



KTH Electrical Engineering

Contention-based Multiple Access Architectures for Networked Control Systems

CHITHRUPA RAMESH

Licentiate Thesis
Stockholm, Sweden 2011

KTH School of Electrical Engineering
Automatic Control Lab
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i telekommunikation fredagen den 18 Februari 2011, klockan 10:15 i sal D3, Kungliga Tekniska högskolan, Lindstedtsv 5, Stockholm.

© Chithrupa Ramesh, February 2011

Tryck: Universitetsservice US AB

Abstract

Networked Control Systems (NCSs) use a wireless network for communication between sensors and controllers, and require a Medium Access Controller (MAC) to arbitrate access to the shared medium. Traditionally, a MAC for control systems is chosen primarily based on the delay it introduces in the closed loop. This thesis focuses on the design of a contention-based MAC, in a time-varying, resource-constrained network for closed loop systems.

In this thesis, we advocate the use of a state-aware MAC, as opposed to an agnostic MAC, for NCSs. A state-aware MAC uses the state of the plant to influence access to the network. The state-aware policy is realized using two different approaches in the MAC: a regulatory formulation and an adaptive prioritization.

Our first approach is a regulatory MAC, which serves to reduce the traffic in the network. We use a local *state-based scheduler* to select a few critical data packets to send to the MAC. We analyze the impact of such a scheduler on the closed loop system, and show that there is a dual effect for the control signal, which makes determining the optimal controller difficult. We also identify restrictions on the scheduling criterion that result in a separation of the scheduler, observer and controller designs.

Our second approach is a prioritized MAC that uses state-based priorities called *Attentions*, to determine access to the network. We use a dominance protocol called *tournaments*, to evaluate priorities in a contention-based setting, and analyze the resulting performance of the MAC.

We also consider a NCS that uses a wireless multihop mesh network for communication between the controller and actuator. We design an optimal controller, which uses packet delivery predictions from a recursive Bayesian network estimator.

Acknowledgements

It is time for an overdue acknowledgement of those who have supported me and assisted in completing this thesis. I'm glad to have this opportunity to express my delight in knowing and having these people around me.

My supervisor Karl Henrik Johansson introduced me to networked control, and it is his encouragement and enthusiasm that has made it possible for me to continue my studies. I'm grateful for all the advice and knowledge, on many things, big and small, that you share so willingly with us all. My co-supervisor Henrik Sandberg has been a big support in my attempt to understand control theory. Thank you for being a generous teacher, with your books and your time. Collaborating with both of you has been a learning experience.

My other co-authors, whose work forms a part of this thesis are Lei Bao and Phoebus Chen. Lei is responsible for my current obsession with Bar-Shalom's results. It's your persistent 'why' that helped me understand. I like working through an idea with Phoebus. Even our long digressions have been fun. I'm learning from your attention to detail. I would also like to thank Maben Rabi, who has been a silent collaborator. Your ideas and discussions have influenced my work.

I enjoy the many, many discussions with Pan, PG, José, André, Pablo, Burak and Euhanna. You have undoubtedly contributed to this work. In addition to my supervisors, Pablo, Phoebus, Ather and Jim have read parts of this thesis. Thanks for your detailed comments. I'm very grateful to Aitor and José, for helping me with a few experiments in the last minute. Special thanks to Karin, Anneli and Kathy for patiently answering my questions on Sweden, migration, taxes and even shopping. Thanks to the rest of the team for making the work environment so lively, and for your company.

Shilpa and Marianna, thanks for resolutely ignoring my bad behaviour in the last couple of months, and not giving up on me. I would like to thank my vast family, for cheering me up everyday and encouraging me over skype, phone and email. I'm deeply grateful for all the support you give me, despite being mostly bewildered with my desire to study. I could not do without my mother, Sujatha, and brother, Sharan. You bring madness and joy to my life. Thanks to my dearest, Sandeep, for setting out on this journey with me. And for the spark, humor and much else that you bring to my life.

Chithrupa

Contents

Contents	v
1 Introduction	1
1.1 Multiple Access for NCSs	2
1.2 Motivating Examples	4
1.3 Problem Formulation	7
1.4 Outline	10
1.5 Contributions	11
2 Background	13
2.1 MAC	14
2.2 Optimal Stochastic Control	21
2.3 Networked Control Systems	25
3 Contention-based Multiple Access Architectures	29
3.1 Information Patterns for Different Architectures	29
3.2 Static MAC	32
3.3 Dynamic MAC	34
3.4 Adaptive MAC	35
4 State-based Schedulers	39
4.1 Contributions and Related Work	40
4.2 Preliminaries	41
4.3 Optimal Controller Design	46
4.4 Closed Loop System Architecture	55
4.5 Measurement-based Scheduler	60
4.6 Examples	61
4.7 Summary	67
5 Attention-Based Tournaments	69
5.1 Contributions and Related Work	69
5.2 Problem Formulation	70
5.3 Tournament Access Protocol	72

5.4	Performance Analysis	75
5.5	Simulation Results	78
5.6	Summary	81
6	Control over Mesh Networks	83
6.1	Introduction	83
6.2	Problem Formulation	86
6.3	Network Estimation and Packet Delivery Prediction	91
6.4	FPD Controller	96
6.5	Examples and Simulations	102
6.6	Discussion on Network Model Selection	105
6.7	Summary	106
7	Conclusions and Future Work	107
7.1	Conclusions	107
7.2	Future Work	108
A	Appendix	111
A.1	Appendix to State-based Schedulers	111
	Bibliography	115

Introduction

Control systems have seen rapid changes in the last forty years. Hybrid systems evolved with the introduction of computers and digital technology, and Networked Control Systems (NCSs) are emerging with the proliferation of networks and wireless systems. Computation, communication and sensing have become ubiquitous and inexpensive, and have provided us access to enormous amounts of data. Future applications in aerospace and transportation, information and networks, and intelligent machines and robotics, are expected to create complex systems that current control theory has not been developed to handle. Such systems are likely to be distributed and asynchronous, with packet-based interactions in networked environments. These factors are changing the basic architecture of a closed loop system. There is a need for new technology and theory to enable the construction of complex, distributed systems.

NCSs use a shared network for communication between sensors and controllers or controllers and actuators. The endeavour to use a wireless network to close the loop is part of a larger trend, which began with the ability to attach a little intelligence and a wireless transceiver on to all the devices around us. Thus, the technology to enable the future is almost ready, and already in use. Wireless sensor networks were meant to enable easy monitoring of the world around us, and a plethora of applications from military and biomedical sensing, environmental and building monitoring, and home surveillance have driven research on this front. The next wave of applications from air traffic control, intelligent transport and building systems have extended the scope of wireless sensor networks to actuation, using data collected from the sensor nodes. The result is a closed loop system, which uses a wireless network for its communications.

Applications in aerospace and intelligent transport systems clearly need wireless. However, other applications can also benefit from adopting these new technologies. Wireless networks eliminate costs incurred in the installation and maintenance of wiring. Wireless sensor networks are easy to deploy, making it possible to modify existing systems by adding sensors on the fly. Consequently, they offer possibilities of better sensing, which could lead to better control. Wireless is sometimes the only

choice, for example, in a network of mobile agents. A wireless system is an enabler of our information-driven world, where decisions can be taken anywhere and reach everywhere. There is also a compulsion to follow the trend; If wired sensors will soon be replaced with wireless nodes, how can closed loop systems remain wired?

Thus, future applications have created a need for wireless in the closed loop, and the technology to fulfill that need already exists. Then, can we plug wireless nodes into existing systems, and achieve working solutions for the future? No. There is a disconnect between classical control theory and the new closed loop architecture. Classical control theory assumes that there is a dedicated point-to-point link between the components of the closed loop system. Wireless networks can cause packet losses and delays. What should be done when the measurement packet does not arrive or the control packet is not delivered? Wireless links have limited bandwidth. How do we transmit real-valued measurements and controls through such links? Networks are interference constrained, as the shared medium does not support simultaneous transmissions. How do we accommodate multiple simultaneous transmission requests from nodes in the network? There is a need for new theory on complex, distributed, networked systems, which lies at the convergence of the classical theories of control, communication and computing. Networked control theory is part of the attempt to fill this void. It aims to design scalable and modular systems to deal with the explosion of information from sensor networks, and find distributed sensing, control and actuation methods.

The EU project, SOCRADES (Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices), dealt with some of these research issues. As part of this project, wireless networked control experiments were conducted in a mining plant at Boliden, which is shown in Fig. 1.1. The design of the closed loop system had to overcome the harsh radio environment, which introduced delays and packet losses in the wireless communication between the nodes.

1.1 Multiple Access for NCSs

Future applications and technology motivate us to consider networks that are characterized by high node densities, low data rates and energy constraints. In addition, sensor nodes in a control network are likely to have small and equal packet sizes. However, the packet generation rates need not be low, relative to the data rates supported by the network. Large networks, such as the ones we consider, are organized without much hierarchy, due to similar energy constraints on all the nodes. Network planning and deployment is a time and energy-consuming task, and impractical for such networks. This is especially true for applications which involve a frequently changing network, possibly due to the presence of mobile agents or other moving objects.

An example of the networks we consider in this thesis is illustrated in Fig. 1.2. The communication between the sensors and controllers, and the controllers and actuators, of M NCSs occurs over a shared network (\mathcal{N}). This network is also



Figure 1.1: Wireless sensors were introduced in the froth-flotation processes at the mining plant in Boliden. This is a real example of networked control, carried out in the EU project SOCRADES. (Courtesy of Boliden)

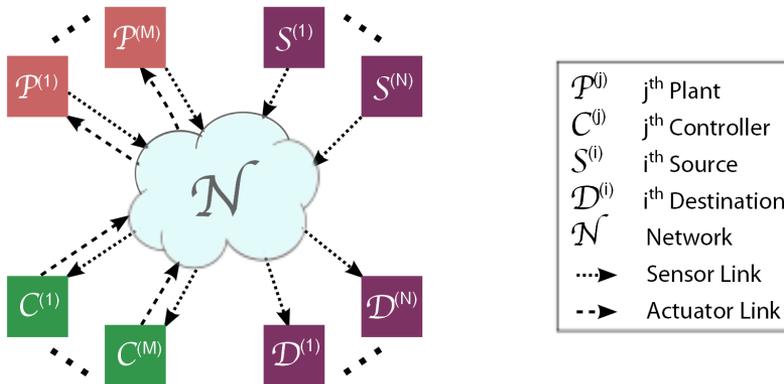


Figure 1.2: A heterogeneous multiple access network: M closed loop systems, consisting of a plant and a controller each, share the network with N other communication flows between generic source-destination pairs.

used for N other communication flows between the generic source-destination pairs depicted in the figure. The wireless medium, used by the network \mathcal{N} , is interference constrained, and permits only a single transmission at a time. Multiple simultaneous transmissions cause a collision, and the data from all the transmissions are lost. Thus, a Medium Access Controller (MAC) is required in every sensor's protocol stack to arbitrate access to the shared network.

The MAC can implement a contention-free or a contention-based multiple access method, each of which has its own challenges. Contention-free methods require a centralized network coordinator to assign a schedule for the users, which determines the order of network access. The schedule can only accommodate a fixed number of nodes commensurate to the capacity of the network, which is a limitation in a dense network. The schedule itself can be static or dynamic. A static schedule is never altered. This results in over-provisioning of resources, and is a severe limitation in a dense network. Dynamic resource allocations can solve this problem. Now, a new schedule is issued every few frames, based on the closed loop system performance. The problem with implementing such a scheduler in a wireless network is that the information is at the plant and not at the scheduler. This information needs to be collected, a schedule needs to be designed and then distributed to all the nodes in the network. In a wireless network, information collection and distribution is subject to the vagaries of the wireless medium, and some data may be lost. Furthermore, these operations constitute a significant overhead in an energy and resource-constrained network.

In contrast, contention-based methods facilitate an easy deployment on nodes. Contention-based multiple access methods work by resolving contention among simultaneous transmissions. Sometimes, the mechanism successfully resolves contention, and all the packets are transmitted at different time instants. However, this is not always the case. When the mechanism fails to resolve contention, a collision occurs and all packets are lost. The drawback is that such methods result in *random access*, rather than giving access to the closed loop system that needs to communicate the most. This could significantly deteriorate the performance of a closed loop system.

Thus, the design of a MAC for NCSs is a challenging problem, and calls for innovative solutions. The main objective of this thesis is to define architectures for closed loop systems, which use a contention-based multiple access network on the sensor link.

1.2 Motivating Examples

We consider three examples to illustrate the unsuitability of existing solutions. In the first example, we allow nodes to transmit whenever a packet is generated. The resulting collisions motivate the need for a MAC. In the next two examples, we consider a small network with a contention-free MAC and a contention-based MAC, respectively, and illustrate their limitations for NCSs.

The first example illustrates the limitations of a broadcast medium, such as a wireless network. We allow two systems to use a broadcast network, without an explicit access mechanism.

Example 1.1

We use a simple control process, with wireless sensing and actuating links, to regulate the level of water in a tank, as shown, to the left, in Fig 1.3. The links use a shared medium, but simply transmit when they generate a packet. We perform a reference tracking experiment, and observe the effect of interference from both the links. Next, we introduce another such control process, and repeat the same experiment with four interfering links, as shown, to the right, in Fig. 1.3.

The results obtained from these experiments conducted on real systems are shown in Fig. 1.4. In the first experiment, interference from the two links causes 14% of the packets to be lost. In the second experiment, interference from the four links using the same medium causes as many as 45% of the packets to be lost. Consequently, the performance of reference tracking in the second experiment is worse than in the first experiment.

Interference from just four links can cause severe packet losses in a broadcast medium. Thus, a shared network requires a protocol to arbitrate channel access among its users.

The second example illustrates the drawbacks of a contention-free multiple access method, for both static and dynamic resource allocations. It uses a well-known contention free multiple access method, i.e., Time-Division Multiple Access (TDMA).

Example 1.2

We consider many wireless tank processes, in a TDMA network, with a packet transmission time of T_{tx} seconds. Each wireless tank process is sampled every T_s seconds. Then, the maximum number of processes that can be supported by this network is given by $M = T_s/T_{tx}$.

Now, consider a static schedule for a network of M processes. Once the water level in the tank has been regulated, the nodes do not have much to convey until the next reference change. However, each node is still allotted a slot every T_s seconds. Next, consider a dynamic schedule for a network of $2M$ processes. Now, slots can be allotted to nodes that require the channel the most. However, the nodes that require access to the network must notify the scheduler. When the schedule is changed, all $2M$ nodes must be notified of the change. If any of these notifications are lost, two nodes may access the channel simultaneously, resulting in a collision.

This experiment shows that a contention-free MAC requires a centralized coordinator, and is not scalable as the number of nodes in the network increases. Also, static allocations result in a waste of scarce resources, and dynamic allocations are difficult to implement in a lossy medium.

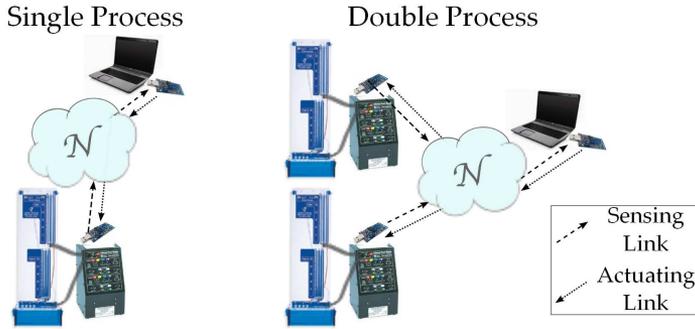


Figure 1.3: The setup to the left depicts a single coupled tank process which uses a wireless network to communicate with the controller. In the setup on the right side, two such processes share the network. The sensor nodes broadcast when they have data to transmit, without an explicit channel access mechanism in Example 1.1, or using a random access MAC in Example 1.3. The resulting collisions affect the closed loop system performance.

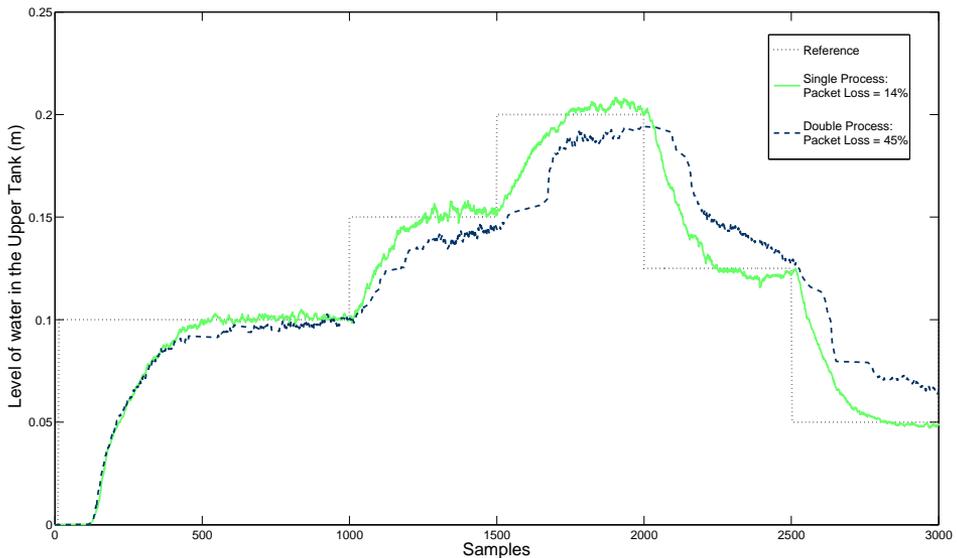


Figure 1.4: A comparison of two experiments: the solid line corresponds to the results of the reference tracking experiment with a single process using the broadcast medium, whereas the dashed line corresponds to the same experiment with two processes sharing the medium. In the second experiment, one of the processes suffers a packet loss of 45%, and consequently, its closed loop performance is worse.

The third example illustrates the drawbacks of a contention-based multiple access method, for NCSs. It uses the Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) protocol, and presents the effect of random losses on a closed loop system.

Example 1.3

We perform the same reference tracking experiment on the wireless process setups, shown in Fig. 1.3. Now, the nodes use CSMA/CA to determine access to the network.

With the single process setup, there are two links simultaneously using the network, and the performance of reference tracking in Fig. 1.5 shows a remarkable improvement over the experiments in Example 1.1, where the nodes did not use an explicit access mechanism. However, with two processes, and four links sharing the network, this protocol results in random losses. The reference tracking is noisier, and delayed, as shown in Fig. 1.5.

Despite the implementation advantage offered by contention free MACs, the randomness of multiple access makes this MAC unsuitable for critical closed loop systems.

Thus, there is a need to examine other possibilities for multiple access with NCSs. The MAC should result in an easily deployable, ad hoc, scalable solution, which can offer a performance guarantee for closed loop systems. Traditionally, MACs for control systems have been chosen based on the delay they introduce into the closed loop system. In this thesis, our focus is to design a MAC which can deal with congestion, not delay. The primary reason for the shift in focus is that the networks we consider are not fixed in size, and are likely to change frequently. Now, the MAC must be capable of adapting to the traffic in the network. To provide performance guarantees for NCSs, we explore the design of *state-aware* contention-based multiple access methods, as opposed to *agnostic* contention-based methods. The state-aware method permits a usage of the plant state to influence access to the network.

1.3 Problem Formulation

We consider a network of M control loops, with each loop consisting of a plant $\mathcal{P}^{(j)}$ and a controller $\mathcal{C}^{(j)}$ for $j \in \{1, \dots, M\}$. We refer to the link between the sensors and the controllers as the sensor link, and the link between the controllers and the actuators as the control link. The loops share access to a common medium on the sensor link, along with N other generic source-destination pairs. To focus on the implications of a MAC on the sensor link, we assume that the communication between the controllers and the corresponding actuators occurs over a point-to-point network, not a shared network. This is also a likely scenario of operation, as actuators require cables drawn from a power source, and are thus wired to the controllers.

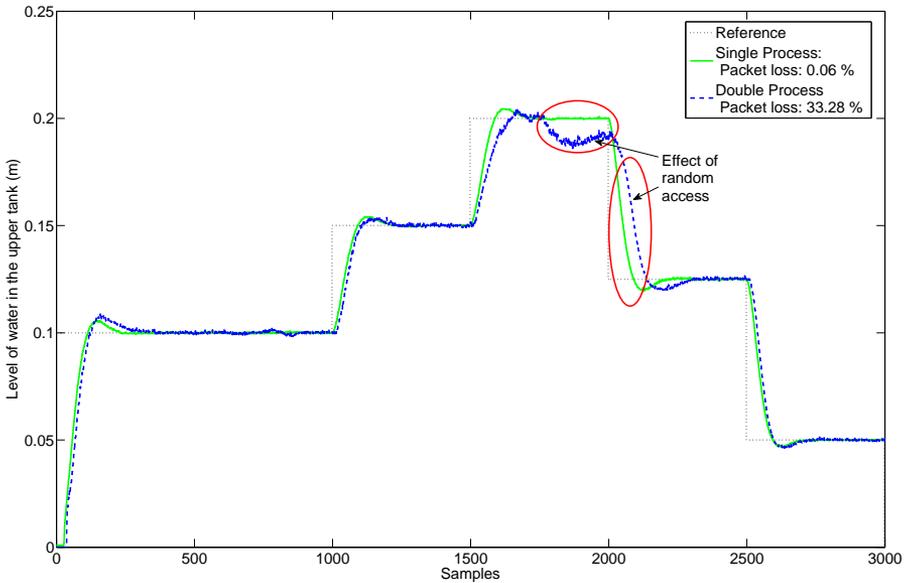


Figure 1.5: Effect of random access: When two processes share the medium and use CSMA/CA in the MAC, one of the processes suffers a packet loss of 33%. This reflects on its closed loop performance, and the reference tracking is much noisier. Also, it does not track the reference for a small region indicated above, and there is a delay compared to the single process.

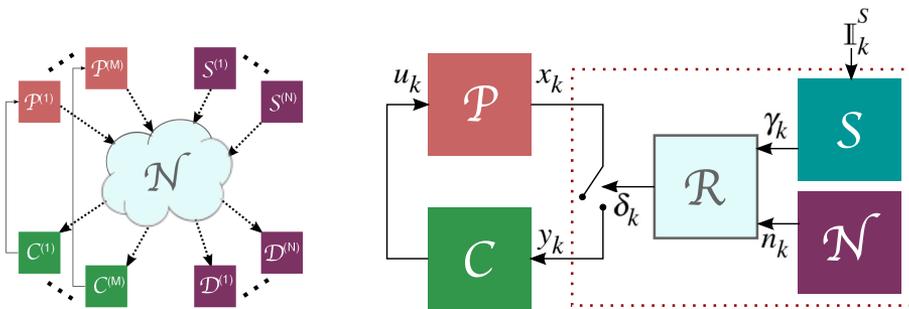


Figure 1.6: A multiple access network on the sensor link (the actuation links do not use the network), and a model for this network from the perspective of a single closed loop system in the network.

From the perspective of a single control loop, this system can be modelled as shown in Fig. 1.6. We drop the index j in this figure for simplicity. The block \mathcal{N} represents the network as seen by this loop, and the block \mathcal{R} denotes the contention resolution mechanism (CRM), which determines whether the control loop or the rest of the network gets to access the shared medium. Each of the blocks in the model are explained below.

Plant: The plant \mathcal{P} has state dynamics given by $x_{k+1} = Ax_k + Bu_k + w_k$, where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and w_k is i.i.d. zero-mean Gaussian with covariance matrix R_w . The initial state x_0 is zero-mean Gaussian with covariance matrix R_0 .

Scheduler: There is a local scheduler \mathcal{S} , situated in the sensor node, between the plant and the controller, which generates a channel access request, denoted by γ_k . The scheduling criterion is denoted by the policy f , which is defined on the information pattern of the scheduler $\mathbb{I}_k^{\mathcal{S}}$, and is given by $\gamma_k = f_k(\mathbb{I}_k^{\mathcal{S}})$. This information pattern consists of the system variables which are known to the scheduler, and can consist of a random variable, or the transmission history, or even the state of the plant. It determines whether the MAC is agnostic or state-aware.

Network: The network \mathcal{N} generates traffic, as denoted by n_k . The network can consist of heterogenous sources, and thus, n_k need not be generated by an information pattern similar to $\mathbb{I}_k^{\mathcal{S}}$.

Contention Resolution Mechanism: The block \mathcal{R} resolves contention between multiple simultaneous channel access requests. When the contention is resolved in favour of our control loop, $\delta_k = 1$, and otherwise 0. Thus, the MAC output δ_k is given by $\delta_k = \mathcal{R}(\gamma_k, n_k)$.

Controller: The measurement across the network is given by $y_k = \delta_k x_k$. The control law g is given by $u_k = g_k(\mathbb{I}_k^{\mathcal{C}})$, where the information available to the controller is given by $\mathbb{I}_k^{\mathcal{C}} = \{\mathbf{y}_0^k, \boldsymbol{\delta}_0^k, \mathbf{u}_0^{k-1}\}$. The bold font denotes a set of variables such as $\mathbf{a}_t^T = \{a_t, a_{t+1}, \dots, a_T\}$.

The controller tries to minimize the objective function (see 2.2.1), defined over a horizon N , and given by

$$J = \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{s=0}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s) \right],$$

where Q_0 , Q_1 and Q_2 are positive definite weighting matrices.

Observer: Since the state is not always available to the controller, some closed loop systems use an observer (\mathcal{O}) to generate an estimate of the state. This estimate is denoted $\hat{x}_{k|k}$ and given by $\mathbb{E}[x_k | \mathbb{I}_k^{\mathcal{C}}]$. The resulting estimation error is defined as

$\tilde{x}_{k|k} \triangleq x_k - \hat{x}_{k|k}$, with error covariance $P_k = \mathbb{E}[\tilde{x}_{k|k} \tilde{x}_{k|k}^T | \mathbb{I}_k^C]$.

In this thesis, we search for ways of influencing the channel access, i.e. δ_k , to minimize the control cost J and the estimation error P_k , in the presence of a multiple access network on the sensor link of the closed loop system. To this end, we address the following questions:

1. **Modelling:** What classes of multiple access architectures arise when we have different information patterns at the scheduler?
2. **Realization:** In what ways can we use these information patterns to help design MAC protocols?
3. **Analysis:** What is the impact of different multiple access methods on the design and performance of the closed loop system and the network?
4. **Design:** What choices of the scheduling policy f , the control policy g and the contention resolution mechanism \mathcal{R} must be made to achieve a certain performance?

1.4 Outline

The rest of the thesis is organized as follows. Chapter 2 provides a review of existing literature pertaining to multiple access methods and networked control systems. This chapter also introduces Linear Quadratic Gaussian (LQG) control theory, which we use to quantify the control performance in the rest of this thesis.

Chapter 3 provides a framework to answer the questions posed in this thesis. It provides an introduction to the state-aware MAC, and directly answers question one and two above. Each of the next two chapters outline a realization of the state-aware MAC.

Chapter 4 contains the main results of this thesis. It provides a theoretical analysis of state-aware MACs and answers a part of the third question listed above. It also presents a closed loop system architecture with design solutions as listed in the fourth question above.

Chapter 5 provides an example of a state-aware MAC whose performance analysis can be evaluated, and answers parts of the third and fourth questions listed above.

Chapter 6 addresses a different problem setup. Here, the closed loop system has a mesh network on the control link, and we derive the optimal controller which uses estimates provided by a network estimator. This work is part of a larger attempt to define and simplify the interface between a network and the application layer, or the control loop in our case.

Chapter 7 provides some notes for future work.

