER V

# OPTIMAL CROSS-LAYER DESIGN OF SAMPLING RATE ADAPTATION AND NETWORK SCHEDULING FOR WIRELESS NETWORKED CONTROL SYSTEMS

## Introduction

The integration of physical systems through computing and networking has become a trend, known as Cyber-Physical Systems (CPS). Many real-world CPS such as automotive vehicles and distributed robotics, are monitored and controlled by Networked Control Systems (NCS), where information among sensors, controllers and actuators is exchanged via a communication network. NCS are increasingly deployed over wireless networks, as they provide great convenience in terms of deployment and mobility support [37, 43]. However in a wireless networking environment, the stability and performance of the control system are greatly affected by its limited and dynamic resource availability.

Three major approaches have been investigated in the literature to address the challenges in designing wireless NCS. The first approach, independent of the network protocol design, investigates the design of the control layer with a goal of achieving the desired performance despite of the underlying network uncertainties (*e.g.*, [37, 101]). Alternatively, the network-centric approach focuses on reliable and timely packet deliveries, independent of the control system. Yet without the knowledge and support from the other components of the NCS, these approaches

can hardly achieve both stability and optimal performance simultaneously (*e.g.*, [79, 12]). To ensure the stability and optimize the performance of NCS, co-design of the control system and the networking system has been investigated. Existing work ([11, 71, 87]) either makes simplifying assumptions on the network models or involves too many interactions between the control and the networking systems, which prevents efficient layer abstraction and encapsulation, hindering broader adoption for real-world deployment.

In this chapter, we consider NCS consisting of multiple physical plant and digital controller pairs communicating via a multi-hop wireless network, where the plants follow the reference trajectories provided by the controllers. The performance of the NCS is characterized by the tracking errors of the plants which are introduced from two sources: (1) discretization of the controller and the noise disturbance from the operating environment; (2) packet delay and loss caused by network congestion and dynamics. Both sources of error are related to the sampling rate of the control system. Intuitively, high sampling rates allow frequent state updates and provide NCS with better capability to reduce the effect of environmental disturbances. On the other hand, high sampling rates increase the network load, which increases the possibility of packet loss and delay [64].

We transform the NCS performance objective in terms of tracking error minimization into an optimization problem. The optimization aims at maximizing a utility function that characterizes the relationship between the sampling rate and the capability of disturbance rejection of the control system (i.e., minimizing the discretization-induced tracking error); and the constraints of the sampling

rate come from the wireless network capacity and the requirement of packet (i.e., bounding the network-induced tracking errors). The solution to this optimization problem leads to a cross-layer design of control system sampling rate adaptation and network scheduling, where the sampling rate adaptation determines the bandwidth demands of the network, and the scheduling at the media access control layer resolves the location-dependent interference and determines the available resource capacity of each wireless link.

This sample rate optimization problem, however, is non-trivial to solve. The tight coupling of the sampling rate and the required delay bound of the control system (i.e., the delay needs to be less than the sampling time) poses a nonlinear constraint, which has never been addressed in the existing rate optimization solutions ([2, 21]). To solve this problem, we present a coupled-loop approach. In the inner loop, a relaxed problem, where the delay bound is fixed and independent of the sampling rate, is solved via dual decomposition. In particular, a double-price scheme is employed to regulate the sampling rate traffic demand and the wireless capacity supply. The capacity price regulates the resource usage at the wireless link level, and the delay price regulates the relationship between the achieved packet delay and the required delay bound at the end-to-end flow level. The control system then adapts its sampling rate based on its utility function so that its net profit, which is the difference between the utility and the cost (product of price and rate), is maximized. The outer loop determines the optimal delay bounds progressively based on the converged sampling rate from the inner loop.

The proposed algorithm naturally leads to a distributed cross-layer implementation.

The main contributions of this chapter are summarized as follows. First, we present a new formulation for NCS performance optimization by decoupling its performance metric (tracking error) into two parts – discretization and network effect, which are formulated into the objective and the constraints of an optimization problem respectively. This formulation leads to a cross-layer joint design of sampling rate adaptation and network scheduling which can be easily deployed on existing control systems and networks. We employ a control design approach based on passivity, and we formally prove that the stability and the performance optimality of NCS can be simultaneously achieved. Second, we present a distributed algorithm that solves the NCS performance optimization problem and resolves the complex interdependency between delay and sampling rate. By introducing a novel Virtual Link Capacity Margin (VLCM) parameter that can be adjusted to control the delay and the rate over a wireless link, our solution does not depend on a specific model of packet arrival processes and is suitable for NCS systems with packet arrivals that are not characterized by Poisson processes (which is an assumption usually used in networking delay analysis). Third, our solution is evaluated in an integrated simulation environment that consists of Matlab and ns-2 [99]. Using ns-2 – a packet-level network simulator that implements all the details of the network protocol stack, allows highly accurate evaluation of network effects on the NCS performance, which is impossible by using Matlab/Simulink alone.

The remainder of this chapter is organized as follows. In Sec. V, we present the control system model and the wireless network model. In Sec. V and V, we formulate the problem of optimal rate allocation and derive the double-price-based rate adaptation algorithm. We evaluate the algorithm in different multi-hop scenarios using our Networked Control System Wind Tunnel (NCSWT) simulation tool in Sec. VI.

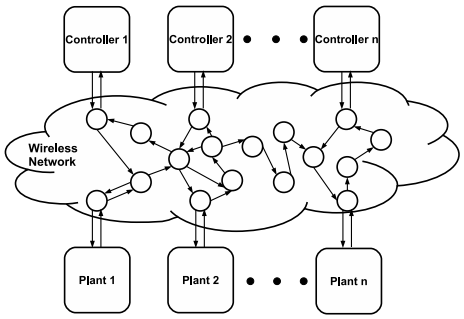


Figure V.1: NCS over multi-hop wireless networks

## Problem Description

We consider NCS consisting of multiple plants and digital controllers communicating via a multi-hop wireless network, as shown in Fig. V.1. The objective of the control system is that the plants follow the reference trajectories provided by the controllers to complete certain tasks. For example, in a manufacturing factory, a group of robotic operators perform the task of moving objects from one place to another. The network controllers receive desired reference trajectory from the

operators and are responsible for ensuring the movement of each robot tracks the desired trajectory.

## Control System Model

A continuous-time plant is described by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) + B_w w(t) \quad \text{(V.1)}$$

$$y_p(t) = C_p x_p(t) \quad \text{(V.2)}$$

where $x_p(t) \in \Re^n$ denotes the plant state, $u_p(t) \in \Re^m$ denotes the control input, $w(t) \in \Re^m$ is the disturbance input, and $y_p(t) \in \Re^m$ is the plant output. $A_p$, $B_p$, and $B_w$ define the plant state matrices and $C_p$ defines the plant output matrix.

The state-space representation of the continuous-time controller is

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) \quad \text{(V.3)}$$

$$y_c(t) = C_c x_c(t) + D_c u_c(t) \quad \text{(V.4)}$$

where $x_c(t) \in \Re^n$ denotes the controller state, and $u_c \in \Re^m$ denotes the error signal, or the difference between the plant output $y_p(t) \in \Re^m$ and the reference signal input $r(t) \in \Re^m$. $A_c$ and $B_c$ define the controller state matrices, while $C_c$ and $D_c$ define the controller output matrices. Let the reference signal denote by

$r(t)$. The tracking error of the system is

$$e_{rr}(t) = r(t) - y_p(t) \tag{V.5}$$

The controller is implemented as a discrete-time control system. We consider sampling instants $t_k \in \mathbb{R}$, $k = 0, 1, \cdots$, with $t_{k+1} > t_k, t_0 = 0$ and we define the sampling interval as $T_k = t_{k+1} - t_k$. In order to simplify the notations, let $x(k+1)$ represent $x(t_{k+1})$, the signal $x(t)$ sampled at time instant $t_{k+1}$.

### Wireless Network Model

We model a multi-hop wireless network as a directed graph $G = (V, L)$, where $V$ is the set of wireless nodes in the network. The nodes communicate with each other via directed wireless links $l \in L$. Such a network supports a set of control systems $H$. For each $h \in H$, the traffic from the controller to the plant and the traffic backwards generate two end-to-end flows denoted as $F(h)$. We collect all end-to-end flows in the network into a set $F$. An end-to-end flow $f$ may go through multiple hops in the network and traverse a sequence of links defined by the routing policy. We use set $L(f)$ to represent all the links along the route of flow $f$ and $F(l)$ to denote all the flows that traverse link $l$.

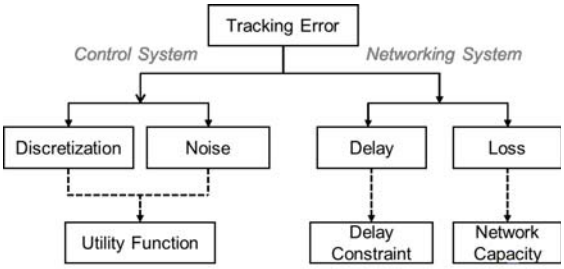


Figure V.2: Decompose tracking error based on its source

NCS Performance Optimization

The NCS performance can be characterized by the tracking error of the control systems. The main focus of this chapter is to minimize the tracking error of the NCS deployed over the multi-hop wireless network while maintaining certain level of fairness among the plant-controller pairs. As shown in Fig. V.2, there are two main sources of error. When a continuous-time control system is discretized, its response to environmental disturbances degrades compared to the response of the idealized continuous system. The level of the degradation depends on the sampling rate, which determines how well the digital controller approximates the continuous controller. High sampling rate allows frequent state updates and thus provides better capability to reduce the effect of environmental disturbances and minimize the tracking error. Packet loss and delay also deteriorate the tracking error. We focus on the congestion-induced packet loss and delay. Network congestion appears when the traffic demand overwhelms the capacity supply. While the sampling rate determines the network traffic demand, the network resource

management mechanisms such as media access control scheduling allocate appropriate capacity to each wireless link.

Optimizing the NCS performance requires the coordination between the control system and the networking system. The control system needs to have the capability to adapt its sampling rate based on the resource utilization information from the network. The networking system should schedule its wireless transmission to meet the resource needs from the control system. This chapter studies how to minimize the NCS tracking error via joint sampling rate adaptation and networking scheduling. Note that this chapter assumes fixed network routing, which is known a priori.

## Optimization Framework For Tracking Error Minimization

In this section, we present the control system design and formulate the problem of NCS tracking error minimization as a sampling rate optimization problem. We first show that our passivity-based control system design is able to ensure system stability with time-varying sampling time. Then we define the optimization objective through a utility function which characterizes the relationship between the sampling rate and the capability of disturbance rejection of the control system (*i.e.*, minimizing the discretization-induced tracking errors). The optimization constraints are based on the wireless network schedulability and the NCS delay requirement.

Passivity-based control system – ensuring system stability with time-varying sampling time

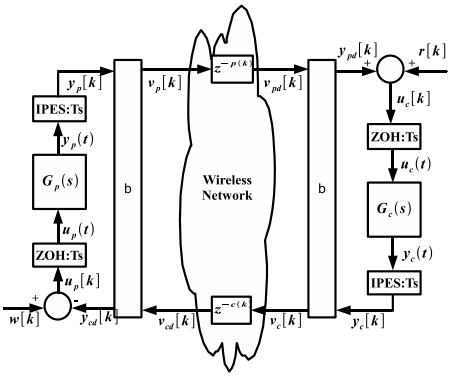


Figure V.3: Passivity Based Control Architecture Over Wireless Networks

Fig. V.3 shows the passivity-based control system architecture. A passive system is defined as a system with bounded output energy such that the system does not produce more energy than what is initially stored. We assume the plant system is passive. A large class of systems can be "passified" by adding local control and filter components [49][55]. The controller $G_c(s)$ is designed so that the plant tracks the reference $r(k)$ and is also assumed to be passive. The control architecture uses (1) a discretization approach defined by the Inner Product Equivalent Sampling and Hold (IPESH) transform, which is composed by the Inner Product Equivalent Sampling (IPES) and Zero Order Hold (ZOH) blocks and (2) a bilinear transform $b$ for converting the control signals into wave variables for communication over a wireless network [55, 56]. These transformations ensure that the NCS

are passive and stable in the presence of time-varying delay and packet loss.[1] Next, we show that the NCS are ensured stable with time varying sampling time, which allows us to use sampling rate adaptation.

A passive continuous-time linear time invariant (LTI) system can be converted to a discrete-time passive system at a varying sampling time, $T_k$, with the discrete-time state space equations described as

$$x(k+1) = \Phi_k x(k) + \Gamma_k u(k) \tag{V.6}$$

$$y(k) = C_{dk} x(k) + D_{dk} u(k) \tag{V.7}$$

In [55][110], it is shown that in order to obtain a passive discrete-time equivalent of a LTI passive continuous-time system for a given fixed sampling time $T_k$, the IPESH is used to compute the system coefficients, $\Phi_k$, $\Gamma_k$, $C_{dk}$ and $D_{dk}$ to preserve passivity.

Discretization with time-varying sampling time can be performed by applying the IPESH for each resulting sampling time, $T_k$, hence ensuring passivity of the discretization at each sampling time and thus the overall passivity of the discrete-time system for a given time interval. This implies that the new system coefficients are redefined as $\Phi_k = \Phi(T_k)$, $\Gamma_k = \Gamma(T_k)$, $C_{dk} = C_d(T_k)$ and $D_{dk} = D_d(T_k)$. By ensuring the passivity of the discrete-time system, the stability is also ensured.

[1] We refer readers to [55, 56] for a detailed description and proofs.

Utility function - modeling error from discretization

To characterize the impact of the sampling rate on the tracking error, we first introduce a utility function which characterizes the disturbance rejection capability of the discrete-time system compared with its continuous-time counterpart.

**Continuous-time control system**

The covariance matrix of the zero-mean white noise process of the continuous-time system can be defined as

$$E[w(t)w^T(t+\tau)] = Q\delta(\tau) \tag{V.8}$$

where $E$ denotes the expected value and $Q$ represents the power spectral density of $w$, or the continuous-time noise covariance matrix. The power spectral density can also be referred to as the "white noise intensity" or mean-square spectral density. The continuous-time state covariance matrix $P_c$ can be described by

$$P_c(t) = E[x(t)x^T(t)] \tag{V.9}$$

Based on the knowledge of $Q$, the steady state value of the state covariance can be obtained by [60]

$$A_{cl}P_c + P_cA_{cl} + B_{wcl}QB_{wcl}^T = 0 \tag{V.10}$$

where the matrices $A_{cl}$ and $B_{wcl}$ denote the closed loop matrices of the continuous-time system, or the coefficients of $x(t)$ and $w(t)$ respectively. From the resulting state covariance matrix, the root mean square of a state can then be determined. The Root-Mean-Square (RMS) of the plant states is equivalent to the standard deviation of one of the plant states. For example, if a system has only one plant state variable $x_p$, and its plant state covariance is $v(x_p)$, the RMS of the plant state is equal to $\sqrt{v(x_p)}$. When a system has several plant state variables, we can use the plant state covariance from one of them to calculate the RMS of all the plant states.

**Discrete-time control system**

Based on the knowledge of the continuous-time noise covariance matrix $Q$, the discrete-time noise covariance matrix $Q_d$ can be obtained using the Van Loan's algorithm [30] and can be defined as

$$Q_d = \int_0^{T_f} \Phi(\tau) B_{wcl} Q B_{wcl}^T \Phi^T(\tau) d\tau \tag{V.11}$$

where $\Phi$ is the closed loop matrix, or the discrete-time state coefficient of $x(k)$, and $B_{wcl}$ denote the closed loop matrix of the continuous-time system, or the coefficient of $x(t)$.

The steady state discrete-time state covariance matrix can then be obtained from the following equation

$$\Phi P_d \Phi^T + Q_d = P_d \tag{V.12}$$

From the resulting state covariance matrix, the discrete RMS of the plant state can then be determined in a similar way as the continuous-time case.

**Utility function formulation**

We now define the utility function of a control system as a function of its sampling rate $1/T_k$ using the ratio of RMS between the discrete-time system with its continuous-time counterpart. Thus, the utility function reflects the degradation amount of the system response to the white noise compared to the continuous closed loop system.

$$U(1/T_k) = \frac{RMS_{\texttt{continuous}}}{RMS_{\texttt{discrete}}(T_k)} \tag{V.13}$$

To demonstrate the definition of our utility function, we consider a single-input-single-output (SISO) LTI system without loss of generality. As shown in Fig. V.4, its utility function is a strictly concave function of the sampling rate. The concavity of the utility function reflects the marginal return on the control performance when its sampling rate increases.

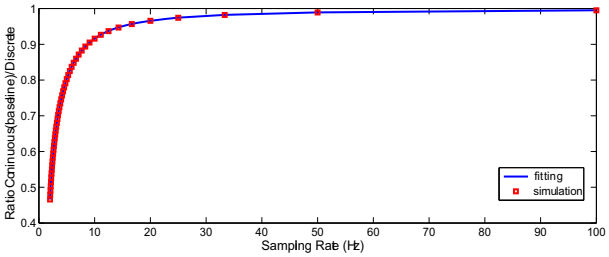


Figure V.4: Example utility function for the control system where the transfer function of the plant is $G_p(s) = \frac{1}{Js}$, transfer function of the controller is $G_c(s) = \frac{K_p + K_d s}{s}$, with $J = 2.93$, $K_d = 32.1$ and $K_p = 8.2$.

**Relationship between utility function and tracking error**

In a closed-loop continuous-time system, the system response of the plant can be described as

$$\begin{aligned} x_p(t) &= e^{A_{cl}t}x_p(0) + e^{A_{cl}t}\int_0^t e^{-A_{cl}\tau}B_{cl}r(\tau)d\tau \\ &\quad + e^{A_{cl}t}\int_0^t e^{-A_{cl}\tau}B_{wcl}w(\tau)d\tau \end{aligned} \tag{V.14}$$

$$\begin{aligned} y_p(t) &= C_{cl}e^{A_{cl}t}x_p(0) + C_{cl}e^{A_{cl}t}\int_0^t e^{-A_{cl}\tau}B_{cl}r(\tau)d\tau \\ &\quad - C_{cl}e^{A_{cl}t}\int_0^t e^{-A_{cl}\tau}B_{wcl}w(\tau)d\tau \end{aligned} \tag{V.15}$$

Recall that the tracking error of the system $e_{rr}(t) = r(t) - y_p(t)$. From (V.15), the output response of the plant has two main components that contribute towards the tracking error. The first component is the plant response to the reference input $r(t)$, and the other is the plant response to the disturbance input $w(t)$. The

passive controller is designed to ensure the plant's response to the reference input minimizes the tracking error. The system achieves a certain level of disturbance rejection. The contribution of the input disturbance can be characterized by the covariance of the tracking error.

From (V.15) and the fact that $r(t)$ is not stochastic, we have $E[r(t)y^T(t)] = E[r^T(t)y(t)] = E[r(t)r^T(t)] = 0$. The covariance of the tracking error can be described by

$$C_e(t) = E[e(t)e^T(t)] = E[y(t)y^T(t)] \tag{V.16}$$

This essentially implies that the covariance of the error is equal to the output covariance. Based on the knowledge of $Q$, the steady state value of the output covariance is [60]

$$C_e = C_{cl}P_cC_{cl}^T \tag{V.17}$$

Capacity and delay constraints – bounding error from network

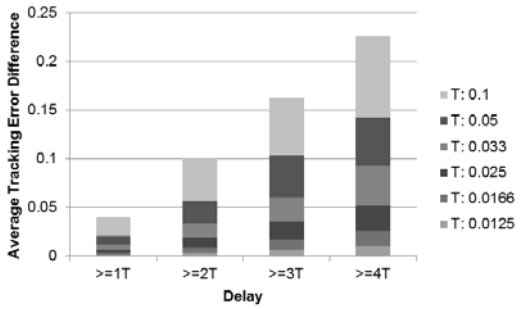


Figure V.5: Impact of Delay On the NCS Average Tracking Error

**Capacity constraint**

To limit the effect of packet loss caused by network congestion on the tracking error, we need to restrain the network load within its capacity. Wireless network communication is subject to location dependent interference. Thus the achievable capacity of each wireless link is related to the scheduling algorithm. We adopt the conflict graph concept to model wireless interference [46]. Each vertex in the conflict graph represents a wireless link of the original network and there is an edge between two vertices if their corresponding wireless links interfere with each other. The communications along wireless links are scheduled on a slotted time basis. In each time slot, one independent set[2] $I$ of the conflict graph is selected and only the links corresponding to the vertices in $I$ are allowed to transmit because they are interference free. Let $c_l$ be the channel capacity. A $L$-dimension column vector $r^I$ is used to represent the capacity vector of $I$, where $r_l^I = c_l$ if $l \in I$, and $r_l^I = 0$ otherwise. We adopt the concept of feasible capacity region $\Lambda$ to model the feasible link capacity allocation [16]. The feasible capacity region is a convex hull, which is defined as $\Lambda := \sum_I \alpha_I r^I$, where $\sum_I \alpha_I = 1$ and $\alpha_I \geq 0$. *Scheduling* essentially determines the capacity allocation $\hat{\boldsymbol{c}} = (\hat{c}_l, l \in L)$ of the links, where $\hat{c}_l$ is the average capacity over time based on the scheduling. Obviously, $\hat{\boldsymbol{c}} \in \Lambda$. To limit the packet congestion loss, the aggregated traffic load on any wireless link $l \in L$ should be no more than its achievable capacity $\hat{c}_l$.

[2]The independent set of a graph is a set of vertices within which no edge exists between any two vertices.

**Delay effect on tracking error**

To determine the effect of delay on the tracking error, we perform a set of simulation studies using NCSWT [99] over NCS with one pair of plant and controller. Based on the assumption that the discrete plant/controller systems update and process data received only at sampling instants, the delay viewed from the control systems' perspective are integral multiples of the sampling interval. We vary the sampling time and manually introduce delay which are integral multiples of the sampling time. Then we evaluate the average tracking error difference, which is the difference between the time-averaged tracking error with delay introduced and that without any delay.

From the experiment, we observe that when the delay is within one sampling time, the tracking error difference remains zero. Fig. V.5 shows the effect of delay on the tracking error difference when it is larger than the sampling time. We observe that the error increases superlinearly when the delay increases beyond one sampling time. Based on the observations, we bound the average end-to-end delay of control system flows to their system sampling time.

**Controlling delay with $VLCM$**

Providing delay assurance is notoriously difficult in wireless networks. The main difficulty comes from the complex interactions between traffic arrival and departure, which is shaped by the network scheduling. Most of the existing works on delay analysis make explicit assumptions on the packet arrival process (*e.g.*,

Poisson arrivals) [35], which do not reflect the NCS traffic characteristics. Here we employ a general method which is not limited to a predefined packet arrival process. In order to regulate the maximum allowable rate $m_l$, we introduce a parameter *Virtual Link Capacity Margin* $(VLCM)$ $\sigma_l$ of link $l$ defined by

$$\sigma_l = \hat{c}_l - m_l, \text{ with } m_l < \hat{c}_l, \forall l \in L \tag{V.18}$$

We regard the link delay (*i.e.*, average packet delay along the link) as a function of the $VLCM$ $\varphi(\sigma_l)$. Then the average delay of flow $f$ is the sum of all link delay along its route.

### Optimization Framework

Recall that each control system is associated with two flows. Let $z_h = \frac{sample\_size}{T_h}$ be the traffic rate of one flow for the control system $h$, where $sample\_size$ is the size of the sample and $T_h$ is its sampling time. $T_{h(f)}$ is the sampling time of control system $h$ which flow $f$ is associated with. Thus, the maximum allowable rate satisfies $m_l \geq \sum_{h \in H: f \in F(h) \& f \in F(l)} z_h$. We overload $U_h$ as a function of traffic rate for control system $h$, as defined by Eq. (V.13). Now we formulate the optimal sampling rate allocation problem as follows:

$$W: \quad \max \quad \sum_{h \in H} U_h(z_h) \tag{V.19}$$

$$\text{s.t.} \quad \sum_{h \in H: f \in F(h) \cap F(l)} z_h \leq \hat{c}_l - \sigma_l, \forall l \in L \tag{V.20}$$

$$\sum_{l \in L(f)} \varphi(\sigma_l) \leq T_{h(f)}, \forall f \in F \tag{V.21}$$

$$\text{over} \quad \hat{\boldsymbol{c}} \in \Lambda \tag{V.22}$$

The objective of the nonlinear problem is to maximize the aggregate utility of all control systems in the network. This objective minimizes the tracking error induced by discretization and maintains certain fairness among all the plant-controller pairs [50]. Inequality (V.20) represents the wireless capacity constraint for each wireless link. Note that the $VLCM$ $\sigma_l$ is introduced here to control the link delay. Inequality (V.22) defines the scheduling feasibility. Inequality (V.21) is the flow delay constraint where the average flow delay is bounded by the sampling time of its control system. It is important to note that there is a possibility that the optimal solution of sampling time that minimizes the tracking error may fall below the delay bound. We choose to incorporate this delay bound (V.21) in our problem formulation for two reasons. First, from Fig. V.5, we observe that the tracking error increases super-linearly with respect to delay when the delay goes beyond the sampling time; while the utility only increases sub-linearly with respect to sampling rate. Intuitively, this implies the marginal benefit of increasing the sampling rate is overweighed by the marginal penalty of pushing the delay beyond the sampling time. Based on this intuition, we bound the average delay by the sampling time. On the other hand, without this delay bound constraint, providing a formulation that fully captures the complex interaction among sampling time/rate, delay, delay-introduced error, and discretization-introduced error

will lead to an intractable optimization problem, where identifying a distributed solution is even harder.

## Distributed Cross-Layer Algorithm

### Solution overview

Problem $W$ is non-trivial due to the complicated interactions between the $VLCM$s, the sampling rate and the end-to-end delay. The tight coupling of the sampling rate and the required delay bound of the control system (i.e., the delay needs to be less than the sampling time) poses a nonlinear constraint, which has never been addressed in the existing rate optimization solutions ([2, 21, 72]). To solve this problem, we first relax the delay constraint and consider the optimization problem with a fixed delay requirement. Then we show how to adjust the delay requirement to achieve the optimal solution of the original problem $W$.