1.5 Contributions

This thesis is based on the following publications. The order of the authors indicates the workload, where the first author performed most of the work.

- Chapter 3 is an extension to some of the work presented in:

Chithrupa Ramesh, Henrik Sandberg and Karl H. Johansson, “LQG and **Medium Access Control**”, in Preprints of the 1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys2009), Venice, Italy, September 2009.

- Chapter 4 is based on the following papers, in addition to the paper listed above:

Chithrupa Ramesh, Henrik Sandberg and Karl H. Johansson, “Design of **State-based Schedulers** for a Network of Control Loops”, EES Department, KTH Royal Institute of Technology, Tech. Rep. TRITA-EE 2011 : 005, January 2011.

Chithrupa Ramesh, Henrik Sandberg, Lei Bao and Karl H. Johansson, “On the Dual Effect in State-based Scheduling of Networked Control Systems”, submitted to ACC 2011.

- Chapter 5 is based on the work presented in:

Chithrupa Ramesh, Henrik Sandberg and Karl H. Johansson, “Multiple access with **attention-based tournaments** for monitoring over wireless networks”, in Proceedings of the European Control Conference 2009, Budapest, Hungary, August 2009.

- Chapter 6 is based on the work presented in the following tech report:

Phoebus Chen, Chithrupa Ramesh and Karl H. Johansson, “Network Estimation and Packet Delivery Prediction for Control over Wireless Mesh Networks”, submitted to IFAC World Congress 2011, (with corresponding Tech. Rep. **TRITA-EE:043**).

Background

In this chapter, we present an overview of some of the previous work in NCSs, along with a review of basic concepts in multiple access technology and LQG control. [Hespanha et al. \(2007\)](#) define networked control systems as spatially distributed systems in which the communication between sensors, actuators and controllers occurs through a shared band-limited digital communication network.

Control theory was already in use in the design of large industrial manufacturing systems by the early 70s. Such applications required modular, easily scalable, distributed control systems with integrated diagnostics. Point-to-point wiring was expensive to install and maintain for such large systems. Thus, these systems had to use a common-bus architecture, making these the first applications of closed loop control through a shared network ([Halevi and Ray, 1988a,b](#)).

Research on such NCSs began as early as in the 70s, and continued through the late 80s and early 90s. The focus was then on common bus architectures, and the need for a suitable MAC protocol to arbitrate access to the shared medium. The MAC introduced a varying delay in the closed loop system, which was the focus of much study and comparison ([Ray, 1987](#); [Lian et al., 2001](#)). There were studies on the effect of delays in a closed loop system ([Nilsson, 1998](#)), and on analyzing closed loop stability under real-time delays ([Zhang et al., 2001](#)). The network quality of service (QoS) was defined in terms of delay, with modifications to network protocols proposed to guarantee a certain QoS. Some of these methods included rate adaptations ([Hong, 1995](#)) and the prioritization of real time traffic over other categories of traffic. Other methods suggested include the use of deadbands to reduce the communication load in shared networks ([Yook et al., 2002](#); [Otanez et al., 2002](#)).

More recently, applications such as robotics, aerospace and intelligent transportation systems have motivated the use of wireless networks for communication on the sensor and control links of a closed loop system. [Murray et al. \(2003\)](#) list NCSs as one of the future research directions in control, and provide an interesting overview of the motivations and origins of NCSs. Wireless networks introduce additional problems such as packet losses and bandwidth constraints, which have been

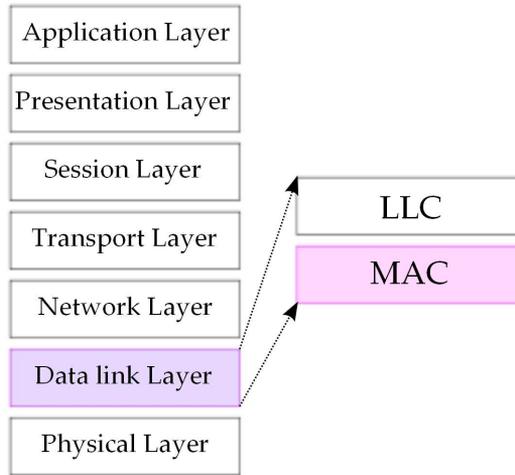


Figure 2.1: The multiple access protocol is implemented in the MAC layer of the network stack.

the focus of recent studies on NCSs (Hespanha et al., 2007). However, the original problems plaguing shared networks remain even today, as wireless networks are interference constrained. The medium cannot support multiple simultaneous transmissions, requiring mechanisms that arbitrate channel access. Wireless networks make it hard to implement some of the older token-based or prioritized channel access methods, which were the chosen solutions for control systems on common bus architectures (Lian et al., 2001). Thus, there is a need to re-examine some of this work and to identify medium access methods suited to control systems which use wireless networks. In the rest of this chapter, we present a review of relevant work culled from the long history of networked control systems. We begin with an introduction to multiple access methods and LQG control.

2.1 MAC

In this section, we introduce various multiple access techniques, and present some of the MAC protocols used in control networks over the last few years. When point-to-point channels are not available, broadcast channels must be used, with techniques to minimize interference from other users who share the same channel. Access schemes to such broadcast channels are known as *Multiple Access Protocols* (Rom and Sidi, 1990). These protocols are implemented in the Medium Access Control (MAC) layer, which forms a sub-layer to the second layer of the OSI network model. This is shown in Fig. 2.1.

Numerous multiple access protocols have been suggested for different applications until date. These are broadly classifiable as centralized or distributed proto-

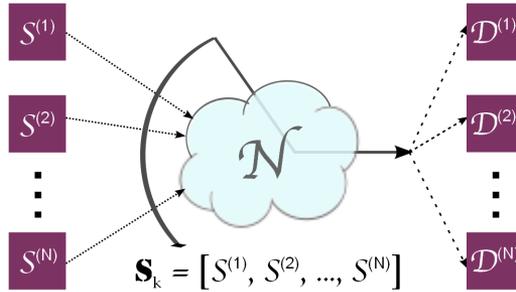


Figure 2.2: The network resources are partitioned and allocated using a schedule (\mathbf{S}_k), which determines the order of access, in a contention-free MAC. The allocations can be static, when the schedule does not vary with time, or dynamic, when it does.

cols, based on the hierarchies built into the protocols. Non-central protocols, where all nodes implement the same set of rules, with no centralized coordinator which has its own set of rules, are the ones that we focus on in this thesis. The protocols are also classified as contention-based or conflict free protocols with static or dynamic allocations. We described these terms in detail below.

2.1.1 Contention-free MACs

Contention-free protocols ensure that a transmission is always successful in the MAC, when the physical medium does not cause any losses. This is achieved by allocating the channel to the users in a static or dynamic manner, as illustrated in Fig. 2.2. These are also called as conflict-free protocols, and the order of channel allocations is referred to as a schedule.

The channel resources can be divided among the users in time, frequency or using codes, resulting in the following conflict-free methods:

- Time Division Multiple Access (TDMA)
- Frequency Division Multiple Access (FDMA)
- Code Division Multiple Access (CDMA)

We use TDMA as a typical example of conflict-free protocols in this thesis. These protocols typically result in a fixed delay, depending on the size of the network. If the resources required by each user are fixed, then such a protocol is not scalable (refer to Example 1.2), despite its desirous property of guaranteeing transmissions. Also, static allocations require over-provisioning of resources. To counter this disadvantage, a dynamic schedule can be drawn up for the network. However, this requires exchange of information between the users and a central scheduler, which is not easy to accomplish in a wireless network.

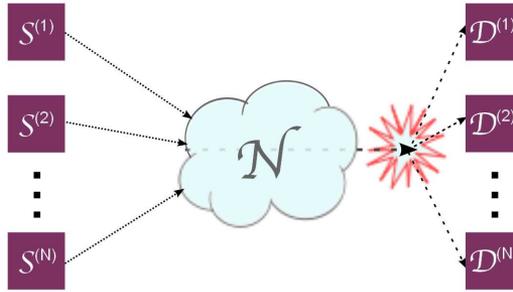


Figure 2.3: The nodes attempt to access the network resources, and any resulting contention is resolved using the protocol. The contention resolution can be successful, resulting in a transmission, or unsuccessful, resulting in a collision.

2.1.2 Contention-based MACs

In a contention-based protocol, a transmitting user is not always successful, as illustrated in Fig. 2.3. The protocol prescribes a mechanism to resolve contention for the same resources, in such a manner so as to eventually be successful in transmitting all the messages. In finite time, when the contention resolution mechanism fails to select a single message from the multiple simultaneous requests, it results in a collision. All the packets interfere with each other, and none are transmitted successfully. These collisions use up channel resources, and do not result in a transmission.

Some common families of contention-based protocols are listed below:

- Aloha-type protocols
- Carrier Sensing Multiple Access (CSMA)

The contention resolution mechanism can be static or dynamic, and we look at this classification in more detail in Chapter 3. Despite the obvious disadvantage of a possible collision, or the lack of a transmission guarantee, these protocols are popular in practice, as they are easy to deploy in an ad hoc manner. We look at some specific examples from these families of protocols.

Aloha

This is the first random access technique introduced in networking literature (Abramson, 1970). *Pure Aloha* is the basic protocol from this family. The idea is very simple: a node attempts transmission as soon as it generates a data packet. If the transmission is unsuccessful, due to a collision with other packets, the data packet is scheduled for retransmission at a random time in the future, independent of other users. We define the throughput as the fraction of time that useful information is

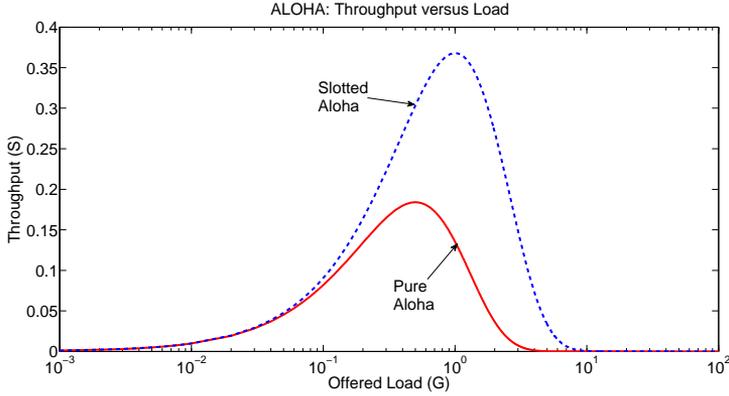


Figure 2.4: The throughput versus load plot for pure Aloha and Slotted Aloha.

carried on the network. Then, the throughput S of this protocol has been derived (Rom and Sidi, 1990) as

$$S = Ge^{-2G}, \quad (2.1)$$

where, $G \triangleq gT$. In this expression, g is the packet arrival rate and T is the packet transmission time.

Slotted Aloha is a popular variation of this protocol, where the transmission time is divided into slots of duration T , and users are restricted to start a transmission only at the slot boundaries. This modification ensures a smaller vulnerability period for a packet, and results in a higher throughput, given by $S = Ge^{-G}$. Note that this improvement is achieved at a cost. Slotted Aloha requires synchronization between the nodes in the network. A plot of the throughput versus load for both these protocols is given in Fig. 2.4.

These protocols exhibit poor performance due to the "impolite" behaviour of users, who do not wait for an idle channel before commencing transmission. *Carrier sensing* is a mechanism that permits nodes in the network to monitor the channel for transmissions from other nodes. Carrier Sensing Multiple Access (CSMA) protocols sense the channel, and only transmit when the channel is idle. If the channel is busy, they resort to different mechanisms for retransmission. Some of these are described below.

***p*-persistent CSMA**

Non-persistent CSMA is a simple protocol, where a user who finds the channel busy schedules a transmission to a random time in the future. The problem with this protocol is that the channel is sometimes idle, despite there being a number of users with a packet to transmit. To remedy this, *1-persistent CSMA* requires a user who finds the channel busy, to wait persistently and transmit as soon as the channel becomes idle. This leads to guaranteed collisions when there are many users with a

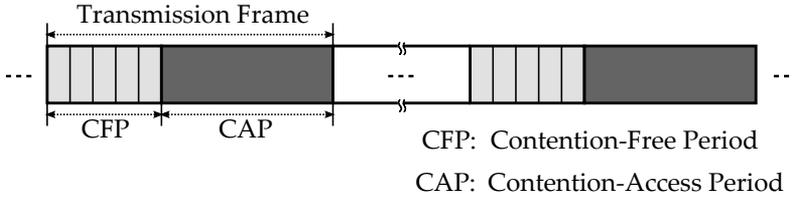


Figure 2.5: In a Hybrid MAC, the transmission time is divided into frames, and each frame consists of a contention-free period and a contention-access period.

packet to transmit. A tradeoff between both these protocols is *p-persistent CSMA*, where a user who finds the channel busy persists to wait and transmits as soon as the channel becomes idle with the persistence probability p . With probability $1 - p$, the user delays transmission by τ seconds. These protocols were first proposed and analyzed in Kleinrock and Tobagi (1975). We use this protocol in the thesis as an abstraction of the more complicated *CSMA/CA*, which is described below.

CSMA/CA

This is a more sophisticated CSMA protocol, where CA stands for collision avoidance. The operation is identical to before, and if the channel is found busy, the packet is scheduled for transmission sometime in the future. However, here, the retransmission mechanisms are well defined, and the most popular scheme is the exponential backoff mechanism which is described below. At each packet transmission, a backoff time is uniformly chosen in the range $(0, w - 1)$, where w is the maximum value of the backoff time, and is called the contention window. In the exponential backoff scheme, the contention window is doubled with the number of failed transmissions. This allows the user to increase its MAC delay and waiting time proportional to the perceived traffic load of the network. Thus, like in most other contention-based methods, the transmission delay can be almost non-existent with low traffic in the network, and scale upwards as the traffic increases. The problem with these methods is not the delay, but the saturation of the network with high traffic, as this can result in zero throughput, effectively choking all the traffic in the network. One implementation of this protocol can be found in the Distributed Coordination Function used in the IEEE 802.11 standard, and the throughput has been analyzed by Bianchi (2000).

2.1.3 Hybrid MAC

Some protocols combine the advantages of both conflict free and contention-based methods with a hybrid MAC. Here, the transmission time is divided into frames. Each frame is further divided into a contention-free period and a contention-access period, as shown in Fig. 2.5. The slots in the contention-free period are reserved for nodes which request for them using the contention-access period. Securing an

allotment of the contention-free slots guarantees multi-packet transmissions. A node is also permitted to use only the contention-access period, if desired. The hybrid MAC is used in the IEEE 802.15.4 standard (IEEE, 2006) and a variation of this idea is used in the WirelessHART standard (HART Communication Foundation, 2007).

2.1.4 Protocols

In this section, we briefly outline the MAC layer of a few protocols, relevant to control systems.

- **CAN Bus:** This protocol is used by the smart distributed system (SDS), DeviceNet and CAN Kingdom. CAN is a serial communication protocol, which offers good performance for time-critical industrial applications (Robert Bosch GmbH, 1991). Messages are allotted different static priorities, which are used to arbitrate access to the common bus. The arbitration is implemented using a bit-dominance strategy, which is described below. A node with a packet to transmit, attempts to secure the transmission slot by winning a tournament slot. In the tournament slot, nodes transmit their priority bits, starting with the most significant bit. Other nodes listen to the network during their recessive bits, and drop out of contention when they hear a dominant bit, as it indicates that a node of higher priority desires access to the channel. The last node remaining secures the transmission slot. This protocol guarantees that the allotted priorities are observed during contention resolution, but it is hard to implement this protocol in a wireless network. An adaptation for wireless networks is discussed in Pereira et al. (2007), and a modification to the priority allocation is presented in Chapter 5.
- **Token Bus:** This protocol is used by process field bus (PROFIBUS), manufacturing automation protocol (MAP), ControlNet and fiber distributed data interface (FDDI). The nodes in a token bus network are arranged logically into a ring, and each node knows the network address of its predecessor and successor in the ring. The node with the token is permitted to transmit until the end of the data packet, or until it runs out of time, whichever occurs earlier. Then, the token is forwarded to the next node in the logical ring. This mechanism guarantees a maximum waiting time before transmission, and makes the network deterministic. However, the token is susceptible to the vagaries of the wireless medium, such as packet losses and hidden terminals, and is hence not popular in wireless networks.
- **Distributed Coordination Function:** This protocol is implemented in the IEEE 802.11 standard and uses CSMA/CA with exponential backoff. However, it results in *random access*, which could significantly deteriorate the performance of a closed loop system (Liu and Goldsmith, 2004). It is also hard to analyze, as shown by Bianchi (2000).

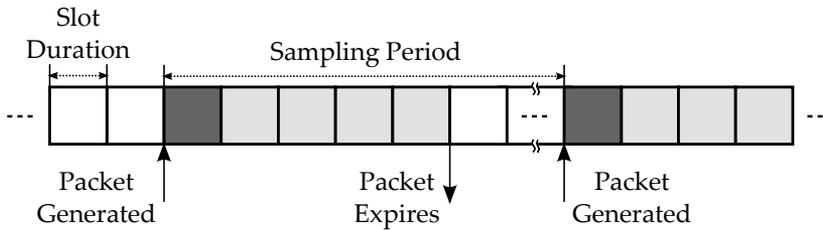


Figure 2.6: A packet is generated at the start of the time slot. The transmission time for a packet is equal to the slot time. The sampling period is much greater than the slot duration, and the data packet is either delivered or dropped by the expiry instant, which occurs before the next sampling instant.

- Beacon-enabled Hybrid MAC: This name is used to denote the MAC layer specified by the IEEE 802.15.4 standard (IEEE, 2006) for wireless sensor networks. It uses Slotted CSMA/CA in the contention-access period, and TDMA in the contention-free period. The PAN coordinator is responsible for allocating available slots in the contention-free period, also known as GTS slots.
- WirelessHART Hybrid MAC: This is another example of a hybrid MAC, as specified by the WirelessHART protocol (HART Communication Foundation, 2007). Here, the transmission time is divided into slots, and the MAC uses TDMA. However, each slot can be allotted to more than one node, and contention is permitted within the slot using CSMA/CA.

2.1.5 Time Scales

A natural unit of time for the description of the MAC is the time slot. In some protocols such as TDMA or p -persistent CSMA, the slot length can be equal to the packet transmission time. In other protocols, such as CSMA/CA, the slot length is much smaller than the packet transmission time. In most of this thesis, we use a p -persistent MAC as an abstraction of the contention resolution mechanism, and accordingly assume that the control network time slot is equal to the transmission time of a single data packet.

At the application layer, a more natural unit of time for the description of discrete time control theory is the sampling time of the physical process. In the rest of this thesis, we assume that the sampling time is much larger than the slot time, as shown in Fig. 2.6. The data packet is delivered or lost by the expiry instant, which occurs prior to the next sampling instant. We do not consider an overlap in the time scales of the MAC and the physical process.

2.2 Optimal Stochastic Control

In this section, we present an introduction to optimal stochastic control, specifically the LQG problem.

2.2.1 LQG Control

This section deals with the general control problem of minimizing the mathematical expectation of a quadratic criterion in the state and control variables of a linear multiple input multiple output (MIMO) plant. The theory of optimal control and estimation present a unified design procedure for the LQG problem, as outlined in Athans (1971). We consider the case of partial state information presented in Åström (1970), but modify the presentation along the lines of the proof in Bar-Shalom and Tse (1974).

Consider a linear MIMO system with additive white, but not necessarily Gaussian noise, given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (2.2)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and w_k is independent and identically distributed (i.i.d.) zero-mean with covariance matrix R_w . The initial state x_0 is zero-mean with covariance matrix R_0 . Also, consider a general measurement model

$$z_k = h_k(x_k, v_k), \quad (2.3)$$

where v_k is the measurement noise with known, but arbitrary, statistics. The measurement noise sequence is restricted to be independent of the process noise w_k . The objective function, defined over a horizon N is given by

$$J = \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{s=0}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s) \right], \quad (2.4)$$

where Q_0, Q_1 and Q_2 are positive definite weighting matrices. Note that the cost to be minimized is quadratic. We define a few terms before we present a derivation of the optimal controller for the above system. Also, we use the bold font to denote a set of variables such as $\mathbf{a}_t^T = \{a_t, a_{t+1}, \dots, a_T\}$.

2.2.2 Dual Effect

The control u_k might affect the future state uncertainty, in addition to its direct effect on the state. This is called the dual effect of control (Feldbaum, 1961).

Definition 2.1 (No Dual Effect (Bar-Shalom and Tse, 1974)). *A control signal is said to have no dual effect of order $r \geq 2$, if*

$$\mathbb{E}[M_k^T | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}] = \mathbb{E}[M_k^T | x_0, \mathbf{w}_0^{k-1}, \mathbf{v}_0^k],$$

where $M_k^T = \mathbb{E}[\|x_k - \mathbb{E}[x_k | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}]\|^r | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}]$ is the r^{th} central moment of x_k conditioned on the information at the controller $\{\mathbf{z}_0^k, \mathbf{u}_0^{k-1}\}$.

2.2.3 The Certainty Equivalence Principle

There are three closely related terms: a certainty equivalent controller, the certainty equivalence principle and the separation principle. We define each of these with respect to the deterministic optimal controller, with full state information, for the above problem setup (Åström, 1970; Bar-Shalom and Tse, 1974).

Definition 2.2 (Certainty Equivalent Controller). *A certainty equivalent controller uses the deterministic optimal controller, with the state x_k replaced by the estimate $\hat{x}_{k|k}$, as an ad hoc control procedure.*

Sometimes, there is no loss in optimality in using a certainty equivalent controller. Then, we say that the Certainty Equivalence Principle holds.

Definition 2.3 (Certainty Equivalence Principle). *The certainty equivalence principle holds if the closed-loop optimal controller has the same form as the deterministic optimal controller with the state x_k replaced by the estimate $\hat{x}_{k|k} = \mathbb{E}[x_k | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}]$.*

The separation property is weaker than the certainty equivalence property. Here, the form of the optimal controller can be different from that of the deterministic optimal controller. However, the control signal is still derived using the estimate of the state alone.

Definition 2.4 (Separation Principle). *The closed loop optimal control has the separation property if it depends on the data only through the estimate $\hat{x}_{k|k}$.*

2.2.4 An Expression for the LQG Cost

We now present the main result, an expression for the minimizing controller and the resulting cost. This section closely follows the proof of the theorem presented in Bar-Shalom and Tse (1974) and the last chapter of Åström (1970).

Theorem 2.1. *The optimal stochastic control for the system with linear dynamics (2.2), measurement equation (2.3) and cost (2.4), has the certainty equivalent property, if and only if, the control has no dual effect of second order. The minimizing control policy is then given by $u_k = -L_k \hat{x}_{k|k}$, and the resulting control cost is given by*

$$J_0 = \hat{x}_0^T S_0 \hat{x}_0 + \text{tr}\{S_0 P_0\} + \sum_{n=0}^{N-1} \text{tr}\{S_{n+1} R_w + (L_n^T (Q_2 + B^T S_{n+1} B) L_n) P_{n|n}\}, \quad (2.5)$$

where L_k and S_k are given by (2.9) and (2.10) respectively.

Proof. We present the classical proof using dynamic programming for the given problem setup, and show that a recursive solution to the optimization problem at

any time step can be found only if the control signal has no dual effect of order 2. We also derive the form of the optimal controller and the resulting control cost, when there is no dual effect of the controls.

We begin our proof with a derivation of the functional equation for our problem. Consider a time k such that $0 \leq k < N$. The expected loss can be written as a sum of two terms

$$\begin{aligned} & \mathbb{E}[x_N^T Q_0 x_N + \sum_{n=0}^{N-1} (x_n^T Q_1 x_n + u_n^T Q_2 u_n)] \\ &= \mathbb{E}[\sum_{\ell=0}^{k-1} (x_\ell^T Q_1 x_\ell + u_\ell^T Q_2 u_\ell)] + \mathbb{E}[x_N^T Q_0 x_N + \sum_{s=k}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s)]. \end{aligned}$$

It is clear that the first term does not depend on the control signals \mathbf{u}_k^{N-1} . Thus, to find the optimal control signals from k until $N-1$, we need to minimize the second term. We do this as

$$V_k = \min_{u_k, \dots, u_{N-1}} \mathbb{E}[x_N^T Q_0 x_N + \sum_{s=k}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s) | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}],$$

where the net cost can be written as $J_0 = \mathbb{E}[V_0]$. Using the same idea as before, we rewrite this expression as

$$\begin{aligned} V_k &= \min_{u_k} x_k^T Q_1 x_k + u_k^T Q_2 u_k \\ &+ \min_{u_{k+1}, \dots, u_{N-1}} \mathbb{E}[x_N^T Q_0 x_N + \sum_{s=k+1}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s) | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}] \\ &= \min_{u_k} \mathbb{E}[x_k^T Q_1 x_k + u_k^T Q_2 u_k + V_{k+1} | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}]. \end{aligned} \quad (2.6)$$

We now have the Bellman equation, which is a recursive minimization problem, with the initial condition given by the functional equation V_N at $k = N$.

For the optimal control policy, which minimizes the quadratic cost J (2.4), we try to find a solution to the Bellman equation, of the form

$$V_k = \mathbb{E}[x_k^T S_k x_k | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}] + s_k, \quad (2.7)$$

where S_k is a positive semi-definite matrix and both S_k and s_k are not functions of the applied control signals \mathbf{u}_0^{k-1} . We now prove that a solution of this form can be found for the system under consideration.

At time N , the functional has a trivial solution (2.7) with $S_N = Q_0$ and $s_N = 0$. This solution can be propagated backwards, in the absence of a dual effect. To show this, we use the principle of induction, and assume the solution to hold at time $k+1$.

Then, at time k , we have

$$\begin{aligned}
V_k &= \min_{u_k} \mathbb{E}[x_k^T Q_1 x_k + u_k^T Q_2 u_k + x_{k+1}^T S_{k+1} x_{k+1} + s_{k+1} | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}] \\
&= \min_{u_k} \mathbb{E}[x_k^T (Q_1 + A^T S_{k+1} A) x_k | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}] \\
&\quad + u_k^T (Q_2 + B^T S_{k+1} B) u_k + \hat{x}_{k|k}^T A^T S_{k+1} B u_k + u_k^T B^T S_{k+1} A \hat{x}_{k|k} \\
&\quad + \text{tr}\{S_{k+1} R_w\} + \mathbb{E}[s_{k+1} | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}],
\end{aligned} \tag{2.8}$$

where $\hat{x}_{k|k} = \mathbb{E}[x_k | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}]$ is the minimum mean-square error (MMSE) estimate (Kailath et al., 2000) of the state obtained from the measurements. The solution to the above minimization problem is given by

$$\begin{aligned}
u_k &= -L_k \hat{x}_{k|k}, \\
\text{where, } L_k &= (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A.
\end{aligned} \tag{2.9}$$

Substituting the expression for u_k into V_k gives us a solution of the form in (2.7), with

$$\begin{aligned}
S_k &= Q_1 + A^T S_{k+1} A - A^T S_{k+1} B (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A, \\
s_k &= \text{tr}\{S_{k+1} R_w\} + \mathbb{E}[s_{k+1} | \mathbf{z}_0^k, \mathbf{u}_0^{k-1}] \\
&\quad + \text{tr}\{A^T S_{k+1} B (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A P_{k|k}\},
\end{aligned} \tag{2.10}$$

where the matrix S_k is positive semi-definite and not a function of the applied controls \mathbf{u}_0^{k-1} . The scalar s_k is not a function of the applied controls \mathbf{u}_0^{k-1} if and only if $P_{k|k}$ has no dual effect (Bar-Shalom and Tse, 1974). Since the optimal control signal (2.9) is a function of only the estimate $\hat{x}_{k|k}$, the Certainty Equivalence Principle holds.

Using the above equations, we can find an expression for the control cost using

$$\begin{aligned}
J_0 &= \mathbb{E}[V_0] = \mathbb{E}[\mathbb{E}[x_0^T S_0 x_0 | z_0] + s_0] \\
&= \hat{x}_0^T S_0 \hat{x}_0 + \text{tr}\{S_0 P_0\} + \mathbb{E}\left[\sum_{s=0}^{N-1} \text{tr}\{S_{s+1} R_w\}\right] \\
&\quad + \mathbb{E}\left[\sum_{s=0}^{N-1} \text{tr}\{A^T S_{s+1} B (Q_2 + B^T S_{s+1} B)^{-1} B^T S_{s+1} A P_{s|s}\}\right] \\
&= \hat{x}_0^T S_0 \hat{x}_0 + \text{tr}\{S_0 P_0\} + \mathbb{E}\left[\sum_{s=0}^{N-1} \text{tr}\{S_{s+1} R_w\}\right] \\
&\quad + \mathbb{E}\left[\sum_{s=0}^{N-1} \text{tr}\{(L_s^T (Q_2 + B^T S_{s+1} B) L_s) P_{s|s}\}\right],
\end{aligned}$$

where the second last equation was obtained by substituting for s_0 and the last equation was obtained using (2.10) and (2.9). Thus, we obtain the expression given in (2.5). \square

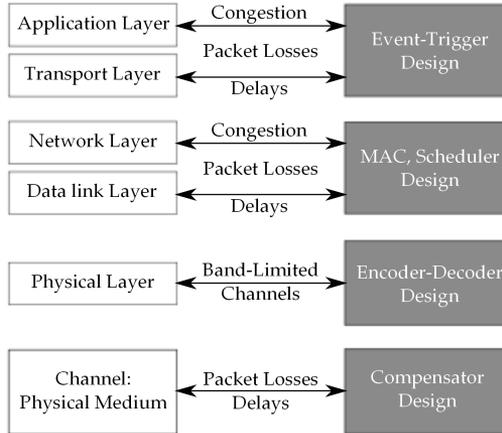


Figure 2.7: The network protocol is implemented as a stack consisting of some of the layers depicted above. We relate the problems tackled in NCSs literature to the layers in the stack.

2.3 Networked Control Systems

In this section, we present an overview of the work in NCSs. Most of the topics presented here lie at the intersection of control and communication theories, through the exchange of information from interconnected dynamical systems over imperfect channels (Hespanha et al., 2007).

Wireless networks are quite different from the perfect point-to-point channels typically assumed in classical control theory. These networks drop packets, or introduce delays into the closed loop system. Much of the literature on NCSs analyzes the impact of packet losses and delays on the closed loop system. In our presentation of this work below, we look at the impact of each layer in a typical protocol stack on NCSs, as shown in Fig. 2.7.

2.3.1 Control Design for Lossy Networks

The wireless medium is inherently lossy, and may cause packets to be dropped along the sensor link or the control link. Network protocols such as TCP have been designed to cope with such failures through the use of a retransmission policy, which ensures that the packet is eventually delivered to the destination. However, this property may not be very useful for a closed loop system, as these systems are not delay tolerant. A packet that is delivered beyond an acceptable delay is treated as a lost packet. The impact of such packet losses on optimal control and estimation has been well studied by Matveev and Savkin (2003), Smith and Seiler (2003), Schenato et al. (2007), Gupta et al. (2007), and others. Most of these studies consider packet losses on the sensor and control links, which they model with binary ran-

dom variables. Typically, the packet loss indicators on both links are assumed to be i.i.d. Some of these studies also consider packet loss distributions with correlations (Gupta et al., 2007).

The main result from this area of work relevant to this thesis is the result on separation. Many of the above authors have established that the separation principle holds under both i.i.d and correlated packet drop sequences, so long as the applied control input is made available to the observer. This is possible with a network protocol that returns an acknowledgement of a packet delivery. When such an acknowledgement is not available, the applied control signal is unknown. Then, the separation principle no longer holds, and the optimal control policy is hard to define (Schenato et al., 2007). Other results in this area include derivations of a critical probability of packet loss, below which the estimation error at the observer does not remain bounded, and also of upper and lower bounds on the achievable error covariance matrices.

2.3.2 Encoder Design for Limited Data-Rate Channels

Any transmission medium has a finite bandwidth. However, the wireless medium is scarce, making the bandwidth constraint more severe. Thus, sensor or control links are constrained to use the limited data rates supported by the wireless channel. Now, it is not possible to transmit a real-valued state or control signal on such a channel. The real values must be quantized at the sender and estimated at the receiver, to reconstruct the original values. These tasks are performed by an encoder and a decoder, respectively. Many studies analyzing the design of an optimal encoder and decoder for a closed loop system were carried out by Bansal and Basar (1989), Borkar and Mitter (1997), Tatikonda et al. (2004), Nair et al. (2007) and others.

An important result in this area is the establishment of the property of certainty equivalence with a state-based encoder and decoder in the closed loop. Tatikonda et al. (2004) construct an auxiliary system from the unforced process, with an equivalent encoder-decoder pair. They show that the sigma-fields corresponding to the quantized outputs for this system are nested within the sigma-fields corresponding to the quantized outputs in the complete system. They also derive the same estimate of the unforced state in both setups, thus establishing that the certainty equivalence principle holds. Bao et al. (2010) have commented on the importance of side information in realizing an architecture where the certainty equivalence principle holds. This relates to the availability of the applied control signal at the observer. Other results include designs for the optimal encoder-decoder pair.

2.3.3 MAC Design for Delayed and Congested Links

A MAC protocol determines the channel access strategy for multiple nodes sharing the same medium. Both contention-free and contention-based MACs introduce a delay in the closed loop system. In addition, the retransmission scheme used by a contention-based MAC determines the traffic contributed to the network by the

MAC. A limited retransmission policy could reduce the reliability, or the probability of a successful transmission. Excessive retransmissions can increase the traffic in the network, leading to congestion, and consequently, zero throughput from the MAC.

The most important drawback of MACs was considered to be the time-varying delay they introduced into closed loop systems. The impact of delays on a closed loop system was studied by Nilsson (1998), who considered different time varying delays on the sensor and control links. The delays are assumed to be independent random variables with known probability distributions. However, with the use of time-stamped data in the network, the delays are known when the measurement arrives. Nilsson derives the optimal control policy, and shows that it is a delay-dependent function of the current state and the past control. He also extends this to correlated delays across links. Much of the earlier work presents a comparison between various MAC protocols, and evaluates their suitability to control applications, such as in Lian et al. (2001) and Ray (1987). Lian et al. (2001) compare CAN, Fieldbus and the DCF of IEEE 802.11, and conclude that CAN is better suited to networks with short and prioritized messages, while token bus is better suited to networks with large messages. However, neither of these can be adapted easily to wireless networks.

In this thesis, we look at the design of a suitable wireless MAC for NCSs. The time constants of systems considered here are low enough to ensure that the delay introduced in the MAC during regular operation can be neglected. However, when the traffic load in the network is high, the throughput of a contention-based MAC can reduce to zero. Now, the MAC introduces infinite delay into the closed loop. Retransmissions do not solve the problem, as they add to the traffic and worsen the congestion in the network. Thus, the MAC must be designed to reduce the traffic in the network when there is congestion. This is the primary focus of the work presented in this thesis.

2.3.4 Event-based System Design for Congested Networks

In NCSs, where the sensor-controller link operates over an interference constrained network, there is much benefit in reducing the sampling rate or the data generated by the sensor (Yook et al., 2002; Otanez et al., 2002). An optimal method on reducing the communication load for NCSs was presented in Xu and Hespanha (2004). This approach has driven the design of event-based sampling systems (Rabi, 2006; Tabuada, 2007), which have been shown to outperform periodically sampled systems under certain conditions (Åström and Bernhardsson, 2002). We approach the same problem from a different perspective in Chapter 4, but one that leads to a *network-aware* design of event triggering methods.

Åström and Bernhardsson (2002) have shown that event-based systems are mathematically equivalent to Lesbegue sampling, which may be a useful alternative over periodic sampling or Reimann sampling, for multi-rate sampling and networked systems. A generalization of this work was presented in Rabi (2006), where the joint choice of stopping times and feedback control signals were derived for certain prob-

lems. [Tabuada \(2007\)](#) presents a Lyapunov-based approach for event-triggering. The relative error criterion determines the event-triggering instants, and guarantees stability for a bounded input disturbance. This approach has been extended and integrated with the dead-band approach on event-triggering in recent work by [Donkers and Heemels \(2010\)](#). There has also been some work on a self-triggering approach ([Anta and Tabuada, 2010](#)), where the node decides on the next sampling instant in such a manner so as to guarantee stability. [Molin and Hirche \(2009\)](#) deal with the optimal closed loop design for event-based systems, which is related to the work presented in Chapter 4.

Most of the above work on event-based systems assumes a perfect channel. However, it is well-acknowledged that packet losses and delays can severely affect the performance of event-based systems. To take into account the effects of an imperfect channel, we need to be able to analyze event-based traffic in a shared network. This is not easy to accomplish as the multiple access channel introduces correlations between independent data packets. Some of the earlier work in this area includes an empirical analysis of event based systems with CSMA/CA ([Cervin and Henningsson, 2008](#)), which highlights the difficulties in analyzing such a MAC, and a complete analysis with ALOHA ([Rabi and Johansson, 2009](#)). [Henningsson and Cervin \(2010\)](#) present a simple model, based on a steady state analysis of event-based systems using a shared medium. More recently, [Blind and Allgöwer \(2011a,b\)](#) have successfully analyzed event-based systems which use Aloha and Slotted Aloha, but with an event-triggering law that has been chosen to result in independent packets.

Contention-based Multiple Access Architectures

This chapter presents a framework for the design of MAC for NCSs. We examine different architectures for MACs which result in different properties for the channel access. Most of the MAC protocols in use are designed in accordance with the OSI model, which emphasizes modular layers and sub-layers, in order to achieve interoperable protocols that function independent of the application layer. Thus, the MAC protocols we examined in the previous chapter are agnostic to the control systems that use them. However, these protocols are not well suited for NCSs.

In this chapter, we look at ways to influence the randomness of channel access in contention-based MACs. We find that we can classify contention-based multiple access architectures as static, dynamic or adaptive to the application layer. We examine each of these architectures in detail, and analyze the impact on the closed loop system design. In particular, we identify the optimal estimator and control policy for each of these architectures, which minimize a quadratic cost. These architectures are not all new, and we identify examples from the literature, wherever possible, that satisfy our definition. One of these classes, the adaptive MAC, is a state-aware MAC, which adapts its channel access probability to the state of the plant. This enables the MAC to use the plant state to influence the randomness of its channel access.

We also look at how to realize these architectures effectively, and the next two chapters present an analysis of a realization each of the adaptive MAC.

3.1 Information Patterns for Different Architectures

Consider a model of a closed loop system which has a contention-based multiple access network on its sensor link, as shown in Fig. 3.1. The plant \mathcal{P} has state dynamics given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (3.1)$$

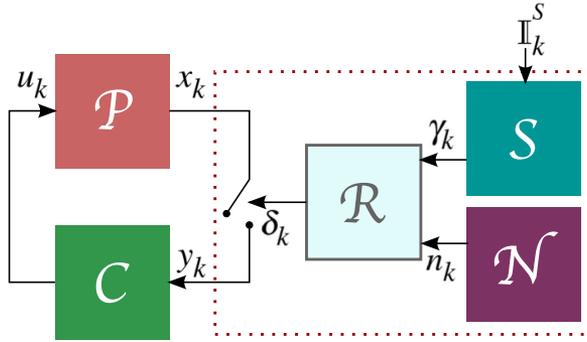


Figure 3.1: A multiple access network model from the perspective of a single closed loop system in the network

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and w_k is i.i.d. zero-mean Gaussian with covariance matrix R_w . The initial state x_0 is zero-mean Gaussian with covariance matrix R_0 .

A local scheduler \mathcal{S} , situated in the sensor node between the plant and the controller, generates a channel access request denoted by γ_k . The scheduling criterion f uses the information pattern of the scheduler $\mathbb{I}_k^{\mathcal{S}}$ to generate the channel access request, with

$$\gamma_k = f_k(\mathbb{I}_k^{\mathcal{S}}), \quad (3.2)$$

where the information pattern can take different values, as described later in this section.

The MAC generates an output δ_k , in response to the channel requests, γ_k , from this plant, and n_k , from the rest of the network. The contention-resolution mechanism \mathcal{R} determines the MAC output, as given by

$$\delta_k = \mathcal{R}(\gamma_k, n_k). \quad (3.3)$$

The measurement across the network is given by $y_k = \delta_k x_k$. The control law g denotes an admissible policy for the finite horizon N defined on the information pattern of the controller, $\mathbb{I}_k^{\mathcal{C}}$, and is given by

$$u_k = g_k(\mathbb{I}_k^{\mathcal{C}}), \text{ where, } \mathbb{I}_k^{\mathcal{C}} = \{\mathbf{y}_0^k, \delta_0^k, \mathbf{u}_0^{k-1}\}. \quad (3.4)$$

The objective function, defined over a horizon N is given by

$$J(f, g) = \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{s=0}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s) \right], \quad (3.5)$$

where Q_0, Q_1 and Q_2 are positive definite weighting matrices.

In some of the sections in this chapter, the controller is shown to be composed of an observer and a time-varying gain block. The input to the observer is the signal

$y_k = \delta_k x_k$. The observer generates the estimate $\hat{x}_{k|k}$ as given by

$$\hat{x}_{k|k} = \bar{\delta}_k \hat{x}_{k|\tau_k} + \delta_k x_k, \quad (3.6)$$

$$\hat{x}_{k|\tau_k} = A \hat{x}_{k-1|k-1} + B u_{k-1}, \quad (3.7)$$

where $\bar{\delta}_k = 1 - \delta_k$ takes a value 1 when the packet is not transmitted. In such a case, the estimate is given by $\hat{x}_{k|\tau_k}$, a model based prediction from the last received data packet at time τ_k .

We use the information pattern \mathbb{I}_k^S to identify three multiple access architectures for NCSs:

- *Static MAC*: Static MAC protocols are random access methods with a fixed channel access probability. The access probability is independent of the current data or the past history of transmissions. The scheduling criterion f uses the information pattern $\mathbb{I}_k^{S,s}$, given by

$$\mathbb{I}_k^{S,s} = \{\alpha_k\}, \quad (3.8)$$

where α_k is a binary random variable, independent of the initial state x_0 and the process noise \mathbf{w}_0^{k-1} . This information pattern turns the scheduling policy in this method into a binary random number generator, such as a coin flip.

- *Dynamic MAC*: Dynamic MAC protocols are random access methods with a channel access probability that evolves over time. The access probability is still independent of the current data, but now depends on the past history of transmissions. The scheduling criterion f uses the information pattern $\mathbb{I}_k^{S,d}$, given by

$$\mathbb{I}_k^{S,d} = \{\boldsymbol{\gamma}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}\}, \quad (3.9)$$

where the bold font denotes a set of variables such as $\mathbf{a}_t^T = \{a_t, a_{t+1}, \dots, a_T\}$ and $\boldsymbol{\delta}$ denotes the MAC output to the channel access request γ from this plant (3.3). This information pattern induces memory into the scheduling policy.

- *Adaptive MAC*: Adaptive MAC protocols are random access methods with a channel access probability that depends on the current data packet, and possibly, evolve over time as well. The scheduling criterion f uses the information pattern $\mathbb{I}_k^{S,a}$, given by

$$\mathbb{I}_k^{S,a} = \{\mathbf{x}_0^k, \mathbf{y}_0^{k-1}, \boldsymbol{\gamma}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}, \mathbf{u}_0^{k-1}\}, \quad (3.10)$$

where x denotes the state of the plant (3.1), y denotes the measurement available across the network, and $\boldsymbol{\delta}$ denotes the MAC output to the channel access request γ from this plant (3.3). This information pattern results in a state-aware MAC.

In the rest of this chapter, we elaborate upon each of these architectures, and analyze the impact of these information patterns on the closed loop system design.

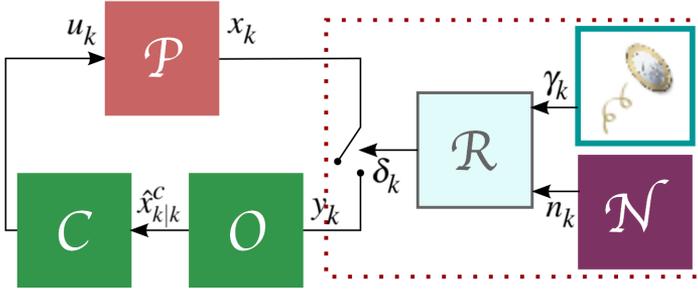


Figure 3.2: A multiple access network model for a closed loop system with a static MAC. Note that the outcome of a coin toss determines whether the packet accesses the channel or not, in this MAC.

3.2 Static MAC

Static MAC protocols are random access methods with a fixed channel access probability, when analyzed at the sampling time scale, independent of the current data or the past history of transmissions. The information pattern \mathbb{I}_k^s , given by (3.8), consists of only the outcome of a binary random process such as a coin-toss. This is illustrated in Fig. 3.2. An example of the static MAC is any protocol that terminates its operation, including all the retransmission attempts, within the sampling interval. This includes CSMA/CA if the worst-case delay is less than the sampling interval.

Here, γ and δ are binary random variables, independent of the data. A coin toss determines the channel access request γ_k . When $\gamma_k = 1$, the data packet is transmitted to the medium, and p_α is the probability of this event. Then, $\bar{p}_\alpha = 1 - p_\alpha$ is the probability of not accessing the shared medium, which happens when $\gamma_k = 0$. The traffic in the network at time k determines the MAC output δ_k . Let p_δ be the probability of a successful transmission, i.e., when $\delta_k = 1$. Again, $\bar{p}_\delta = 1 - p_\delta$ is the probability of a transmission failure, i.e., when $\delta_k = 0$. This can occur when the plant does not attempt to access the medium, or when there are multiple simultaneous transmissions resulting in a collision. Note that to find p_δ , we need to analyze the performance of the given contention resolution mechanism. This is possible for the MACs listed in Chapter 2. We now compute the estimation error covariance at the observer and the LQG cost for a closed loop system which uses a static MAC.

Lemma 3.1. *For the closed loop system given by the plant in (3.1), with a static MAC (3.2,3.8), a multiple access network on the sensor link (3.3), the controller in (3.4) and the LQG cost in (3.5), we state the following results:*

- i) *The estimate given by (3.6) minimizes the mean-squared error.*

ii) The estimation error covariance at the observer is given by

$$P_{k|k} = \bar{p}_\delta \left(\sum_{i=0}^k \Pr(\tau_k = i) \sum_{s=1}^{k-i} A^{s-1} R_w A^{s-1T} \right. \\ \left. + \Pr(\tau_k = -1) \sum_{s=1}^{k-i} A^{s-1} R_w A^{s-1T} + A^k R_0 A^{kT} \right), \quad (3.11)$$

where τ_k is the time index of the last received packet at time k .

iii) The certainty equivalence principle holds and the optimal control law is given by

$$u_k = -L_k \hat{x}_{k|k}, \quad (3.12)$$

where $L_k = (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A$.

iv) An expression for the LQG cost is given by

$$J_0 = \hat{x}_0^T S_0 \hat{x}_0 + \text{tr}\{S_0 P_0\} \\ + \sum_{n=0}^{N-1} \text{tr}\{S_{n+1} R_w + (L_n^T (Q_2 + B^T S_{n+1} B) L_n) P_{n|n}\}, \quad (3.13)$$

where L_k is given by (3.12). Also, $S_N = Q_0$ and S_k is obtained by the reverse iteration

$$S_k = Q_1 + A^T S_{k+1} A - A^T S_{k+1} B (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A, \quad (3.14)$$

Proof. Evaluating the expression $\mathbb{E}[x_k | \mathbb{I}_k^C]$, we get

$$\mathbb{E}[x_k | \mathbb{I}_k^C] = \begin{cases} \mathbb{E}[x_k | \delta_k = 1, y_k = x_k] & \delta_k = 1 \\ \mathbb{E}[x_k | \delta_{\tau_k} = 1, \boldsymbol{\delta}_{\tau_k+1}^k = 0, y_{\tau_k} = x_{\tau_k}, \mathbf{u}_0^{k-1}] & \delta_k = 0 \end{cases} \\ = \begin{cases} x_k & \delta_k = 1 \\ \mathbb{E}[A^{k-\tau_k} x_{\tau_k} + \sum_{\ell=1}^{k-\tau_k} (A^{\ell-1} B u_{k-\ell} + A^{\ell-1} w_{k-\ell}) \\ \quad | \delta_{\tau_k} = 1, \boldsymbol{\delta}_{\tau_k+1}^k = 0, y_{\tau_k} = x_{\tau_k}, \mathbf{u}_0^{k-1}] & \delta_k = 0 \end{cases} \\ = \begin{cases} x_k & \delta_k = 1 \\ A^{k-\tau_k} x_{\tau_k} + \sum_{n=1}^{k-\tau_k} A^{n-1} B u_{k-n} \\ \quad + \mathbb{E}[\sum_{\ell=1}^{k-\tau_k} A^{\ell-1} w_{k-\ell} | \boldsymbol{\delta}_{\tau_k+1}^k = 0] & \delta_k = 0 \end{cases} \\ = \begin{cases} x_k & \delta_k = 1 \\ A \hat{x}_{k-1|k-1} + B u_{k-1} & \delta_k = 0 \end{cases},$$

where the last equality sign is obtained using

$$A^{k-\tau_k} x_{\tau_k} + \sum_{n=1}^{k-\tau_k} A^{n-1} B u_{k-n} = A \hat{x}_{k-1|k-1} + B u_{k-1},$$

and $\mathbb{E}[\sum_{\ell=1}^{k-\tau_k} A^{\ell-1} w_{k-\ell} | \delta_{\tau_k+1}^k = 0] = 0.$

The last equation above holds when channel access is determined independent of the process noise, as in this case. Now, since the estimate in (3.6) is equal to $\mathbb{E}[x_k | \mathbb{I}_k^C]$, we know that this is a minimum mean-squared error (MMSE) estimate (Kailath et al., 2000).

The probability distribution of τ_k can be defined as

$$\Pr(\tau_k = i) = \begin{cases} (\bar{p}_\delta)^{k-i} \cdot p_\delta & 0 \leq i \leq k \\ (\bar{p}_\delta)^{k+1} & i = -1 \end{cases}.$$

Using this expression, and with the knowledge that the full state is sent with a successful transmission, the estimation error covariance can be computed as (3.11).

From the expression for the error covariance in (3.11), it is clear that there is no dependence on the applied control signals. Thus, there is no dual effect of order 2, and using the results of Theorem 2.1, we know that the certainty equivalence principle holds. Now, the optimal control law is given by (3.12) and the expression for the control cost remains the same as in the case with partial state information, and is given by (3.13). \square

Recall from Example 1.3 that a random access MAC (or the static MAC) is not well-suited for delay critical systems which require a performance guarantee, such as the systems we consider in this thesis. The results derived here are similar to those obtained for an i.i.d binary erasure channel with packet loss probability \bar{p} (Schenato et al., 2007).

3.3 Dynamic MAC

Dynamic MAC protocols are random access methods with a channel access probability that evolves over time. The information pattern $\mathbb{I}_k^{\delta,d}$, given by (3.9), consists of the transmission history of the node. We use the memory of the MAC to influence the outcome of channel access. The probabilities defined in this section are evaluated at the sampling time scale. The Dynamic MAC is illustrated in Fig. 3.3.

Here, γ and δ are random variables, independent of the data. Like before, the data packet may or may not be successfully transmitted depending on the traffic in the network. We use p_δ to denote the probability of a successful transmission, i.e., when $\delta_k = 1$. and $\bar{p}_\delta = 1 - p_\delta$ to denote the probability of a transmission failure, i.e., when $\delta_k = 0$. We now compute the estimation error covariance at the observer and the LQG cost for a closed loop system which uses a dynamic MAC.

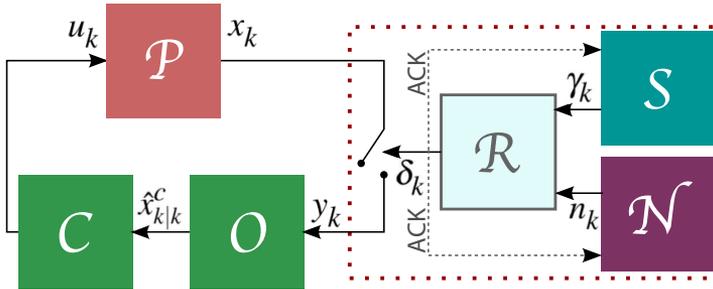


Figure 3.3: A multiple access network model for a closed loop system with a dynamic MAC. Note that an explicit acknowledgement (ACK) is required to make the transmission history available to the MAC at the next time step.

Lemma 3.2. *For the closed loop system given by the plant in (3.1), with a dynamic MAC (3.2,3.9), a multiple access network on the sensor link (3.3), the controller in (3.4) and the LQG cost in (3.5), we state the following results:*

- i) The estimate given by (3.6) minimizes the mean-squared error.*
- ii) The estimation error covariance at the observer is given by 3.11.*
- iii) The certainty equivalence principle holds and the optimal control law is given by (3.12).*
- iv) An expression for the LQG cost is given by (3.13).*

The proof is identical to the proof of Lemma 3.1.

The drawback of using such a MAC is that the performance of a closed loop system does not solely depend on its transmission history. As a simple example, an unstable plant may require channel access even if all its previous packets have been successfully transmitted, as opposed to a stable plant. The next class of MACs are better suited to emphasize the closed loop system performance.

3.4 Adaptive MAC

Adaptive MAC protocols are random access methods with a channel access probability that depends on the current data packet. The information pattern $\mathbb{I}_k^{S^a}$, given by (3.10), consists of the complete information of the plant history. This is represented in Fig. 3.4.

Note that the controller does not contain an observer block explicitly. This is because the transmission history now depends on the state, and it is not clear if the certainty equivalence principle holds. Thus, we need to understand the impact of using an adaptive MAC on the closed loop system, which we do in Chapter 4.

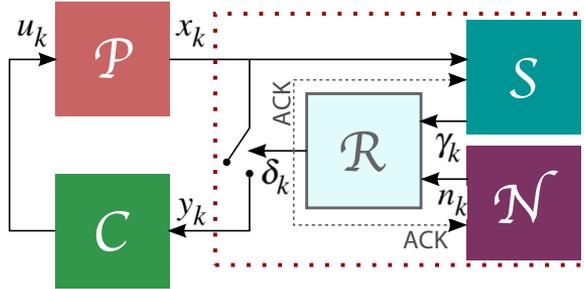


Figure 3.4: A multiple access network model for a closed loop system with an adaptive MAC. Note that an explicit acknowledgement (ACK) is required to make the transmission history available to the MAC at the next time step.