### Cross-layer algorithm with fixed delay

The optimization framework with a fixed delay requirement can be written as

$$W1: \quad \max \quad \sum_{h \in H} U_h(z_h) \tag{V.23}$$

$$\text{s.t.} \quad \sum_{h \in H: f \in F(h) \cap F(l)} z_h \leq \hat{c}_l - \sigma_l, \forall l \in L \tag{V.24}$$

$$\sum_{l \in L(f)} \varphi(\sigma_l) \leq D_{h(f)}, \forall f \in F \tag{V.25}$$

$$\text{over} \quad \hat{\boldsymbol{c}} \in \Lambda \tag{V.26}$$

where the constraint (V.21) is replaced by (V.25), in which $D_{h(f)}$ is the delay requirement of control system $h$.

**Double-Price Algorithm**

Direct solution to $W1$ requires global coordination of all network components, such as flows and links, which is computationally expensive. We consider its dual decomposition. Let $\boldsymbol{\nu} = \{\nu_l, l \in L\}$ and $\boldsymbol{\mu} = \{\mu_f, f \in F\}$ be the Lagrange multipliers with respect to constraints (V.24) and (V.25) respectively. The Lagrangian of $W1$ is:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{z}, \boldsymbol{\nu}, \boldsymbol{\sigma}, \boldsymbol{\mu}, \hat{\boldsymbol{c}}) \\
&= \sum_{h \in H} U_h(z_h) - \sum_{l \in L} \left( \nu_l \sigma_l + \sum_{f \in F(l)} \varphi(\sigma_l) \mu_f \right) \\
&\quad - \sum_{h \in H} \left( z_h \sum_{l \in L(f) \& f \in F(h)} \nu_l \right) + \sum_{f \in F} \mu_f D_{h(f)} + \sum_{l \in L} \nu_l \hat{c}_l
\end{aligned}$$

The dual of $W1$ is

$$\bar{D}(\boldsymbol{\nu}, \boldsymbol{\mu}) = \min_{\boldsymbol{\nu} \geq 0, \boldsymbol{\mu} \geq 0} D(\boldsymbol{\nu}, \boldsymbol{\mu}) \tag{V.27}$$

where

$$
\begin{aligned}
& D(\boldsymbol{\nu}, \boldsymbol{\mu}) \qquad\qquad (V.28)\\
= \; & \max_{\boldsymbol{z},\boldsymbol{\sigma},\hat{\boldsymbol{c}}} L(\boldsymbol{z}, \boldsymbol{\nu}, \boldsymbol{\sigma}, \boldsymbol{\mu}, \hat{\boldsymbol{c}}) \\
= \; & \max_{\boldsymbol{\sigma}} \left\{ -\sum_{l \in L} \left( \nu_l \sigma_l + \sum_{f \in F(l)} \varphi(\sigma_l) \mu_f \right) \right\} \\
& + \max_{\boldsymbol{z}} \left\{ \sum_{h \in H} \left( U_h(z_h) - z_h \sum_{l \in L(f) \& f \in F(h)} \nu_l \right) \right\} \\
& + \max_{\hat{\boldsymbol{c}}} \left\{ \sum_{l \in L} \nu_l \hat{c}_l \right\} + \sum_{f \in F} \mu_f D_{h(f)}
\end{aligned}
$$

The solution $(\boldsymbol{z}^*, \boldsymbol{\sigma}^*, \hat{\boldsymbol{c}}^*)$ to (V.28) should satisfy:

$$
z_h^* = \arg\max_{z_h} \left\{ \sum_{h \in H} \left( U_h(z_h) - z_h \sum_{l \in L(f) \& f \in F(h)} \nu_l \right) \right\} \qquad (V.29)
$$

$$
\sigma_l^* = \arg\max_{\sigma_l} \left\{ -\sum_{l \in L} \left( \nu_l \sigma_l + \sum_{f \in F(l)} \varphi(\sigma_l) \mu_f \right) \right\} \qquad (V.30)
$$

$$
\hat{c}_l^* = \arg\max_{\hat{c}_l \in \Lambda} \left( \sum_{l \in L} \nu_l \hat{c}_l \right) \qquad (V.31)
$$

Here the multiplier $\nu_l$ can be seen as the implicit congestion price [72] of link $l$, which represents the cost of delivering a unit of data through link $l$. The multiplier $\mu_f$ can be interpreted as the implicit delay price of flow $f$, which represents the cost of imposing a unit of delay on flow $f$. If $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$ are given, we can

obtain the maximizers $z_h^*$ and $\sigma_l^*$ by taking the derivative with respect to $z_h$ and $\sigma_l$ respectively.

$$z_h^*(\kappa_h) = U_h^{'-1}(\kappa_h).\ \kappa_h = \sum_{l \in L(f) \& f \in F(h)} \nu_l, \forall h \in H \tag{V.32}$$

$$\sigma_l^*(\lambda_l, \nu_l) = \varphi_l^{'-1}(\frac{-\nu_l}{\lambda_l}).\ \lambda_l = \sum_{f \in F(l)} \mu_f, \forall l \in L \tag{V.33}$$

(V.32) implies that the optimal sampling rate of a control system $h$ is determined by its price $\kappa_h$, which is the aggregated price of the links along its flow routes. (V.33) implies that the optimal $VLCM$ of a link is relevant to its congestion price $\nu_l$ and link margin price $\lambda_l$. The intuition is: 1) the congestion price determines the available capacity margin that can be used for $VLCM$ adjustment; and 2) the link margin price implicitly reflects the overall delay requirement (from all of its supporting flow delay requirement) on its $VLCM$. The maximizer $\hat{c}_l^*$ can be generated from a maximum weight based scheduling policy.

Now $W1$ is converted into three sub-problems: the sampling rate adaptation problem (V.29), the *VLCM* assignment problem (V.30) and the scheduling problem (V.31). The link congestion price $\boldsymbol{\nu}$ and the flow delay price $\boldsymbol{\mu}$ can be computed iteratively, from the opposite direction to the gradient $\nabla(L(\boldsymbol{\nu}, \boldsymbol{\mu}))$ [9].

This adaptation approach is called *double-price* scheme. Based on the information of two price signals, the algorithm iteratively reaches a global optimum. The property of this algorithm is formally characterized in Proposition 1 and Proposition 2.

**Proposition 1** There is no duality gap between (V.23) and (V.27). For any $(\boldsymbol{\nu}^*, \boldsymbol{\mu}^*)$ that minimizes (V.28), if $(\boldsymbol{z}^*, \boldsymbol{\sigma}^*, \hat{\boldsymbol{c}}^*)$ solves (V.29), then $(\boldsymbol{z}^*, \boldsymbol{\sigma}^*, \hat{\boldsymbol{c}}^*)$ is the unique maximizer of (V.19).

**Proposition 2** If $||\boldsymbol{\beta}||_2$ and $||\boldsymbol{\gamma}||_2$ are sufficiently small, starting from any initial values $\boldsymbol{z}(0)$, $\boldsymbol{\sigma}(0)$, $\hat{\boldsymbol{c}}(0)$ and prices $\boldsymbol{\nu}(0) \geq 0$, $\boldsymbol{\mu}(0) \geq 0$, the cross-layer algorithm converges to the optimal solution $(\boldsymbol{z}^*, \boldsymbol{\sigma}^*, \hat{\boldsymbol{c}}^*, \boldsymbol{\nu}^*, \boldsymbol{\mu}^*)$. The proof of these two propositions are provided in [92].

### Cross-Layer Rate Allocation Implementation

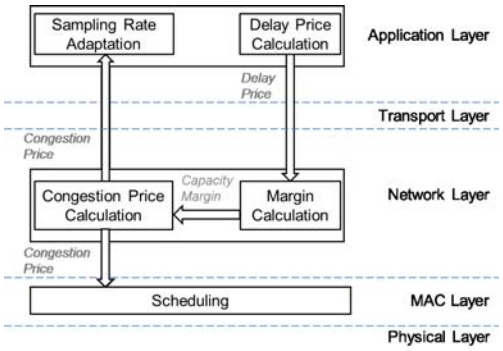


Figure V.6: NCS over multi-hop wireless networks

Our algorithm naturally leads to a cross-layer implementation via joint *VLCM* assignment, sampling rate adaptation and scheduling, as shown in Fig. V.6. Scheduling is performed at the MAC layer. At the network layer, the margin calculation generates the optimal *VLCM*s for a wireless interface queue; the congestion price

calculation provides per-hop congestion price, which reflects the level of congestion at this queue. They can be implemented as part of the queue management mechanism. At the application layer, the per-hop congestion price is aggregated to calculate the sampling rate; the end-to-end delay is measured to calculate the delay price.

Our algorithm implementation only requires the knowledge of the first order derivative of the link delay with respect to the capacity margin $\frac{\partial\varphi(\sigma_l)}{\sigma_l}$ based on (V.33), rather than some statistical characteristics, such as the mean or the variance of the packet arrival rate. The derivative of link delay can be profiled online. According to (V.31), we need to find a scheduling policy so that the aggregate link weight $\sum_{l\in L}\nu_l\hat{c}_l$ could be maximized. We achieve this by using a maximum matching based scheduling policy [72].

### Delay Bound Tuning

After obtaining the optimal sampling rate solution to the problem $W1$ with fixed delay requirement, we now solve the original optimal problem $W$ by determining the optimal delay requirements for all NCS. We proceed in two steps. First we determine the ranges of the delay requirements. Then, we adjust the delay requirements to find the ones which yield the optimal sampling rate allocation within the range.

**Range of delay requirement determination**

The lower bound $\underline{\boldsymbol{D}} = (\underline{D}_h, h \in H)$ of the delay range can be computed via the optimization problem of

$$\underline{W}: \quad \max \quad \sum_{h \in H} U_h(z_h) \tag{V.34}$$

$$\text{s.t.} \quad \sum_{h \in H: f \in F(l) \& f \in F(h)} z_h \leq \hat{c}_l, \forall l \in L \tag{V.35}$$

$$\text{over} \quad \hat{\boldsymbol{c}} \in \Lambda \tag{V.36}$$

This is a simplified form of $\underline{W}$, with the $VLCM$ $\sigma_l = 0$ for all $l \in L$ and without the delay constraints. The solution to this problem $\underline{\boldsymbol{z}}$ is the maximum achievable sampling rate considering only the network capacity constraint. This maximum achievable rate corresponds to the minimum sampling time of the NCS $T_h(\underline{z}_h)$. As our delay constraint in the original problem $W$ is that the flow delay should not exceed one sampling time, we can treat the minimum sampling time as the lower delay bound $\underline{\boldsymbol{D}} = (T_h(\underline{z}_h), h \in H)$.

Fixing the sampling rate to $\underline{\boldsymbol{z}}$, we allow the maximum amount of traffic satisfying only the network capacity constraint to be injected into the network. Thus the measured delay $\bar{\boldsymbol{d}} = (\bar{d}_f, f \in F)$ is the upper bound of the end-to-end delay. If $\bar{d}_f \leq T_h(\underline{z}_h), \forall f \in F(h), \forall h \in H$, then $\underline{\boldsymbol{z}}$ will also be the optimal sampling rate for the original problem $W$. If there exists $\bar{d}_f > T_h(\underline{z}_h)$, then we set the upper bound of the delay requirement to $\bar{\boldsymbol{D}} = \bar{\boldsymbol{d}}$.

**Optimal delay requirement adjustment**

Starting from the lower bound of the delay requirement, we adjust the delay requirement of each control system based on the algorithm shown in Table V.1. In the algorithm, we gradually adjust the delay requirements of all the systems from their lower bounds until 1) they are all smaller than the corresponding optimal sampling times based on problem $W1$; and 2) at least for one system, the difference between its delay requirement and the optimal sampling time is within the constant bound $\epsilon$. The initial adjustment size $a_h, h \in H$ is set to half of the difference between $\underline{D}_h$ and $\bar{D}_h$, and becomes half of its previous value on each round of iteration.

Table V.1: Delay requirement adjustment

| **Adjustment of Delay Requirement $\boldsymbol{D}$** |
|---|
| 0) *initialization*<br>let $\boldsymbol{D} = \underline{\boldsymbol{D}}$, $\boldsymbol{a} = (\bar{\boldsymbol{D}} - \underline{\boldsymbol{D}})/2$ be the initial adjustment sizes,<br>$\epsilon$ be a sufficiently small constant, $\boldsymbol{\epsilon}$ be a vector of $\epsilon$;<br>1) compute $\boldsymbol{z}$ by solving $W1$ where the delay requirements are $\boldsymbol{D}$;<br>derive the corresponding sampling time $\boldsymbol{T}(\boldsymbol{z})$;<br>If $\boldsymbol{D} \leq \boldsymbol{T}(\boldsymbol{z})$ and $\exists h, T_h(z_h) - D_h \leq \epsilon$, stop;<br>2) If $\boldsymbol{D} < \boldsymbol{T}(\boldsymbol{z}) - \boldsymbol{\epsilon}$, increase its delay requirement:<br>$\boldsymbol{D} = \boldsymbol{D} + \boldsymbol{a}$<br>3) If $\exists h, D_h > T_h(z_h)$, decrease the delay requirement:<br>$\boldsymbol{D} = \boldsymbol{D} - \boldsymbol{a}$<br>and reduce the adjustment size:<br>$\boldsymbol{a} = \boldsymbol{a}/2$<br>repeat 1) to 3) |

**Theorem 1** The adjustment algorithm in Table V.1 converges within in $\lceil \log_2 a_h/\epsilon \rceil$ number of iteration.

**Proof**: We start with a simple scenario with one pair of plant and controller in a single-hop topology.

We first show the existence of the optimal delay requirement in the simple scenario. In this scenario, $d_f$ is a strictly decreasing function of $T_h$. In the cross-layer algorithm with fixed delay, given a reachable delay requirement $D_h$, the algorithm can drive the average end-to-end delay $d_f$ of a flow close to the delay requirement: $0 \leq D_h - d_f \leq \epsilon$. Thus, we can consider $D_h$ as a decreasing function of $T_h$. From the definition of the lower and upper bound of the delay requirements $\underline{D}_h$ and $\bar{D}_h$, we have

$$\begin{aligned} \bar{D}_h &> T_h(\underline{z}_h) \\ \underline{D}_h &< T_h(\bar{z}_h) \end{aligned}$$

Based on the fixed point theorem, there exists

$$0 \leq \tilde{D}_h - T_h(\tilde{z}_h) \leq \epsilon$$

According to the binary search algorithm, the maximum number of iteration will be $\lceil \log_2 a_h/\epsilon \rceil$. This proves the convergence of the optimal delay requirement algorithm in the simple scenario. □

Next we will prove the convergence of the delay requirement adjustment algorithm in the general cases.

**Lemma 1** The maximum number of iteration in the delay requirement adjustment algorithm is $\lceil \log_2 \max(a_h)/\epsilon \rceil$.

**Proof**: During each iteration, the binary search algorithm reduces the search space for the optimal delay requirement by half. Moreover, according to condition 3) of our algorithm, as long as the delay requirement of any system is greater than its corresponding sampling time, the delay requirements of each system $h$ will decrease $a_h$. Thus, the ranges of the delay requirements always satisfy one of the following conditions:

- The lower bound of the delay requirement is smaller than the corresponding sampling time, and the upper bound of the delay requirement is larger than the corresponding sampling time.

- Both the lower bound and the upper bound of the delay requirements are smaller than the corresponding sampling times.

As the binary search algorithm reduces the search space by half during each iteration, after $\lceil \log_2 \max(\boldsymbol{a})/\epsilon \rceil$ number of iterations, the search space of system $h$ becomes

$$\frac{(\bar{D}_h - \underline{D}_h)}{2^{\lceil log_2^{\max(\bar{\boldsymbol{D}} - \underline{\boldsymbol{D}})/\epsilon} \rceil}} \leq \epsilon \cdot \frac{\bar{D}_h - \underline{D}_h}{\max(\bar{\boldsymbol{D}} - \underline{\boldsymbol{D}})} \leq \epsilon$$

Thus, the algorithm converges after at most $\lceil \log_2 \max(\boldsymbol{a})/\epsilon \rceil$ number of iterations. Note that when all the systems have same utility function and network topology in terms of route and inteference, our algorithm converges after $\log_2 \boldsymbol{a}/\epsilon$ number of iterations, and leads to the optimal solution. □

In general cases, when the routes of different systems overlap or interfere with each other, $D_h$ not only is a function of $T_h$, but is also affected by sampling time of other systems. This complicates the search for the optimal solutions. Our algorithm simplifies the problem by providing a feasible but sub-optimal solution.

## Performance Evaluation

In this section, we evaluate our cross-layer sampling rate adaptation and network scheduling algorithm using an integrated simulation tool named Networked Control System Wind-Tunnel (NCSWT) [99]. NCSWT integrates two simulators Matlab and ns-2, which allows us to simulate the control system models in Matlab/Simlink and the networking systems in ns-2. Using ns-2, a packet-level network simulator that implements all the details of the network protocol stack, we can perform highly accurate evaluation of the network effects on the NCS performance, including queueing delay and network scheduling, which is impossible by using Matlab/Simulink alone.

## Simulation Setup

In our experiments, the NCS consists of three pairs of plants and controllers. Each of the three plant systems used in the experiments is the model of a single joint of a robotic arm. They are described by the continuous time state space representation as defined in (V.1) and (V.2), with the parameters $A_p = 0, B_p = 1, C_p = 0.3413$. Each of the controllers is described as in (V.3) and (V.4) with $A_c = 0, B_c = 1, C_c = 32.1, D_c = 8.2$. The plants and controllers are discretized based on the sampling time $T_h$ to obtain the discrete time equivalent. The utility function used in the experiments is the same as the function presented in Section V. The objective is the joint velocity of each robotic arm tracks a sinusoidal reference input $r[k] = sin(\omega k)$ for $k = 0, 1, 2, \cdots$ with $\omega = \frac{2\pi}{80}$. The disturbance inputs for Plant2 and Plant3 are white noise with the power spectral density of $1$. Plant1 does not have any white noise input. In the wireless network, the interference range and the transmission range are set to 250m. The capacity of the wireless channel is $2Mbps$. The packet size is $260$ bytes. Each simulation runs for $180$ seconds.

Four aspects of the system are evaluated after the first period of the reference signal when the adaptation converges:

1. The average tracking error $\bar{e}_{rr}$, which is the average absolute difference between the plant output and its reference signal.[3];

[3]The results with the optimal delay requirements are presented with the mean and its range of error based on 7 times of simulation runs.

2. The sampling time $T_h$;

3. The end-to-end delay of flows associated with system $h$;

4. The channel utilization, which is the ratio of the total network load to the channel capacity.

## Simulation Results

### Single-hop Scenario

In the first experiment, there are six nodes in the wireless network, each hosting either a plant or a controller. All the nodes are within the transmission range of each other, forming a single-hop network topology.

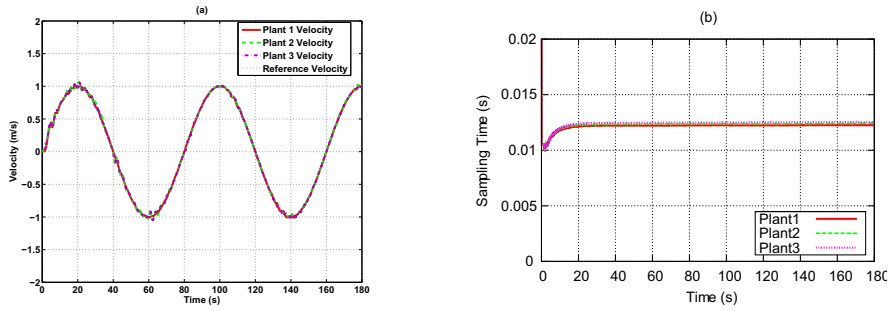


Figure V.7: Velocity Outputs with the Optimal Delay Requirement in Single-hop

Fig. V.7 presents the simulation results with the optimal delay requirement derived from the delay requirement adjustment algorithm. Fig. V.7 (a) shows the plant outputs, and Fig. V.7 (b) illustrates the sampling time convergence of the

three plant-controller pairs. The sampling time quickly converges, and the plant outputs closely follow the reference trajectory. In Plant2 and Plant3, white noise is introduced at a period of 15 seconds, when the outputs deviate from the reference trajectory. Their gaps quickly diminish after a short period of time.

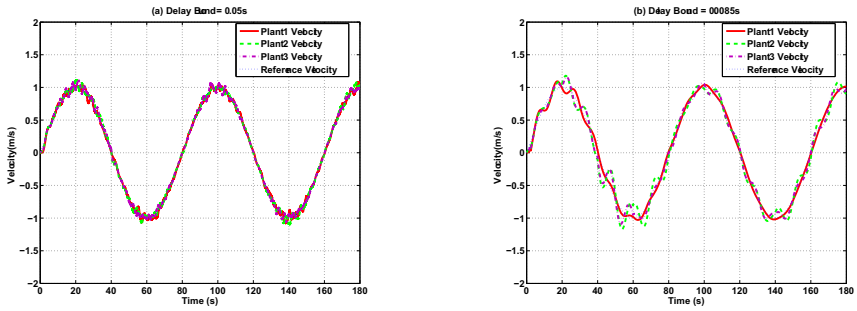


Figure V.8: Velocity Outputs with Different Delay Requirements in Single-hop

Table V.2: Performance Metrics for Different Delay Requirements

| **Delay Requirements** | **Average Track Error** | **Sampling Time** | **Average Delay** | **Channel Utilization** |
|---|---|---|---|---|
| Optimal | $0.007 \pm 0.0043$ | 0.0125 | 0.0121 | 50 % |
| 0.05 | 0.0251 | 0.0087 | 0.0481 | 72 % |
| 0.0085 | 0.0428 | 0.2013 | 0.0087 | 3 % |

Next we compare the performance of the NCS with fixed delay requirements, which are different from the optimal one in Tab. V.2. Fig. V.8 (a) illustrates the plant outputs using the delay requirement of $0.05s$. Fig. V.8 (b) demonstrates the plant outputs using the delay requirement of $0.0085s$. We observe that both

outputs are much worse than those in Fig. V.7. In addition, Plant2 and Plant3 suffer from larger oscillations than Plant1, and cannot track the reference trajectory closely, as shown in Fig. V.8 (b). With a larger delay requirement, the control systems are allowed to send packets with a larger sampling rate, which increases the traffic load of the networks. The average end-to-end delay experienced by the control systems is more than 5 times of the sampling time. Thus, the outputs exhibit a lot of oscillation. On the other hand, a small delay requirement leads to small sampling rates, which degrade the system capability of white noise rejection. Thus, the controller cannot be notified in time about the occurrence of the white noise disturbance.

**Multi-hop Scenario**

Direct communication in wireless networks requires two nodes to be within the transmission range of each other. When they are out of range, intermediate nodes can provide relays to route packets. We evaluate our solution over a multi-hop wireless network with 12 nodes organized in a grid topology. The plants and controllers are deployed on nodes at the network edges. Plant2 resides in the middle of the network. The paths of all the control system pairs are set up using the shortest-path routing algorithm.

Fig. V.9 (a) shows the velocity outputs of the three plants with the optimal delay requirements. Tab. V.3 compares their performance metrics. Compared with the single hop case, the plants experience larger oscillation at the beginning of the

Table V.3: Performance Metrics with Optimal Requirements

| | **Delay Requirements** | **Average Track Error** | **Sampling Time** | **Average Delay** |
|---|---|---|---|---|
| Plant1 | 0.024 | 0.0020 $\pm$ 5.705e-5 | 0.026 | 0.031 |
| Plant2 | 0.035 | 0.0186 $\pm$ 5.797e-7 | 0.038 | 0.048 |
| Plant3 | 0.024 | 0.0159 $\pm$ 3.549e-4 | 0.026 | 0.023 |

simulation. Because it takes longer time to set up the routes between the plant and controller pairs. Plant1 does not have white noise disturbance, so after convergence its velocity output follows the reference signal closely. Although Plant2 and Plant3 have the same amount of white noise input, Plant2 has larger oscillation than Plant3. This is because flows of Plant2 experience larger interference than those of Plant3. As a result, it has a larger sampling time and is more vulnerable to noise.

Table V.4: Performance Metrics of the NCS with Fixed Sampling Time

| | **Average Track Error** | **Sampling Time** | **Average Delay** |
|---|---|---|---|
| Plant1 | 0.0132 | 0.0117 | 0.0807 |
| Plant2 | 0.0223 | 0.0176 | 0.2044 |
| Plant3 | 0.0139 | 0.0117 | 0.0519 |

We further run the experiment with fixed sampling time for the three control systems. In Tab. V.4, we show the average tracking error with the fixed sampling time. Comparing with the errors under optimal sampling time, we observe that the three plants experience larger tracking error. Their sampling rates are about twice of the optimal rates, which leads to much longer delay in a multi-hop network.

When the average delay exceeds the sampling time by orders of magnitude, the tracking error increases significantly.

**Impact of wireless random packet loss**

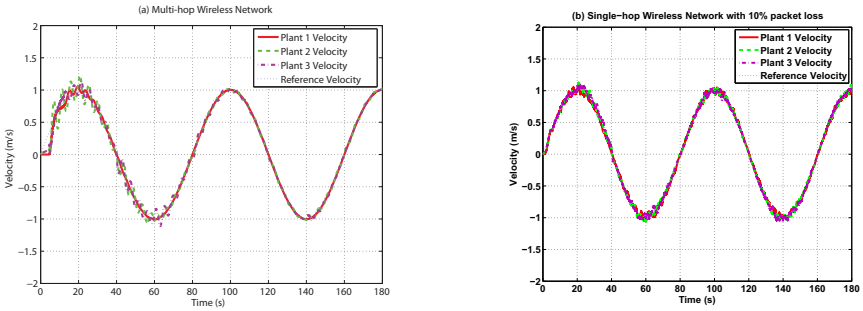


Figure V.9: Velocity outputs

We now set up a single-hop wireless network with 10% random packet loss that may be caused by wireless interference or noise. In Fig. V.9 (b), we show the velocity outputs of the three plants with the optimal delay requirements. Compared with the case without any packet losses, plants experience larger oscillation at the peaks of the sinusoidal outputs. At the peaks, the acceleration of reference velocities changes to the opposite direction, as communication between the plants and the controllers becomes unreliable due to random packet loss, plants need longer time to know the changes. While after the peaks, plants quickly track the reference closely.