We now look at how to use this theoretical model to realize a MAC protocol. We outline a number of ways in which this can be done:

- *Tuning Existing Protocols:* Here, the state or the information set is used to tune the probability of channel access by changing the parameters of existing MAC protocols. This includes tuning parameters such as back-off window lengths, idle times (arbitration inter-frame space or DCF inter-frame space), number of permissible retransmissions, etc. Some of these have been attempted to implement QoS guarantees in 802.11e (Bianchi et al., 2005), and certain modifications have proven to be more effective than others. However, this may not be a very effective way of realizing an adaptive MAC, because such modifications only influence the access probability. The probability of a successful transmission is still determined by the net traffic, and not just the mechanisms of a single user. From the expression for the throughput of a simple MAC such as Aloha (2.1), it is clear that a linear relationship exists between the input and the output for a very small region of operation. Below and beyond this region, for any MAC protocol, the modifications are not likely to have the desired impact.
- *Regulating the Data Source:* Directly using the state of the plant to determine an access probability may result in a MAC that is difficult to implement and analyze. Instead, we use the state of the plant to select packets to send to the MAC, motivated by an understanding of the two roles played by a MAC. Any random access method works by resolving contention between simultaneous channel access requests, thus spreading traffic that arrives in bursts. The carrier sense multiple access with collision avoidance (CSMA/CA) method does this by assigning a random back-off to packets that attempt to access a busy channel, thus spreading the traffic over a longer interval of time. Similarly, the p-persistent CSMA method does this by probabilistically limiting access to the channel and permitting a number of retransmissions if

the channel is busy. However, all of these methods permit only a finite number of retransmissions, beyond which the packet is discarded. We appropriate this latter role of discarding packets to a local *state-based scheduler*, which sends fewer, but more important packets to the MAC for transmission across the network. The state-based scheduler regulates the flow of data from the node, with the aim of getting the best out of the network, given the other traffic in the network. We look at state-based schedulers and their design in Chapter 4.

- *Introducing New Protocols:* It might be possible to adapt existing protocols from the wired to the wireless world, such as CAN bus, which permit less random interactions between nodes, in a contention-based setting. This method holds some promise, especially for delay critical systems which require performance guarantees. We look at analyzing such a MAC in Chapter 5.

Thus, we take our premise of a state-aware MAC to an event-triggered formulation in Chapter 4, which regulates the flow of data from the sensor nodes, and to a CAN-bus realization for wireless networks in Chapter 5.

State-based Schedulers

In this chapter we explore the analysis and design of a state-aware contention-based MAC for the NCSs shown in Fig. 4.1. We consider state-aware MACs, as opposed to agnostic MACs, as we wish to influence the randomness of channel access in favour of the state of the plant in a closed loop system. Directly using the state of the plant to determine an access probability would give us the successful transmission probability that we desire in a small operating region of the MAC alone, where the input-output response is nearly linear. This region of operation is determined by the net traffic in the network, and cannot be influenced by the plant in consideration alone. Instead, we use a local *state-based scheduler*, which sends fewer, but more important packets to the MAC for transmission across the network. This strategy accomplishes a reduction in traffic, and provides a means of remaining in the non-saturated region of operation of a MAC, irrespective of the traffic in the network.

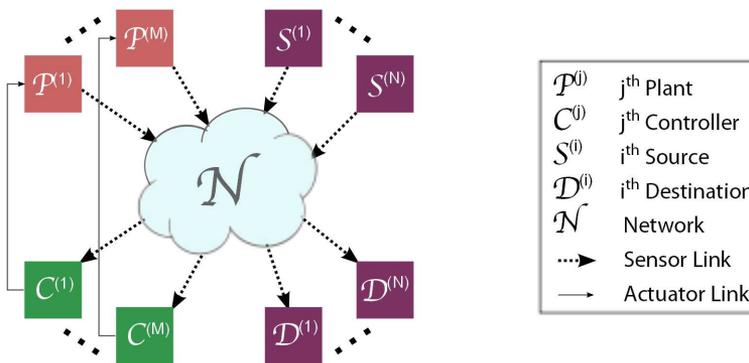


Figure 4.1: A network of M control loops, with each loop consisting of a plant $\mathcal{P}^{(j)}$ and a controller $\mathcal{C}^{(j)}$ for $j \in \{1, \dots, M\}$. The loops share access to a common medium on the sensor link, along with N other communication flows from generic source-destination pairs.

With the reduced traffic, the contention resolution mechanism may also be able to successfully handle all the transmission requests, within the maximum number of retransmissions permitted by the protocol. Thus, the state-based scheduler appropriates the MAC's role of discarding packets, and acts as a regulator of the data flow from the source. Now, the contention resolution mechanism retains responsibility for its primary role alone, that of resolving contention between simultaneous channel access requests, by spreading traffic that arrives in bursts.

Event-based systems were conceived with the same goal of reducing network traffic by regulating data flow from a plant. This makes the state-based scheduler an event-triggering mechanism, but one which operates under the premise of a shared, lossy network. Thus, our approach leads to a *network-aware* design of event triggering methods.

4.1 Contributions and Related Work

There are two main contributions in this chapter. The first contribution is an analysis of the impact of having a state-based scheduler in the closed loop. A state-based scheduler permits the information available to the controller to be altered with the plant state. Here, this information is not entirely random, like in the case of packet losses due to a noisy channel (Schenato et al., 2007; Gupta et al., 2007), and it can result in a sharply asymmetrical estimation error with and without a transmission, unlike in the case of encoder design over limited data rate channels (Tatikonda et al., 2004; Nair et al., 2007).

In this scenario, we ask if we can use the controller to move the plant state across the threshold and force a transmission? If this were possible, the controller would play two roles: the first being to control the plant, and the second being to control the information available at the next time step. It relates to the classical concept of a dual effect, as described in Feldbaum (1961). The answer to this question determines the ease of optimal controller design.

We examine our system and find that there is a dual effect with a state-based scheduler in the closed loop, and thus, the certainty equivalence principle does not hold. Hence, the optimal state-based scheduler, estimator and controller designs are coupled. A restriction on the input arguments to the state-based scheduler, such that these arguments are no longer a function of the past control actions, renders the setup free of a dual effect, and enables the certainty equivalence principle to hold. These results can be seen as an interpretation, within the state-based scheduler setup, of the classical work on dual effect, certainty equivalence and separation by Witsenhausen (1971) and Bar-Shalom and Tse (1974), and on adaptive control by Feldbaum (1961), Åström and Wittenmark (1995) and others (Filatov and Unbehauen, 2000).

The second contribution of this paper is the *dual predictor architecture*, which is our proposed solution to the design of the closed loop system, including the state-based scheduler. In this architecture, we use an innovation-based scheduler,

which permits a separation in design of the scheduler, estimator and controller. The state-based scheduler thresholds the squared difference of the innovation contained in the latest measurement, with respect to the estimator across the network. This results in an optimal certainty equivalent controller, and a simple observer which generates the minimum mean-squared error (MMSE) estimate.

The rest of the chapter is organized as follows. In Section 4.2, we present definitions and properties of the system. In Section 4.3, we derive the theoretical results for the case when full state information is available, with and without network traffic. We also present the dual predictor architecture for this case. Then, in Section 4.5, we extend our results to a system without full state information. We present an example, which illustrates our notion of network-aware event-triggering, in Section 4.6.

4.2 Preliminaries

We present the problem setup and a few important definitions in this section.

4.2.1 Problem Formulation

We consider a network of M control loops, as shown in Fig. 4.2. Each control loop consists of a plant $\mathcal{P}^{(j)}$, a state-based scheduler $\mathcal{S}^{(j)}$ and a controller $\mathcal{C}^{(j)}$ for $j \in \{1, \dots, M\}$. The loops share access to a common medium on the sensor link. From the perspective of a single control loop, this system can be modelled as shown in Fig. 4.3. We drop the index j in this figure for simplicity. The block \mathcal{N} represents the network as seen by this loop, and the block \mathcal{R} is the contention resolution

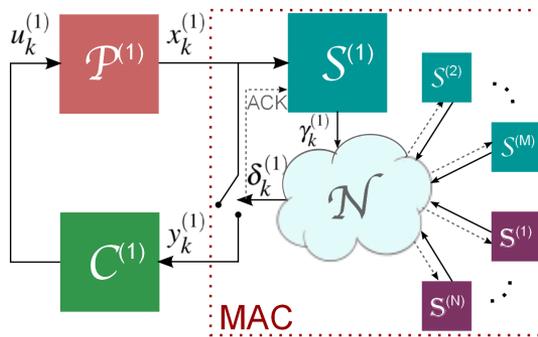


Figure 4.2: A plant ($\mathcal{P}^{(1)}$) with a state-based scheduler ($\mathcal{S}^{(1)}$) and a controller ($\mathcal{C}^{(1)}$) situated across a shared network (\mathcal{N}). Note that the schedulers ($\mathcal{S}^{(j)}$, $j \in \{2, \dots, M\}$) belong to other closed loop systems with their own plants and controllers each, which are not shown in this illustration. There are also other generic sources ($\mathcal{S}^{(i)}$, $i \in \{1, \dots, N\}$) in the shared network.

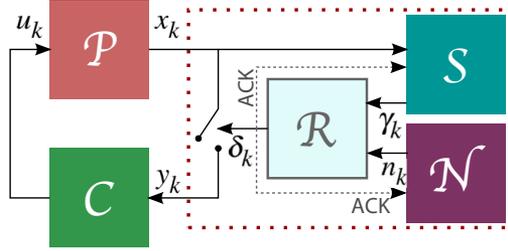


Figure 4.3: A multiple access network model from the perspective of a single closed loop system in the network

mechanism (CRM), which determines whether the control loop or the rest of the network gets to access the shared medium. Compared to the general problem setup presented in Chapter 1, the local state-based scheduler converts the channel access request γ_k into a binary random variable. Accordingly, the variable n_k is also considered to be a binary random variable, or a network traffic indicator. The problem formulation presented below assumes that the full state is available. The problem setup with partial state information is presented in Section 4.5. Each of the blocks in Fig. 4.3 are explained below.

Plant: The plant \mathcal{P} has state dynamics given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (4.1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and w_k is i.i.d. zero-mean Gaussian with covariance matrix R_w . The initial state x_0 is zero-mean Gaussian with covariance matrix R_0 .

State Based Scheduler: There is a local scheduler \mathcal{S} , situated in the sensor node, between the plant and the controller, which decides if the state is to be sent across the network or not. The scheduler output is denoted γ_k , where $\gamma_k \in \{0, 1\}$. It takes a value 1 when the state x_k is scheduled to be sent and 0 otherwise. The scheduling criterion is denoted by the policy f , which is defined on the information pattern of the scheduler $\mathbb{I}_k^{\mathcal{S}}$, and is given by

$$\gamma_k = f_k(\mathbb{I}_k^{\mathcal{S}}), \text{ where, } \mathbb{I}_k^{\mathcal{S}} = \{\mathbf{x}_0^k, \mathbf{y}_0^{k-1}, \gamma_0^{k-1}, \delta_0^{k-1}, \mathbf{u}_0^{k-1}\}. \quad (4.2)$$

Again, we use bold font to denote a set of variables such as $\mathbf{a}_t^T = \{a_t, a_{t+1}, \dots, a_T\}$.

Network: The network \mathcal{N} generates traffic, as is indicated by $n_k \in \{0, 1\}$. It takes a value 1 when the network traffic attempts to access the channel, and 0 otherwise. The network traffic is considered to be stochastic, as it could be generated by another such control loop, or by any other communicating node in the network. Thus, n_k is a binary random variable. It is not required to be i.i.d., and this is elaborated

upon below.

CRM: The CRM block \mathcal{R} resolves contention between multiple simultaneous channel access requests, i.e. when $\gamma_k = 1$ and $n_k = 1$. If the CRM resolves the contention in favour of our control loop, $\delta_k = 1$, and otherwise 0. The CRM can be modelled as the MAC channel response \mathcal{R} , with MAC output δ_k given by

$$\delta_k = \mathcal{R}(\gamma_k, n_k). \quad (4.3)$$

For brevity, we also define $\bar{\delta}_k = 1 - \delta_k$, which takes a value 1 when the packet is not transmitted. The MAC channel response \mathcal{R} is modelled as a discrete memoryless channel at the sampling time scale, requiring the CRM to resolve contention with respect to this plant's packet before the next sampling instant. This can be thought of as a limitation on the sampling rates supported by the model. Other network traffic may be sampled at different rates, and need not be independent across the sampling instants of this control system. Thus, n_k is not required to be i.i.d.

Measurement: The measurement across the network is denoted y_k . It is a non-linear function of the state x_k , and is given by

$$y_k = \delta_k x_k = \begin{cases} x_k & \delta_k = 1 \\ 0 & \delta_k = 0 \end{cases}. \quad (4.4)$$

A successful transmission results in the full state being sent to the controller. However, even non-transmissions convey information as the scheduler output δ_k can be treated as a noisy and coarsely quantized measurement of the state x_k .

Controller: The control law g denotes an admissible policy for the finite horizon N defined on the information pattern of the controller, \mathbb{I}_k^C , and is given by

$$u_k = g_k(\mathbb{I}_k^C), \text{ where, } \mathbb{I}_k^C = \{\mathbf{y}_0^k, \boldsymbol{\delta}_0^k, \mathbf{u}_0^{k-1}\}. \quad (4.5)$$

The objective function, defined over a horizon N is given by

$$J = \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{s=0}^{N-1} (x_s^T Q_1 x_s + u_s^T Q_2 u_s) \right], \quad (4.6)$$

where Q_0, Q_1 and Q_2 are positive definite weighting matrices.

In the rest of the chapter, we address the following questions -

1. What is the optimal control policy g^* for a given scheduling policy f , which minimizes the cost J in (4.6)?
 - i) Does the control signal in the above problem setup exhibit a dual effect?

- ii) Can we find an equivalent system, in the sense of Witsenhausen (see Definition 4.6), for this problem, with no dual effect?
 - iii) Will the optimal control design be different for a cost function which also penalizes transmissions using the shared medium?
2. What class of scheduling functions f can be chosen to simplify the design of the closed loop system?
 - i) What are the restrictions on the function f for the Certainty Equivalence Principle to hold in the above problem setup?
 - ii) How does the function f influence the design of the optimal observer?
 3. How can we design a suboptimal, but simple closed loop system architecture which ensures separation of the scheduler, controller and observer?

4.2.2 Definitions and Properties

We present a few definitions and properties that are used in the rest of the chapter.

Definition 4.1 (Unforced Process). *An auxiliary unforced process can be defined for any closed loop system, by removing the effect of the applied control signals from the state. The resulting unforced state is denoted \bar{x}_k , and given by*

$$\begin{aligned}\bar{x}_k &= x_k - \sum_{\ell=1}^k A^{\ell-1} B u_{k-\ell} \\ &= A^k x_0 + \sum_{\ell=1}^k A^{\ell-1} w_{k-\ell}.\end{aligned}\tag{4.7}$$

Information Patterns The information patterns defined in (4.2,4.5) in the section above, contain many system variables which are derived from one another. We can define an array of primitive random variables for the system, given by $\omega_0^k = [x_0, \mathbf{w}_0^{k-1}, \mathbf{n}_0^k]$, and express the information sets in terms of these variables, along with the scheduling and control policies. Thus, the information sets can be reduced to a sufficient statistic, using ω , f and g . These definitions also highlight the role of the design policies f and g on the system variables.

Definition 4.2 (Information Patterns). *The information patterns can be redefined in terms of the array of primitive random variables ω_0^k , and the control and scheduling policies, as given by*

$$\mathbb{I}_k^S = \{\mathbf{x}_0^k, \mathbf{y}_0^{k-1}, \gamma_0^{k-1}, \delta_0^{k-1}, \mathbf{u}_0^{k-1}\} = \{\omega_0^{k-1}, f_0^{k-1}, g_0^{k-1}\}, \tag{4.8}$$

$$\mathbb{I}_k^C = \{\mathbf{y}_0^k, \delta_0^k, \mathbf{u}_0^{k-1}\} = \{\mathbf{y}_0^k, \delta_0^k, g_0^{k-1}\}. \tag{4.9}$$

Note that an explicit acknowledgement (ACK) of a successful transmission is required for δ_k to be available to the scheduler. However, once it is known to the scheduler, knowledge of the applied policies f and g from time 0 to k are sufficient to reconstruct the variables n_k , y_k and u_k . The information pattern \mathbb{I}_k^C cannot be rewritten in terms of the scheduling policy f even if it is known to the controller, as the inputs to the scheduler are not available to permit a reconstruction of γ .

Last Received Packet Index The time index of the last received packet is denoted τ_k at time k , and is given by

$$\tau_k = \max\{t : \delta_t = 1, \text{ for } -1 \leq t \leq k\} \text{ and } \delta_{-1} = 1, \quad -1 \leq \tau_k \leq k. \quad (4.10)$$

The definition is illustrated in Fig. 4.4. An iterative relationship for τ_k can be found as

$$\tau_k = \bar{\delta}_k \tau_{k-1} + \delta_k k, \text{ and } \tau_{-1} = -1. \quad (4.11)$$

If a packet arrives at current time k , the last received packet index $\tau_k = k$. But, if there is no packet at time k , then the last received packet index is the same as the last received packet index from time $k - 1$, i.e. $\tau_k = \tau_{k-1}$. This implies that $\tau_k \in \{-1, 0, \dots, k\}$.

Dual Effect: The Dual Effect property is redefined here with the notation discussed above for the sake of clarity. This property is proven for the above problem setup in Sec. 4.3.1.

Definition 4.3 (No Dual Effect (Bar-Shalom and Tse, 1974)). *A control signal is said to have no dual effect of order $r \geq 2$, if*

$$\mathbb{E}[M_k^r | \mathbb{I}_k^C] = \mathbb{E}[M_k^r | x_0, \mathbf{w}_0^{\tau_k}, \mathbf{n}_0^k], \quad (4.12)$$

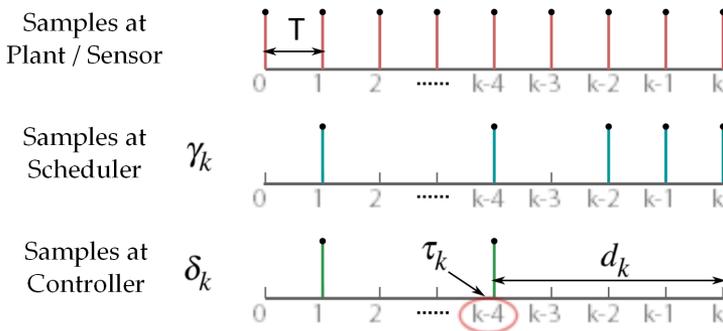


Figure 4.4: An illustration of the delay since the last received packet d_k and the index of the last received packet τ_k . The sampling period of the sensor or plant is a constant T . The scheduler picks a few samples to transmit, as given by γ_k and only some of those are received by the controller, as given by δ_k .

where $M_k^r = \mathbb{E}[\|x_k - \mathbb{E}[x_k|\mathbb{I}_k^C]\|^r | \mathbb{I}_k^C]$ is the r^{th} central moment of x_k conditioned on \mathbb{I}_k^C and τ_k is the time index of the last received measurement at time k .

Note that M_k^r in (4.12) must specifically not be a function of the past control policies g^{k-1} for the control signal to have no dual effect of order r . In other words, if there is no dual effect, the expected future uncertainty is not affected by the controls \mathbf{u}_0^{k-1} . In the presence of a dual effect, the optimal control laws are hard to find (Åström and Wittenmark, 1995).

Certainty Equivalence: The Certainty Equivalence Property is redefined here with the notation discussed above for the sake of clarity.

Definition 4.4 (Certainty Equivalence Principle (Bar-Shalom and Tse, 1974)). *The certainty equivalence principle holds if the closed-loop optimal controller has the same form as the deterministic optimal controller with the state x_k replaced by the estimate $\hat{x}_{k|k} = \mathbb{E}[x_k|\mathbb{I}_k^C]$.*

The conditions under which this property holds for state-based schedulers are presented in Sec. 4.3.3.

Correlated Network Noise: We state a property of feedback systems with a state-based scheduler which share a contention-based multiple access network. Even if the initial states and disturbances of all the plants in the network are independent, the contention-based MAC introduces a correlation between the traffic sources in the network, as noted in Cervin and Henningsson (2008) and Rabi and Johansson (2009).

Property 4.5. *For a network of M plants with state dynamics given by (4.1), which use a state-based scheduler of the form (4.2) and a shared multiple access channel, with channel output given by (4.3), to reach the respective controllers, given by (4.5), the network noise n_k is correlated to the state of the plant x_k .*

Proof. At time $k-1$, the MAC output δ_{k-1} is correlated to the state-based scheduler outcome γ_{k-1} , and the network noise n_{k-1} , from (4.3). The control signal u_{k-1} is a function of the MAC output δ_{k-1} (4.5), and is applied through the feedback to the plant. Thus, x_k is correlated to δ_{k-1} .

Similarly, other network traffic, which consists of feedback loops, will have states at time k that are correlated to δ_{k-1} as well. The state-based scheduler outcomes of the other plants decide the net traffic at time k , and thus, the indicator of network traffic n_k will also be correlated to δ_{k-1} . Thus, the network noise n_k is correlated to the plant state x_k . \square

4.3 Optimal Controller Design

We present the main results of this chapter in this section. We analyze the effects of a state-based scheduler on a control loop in the absence of other network traffic,

i.e. $n_k = 0$. We show that there is a dual effect of the control signal, and that the scheduling policy must be restricted from using the past control inputs for the certainty equivalence principle to hold. Then, we extend these results to the case with network traffic. To begin with, we have $n_k = 0$, for all k , and consequently, the MAC output is given by $\delta_k = \gamma_k$, for all k for a discrete, memoryless channel.

4.3.1 Dual Effect with State-based Scheduling

For the problem defined in section 4.2.1, we observe the following result.

Theorem 4.1. *For the closed loop system with the plant (4.1), state-based scheduler (4.2), no network traffic ($n_k = 0, \forall k$) and controller (4.5), the control signal has a dual effect of order $r = 2$.*

Proof. We examine the estimation error covariance $P_{k|k}$, and show that it is a function of the applied control signals \mathbf{u}_0^{k-1} . For the estimate defined by $\hat{x}_{k|k} \triangleq \mathbb{E}[x_k | \mathbb{I}_k^C]$, the estimation error can be written as

$$\begin{aligned} \tilde{x}_{k|k} &= x_k - \mathbb{E}[x_k | \mathbb{I}_k^C] \\ &= A^k x_0 + \sum_{\ell=1}^k A^{\ell-1} (B u_{k-\ell} + w_{k-\ell}) - \mathbb{E}[A^k x_0 + \sum_{\ell=1}^k A^{\ell-1} (B u_{k-\ell} + w_{k-\ell}) | \mathbb{I}_k^C] \\ &= A^k x_0 + \sum_{\ell=1}^k A^{\ell-1} w_{k-\ell} - \mathbb{E}[A^k x_0 + \sum_{\ell=1}^k A^{\ell-1} w_{k-\ell} | \mathbb{I}_k^C] \\ &= \bar{x}_k - \mathbb{E}[\bar{x}_k | \mathbb{I}_k^C], \end{aligned}$$

where \bar{x}_k is the state of the unforced process (see Definition 4.1). From (4.4), we know that a successful transmission results in the full state being sent to the controller. Thus, we have

$$\mathbb{E}[\bar{x}_k | \mathbb{I}_k^C] = \begin{cases} \bar{x}_k & \delta_k = 1 \\ \mathbb{E}[\bar{x}_k | \mathbb{I}_k^C, \delta_k = 0] & \delta_k = 0 \end{cases}. \quad (4.13)$$

The scheduler outcome, and consequently δ_k , are influenced by the applied control inputs \mathbf{u}_0^{k-1} in a state-based scheduler such as (4.2).

Now, the estimation error is clearly dependent on the applied controls, as seen in

$$\tilde{x}_{k|k} = \begin{cases} 0 & \delta_k = 1 \\ x_k - \mathbb{E}[x_k | \mathbb{I}_k^C, \delta_k = 0] & \delta_k = 0 \end{cases}.$$

Then, the error covariance, $P_{k|k} \triangleq \mathbb{E}[\tilde{x}_{k|k} \tilde{x}_{k|k}^T | \mathbb{I}_k^C]$, is given by

$$P_{k|k} = \begin{cases} 0 & \delta_k = 1 \\ \mathbb{E}[\tilde{x}_{k|k} \tilde{x}_{k|k}^T | \mathbb{I}_k^C, \delta_k = 0] & \delta_k = 0 \end{cases}. \quad (4.14)$$

The covariance $P_{k|k}$ is zero if the scheduling criterion in (4.2) is fulfilled, and non-zero otherwise. Clearly, $P_{k|k}$ is a function of the past controls. Hence, $P_{k|k}$ does not satisfy the condition (4.12) required to have no dual effect. Thus, we see that the system (4.1)–(4.5) exhibits a dual effect of order $r = 2$. \square

In this setup, there is an incentive for the control policy to modify the estimation error along with controlling the plant.

4.3.2 Equivalent Schedulers vs. Equivalent Systems

Every state-based scheduler f in (4.2) can be transformed into an equivalent scheduler \tilde{f} , such as

$$\gamma_k = \tilde{f}_k(x_0, \mathbf{w}_0^{k-1}). \quad (4.15)$$

The applied controls \mathbf{u}_0^{k-1} are known at time k , and hence, such a transformation can always be accomplished. We now examine the question of whether the closed loop system with this equivalent scheduler, is equivalent to the original system. Witsenhausen (1971) defines an equivalent design, which gives us the following definition when applied to our problem.

Definition 4.6. *An equivalent control design g_{eq} for the optimal controller g^* , which minimizes the cost criterion (4.6) for the system defined by (4.1)–(4.5), satisfies the equivalence relationship given by*

$$\mathbf{u}^* = \Upsilon(\boldsymbol{\omega}, g^*) = \Upsilon(\boldsymbol{\omega}, g_{eq}), \quad (4.16)$$

where Υ is obtained by recursive substitution for the control signals in the system equations with the respective control policy and the primitive random variables $\boldsymbol{\omega}$.

Let $\{\mathcal{P}, \bar{f}, \bar{g}\}$ denote the system with the plant given by (4.1), with \bar{f} as the given scheduler and \bar{g} as the optimal controller for the cost criterion (4.6) applied to this system. We now note the following result.

Lemma 4.2. *For two schedulers f , given by (4.2), and \tilde{f} , given by (4.15), which result in the same schedules for the closed loop system with a plant (4.1), no network traffic ($n_k = 0, \forall k$) and controller (4.5), $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$ is not an equivalent system to $\{\mathcal{P}, f, g^*\}$, in the sense of Witsenhausen.*

Proof. Definition 4.6 requires the control signals obtained using the policies g^* and \tilde{g} to be equal. In this proof, we find the optimal control policies for $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$ and $\{\mathcal{P}, f, g^*\}$, and show that they do not result in the same control signals.

For the optimal control policy, which minimizes the quadratic cost J (4.6), to be certainty equivalent, we need to find a solution to the Bellman equation (Åström, 1970), which is a one-step minimization of the form

$$V_k = \min_{u_k} \mathbb{E}[x_k^T Q_1 x_k + u_k^T Q_2 u_k + V_{k+1} | \mathbb{I}_k^C]. \quad (4.17)$$

In general, without defining a structure for the estimator, [Bar-Shalom and Tse \(1974\)](#) give us the solution to the functional, of the form

$$V_k = \mathbb{E} \left[x_k^T S_k x_k | \mathbb{I}_k^C \right] + s_k , \quad (4.18)$$

where S_k is a positive semi-definite matrix and both S_k and s_k are not functions of the applied control signals \mathbf{u}_0^{k-1} . We now prove that a solution of this form can be found for $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$, but not for $\{\mathcal{P}, f, g^*\}$.