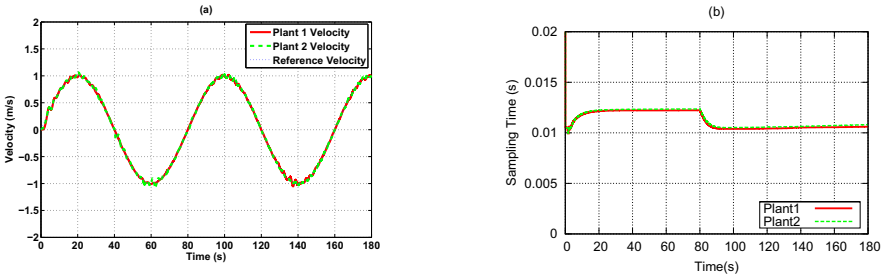


Figure V.10: Velocity Outputs with Background Traffic in Single-hop

**Impact of background traffic**

We then set up a single-hop wireless network with two pairs of plant and controller and two pairs of background traffic. All of them start at the beginning, while the two background Constant Bit Rate (CBR) traffic flows leave the channel at 80 seconds. The two pairs of background traffic both use fixed sampling time of $0.02s$. At the beginning as in Fig. V.10 (b), plants' sampling time quickly converges to around $0.012s$ with the plant outputs stabilize to the sinusoidal reference. When the background traffic leaves, plants' sampling time re-converges to around $0.010s$. and plant outputs still closely follow their reference. This experiment demonstrates that with our algorithm, NCS smoothly handles background traffic in the network.

**Impact of NCS dynamics**

In this experiment, Plant1 and Plant2 start first, with Plant3 starting at 80 seconds. At the beginning as in Fig. V.11 (b), the sampling times of Plant1 and Plant2

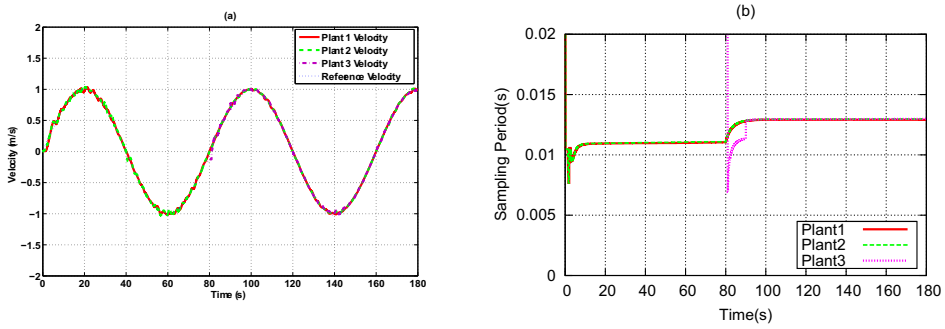


Figure V.11: Velocity Outputs with Dynamic NCS Join in Single-hop

quickly converge to $0.011s$ and their plant outputs stabilize to the sinusoidal references as in Fig. V.11 (a). When the third pair joins in, all the sampling times re-converge to around $0.013s$ after short period of time. Plant1 and Plant2 do not experience much additional oscillation, and their sampling times transit smoothly to the new values. It is because their scheduling is controlled by price, which does not change abruptly.

**Impact of utility function**

Last we change the transfer function of Plant3 to $G(s) = \frac{1}{5s}$. The parameters of its continuous time state space representation become $A_p = 0, B_p = 0.5, C_p = 0.4$. Its utility function is plotted in Fig. V.12. Compared with the old transfer function as in Fig. V.4, only $J$ is changed from $2.93$ to $5$. This leads to a faster increase of the utility function as a function of sampling rate. With the new utility function, when sampling rate is smaller than $10Hz$, Plant3 is more sensitive to price change; when sampling rate is larger than $10Hz$, it is more robust to price

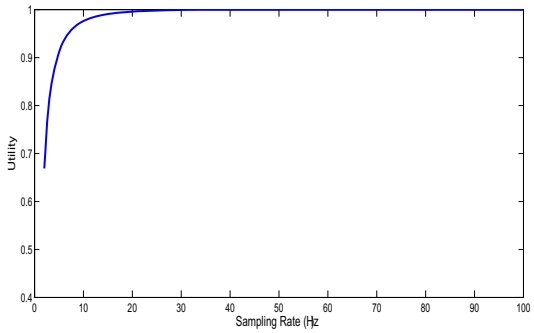


Figure V.12: Utility function for the control system where the transfer function of the plant is $G_p(s) = \frac{1}{Js}$, transfer function of the controller is $G_c(s) = \frac{K_p + K_d s}{s}$, with $J = 5$, $K_d = 32.1$ and $K_p = 8.2$.

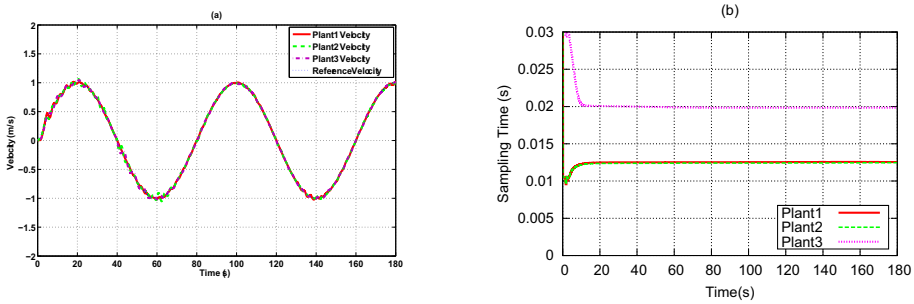


Figure V.13: Velocity Outputs with Plants using Different Utility in Single-hop

change. As a result, in Fig. V.13 (a), Plant3 experiences relatively smaller oscillation than Plant2. And the converged sampling time of Plant3 is larger than that of Plant1 and Plant2 as in Fig. V.13 (b). This is because the utility function of Plant3 is smoother when the sampling time is relatively small (sampling time is inverse of sampling rate in the figure), steeper when the sampling time is relatively large. So the velocity of Plant3 is less sensitive to price changes, in turn more robust to network dynamics, as compared with Plant2. In addition, because of the change of

the utility function, the correlation between price and sampling time also changes. Same price in the network is then interpreted to different sampling times when plants use different utility functions.

# CHAPTER VI

# END-TO-END FAIRNESS OVER NON-CONVEX CAPACITY REGION IN IEEE 802.11-BASED WIRELESS NETWORKS

## Introduction

Fair resource allocation for *end-to-end flows* in multi-hop wireless networks is an essential but challenging problem. The key challenge comes from the location-dependent resource contention. Essentially, the packet transmission along a wireless link will block the packet transmission in its vicinity. As a result, the achievable capacity of a wireless link depends on the scheduling algorithm which determines the set of active transmissions at each time slot. Existing approaches to this problem largely fall into two categories: 1) Joint optimization of scheduling and rate allocation [29, 66, 109, 131, 136] where the resource capacity region is rigorously defined by the schedulability of link-level flows. Though these methods provide theoretically sound solutions for fair resource allocation, they can hardly be implemented in a cost-efficient way due to the intrinsic complexity in the multi-hop scheduling, and are rarely seen in practical deployment due to their incompatibility with the standard IEEE 802.11; 2) Heuristic rate allocation solutions [129, 95], which simplify the scope of resource sharing regions using different neighborhood models and partially rely on the underlying packet scheduling, *e.g.*, IEEE 802.11, to resolve the resource contention among link-level

flows. These works are practical for implementation, but their fairness properties are usually only evaluated on special topologies, like diamond or stack [95].

To date, the following essential questions still remain unanswered: 1) *How to define end-to-end fairness under IEEE 802.11.* It is known that the capacity region of IEEE 802.11 is non-convex [47]. Under the non-convex capacity region, the proportionally fair rate allocation is not unique and not consistent with the trivial fairness in simple analyzable network scenarios. Defining fairness under non-convex regions needs a fresh treatment. 2) *How to achieve end-to-end fairness*. End-to-end fairness is usually achieved via rate allocation algorithms, whose stability and convergence are critical for the overall network performance. The new definition of fairness has to be implementable via a stable rate controller.

In this chapter, we present a new fairness model for IEEE 802.11 wireless network where the capacity region is non-convex. To characterize the desired fairness property, we adopt an axiomatic approach based on the game theoretic framework. Recall that Nash Bargaining Solution (NBS), only defined over convex regions, from cooperative game theory has been applied in [132] as a unified framework for network fairness and efficiency. In particular, NBS is consistent with the concept of proportional fairness defined by the optimal resource allocation framework [50]. Here we seek a solution that is unique, coincides with the NBS under the convex cases and approximates the NBS under the non-convex cases. Our new fairness model is based on the Nash extension solution (NES), which is shown to meet all these requirements [23, 22]. The NES can be constructed from the NBS of the smallest convex regions that contain the original problem with non-convex

regions. The solution to the convex problem is then scaled until all the rate allocation intersects with the non-convex region. We further present an efficiency enhanced version of Nash extension solution, which pushes the NES to the strong Pareto frontier by increasing the benefits of some players when the utilities of the rest players remain the same.

Based on the construction method of NES and its extension solution, we present a time-decomposed price-based rate allocation algorithm. To best characterize the convex hull that contains the non-convex region as defined by the NES concept, we adopt the maximal clique approximation model [131]. The price-based rate allocation algorithm is similar to the traditional dual-based rate allocation algorithm. With the clique-formed price-based algorithm, the NES is approached by scaling the capacity of convex cliques. To do so, a region scaler $\beta$ is defined in a slowly varying system and iteratively scales the capacity of the regions to approach the IEEE 802.11 achievable capacity. We prove that both the price-based rate allocation algorithm and the scaler update algorithm are stable by Lyapunov functions, and the overall system is stable when the scaler varies slowly. The algorithm is further extended to utilize the slack resource in the network, which leads to the efficiency enhanced NES. Fully distributed implementation of the rate allocation algorithms based on the gossip protocol is also presented. The simulation study over a variety of topologies (e. g., random and dynamic) validates the performance of our algorithms and demonstrates this theoretically sound new fairness model for IEEE 802.11 networks.

This chapter makes the following contributions to the field. First, it defines a new fairness model for non-convex capacity regions based on Nash extension solution. It further presents an efficiency enhanced fairness model to address the limit of weak Pareto optimality of NES. Both models will lead to a unique rate allocation solution under non-convex capacity regions. While the NES coincides with the fair allocation of NBS under the convex cases and approximates the NBS under the non-convex cases, the efficiency enhanced model leverages fairness to improved resource utilization. Second, we present a time-decomposed price-based rate allocation algorithm that iteratively converges to the fair solution and prove its stability. Distributed implementation of the algorithm is also presented and evaluated using a simulation-based study.

The rest of this chapter is organized as follows. We first describe the problems in Sec. VI. A new fairness model in IEEE 802.11 is defined in Sec. VI. The rate-based algorithm to implement the new fairness model is presented and its stability is formally proved in Sec. VI, VI and VI. Finally we present our evaluation results in Sec. VI.

## Problem Description

Non-convex capacity region in IEEE 802.11 networks

The problem of identifying the achievable capacity region is extremely complicated in IEEE 802.11 due to its nature of random-access-based scheduling. [47] points out two important properties of the capacity region in IEEE 802.11:

- The capacity region under IEEE 802.11 scheduling is not convex.
- Asymmetric knowledge of the source or destination nodes leads to asymmetric capacity regions.

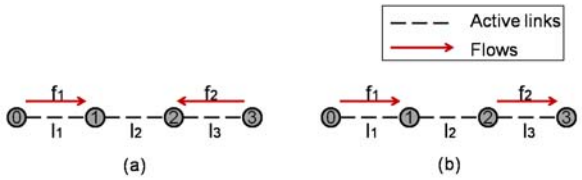


Figure VI.1: Scenarios of two contending one-hop flows

Here we demonstrate these two properties via a ns-2 based simulation study. Fig. VI.1 shows the simulation scenarios and Fig. VI.2 plots their achievable capacity regions respectively.

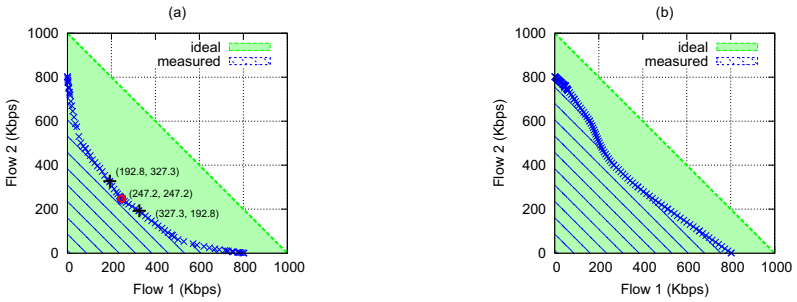


Figure VI.2: Achievable capacity regions of two contending one-hop flows under IEEE 802.11. In the experiment, RTS/CTS is enabled. The channel capacity is 1Mbps. The packet size is 1000 bytes.

Limitation of existing fairness models over the non-convex problem

We first review the existing fairness models and identify their limitations when the network capacity region is not convex. Let $\boldsymbol{x} = (x_f, f \in \mathcal{F})$ denote the rate vector of the end-to-end flows $f \in \mathcal{F}$ and $\Pi$ be the set of all feasible rate vectors. Two widely used fairness criteria are proportional fairness and max-min fairness. Recent literature [93] has shown that the max-min fairness model leads to severe resource inefficiency in wireless networks, as all the flow rates are bounded by the worst one, without considering the differences of their location-dependent resource consumption. Thus, we focus on the proportional fairness model in the discussion below.

According to [50], a rate allocation $\hat{\boldsymbol{x}} = (\hat{x}_f, f \in \mathcal{F}) \in \Pi$ satisfies *proportional fairness*, if

$$\forall \boldsymbol{x} \in \Pi, \sum_{f \in \mathcal{F}} \frac{x_f - \hat{x}_f}{\hat{x}_f} \leq 0 \tag{VI.1}$$

Intuitively, if $\hat{\boldsymbol{x}}$ satisfies the proportional fairness, the rate increase of one user ($f \in \mathcal{F}$) will result in the rate decrease of another user $f' \in \mathcal{F}$. Moreover, the proportion by which $\hat{x}_{f'}$ decreases will be larger than the proportion of $x_f$'s increase. It has been shown in [50] that the proportionally fair rate allocation can be derived from the solution of the network utility maximization problem, where the utility function is a logarithm function, *i.e.*, $\max \sum_{f \in \mathcal{F}} \ln(x_f), \boldsymbol{x} \in \Pi$. Further, the work of [132] has shown that the proportionally fair rate allocation is also the solution of Nash Bargaining Solution (NBS) problem. When $\Pi$ is a convex set, there exists a unique proportionally fair rate allocation vector. When the feasible rate vector set is non-convex, as in the case of IEEE 802.11[1], there may exist multiple allocation vectors that satisfy proportional fairness, as shown in Fig. VI.2(a). The two points $(192.8, 327.3)$ and $(327.3, 192.8)$ marked by black "+" signs both satisfy the proportional fairness criteria (Inequality (VI.1)). Further, both points deviate from the trivially fair point $(247.2, 247.2)$ marked by the red circle. To summarize, the above observations significantly challenge the feasibility of using proportional fairness in IEEE 802.11-based wireless network, as 1) the proportionally fair allocation is not unique; 2) the proportionally fair allocation is not consistent with the trivial fairness even in simple network scenarios, when the feasible rate vector set is non-convex.

[1]Since the capacity region, defined as the set of feasible edge rate vectors, is shown to be non-convex under IEEE 802.11 [47], it is trivial to show that the set of feasible end-to-end flow rate vector will also be non-convex.

**Nash-Extension-Solution-based Fairness Model**

In this section, we define a new fairness model for IEEE 802.11 wireless networks where the capacity region is non-convex. There are three basic requirements for the new fairness model.

1. Uniqueness. Only one rate vector within the capacity region is the fair allocation point.

2. Pareto optimality. No alternative allocation within the capacity region can improve the rate of any individual in the fair allocation.

3. Convexity consistence. The fair allocation is consistent with the proportional fairness (*e.g.*, NBS), when the capacity region is convex,

To characterize the desired fairness properties, we seek a solution in the game theoretic framework that is unique, coincides with the NBS under convex cases and approximates the NBS under non-convex cases. The Nash extension solution has been shown to meet all these requirements in the cooperative game theory [23, 22]. Here we formally define the Nash extension solution in the settings of wireless rate allocation.

In a multi-hop wireless network, the end-to-end flows $F$ constitute the set of players in the game, which compete for the shared wireless bandwidths. The rate allocation vector $\boldsymbol{x} = \{x_f, f \in F\}$ denotes the utility vector of all the players.

$\Pi \subset \Re^{\mathcal{F}}$ ($\mathcal{F} = |F|$) is the set of all feasible utility vectors. To formally define the bargaining problem, we first introduce the following definitions and notations[2]:

- ***d**-comprehensive*. Give a point $\boldsymbol{d} \in \Re^{\mathcal{F}}$ and a set $\Pi \subset \Re^{F}$. $\Pi$ is $\boldsymbol{d}$-comprehensive if $\boldsymbol{d} \leq \boldsymbol{x} \leq \boldsymbol{z}$ and $\boldsymbol{z} \in \Pi$ implies $\boldsymbol{x} \in \Pi$;
- *convex hull of a set* $\Pi \subset \Re^{\mathcal{F}}$ is the smallest convex set containing the set $\Pi$, specified as $con(\Pi)$;
- *weak Pareto frontier of* $\Pi$ is denoted as $WP(\Pi) \equiv \{\boldsymbol{x} \in \Pi | \boldsymbol{y} \gg \boldsymbol{x}$ implies $\boldsymbol{y} \notin \Pi\}$;
- *strong Pareto frontier of* $\Pi$ is denoted as $SP(\Pi) \equiv \{\boldsymbol{x} \in \Pi | \boldsymbol{y} \geq \boldsymbol{x}$ implies $\boldsymbol{y} \notin \Pi\}$.

Let $\Pi \in \Re^{\mathcal{F}}$ be the set of all feasible rate allocations. It is non-empty, compact, but not necessarily convex. Further, we denote the initial agreement point of the game as $\boldsymbol{x}^0$, which represents the initial utilities of the players without any cooperation in order to enter the game. The *domain of Nash extension (NE) bargaining problem* considered in the context of wireless rate allocation is denoted as $\Sigma_B = \{(\Pi, \boldsymbol{x}^0)\}$, which satisfies 1) $\Pi$ is $\boldsymbol{x}^0$-comprehensive; 2) $\{\boldsymbol{x} \in \Pi | \boldsymbol{x} \gg \boldsymbol{x}^0\} \neq \emptyset$, which is a set of rate allocations $\boldsymbol{x}$ that are acceptable to all the flows.

[2]The vector inequality notations in this chapter are defined as follows: 1) $\boldsymbol{x} > \boldsymbol{y}$ means that $x_i \geq y_i, \forall i \in F$ with strict inequality for at least one $i$; 2) $\boldsymbol{x} \geq \boldsymbol{y}$ means that $x_i \geq y_i, \forall i \in F$; 3) $\boldsymbol{x} \gg \boldsymbol{y}$ means that $x_i > y_i, \forall i \in F$.

Then a NE bargaining solution becomes a function $\varphi : \Sigma_B \to \Re^F$, such that $\forall(\Pi, \boldsymbol{x}^0) \in \Sigma_B, \varphi(\Pi, \boldsymbol{x}^0) \in \Pi$. It specifies a rate allocation within the feasible utility set, with three reasonable properties satisfied: 1) fair allocation among all players; 2) efficient usage of resources; 3) consistency with NBS when $\Pi$ is convex. These properties are precisely encapsulated by the following axioms of NE solution.

*Definition 1 Nash Extension Solution (NES).* A bargaining solution $\varphi : \Sigma_B \to \Re^F$ is a Nash extension solution, if $\boldsymbol{x}^* = \varphi(\Pi, \boldsymbol{x}^0)$ satisfies the *Axioms* A1-A6.

A1 Individual rationality: $\boldsymbol{x}^* \geq \boldsymbol{x}^0$;

A2 Feasibility: $\boldsymbol{x}^* \in \Pi$;

A3 Weak Pareto optimality: $\forall \boldsymbol{x}$, if $\boldsymbol{x} \gg \boldsymbol{x}^*$, then $\boldsymbol{x} \notin \Pi$;

A4 Scale invariance: $\forall \lambda, \varphi(\lambda(\Pi), \lambda(\boldsymbol{x}^0)) = \lambda(\varphi(\Pi, \boldsymbol{x}^0))$, where $\lambda$ is an affine transformation on $\Re^F$: $\lambda(\boldsymbol{x}) = a + b\boldsymbol{x}, b > 0$, $\lambda(\Pi) = \{\boldsymbol{z} \in \Re^F | \boldsymbol{z} = \lambda(\boldsymbol{x}), \boldsymbol{x} \in \Pi\}$;

A5 Independence of Irrelevant Alternatives: If $\Pi' \subset \Pi$, $\boldsymbol{z}^0 = \boldsymbol{x}^0$, and $\varphi(\Pi, \boldsymbol{x}^0) \in \Pi'$, then $\varphi(\Pi', \boldsymbol{z}^0) = \varphi(\Pi, \boldsymbol{x}^0)$;

A6 Symmetry: If for all permutation operator $\phi$, $\phi(\Pi) = \Pi$ and $\phi(\boldsymbol{x}^0) = \boldsymbol{x}^0$ then $\varphi_f(\Pi, \boldsymbol{x}^0) = \varphi_{f'}(\Pi, \boldsymbol{x}^0), \forall f, f' \in \mathcal{F}$;

A7 Ethical monotonicity: If $\Pi \subset \Pi', \boldsymbol{x}^0 = \boldsymbol{z}^0$, and $\varphi(con(\Pi'), \boldsymbol{z}^0) \in con(\Pi)$, then $\varphi(\Pi, \boldsymbol{x}^0) \leq \varphi(\Pi', \boldsymbol{z}^0)$.

The above axioms encapsulate both concepts of fairness in rate allocation (A4-A7) and weak Pareto optimality in resource utilization (A3). Among these axioms, A1-A6 are similar to the definition of NBS [130]. The scale invariance axiom (A4) states that the NE bargaining solution is scale-invariant, *i.e.*, the solution does not change if the utility is positively scaled. The symmetry axiom (A6) implies that the bargaining solution does not depend on any specific labels and no player is superior than others, *i.e.*, players with the same initial points and objectives will achieve the same final utility. The ethical monotonicity axiom (A7) requires that if the feasible set of a bargaining problem is reduced in a way that leaves the "ethical point" unchanged, then no agent should benefit from the decrease of opportunities. This captures the notion of "fairness" that all agents should share any such gains or losses. Here the "ethical point" is defined as the NBS of the convex hull of the feasible set ($con(\Pi)$). The NES compromises away from the most ethically desirable point to the maximal feasible allocation in a way that it distributes the losses over all the players [23]. As a result, the NES is coincident with the NBS if the problem is convex, and approximates the NBS under non-convex cases.

The weak Pareto optimality (A3) shows that no other distributions of the resource will strictly benefit all the players simultaneously, which indicates efficient resource utilization when fairness is guaranteed. However, the weak Pareto optimality condition implies that there are solutions which can increase the benefits of some players with the utilities of the rest players remaining the same. Thus,

we further present an efficiency enhanced version of Nash extension solution as follows:

*Definition 2 Nash Extension Solution with Efficiency Enhancement(NESEE).* A Nash extension solution $\varphi^e : \Sigma_B \rightarrow \Re^{\mathcal{F}}$ is a Nash extension solution with efficiency enhancement, if $\boldsymbol{x}^e = \varphi^e(\Pi, \boldsymbol{x}^0)$ satisfies the following conditions:

C1 Strong Pareto optimality: $\boldsymbol{x}^e \in SP(\Pi)$;

C2 Consistency to Nash Extension Solution: $\boldsymbol{x}^* \leq \boldsymbol{x}^e$, where $\boldsymbol{x}^*$ is the Nash extension solution.

## NES Implementation in Wireless Networks

In this section, we show how Nash extension solution can be formulated as a rate allocation problem in wireless networks. The works of [23, 22] show that the NES over a non-convex set can be constructed via its convex hull as follows. First, let the mapping $M : \Sigma_B \rightarrow \Re^{\mathcal{F}}$ defined as:

$$M(\Pi, \boldsymbol{x}^0) \equiv line(\varphi(con(\Pi), \boldsymbol{x}^0), \boldsymbol{x}^0). \tag{VI.2}$$

$M(\Pi, \boldsymbol{x}^0)$ is the line segment connecting $\boldsymbol{x}^0$ to the NES[3] of the convex hulls $con(\Pi)$ and $\boldsymbol{x}^0$. Then the NES can be constructed as:

$$\varphi(\Pi, \boldsymbol{x}^0) \equiv \max_{\boldsymbol{x}} \{\boldsymbol{x} \in M(\Pi, \boldsymbol{x}^0) \cap \Pi\}. \tag{VI.3}$$

[3]It is also the NBS, since the vector set is convex.

where "max" indicates the maximal element with respect to the partial order on $\Re^{\mathcal{F}}$. The point $\varphi(\Pi, \boldsymbol{x}^0)$ is the intersection of the weak Pereto frontier of $\Pi$ and the line segment $M$. Since $M$ is a line segment, this maximal element exists and is unique. Thus, NES is nonempty and unique on $\Sigma_B$. Based on this construction

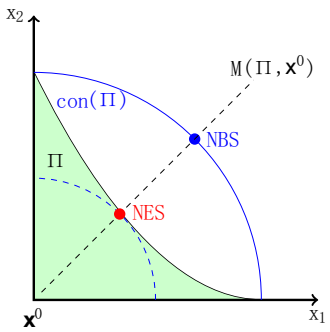


Figure VI.3: An Example of the NES Implementation

method, four sub-problems need to be solved to obtain $\boldsymbol{x}^*$ for the NES implementation, with an illustrative example in Fig. VI.3.

1. Identify $con(\Pi)$, the smallest convex set containing $\Pi$;
2. Identify NBS of $con(\Pi)$;
3. Identify the line segment $M(\Pi, \boldsymbol{x}^0)$ as in Eq. (VI.2);
4. Identify $\varphi(\Pi, \boldsymbol{x}^0)$, the NES on $M$ as in Eq. (VI.3).

Identify $con(\Pi)$

The convex hull $con(\Pi)$ needs to satisfy the following three properties.