First consider the system $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$. At time N , the functional has a trivial solution with $S_N = Q_0$ and $s_N = 0$. This solution can be propagated backwards, in the absence of a dual effect. To show this, we use the principle of induction, and assume that a solution of the form (4.18) holds at time $k + 1$. Then, at time k , we have

$$\begin{aligned} V_k &= \min_{u_k} \mathbb{E} [x_k^T Q_1 x_k + u_k^T Q_2 u_k + x_{k+1}^T S_{k+1} x_{k+1} + s_{k+1} | \mathbb{I}_k^C] \\ &= \min_{u_k} \mathbb{E} [x_k^T (Q_1 + A^T S_{k+1} A) x_k | \mathbb{I}_k^C] + \text{tr}\{S_{k+1} R_w\} + \mathbb{E}[s_{k+1} | \mathbb{I}_k^C] \\ &\quad + u_k^T (Q_2 + B^T S_{k+1} B) u_k + \hat{x}_{k|k}^T A^T S_{k+1} B u_k + u_k^T B^T S_{k+1} A \hat{x}_{k|k} . \end{aligned}$$

The optimal control is found to be

$$\begin{aligned} \tilde{u}_k &= -L_k \hat{x}_{k|k} , \\ \text{where, } L_k &= (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A . \end{aligned} \quad (4.19)$$

Substituting the expression for \tilde{u}_k into V_k gives us a solution of the form in (4.18), with

$$\begin{aligned} S_k &= Q_1 + A^T S_{k+1} A - A^T S_{k+1} B (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A , \\ s_k &= \text{tr}\{S_{k+1} R_w\} + \mathbb{E}[s_{k+1} | \mathbb{I}_k^C] \\ &\quad + \text{tr}\{A^T S_{k+1} B (Q_2 + B^T S_{k+1} B)^{-1} B^T S_{k+1} A P_{k|k}\} , \end{aligned} \quad (4.20)$$

where the matrix S_k is positive semi-definite and not a function of the applied controls \mathbf{u}_0^{k-1} . The scalar s_k is not a function of the applied controls \mathbf{u}_0^{k-1} if and only if $P_{k|k}$ has no dual effect ([Bar-Shalom and Tse, 1974](#)). From the expression for the error covariance $P_{k|k}$ (4.14), it is clear that a scheduling criterion that is not a function of the past control actions, such as (4.15), results in no dual effect. Under this condition, s_k is not a function of the applied controls \mathbf{u}_0^{k-1} and the proof by induction is complete. Since the optimal control signal (4.19) is a function of only the estimate $\hat{x}_{k|k}$, the Certainty Equivalence Principle holds.

Now, consider the system $\{\mathcal{P}, f, g^*\}$. Solving the backward recursion as we did above, we find that V_N and V_{N-1} have a solution of the form (4.18), with $S_N = Q_0$ and $s_N = 0$, and S_{N-1} and s_{N-1} given by (4.20) with $k = N - 1$. However, V_{N-2}

results in a different minimization problem for this system because of the dual effect in $\{\mathcal{P}, f, g^*\}$. The optimal control signal u_{N-2}^* can be obtained by solving

$$\begin{aligned} \frac{\partial V_{N-2}}{\partial u_{N-2}^*} &= 2u_{N-2}^{*T}(Q_2 + B^T S_{N-1} B) + 2\hat{x}_{N-2|N-2}^T A^T S_{N-1} B \\ &\quad + \frac{\partial}{\partial u_{N-2}^*} \left(\text{tr} \{ A^T S_N B (Q_2 + B^T S_N B)^{-1} B^T S_N A \cdot \mathbb{E} [P_{N-1|N-1}^C] \} \right). \end{aligned}$$

Multiplying the above expression with $(Q_2 + B^T S_{N-1} B)^{-1}$ from the right and using (4.19) to substitute for the second term, we obtain the simpler equation

$$\frac{\partial V_{N-2}}{\partial u_{N-2}^*} = 2(u_{N-2}^{*T} - \tilde{u}_{N-2}^T) + \frac{\partial}{\partial u_{N-2}^*} \left(\text{tr} \{ \mathbb{K}_{N-2} \mathbb{E} [P_{N-1|N-1}^C] \} \right) = 0, \quad (4.21)$$

where, we set $\mathbb{K}_{N-2} = (Q_2 + B^T S_{N-1} B)^{-1} A^T S_N B (Q_2 + B^T S_N B)^{-1} B^T S_N A$. There is an additional term in (4.21), related to the estimation error covariance $P_{N-1|N-1}$, which is not equal to zero as implied by the dual effect property from Theorem 4.1. Due to this term, the above minimization problem is not linear, and the solutions \tilde{u}_{N-2} and u_{N-2}^* are not equal. From this point on, the cost-to-go for the optimal control policy g^* does not have a solution of the form given by (4.18). Hence, the control signals $\{\tilde{\mathbf{u}}_0^{N-3}$ and $\{\mathbf{u}^*\}_0^{N-3}$ will not be equal. Now, the joint distribution of all system variables could be quite different for schedulers \tilde{f} and f . Thus, the described transformation of the scheduling criterion does not result in an equivalent class construction. \square

Due to the dual effect, the optimal control action takes on two roles. One, to control the plant, and the other, to probe the plant state which could result in an improved estimate (Åström and Wittenmark, 1995). In the certainty equivalent setup, the probing action cannot be implemented due to the lack of a dual effect and the resulting control actions will not remain the same.

4.3.3 Conditions for Certainty Equivalence

From the previous discussions, it is clear that a scheduling criterion independent of the past control actions results in no dual effect. This result is presented below.

Corollary 4.3. *The optimal controller for the system with the plant (4.1), a given state-based scheduler, no network traffic ($n_k = 0, \forall k$) and controller (4.5), with respect to the cost in (4.6), is certainty equivalent if and only if the scheduling decisions are not a function of the applied control actions, such as in (4.15).*

Proof. In the proof of Lemma 4.2, it is clear from (4.19) that the optimal control policy \tilde{g} for the system $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$ is certainty equivalent. \square

Corollary 4.3 gives a condition on the scheduler to guarantee certainty equivalence. Note that the resulting closed loop system is not optimal in general. A scheduler with a dual effect may result in a better design with lower cost.

4.3.4 An Improvement on Certainty Equivalent Control

In a problem setup where the transmission status is indicated by a binary random variable such as δ_k , and the information patterns at the scheduler and controller are given by (4.8) and (4.9), respectively, the certainty equivalence principle only holds when the transmission loss cannot be influenced by the control policy. Then, the effect of the estimation error on the control cost can be quantified with an additive term (Åström, 1970). However, when there is a dual effect, the estimation error can be modified by the control policy. Thus, the dual effect property can lead to an improvement on certainty equivalent control. This result, which summarizes the findings of this section, is presented below.

Theorem 4.4. *An optimal controller g^* for the system with the plant (4.1), state-based scheduler (4.2), no network traffic ($n_k = 0, \forall k$) and controller (4.5), with respect to the cost criterion (4.6), results in a control cost $J_{SS} < J_{CE}$, where J_{CE} is the cost using a certainty equivalent controller.*

Proof. Let us assume that the certainty equivalent control policy \tilde{g} , defined in (4.19), is optimal for the given system with respect to the cost criterion (4.6). Then, it should be the solution to (4.21). Substituting for u_{N-2}^* in (4.21) with \tilde{u}_{N-2} from (4.19), we get

$$\frac{\partial}{\partial u_{N-2}^*} \left(\text{tr} \{ \mathbb{K}_{N-2} \mathbb{E} [P_{N-1|N-1} | \mathbb{I}_{N-2}^C] \} \right) = 0.$$

This is only possible if $\mathbb{E} [P_{N-1|N-1} | \mathbb{I}_{N-2}^C]$ is not a function of u_{N-2}^* , i.e., if the control signal is free of a dual effect of order 2. This is not true for the given system, from the results of Theorem 4.1 and Lemma 4.2. Thus, there is a contradiction, and the certainty equivalent policy \tilde{g} is not the optimal controller for the system given here. \square

The gradient $\frac{\partial}{\partial u_{N-2}^*} (\text{tr} \{ \mathbb{K}_{N-2} \mathbb{E} [P_{N-1|N-1} | \mathbb{I}_{N-2}^C] \})$ in (4.21) provides an improving direction for the net control cost in the presence of a dual effect. The optimal controller g^* utilizes this incentive, as seen in the proof of Lemma 4.2.

4.3.5 Penalizing Network Usage

We have proven, in Theorem 4.1 and Lemma 4.2, that the applied controls play a significant role in a state-based scheduler and cannot be removed from the scheduler inputs to create an equivalent setup without a dual effect. However, the minimizing solution to a cost criterion can render the effect of the applied controls redundant. To see an example of this, consider the problem of finding the jointly optimal scheduler-controller pair for the classical LQG cost criterion in (4.6). Since there is no penalty on using the network, the optimal scheduler policy is to transmit all the time. Now, the structure of the closed loop system does not resemble the

one presented in Theorem 4.1, and consequently, that result does not hold. In this scenario, there is no incentive for the controller to influence the transmissions.

Now, consider a cost criterion which penalizes the use of the network, such as

$$J_\Lambda = \min_{\mathbf{u}_0^{N-1}, \delta_0^{N-1}} \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{s=0}^{N-1} x_s^T Q_1 x_s + u_s^T Q_2 u_s + \Lambda \delta_s \right], \quad (4.22)$$

where Q_0, Q_1 and Q_2 are positive definite weighting matrices and $\Lambda > 0$ is the cost of using the network. The optimal state-based scheduling policy chooses a schedule in relation to the penalty Λ , such that the average network use, i.e. $\mathbb{E}[\delta_k]$, decreases as Λ increases. Thus, we state the following result.

Corollary 4.5. *For the closed loop system, with the plant (4.1), state-based scheduler (4.2), no network traffic ($n_k = 0, \forall k$) and controller (4.5), the control signals derived from the jointly optimal scheduler-controller pair, which minimize the cost criterion in (4.22), exhibit a dual effect of order $r = 2$.*

Proof. Let us assume that the scheduler from the jointly optimal scheduler-controller pair does not depend on the applied controls, such that the estimation error covariance $P_{k|k}$ is not a function of \mathbf{u}_0^{k-1} . Using the result of Corollary 4.3, we know that the optimal controller for this scheduler is certainty equivalent, and then, the net cost is given by

$$J_\Lambda = J_{CE} + \sum_{s=0}^{N-1} \Lambda \tilde{\delta}_s, \quad (4.23)$$

where $\tilde{\delta}_s$ is the optimal scheduled sequence.

For the above scheduler, we can always find an equivalent scheduler which uses the applied controls, such as in (4.2), and results in the same schedule $\tilde{\delta}_s$ for all s . However, from Theorem 4.1 and Lemma 4.2, we know that the estimation error $P_{k|k}$ will now be a function of the applied controls. Then, if we fix this state-based scheduler, we know, from Theorem 4.4, that there exists an optimal controller which results in a cost $J_{SS} < J_{CE}$, for a given state-based scheduler. The net cost is now given by

$$J_\Lambda = J_{SS} + \sum_{s=0}^{N-1} \Lambda \tilde{\delta}_s,$$

which is less than the cost in (4.23). Thus, we have shown that there exists a scheduler-controller pair that results in a lower cost. This is a contradiction, and hence, the scheduler from the jointly optimal scheduler-controller pair must depend on the applied controls, resulting in a dual effect of the control signal. \square

This provides the controller an incentive to modify the transmission outcome. As a result, the optimal scheduler and controller designs in this problem are coupled.

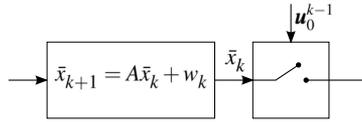


Figure 4.5: Due to the asymmetry in the amount of information available at the controller with and without a transmission, the applied controls influence the estimate of the unforced process.

4.3.6 A Discussion on the Dual Effect in this Setup

The results in this section show that the applied controls can push the state across the scheduler threshold, and influence the transmission outcome. *The applied controls influence the estimate of the unforced process*, as shown in Fig 4.5. This is a consequence of sending the full state in case of a transmission success as against sending a coarse quantization of the state (δ_k) in case of no transmission. The unequal information in the measurement y_k , with and without a transmission, provides a probing incentive for the controller.

These results imply that the dual effect is visible in any control signal applied to the plant, not just the optimal one, as the control signal will always influence the estimation error, irrespective of whether it has been designed to do so or not. In this context, the dual effect can be best explained as a coupling between the control and scheduling policies. The information at the scheduler contains all the information available to the controller. Despite this, the control policy can signal to the estimation error as the scheduler outcome is a function of the control design.

The dual effect and certainty equivalence properties have been noted previously in other problems for NCSs. We discuss these other occurrences and the connections to our problem setup below.

Packet Drops over a Lossy Network Packet drops in a lossy network are not influenced by the applied controls. Hence, certainty equivalence holds, when there are packet drops on the sensor or controller links or both, as shown in [Schenato et al. \(2007\)](#) and [Gupta et al. \(2007\)](#). However, this only applies if there is an ACK of packets received or lost.

Importance of Side Information In any NCS problem, the classical information pattern must be reconstructed for the certainty equivalence principle to hold ([Witsenhausen, 1968](#)). This may require one or more explicit side information channels to convey acknowledgements of received packets back to the transmitters. Certainty equivalence holds only when full side information is available ([Bao et al., 2010](#); [Schenato et al., 2007](#)).

Encoder Design over Limited Data Rate Channels The problem of encoder design on the sensor link has been shown to have no dual effect in [Tatikonda et al. \(2004\)](#). In this problem, the encoder output is the only measurement available

across the channel, and thus, the applied controls cannot influence the estimate of the unforced process, unlike in (4.13). Due to this, it is possible to show that an information pattern with the encoder inputs derived from the unforced state history alone form a subset of the information pattern \mathbb{I}_k^S , and that the estimate derived from such an encoder's output across the channel is equal to $\mathbb{E}[x_k | \mathbb{I}_k^C]$.

Event-based Systems Molin and Hirche (2009, 2010) deal with a similar problem formulation as in Section 4.2.1, and with a cost function such as (4.22). They use a transformation similar to the one presented for the encoder design problem in Nair et al. (2007). There are, however, subtleties in defining an equivalence class for a state-based scheduler. Using an equivalent scheduler need not result in an equivalent system, as shown in Lemma 4.2.

4.3.7 Effect of State-based Schedulers in a network

In this subsection, we re-analyze the effects of a state-based scheduler on the control loop in the *presence* of other network traffic. Thus, we have $n_k \neq 0$ and a channel output given by (4.3). Recall from Property 4.5, that the network traffic indicator n_k is correlated to the state of the plant x_k . The certainty equivalence property need not hold for plants where the measurement noise is correlated to the process noise (Bar-Shalom and Tse, 1974). To focus on the effect of state-based schedulers on the closed loop system, the results presented in the previous subsection did not include network traffic. Now, we re-derive some of the above results for the system in the presence of network traffic.

Theorem 4.6. *For the closed loop system defined by (4.1)–(4.5), the control signal has a dual effect of order $r = 2$.*

Proof. The MAC output δ_k (4.3) is clearly still a function of the applied controls, through the state-based scheduler outcome. Thus, the estimation error covariance $P_{k|k}$, in (4.14), remains a function of the applied controls \mathbf{u}_0^{k-1} . Since $P_{k|k}$ does not satisfy the condition (4.12) required to have no dual effect, we see that the system (4.1)–(4.5) exhibits a dual effect of order $r = 2$. \square

With the above result, Lemma 4.2 can be easily extended to include the case with network traffic. However, it is not as straightforward to extend Corollary 4.3. When the measurement noise is correlated to the process noise, certainty equivalence need not hold. To see why, recall the proof of Lemma 4.2, where we derive a solution of the form $V_k = \mathbb{E}[x_k^T S_k x_k | \mathbb{I}_k^C] + s_k$ for the Bellman equation (4.17). Now, if w_k is correlated to the variables in the information set \mathbb{I}_k^C , specifically \mathbf{n}_0^k , the minimization with respect to u_k in (4.19) must include the term $\text{tr}\{S_{k+1} R_w\}$. Then, the optimal controller will not have the form shown in (4.19), and thus, certainty equivalence will not hold.

We need to prove that w_k is independent of \mathbf{n}_0^k for the certainty equivalence property to hold, which we do below.

Corollary 4.7. *The optimal controller for the closed loop system defined by (4.1)–(4.5), with respect to the cost criterion (4.6), is certainty equivalent if the network traffic indicator n_k is independent of the process noise w_k , and, if the scheduling decisions are not a function of the applied control actions, i.e. if*

$$\gamma_k = \check{f}_k(x_0, \mathbf{w}_0^{k-1}, \mathbf{n}_0^{k-1}). \quad (4.24)$$

Proof. Note that n_k is only correlated to δ_0^k and thus, to the signals \mathbf{w}_0^{k-1} , from Property 4.5. As the process noise is i.i.d, n_k is independent with respect to w_k . A scheduler of the form (4.24) provides no incentive to the controller, and thus, Certainty Equivalence holds. \square

Now, it is directly possible to extend the results of Theorem 4.4 and Corollary 4.5 to include the effect of network traffic.

4.4 Closed Loop System Architecture

In this section, we identify a property of the scheduling policy that results in a simplification of the design of the closed loop system. This enables us to propose a dual predictor architecture for the closed loop system, which results in a separation of the scheduler, observer and controller designs.

4.4.1 Observer Design

In this section, we propose a structure for the estimator at the controller. Due to the non-linearity of the problem, the estimate in general can be hard to compute.

The estimation error is reset to zero with every transmission, as we send the full state. Consider one such reset instance, a time k such that $\delta_k = 1$. The state is sent across the network, $y_k = x_k$, so the estimate $\hat{x}_{k|k} = x_k$. A suitable control signal u_k is found and applied to the plant, which results in the next state x_{k+1} . Now, the scheduler can generate one of two outcomes. We consider each case, and find an expression for the estimate, below:

- a) $\delta_{k+1} = 0$: We need an estimate of w_k . We use the scheduler output as a coarse quantized measurement to generate this, as follows:

$$\begin{aligned} \hat{x}_{k+1|k+1} &= \mathbb{E}[x_{k+1} | \mathbb{I}_{k+1}^C, \delta_{k+1} = 0] \\ &= \mathbb{E}[Ax_k + Bu_k + w_k | \mathbf{y}_0^{k+1}, \delta_0^k, \delta_{k+1} = 0, \mathbf{u}_0^k] \\ &= Ax_k + Bu_k + \mathbb{E}[w_k | \mathbf{y}_0^{k+1}, \delta_0^k, \delta_{k+1} = 0, \mathbf{u}_0^k] \\ &= Ax_k + Bu_k + \mathbb{E}[w_k | \hat{f}(w_k) = 0], \\ \tilde{x}_{k+1|k+1} &\triangleq x_{k+1} - \hat{x}_{k+1|k+1} = w_k - \mathbb{E}[w_k | \hat{f}(w_k) = 0], \end{aligned} \quad (4.25)$$

where, $\hat{f}(w_k) \equiv f(Ax_k + Bu_k + w_k | x_k, u_k)$.

- b) $\delta_{k+1} = 1$: The estimation error is zero as $\hat{x}_{k+1|k+1} = x_{k+1}$.

The transformation to \dot{f} in (4.25), is not intended to remove the dual effect, but merely serves to remove the known variables from the expression. The dual effect has influenced the packet's transmission, i.e., the value of δ_{k+1} .

To see this more clearly, we look at the next time instant. Now a signal u_{k+1} is generated, and applied to the plant. We note that $x_{k+2} = A^2x_k + ABu_k + Bu_{k+1} + Aw_k + w_{k+1}$. The state x_{k+2} is either sent to the controller or not depending on the scheduler outcome δ_{k+2} . Again, we look at both cases, and derive an expression for the estimate:

- i) $\delta_{k+2} = 0$: We now need to estimate $Aw_k + w_{k+1}$, as the rest is completely known from x_{k+2} . We use both scheduler outputs δ_{k+1} and δ_{k+2} to generate an estimate of the unknown variables as

$$\begin{aligned}\hat{x}_{k+2|k+2} &= A^2x_k + ABu_k + Bu_{k+1} \\ &\quad + \mathbb{E}[Aw_k + w_{k+1} | \dot{f}(w_k) = 0, \dot{f}(Aw_k + w_{k+1}) = 0], \\ \tilde{x}_{k+2|k+2} &= Aw_k + w_{k+1} \\ &\quad - \mathbb{E}[Aw_k + w_{k+1} | \dot{f}(w_k) = 0, \dot{f}(Aw_k + w_{k+1}) = 0].\end{aligned}$$

- ii) $\delta_{k+2} = 1$: The estimation error is zero as $\hat{x}_{k+2|k+2} = x_{k+2}$.

This process can be continued recursively through a non-transmission burst, until finally a measurement is received and the estimation error is reset to zero. Thus, the estimate at any time k is given by

$$\hat{x}_{k|k} = \begin{cases} x_k & \delta_k = 1 \\ A^{k-\tau_k} x_{\tau_k} + \sum_{s=1}^{k-\tau_k} A^{s-1} Bu_{k-s} & \delta_k = 0 \\ + \mathbb{E}\left[\sum_{s=1}^{k-\tau_k} A^{s-1} w_{k-s} \mid \dot{f}_k, \dots, \dot{f}_{\tau_k+1} = 0\right] & \end{cases}, \quad (4.26)$$

where τ_k is the time index of the last received measurement at time k , as defined in (4.10), and the argument to the function \dot{f}_t is given by the term $\sum_{s=1}^{t-\tau_t} A^{s-1} w_{t-s}$.

4.4.2 State-based Scheduler Design: Symmetric Schedulers

The computation of the term $\mathbb{E}[\sum_{s=1}^{k-\tau_k} A^{s-1} w_{k-s} | \dot{f}_k, \dots, \dot{f}_{\tau_k+1} = 0]$, for a burst of non-transmissions of length greater than one, makes the estimate given in (4.26) hard to evaluate. This is because the quantized noise is not Gaussian. As a sub-

optimal, but simplified approach, consider the scheduling criterion given by

$$\gamma_k = f_{|\cdot|} \left(\sum_{s=1}^{k-\tau_{k-1}} A^{s-1} w_{k-s} \right) \quad \forall k, \quad (4.27)$$

where, $f_{|\cdot|}(r) = f_{|\cdot|}(-r)$.

Since τ_k is not defined without the MAC output δ_k in (4.10), we replace it with τ_{k-1} , which is also a measure of the non-transmission burst. The notation for the scheduling criterion $f_{|\cdot|}$ denotes a symmetric function of its arguments. Choosing the scheduler in this manner results in a zero mean estimate from the quantized noise when there is no transmission. Now, the estimate is easy to compute and a certainty equivalent control can be applied. This observation is summarized below, and is used to design the scheduler presented in section 4.4.3.

Proposition 4.8. *For the system (4.1)–(4.6), using the symmetric scheduling policy defined in (4.27) results in separation between the estimator and the scheduler, as well as an optimal certainty equivalent controller.*

4.4.3 The Dual Predictor Architecture

In this section, we examine closed loop design of the entire system. From the results of Theorems 4.4, 4.6 and Proposition 4.8, it is clear that the scheduler, observer and controller designs are coupled. It is not possible to design the optimal scheduling policy independently and combine it with a certainty equivalent controller and optimal observer to get the optimal closed loop system design. At the same time, solving for the jointly optimal scheduler, observer and controller is a hard problem.

Thus, we propose an architecture, shown in Fig. 4.6, for a design of the state-based scheduler, and the corresponding optimal controller and observer. There are 2 estimators in this architecture, and hence, we call it a dual predictor architecture (Ramesh et al., 2009). This architecture has been referred to previously in Xu et al. (2004), in the context of mobile networks. The scheduler, observer and controller blocks are described below.

Scheduler (S): The scheduler output γ_k is given by

$$\gamma_k = f(x_k, \hat{x}_{k|\tau_{k-1}}) = \begin{cases} 1 & |x_k - \hat{x}_{k|\tau_{k-1}}|^2 \geq \epsilon \\ 0 & \text{otherwise} \end{cases}. \quad (4.28)$$

Here, $\hat{x}_{k|\tau_{k-1}}$ is the estimate at the controller at time k if the current packet is not scheduled for transmission. To realize such a scheduling policy, the observer must be replicated within the scheduler, and for the observer to be able to subtract out the applied control, the controller must also be replicated within the scheduler. An explicit ACK is required to make these blocks work.

Theorem 4.9. For the closed loop system given by the plant in (4.1), the state-based dual predictor architecture, (4.28)–(4.31) and the cost criterion in (4.6), we state the following results:

- i. The estimate given by (4.29) minimizes the mean-squared estimation error.
- ii. The control signal in this architecture does not have a dual effect.
- iii. The certainty equivalence principle holds and the optimal control law is given by (4.31).
- iv. An expression for the LQG cost in this setup is given by

$$J_{DP} = \hat{x}_0^T S_0 \hat{x}_0 + \text{tr}\{S_0 P_0\} + \sum_{n=0}^{N-1} \text{tr}\{S_{n+1} R_w + (L_n^T (Q_2 + B^T S_{n+1} B) L_n) P_{n|n}\}, \quad (4.32)$$

where $P_{k|k}$ is the error covariance of the estimate at the observer. Also, $S_N = Q_0$ and S_k is obtained by the backward iteration in (4.20).

Proof. Evaluating the expression $\mathbb{E}[x_k | \mathbb{I}_k^C]$, we get

$$\begin{aligned} \mathbb{E}[x_k | \mathbb{I}_k^C] &= \begin{cases} \mathbb{E}[x_k | \delta_k = 1, y_k = x_k] & \delta_k = 1 \\ \mathbb{E}[x_k | \delta_{\tau_k} = 1, \boldsymbol{\delta}_{\tau_k+1}^k = 0, y_{\tau_k} = x_{\tau_k}, \mathbf{u}_0^{k-1}] & \delta_k = 0 \end{cases} \\ &= \begin{cases} x_k & \delta_k = 1 \\ \mathbb{E}[A^{k-\tau_k} x_{\tau_k} + \sum_{\ell=1}^{k-\tau_k} (A^{\ell-1} B u_{k-\ell} + A^{\ell-1} w_{k-\ell}) \\ \quad | \delta_{\tau_k} = 1, \boldsymbol{\delta}_{\tau_k+1}^k = 0, y_{\tau_k} = x_{\tau_k}, \mathbf{u}_0^{k-1}] & \delta_k = 0 \end{cases} \\ &= \begin{cases} x_k & \delta_k = 1 \\ A^{k-\tau_k} x_{\tau_k} + \sum_{n=1}^{k-\tau_k} A^{n-1} B u_{k-n} \\ \quad + \mathbb{E}[\sum_{\ell=1}^{k-\tau_k} A^{\ell-1} w_{k-\ell} | \boldsymbol{\delta}_{\tau_k+1}^k = 0] & \delta_k = 0 \end{cases} \\ &= \begin{cases} x_k & \delta_k = 1 \\ A \hat{x}_{k-1|k-1} + B u_{k-1} & \delta_k = 0 \end{cases}, \end{aligned}$$

where the last equality sign is obtained using the facts that

$$A^{k-\tau_k} x_{\tau_k} + \sum_{n=1}^{k-\tau_k} A^{n-1} B u_{k-n} = A \hat{x}_{k-1|k-1} + B u_{k-1},$$

$$\text{and } \mathbb{E}[\sum_{\ell=1}^{k-\tau_k} A^{\ell-1} w_{k-\ell} | \boldsymbol{\delta}_{\tau_k+1}^k = 0] = 0.$$

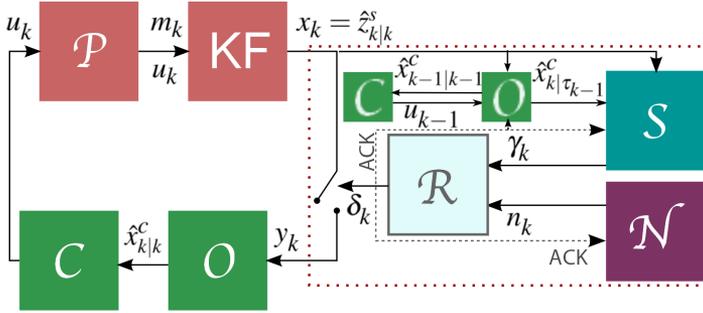


Figure 4.7: Measurement-based Dual Predictor Architecture: the innovations to the observer serve as input to the scheduler. The resulting setup is certainty equivalent. The observer is simple, and computes the minimum mean-square error estimate.

The last equation above is due to the use of a scheduling policy which is symmetric in its arguments, as defined in (4.27). Now, since the estimate in (4.29) is equal to $\mathbb{E}[x_k | \mathbb{I}_k^C]$, we know that this is a MMSE estimate (Kailath et al., 2000).

The error covariance at the estimator is given by (4.14), where, from (4.28) and (4.3), it is clear that the scheduler outcome γ_k and the MAC output δ_k do not depend on the applied controls u_0^{k-1} . Thus, the error covariance satisfies the definition in (4.12), and the control signal in this architecture does not have a dual effect.

From the above conclusion, note that the scheduling policy in (4.28) is of the form (4.24). Thus, from Corollary 4.7, we know that the optimal controller for this setup is certainty equivalent. Then, the optimal control signal is given by (4.19), which has the same form as the controller in this architecture (4.31). The expression for the control cost remains the same as in the case with partial state information, and is given by (4.32). \square

4.5 Measurement-based Scheduler

In this section, we extend the above results to a measurement-based system, or a system without full state information. We show that by placing an optimal observer, a Kalman Filter (KF) at the sensor, to estimate the state of the linear plant, and basing the scheduler decisions on this estimate, instead of on the measurement, we are able to re-establish the same problem formulation as before.

Consider a linear plant with a state z_k , and a measurement m_k given by

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k + w_{z,k}, \\ m_k &= Cz_k + v_{z,k}, \end{aligned} \quad (4.33)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $w_{z,k}$ is i.i.d. zero-mean Gaussian affecting the state z with covariance matrix $R_{w,z}$. The initial state z_0 is zero-mean Gaussian with

covariance matrix $R_{z,0}$. Also, the measurement $m \in \mathbb{R}^m$ and the matrix $C \in \mathbb{R}^{n \times m}$. The measurement noise $v_{z,k}$ is a zero mean i.i.d Gaussian process with covariance matrix $R_{v,z} \in \mathbb{R}^{m \times m}$, and it is independent of the process noise.

We can always place a Kalman Filter at the sensor node, which receives every measurement m_k from the sensor and updates its estimate ($\hat{z}_{k|k}^s$) as

$$\hat{z}_{k|k}^s = A\hat{z}_{k-1|k-1}^s + Bu_{k-1} + K_{f,k}e_k, \quad (4.34)$$

where, $K_{f,k}$ denotes the gain of the Kalman filter and e_k denotes the innovation in the measurement. The innovation can be proven to be Gaussian with zero-mean and covariance $R_{e,k}$. The error covariances for the predicted estimate and the filtered estimate are denoted P_k^s and $P_{k|k}^s$ respectively. These terms are given by

$$\begin{aligned} e_k &= m_k - C(A\hat{z}_{k-1|k-1}^s + Bu_{k-1}), \\ K_{f,k} &= P_k^s C^T R_{e,k}^{-1}, \\ R_{e,k} &= CP_k^s C^T + R_{v,z}, \\ P_k^s &= AP_{k-1|k-1}^s A^T + R_{w,z}, \\ P_{k|k}^s &= P_k^s - K_{f,k} R_{e,k} K_{f,k}^T. \end{aligned}$$

Now, if we use the estimate to define the state x of a linear plant, such that $x_k \triangleq \hat{z}_{k|k}^s$, we have a linear plant disturbed by i.i.d Gaussian process noise $w_k = K_{f,k}e_k$. Thus, we have re-established the problem setup from section 4.2.1, and the results from before can be applied to this plant. Note that the scheduler is now defined with respect to the estimate $\hat{z}_{k|k}^s$ and not the measurements m_k .

4.6 Examples

We present three examples in this section. The first example describes the problem setup, and illustrates the motivation for the problem. The second example illustrates the results of Theorem 4.1 and Lemma 4.2, which identify the dual role of the applied controls towards the information available to the controller. The third example illustrates the dual predictor architecture and provides an example of network-aware event triggering.

4.6.1 An Example of a Multiple Access NCS

This example illustrates the role of a state-based scheduler in our problem formulation in Section 4.2.1, where a number of closed loop systems share a contention-based multiple access network on the sensor link. We use a p -persistent CSMA protocol in the MAC. The observer and controller are chosen for simplicity of design, not as optimizers of any cost. We look at the performance of this network of control loops, with and without the state-based scheduler.

Table 4.8: A comparison of control costs with (J_{SS}) and without (J_{CN}) a state-based scheduler in the closed loop, from Example 4.1

Plant Type	$\mathcal{P}^{[T^1]}$	$\mathcal{P}^{[T^2]}$	$\mathcal{P}^{[T^3]}$
J_{CN}	45.3074	10.0028	6.1213
J_{SS}	23.5785	8.3489	5.3803

Example 4.1

We consider a heterogeneous network of 20 scalar plants, indexed by $j \in \{1, \dots, 20\}$. There are three different types of plants, $\mathcal{P}^{[T^1]}$, $\mathcal{P}^{[T^2]}$ and $\mathcal{P}^{[T^3]}$, given by

$$x_{k+1}^{(j)} = a^{[i]} x_k^{(j)} + u_k^{(j)} + w_k^{(j)}, \quad (4.35)$$

where $a^{[i]} \in \{1, 0.75, 0.5\}$, and $R_w^{[i]} \in \{1, 1.5, 2\}$, for the plant $\mathcal{P}^{[T^i]}$. The systems numbered $j \in \{1, \dots, 6\}$ are of type $\mathcal{P}^{[T^1]}$, $j \in \{7, \dots, 13\}$ are of type $\mathcal{P}^{[T^2]}$ and $j \in \{14, \dots, 20\}$ are of type $\mathcal{P}^{[T^3]}$. The plants are sampled with different periods given by $T^{[i]} \in \{10, 20, 25\}$, for the different types of plants, respectively. The state-based scheduler uses the criterion $x_k^{(j)^2} > \epsilon^{(j)}$. A p -persistent MAC, with synchronized slots, which permits three retransmissions is used. The persistence probability is given by $p_\alpha^{(r)}$, where r denotes the retransmission index, and $p_\alpha^{(r)} = \{1, 0.75, 0.5\}$ for $r \in \{1, \dots, 3\}$. The LQG criterion in (4.6), with $N = 10$ and $Q_0 = Q_1 = Q_2 = 1$ is used to design a certainty equivalent controller (4.19) as an ad hoc policy, not an optimal one, as we know from Corollary 4.7. The observer calculates a simple estimate as given by (4.29)-(4.30).

We look at the performance of a closed loop system in this network without a state-based scheduler, i.e., when $\epsilon^{(j)} = 0$ for all j . The cost of controlling the plants in the contention-based network, is denoted $J_{CN}^{[i]}$, and the values are listed in Table 4.8. We compare these to the costs obtained with a state-based scheduler in the closed loop system, denoted $J_{SS}^{[i]}$, when $\epsilon^{(j)} = 2.5$. There is a marked improvement with a state-based scheduler in the closed loop. Fig 4.9 depicts the state and the control signal for the first plant in this network.

The above improvement is obtained due to fewer collisions in the contention-based MAC. The non-zero scheduling threshold reduces the traffic in the network, and increases the probability of a successful transmission for all the plants in the network.

4.6.2 A 2-Step Horizon Example

We now look at a simple example to see the computational difficulties in identifying optimal estimates and controls for a system with a state-based scheduler in the closed loop. We also show that for an equivalent scheduler such as \tilde{f} in Section

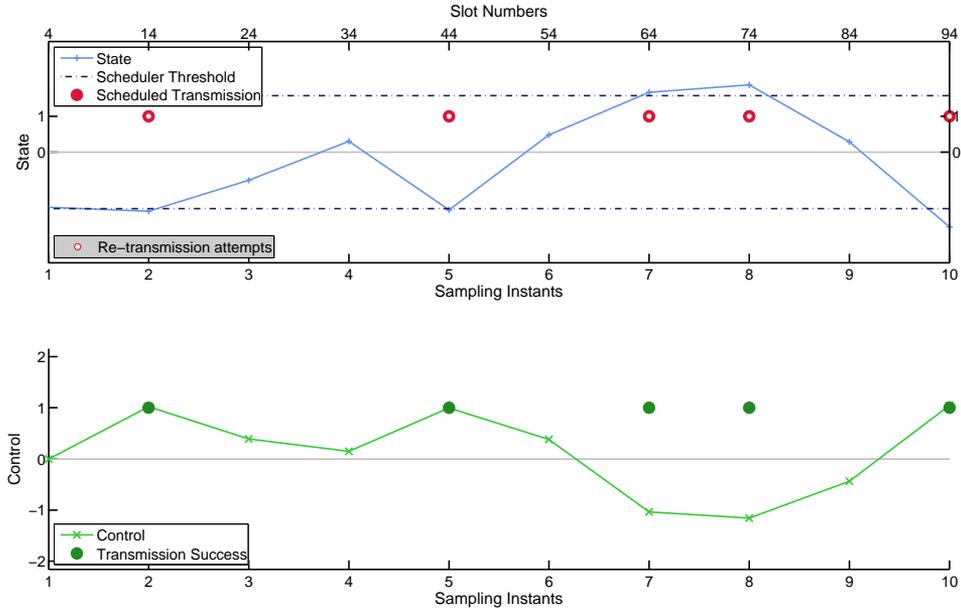


Figure 4.9: The state and the control signal with the channel use pattern. Note that the requested bound on the state, which is marked with a dotted line, is sometimes exceeded due to network traffic. Also, the control signal corresponds closely to the state only when there is a successful transmission.

4.3.2, which renders the control signal free of a dual effect, the entire plant is altered, so the equivalence construction does not work.

Example 4.2

Consider a scalar plant, given by $x_{k+1} = ax_k + bu_k + w_k$, with $a, b \in \mathbb{R}$ and $x_0, w_k \sim \mathcal{N}(0, 1)$. The scheduling law is given by

$$\delta_k = \begin{cases} 1 & x_k \geq 0.5 \\ 0 & \text{otherwise} \end{cases}.$$

Our aim is to find both the optimal controller, with dual effect, and the certainty equivalent controller for the equivalent scheduler and show that these result in different control actions for the same scheduling sequence. The controllers are designed to minimize the LQG cost (4.6), for a horizon of two steps, i.e., $N = 2$, and with $Q_0, Q_1, Q_2 \in \mathbb{R}$. We first derive the optimal controller with dual effect. Then, for the same schedule, we define the certainty equivalent controller, assuming that an equivalent scheduler of the form \tilde{f} in (4.15) has been designed. We compare the resulting control actions, and comment on the differences.

Estimator: We derive the estimates $\hat{x}_{0|0}$ and $\hat{x}_{1|1}$, the estimation errors $\tilde{x}_{0|0}$ and $\tilde{x}_{1|1}$, and the estimation error covariances $P_{0|0}$ and $P_{1|1}$, in Appendix A.1.1. The estimates are obtained using (4.26) in (A.1) and (A.4). Since the estimation error is non-Gaussian, we need to derive the probability density functions of the estimation errors at each time instant. This makes the computation of the estimation errors in (A.2) and (A.5), and the error covariances in (A.3) and (A.6), hard.

Optimal Controller: To solve for the optimal control signals, we use V_1 and V_0 from (4.17). The complete derivation of V_1 and V_0 are presented in the Appendix A.1.2. We find the control signal u_1 that minimizes V_1 , and get

$$u_1 = -\frac{abQ_0}{Q_2 + b^2Q_0}\hat{x}_{1|1}. \quad (4.36)$$

Then, to find u_0 , we take a partial derivative of the expression for V_0 with respect to u_0 and get

$$\frac{\partial V_0}{\partial u_0} = 2u_0(Q_2 + b^2S_1) + 2\hat{x}_{0|0}abS_1 + \frac{a^2Q_0^2b^2}{Q_2 + b^2Q_0} \cdot \frac{\partial}{\partial u_0} \left(\mathbb{E}[P_{1|1}|\mathbb{I}_0^c] \right) = 0. \quad (4.37)$$

This can be simplified using the expression for $P_{1|1}$. When $\delta_0 = 1$, we have

$$\frac{\partial V_0}{\partial u_0} = 2u_0(Q_2 + b^2S_1) + 2\hat{x}_{0|0}abS_1 - \frac{a^2Q_0^2b^2}{Q_2 + b^2Q_0}b(w_{0,max} - \bar{w}_0)^2\phi_{w_0}(w_{0,max}) = 0,$$

where $w_{0,max} = 0.5 - ax_0 - bu_0$. The final equation is obtained using Leibnitz rule. For the case when $\delta_0 = 0$, we have

$$\frac{\partial V_0}{\partial u_0} = 2u_0(Q_2 + b^2S_1) + 2\hat{x}_{0|0}abS_1 - \frac{a^2Q_0^2b^2}{Q_2 + b^2Q_0}b(e_{max} - \bar{e}_{\delta_0})^2\phi_{e_{\delta_0}}(e_{max}) = 0,$$

where $e_{max} = 0.5 - bu_0$ and again, Leibnitz rule was used. Solving these equations give the optimal u_0 for $\delta_0 = 1$ and 0, respectively.

CE Controller: For the same scheduler outcomes δ_0, δ_1 obtained through an equivalent scheduler which has no dual effect, the certainty equivalent controller gives us the control signals

$$\begin{aligned} u_1 &= -\frac{AbQ_0}{Q_2 + b^2Q_0}\hat{x}_{1|1} \\ u_0 &= -\frac{AbS_1}{Q_2 + b^2S_1}\hat{x}_{0|0} \end{aligned} \quad (4.38)$$

Note that the u_1 is found by minimizing V_1 , which results in the same expression as for the optimal controller (4.36). However, u_0 for the CE controller is obtained by solving the equation

$$2u_0(Q_2 + b^2S_1) + 2\hat{x}_{0|0}abS_1 = 0. \quad (4.39)$$

Discussion: A comparison of the control signals for the CE controller (4.38) with u_1 and u_0 obtained in (4.36) and (4.37), shows that the signal u_1 remains the same. However, u_0 is different, and displays a dual effect in the optimal controller. From (4.39), it is clear that the additional term in (4.37) alters the solution for the optimal controller.

This observation can be explained as follows. In a controller with a dual effect, the control signal can be chosen to probe the plant state in order to improve the quality of the estimate. However, there is no motive in improving the estimate in a one-step optimization process. Thus, u_1 is the same for both controllers. When the optimization is performed over two steps, a probing effect in the first step can improve the estimate and the corresponding control applied in the next step. Thus, u_0 is different in the optimal controller for a state-based scheduler.

This example shows us that even the same schedule can result in a different control sequence for a system without a dual effect. Thus, an equivalent construction for the scheduler does not result in an equivalent system in our setup.

4.6.3 An Example of the Dual Predictor Architecture

In this example, we present the dual predictor architecture, as applied to a shared network. We tune the threshold of the state-based scheduling law to probabilistically guarantee an achievable control performance, given the traffic over the network. We use a homogenous network in this example to simplify the comparison of control cost versus the scheduling threshold.

Example 4.3

We consider a shared network of 20 scalar plants, indexed by $j \in \{1, \dots, 20\}$ and given by (4.35), where $a^{(j)} = 1$ and $R_w^{(j)} = 1$ for all j . The plants are sampled with a period given by $T = 10$. The innovations-based scheduler uses a similar criterion to (4.28), where ϵ is the threshold of the scheduler. A p -persistent MAC, with synchronized slots, which permits three retransmissions is used. The persistence probability is given by $p_\alpha^{(r)}$, where r denotes the retransmission index and $p_\alpha^{(r)} = \{1, 0.75, 0.5\}$ for $r \in \{1, \dots, 3\}$. The LQG criterion in (4.6), with $N = 10$ and $Q_0 = Q_1 = Q_2 = 1$ is used to design the optimal certainty equivalent controller (4.19). The observer calculates the MMSE estimate given by (4.29)-(4.30).

The effect of varying ϵ on the control cost is shown in Fig. 4.10. For very high values of ϵ , the network is under-utilized, and almost all the transmissions are successful. However, the control cost is high as the number of transmissions is low. As we decrease ϵ , the control cost initially decreases due to increased use of the network. However, for very low values of ϵ , the network is over-utilized and this results in collisions. Thus, the control cost increases again, due to dropped packets.

Fig. 4.11 depicts the state and control signal of the first plant obtained from our simulation, for the best value of ϵ picked from the above plot. Note that the esti-

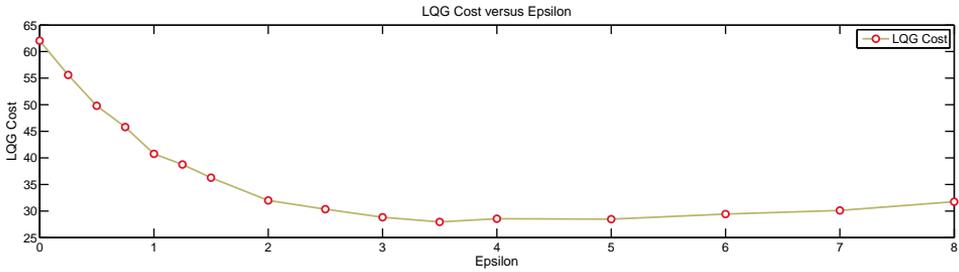


Figure 4.10: The control cost J_{DP} versus the scheduler threshold ϵ . For low thresholds, the high traffic in the network causes collisions, and a high J_{DP} . Very high values of ϵ result in an under-utilized network, and a high J_{DP} due to insufficient transmissions.

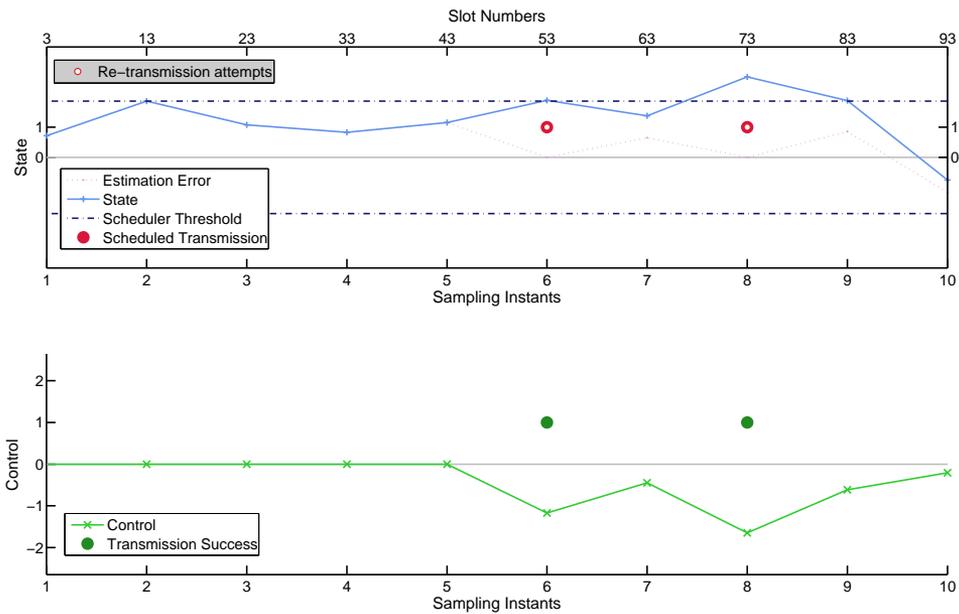


Figure 4.11: The estimation error, state and control signal with the channel use pattern. Note that the requested bound on the predicted estimation error, which is marked with a dotted line, is rarely exceeded. Also, the control signal corresponds closely to the state only when there is a successful transmission.

mation error is bounded, with a probability of 0.94, by the scheduling threshold, for the value $\epsilon^{(1)} = 3.5$, and the resulting control cost is $J_{\text{DP}} = 27.9235$.

It is interesting to note, in Fig. 4.10, that the minima is quite flat. Thus, it is not very important to use the optimal scheduling threshold ϵ .

4.7 Summary

This chapter describes the effects of a state-based schedulers in a network, on the control loop. We found that a state or measurement-based scheduler makes design of the optimal controller and observer hard. In general, certainty equivalence does not hold, unless the scheduler output is not a function of the past applied controls. Furthermore, a scheduling policy which is symmetric in its arguments reduces the complexity of the estimator. We used these results to propose a dual predictor architecture for closed loop systems with a state-based scheduler.

In the next chapter, we look at another example of a state-aware MAC, and analyze the performance of this MAC.

Attention-Based Tournaments

In this chapter, we look at a MAC which resolves contention strictly based on the priority allotted to the data packet. State-based priorities are assigned to data packets, based on the attention that each packet requires from across the network, making this another example of a state-aware MAC. The theoretical analysis on state-based schedulers presented in the previous chapter can also be used to find the optimal controller for the state-aware MAC presented in this chapter. In particular, we use innovation-based priorities in this chapter, which result in an optimal certainty equivalent controller for any closed loop systems using such a MAC.

We present tournaments in the MAC layer as a way to evaluate priorities and assign channel resources in a distributed manner. Priorities based on the attention emphasize the information content in the data to be transmitted and the related process dynamics. We present a performance analysis of this MAC for a heterogeneous network of dynamic linear plants. The use of attention-based priorities results in the same expected value of the net error covariance across the network, which can be obtained using a scheduling policy based on minimizing the per-sample variance of the error in the estimates.

5.1 Contributions and Related Work

Networks for control and estimation differ from other generic networks primarily due to the delay sensitive, critical nature of the dynamic processes involved. Performance optimization over control networks requires communication infrastructure that meets real-time constraints (Lian et al., 2001).

The time-criticality of data requires the introduction of some priority between sensors, while retaining a contention-based architecture. Previous attempts at introducing priorities within CSMA/CA include arbitration inter-frame space or contention window differentiation, such as in IEEE 802.11e. These are probabilistic measures that make it hard to analyze performance (Bianchi et al., 2005). A more certain method of ensuring priority is called for here. In this chapter, we introduce a prioritized access scheme with reserved slots, which are won through a tourna-

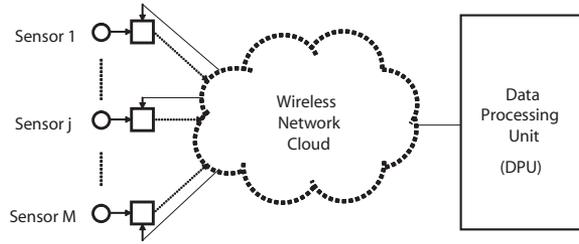


Figure 5.1: A typical scenario depicting Monitoring over a Wireless Network, along with the Multiple Access Solution presented in the paper.

ment. The idea of a tournament to resolve contention based on static priorities is already prevalent in literature, in the CAN Bus Protocol (Robert Bosch GmbH, 1991) and its recent adaptation to wireless networks in WiDOM (Pereira et al., 2007). However, the priority mechanism in our proposal is adapted to the plant state, and priorities are assigned to data packets, not to nodes.

Priorities adapted to the information in the current measurements are hard to assign for heterogenous sensors measuring vastly different physical quantities. We present a method of assigning these priorities. A node evaluates the criticality of the current measurement to be transmitted to the controller or monitoring unit and assigns an appropriate priority. The priority is a measure of the attention that a packet requires from the controller. A suitable *Attention Factor* is introduced. It ensures that the performance of the multiple access scheme converges to a centralized scheduling policy based on minimizing the per-sample variance of the error in the estimates obtained with limited communication resources. The corresponding throughput and delay analysis of such a Tournament Access Protocol are presented.

Note that the above formulation is suited to a more general setting, as the priorities are based on the innovations in the measurement. Thus, this formulation applies to both closed loop systems and other generic source-destination pairs. The outline of the chapter is as follows. Section 5.2 formulates the problem, for a network of control loops. The Attention Factor and the Tournament Access Protocol are given in Section 5.3. Section 5.4 presents the performance analysis of the protocol. Simulation results illustrating the system behaviour are given in Section 5.5.

5.2 Problem Formulation

Fig. 5.1 illustrates a network with M sensors attached to M independent physical processes, whose states are estimated over a wireless network by a data processing unit. From the perspective of a single sensing link, this system can be modelled as shown in Fig. 5.2. The process model is known to both the sensor and the data processing unit. The task of estimation is performed in the sensor node itself as it is assumed to have sufficient processing capability. The estimated state could be

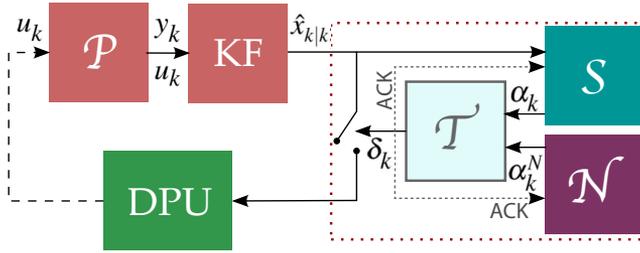


Figure 5.2: A model of the tournament access protocol from the perspective of a single sensing link. Note that this sensing link may or may not be part of a closed loop system. The actuating link only exists when it is part of a closed loop system.

used for monitoring, control or detection applications. The block \mathcal{N} represents the network as seen by this loop, and the block \mathcal{T} is the tournament access protocol (TAP), which determines whether this link or the rest of the network gets to access the shared medium. Note that in this chapter, the channel access request α_k is not a binary random variable. It is the priority allotted to the current data packet. Similarly, α_k^N is also not binary. Each of the blocks in Fig. 5.2 are explained below.

Sensor: We describe one of the sensors and its underlying process below. Each node senses a physical process \mathcal{P} with state $x \in \mathbb{R}^n$ and communicates the sensor reading $y \in \mathbb{R}^m$. The state and measurements are related by the process model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + v_k, \end{aligned} \quad (5.1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{m \times n}$ are constant matrices. If the physical process is not part of a control loop, there is no control term in the state equation, i.e., $B = 0$. The process noise w and the measurement noise v are i.i.d. zero mean Gaussian with covariance matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$, respectively. The initial state x_0 is zero-mean Gaussian with covariance matrix Q_0 .