- Feasibility, in terms of the schedulability of the region. The convex hull $con(\Pi)$ has to contain the IEEE 802.11 schedulable region $\Pi$. Thus we look for a region that satisfies the necessary condition of schedulability.
- Convexity, in terms of the shape of the capacity region.
- Smallest set, in terms of the size of the capacity region. Thus we look for a region that approximates the sufficient condition of schedulability.

Characterizing the channel resource sharing region is related to the issue of schedulability, *i.e.*, whether the edge rate vector $\boldsymbol{y}$ is schedulable given the channel capacity and the network topology. However, it is known [45] that satisfying the sufficient and necessary condition of schedulability in a wireless network involves finding the independence number of a graph, which is an NP-hard problem. Several approximation models of resource sharing units are proposed in the existing literature [45, 131, 53, 3]. In this chapter, we adopt the maximal clique approximation [131] to model the capacity regions as it has a better approximation factor compared with the interference set models [53, 3]. In this clique-based wireless resource allocation framework, the resource sharing regions are characterized by *maximal cliques* in the *wireless link contention graph* $G_c$ of the network. In a wireless link contention graph, the vertices in a maximal clique represent a maximal resource sharing region. They are mutually contending wireless links,

among which at most one subflow may transmit at any given time. Formally, let $Q$ be the set of all maximal cliques in $G_c$. For a maximal clique $q$ in the wireless link contention graph $G_c$, $V(q) \subseteq L$ is the set of its vertices. The resource constraint under this clique model is formulated based on the necessary condition of schedulability, as $\sum_{l \in V(q)} y_l \leq C, \forall q \in Q$, where $C$ is the channel capacity.

After the feasible capacity regions have been constructed, next we adjust it to the tightest convex set. For each capacity region, the maximum achievable capacity under IEEE 802.11 is proved to depend on the number of users in the region [10], which can be computed analytically, if the ideal value of the capacity is known. Under IEEE 802.11, we are able to identify one Pareto point in this region. Consider the scenario where only one link is active, this active link itself can achieve the maximum capacity. This non-convex capacity region is tightly bounded by this maximum achievable capacity on the dimension of each wireless link. Thus we define $C$ as the maximum achievable capacity of a single active link for each clique. This set of linear equations defines a convex edge rate vector region which we consider to be the smallest one that contains the IEEE 802.11 capacity region. Let the clique-flow matrix $\boldsymbol{R} = \{R_{qf}\}$ represent the "resource usage pattern" of each flow where $R_{qf} = |V(q) \bigcap E(f)|$ represents the number of subflows that flow $f$ has in clique $q$. The convex hull $con(\Pi)$ of the flow rate vector can be derived from $\boldsymbol{Rx} \leq \boldsymbol{C}$.

Identify NBS of $con(\Pi)$

If we take the initial agreement point $\boldsymbol{x}^0$ as a vector of zero, then the NBS of $con(\Pi)$ can be derived from the following rate allocation problem [130, 132].

$$\hat{\mathbf{P}}: \quad \textbf{maximize} \quad \sum_{f \in F} \ln(x_f) \tag{VI.4}$$

$$\textbf{subject to} \quad \boldsymbol{R} \cdot \boldsymbol{x} \leq \boldsymbol{C} \tag{VI.5}$$

$$\boldsymbol{x} \geq \mathbf{0} \tag{VI.6}$$

The objective function Eq.(VII.2) has the form of the proportional fairness, which maximizes the aggregated utility of all flows. Constraint (VII.3) characterizes $con(\Pi)$. Constraint (VII.4) shows that the rate of each flow is greater than the initial agreement point.

Identify $M(\Pi, \boldsymbol{x}^0)$

The line segment $M$ is determined by the two end points: the NBS of the convex problem $\hat{x} = \varphi(con(\Pi), \boldsymbol{x}^0)$ and $\boldsymbol{x}^0$. Let $\boldsymbol{x}^m$ be a point on the line segment $M$, then $\forall i, j \in F, i \neq j$, the flow rates need to satisfy

$$\frac{x_i^m - x_i^0}{x_j^m - x_j^0} \equiv \frac{\hat{x}_i - x_i^0}{\hat{x}_j - x_j^0}$$

Assuming that the minimum rate vector $\boldsymbol{x}^0 = \mathbf{0}$, the above equation can be written in the vector form as

$$\boldsymbol{x}^m = \beta\hat{\boldsymbol{x}}, \beta \in [0, 1]. \tag{VI.7}$$

With constraint (VII.3), we have $\boldsymbol{R} \cdot \boldsymbol{x}^m \leq \beta\boldsymbol{C}$. Thus, points on the segment line $M$ can be solved in the following non-linear optimization problem by varying *region scaler* $\beta \in [0, 1]$.

$$\begin{aligned} \mathbf{P}_m: \quad & \textbf{maximize} \quad \sum_{f \in F} \ln(x_f) \\ & \textbf{subject to} \quad \boldsymbol{R} \cdot \boldsymbol{x} \leq \beta\boldsymbol{C} \\ & \qquad\qquad\quad \boldsymbol{x} \geq 0 \end{aligned} \tag{VI.8}$$

Identify $\varphi(\Pi, \boldsymbol{x}^0)$

After $M$ being determined, we need to find the point on the line corresponding to its intersection with $\Pi$. The NES is the intersection of $M$, the solution to the non-linear optimization problem $\mathbf{P}_m$ and the feasible allocation set $\Pi$:

$$\varphi(\Pi, 0) = \{\boldsymbol{x}^* | \boldsymbol{x}^* \in WP(\Pi), \boldsymbol{R} \cdot \boldsymbol{x}^* = \beta\boldsymbol{C}, \beta \in [0, 1]\} \tag{VI.9}$$

Assuming that the achievable capacity vector under IEEE 802.11 is $\tilde{\boldsymbol{C}}$ for the clique approximation of the resource sharing units, identifying NES is equivalent

to find a $\beta$, which solves the linear optimization problem below:

$$\begin{aligned} \mathbf{M}: \quad & \textbf{maximize} \quad \beta \\ & \textbf{subject to} \quad \beta \boldsymbol{C} \leq \tilde{\boldsymbol{C}} \\ & \qquad\qquad\quad 0 \leq \beta \leq 1 \end{aligned} \tag{VI.10}$$

## NES-Fair Rate Allocation Algorithm

Based on the construction of NES, we present a rate allocation algorithm that achieves NES-fairness in wireless networks. The algorithm consists of two dynamic systems evolving at two timescales:

- The boundary-layer system, where the region scaler $\beta$ remains constant. Flows adapt their rates towards the fixed target capacity of $\beta \boldsymbol{C}$ according to $\mathbf{P}_m$.
- The reduced system or slow system, where the flow rates have converged for a particular set of $\beta \boldsymbol{C}$. The capacity regions, or the cliques then tune the region scaler $\beta$ to approach the achievable capacity under IEEE 802.11.

The Boundary-layer System

With the value of $\beta$ fixed, the optimal resource allocation of problem $\mathbf{P}_m$ can usually be solved by a price-based dual problem $\mathbf{D}$ of $\mathbf{P}_m$: $\min_{\boldsymbol{\mu}\geq 0} D(\boldsymbol{\mu})$.

$$\begin{aligned} D(\boldsymbol{\mu}) &= \sum_{f\in F} \max_{x_f}(\ln(x_f) - x_f \sum_{q:E(f)\cap V(q)\neq\emptyset} \mu_q R_{qf}) \\ &+ \sum_{q\in Q} \mu_q \beta C_q \end{aligned} \tag{VI.11}$$

where the vector of Lagrange multipliers $\boldsymbol{\mu} = (\mu_q, q \in Q)$ is called price and interpreted as the cost of a unit flow accessing the resource sharing region characterized by the maximal clique $q$. The dual problem $\mathbf{D}$ is solved via a gradient projection algorithm [131]. This leads to an iterative algorithm that can be executed in a distributed way to achieve the optimal solution of $\mathbf{P}_m$. In particular, the price $\mu_q$ at clique $q$ is adjusted as follows:

$$\mu_q(t+1) = [\mu_q(t) - \gamma(\beta C_q - \sum_{f:E(f)\cap V(q)\neq\emptyset} x_f \cdot R_{qf})]^+ \tag{VI.12}$$

The flow rate $x_f$ can be adjusted at the source of the flow when it receives an ack packet from the destination node, so that its net benefit (difference between utility and cost) is maximized:

**maximize** $\ln(x_f) - \lambda_f x_f$

It is easy to see that $x_f = 1/\lambda_f$ at optimum. Here $\lambda_f$ is the price of a flow $f$: $\lambda_f = \sum_{q:E(f)\cap V(q)\neq\emptyset} R_{qf}\mu_q$.

The Reduced System

In the reduced system, $\beta$ is adjusted towards the optimal solution to $\mathbf{M}$. Expanding constraint (VI.10) in a matrix form:

$$\begin{bmatrix} \beta C_1 & - & \tilde{C}_1 \\ \beta C_2 & - & \tilde{C}_2 \\ & \vdots & \\ \beta C_{|Q|} & - & \tilde{C}_{|Q|} \end{bmatrix} \leq 0.$$

The inequations can be replaced by: $\max_{q\in Q}(\beta C_q - \tilde{C}_q) \leq 0$.

Hence, $\beta$ can be updated according to the ordinary differential equation (ODE):

$$\dot{\beta} = -\alpha\gamma \max_{q\in Q}(\beta C_q - \tilde{C}_q(\boldsymbol{x}^m)) \tag{VI.13}$$

where $\alpha > 0$ is a damping factor. The ODE increases the region scaler when all the cliques are under-utilized, *i.e.*, $\forall q \in Q, \beta C_q < \tilde{C}_q$; and decreases the region scaler when any clique is over utilized, *i.e.*, $\exists q \in Q, \beta C_q > \tilde{C}_q$.

Stability of the Time-decomposed Algorithm

We first show that the two systems are stable individually. Then we study the condition when the entire system is semi-globally stable and converges to the equilibrium point exponentially fast.

First, the boundary-layer model of the system is shown to be globally asymptotically stable in [131]. Here we focus on the stability of the reduced system.

**Theorem 1** Let $\mathcal{W}(\beta) = \sum_{q\in Q}(\beta C_q - \tilde{C}_q)^2$. The strictly convex function $\mathcal{W}$ is a Lyapunov function for the system of Eq. (VI.8) and Eq. (VI.13). The unique value $\beta$ at $\mathcal{W} = 0$ is a stable point of the system, to which all trajectories converge.

$$\begin{aligned}
\textbf{Proof}: \frac{d\mathcal{W}(\beta)}{dt} &= \frac{\partial \mathcal{W}(\beta)}{\partial \beta}\frac{d\beta}{dt} \\
&= 2C_q \sum_{q\in Q}(\beta C_q - \tilde{C}_q(\boldsymbol{x}^m)) \\
&\quad \cdot(-\alpha\gamma)\max_{q\in Q}(\beta C_q - \tilde{C}_q(\boldsymbol{x}^m)) \\
&\leq -2\alpha\gamma C_q \sum_{q\in Q}(\beta C_q - \tilde{C}_q(\boldsymbol{x}^m))^2
\end{aligned}$$

Therefore, $\mathcal{W}(\beta) = 0$ at $\max_{q\in Q}(\beta C_q - \tilde{C}_q(\boldsymbol{x}^m)) = 0$. By LaSalle's invariance principle, the reduced system Eq. (VI.8), (VI.13) converges to the optimal point of the system. □

Finally, we show the overall system is semi-globally exponentially stable.

**Theorem 2** The boundary-layer system given by (VI.12) is locally exponentially stable uniformly in $\{\beta\}$. That is, there exists a $r > 0$, such that

$$||\boldsymbol{\mu}(t, ||\boldsymbol{x}(0)||, \beta) - \boldsymbol{\mu}^*(\beta)|| \leq K||x(0) - x^*(\beta)||exp[-\gamma t],$$

$\forall(\boldsymbol{\mu}, \beta) \in \{||\boldsymbol{\mu} - \boldsymbol{\mu}^*(\beta)|| \leq r\} \times \mathcal{B}_\beta$.

**Proof**: Assuming that $\mu_q > 0, q \in Q$, (VI.12) can be written in the continuous form: $\dot{\mu}_q = -\gamma(\beta C_q - \sum_{f \in F} x_f \cdot R_{qf})$. Linearizing the boundary-layer system (VI.12) around the equilibrium point we get: $\delta\dot{\boldsymbol{\mu}} = A\delta\boldsymbol{\mu}$, where $A$ is the diagonal matrix defined as

$$-\gamma \begin{bmatrix} \sum_{f \in F}(R_{1f}x_f^*)^2 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \vdots & 0 \\ 0 & \cdots & \sum_{f \in F}(R_{Qf}x_f^*)^2 \end{bmatrix}$$

with $x_f^* = x_f^*(\boldsymbol{\mu}^*(\beta))$. Since $A$ is strictly negative definite, *i.e.*,$A < 0$, all the eigenvalues of $A$ are negative. Therefore, the system is locally exponentially stable. The eigenvalues are continuous functions of the elements of $A$, the elements of $A$ are continuous functions of $\beta$ where $\beta$ lies in a compact set $\mathcal{B}_\beta$. Hence $A$ is uniformly Hurwitz in $\beta$. □

Efficiency Enhancement

First we introduce two states of a clique: balanced and active. If the measured capacity of a clique $\tilde{C}_q$ is close to the scaled capacity $\beta C_q$, the clique is balanced; otherwise, it is active. Formally, a flag $h_q$ marks the clique state information,

$$h_q = \begin{cases} B & \text{balanced, if } |\beta C_q - \tilde{C}_q| < \varepsilon \\ A & \text{active, otherwise.} \end{cases}$$

where $\varepsilon$ is a constant value satisfying $\varepsilon \ll C_q$.

By the definition of NES, the time-decomposed algorithm can achieve the same rate proportion among different users as in their convex counterparts. However, at weak Pareto optimality, the flow rates may no longer increase when certain region is balanced, while it is possible that some flows can utilize the residual resource without affecting the balanced regions. This condition is precisely characterized in our proposed Nash Extension Solution with Efficiency Enhancement (NESEE).

To achieve NESEE, we define a slack clique set $S_q$ to keep track of regions with residual capacities. If all the cliques a flow traverses are in $S_q$, the flow can increase its rate without affecting the balanced regions. These flows are recorded by a slack flow set $S_f$.

Table. VI.1 describes the NESEE algorithm in detail. Precisely, when the maximum difference between the scaled capacity $\beta C_q$ and the measured capacity $\tilde{C}_q$ of all the active cliques is within the range of $[-\varepsilon, 0]$, the status of the clique

$q$ with the maximum difference will be set to balanced. At the same time, all the flows passing $q$ will be notified and they will no longer update their price. If active cliques exist, the region scaler will continuously increase to further distribute the available resource in the network.

Table VI.1: Implementation of NESEE algorithm

| **NESEE Rate Adaptation** |
|---|
| 1) *Initialization*<br>$\forall q \in Q, h_q = A.$<br>Create the slack clique set $S_q$ and slack flow set $S_f$,<br>which contain all the cliques and all the flows respectively. |
| 2) $S_q$ *filtering*<br>If $-\varepsilon < \max_{q \in S_q}(\beta C_q - \tilde{C}_q) < 0$, let $i = \arg\max_{q \in S_q}(\beta C_q - \tilde{C}_q)$,<br>$S_q \Leftarrow S_q - \{i\}$. |
| 3) $S_f$ *filtering*<br>$\forall f \in S_f$, if $f$ passes clique $i$, set $S_f \Leftarrow S_f - \{f\}$. |
| 4) *Scaler update*<br>$\beta(t+1) = \beta(t) - \alpha\gamma \max_{q \in S_q}(\beta(t) C_q - \tilde{C}_q).$ |
| 5) *Price update*<br>$\forall q \in S_q, \mu_q(t+1) = [\mu_q(t) - \gamma(\beta C_q - \sum_{f: E(f) \cap V(q) \neq \emptyset} x_f \cdot R_{qf})]^+$ |
| 6) *Flow rate adaptation*<br>$\forall f \in S_f, x_f(t+1) = 1/\lambda_f(t+1).$<br>If $S_f \neq \emptyset$, go to 2); Else Done. |

## Fully Distributed Rate Allocation Algorithm

The algorithms presented in Section VI require the update of $\beta$ based on the knowledge of the achievable capacities of all the cliques in the reduced system,

which is done in a centralized manner. Nevertheless, the unique graphical properties of the wireless link contention graph have the potential to facilitate decentralized algorithms. We allow flows communicate with the cliques they traverse to help propagate the scaler information.

In the distributed algorithms, each clique has two states: frozen and active. If the value of a clique's region scaler is obtained from another clique, its state is frozen; otherwise it is active. Formally, the state of clique $q$ is marked by flag $h_q$,

$$h_q = \begin{cases} O & \text{frozen, if } id_q \neq q \\ A & \text{active, otherwise.} \end{cases}$$

We also define a region scaler information set $B$ containing: $\beta$, the value of the region scaler; $id$, an id identifer marking the source clique from which $\beta$ is acquired; $t$: a time stamp to ensure the freshness of $\beta$. Each clique $q$ maintains a set $B_q$. Each flow also has a set $B_f$ to help facilitate the scaler exchanges among cliques.

### Distributed NES Algorithm

We present a distributed implementation of the NES algorithm in Table. VI.2. In the flow-clique signal exchange phase, a larger or older scaler will always get replaced. When a flow $f$ reaches its destination, it will feedback $B_f$ to the source node to keep track of the $\beta$ information.

Table VI.2: Distributed Implementation of NES algorithm

**Distributed NES Rate Adaptation**

1) *Initialization*
   $\forall q \in Q, \beta_q = 1, id_q = q, t_q = 0, h_q = A.$
   $\forall f \in F, \beta_f = 1, id_f = -1, t_f = 0.$
2) *Flow-Clique signal exchange*
   When a flow $f$ passes clique $q$,
   if $id_f != id_q \&\& \beta_q < \beta_f || id_f == id_q \&\& t_f < t_q$,
   $B_f = B_q$
   if $id_f != id_q \&\& \beta_f < \beta_q || id_f == id_q \&\& h_q = O \&\& t_q < t_f$,
   $B_q = B_f, h_q = O.$
3) *Clique price update*
   $\mu_q(t+1) = [\mu_q(t) - \gamma(\beta_q C_q - \sum_{f:E(f)\cap V(q)\neq\emptyset} x_f \cdot R_{qf})]^+$
4) *Clique scaler update*
   if $h_q == O$ and $\beta_q C_q - \tilde{C}_q > \varepsilon$, then $h_q = A$
   if $h_q == A, \beta_q(t+1) = \beta_q(t) - \alpha\gamma(\beta_q(t) C_q - \tilde{C}_q)$.
5) *Flow rate adaptation*
   $\forall f \in F, x_f(t+1) = 1/\lambda_f(t+1).$

As every flow may traverse several cliques, and each clique is passed by several flows, the minimum value of $\beta_q$ can quickly get propagated by the flow-clique signal exchange.

### Efficiency Enhancement

To further utilize the available resource in the network when flow rates have been allocated according to the distributed NES algorithm, each frozen clique maintains an additional self region scaler $\beta_{q_s}$ other than the global scaler set $B_q$. $\beta_{q_s}$ adapts based on the capacity information of the clique itself. An additional self price $\mu_{q_s}$ of clique $q$ is created, which updates based on $\beta_{q_s}$.

To determine which price to use, a flow $f$ carries a flow state $h_f$ and a clique state recorder $h_{f_q}$. Both of them have two states: active and frozen. $h_{f_q}$ is set by the cliques it passes:

$$h_{f_q} = \begin{cases} A & \text{active, if } \exists q \in Q(f), h_q = A \\ O & \text{frozen, otherwise.} \end{cases}$$

After the flow reaches its destination, $h_{f_q}$ is feedback to the source node to set $h_f = h_{f_q}$. If $h_f$ is active, flow $f$ uses the normal price $\mu_q$ of clique $q$; otherwise, it will use the self price $\mu_{q_s}$ to calculate its flow price $\lambda_f$. In this way, a flow whose traversed cliques are all frozen can further increase its rate to utilize the available resources of these cliques. Table. VI.3 describes the algorithm for the distributed NES with efficiency enhancement.

Table VI.3: Distributed Implementation of NESEE algorithm

**Distributed NESEE Rate Adaptation**

1) *Initialization*
$\forall q \in Q, \beta_q = 1, id_q = q, t_q = 0, h_q = A.$
$\forall f \in F, \beta_f = 1, id_f = -1, t_f = 0, h_f = A, h_{f_q} = O.$

2) *Flow-Clique signal exchange*
When a flow $f$ passes clique $q$,
if $id_f != id_q \&\& \beta_q < \beta_f || id_f == id_q \&\& t_f < t_q$,
$B_f = B_q$.
if $id_f != id_q \&\& \beta_f < \beta_q || id_f == id_q \&\& h_q = O \&\& t_q < t_f$,
$B_q = B_f, h_q = O$.

3) *Price update*
Clique $q$ updates its price
if $h_q == O$ and $\sum_{f:E(f)\cap V(q)\neq\emptyset} x_f \cdot R_{qf} > \beta_q C_q$,
$\mu_{q_s}(t+1) = [\mu_{q_s}(t) - \gamma(\beta_{q_s} C_q - \sum_{f:E(f)\cap V(q)\neq\emptyset} x_f \cdot R_{qf})]^+$.
else, $\mu_q(t+1) = [\mu_q(t) - \gamma(\beta_q C_q - \sum_{f:E(f)\cap V(q)\neq\emptyset} x_f \cdot R_{qf})]^+$,
$\mu_{q_s} = \mu_q$.
Flow $f$ updates its price
if $h_f = A, \lambda_f(t+1) = \sum_{q:E(f)\cap V(q)\neq\emptyset} R_{qf}\mu_q(t+1)$
else, $\lambda_f(t+1) = \sum_{q:E(f)\cap V(q)\neq\emptyset} R_{qf}\mu_{q_s}(t+1)$

4) *Clique scaler update*
if $h_q == O$, $\beta_{q_s}(t+1) = \beta_{q_s}(t) - \alpha\gamma(\beta_{q_s}(t)C_q - \tilde{C}_q)$
if $\beta_{q_s} < \beta_q$, then $h_q = A$.
if $h_q == A$,
$\beta_q(t+1) = \beta_q(t) - \alpha\gamma(\beta_q(t)C_q - \tilde{C}_q), h_{f_q} = A$.

5) *Flow adaptation*
$\forall f \in F, x_f(t+1) = 1/\lambda_f(t+1), h_f = h_{f_q}, h_{f_q} = O.$

## Simulation Evaluation

### Simulation Setup

We evaluate our new fairness models and the associated resource allocation algorithms using ns-2. All the simulation uses the IEEE 802.11 protocol with RTS/CTS enabled. The wireless network has a channel capacity of $1Mbps$ with the two-ray ground reflection propagation model, and the routing algorithm selects the shortest path. Both the transmission range and interference range are 250m. The achievable clique capacity $\tilde{C}_q$ is estimated by aggregating the achievable bandwidth of all the wireless links [103] the clique contains.

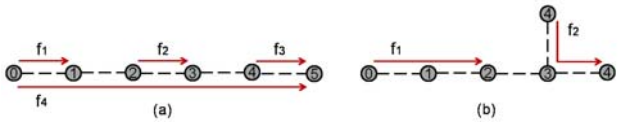


Figure VI.4: Simulation Scenarios

We consider two wireless network topologies as shown in Fig. VI.4. The following aspects of the algorithms are evaluated:

- Throughput. It is the aggregated rate of all the flows in the network.
- Fairness. To evaluate the fairness model of IEEE 802.11 with non-convex capacity region, we compare it with the fairness model of the same network

under ideal scheduling where a convex capacity region can be achieved. In particular, we define the normalized throughput of a flow $\bar{x}_f$ as $\frac{x_f}{\hat{x}_f}$, where $x_f$ is the converged rate, and $\hat{x}_f$ is its ideal rate under the convex hull [131]. Then we leverage Jain's fairness index defined over the normalized throughput $J = \frac{(\sum_{f=1}^{F} \bar{x}_f)^2}{n \sum_{f=1}^{n} \bar{x}_f^2}$, $J \in [0, 1]$ to characterize our fairness model in comparison with the ideal one. A larger $J$ indicates a closer approximation.

### Convergence

Compared with the price-based algorithm for convex capacity regions [131], the convergence speed of our algorithms is relatively slower. As our algorithms take the additional step of setting the optimal region scalers, they involve extra processing and space overhead. We now show the convergence behavior of the four fairness resource allocation algorithms in the topology of Fig. VI.4 (a), without flow $f_4$. The instantaneous flow rates are shown in Fig. VI.5. The flow rates quickly converge in each algorithm. We observe that 1) in the NES algorithms, the flow rates of $f_1$ and $f_3$ are the same, both twice of that of $f_2$; 2) in the NESEE algorithms, the rate of $f_3$ further increases with the rates of $f_1$ and $f_2$ decrease for certain amount. 3) The NESEE algorithms converge relatively slower than the NES algorithms, and the distributed NESEE experiences more oscillation than the centralized NESEE.