KF: The best estimate of the state can be obtained from a KF placed at the sensor node itself, which receives every measurement y_k from the sensor and updates its filtered estimate ($\hat{x}_{k|k}$) and predicted estimate ($\hat{x}_{k|k-1}$) as

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_{f,k} e_k, \\ \hat{x}_{k|k-1} &= A\hat{x}_{k|k} + Bu_k, \end{aligned} \quad (5.2)$$

where $K_{f,k}$ denotes the Kalman gain and e_k denotes the innovation in the measurement. The innovation is defined as

$$e_k = y_k - C\hat{x}_{k|k-1}. \quad (5.3)$$

The Kalman gain is defined as

$$\begin{aligned} K_{f,k} &= P_{k|k-1} C^T R_{e,k}^{-1}, \\ R_e(k) &= C P_{k|k-1} C^T + R, \end{aligned}$$

where $R_{e,k}$ is the covariance of the innovation e_k . The prediction error covariance ($P_{k|k-1}$) and the filtered error covariance ($P_{k|k}$) are given by

$$\begin{aligned} P_{k|k-1} &= A P_{k-1|k-1} A^T + Q, \\ P_{k|k} &= P_{k|k-1} - K_{f,k} R_{e,k} K_{f,k}^T. \end{aligned}$$

State Based Priorities: There is a local scheduler \mathcal{S} , situated in the sensor node, between the plant and the network, which calculates the state-based priority, α_k , of the data packet. This block is formulated using a policy f , as given by

$$\alpha_k = f_k(I_k^S), \text{ where, } \mathbb{I}_k^S = \{ \{ \hat{x} \}_{0|0}^{k|k}, \mathbf{y}_0^{k-1}, \boldsymbol{\alpha}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}, \mathbf{u}_0^{k-1} \}. \quad (5.4)$$

Again, we use bold font to denote a set of variables such as $\mathbf{a}_t^T = \{a_t, a_{t+1}, \dots, a_T\}$.

Network: The network \mathcal{N} generates other traffic, with a symbolic priority α_k^N .

TAP: The tournament access protocol \mathcal{T} resolves contention between multiple simultaneous channel access requests, using the priorities α_k and α_k^N . The MAC output $\delta_k \in \{0, 1\}$ is given by $\delta_k = \mathcal{T}(\alpha_k, \alpha_k^N)$. The tournament access protocol chooses the packet with the highest priority in a contention-based setting.

DPU: There is a data processing unit situated across the network. This block utilizes the estimate of the state $\hat{x}_{k|k}$ in monitoring, control or detection applications. If there is a controller in the DPU, it issues a control signal using a policy g , defined on the information pattern of the controller, \mathbb{I}_k^C , as given by

$$u_k = g_k(\mathbb{I}_k^C), \text{ where, } \mathbb{I}_k^C = \{ \mathbf{z}_0^k, \boldsymbol{\delta}_0^k, \mathbf{u}_0^{k-1} \},$$

where, $z_k = \delta_k \hat{x}_{k|k}$ is the measurement available across the network.

The problem addressed in this chapter is how to assign adaptive priorities to data packets based on the measurements from the sensors. Each node must evaluate the criticality of its measurement in the context of a heterogenous sensor network, without communicating with its neighbors. In terms of the above formulation, the problem addressed is the design of the functional block f in (5.4). The rest of the chapter deals with the problem of formulating such a distributed adaptive priority and evaluating the performance of the resulting multiple access scheme.

5.3 Tournament Access Protocol

This section discusses the mathematical formulation of the state-based priority, called the attention factor, which is the adaptive priority assigned by each node to

a data packet due for transmission. Optimal properties of this priority measure are derived. Then, a multiple access protocol based on this attention factor is presented.

5.3.1 Attention Factor

The Attention Factor (α_k) is an adaptive priority designed to call the attention of the data processing unit to the current data in the network node or its deviation from the last known value (known to the data processing unit) and the penalty in not being able to transmit this packet.

The formulation below is described with respect to the state model of one node in the network. Recall the process model in (5.1). The network node assigns α_k to its own data packets. At time $k - 1$, the sensor delivers measurement y_{k-1} to its local estimator (KF). With this information, the estimator derives the estimate $\hat{x}_{k-1|k-1}$, which it sends to the DPU over the network. This unit can use its knowledge of the corresponding process model to generate the predicted estimates $\hat{x}_{k|k-1}, \hat{x}_{k+1|k-1}, \dots$. The motivation for allocating channel resources to deliver the next packet is the innovation e_k in the measurement y_k , given by (5.3). The filtered estimate can then be updated as given in (5.2). The risk in not being able to deliver this packet can be evaluated by computing the difference between the predicted estimates $\hat{x}_{k+1|k-1} = A\hat{x}_{k|k-1}$ and $\hat{x}_{k+1|k} = A\hat{x}_{k|k}$. The predicted estimate is used here to further emphasize the process dynamics, making α_k more sensitive to unstable processes.

We denote the attention associated with the measurement y_k (or data packet at time k) by α_k . The increase in the sample variance of the prediction error due to not receiving a packet at time k is denoted $P_{\text{samp}}\{y_k\}$ (or $\bar{\alpha}_k$, which is related to α_k). The notation $\bar{\alpha}_k$ is used in addition to $P_{\text{samp}}\{y_k\}$ to stress its relation to α_k or the attention factor, and $\bar{\alpha}_k$ is given by

$$\bar{\alpha}_k = P_{\text{samp}}\{y_k\} = \text{tr}\{(\hat{x}_{k+1|k} - \hat{x}_{k+1|k-1}) \cdot (\hat{x}_{k+1|k} - \hat{x}_{k+1|k-1})^T\}, \quad (5.5)$$

where $\text{tr}\{\cdot\}$ is the trace operator. $P_{\text{samp}}\{y_k\}$ is an empirical quantity based on knowledge of the measurement y_k . The expected value of $\bar{\alpha}_k$ is then proportional to the increase in the variance of the prediction error at the data processing unit due to not possessing information about the measurement y_k , as shown in

$$\mathbb{E}[\bar{\alpha}_k] = \text{tr}\{AK_{f,k} \mathbb{E}[e_k e_k^T] K_{f,k}^T A^T\} = \text{tr}\{P_{k+1|k-1} - P_{k+1|k}\}.$$

Applying this concept of attention as the adaptive priority used by data packets in the network to gain channel access, we can derive an interesting property. Let $\eta_k^{(j)}$ define the ratio of $P_{\text{samp}}\{y_k\}$ of the j^{th} node to $\mathbb{E}[P_{\text{sys}}\{\mathbf{y}_k\}]$, i.e.,

$$\eta_k^{(j)} = \frac{P_{\text{samp}}^{(j)}\{y_k\}}{\mathbb{E}[P_{\text{sys}}\{\mathbf{y}_k\}]},$$

where, $P_{\text{sys}}\{\mathbf{y}_k\}$ is the net increase in the prediction error variance due to not possessing information about measurements \mathbf{y}_k from all the nodes in the network

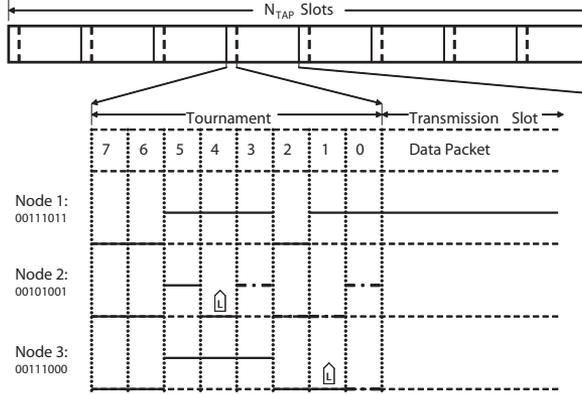


Figure 5.3: The Frame Structure of the Tournament Access Protocol. An example of a tournament is depicted below, where 'L' indicates that the node has lost the tournament.

at time k , and is given by

$$P_{\text{sys}}\{\mathbf{y}_k\} = \sum_{j=1}^M P_{\text{samp}}^{(j)}\{y_k\}.$$

In a prioritized access scheme, the channel is allotted to the data packet which maximizes $\eta_k^{(j)}$. Then,

$$\max_j \mathbb{E}[\eta_k^{(j)}] = \max_j \frac{\text{tr}\{P_{k+1|k-1}^{(j)} - P_{k+1|k}^{(j)}\}}{\sum_j \text{tr}\{P_{k+1|k-1}^{(j)} - P_{k+1|k}^{(j)}\}},$$

states that the data packet (measurement) which results in maximum reduction of the prediction error variance at the estimator is most likely to be allotted channel access. Thus, priorities based on $\bar{\alpha}_k$ minimize the net error variance of the estimates, which is equivalent to an optimal (per sample) scheduling strategy given limited communication resources.

Also, note that the choice of $\bar{\alpha}_k$ for the attention in (5.5), ensures that any closed loop system in this network is free of a dual effect, from Corollary 4.7. Now, the optimal control policy for a cost such as (4.6) is certainty equivalent, as defined in (4.19), and the MMSE estimate across the network is given by (4.29)–(4.30).

5.3.2 Tournament Access Protocol

Now that adaptive priorities have been assigned to the data packets, there remains the task of designing an arbitration policy to resolve contention. In other words, how should the data packets exchange priorities and decide who gets to transmit. The tournament access protocol (TAP) solves this problem.

The frame structure of the tournament access period is presented in Fig. 5.3. There are N_{TAP} tournament slots in each period, and sensors that wish to transmit in this frame must generate an Attention Factor (α) as described in Section 5.3.1. The formulation for $\bar{\alpha}$ in (5.5) is scaled and rounded to an integer α , given by

$$\alpha_k = \text{round} \left(\text{tr} \{ AK_{f,k} e_k e_k^T K_{f,k}^T A^T \} \cdot \frac{A_{\text{max}}}{P_{\text{smax}}} \right), \quad (5.6)$$

where, $P_{\text{smax}} = \text{tr} \{ KK_{f,k} R_{e,k} K_{f,k} K \}$. The discussion in Section 5.3.1 is unaffected by this modification for sufficiently large values of A_{max} , which is the largest value of attention. Here, P_{smax} can be thought to be the maximum tolerable increase in the sample variance of the prediction error due to not possessing information about a measurement from a node attached to a process with identity system matrix ($A = I$). This emphasizes the attention values for dynamic processes. K is a positive integer with which each process dictates its own tolerance limits and influences the increase of α with deviating measurements.

A tournament precedes every transmission slot in the TAP. During the tournament, qualifying packets transmit their Attentions, starting with the most significant bit. Nodes transmit a suitably chosen pulse for a bit of value one and remain silent during the zero bit. As wireless transceivers cannot transmit and receive at the same instant, nodes can listen during the zero bits. A busy channel indicates that they have lost the tournament. The packet(s) with the most number of ones in the attention factor wins the tournament. As the attention factors are assigned by each node, more than one packet can have the same attention factor and win the tournament. Multiple winners are not aware of each other, and cause a collision. Using the same mechanism as in CSMA/CA, nodes are aware of a collision by the lack of an acknowledgment (ACK). Fig. 5.3 illustrates the concept of a tournament between three nodes with attentions 59, 41 and 56 respectively. Nodes 2 and 3 lose the tournament after transmitting 4 and 7 bits of their priorities, as they hear a busy channel during their recessive bits. Node 1 wins the tournament and transmits in the succeeding slot.

5.4 Performance Analysis

The performance of a multiple access scheme is characterized by its throughput and delay. However, in the case of TAP, the probability of transmission conditioned on the attention factor is an important parameter. This computation requires a probability distribution for the attention factor, which is given by

$$\Pr(A_F = \alpha) = \begin{cases} \phi\left(0.5 \frac{P_{\text{smax}}}{A_{\text{max}} \sigma^2}\right) & \alpha = 0 \\ \phi\left(\left(\alpha + 0.5\right) \frac{P_{\text{smax}}}{A_{\text{max}} \sigma^2}\right) - \phi\left(\left(\alpha - 0.5\right) \frac{P_{\text{smax}}}{A_{\text{max}} \sigma^2}\right) & 0 < \alpha < A_{\text{max}} \\ 1 - \phi\left(\left(A_{\text{max}} - 0.5\right) \frac{P_{\text{smax}}}{A_{\text{max}} \sigma^2}\right) & \alpha = A_{\text{max}} \end{cases}, \quad (5.7)$$

where, $\phi(x)$ is the cumulative distribution function of the Chi Squared distribution. The sum of unnormalized squared Gaussian variables ($\text{tr}\{AK_{f,k}e_k e_k^T K_{f,k}^T A^T\}$) with unequal variances has a multivariate Gamma-type distribution, as discussed in [Krishnamoorthy and Parthasarathy \(1951\)](#). However, for process models with $m = 1$ ($e \in \mathbb{R}$), $\text{tr}\{AK_{f,k}e_k e_k^T K_{f,k}^T A^T\} \sim \sigma^2 \chi_1^2$, where χ_1^2 represents the Chi-squared distribution with one degree of freedom and σ^2 is the variance of the scalar term. If the variances of the innovation vector components are equal, then again, $\text{tr}\{AK_{f,k}e_k e_k^T K_{f,k}^T A^T\} \sim \sigma^2 \chi_m^2$, where m is the number of measurements in the process model. A motivation for reducing m is that higher order Chi-squared distributions tend towards a normal distribution, increasing the probability of collisions. This is explained in detail in Section 5.5.

The performance analysis is performed with respect to the average probability density of the attention variables of all the nodes participating in the tournament, which is referred to henceforth as $\mathbb{P}r_{AF}(\alpha)$. This is given by

$$\mathbb{P}r_{AF}(\alpha) = \frac{1}{M} \sum_{j=1}^M \mathbb{P}r^{(j)}(A_F = \alpha), \quad \text{for } 0 \leq \alpha \leq A_{\max}, \quad (5.8)$$

where M is the number of nodes, each of which are assumed to have a packet to transmit.

In any tournament slot, a packet can lose the tournament, or win the tournament. After winning the tournament, a packet can collide with another, or succeed in transmission. To derive the probabilities of these events, we define quantities p_L, p_{LE}, p_G in

$$\begin{aligned} p_L &= \mathbb{P}r(L|\acute{\alpha}) = \mathbb{P}r(\alpha < \acute{\alpha}), \quad \text{where, } \mathbb{P}r(\alpha < \acute{\alpha}) = \sum_{\alpha < \acute{\alpha}} \mathbb{P}r_{AF}(\alpha), \\ p_{LE} &= \mathbb{P}r(LE|\acute{\alpha}) = \mathbb{P}r(\alpha \leq \acute{\alpha}), \quad \text{where, } \mathbb{P}r(\alpha \leq \acute{\alpha}) = \sum_{\alpha \leq \acute{\alpha}} \mathbb{P}r_{AF}(\alpha), \\ p_G &= \mathbb{P}r(G|\acute{\alpha}) = \mathbb{P}r(\alpha > \acute{\alpha}), \quad \text{where, } \mathbb{P}r(\alpha > \acute{\alpha}) = 1 - \mathbb{P}r(LE|\acute{\alpha}). \end{aligned} \quad (5.9)$$

These quantities refer to the conditional probability of another packet with attention less than (p_L), less than or equal to (p_{LE}) and greater than (p_G) a given value $\acute{\alpha}$.

Now, we arrive at the conditional probability (conditioned on the attention α) of winning a tournament in N slots against $M - 1$ other packets, as given in

$$\mathbb{P}r(W_{N,M-1}|\alpha) = \sum_{n=0}^{N-1} C_n^{M-1} p_{LE}^{M-1-n} p_G^n, \quad (5.10)$$

where $C_k^n = \frac{n!}{(n-k)!k!}$ refers to the binomial coefficient. This equation states that there can be only up to $N - 1$ packets with attentions greater than any value $\acute{\alpha} \in \alpha$ and that the rest must have attentions less than or equal to $\acute{\alpha}$. There can be more

than $N - 1$ packets with attentions greater than $\acute{\alpha}$, but these additional packets must have equal attentions and collide. To simplify the analysis, we assume that the probability of collisions in previous slots is negligible. This assumption is valid for correct choice of the parameter A_{\max} , as shown in Section 5.5. Also, a design of TAP based on this assumption increases the throughput.

The conditional probability of losing tournaments in all N slots against $M - 1$ packets is then given by

$$\Pr(L_{N,M-1}|\alpha) = 1 - \Pr(W_{N,M-1}|\alpha) .$$

The conditional probability of succeeding in transmission in N slots against $M - 1$ packets is given in

$$\Pr(Tx_{N,M-1}|\alpha) = \sum_{n=0}^{N-1} C_n^{M-1} p_L^{M-1-n} p_G^n , \quad (5.11)$$

which differs from (5.10) by requiring that the other packets have attentions strictly less than any value $\acute{\alpha} \in \alpha$. Finally, the conditional probability of a collision under these circumstances is given by

$$\Pr(C_{N,M-1}|\alpha) = \Pr(W_{N,M-1}|\alpha) - \Pr(Tx_{N,M-1}|\alpha) .$$

The probability of successfully transmitting a packet in N_{TAP} slots against $M - 1$ other packets is obtained by setting $N = N_{\text{TAP}}$ in (5.11). We can then define the probability of a successful transmission ($\Pr(\delta_k = 1)$) for a specific sensor link, such as the one shown in Fig. 5.2, as

$$\Pr(\delta_k = 1) = \sum_{\alpha_k} \Pr(Tx_{N,M-1}|\alpha_k) \Pr_L(\alpha_k) ,$$

where, $\Pr_L(\alpha_k)$ is the probability density function of the attention of the link under consideration, which can be quite different from the average probability density function $\Pr_{AF}(\alpha_k)$ for the network. We can now substitute for the probability of a successful transmission in the expression for the error covariance in (3.11), and in the control cost in (4.32).

We can also define an attention specific throughput, or the fraction of time that useful information is carried on the network, as given by

$$S_{\text{TAP}}(\alpha) = \frac{\Pr(Tx_{N,M-1}|\alpha) \Pr_{AF}(\alpha) \text{Len}(P)}{T_{\text{TAP}}} ,$$

where $\text{Len}(P)$ is the packet payload size and T_{TAP} is the length of the Tournament Access Period (TAP). The attention specific throughput is a more useful parameter as it is a direct indicator of the performance of our prioritized MAC scheme. Packets with different attention factors (or priorities) view the medium as a channel

with throughput $S_{\text{TAP}}(\alpha)$, where α is the attention of the packet. Averaged over $\Pr_{AF}(\alpha)$, the mean throughput (S_{TAP}) is given by

$$S_{\text{TAP}} = \sum_{A=0}^{A_{\max}} S_{\text{TAP}}(\alpha) \approx \frac{N_{\text{TAP}} \cdot \text{Len}(P)}{T_{\text{TAP}}}.$$

For $M > N_{\text{TAP}}$ and a negligible probability of collision, this can be expressed more simply as shown.

We define the average conditional delay ($\mathbb{E}[d|\alpha]$) in terms of the number of frames by which the packet has been delayed, as given in

$$\mathbb{E}[d|\alpha] = 1 \cdot \Pr(Tx|\alpha_{[1]}) + 2 \cdot (1 - \Pr(Tx|\alpha_{[1]})) \Pr(Tx|\alpha_{[2]}),$$

where $\Pr(Tx|\alpha_{[d]})$ refers to $\Pr(Tx_{N,M-1}|\alpha_{k+d})$ from (5.11) for $N = N_{\text{TAP}}$ slots in the d^{th} frame after the packet was generated. Higher order terms are neglected, as the probability of their occurring is designed to be small. The attention factor as a function of delay is given by $\alpha_{k+d} = \text{tr}\{AK_{f,k+d}e_{k+d}e_{k+d}^T K_{f,k+d}^T A^T\}$.

The above analysis has assumed M nodes with a packet each in every frame. A more realistic scenario would be to consider M nodes with average packet generation rates $\{\lambda_m\}$. Then,

$$M\bar{\lambda} \leq N_{\text{TAP}}, \quad \text{where, } \bar{\lambda} = \frac{\sum_{m=0}^M \lambda_m}{M},$$

provides a limit on the average rate ($\bar{\lambda}$) for a cluster of linear processes, where N_{TAP} is the number of packets that this Medium Access Control Layer can support per frame.

5.5 Simulation Results

In this section, we simulate TAP in Matlab and present results that give us an insight into how the attention factor varies with system dynamics and delays. These results also provide a validation of our analysis.

5.5.1 Results

Consider a network with $M = 20$ nodes. Each node generates a packet to transmit, and these packets vie for $N_{\text{TAP}} = 10$ tournament slots. The maximum value that the attention can take is $A_{\max} = 255$, which is sufficiently large to prevent frequent collisions while maximizing throughput.

The parameter K , referred to in (5.6), can be set to 5, which is sufficient to produce a decaying probability distribution curve, as shown in Fig. 5.4a. Lower values of K result in a peak at the higher end of \Pr_{AF} (defined in (5.8)), and higher values of K under utilize the range of \Pr_{AF} . Critical nodes must set a

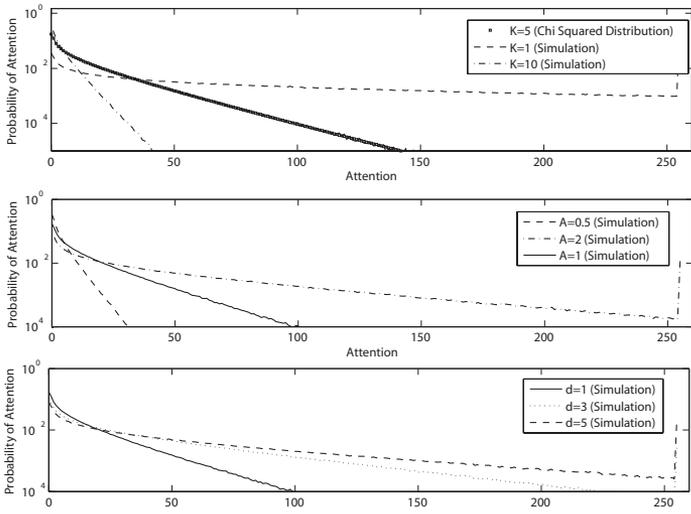


Figure 5.4: This figure shows the probability distribution of the attention factor (a) for different values of K , (b) for processes with different dynamics (different A) and (c) with delay (d).

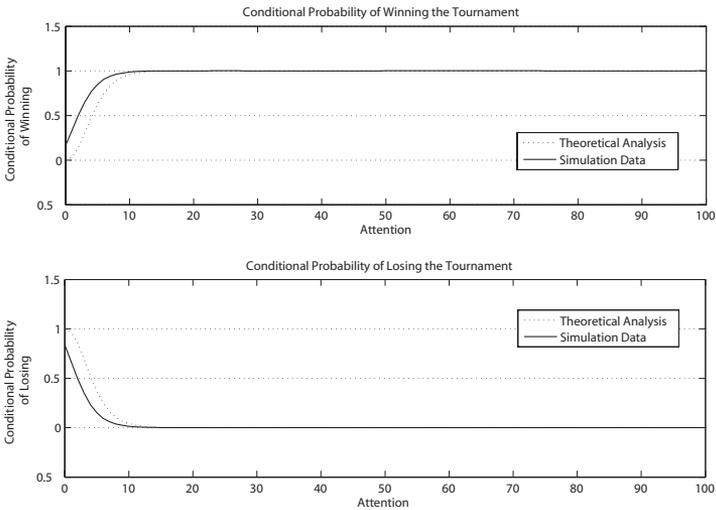


Figure 5.5: This figure shows the conditional probabilities of winning and losing in TAP. Note that the simulated data matches the analytical values.

lower K to generate packets with high attention values for small deviations in the measurements.

In these simulations, we consider models with a single state variable and measurement ($n = 1, m = 1$ in (5.1)). Fig. 5.4b illustrates the variation in the probability distribution of the attention factor for process models with different dynamics. Stable processes ($A = 0.5$) probabilistically generate lower values of attentions than unstable processes ($A = 2$). Fig. 5.4c illustrates the variation in the probability distribution of the attention factor with delays. Note that a delayed packet is more likely to generate a higher attention factor. Thus, the attention factor satisfies the basic requirements of an adaptive priority based on measurements, namely that it highlights the process dynamics and any suffered delay.

Next, the tournament access period was simulated in Matlab. A homogenous network was considered for simplicity, with model parameter $A = 1$. The attention factor was generated in the same frame as the data packet ($d = 1$). The results matched the analysis closely, as shown in Figs. 5.5 and 5.6. The conditional probability of winning, transmission and losing are close to the analytical values because the probability of a collision is negligibly small for high values of α .

The conditional probabilities of winning and transmission are almost 1 for packets with high attentions. The peak in the conditional probability of collision (Fig. 5.6) can be explained from the probability density function of α . The probability distribution of the attention $\mathbb{P}_{AF}(\alpha)$ indicates that there are few packets with large attentions. These are most likely to win the tournament in the first few slots and transmit without collision. Hence, the curve falls to nearly 0 for high values of α . Packets with lower values of α mostly win the tournament in the last few slots, and since there are many such packets, collisions are very likely. Finally, the packets with very low values of attention do not win the tournament often, and hence the probability of collision is low for these values.

5.5.2 Discussion

We can now make some inferences from these results, which could help us improve the design of TAP. We would like to design TAP such that the conditional probability of transmitting in N_{TAP} slots, $\mathbb{P}_{\text{T}}(Tx_{N_{\text{TAP}}, M-1} | \alpha) \approx 1$, and the conditional probability of a collision, $\mathbb{P}_{\text{T}}(C_{N_{\text{TAP}}, M-1} | \alpha) \approx 0$, for higher values of attentions. To ensure these properties hold, it is essential that the probability distribution of α tapers away consistently. If the curve were more bell-shaped, for instance, the probability of collision would be higher for the mid-range values of α as there would be more packets with these values.

From Fig. 5.6, it is clear that the throughput of TAP will be significantly affected due to a large number of collisions in the final few slots. This can be treated as a design constraint and the number of slots (N_{TAP}) chosen to be fewer than the number of packets with attentions greater than a value A_{min} . A_{min} should be chosen from Fig. 5.6, such that for $\alpha > A_{\text{min}}$, $\mathbb{P}_{\text{T}}(C_{N, M-1} | \alpha) \approx 0$.

But, this does not solve the problem, since a well designed $\mathbb{P}_{AF}(\alpha)$ will ensure

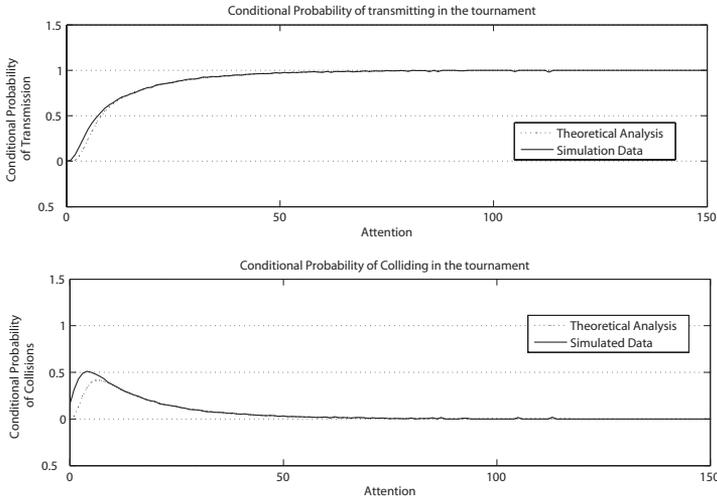


Figure 5.6: Conditional Probabilities of Transmission and Collision in TAP. The conditional probability of collision shows a peak at low values of attention, which indicates that TAP is not well suited for this range of attentions. TAP could be used along with Slotted CSMA/CA in a Hybrid MAC, with the tournament slots reserved for packets with higher attentions.

that most packets in the network have low values of attentions. These packets should clearly not use the tournament, but must still be given a chance to contend for the channel. These packets would be better off using a contention access scheme such as Slotted CSMA/CA, since their priorities are nearly equal. Thus, TAP is well suited for a hybrid Medium Access Control along with Slotted CSMA/CA. The tournament slots are simply reserved slots for packets with higher priorities.