The reason behind the observations can be intuitively explained as follows. Considering the link contentions, $f_1, f_2$ are in one clique $q_1$, and $f_2, f_3$ are in

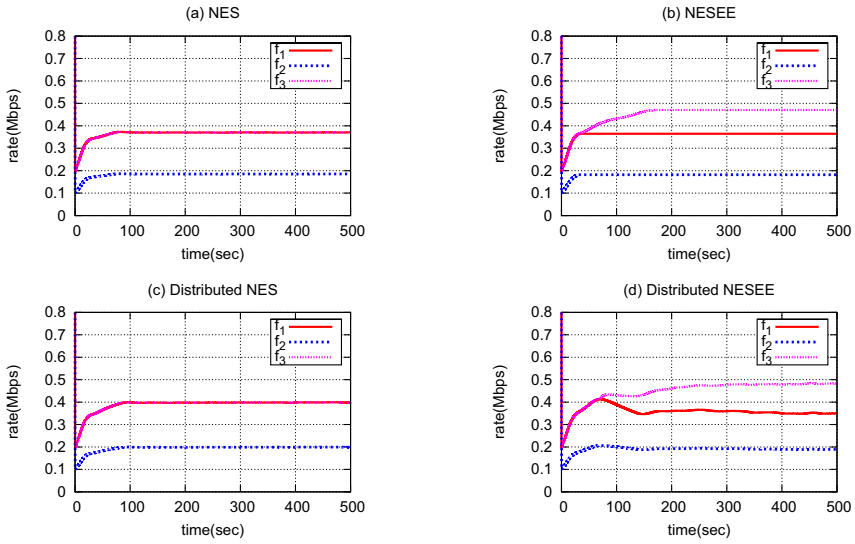


Figure VI.5: Flow Rate Convergence of a Static Network

another clique $q_2$. In the NES algorithms, both cliques will have the same capacity scaler. As both cliques also contain the same number of active links, they will have the same clique price, which leads to the same rates for $f_1$ and $f_3$. $f_2$ passes both cliques, so its price is the sum of the two clique prices, leading to a flow rate half of $f_1$ or $f_3$. According to Fig. VI.2, both $q_1$ and $q_2$ form non-convex regions. $f_2$ is affected by the hidden terminal problem, and $f_1$ is further affected by the hidden terminal problem. This results in asymmetric bandwidths, with $q_2$ viewing larger capacity than $q_1$. In the NES algorithms, the capacity scaler is constrained by the capacity of $q_1$. While in the NESEE algorithms, $f_3$ utilizes the remaining capacity of $q_2$ when the network resource has been allocated by the NES algorithms, as it is independent of $q_1$. Thus, NESEE takes a little longer time to converge. With the rate increase of $f_3$, $q_1$ views fewer capacity due to

the hidden terminal problem. This causes the rate changes of $f_2$ and $f_3$. When the distributed NESEE keeps track of the available capacities of the time-varying channels in the wireless network, the flows experience more oscillation than the centralized NESEE.

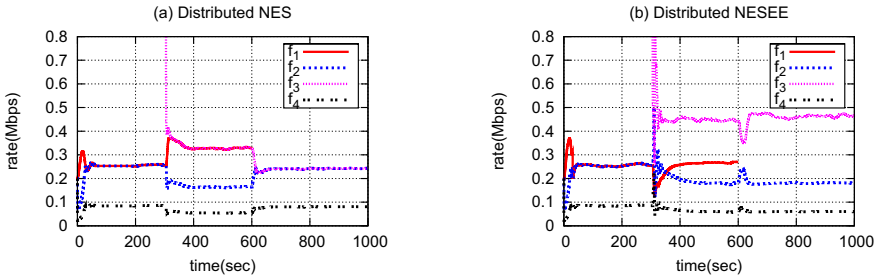


Figure VI.6: Flow Rate Convergence of a Dynamic Network

Next, we test the convergence of our distributed algorithms under network dynamics. We use the topology of Fig. VI.4 (a). Flow $f_1, f_2, f_4$ joins the network at the beginning, $f_3$ joins at $200$ seconds, and later $f_1$ leaves the network at $400$ seconds. We observed that 1) The flow rates quickly re-converge each time after network dynamic occurs as shown in Fig. VI.6. 2) Flow $f_4$ always converges faster than other flows. 3) The distributed NESEE takes longer time to converge when all the flows are in the network. For 2), $f_4$ has more hops, which provide it with a broader view and more frequent update of the region scaler in the network. For 3), more flows lead to longer convergence time considering the redistribution of the slack network resource.

Comparison Studies

To evaluate the optimality of our fairness algorithms, we perform a series of comparison studies with the topologies in Fig. VI.4. Note that we do not use flow $f_4$ in Fig. VI.4 (a).

**Centralized NES and NESEE**

Table VI.4: Performance Metrics for CNES & CNESEE

| **Topology** | **Fairness Model** | **Fairness Index** | **Throughput (Mbps)** |
|---|---|---|---|
| Fig. VI.4 (a) | CNES | 1 | 0.984 |
| | CNESEE | 0.985 | 1.017 |
| | DNES | 1 | 0.925 |
| | DNESEE | 0.978 | 1.018 |
| | TCP | 0.333 | 0.663 |
| Fig. VI.4 (b) | CNES | 1 | 0.436 |
| | CNESEE | 0.898 | 0.548 |
| | DNES | 1 | 0.440 |
| | DNESEE | 0.784 | 0.496 |
| | TCP | 0.502 | 0.308 |

Comparing the performance metrics of Centralized NES (CNES) and Centralized NESEE (CNESEE) as in Tab. VI.4, the fairness indices of the CNES in both topologies are 1, showing that the CNES always follows the ideal fair allocation of its convex counterpart. With the efficiency enhancement, the throughput achieved by the CNESEE is higher than CNES, which in turn costs lower fairness indices.

**Centralized and Distributed**

Next we compare the performance of the centralized the distributed algorithms. The Distrusted NES (DNES) perfectly follows the fair allocation as the CNES, while the achieved throughput varies somewhat considering the fluctuation of the wireless network. The fairness index of the Distributed NESEE (DNESEE) is smaller than the CNESEE, as in the distributed algorithm, the distribution of the remaining capacity will lead to the rate increase of certain flows, which can decrease the previous rate allocation of some other flows. This can also cause relatively smaller improvement of the throughput as in Fig. VI.4 (b).

**Distributed and TCP**

We further compare the performance of TCP with our distributed algorithms. We use standard TCP on IEEE 802.11, so the only difference from our algorithm is the rate allocation. Both the fairness index and throughput are much lower than those of our algorithms. In either topology, only one TCP flow occupies the network ($f_3$ in Fig. VI.4 (a), and $f_2$ in Fig. VI.4 (b)) while all the others get starved. This is caused by the hidden terminal problem. With the severe asymmetry, TCP also hardly achieves high throughput.

Random Networks

We finally study the applicability of the distributed fairness algorithms in random topologies. We simulate 5 topologies. Each topology has 25 nodes randomly deployed over a $1000m \times 1000m$ area, with 5 flows randomly generated.

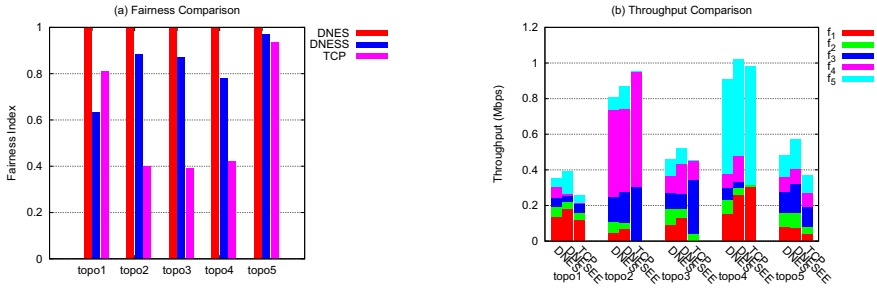


Figure VI.7: Performance Metrics in Random Topologies

Fig. VI.7 displays the fairness indices and the throughput of the flows using the distributed algorithms and TCP. We observe that the distributed NES always allocates the network resource fairly. The distributed NESEE improves the overall throughput from distributed NES, but with the cost of relatively lower fairness indices. Comparing with TCP, our distributed NESEE has larger fairness index and larger throughput most of the time. When the distributed NESEE has a lower fairness index than TCP as in topology 1, it in turn achieves higher throughput; vice versa in topology 2. These results are consistent with those from the comparison studies.

# CHAPTER VII

# UNDERSTANDING THE IMPACT OF NEIGHBORHOOD INFORMATION ON END-TO-END FAIRNESS IN MULTI-HOP WIRELESS NETWORK

## Introduction

A multi-hop wireless network consists of a collection of wireless nodes without a fixed infrastructure. Nodes within the transmission range of each other communicate directly, while nodes that are far away communicate via relays of intermediate nodes. In such a network, each *end-to-end flow* traverses multiple hops from a source to a destination. It is important to allocate the limited bandwidth resources in multi-hop wireless networks to contending flows, in a way that is both *efficient* with respect to resource utilization, and *fair* across contending multi-hop flows.

Resource allocation for end-to-end flows is an extremely challenging problem in multi-hop wireless networks. The key challenge comes from the complicated wireless resource contention model, namely location-dependent contention and spatial channel reuse. Depending on the scheduling algorithm, transmission of unit flow along a link may block the transmissions of different sets of flows, thus virtually uses different amounts of resources. In recent years, significant progress has been made on this topic. Existing works on this problem largely fall into two categories: 1) Theoretical methods for joint optimization of scheduling

and rate allocation [29] [66] [109] [131], where the scope of resource sharing regions is rigorously defined by the schedulability of link-level flows. Though these methods provide a theoretically sound optimal and fair resource allocation solution, they can hardly be implemented in a cost-efficient way due to the intrinsic complexity in the optimal multi-hop scheduling; 2) Heuristic rate allocation solutions [129] [94], which simplify the scope of resource sharing regions using different neighborhood models and partially rely on the underlying packet scheduling (*e.g.,* IEEE 802.11) to resolve the resource contention among link-level flows. These works are practical for implementation, but their fairness properties are usually only evaluated on special topologies, as their concept of fairness is not well-defined on general topologies. To date, the following essential question on end-to-end fairness still remains unanswered: *how well these practical heuristic neighborhood-aware rate allocation solutions approximate the optimal point defined in the theoretical resource allocation framework*.

The objective of this chapter is to *evaluate the impact of different neighborhood models on end-to-end fairness*. Towards this goal, we first establish a baseline fairness model using the optimal resource allocation framework [131]. In this framework, the optimization objective is defined as maximizing the aggregated utility of all flows and the resource constraints are characterized by the necessary condition of schedulability defined using maximal cliques in the contention graph of the wireless network. It is shown [50] that this optimal resource allocation framework can achieve different fairness models, when different utility functions

are specified. In this work, we consider the *proportional fairness* model as our baseline fairness model, which can be achieved under logarithm utility function.

To evaluate different heuristic rate allocation solutions that use various neighborhood models to approximate the resource sharing regions, we consider the price-based model which provides a distributed solution to resource allocation. In the price-based model, price represents the penalty of the resource usage incurred by unit flow (*i.e.*, congestion penalty). The price of a flow is the aggregated price of the links it traverses. And the link price is the sum of the prices of all the maximal cliques (scope of resource sharing regions), to which it belongs. The rate of a flow is then determined by its price so that its net profit, which is the difference between its utility and the cost its pays, is maximized. Obviously, when the resource sharing regions are approximated by different neighborhood models in the heuristic solutions, link prices will be manifested as different values. We introduce a normalized fairness index to quantify the deviations of these heuristic neighborhood-aware solutions from baseline fairness models,

Six different neighborhood models are evaluated in this chapter: 1) clique approximation, which provides clique price estimation for the optimal clique-based solution in IEEE 802.11-style networks via the achievable capacity measurement; 2) asymmetric neighborhood (1-hop and 2-hop, respectively), which approximates the resource sharing region using link-centered neighborhood with different scopes, as in the work of [95]; 3) symmetric neighborhood (1-hop and 2-hop, respectively), which improves the asymmetric neighborhood model by ensuring the maximal neighborhood knowledge symmetrically shared by the links

within the neighborhood; and 4) single link, where only the congestion penalty observed on its own link is considered, as in traditional TCP.

This chapter makes the following contributions to the field. First, it presents a price-based fairness model and a normalized fairness index model, where heuristic rate allocation solutions with different neighborhood information can be compared within a common framework. With the advance of the wireless communication technology, medium access and routing protocols, the solution space of the fairness problem may continue to evolve, but its nature of fairness resource allocation remains unchanged. This fairness theoretical framework can effectively decouple the "core" of the problem and its other components, so that the basic problem formulation and its solving methodology survive. Second, extensive simulation study is conducted over a variety of carefully designed and random topologies. Our study makes two significant observations: 1) symmetric knowledge on the construction of a neighborhood is important in achieving fairness. Such a knowledge has not been considered in any of the existing neighborhood-aware rate allocation algorithms; 2) while using 1-hop neighborhood information brings noticeable gain in fairness compared with link-only solutions, the knowledge of 2-hop neighborhood information does not bring additional benefit from 1-hop neighborhood information.

The rest of this chapter is organized as follows. We first introduce the price-based fairness model in multi-hop wireless networks in Sec. VII. Then the neighborhood models to be evaluated and the fairness index model are presented in Sec. VII. Finally we present our evaluation results in Sec. VII.

## System Models

### Network Model

We model a multi-hop wireless network as a bidirectional graph $G = (V, E)$, where $V$ is the set of nodes. $E \subseteq 2^V$ denotes the set of wireless links. A wireless link $e \in E$ is represented by its end nodes $i$ and $j$, i.e., $e = (i, j)$.

Let us consider a set of *end-to-end flows*, denoted as $F$. Each flow $f \in F$ has a rate of $x_f$. We use $\boldsymbol{x} = \{x_f, f \in F\}$ to denote the flow vector. Flow $f$ goes through multiple hops in the network, passing a set of wireless links $E(f)$. A single-hop data transmission in the flow $f$ along a particular wireless link is referred to as a *subflow* of $f$. Obviously, there may exist multiple subflows along the same wireless link. We use the notation $L(L \subseteq E)$ to represent a set of wireless links in $G$, such that each of the wireless links in $L$ carries at least one subflow. A link $l \in L$ is called an active link. The rate vector of active links is denoted as $\boldsymbol{y} = \{y_l, l \in L\}$, where $y_l = \sum_{f:l \in E(f)} x_f$.

### Resource Sharing and Contention Model

Flows in the multi-hop wireless network contend for wireless resource in a location-dependent manner. In this chapter, we consider the protocol model [36], where two subflows contend with each other if either the source or destination of one subflow is within the interference range of the source or destination of the

other. Formally, a wireless link contention graph is defined as $G_c = (V_c, E_c)$, where the vertex set corresponds to the wireless links $V_c = L$, and there exists an edge between two vertices if the subflows along these two wireless links contend with each other.

The key challenge of optimal resource allocation in multi-hop wireless network comes from the modeling of wireless resource capacity region. A wireline link is simply a resource sharing unit, and thus represents an independent dimension for the resource capacity region. However, in the case of multi-hop wireless networks, channel resource is shared in a location-dependent way. Thus the achievable capacity of a wireless link is interrelated with other wireless links in its vicinity. As a matter of fact, characterizing the channel resource sharing region is related to the issue of schedulability, *i.e.*, whether rate vector $\boldsymbol{y}$ is schedulable given the channel capacity and the network topology. However, it is known [45] that establishing the sufficient and necessary condition of schedulability in a wireless network involves finding the independence number of a graph, which is an NP-hard problem.

Several approximation models of resource sharing units are proposed in the existing literature [45, 131, 53, 3]. In this chapter, we adopt the maximal clique approximation model [131] as the baseline fairness model for two of its properties: 1) better approximation factor compared with the interference set models [53, 3] and 2) similarity to the neighborhood approximation models which are widely used in the existing rate allocation heuristics [129, 95]. In this clique-based wireless resource allocation framework, the resource sharing regions are characterized

by *maximal cliques*[1] in the *wireless link contention graph* of the network. In a wireless link contention graph, the vertices in a maximal clique represent a maximal resource sharing region. They are mutually contending wireless links, among which at most one subflow may transmit at any given time. Formally, let $Q$ be the set of all maximal cliques in $G_c$. For a maximal clique $q$ in the wireless link contention graph $G_c$, $V(q) \subseteq L$ is the set of its vertices. The resource constraint under this clique model is formulated based on the necessary condition of schedulability, as follows:

$$\forall q \in Q, \sum_{l \in V(q)} y_l \leq C_q \tag{VII.1}$$

where $C_q$ is the schedulable region of clique $q$.

## Baseline End-to-end Fairness Model

### Optimal Resource Allocation

We first briefly review the optimal resource allocation framework under clique model and its relationship to end-to-end fairness [50].

Let the utility function for an end-to-end flow $f \in F$ be $U_f(x_f)$, with its associated end users at rate $x_f$. This function is increasing, strictly concave and

[1] *In a graph, a complete subgraph is referred to as a* clique. *A* maximal clique *is defined as a clique that is not contained in any other cliques.*

continuously differentiable. The problem of optimal rate allocation can be formulated as:

$$\mathbf{P}: \quad \textbf{maximize} \quad \sum_{f \in F} U_f(x_f) \tag{VII.2}$$

$$\textbf{subject to} \quad \boldsymbol{R} \cdot \boldsymbol{x} \leq \boldsymbol{C} \tag{VII.3}$$

$$\boldsymbol{x} \geq 0 \tag{VII.4}$$

The objective function (VII.2) is to maximize the aggregated utility of all flows. In inequality (VII.3), the matrix $\boldsymbol{R}$ represents the "resource usage pattern" of each flow. If we treat a maximal clique as an independent resource, $\boldsymbol{R} = \{R_{qf}\}$ is the clique-flow matrix. Note that $\sum_{f \in F} R_{qf} x_f = \sum_{l \in V(q)} y_l$, the constraint comes from the schedulability condition (VII.1). Further, decomposing $\boldsymbol{R} = R^A_{ql} \cdot R^P_{lf}$, where $R^A_{ql}$ can be understood as the scheduling matrix and defined as follows:

$$R^A_{ql} = \begin{cases} 1 & \text{if } l \text{ belongs to } q \\ 0 & \text{otherwise} \end{cases}$$

and $R^P_{lf}$ is the routing matrix as

$$R^P_{lf} = \begin{cases} 1 & \text{if } f \text{ passes through } l \\ 0 & \text{otherwise} \end{cases}$$

When a network contains only link-level flows, $R^P$ becomes an identity matrix. $R_{qf} = |V(q) \cap E(f)|$ represents the number of subflows that flow $f$ has in the clique $q$. Let $\boldsymbol{C} = \{C_q, q \in Q\}$ be the vector of achievable channel capacities in each of the cliques. If the channel capacity of clique $q$ under ideal scheduling is $C$, $C_q \leq C$. This optimization constraint characterizes the schedulability condition of wireless channel resource.

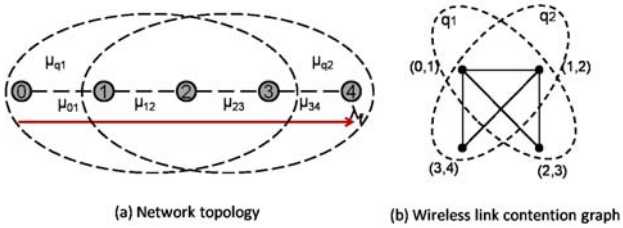


Figure VII.1: Wireless Flow Price Model: An Example

We present an example to illustrate the concepts and notations defined so far. Fig.VII.1(a) shows the topology of the network, with its ongoing flows. The corresponding wireless link contention graph is shown in Fig.VII.1(b). We assume that the interference range is the same as transmission range in this chapter. In this example, there is one flow $f = \{0 \rightarrow 4\}$. As such, there are two maximal cliques in the contention graph: $q_1 = \{(0,1),(1,2),(2,3)\}$ and $q_2 = \{(1,2),(2,3),(3,4)\}$. The aggregated rate $y_{ij}$ of all subflows along a wireless link $\{i,j\}$ equals to the rate of the flow. That is, $y_{ij} = x$. In each clique, the aggregated rate may not

exceed the corresponding channel capacity:

$$\begin{aligned} y_{01} + y_{12} + y_{23} &\leq C_1 \\ y_{12} + y_{23} + y_{34} &\leq C_2 \end{aligned}$$

where

$$R^A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \text{ and } R^P = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

$\mathbf{P}$ achieves Pareto optimality with respect to the resource utilization and end-to-end fairness when appropriate utility functions are specified and when the feasible region of $\boldsymbol{C}$ is convex and compact [131]. In , *proportional fairness* can be achieved when the utility function takes the logarithm function form $U_f(x_f) = \log x_f$.

Price-based Fairness Model

The optimal resource allocation in terms of both utilization and fairness is achieved by solving the dual problem $\mathbf{D}$ of $\mathbf{P}$ as follows:

$$\mathbf{D}: \quad \min_{\boldsymbol{\mu} \geq 0} D(\boldsymbol{\mu}) \tag{VII.5}$$

where

$$D(\boldsymbol{\mu}) = \sum_{f \in F} \max_{x_f} (U_f(x_f) - x_f \sum_{q: E(f) \cap V(q) \neq \emptyset} \mu_q R_{qf}) + \sum_{q \in Q} \mu_q C_q \quad \text{(VII.6)}$$

a vector of Lagrange multipliers $\boldsymbol{\mu} = (\mu_q, q \in Q)$ is called price. It is interpreted as the cost of a unit flow accessing the resource sharing region characterized by the maximal clique $q$. In other words, $\mu_q$ is the *price* of clique $q$.

The dual problem $\mathbf{D}$ is solved via a gradient projection algorithm [131]. This leads to an iterative algorithm that can be executed in a distributed way to achieve the optimal solution of $\mathbf{P}$. In particular, the price $\mu_q$ at clique $q$ is adjusted as follows:

$$\mu_q(t+1) = [\mu_q(t) - \gamma(C_q - \sum_{q: E(f) \cap V(q) \neq \emptyset} x_f \cdot R_{qf})]^+ \quad \text{(VII.7)}$$

The optimal flow rate for $f$ can be adjusted at the source of the flow so that its net benefit (difference between utility and cost) is maximized:

$$\textbf{maximize } U_f(x_f) - \lambda_f x_f \quad \text{(VII.8)}$$

The optimal flow rate for $f$ can be derived from

$$U_f'(x_f) - \lambda_f = 0 \tag{VII.9}$$

Here $\lambda_f$ is the price of a flow $f$ and can be interpreted in the following two alternative ways.

$$\lambda_f = \sum_{q:E(f)\cap V(q)\neq\emptyset} R_{qf}\mu_q \tag{VII.10}$$

$$= \sum_{l:l\in E(f)} \mu_l = \sum_{l:l\in E(f)} \sum_{q:l\in V(q)} \mu_q \tag{VII.11}$$

In Eq. (VII.10), flow $f$ needs to pay for all the resource sharing regions (*i.e.*, maximal cliques in our baseline model) it uses. For each maximal clique, the cost is the product of the number of wireless links that $f$ traverses in this region and its price. In the second representation (VII.11), flow price is the aggregated price of all wireless links it passes. For each wireless link, its price is the aggregated price of all the regions that it belongs to.

As shown in Fig. VII.1, let the prices of the two cliques be $\mu_1$ and $\mu_2$. The price of flow $f$ that traverses these two cliques is given by $\lambda_f = 3\mu_1 + 3\mu_2$, which is the sum of the product of the number of subflows of $f$ in each clique and the price of this clique. Alternatively, the price can also be written as $\lambda_f = \mu_1 + (\mu_1 + \mu_2) + (\mu_1 + \mu_2) + \mu_2$, which is the sum of the prices of its subflows.

Discussion

This pricing model reflects the fundamental concept of fairness in a wireless network. Essentially, if two subflows are within the same resource sharing region, they should share the same price as the cost of using this region. If a link belongs to multiple regions, all the region prices should be aggregated into the link price. As the transmission along this link would influence all the links in these regions, it should receive the penalty from all. Obviously, when the resource sharing regions are approximated by different neighborhood models in heuristic solutions [129, 95], link prices will be manifested as different values, thus affect the fairness properties of end-to-end flow rate allocation.

## Neighborhood Models

In our baseline fair resource allocation model (which we denote as *OPT*), a maximal clique is regarded as an *independent resource sharing region* with capacity $C$. While theoretically sound, this model has two limitations that need to be addressed before it can be applied as a practical solution of wireless rate allocation. First, Eq. (VII.1) only gives an upper bound on the rate allocations to the wireless links. In practice, however, such a bound may not be tight, especially with carrier-sensing-multiple-access-based wireless networks such as IEEE 802.11. In this case, the achievable channel capacity $C_q$ needs to be estimated at each region $q$. Second, calculating the price of a maximal clique involves information exchange within the 3-hop neighborhood [131], which is inevitably expensive. To

limit the communication overhead, different neighborhood models can be used to approximate the maximal clique construction. The goal of this chapter is to examine the impact of various approximation models on fairness. The optimal rate allocation problem can still be formulated as $\boldsymbol{P}$, while the matrix $\boldsymbol{R}$ represents the "resource usage pattern" of each flow in the approximation models, and $\boldsymbol{C}$ is the vector of achievable channel capacities observed by different neighborhoods. A wireless link neighborhood graph $G_n = (V_n, E_n)$ is used to describe the views of the resource sharing regions in the different models. $V_n = L$ is the vertex set corresponds to the active wireless links, and $E_n$ is the edge set in which an edge between two vertices exists if the two links are considered contend with each other in the neighborhood model. Specially, when the resource sharing regions are the same as *OPT*, the neighborhood graph is identical to the link contention graph. In particular, we consider the following approximation models, *LNK, ANB-1, ANB-2, SNB-1, SNB-2, CLQ*.

In the neighborhood approximation models, let $N$ be the set of all the neighborhoods in $G_n$, where $n \in N$ represents a neighborhood in the model. The capacity of the neighborhood $n$ is defined as $\boldsymbol{C}_n = \{C_n, n \in N\}$. In particular, it uses the approach presented in [103] to measure the achievable bandwidth ($C_l$) of each wireless link $l$ based on its historical data transmission results. And $C_n$ aggregates the available bandwidth within the neighborhood, $C_n = \sum_{l \in n} C_l, (l \in L)$. The rate allocation methods of these approximation models follow the similar price-based approach as in the baseline model. Specifically, let $\mu_n$ be the neighborhood

price, it is updated based on the following equation

$$\mu_n(t+1) = [\mu_n(t) - \gamma(C_n - \sum_{f \in F} x_f \cdot R_{nf})]^+ \tag{VII.12}$$

where $R_{nf} = R^A_{nl} \cdot R^P_{lf}$ is the number of subflows of $f$ in the neighborhood $n$. The link price $\mu_l = \sum_{l \in V(n)} \mu_n$ is the aggregated price of all the neighborhoods it belongs to. The corresponding flow rate for $f$ can still be derived from Eq. (VII.9), where $\lambda_f = \sum_{l:l \in E(f)} = \sum_{l:l \in E(f)} \sum_{n:l \in V(n)} \mu_n$. The following lemma formally proves Eq. (VII.12) is globally and asymptotically stable, which converges to a unique point under convex capacity regions.

Let $\mathcal{V}$ be defined as

$$\mathcal{V}(\mu) = \sum_{f \in F} log(\sum_{n \in f} \mu_n) - \sum_{n \in N} \int_0^{\mu_n} p_n(\eta) d\eta$$

and for $n \in N$, $p_n(0) = 0$ and $p_n(\eta), \eta \geq 0$, is a continuous, strictly increasing function of $\mu$.

**Lemma 1** *$\mathcal{V}$ is a strictly concave function and a Lyapunov function for the system of equations (VII.12) and (VII.9). The unique value $\mu$ maximizes $\mathcal{V}$ and is also a stable point of the system where all trajectories converge.*

*Proof.* We first prove the strict concavity of $\mathcal{V}$. The first term $\sum_{f \in F} log(\sum_{n \in f} \mu_n)$ is strictly concave and its derivative to $\mu_n$ is strictly decreasing considering the properties of $log$ function. The second term is convex. Let $p_n(\eta) = \tilde{C}_n \eta / (\eta + \varepsilon)$,

for $n \in N, \eta > 0$. $\tilde{C}$ is the actual achievable capacity, with $\tilde{C}_n \leq C_n$ and defined as

$$\tilde{C}_n = \max_{\mu_n} \sum_{f \in n} x_f(\mu_n, \cdot) \tag{VII.13}$$

$p_n(\eta)$ is a continuous and strictly increasing function of $\eta$, and it arbitrarily closely approximates $\tilde{C}_n$ for a small positive $\varepsilon$. Thus $\mathcal{V}$ is strictly concave on $\mu > 0$ with an unique interior maximum $\mu$. By setting $\mathcal{V}'(\mu) = 0$.

$$\begin{aligned} \frac{\partial}{\partial \mu_n}\mathcal{V}(\mu) &= \sum_{f:n\in f} \frac{1}{\sum_{k\in f}\mu_k} - p_n(\mu_n) \\ &= \sum_{f:n\in f} \frac{1}{\lambda_f} - p_n(\mu_n) \\ &\geq \sum_{f:n\in f} \frac{1}{\lambda_f} - C_n \end{aligned} \tag{VII.14}$$

As $\sum_{n:E(f)\cap V(n)\neq\emptyset} x_f \cdot R_{nf} = \sum_{f:n\in f} x_f$ and with $x_f = 1/\lambda_f$ from Eq. (VII.9),

$$\begin{aligned} \frac{d}{dt}\mathcal{V}(\mu(t)) &= \sum_{n\in N} \frac{\partial \mathcal{V}}{\partial \mu_n} \cdot \frac{d}{dt}\mu_n(t) \\ &\geq \gamma \sum_{n\in N} \Big( \sum_{n:E(f)\cap V(n)\neq\emptyset} x_f \cdot R_{nf} - C_n \Big)^2 \end{aligned}$$

establishes that $\mathcal{V}$ is strictly increasing with $t$ unless $\mu_(t) = \mu$, where the unique value $\mu$ maximizes $\mathcal{V}$. Thus function $\mathcal{V}$ is a Lyapunov function for the system (VII.12) and (VII.9), and the lemma follows. □

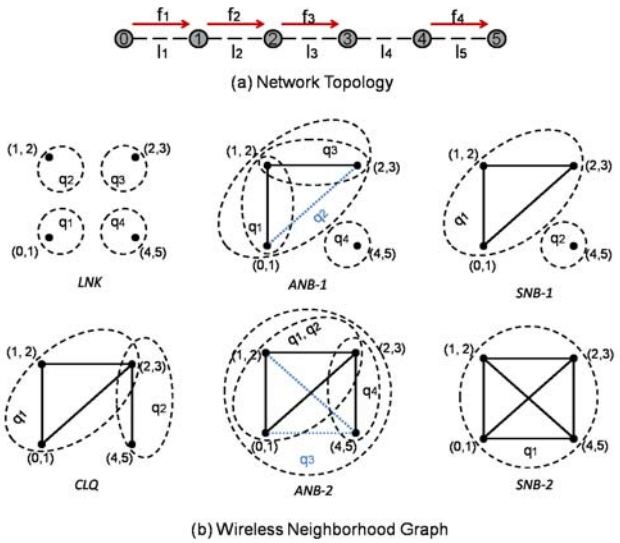


Figure VII.2: A Special Scenario

We use a special scenario as an example in Fig. VII.2 (a) to illustrate the different neighborhood models. Since the topology contains only per-hop flows, the routing matrix $\boldsymbol{R}^P$ is a $4 \times 4$ identity matrix. We can solely consider the scheduling matrix $\boldsymbol{R}^A$. Fig. VII.2 (b) shows the wireless neighborhood graphes of the different neighborhood models.

Definition of the approximation models

We first define the basic concepts of the approximation models.

**Link only (LNK)**

In this model, a link only considers itself as a resource sharing region, thus the scheduling matrix $\boldsymbol{R}^A$ is a $4 \times 4$ identity matrix, and $\boldsymbol{C} = \{C_l\}$ is the achievable capacity of each link. Its neighborhood graph is just a collection of isolated nodes as in Fig. VII.2 (b). The link capacities rely entirely on the underlying (possibly suboptimal) MAC scheduling protocol (*i.e.*, IEEE 802.11 in our study) to resolve the resource allocation among wireless links. This model well characterizes the TCP rate control mechanism.

**Asymmetric neighborhood: 1-hop (ANB-1) and 2-hop (ANB-2)**

Under this model, a resource sharing region is centered at each wireless link and includes its $n$-hop neighboring links ($n = 1$ or 2).

$$\boldsymbol{R}^A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

in the 1-hop and 2-hop neighboring models separately.

This model aligns well with the existing heuristic neighborhood-aware rate allocation solutions [129, 95]. However, as these heuristic solutions only pass the signals of a congested link to all the contention regions that contain the link, the constructed asymmetric neighborhood regions cannot be reflected directly by the price-based approach.

**Symmetric neighborhood: 1-hop (SNB-1) and 2-hop (SNB-2)**

As the price-based approach is driven by the net benefit instead of certain congested links in the heuristic neighborhood-aware rate allocation solutions, this symmetric model approximates those heuristic solutions with the price-based method. It ensures that the maximal neighborhood knowledge is symmetrically shared by the links within the neighborhood. Specifically, each link will first generate its own neighborhood as a neighborhood candidate. If a neighborhood of a link is contained by the neighborhood of another link, it will adopt the larger neighborhood. Thus, in the scheduling matrix $\boldsymbol{R}^A = \{R_{nl}\}$, $n$ is a neighborhood that is not contained by any other neighborhoods.

$$\boldsymbol{R}^A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

in the 1-hop and 2-hop neighborhood models separately. In the 1-hop neighborhood, as the neighborhoods of links 1 and 3 are contained by that of link 2, both of

them are removed. Similarly in the 2-hop neighborhood, the neighborhood graph contains only one large neighborhood overshadowing the other three smaller ones.

**Clique approximation (CLQ)**

This approximation model adopts the same definition of resource sharing region (*i.e.*, maximal clique) as the baseline model (OPT), with the scheduling matrix of

$$R_a^A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

and the contention graph is shown in Fig. VII.2 (b).

The difference of this model from the baseline model is that it does not assume an ideal scheduling algorithm which works with the rate allocation algorithm to jointly optimize the resource allocation. Instead, it performs online estimation of the region capacity $C_q$ under the IEEE 802.11 protocol. The region capacity $C_q = \sum_{l \in V(q)} C_l$.

Evaluation of the approximation models

The performance and correlation between the different models are explained here.

**LNK & SNB-1**

The result of *LNK* is topology dependent, and with no price regulation. While the prices of *SNB-1* are regulated within each neighborhood composed of 1-hop neighboring links, or the directly connected ones.

*SNB-1* does not take into account the contention between the unattached links which are still within the interference range of each other. Thus the *SNB-1* model performs the same as *LNK* in a topology where active links are at least 2 hops away from each other. For example, in Fig. VII.2, link $l_5$ considers itself as an independent neighborhood in *SNB-1* like in *LNK*.

As *SNB-1* does consider information sharing between directly connected neighbors, it can correctly find more neighbors via its one-hop neighbors. In Fig. VII.2, *SNB-1* considers $l_1$ and $l_2$ as neighbors, but *LNK* does not. Thus, the number of neighbors found by *LNK* is always greater than or equal to that of *SNB-1*, accordingly, the sizes of neighbors, considering the number of links contained by each neighbor, will not larger in *LNK* than those of *SNB-1* .

**SNB-1 & SNB-2**

Comparing *SNB-1* and *SNB-2*, the number of neighbors found by *SNB-1* is always larger than or equal to that of *CLQ*, and the number of neighbors found by *SNB-2* is always smaller than or equal to that of *CLQ*.

*SNB-1* performs the same as *CLQ* if all the flows form a connected graph. Considering a graph constructed by all the links that have flows on it in the topology, we call it a flow graph. Assuming symmetric links, in the graph, if any node can find a route from one node to any other node, it becomes a connected flow graph. As links can find their 2-hop neighbors via the 1-hop neighbors, the *SNB-1* model performs the same as the *CLQ* model. For example in Fig. VII.2 (b), *SNB-1* has the same neighborhood $q_1$ as *CLQ*. However, as long as the graph is not a connected flow graph, *SNB-1* will result in different neighborhoods from *CLQ* like $q_2$ in Fig. VII.2 (b).

*SNB-2* performs the same as *CLQ* when all the links interfere with each other. Usually *SNB-2* has a much larger neighborhood in terms of number of links contained comparing with other models. It always tends to regulate the prices of larger groups of links equally. Like in Fig. VII.2 (b), *SNB-2* considers all the links within the same neighborhood. But in a small dense region where almost all the flows may contend with each other, *SNB-2* can sometimes out-perform *CLQ*.

To sum up. *SNB-1* does not carry its full responsibility of the contention, while *SNB-2* over-emphasizes the contention in the network.

**ANB-n & SNB-n**

In the *ANB-n* models, each link has its own view of neighborhood. However, as the links do not coordinate with other links, this may cause information asymmetry. In Fig. VII.2 (b) in *ANB-1*, $q_2$ considers $l_1$ and $l_3$ are neighbors, but $l_1$ and

$l_3$ do not treat each other as a neighbor. The same problem exists in *ANB-2* when $q_3$ considers all the links are neighbors, but at least $l_1$ and $l_5$ do not see each other as neighbors. The neighborhood covering can lead to severe problems, as each contention region has its own price regulation and the price regulation line of the covered neighborhood may not overlap with that of the covering neighborhood. So the links viewing the larger neighborhood will over-use the channel if available capacity is sensed, or get over-punished when the demand in the neighborhood is high. Affected by the covering problems of the neighborhood information, *ANB-n* cannot provide reasonable results.

With *SNB-n* when all the covered neighborhoods are removed, the asymmetry problem no longer exist, which is why in the future comparison, the results of *ANB-n* may not be displayed.

## Summary

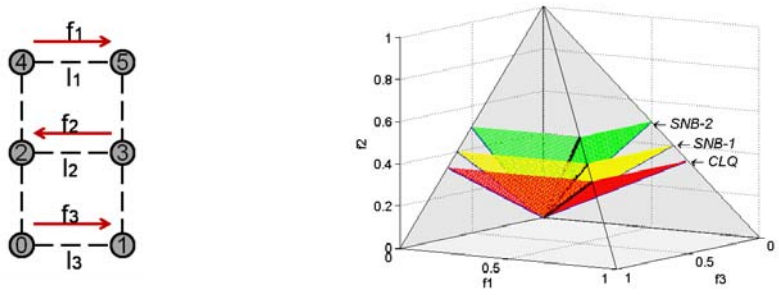


Figure VII.3: Scenario to show price regulation lines

Fig. VII.3 shows the price regulation under different neighborhood models in the ideal capacity region. In the 3-flow topology on the left, the ideal region is a pyramid. As $f_1$ and $f_3$ do not contend with each other, their capacity region is a square. While $f_1$ and $f_2$, as well as $f_2$ and $f_3$ both contend with each other, their capacity regions are isosceles right triangles under ideal scheduling. The 3 flows belong to the same neighborhood in the *SNB-2* model, each of which gets the same neighborhood price. Under the price regulation, the rate of each flow will increase equally. If viewed by flow pairs, without considering $f_3$, the rates of $f_1$ and $f_2$ will increase along the plane of $x_1 = x_2$, and without considering $f_1$, the rates of $f_2$ and $f_3$ will increase along the plane of $x_2 = x_3$. The two planes at the top in Fig. VII.3 intersect in the dark line in the middle, which is the price regulation line of the three flows in the *SNB-2* model. The regulation line starts from 0, and ends when it hits the exterior boundary of the capacity region.

In the *CLQ* model, there are two neighborhoods. As $f_2$ belongs to both of them, its price is the sum of the two neighborhood prices. Thus, with the price regulation, in the capacity region of $f_1$ and $f_2$, the rates will increase along the plane of $x_1 : x_2 = 2 : 1$. Similarly, the rates of $f_2$ and $f_3$ will increase along the plane of $x_3 : x_2 = 2 : 1$. The two planes at the bottom of Fig. VII.3 also intersect in a line in the middle, which regulates the rates of the three flows to increase from 0 to the exterior of the capacity region in the *CLQ* model.

Note that the *SNB-1* model has the regulation line between the *SNB-2* and *CLQ* models. In this example, *SNB-1* as well as *ANB-1* has the same neighborhood of *LNK* model, which follows the IEEE 802.11 scheduling. It regulates $f_1$ and $f_2$, $f_2$

and $f_3$ with $x_1 : x_2 = x_3 : x_2 = a : 1$, where $a \in (1, 2)$. This results the price regulation line between those of *SNB-2* and *CLQ* models.

## Discussion

Two fairness models are often used in network resource allocation, namely max-min fairness [8] and proportional fairness [50]. Existing works on fair wireless rate allocation [129] [95] usually use the max-min fairness model to evaluate their solutions. In this chapter, we adopt the proportional fairness model where region prices and link prices are used as the basis of fairness definition. This approach reflects the recent work on fairness [14], which states that comparing flow rates should not be used for fairness indexing in production networks. Instead, the fairness should be determined by how flows share out the cost in the network. Our fairness model is also easier to be generalized to handle the different neighborhood models.

It is also important to note that our approximation models provide an *abstraction* for the heuristic neighborhood-aware rate allocation solutions. There may not fully characterize the details of individual solutions. For example, the work of [129] actually uses a node-centered neighborhood model, where the neighborhood region includes the node itself and the nodes which can interfere with this node's signals. This creates another level of asymmetry even from our *NB1a* model, as the sender and receiver nodes have different neighborhood regions. The implementation details are also abstracted out in our model. For example,

to calculate the region price (congestion signal), different solutions have applied different methods, including using distributed queue length, channel conditions, etc, and delivered the price using the active queue management mechanism. The relationship along price-based resource allocation, active queue management, and congestion control has been extensively discussed in [108]. Our approximation models abstract out these details so that they will not introduce unnecessary noise to our study and thus allow us to focus on evaluating the impact of neighborhood information.

## Simulation Evaluation

### Simulation Setup

We implement the approximation models in ns-2. Unless explicitly mentioned, all the experiments use the following settings. RTS/CTS is enabled in the IEEE 802.11. The bandwidth of the channel is 1Mbps, and its propagation model is the two-ray ground reflection model. The transmission range and interference range are both 250m.

Evaluation Metrics

To compare the fairness achieved by the different neighborhood models and evaluate the impact of neighborhood information, we introduce the following evaluation metrics.

**Normalized Link Price**

Let the allocated rate vector in the baseline fairness model be $\boldsymbol{x}^* = \{x_f^* | f \in F\}$, and the rate vector of the neighborhood model to be evaluated as $\boldsymbol{x} = \{x_f | f \in F\}$. The normalized flow rate vector is then defined as $\bar{\boldsymbol{x}} = \{\bar{x}_f, f \in F\}$, where $\bar{x}_f = \frac{x_f}{x_f^*}$.

**Normalized Flow Rate**

Let the converged link price vector in the baseline fairness model be $\boldsymbol{\mu}_l^* = \{\mu_l^* | l \in L\}$, and the link price vector of the neighborhood model to be evaluated as $\boldsymbol{\mu}_l = \{\mu_l | l \in L\}$. The normalized flow rate vector is then defined as $\bar{\boldsymbol{\mu}}_l = \{\bar{\mu}_l, l \in L\}$, where $\bar{\mu}_l = \frac{\mu_l}{\mu_l^*}$.

**Normalized Fairness Index**

The rate-based fairness index for $\boldsymbol{x}$ is then defined as follows:

$$\mathcal{J}(\boldsymbol{x}) = \frac{(\sum_{f \in F} \bar{x}_f)^2}{|F| \sum_{f \in F} \bar{x}_f^2} \tag{VII.15}$$

where $|F|$ is the number of flows in $F$. This fairness index is bounded between 0 and 1. The higher the fairness index is, the better a rate allocation achieves fairness.

**Normalized Utility Function**

As the optimization objective is captured in the utility function, the efficiency level of the neighborhood models can be evaluated in terms of the optimization criterion of

$$\mathcal{U}(\boldsymbol{x}) = \frac{\sum_{f \in F} U_f(x_f)}{\sum_{f \in F} U_f(x_f^*)} \tag{VII.16}$$

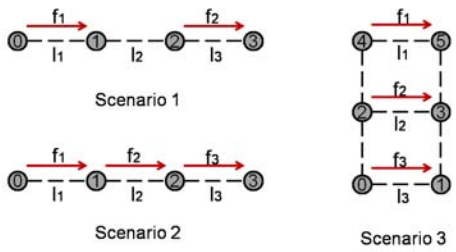


Figure VII.4: Simple Scenarios

Formulation Validation

We first experiment on the topology of Fig. VII.2. Fig. VII.5 shows that following Eq. (VII.12), the link prices and the flow rates all converge quickly in

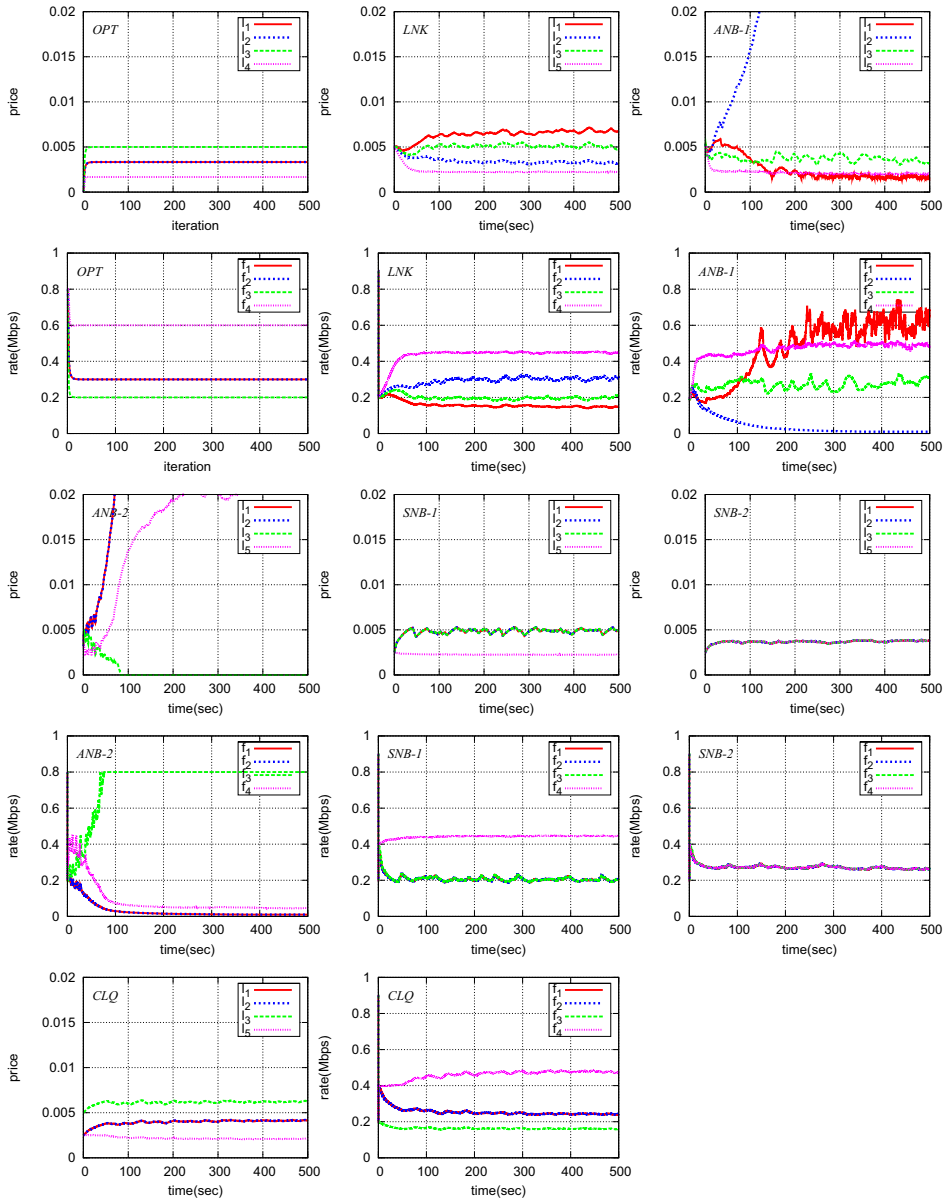


Figure VII.5: Instantaneous Price and Rate of Scenario 1

Table VII.1: Evaluation Metrics for the Special Scenario

| | LNK | ANB-1 | ANB-2 | SNB-1 | SNB-2 | CLQ |
|---|---|---|---|---|---|---|
| $\mathcal{J}(\boldsymbol{x})$ | 0.933 | 0.685 | 0.269 | 0.967 | 0.889 | 1 |
| $\mathcal{U}(\boldsymbol{x})$ | 0.919 | 0.854 | 0.634 | 0.921 | 0.929 | 0.927 |

each neighborhood model. Fig. VII.6 displays the link prices, the normalized link prices and the normalized flow rates of the models after convergence. Table. VII.1 presents the values of the normalized fairness index and utility function of the models.

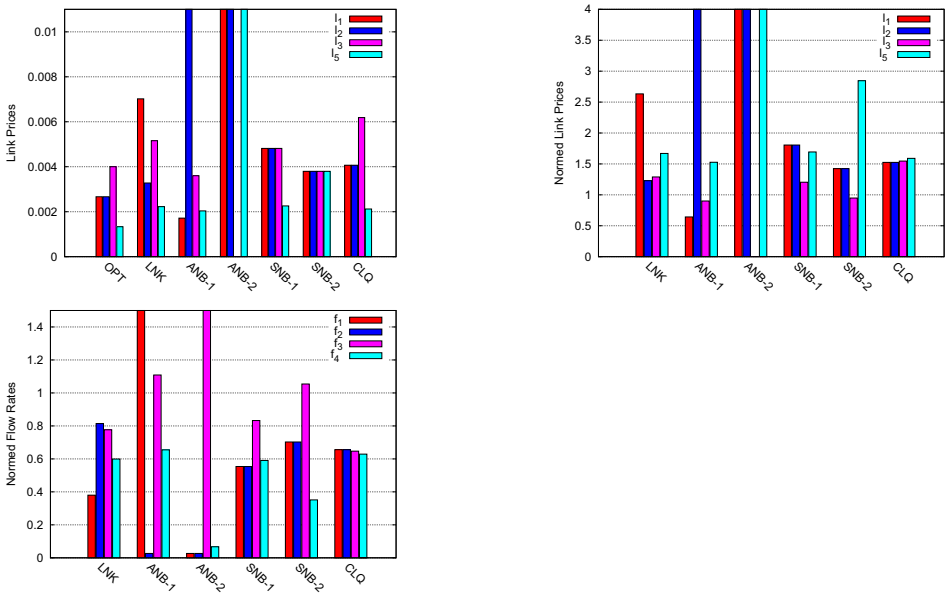


Figure VII.6: Price and Rate Comparison for the Special Scenario

In the *OPT* model, $f_1$ and $f_2$ only belong to clique $q_1$, so they have the same price of $q_1$. $f_4$ only belong to clique $q_2$, so it takes the price of $q_2$. $f_3$ belongs to both cliques, so its price is the addition of the two clique prices.

In the *LNK* model, each link converges to different prices which directly reflect the underlying IEEE 802.11 scheduling. Though intuitively the *LNK* model does not form any concept of regions, noticeably it does not perform far from the *OPT* model. In the example, only $f_1$ on link $l_1$ drifts from the baseline model. It verifies the result in [47] that the performance of 802.11 is surprisingly good and it is decent enough to be used as the underlying protocol.

In the *ANB-n* models, some of the prices and rates converge to extremes. With each link arranging its own neighborhood, the simulation results show that these models do not work well with the price-based approach.

In the *SNB-n* models, some links converge to the same price, as links gather into different regions, and links view the same price within the same neighborhood.

In the *CLQ* model, the link prices converge to the similar share to the *OPT* model, while the absolute values of the link prices and flow rates are different, as the *CLQ* is based on 802.11 instead of an ideal scheduling protocol. Note that the fairness level of *CLQ* is the same as *OPT*. Though there are two exposed terminal pairs in the topology, $f_1$, $f_3$ and $f_3$ and $f_4$, as in each clique, and the two cliques share the flow $f_3$. Considering in an exposed terminal pair the flow cannot hear the other flow as the "bad" flow, and the other flow as "good" flow. Then both $q_1$

and $q_2$ have a "bad" flow and a "good" flow. In such an even distribution of the problem, *CLQ* can still result in relatively fair resource distribution.

### Simple Topologies

Three simple scenarios in Fig. VII.4 are then evaluated to compare the different neighborhood models. Fig. VII.7,VII.8,VII.9 display the link prices, the normalized link prices and the normalized flow rates of the models after convergence for the three scenarios.

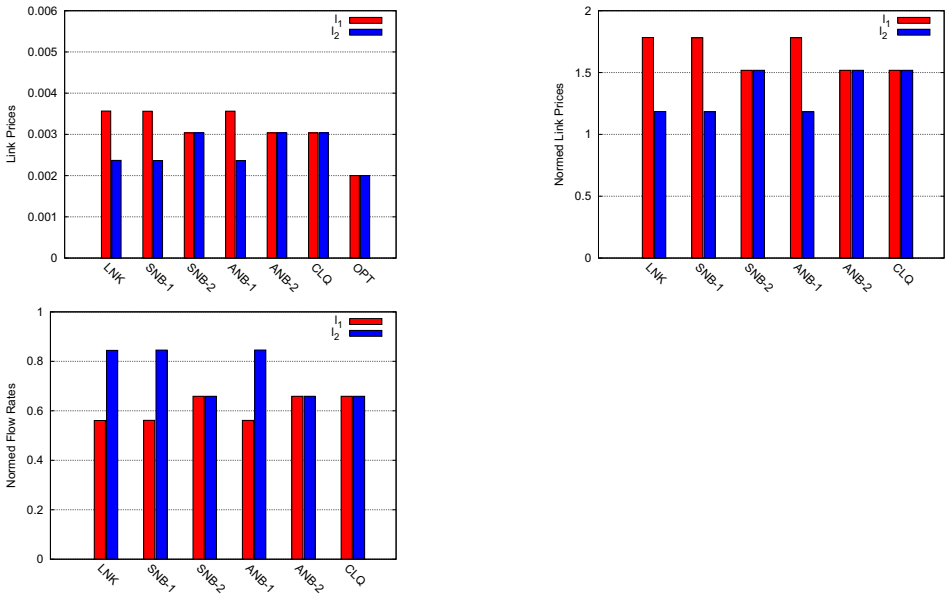


Figure VII.7: Price and Rate Comparison for Scenario 1

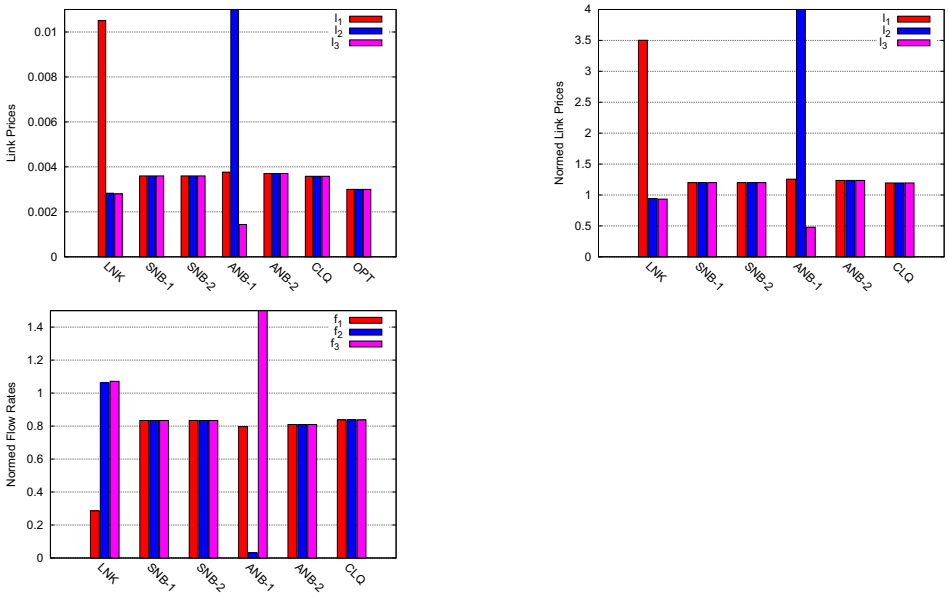


Figure VII.8: Price and Rate Comparison for Scenario 2

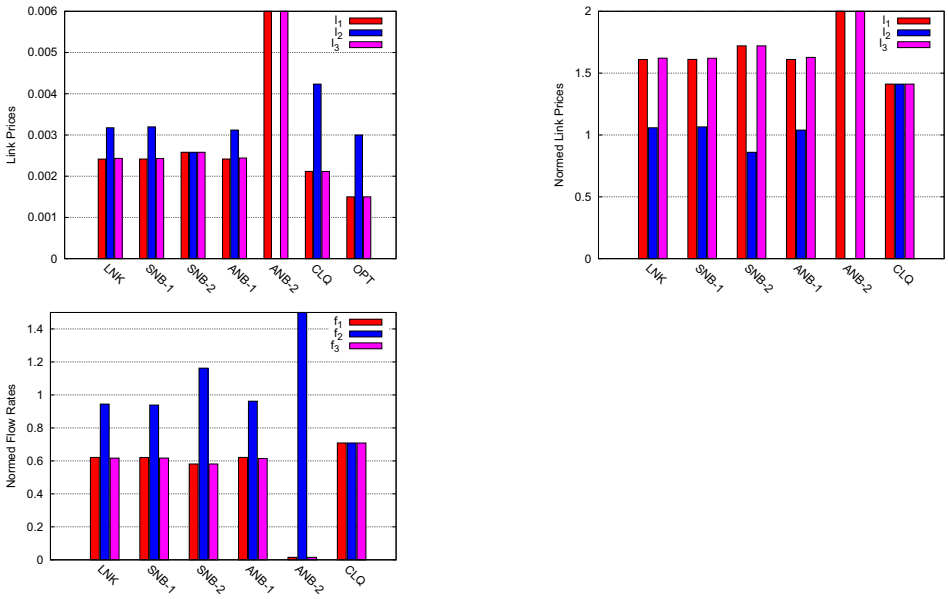


Figure VII.9: Price and Rate Comparison for Scenario 3

**LNK & SNB-1**

Scenario 1 shows that the *LNK* model and the *SNB-1* model can have the same performance. In scenario 1, as the two flows interfere with each other, they are in the same clique in the *OPT* and *CLQ* models, and they get the same price. However, with no price regulation in the *LNK* model, the two flows are affected by the exposed terminal problem and result in different prices and rates. In scenario 1, the *SNB-1* model performs the same as the *LNK* model, as the two links are not directly connected.

Scenario 2 shows that the *SNB-1* model can perform better than the *LNK* model. In scenario 2, the three flows all interfere with each other, they are still in the same clique in the *OPT* and *CLQ* models. The *LNK* follows the underlying scheduling and the three flows do not converge to the same rate. While in the *SNB-1* model, $l_1$ gets $l_3$ as its neighbor via $l_2$, and the same for $l_3$. Thus three flows converge to the same rate in the *SNB-1* model.

**SNB-1 & SNB-2**

In scenario 2, *SNB-1* performs the same as *CLQ*, when scenario 2 forms a connected flow graph. However in scenario 1 and 3, *SNB-1* performs differently from *CLQ* for it does not handle unattached links but grouping them into different neighborhoods.

In scenario 1, *SNB-2* performs the same as *CLQ* because *SNB-2* considers 2-hop neighboring. However in scenario 3, *SNB-2* still treats all the links in one

neighborhood, while $l_1$ and $l_3$ do not interfere with each other like in *CLQ*. Thus, in the *CLQ* model, the price of $l_2$ is the addition of the prices of $l_1$ and $l_3$, but in the *SNB-2* model, all of them have the same price. Similar in Fig. VII.2, all the links are of the same price for $l_3$ interfere with all the other links, though for example, $l_1$ and $l_5$ do not interfere each other, which is considered in *CLQ*.

**ANB-n & SNB-n**

The *ANB-n* models have the problem of information asymmetry. In scenario 2, in the *ANB-1* model, as $l_1$ and $l_3$ have different neighborhoods, they converge to different price values under the effect of underlying IEEE 802.11 scheduling. But the link $l_2$ has a larger neighborhood and considers both $l_1$ and $l_3$ as neighbors. While $l_1$ and $l_3$ utilize the capacity viewed by $l_2$, $l_2$ aggregates the demands on both $l_1$ and $l_3$ as its neighborhood demand. Due to this information asymmetry, the price of $l_2$ increases continuously and the rate of $f_2$ drops close to 0. Accordingly, $l_1$ and $l_3$ gain the resource of $l_2$. Similarly in scenario 3, the *ANB-2* model encounters the same problem, but the link $l_2$ interferes with both $l_1$ and $l_3$ gets all the resources with the other two links starved.

The *ANB-n* models cannot provide reasonable results in most scenarios, and we will replace it with *SNB-n* in all the subsequent experiments.

Table VII.2: Evaluation Metrics for the simple scenarios

| | | *LNK* | *SNB-1* | *SNB-2* | *ANB-1* | *ANB-2* | *CLQ* |
|---|---|---|---|---|---|---|---|
| $\mathcal{J}$ | $scen1$ | 0.961 | 0.961 | 1 | 0.961 | 1 | 1 |
| | $scen2$ | 0.828 | 1 | 1 | 0.567 | 1 | 1 |
| | $scen3$ | 0.957 | 0.959 | 0.889 | 0.953 | 0.342 | 1 |
| $\mathcal{U}$ | $scen1$ | 0.940 | 0.940 | 0.933 | 0.940 | 0.933 | 0.933 |
| | $scen2$ | 0.935 | 0.969 | 0.969 | 0.830 | 0.964 | 0.969 |
| | $scen3$ | 0.946 | 0.946 | 0.950 | 0.947 | 0.600 | 0.945 |

**CLQ & OPT**

Note that in all the three scenarios, the normalized fairness indexes of *CLQ* are 1. Although both scenario 1 and 2 have exposed terminal problem, but as the problem is within one clique, *CLQ* can drive the prices equal. Also with symmetric topology as in scenario 3, the *CLQ* model can exactly follow the *OPT* model.

**Discussion**

Table VII.2 displays the normalized fairness indices and utility function values of all the neighborhood models. Note that the fairness index and the utility function no longer consist with each other. That is, a model with the maximum fairness index among all the models may not have the maximum value of the aggregated utility function. In scenario 2, $\mathcal{V}$ and $\mathcal{U}$ are both the maximum for the *CLQ* model. However, all the other scenarios including the special scenario no longer have the same model for the maximum values.

Common Topologies

The chain topology and the stack topology in Fig. VII.10 are commonly studied by several papers [47, 129, 95]. the converged link prices and flow rates are plotted in Fig. VII.11 and VII.12 separately, with $\mathcal{J}$ and $\mathcal{U}$ summarized in Table VII.3.

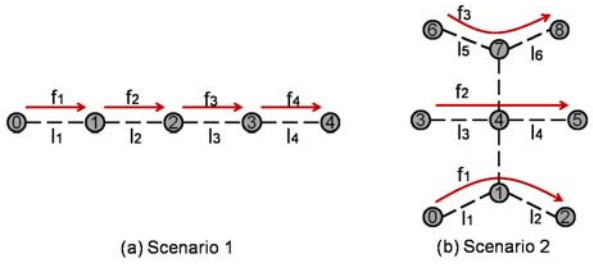


Figure VII.10: Common Scenarios

In scenario 1, the *LNK* model is severely affected by the exposed terminal problem. It shows the most asymmetric distribution of rates, with the $l_1$ and $l_2$ of large prices, and $l_3$ and $l_4$ of small prices.

In the *OPT* model, $l_1$ and $l_4$ are at the symmetric position and converge to the same price. However in the *CLQ* model, flows $f_1$ and $f_4$ have the largest and smallest normalized rates separately. The exposed terminal problem also affects the capacity estimation on the links, which leads to the asymmetric convergence of links $l_1$ and $l_4$. Thus, although the *CLQ* model approximates the *OPT* model and try to regulate the link prices, it inevitably inherits some properties of the underlying protocol. *CLQ* is greatly affected by the direction of the flows. In the

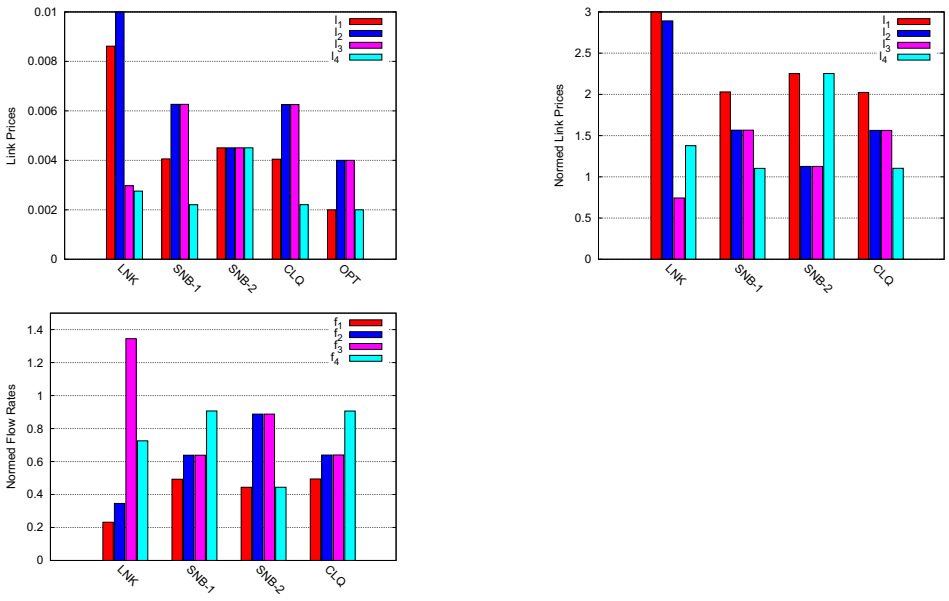


Figure VII.11: Price and Rate Comparison for Common Topology 1

clique of $l_1$, $l_2$, and $l_3$, it contains two "bad" flows and one "good" flow, but in the other clique, it contains two "good" flows and one "bad" flow. The clique with more "bad" flows consequently results in a higher price.

In scenario 2, the flow in the middle gets completely starved in the *LNK* model, which conforms to the result of the TCP connection in [95].

The *CLQ* again follows the *OPT* model well in such a symmetrical topology.

Note that *SNB-1* performs much better than *LNK*. With coordination between each two connected active links, the middle flow gets enhanced for resource competition and finally settles with resource over the amount it is supposed to take.

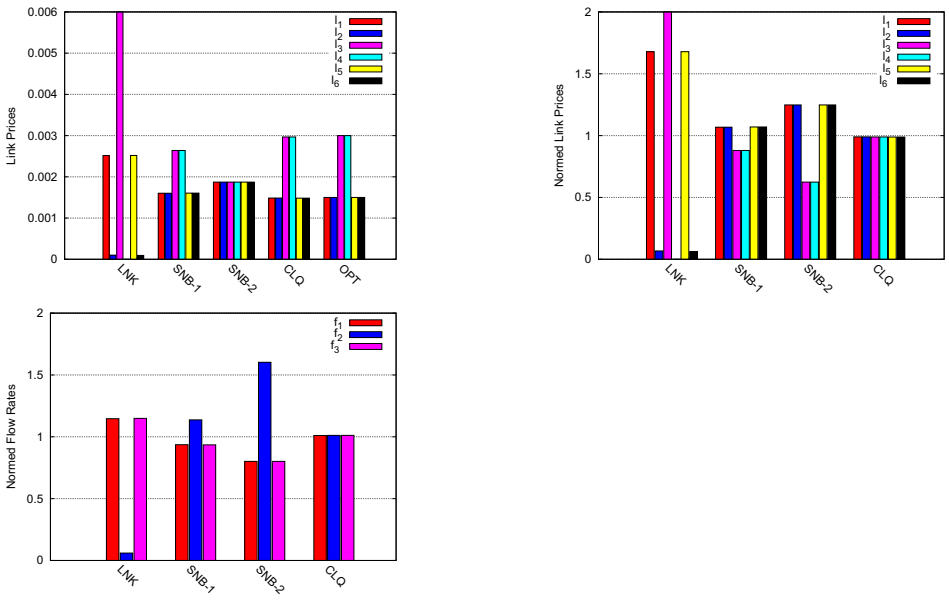


Figure VII.12: Price and Rate Comparison for Common Topology 2

In both scenarios, *SNB-2* acts in an egalitarian manner when other models have varying flow rates and link prices, because all the links are in the same neighborhood in the *SNB-2* model.

**Discussion**

The *CLQ* model can exactly follow the *OPT* model when the topology is symmetric or the nodes with hidden terminal problem symmetric within the same clique. Overall, the *CLQ* model is stable and can perform better than the *LNK* model, and most times better than *SNB-1* and *SNB-2*. As it is a theoretical framework, the construction of clique is complicated compared with other models.

Table VII.3: Evaluation Metrics for the common scenarios

| | | *LNK* | *SNB-1* | *SNB-2* | *CLQ* |
|---|---|---|---|---|---|
| $\mathcal{J}$ | $scen1$ | 0.699 | 0.952 | 0.900 | 0.953 |
| | $scen2$ | 0.701 | 0.991 | 0.889 | 1 |
| $\mathcal{U}$ | $scen1$ | 0.948 | 0.984 | 0.869 | 0.984 |
| | $scen2$ | 0.848 | 1 | 1 | 1 |

The *SNB-1* model enables the information exchange between directly connected links based on the *LNK* model, and this can relieve some of the problems of *LNK* to some extend.

The *SNB-2* model always tends to make the prices of large groups of links evenly. In a topology where links are dense, it may perform well. However when a topology is diverse, it can perform very bad.

### Grid Topologies

A $5 \times 5$ grid topology is studied in this subsection. First 20 per-hop horizontal flows are simulated on the topology and then 5 horizontally paralleled multi-hop flows are tested.

The simulation shows that in the per-hop flow scenario, the *SNB-2* performs much better with fairness index of $0.838$, compared with the *CLQ*, *SNB-1* and *LNK* with fairness indices of $0.650, 0.647, 0.600$. In such a scenario where flows are independent in the sense of originating from different source nodes, *SNB-2* evens the link prices in the largest region and tries to make the short flows coordinate with each other.

In the multi-hop scenario, the fairness indices of *SNB-2, CLQ, SNB-1* and *LNK* become $0.770, 0.762, 0.784, 0.722$. *SNB-1* turns to be the relatively best approach, followed by *CLQ, SNB-2* and *LNK*. In the *SNB-1* model, the clique-concept within a flow is correctly followed, so the flow prices are constrained along the path. In addition, flows do not consider the information from other flows, which in turn is beneficial in a symmetric topology. In this scenario when *SNB-2* tries to equalize link prices, a flow price is not easy to change as it is determined by all the link prices along the path.

Random Topologies

We further study the fairness level of the heuristic approaches in random topologies. 25 nodes are deployed over a $1000$ by $1000$ area. $2$ sets of node topologies are randomly generated. 7 different flow sets of 15 or 25 per-hop flows are established in the topologies as in Fig. VII.13. Topo $1-3$ use the first topology, and topo $4-7$ use the second topology. The rate fairness indices are shown in Table VII.4. The dense topologies have smaller fairness indices compared with the sparse ones.

$15$ flows in topo 1 mainly form three separated groups. *SNB-1* model performs the best in this scenario. In the *SNB-1* model, flows attached with other flows correctly find cliques via its neighbors. Also small regions composed of connected links symmetrically interfere with one another. The other two groups at the bottom distribute their own channels without contending with each other. The *SNB-2*

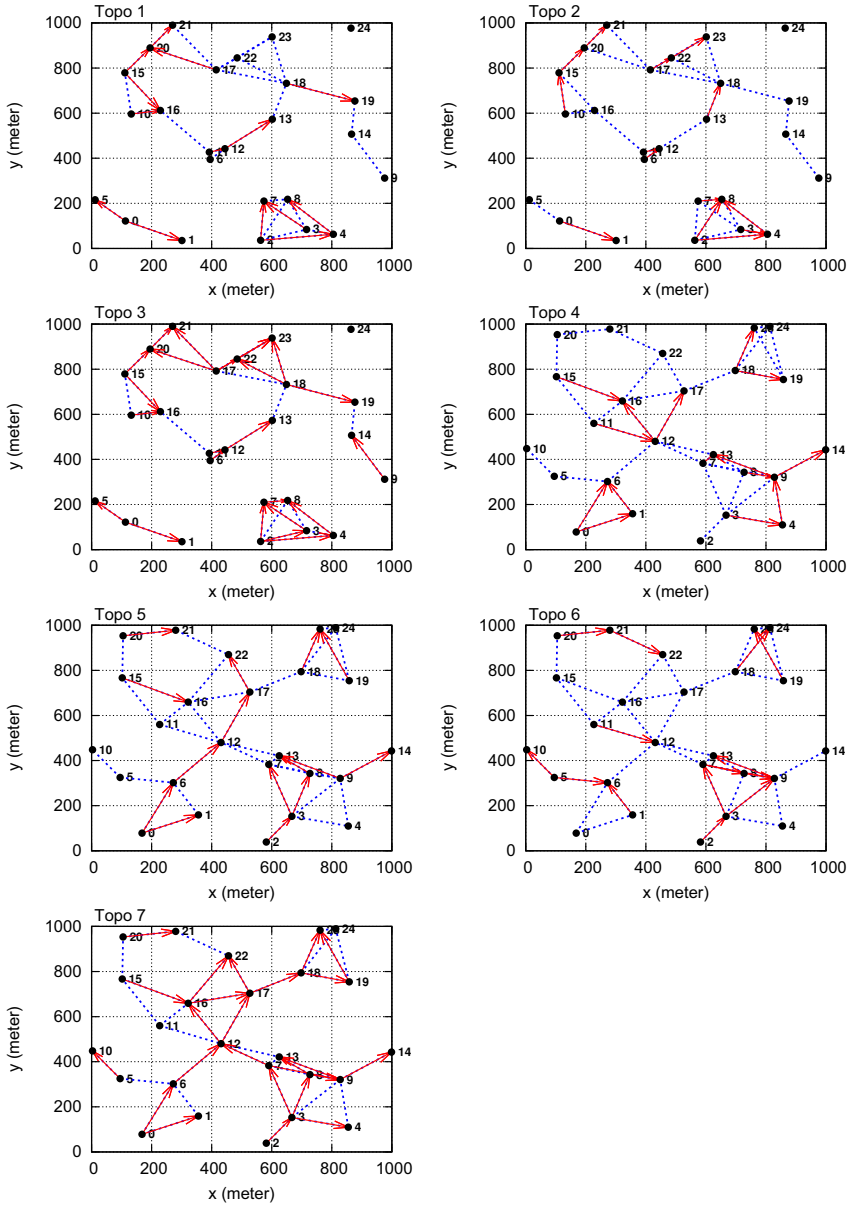


Figure VII.13: Random Topologies

Table VII.4: Flow Fairness Index Comparison in Random Topologies

| | *CLQ* | *LNK* | *SNB-1* | *SNB-2* |
|---|---|---|---|---|
| $topo1$ | 0.918 | 0.869 | 0.910 | 0.434 |
| $topo2$ | 0.927 | 0.832 | 0.520 | 0.404 |
| $topo3$ | 0.626 | 0.535 | 0.623 | 0.672 |
| $topo4$ | 0.904 | 0.745 | 0.831 | 0.940 |
| $topo5$ | 0.908 | 0.803 | 0.892 | 0.337 |
| $topo6$ | 0.949 | 0.816 | 0.611 | 0.341 |
| $topo7$ | 0.667 | 0.562 | 0.623 | 0.604 |

model performs the worst in this scenario, as it evenly distributes the channel resources in each isolated group.

Comparing topo 2 with topo 1, the *SNB-2* model has the same neighborhood as in topo 1, so its fairness index does not get better. The *NB1s* model no longer performs well in topo 2, as the flows become more unattached and asymmetric in the top region.

Topo 3 has $10$ more flows than topo 1 and 2. With increased number of flows in the top isolated region, *SNB-2* model out-performs all the other approaches, especially when the top isolated region becomes symmetric centered by the node in the middle. Also *SNB-1* model performs the second best in this scenario.

Topo 4-7 are implemented on the other node topology, and the simulation results are consistent with the first three topologies. Topo 7 is quite symmetric with two unattached flows on the two sides and connected flows in the middle. The fairness indices of both *SNB-1* and *SNB-2* are larger than the *LNK* model. Note the *CLQ* model almost always has the largest fairness index.

# CHAPTER VIII

# CONCLUDING REMARKS

This dissertation studies the design of wireless networked control systems. When the control loop is closed via wireless networks, the system performance will be greatly affected by the communication properties. Among all the networking problems, resource allocation has been widely investigated to economically assign the available resources. It implicitly influences the end-to-end delay, data loss and throughput of all the control systems. Our work is mainly about optimal and fair resource allocation that improves the performance of wireless networked control systems.

The contributions of the dissertation are as follows. First, the design concern of the NCS is decomposed into two design spaces. The NCS is guaranteed stable by using a passive control structure in the control layer. The overall NCS performance is then optimized by adjusting the network protocol parameters. Through experiment study, the important convex relationship between the retransmission limit of IEEE 802.11 and the NCS performance is revealed. Based on this observation, we present a MAC-layer control algorithm that dynamically tunes the retransmission limit so that the NCS performance can be either bounded by an error threshold or optimized. The results of the simulation study show that our

MAC controller can achieve robust adaption and enable desirable NCS performance under a variety of network conditions including highly lossy channel and with heavy background traffic.

Second, we investigate the problem of NCS performance optimization in terms of tracking error minimization. It presents an optimization formulation where the objective is to maximize a utility function that characterizes the relationship between the sampling rate and the disturbance rejection capability of the control system. The constraints come from the wireless network capacity and the packet requirements. A distributed double-price-based algorithm is presented to solve the problem. Our solution has desired properties from both theoretical and practical aspects. From theoretical perspective, it is shown to achieve both system stability and performance optimality. From the view of practice, it can be naturally deployed over the existing layered networking systems with well-defined cross-layer interactions. Simulation studies conducted in an integrated simulation environment consisting of Matlab/Simulink and ns-2 demonstrate that our algorithm is able to provide agile and stable sampling rate adaptation and achieve optimal NCS performance.

Third, we redefine the concept of fairness for non-convex capacity regions. We adopt an axiomatic approach based on the game theoretic framework and define the new fairness model based on the Nash extension solution (NES). We further present an efficiency enhanced version of Nash extension solution, which pushes the NES to the strong Pareto frontier. Both presented fairness models will lead

to a unique rate allocation solution under non-convex capacity regions, which coincides with the NBS under the convex cases and approximates the NBS under the non-convex cases. we present a time-decomposed price-based rate allocation algorithm that iteratively converges to the fair solution and prove its stability. Distributed implementation of the algorithm is also presented and evaluated using a simulation-based study.

Fourth, we aim at understanding the gap between heuristic fair rate allocation solutions and the optimal solution. It characterizes the heuristic solutions using neighborhood models to differentiate the shared resource regions. Then a price-based fairness model is established where the impact of neighborhood information on end-to-end flow fairness can be evaluated on a common framework. The simulation-based study has revealed several important properties, including the importance of symmetric knowledge on the construction of a neighborhood and the limitation of 2-hop neighborhood information. We believe that these properties discovered in this dissertation have significant implications to future fair rate allocations in multi-hop wireless networks.

# BIBLIOGRAPHY

[1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer, 2007.

[2] A.T. Al-Hammouri, M.S. Branicky, V. Liberatore, and S.M. Phillips. Decentralized and dynamic bandwidth allocation in networked control systems. In *IPDPS*, page 8, Apr. 2006.

[3] M. Alicherry, R. Bhatia, and L. Li. Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks. In *Proc. of ACM MobiCom*, 2005.

[4] G. Alldredge, M. S. Branicky, and V. Liberatore. Play-Back Buffers in Networked Control Systems: Evaluation and Design. In *Proc. American Control Conf.*, Seattle, June 2008.

[5] E. Altman, T. Basar, and R. Srikant. Congestion Control as a Stochastic Control Problem with Action Delays. *Automatica*, 35:1937–1950, 1999.

[6] J. Bai, E. Eyisi, Y. Xue, and X. Koutsoukos. Dynamic Tuning Retransmission Limit of IEEE 802.11 MAC Protocol for Networked Control Systems. In *Proc. of CPSCom*, pages 666–672, 2010.

[7] L. Bao, M. Skoglund, C. Fischione, and K.H. Johansson. Rate allocation for quantized control over noisy channels. In *WiOPT*, pages 1–9, 2009.

[8] D. Bertsekas and R. Gallager. *Data networks*. Prentice-Hall, 1992.

[9] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[10] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Jour. on*, 18(3):535–547, March 2000.

[11] M. S. Branicky, S. M. Phillips, and W. Zhang. Scheduling and feedback co-design for networked control systems. In *Decision and Control, Proc. of the 41st IEEE Conf. on*, volume 2, pages 1211–1217, Dec. 2002.

[12] M.S. Branicky, S.M. Phillips, and W. Zhang. Stability of networked control systems: Explicit analysis of delay. In *Proc. of the American Control Conference*, pages 2352–2357, 2000.

[13] J.H. Braslavsky, R.H. Middleton, and J.S. Freudenberg. Feedback Stabilization Over Signal-to-Noise Ratio Constrained Channels. *Automatic Control, IEEE Trans. on*, 52(8):1391–1403, Aug. 2007.

[14] B. Briscoe. Flow rate fairness: dismantling a religion. *SIGCOMM Comput. Commun. Rev.*, 37(2):63–74, 2007.

[15] D. Carnevale, A. R. Teel, and D. Nesic. A Lyapunov Proof of an Improved Maximum Allowable Transfer Interval for Networked Control Systems. *Automatic Control, IEEE Trans. on*, 52(5):892–897, May 2007.

[16] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle. Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –13, april 2006.

[17] M. Chow and Y. Tipsuwan. Network-based control systems: a tutorial. In *IECON*, volume 3, pages 1593–1602, 2001.

[18] J. Colandairaj, G. W. Irwin, and W. G. Scanlon. Analysis and co-simulation of an IEEE 802.11b wireless networked control system. In *Proc. of the IFAC World Congress*, 2005.

[19] J. Colandairaj, G. W. Irwin, and W. G. Scanlon. Analysis of an IEEE 802.11b wireless networked control system. In *Proc. of the First NeCST Workshop*, 2005.

[20] J. Colandairaj, G. W. Irwin, and W. G. Scanlon. A Co-Design Solution for Wireless Feedback Control. In *Networking, Sensing and Control, IEEE International Conf. on*, pages 404–409, Apr 2007.

[21] J. Colandairaj, G. W. Irwin, and W. G. Scanlon. Wireless networked control systems with qos-based sampling. *Control Theory Applications, IET*, 1(1):430–438, january 2007.

[22] J. P. Conley and S. Wilkie. An Extension of the Nash Bargaining Solution to Nonconvex Problems. *Games and Economic Behavior*, 13(1):26–38, 1996.

[23] J.P. Conley and S. Wilkie. Implementing the nash extension bargaining solution for non-convex problems. *Review of Economic Design*, 1(1):205–216, 1994.

[24] S. Cui, R. Madan, A. Goldsmith, and S. Lall. Cross-Layer Energy and Delay Optimization in Small-Scale Sensor Networks. *Wireless Communications, IEEE Trans. on*, 6(10):3688–3699, Oct. 2007.

[25] D.B. Dacic and D. Nesic. Observer design for linear networked control systems using matrix inequalities. In *Decision and Control, 46th IEEE Conf. on*, pages 3315–3320, Dec. 2007.

[26] Dragan B. Dačić and D. Nešić. Quadratic stabilization of linear networked control systems via simultaneous protocol and controller design. *Automatica*, 43:1145–1155, July 2007.

[27] L. Dugard and Eds. E.I. Verriest. *Stability and Control of Time-Delay Systems (Lecture Notes in Control and Information Sciences)*, volume 228. Heidelberg, Germany: Springer-Verlag, 1997.

[28] N. Elia and S.K. Mitter. Stabilization of linear systems with limited information. *Automatic Control, IEEE Trans. on*, 46(9):1384–1400, Sep. 2001.

[29] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Trans. Netw.*, 15(6):1333–1344, 2007.

[30] G. F. Franklin, J. D. Powel, and M. L. Workman. *Digital Control of Dynamic Systems*. Addison Wesley, 1997.

[31] M.E.M.B. Gaid, A. Cela, and Y. Hamam. Optimal integrated control and scheduling of networked control systems with communication constraints: application to a car suspension system. *Control Systems Technology, IEEE Trans. on*, 14(4):776–787, July 2006.

[32] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[33] I. A. Gravagne, J. M. Davis, J. J. Dacunha, and R. J. Marks. Bandwidth reduction for controller area networks using adaptive sampling. In *ICRA. IEEE International Conf. on*, volume 5, pages 5250–5255, Apr. 2004.

[34] M. Grenier and N. Navet. Fine-Tuning MAC-Level Protocols for Optimized Real-Time QoS. *Industrial Informatics, IEEE Trans. on*, 4(1):6–15, Feb. 2008.

[35] Gagan Raj Gupta and Ness B. Shroff. Delay analysis and optimality of scheduling policies for multihop wireless networks. *IEEE/ACM Trans. Netw.*, 19:129–141, February 2011.

[36] P. Gupta and P.R. Kumar. The Capacity of Wireless Networks. *IEEE Trans. Information Theory*, 46:388–404, 2000.

[37] R.A. Gupta and Mo-Yuen Chow. Networked Control System: Overview and Research Trends. *Industrial Electronics, IEEE Trans. on*, 57(7):2527–2535, July 2010.

[38] W.P.M.H. Heemels, A.R. Teel, N. van de Wouw, and D. Nešić and. Networked Control Systems With Communication Constraints: Tradeoffs Between Transmission Intervals, Delays and Performance. *Automatic Control, IEEE Trans. on*, 55(8):1781–1796, aug. 2010.

[39] J. Hespanha, A. Ortega, and L. Vasudevan. Towards the control of linear systems with minimum bit-rate. In *In Proc. of the Int. Symp. on the Mathematical Theory of Networks and Syst*, 2002.

[40] J. Hespanha and A.R. Teel. Stochastic impulsive systems driven by renewal processes. In *In Proc. of the International Symposium on Mathematical Theory of Networks and Systems (MTNS*, pages 606–618, 2005.

[41] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A Survey of Recent Results in Networked Control Systems. *Proc. of the IEEE*, 95(1):138–162, Jan. 2007.

[42] Z. Huo, H. Fang, and G. Yan. Co-design for NCS robust fault-tolerant control. In *Industrial Technology. ICIT. IEEE International Conf. on*, pages 119–124, Dec. 2005.

[43] G. Irwin, J. Colandairaj, and W. G. Scanlon. An Overview of Wireless Networks in Control and Monitoring. In De-Shuang Huang, Kang Li, and George Irwin, editors, *Computational Intelligence*, volume 4114 of *Lecture Notes in Computer Science*, pages 1061–1072. Springer Berlin / Heidelberg, 2006.

[44] D. Yuan J. Li, H. Yu and X. Guo. Co-design of Networks and Control Systems with Synthesized Controller. *International Jour. of Information and System Sciences*, 7(1):131–140, 2011.

[45] J. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom*, pages 66–80, 2003.

[46] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. *Wirel. Netw.*, 11:471–487, July 2005.

[47] A. Jindal and K. Psounis. The achievable rate region of 802.11-scheduled multihop networks. *IEEE/ACM Trans. Netw.*, 17:1118–1131, August 2009.

[48] P.A. Kawka and A.G. Alleyne. Stability and feedback control of wireless networked systems. In *American Control Conference. Proc. of*, volume 4, pages 2953–2959, June 2005.

[49] A.G. Kelkar and S.N. Joshi. Robust control of non-passive systems via passification [for passification read passivation]. In *American Control Conference*, volume 5, pages 2657 –2661, 1997.

[50] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *The Jour. of the Operational Research Society*, 49(3):237–252, 1998.

[51] D. Kim, Y.S. Lee, W.H. Kwon, and H.S. Park. Maximum allowable delay bounds of networked control systems. *Control Engineering Practice*, 11(11):1301–1313, 2003.

[52] B. R. Klingenberg, U. Ojha, and M. Chow. Predictive constrained gain scheduling For UGV path tracking in a networked control system. In *Proc. of IEEE/RSJ*, IROS, pages 1935–1940, Piscataway, NJ, USA, 2009. IEEE Press.

[53] M. Kodialam and T. Nandagopal. Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proc. of ACM Mobicom*, 2005.

[54] N. Kottenstette. *Control of passive plants with memoryless nonlinearitites over wireless networks*. PhD thesis, University of Notre Dame, 2007.

[55] N. Kottenstette, J. Hall, X. koutsoukos, J. Sztipanovits, and P. Antsaklis. Design of Networked Control Systems Using Passivity. *IEEE Trans. Control Systems Technlology (Accepted for publication)*, 2012.

[56] N. Kottenstette, X. Koutsoukos, J. Hall, J. Sztipanovits, and P. Antsaklis. Passivity-Based Design of Wireless Networked Control Systems for Robustness to Time-Varying Delays. In *Proc. of RTSS*, pages 15–24, Washington, DC, USA, 2008. IEEE Computer Society.

[57] P.R. Kumar. New technological vistas for systems and control: the example of wireless networks. *Control Systems, IEEE*, 21(1):24–37, feb 2001.

[58] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-Down Approach*. Addison-Wesley Publishing Company, USA, 5th edition, 2009.

[59] Michael Lemmon and Xiaobo Sharon Hu. Almost sure stability of networked control systems under exponentially bounded bursts of dropouts. In *Proceedings of the 14th international conference on Hybrid systems: computation and control*, pages 301–310, 2011.

[60] W. S. Levine. *Control System Funcamentals*. CRC Press, 2000.

[61] Q. Li, G. Yi, C. Wang, L. Wu, and C. Ma. LMI-based Stability Analysis of Networked Control Systems with Large Time-varying delays. In *Mechatronics and Automation, Proc. of IEEE International Conf. on*, pages 713–717, June 2006.

[62] Z. Li and M. Chow. Adaptive Multiple Sampling Rate Scheduling of Real-time Networked Supervisory Control System - Part II. In *IECON*, pages 4615–4620, Nov. 2006.

[63] F. Lian, J.R. Moyne, and D.M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *Control Systems, IEEE*, 21(1):66–83, Feb. 2001.

[64] Feng-Li Lian, J.K. Yook, D.M. Tilbury, and J. Moyne. Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems. *Industrial Informatics, IEEE Transactions on*, 2(1):12 – 24, feb. 2006.

[65] V. Liberatore. Integrated Play-Back, Sensing, and Networked Control. In *INFOCOM. 25th IEEE International Conf. on*, pages 1–12, Apr. 2006.

[66] X. Lin and N. Shroff. Joint rate control and scheduling in multihop wireless networks. In *in Proc. of IEEE Conf. on Decision and Control*, pages 1484–1489, 2004.

[67] F. Liu and Y. Yao. Modeling and analysis of networked control systems using hidden Markov models. In *Machine Learning and Cybernetics. Proc. of International Conf. on*, volume 2, pages 928–931, Aug. 2005.

[68] G. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu. Networked Predictive Control of Systems With Random Network Delays in Both Forward and Feedback Channels. *Industrial Electronics, IEEE Trans. on*, 54(3):1282–1297, June 2007.

[69] X. Liu and A. Goldsmith. Wireless communication tradeoffs in distributed control. In *CDC*, volume 1, pages 688–694, Dec. 2003.

[70] X. Liu and A. Goldsmith. Wireless medium access control in networked control systems. In *American Control Conference. Proc. of*, volume 4, pages 3605–3610, July 2004.

[71] X. Liu and A. Goldsmith. Wireless network design for distributed control. In *CDC*, volume 3, pages 2823–2829, Dec. 2004.

[72] S.H. Low and D.E. Lapsley. Optimization flow control. I. Basic algorithm and convergence. *Networking, IEEE/ACM Transactions on*, 7(6):861 –874, dec. 1999.

[73] C. Lozoya, P. Mart, M. Velasco, J. Fuertes, and E. Martn. Simulation study of a remote wireless path tracking control with delay estimation for an autonomous guided vehicle. *The International Jour. of Advanced Manufacturing Technology*, 52:751–761, 2011. 10.1007/s00170-010-2736-x.

[74] R. Luck and A. Ray. Experimental verification of a delay compensation algorithm for integrated communication and control systems. *International Jour. of Control*, 59(6):1357–1372, 1994.

[75] C. Ma, S. Chen, and W. Liu. Maximum allowable delay bound of networked control systems with multi-step delay. *Simulation Modelling Practice and Theory*, 15(5):513–520, 2007.

[76] P. Marti, J. Yepez, M. Velasco, R. Villa, and J. M. Fuertes. Managing quality-of-control in network-based control systems by controller and message scheduling co-design. *Industrial Electronics, IEEE Trans. on*, 51(6):1159–1167, dec. 2004.

[77] S. Mascolo. Classical control theory for congestion avoidance in high-speed Internet. In *Decision and Control. Proc. of the 38th IEEE Conf. on*, volume 3, pages 2709–2714, 1999.

[78] L. A. Montestruque and P. J. Antsaklis. Networked Control Systems: A Model-Based Approach. In *Handbook of Networked and Embedded Control Systems*, pages 601–626. Hristu and Levine, Eds., 2005.

[79] L.A. Montestruque and P. Antsaklis. Stability of model-based networked control systems with time-varying transmission times. *Automatic Control, IEEE Trans. on*, 49(9):1562–1572, Sep. 2004.

[80] P. Naghshtabrizi, J. Hespanha, and A. Teel. Stability of infinite-dimensional impulsive systems with application to network control systems. In *American Control Conf.*, 2007.

[81] G. Nair. Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39(4):585–593, April 2003.

[82] D. Nesic and A.R. Teel. Input-output stability properties of networked control systems. *Automatic Control, IEEE Trans. on*, 49(10):1650–1667, Oct. 2004.

[83] S. Nethi, M. Pohjola, L. Eriksson, and R. Jäntti. Simulation case studies of wireless networked control systems. In *PM2HW2N*, pages 100–104, 2007.

[84] G. Nikolakopoulos, A. Panousopoulou, and A. Tzes. Experimental controller tuning and QoS optimization of a wireless transmission scheme for real-time remote control applications. *Control Engineering Practice*, 16(3):333–346, 2008.

[85] J. Nilsson. *Real-time control systems with delays*. PhD thesis, Dept. Automatic Control, Lund Institute of Technology, Lund, Sweden, Jan. 1998.

[86] Johan Nilsson, Bo Bernhardsson, and Björn Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, 1998.

[87] U. Ojha and M. Chow. Behavioral control based adaptive bandwidth allocation in a system of Unmanned Ground Vehicles. In *IECON*, pages 3123–3128, Nov. 2010.

[88] P.G. Otanez, J.R. Moyne, and D.M. Tilbury. Using deadbands to reduce communication in networked control systems. In *American Control Conf. Proc. of*, volume 4, pages 3015–3020, 2002.

[89] M. Pajic, S. Sundaram, G.J. Pappas, and R. Mangharam. The Wireless Control Network: A New Approach for Control Over Networks. *Automatic Control, IEEE Transactions on*, 56(10):2305 –2318, 2011.

[90] H.S. Park, Y.H. Kim, D. Kim, and W.H. Kwon. A scheduling method for network-based control systems. *Control Systems Technology, IEEE Trans. on*, 10(3):318–330, May 2002.

[91] N. Pereira, B. Andersson, and E. Tovar. WiDom: A Dominance Protocol for Wireless Medium Access. *Industrial Informatics, IEEE Trans. on*, 3(2):120–130, May 2007.

[92] Fan Qiu, Jia Bai, and Yuan Xue. Optimal Rate Allocation in Multi-hop Wireless Networks with Delay Constraints. Technical report, Vanderbilt University, 2011.

[93] B. Radunovic and J.Y. Le Boudec. Rate performance objectives of multihop wireless networks. *Mobile Computing, IEEE Trans. on*, 3(4):334–349, 2004.

[94] S. Rangwala, A. Jindal, K. Jang, K. Psounis, and R. Govindan. Understanding congestion control in multi-hop wireless mesh networks. In *MobiCom '08*, pages 291–302, 2008.

[95] S. Rangwala, A. Jindal, K. Jang, K. Psounis, and R. Govindan. Neighborhood-centric congestion control for multi-hop wireless mesh networks. *ACM/IEEE Trans. on Networking*, to appear 2011.

[96] K. Årzén and A. Cervin. Control and Embedded Computing: Survey of Research Directions. In *Proc. of the 16th IFAC World Congress*. Elsevier, July 2005.

[97] A. Ray and Y. Halevi. Integrated Communication and control systems: Part II-Design Considerations. In *Dynamic Systems, Measurement and Control*, volume 110, pages 374–381, 1988.

[98] H. Rehbinder and M. Sanfridson. Scheduling of a limited communication channel for optimal control. In *Decision and Control. Proc. of the 39th IEEE Conf. on*, volume 1, pages 1011–1016, 2000.

[99] D. Riley, E. Eyisi, J. Bai, X. Koutsoukos, Y. Xue, and J. Sztipanovits. Networked Control System Wind Tunnel (NCSWT) - An evaluation tool for networked multi-agent systems. In *Proc. of SIMUTools*, 2011.

[100] A. Saifullah, C. Wu, P. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen. Near Optimal Rate Selection for Wireless Control Systems. In *Proc. of RTAS*, April 2012.

[101] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S.S. Sastry. Foundations of Control and Estimation Over Lossy Networks. *Proc. of the IEEE*, 95(1):163–187, Jan. 2007.

[102] D. Seto, J.P. Lehoczky, L. Sha, and K.G. Shin. On task schedulability in real-time control systems. In *Proc. of RTSS*, pages 13 –21, 1996.

[103] Samarth H. Shah, Kai Chen, and Klara Nahrstedt. Dynamic bandwidth management in single-hop ad hoc wireless networks. *Mob. Netw. Appl.*, 10:199–217, February 2005.

[104] S. Shakkottai, T.S. Rappaport, and P.C. Karlsson. Cross-layer design for wireless networks. *Communications Magazine, IEEE*, 41(10):74–80, Oct. 2003.

[105] J. Sheu, C. Liu, S. Wu, and Y. Tseng. A priority MAC protocol to support real-time traffic in ad hoc networks. *Wirel. Netw.*, 10:61–69, Jan. 2004.

[106] S. Soucek and T. Sauter. Quality of service concerns in IP-based control systems. *Industrial Electronics, IEEE Trans. on*, 51(6):1249–1258, Dec. 2004.

[107] J.D. Spragins, J.L. Hammond, and K. Pawlikowski. *Telecommunications: Protocols and Designs*. Reading, MA: Addison-Wesley, 1991.

[108] R. Srikant. *The mathematics of internet congestion control*. Birkhauser Boston, 2003.

[109] A. Stolyar. Maximizing Queueing Network Utility Subject to Stability: Greedy Primal-Dual Algorithm. *Queueing Syst. Theory Appl.*, 50(4):401–457, 2005.

[110] S. Stramigioli, C. Secchi, A.J. van der Schaft, and C. Fantuzzi. A novel theory for sampled data system passivity. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 2, pages 1936 – 1941 vol.2, 2002.

[111] M. Tabbara and D. Nesic. Input-Output Stability of Networked Control Systems With Stochastic Protocols and Channels. *Automatic Control, IEEE Trans. on*, 53(5):1160–1175, June 2008.

[112] M. Tabbara, D. Nesic, and A.R. Teel. Stability of Wireless and Wireline Networked Control Systems. *Automatic Control, IEEE Trans. on*, 52(9):1615–1630, Sep. 2007.

[113] S. Tatikonda and S. Mitter. Control under communication constraints. *Automatic Control, IEEE Trans. on*, 49(7):1056–1068, July 2004.

[114] Y. Tipsuwan and M. Chow. Network-Based Controller Adaptation Based On QoS Negotiation and Deterioration. *Proc. of IECON*, pages 1794–1799, 2001.

[115] Y. Tipsuwan and M. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11(10):1099–1111, 2003.

[116] Y. Tipsuwan and M. Chow. Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation-part II: teleoperation. *Industrial Electronics, IEEE Trans. on*, 51(6):1228–1237, Dec. 2004.

[117] Y. Tipsuwan, S. Kamonsantiroj, J. Srisabye, and P. Chongstitvattana. An auction-based dynamic bandwidth allocation with sensitivity in a wireless networked control system. *Comput. Ind. Eng.*, 57:114–124, August 2009.

[118] M. Velasco, J. M. Fuertes, C. Lin, P. Marti, and S. Brandt. A control approach to bandwidth management in networked control systems. In *IECON 2004. 30th Annual Conf. of IEEE*, volume 3, pages 2343–2348, Nov. 2004.

[119] G.C. Walsh, H. Ye, and L.G. Bushnell. Stability analysis of networked control systems. *Control Systems Technology, IEEE Trans. on*, 10(3):438–446, May 2002.

[120] W. Wang, L. Zhang, X. Yin, and Y. Song. Co-design of control and scheduling in networked control system with delay and access constraint. In *Electric Information and Control Engineering (ICEICE), 2011 International Conf. on*, pages 1224–1228, Apr. 2011.

[121] X. Wang, M. Chen, H. Huang, V. Subramonian, C. Lu, and C.D. Gill. Control-Based Adaptive Middleware for Real-Time Image Transmission over Bandwidth-Constrained Networks. *Parallel and Distributed Systems, IEEE Trans. on*, 19(6):779–793, June 2008.

[122] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *Industrial Electronics, IEEE Trans. on*, 49(6):1265–1282, Dec 2002.

[123] W.S. Wong and R.W. Brockett. Systems with finite communication bandwidth constraints. II. Stabilization with limited information feedback. *Automatic Control, IEEE Trans. on*, 44(5):1049–1053, May 1999.

[124] Q. Wu, Y. Li, and Y. Qin. A Scheduling Method Based on Deadline for CAN-based Networked Control Systems. In *Mechatronics and Automation, Proc. of IEEE International Conf. on*, pages 345–350, June 2006.

[125] F. Xia, X. Dai, Z. Wang, and Y. Sun. Feedback based network scheduling of networked control systems. In *Control and Automation. ICCA. International Conf. on*, volume 2, pages 1231–1236, June 2005.

[126] F. Xia, Y. Sun, and Y. Tian. Feedback scheduling of priority-driven control networks. *Computer Standards & Interfaces*, pages 539–547, 2009.

[127] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith. Joint optimization of communication rates and linear systems. *Automatic Control, IEEE Trans. on*, 48(1):148–153, jan 2003.

[128] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith. *Joint optimization of wireless communication and networked control systems*, volume 3355, chapter Switching and Learning in Feedback Systems, pages 248–272. Springer Berlin/Heidelberg, 2005.

[129] K. Xu, M. Gerla, L. Qi, and Y. Shu. TCP Unfairness in ad hoc wireless networks and a neighborhood RED solution. *Wirel. Netw.*, 11(4):383–399, 2005.

[130] Y. Xue, B. Li, and K. Nahrstedt. Channel-relay price pair: towards arbitrating incentives in wireless ad hoc networks. *Wireless Communications and Mobile Computing*, 6:235–245, 2006.

[131] Y. Xue, B. Li, and K. Nahrstedt. Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-based Approach. *IEEE Trans. on Mobile Computing*, 5:347–364, 2006.

[132] Haikel Yaiche, Ravi R. Mazumdar, and Catherine Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Trans. on Networking*, 8:667–678, 2000.

[133] H. Ye, G.C. Walsh, and L.G. Bushnell. Real-time mixed-traffic wireless networks. *Industrial Electronics, IEEE Trans. on*, 48(5):883–890, Oct. 2001.

[134] J. Yepez, P. Marti, and J. M. Fuertes. Control loop scheduling paradigm in distributed control systems. In *IECON. The 29th Annual Conf. of the IEEE*, volume 2, pages 1441–1446, Nov. 2003.

[135] T. Yoo, E. Setton, X. Zhu, A. Goldsmith, and B. Girod. Cross-layer design for video streaming over wireless ad hoc networks. In *Multimedia Signal Processing, IEEE 6th Workshop on*, pages 99–102, Sep. 2004.

[136] Y. Yu and G.B. Giannakis. Cross-Layer Congestion and Contention Control for Wireless Ad Hoc Networks. *IEEE Trans. Wirel. Commun.*, 7(1):37–42, Jan. 2008.

[137] D. Yue, Q. Han, and J. Lam. Network-based robust control of systems with uncertainty. *Automatica*, 41(6):999–1007, 2005.

[138] C. Zhang and X. Liu. Trade-off between the sampling rate and the data accuracy. In *American Control Conf., 2008*, pages 2631–2636, June 2008.

[139] L. Zhang. *Access Scheduling and Controller Design in Networked Control Systems*. PhD thesis, Mechanical Engineering, University of Maryland, 2005.

[140] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of Networked Control Systems. *IEEE Control Systems Magazine*, 21:84–99, 2001.

[141] W. Zhang and M.S. branicky. Stability of networked control systems with time-varying transmission period. Allerton Conf. Communication, Control, and Computing, Urbana, IL, Oct. 2001.

[142] Y.B. Zhao, G.P. Liu, and D. Rees. Integrated predictive control and scheduling co-design for networked control systems. *Control Theory Applications, IET*, 2(1):7–15, Jan. 2008.