Finally, since only packets with α greater than A_{\min} are to use the tournament slots, it could be effective to scale the attentions within the range ($R : A_{\min} \leq \alpha \leq A_{\max}$) over the entire range of α ($0 \leq \alpha \leq A_{\max}$). Here, α_R is the new scaled attention within the range R . This translation retains the order of priorities in α , but uniformly spreads each value over a number of values in the new scale α_R . Now, the probability of another packet with the same priority (α_R) is lowered, which reduces the probability of a collision and increases the throughput of TAP.

5.6 Summary

We have presented another example of a state-aware MAC, with an adaptive priority formulation called the Attention Factor, as well as an arbitration mechanism called the tournament that uses this priority. We were able to analyze the performance of

this multiple access scheme, which is required to quantify the estimation error or the control cost across the network.

To implement tournaments in sensor nodes is not a trivial task. Even if it is possible, it comes at the cost of throughput, as the bit-level synchronization requirements of this protocol are hard to achieve with sensor nodes. Also, this mechanism requires nodes to switch the receiver mode from transmission to reception during the priority bits, and a short turn-around time may be hard to achieve given today's transceiver technology.

Control over Mesh Networks

Much of the current theory of networked control systems uses simple point-to-point communication models as an abstraction of the underlying network. As a result, the controller has very limited information on the network conditions and performs suboptimally. This work models the underlying wireless multihop mesh network as a graph of links with transmission success probabilities, and uses a recursive Bayesian estimator to provide packet delivery predictions to the controller. The predictions are a joint probability distribution on future packet delivery sequences, and thus capture correlations between successive packet deliveries. We look at finite horizon LQG control over a lossy actuation channel and a perfect sensing channel, both without delay, to study how the controller can compensate for predicted network outages.

6.1 Introduction

Increasingly, control systems are operated over large-scale, networked infrastructures. In fact, several companies today are introducing devices that communicate over low-power wireless mesh networks for industrial automation and process control ([Wireless Industrial Networking Alliance, 2010](#); [International Society of Automation, 2010](#)). While wireless mesh networks can connect control processes that are physically spread out over a large space to save wiring costs, these networks are difficult to design, provision, and manage ([Chlamtac et al., 2003](#); [Bruno et al., 2005](#)). Furthermore, wireless communication is inherently unreliable, introducing packet losses and delays, which are detrimental to control system performance and stability.

Research in the area of Networked Control Systems (NCSs) ([Hespanha et al., 2007](#)) addresses how to design control systems which can account for the lossy, delayed communication channels introduced by a network. Traditional tasks in control systems design, like stability/performance analysis and controller/estimator synthesis, are revisited, with network models providing statistics about packet losses and delays. In the process, the studies highlight the benefits and drawbacks of different

system architectures. For example, Figure 6.1 depicts the general system architecture of a networked control system over a mesh network proposed by Robinson and Kumar (Robinson and Kumar, 2007). A fundamental architecture problem is how to choose the best location to place the controllers, if they can be placed at any of the sensors, actuators, or communication relay nodes in the network. One insight from Schenato et al. (Schenato et al., 2007) is that if the controller can know whether the control packet reaches the actuator, e.g., we place the controller at the actuator, then the optimal LQG controller and estimator can be designed separately (the separation principle).

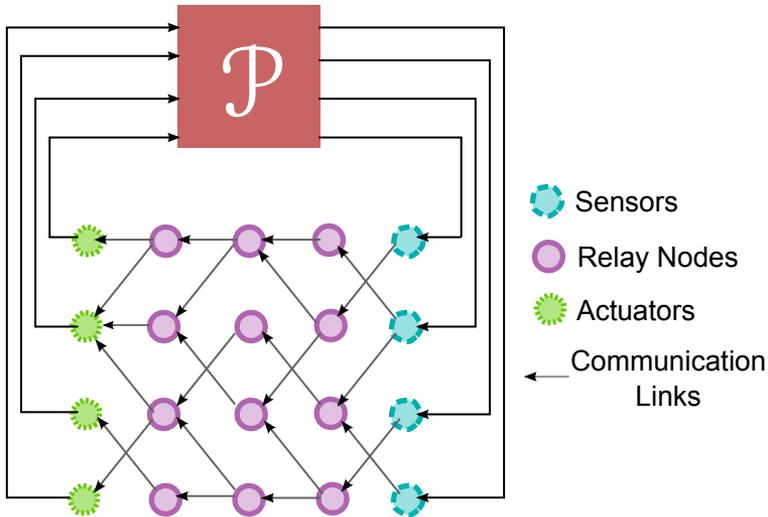


Figure 6.1: A networked control system over a mesh network, where the controllers can be located on any node.

To gain more insights on how to architect and design NCSs, two limitations in the approach of many current NCS research studies need to be addressed. The first limitation is the use of simple models of packet delivery over a point-to-point link or a star network topology to represent the network, which are often multihop and more complex. The second limitation is the treatment of the network as something designed and fixed a priori before the design of the control system. Very little information is passed through the interface between the network and the control system, limiting the interaction between the two “layers” to tune the controller to the network conditions, and vice versa.

6.1.1 Related Works

Schenato et al. (Schenato et al., 2007) and Ishii (Ishii, 2008) study stability and controller synthesis for different control system architectures, but they both model

networks as i.i.d. Bernoulli processes that drop packets on a single link. The information passed through the interface between the network and the control system is the packet drop probability of the link, which is assumed to be known and fixed. Seiler and Sengupta (Seiler and Sengupta, 2005) study stability and \mathcal{H}_∞ controller synthesis when the network is modeled as a packet-dropping link described by a two-state Markov chain (Gilbert-Elliott model), where the information passed through the network-controller interface are the transition probabilities of the Markov chain. Elia (Elia, 2005) studies stability and the synthesis of a stabilizing controller when the network is represented by an LTI system with stochastic disturbances modeled as parallel, independent, multiplicative fading channels.

Some related work in NCSs do use models of multihop networks. For instance, work on consensus of multi-agent systems (Olfati-Saber et al., 2007) typically study how the connectivity graph(s) provided by the network affects the convergence of the system, and is not focused on modeling the links. Robinson and Kumar (Robinson and Kumar, 2007) study the optimal placement of a controller in a multihop network with i.i.d. Bernoulli packet-dropping links, where the packet drop probability is known to the controller. Gupta et al. (Gupta et al., 2009) study how to optimally process and forward sensor measurements at each node in a multihop network for optimal LQG control, and analyze stability when packet drops on the links are modeled as spatially-independent Bernoulli, spatially-independent Gilbert-Elliott, or memoryless spatially-correlated processes.¹ Varagnolo et al. (Varagnolo et al., 2008) compare the performance of a time-varying Kalman filter on a wireless TDMA mesh network under unicast routing and constrained flooding. The network model describes the routing topology and schedule of an implemented communication protocol, TSMP (Pister and Doherty, 2008), but it assumes that transmission successes on the links are spatially-independent and memoryless. Both Gupta et al. (Gupta et al., 2009) and Varagnolo et al. (Varagnolo et al., 2008) are concerned with estimation when packet drops occur on the sensing channel, and the estimators do not need to know network parameters like the packet loss probability.

6.1.2 Contributions

Our approach is a step toward using more sophisticated, multihop network models and passing more information through the interface between the controller and the network. Similar to Gupta et al. (Gupta et al., 2009), we model the network routing topology as a graph of independent links, where transmission success on each link is described by a two-state Markov chain. The network model consists of the routing topology and a global TDMA transmission schedule. Such a minimalist network model captures the essence of how a network with bursty links can have correlated packet deliveries (Willig et al., 2002), which are particularly bad for control when they result in bursts of packet losses. Using this model, we propose a network estimator to estimate, without loss of information, the state of the network

¹Here, “spatially” means “with respect to other links.”

given the past packet deliveries.² The network estimate is translated to a joint probability distribution predicting the success of future packet deliveries, which is passed through the network-controller interface so the controller can compensate for unavoidable network outages. The network estimator can also be used to notify a network manager when the network is broken and needs to be reconfigured or reprovisioned, a direction for future research.

Section 6.2 describes our plant and network models. We propose two network estimators, the Static Independent links, Hop-by-hop routing, Scheduled (SIHS) network estimator and the Gilbert-Elliott Independent links, Hop-by-hop routing, Scheduled (GEIHS) network estimator in Section 6.3. Next, we design a finite-horizon, Future-Packet-Delivery-optimized (FPD) LQG controller to utilize the packet delivery predictions provided by the network estimators, presented in Section 6.4. Section 6.5 provides an example and simulations demonstrating how the GEIHS network estimator combined with the FPD controller can provide better performance than a classical LQG controller or a controller assuming i.i.d. packet deliveries. Finally, Section 6.7 describes the limitations of our approach and future work.

6.2 Problem Formulation

This chapter studies an instance of the general system architecture depicted in Figure 6.1, with a single control loop containing one sensor and one actuator. One network estimator and one controller are placed at the sensor, and we assume that an end-to-end acknowledgement (ACK) that the controller-to-actuator packet is delivered is always received at the network estimator, as shown in Figure 6.2. For simplicity, we assume that the plant dynamics are significantly slower than the end-to-end packet delivery deadline, so that we can ignore the delay introduced by the network. The general problem is to jointly design a network estimator and controller that can optimally control the plant using our proposed SIHS and GEIHS network models. In our problem setup, the controller is only concerned with the past, present, and future packet delivery sequence and not with the detailed behavior of the network, nor can it affect the behavior of the network. Therefore, the network estimation problem decouples from the control problem. The information passed through the network-controller interface is the packet delivery sequence, specifically the joint probability distribution describing the future packet delivery predictions.

6.2.1 Plant and Network Models

The state dynamics of the plant \mathcal{P} in Figure 6.2 is given by

$$x_{k+1} = Ax_k + \nu_k Bu_k + w_k \quad , \quad (6.1)$$

²Strictly speaking, we obtain the probability distribution on the states of the network, not a single point estimate.

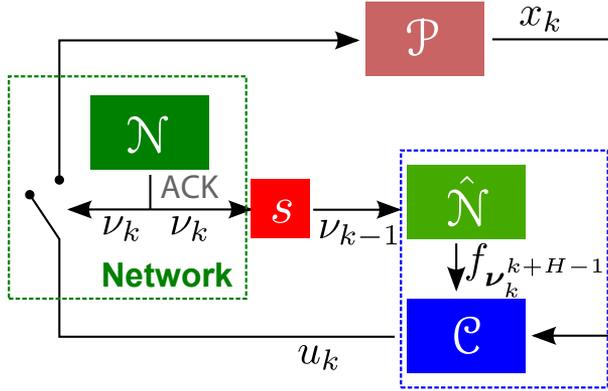


Figure 6.2: A control loop for plant \mathcal{P} with the network on the actuation channel. The network estimator $\hat{\mathcal{N}}$ passes packet delivery predictions $f_{\nu_k}^{k+H-1}$ to the FPD controller \mathcal{C} , with past packet delivery information obtained from the network \mathcal{N} over an acknowledgement (ACK) channel.

where $A \in \mathbb{R}^{\ell \times \ell}$, $B \in \mathbb{R}^{\ell \times m}$, and w_k are i.i.d. zero-mean Gaussian random variables with covariance matrix $R_w \in \mathbb{S}_+^{\ell}$, where \mathbb{S}_+^{ℓ} is the set of $\ell \times \ell$ positive semidefinite matrices. The initial state x_0 is a zero-mean Gaussian random variable with covariance matrix $R_0 \in \mathbb{S}_+^{\ell}$ and is mutually independent of w_k . The binary random variable ν_k indicates whether a packet from the controller reaches the actuator ($\nu_k = 1$) or not ($\nu_k = 0$), and each ν_k is independent of x_0 and w_k (but the ν_k 's are not independent of each other).

Let the discrete sampling times for the control system be indexed by k , but let the discrete time for schedule time slots (described below) be indexed by t . The time slot intervals are smaller than the sampling intervals. The time slot when the control packet at sample time k is generated is denoted t_k , and the deadline for receiving the control packet at the receiver is t'_k . We assume that $t'_k \leq t_{k+1}$ for all k . Figure 6.3 illustrates the relationship between t and k .

The model of the TDMA wireless mesh network (\mathcal{N} in Figure 6.2) consists of

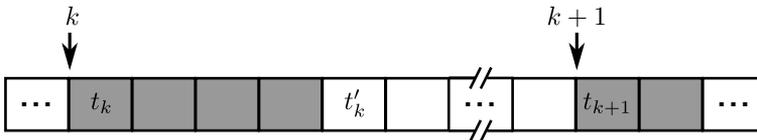


Figure 6.3: The packet containing the control input u_k is generated right before time slot t_k . The packet may be in transit through the network in the shaded time slots, until right before time slot t'_k . Thus, time t_k is aligned with the beginning of the time slot.

a routing topology G , a link model describing how the transmission success of a link evolves over time, and a fixed repeating schedule $\mathbf{F}^{(T)}$. The SIHS network model and the GEIHS network model only differ in the link model. Each of these components will be described in detail below.

The routing topology is described by $G = (\mathcal{V}, \mathcal{E})$, a connected directed acyclic graph with the set of vertices (nodes) $\mathcal{V} = \{1, \dots, M\}$ and the set of directed edges (links) $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$, where the number of edges is denoted E . The source node is denoted a and the sink (destination) node is denoted b . Only the destination node has no outgoing edges.

At any moment in time, the links in G can be either be up (succeeds if attempt to transmit packet) or down (fails if attempt to transmit packet). Thus, there are 2^E possible topology realizations $\tilde{G} = (\mathcal{V}, \tilde{\mathcal{E}})$, where $\tilde{\mathcal{E}} \subseteq \mathcal{E}$ represents the edges that are up.³

At time t_k , the actual state of the topology is one of the topology realizations but it is not known to the network estimator. With some abuse of terminology, we define $G^{(k)}$ to be the random variable representing the state of the topology at time t_k .⁴

This chapter considers the network under two link models, the static link model and the Gilbert-Elliott (G-E) link model. Both network models assume all the links in the network are independent.

The static link model assumes the links do not switch between being up and down while packets are sent through the network. Therefore, the sequence of topology realizations over time is constant. While not realistic, it leads to the simple network estimator in Section 6.3.1 for pedagogical purposes. The a priori transmission success probability of link $l = (i, j)$ is p_l .

The G-E link model represents each link l by the two-state Markov chain shown in Figure 6.4. At each sample time k , a link in state 0 (down) transitions to state 1 (up) with probability p_l^u , and a link from state 1 transitions to state 0 with probability p_l^d .⁵ The steady-state probability of being in state 1, which we use as the a priori probability of the link being up, is

$$p_l = p_l^u / (p_l^u + p_l^d) \quad .$$

The fixed, repeating schedule of length T is represented by a sequence of matrices $\mathbf{F}^{(T)} = (F^{(1)}, F^{(2)}, \dots, F^{(T)})$, where the matrix $F^{(t-1 \pmod{T}+1)}$ represents the links scheduled at time t . The matrix $F^{(t)} \in \{0, 1\}^{M \times M}$ is defined from the set $\mathcal{F}^{(t)} \subseteq \mathcal{E}$ containing the links scheduled for transmission at time t . We assume that nodes can only unicast packets, meaning that for all nodes i , if $(i, j) \in \mathcal{F}^{(t)}$ then for all $v \neq j$, $(i, v) \notin \mathcal{F}^{(t)}$. Furthermore, a node holds onto a packet if the transmission

³Symbols with a tilde ($\tilde{\cdot}$) denote values that can be taken on by random variables, and can be the arguments to probability distribution functions (pdfs).

⁴Strictly speaking, $G^{(k)}$ is a function mapping events to the set of all topology realizations, not to the set of real numbers.

⁵We can easily instead use a G-E link model that advances at each time step t , but it would make the following exposition and notation more complicated.

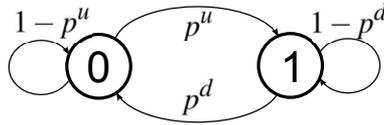


Figure 6.4: Gilbert-Elliott link model

fails and can retransmit the packet the next time an outgoing link is scheduled (hop-by-hop routing). Thus, the matrix $F^{(t)}$ has entries

$$F_{ij}^{(t)} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{F}^{(t)}, \text{ or} \\ & \text{if } i = j \text{ and } \forall v \in \mathcal{V}, (i, v) \notin \mathcal{F}^{(t)} \\ 0 & \text{otherwise.} \end{cases}$$

An exact description of the network consists of the sequence of topology realizations over time and the schedule $\mathbf{F}^{(T)}$. Assuming a topology realization \tilde{G} , the links that are scheduled and up at any given time t are represented by the matrix $\tilde{F}^{(t; \tilde{G})} \in \{0, 1\}^{M \times M}$, with entries

$$\tilde{F}_{ij}^{(t; \tilde{G})} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{F}^{(t)} \cap \tilde{\mathcal{E}}, \text{ or} \\ & \text{if } i = j \text{ and } \forall v \in \mathcal{V}, (i, v) \notin \mathcal{F}^{(t)} \cap \tilde{\mathcal{E}} \\ 0 & \text{otherwise.} \end{cases} \quad (6.2)$$

Define the matrix $\tilde{F}^{(t, t'; \tilde{G})} = \tilde{F}^{(t; \tilde{G})} \tilde{F}^{(t+1; \tilde{G})} \dots \tilde{F}^{(t'; \tilde{G})}$, such that entry $\tilde{F}_{ij}^{(t, t'; \tilde{G})}$ is 1 if a packet at node i at time t will be at node j at time t' , and is 0 otherwise. Since the destination b has no outgoing links, a packet sent from the source a at time t reaches the destination b at or before time t' if and only if $\tilde{F}_{ab}^{(t, t'; \tilde{G})} = 1$. To simplify the notation, let the function δ_κ indicate whether the packet delivery $\tilde{\nu} \in \{0, 1\}$ is consistent with the topology realization \tilde{G} , assuming the packet was generated at t_κ , i.e.,

$$\delta_\kappa(\tilde{\nu}; \tilde{G}) = \begin{cases} 1 & \text{if } \tilde{\nu} = \tilde{F}_{ab}^{(t_\kappa, t'_\kappa; \tilde{G})} \\ 0 & \text{otherwise.} \end{cases} \quad (6.3)$$

The function assumes the fixed repeating schedule $\mathbf{F}^{(T)}$, the packet generation time t_κ , the deadline t'_κ , the source a , and the destination b are implicitly known.

6.2.2 Network Estimators

As shown in Figure 6.2, at each sample time k the network estimator \hat{N} takes as input the previous packet delivery ν_{k-1} , estimates the topology realization using the network model and all past packet deliveries, and outputs the joint probability distribution of future packet deliveries $f_{\nu_k^{k+H-1}}$. For clarity in the following exposition, let $\mathcal{V}_\kappa \in \{0, 1\}$ be the value taken on by the packet delivery random variable

ν_κ at some past sample time κ . Let the vector $\mathbf{V}_0^{k-1} = [\mathcal{V}_0, \dots, \mathcal{V}_{k-1}]$ denote the history of packet deliveries at sample time k , the values taken on by the vector of random variables $\boldsymbol{\nu}_0^{k-1} = [\nu_0, \dots, \nu_{k-1}]$. Then,

$$f_{\boldsymbol{\nu}_k^{k+H-1}}(\tilde{\boldsymbol{\nu}}_0^{H-1}) = \mathbb{P}\text{r}(\boldsymbol{\nu}_k^{k+H-1} = \tilde{\boldsymbol{\nu}}_0^{H-1} | \boldsymbol{\nu}_0^{k-1} = \mathbf{V}_0^{k-1}) \quad (6.4)$$

is the prediction of the next H packet deliveries, where $\boldsymbol{\nu}_k^{k+H-1} = [\nu_k, \dots, \nu_{k+H-1}]$ is a vector of random variables representing future packet deliveries and $\tilde{\boldsymbol{\nu}}_0^{H-1} \in \{0, 1\}^H$.

The SIHS and GEIHS network estimators only differ in the network models. The parameters of the network models — topology G , schedule $\mathbf{F}^{(T)}$, link probabilities $\{p_l\}_{l \in \mathcal{E}}$ or $\{p_l^u, p_l^d\}_{l \in \mathcal{E}}$, source a , sink b , packet generation times t_k , and deadlines t'_k — are known a priori to the network estimators and are left out of the conditional probability expressions.

In Section 6.3, we will use the probability distribution on the topology realizations (our network state estimate),

$$\mathbb{P}\text{r}(G^{(k)} = \tilde{G} | \boldsymbol{\nu}_0^{k-1} = \mathbf{V}_0^{k-1}) \quad ,$$

to obtain $f_{\boldsymbol{\nu}_k^{k+H-1}}$ from \mathbf{V}_0^{k-1} and the network model.

6.2.3 FPD Controller

The FPD controller (\mathcal{C} in Figure 6.2) optimizes the control signals to the statistics of the future packet delivery sequence, derived from the past packet delivery sequence. We choose the optimal control framework because the cost function allows us to easily compare the FPD controller with other controllers. The control policy operates on the information set

$$\mathcal{I}_k^c = \{\mathbf{x}_0^k, \mathbf{u}_0^{k-1}, \boldsymbol{\nu}_0^{k-1}\} \quad . \quad (6.5)$$

The control policy minimizes the linear quadratic cost function

$$\mathbb{E} \left[x_N^T Q_0 x_N + \sum_{n=0}^{N-1} x_n^T Q_1 x_n + \nu_n u_n^T Q_2 u_n \right] \quad , \quad (6.6)$$

where Q_0 , Q_1 , and Q_2 are positive definite weighting matrices and N is the finite horizon, to get the minimum cost

$$J = \min_{u_0, \dots, u_{N-1}} \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{n=0}^{N-1} x_n^T Q_1 x_n + \nu_n u_n^T Q_2 u_n \right] \quad .$$

Section 6.4 will show that the resulting architecture separates into a network estimator and a controller which uses the pdf $f_{\boldsymbol{\nu}_k^{k+H-1}}$ supplied by the network estimator ($\hat{\mathbf{N}}$ in Figure 6.2) to find the control signals u_k .

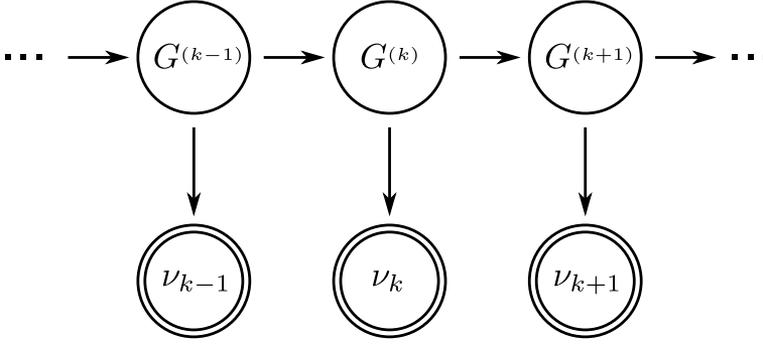


Figure 6.5: Graphical model describing the network estimation problem. ν_k is the measurement output variable at time k , and $G^{(k)}$ is the hidden state of the network.

6.3 Network Estimation and Packet Delivery Prediction

We will use recursive Bayesian estimation to estimate the state of the network, and use the network state estimate to predict future packet deliveries. Figure 6.5 is the graphical model / hidden Markov model (Smyth et al., 1997) describing our recursive estimation problem.

6.3.1 SIHS Network Estimator

The steps in the SIHS network estimator are derived from (6.4). We introduce new notation for conditional pdfs (i.e., α_k, β_k, Z_k), which will be used later to state the steps in the estimator compactly.⁶ First, express $f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})$ as

$$\underbrace{\mathbb{P}\mathbb{R}(\nu_k^{k+H-1} | \mathbf{V}_0^{k-1})}_{f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})} = \sum_{G^{(k-1)}} \underbrace{\mathbb{P}\mathbb{R}(\nu_k^{k+H-1} | G^{(k-1)})}_{\alpha_k(\tilde{\nu}_0^{H-1}; \tilde{G})} \cdot \underbrace{\mathbb{P}\mathbb{R}(G^{(k-1)} | \mathbf{V}_0^{k-1})}_{\beta_k(\tilde{G})}$$

where we use the relation

$$\mathbb{P}\mathbb{R}(\nu_k^{k+H-1} = \tilde{\nu}_0^{H-1} | G^{(k-1)} = \tilde{G}, \nu_0^{k-1} = \mathbf{V}_0^{k-1}) = \mathbb{P}\mathbb{R}(\nu_k^{k+H-1} = \tilde{\nu}_0^{H-1} | G^{(k-1)} = \tilde{G}) \quad .$$

This relation states that given the state of the network, future packet deliveries are independent of past packet deliveries. The expression $\mathbb{P}\mathbb{R}(\nu_k^{k+H-1} = \tilde{\nu}_0^{H-1} | G^{(k-1)} = \tilde{G})$ indicates whether the future packet delivery sequence $\tilde{\nu}_0^{H-1}$ is consistent with

⁶A semicolon is used in the conditional pdfs to separate the values being conditioned on from the remaining arguments.

the graph realization \tilde{G} , meaning

$$\Pr(\boldsymbol{\nu}_k^{k+H-1} = \tilde{\boldsymbol{\nu}}_0^{H-1} | G^{(k-1)} = \tilde{G}) = \prod_{h=0}^{H-1} \delta_{k+h}(\tilde{\nu}_h; \tilde{G}) \quad ,$$

where \prod is the *and* operator (sometimes denoted \wedge). The network state estimate at sample time k from past packet deliveries is $\beta_k(\tilde{G})$ and is obtained from the network state estimate at sample time $k-1$, since

$$\underbrace{\Pr(G^{(k-1)} | \mathbf{V}_0^{k-1})}_{\beta_k(\tilde{G})} = \frac{\overbrace{\Pr(\mathcal{V}_{k-1} | G^{(k-1)})}^{\delta_{k-1}(\mathcal{V}_{k-1}; \tilde{G})} \cdot \overbrace{\Pr(G^{(k-1)} | \mathbf{V}_0^{k-2})}^{\beta_{k-1}(\tilde{G})}}{\underbrace{\Pr(\mathcal{V}_{k-1} | \mathbf{V}_0^{k-2})}_{(Z_k)}} \quad . \quad (6.7)$$

Here, $\Pr(G^{(k-1)} = \tilde{G} | \boldsymbol{\nu}_0^{k-2} = \mathbf{V}_0^{k-2}) = \Pr(G^{(k-2)} = \tilde{G} | \boldsymbol{\nu}_0^{k-2} = \mathbf{V}_0^{k-2}) = \beta_{k-1}(\tilde{G})$ for the static link model because $G^{(k-1)} = G^{(k-2)} = G^{(0)}$. Again, we used the independence of future packet deliveries from past packet deliveries given the network state,

$$\Pr(\nu_{k-1} = \mathcal{V}_{k-1} | G^{(k-1)} = \tilde{G}, \boldsymbol{\nu}_0^{k-2} = \mathbf{V}_0^{k-2}) = \Pr(\nu_{k-1} = \mathcal{V}_{k-1} | G^{(k-1)} = \tilde{G}) \quad .$$

Note that $\Pr(\nu_{k-1} = \mathcal{V}_{k-1} | G^{(k-1)} = \tilde{G})$ can only be 0 or 1, indicating whether the packet delivery is consistent with the graph realization. Finally, $\Pr(\nu_{k-1} = \mathcal{V}_{k-1} | \boldsymbol{\nu}_0^{k-2} = \mathbf{V}_0^{k-2})$ is the same for all \tilde{G} , so it is treated as a normalization constant.

At sample time $k=0$, when no packets have been sent through the network, $\beta_0(\tilde{G}) = \Pr(G^{(0)} = \tilde{G})$, which is expressed in (6.8d) below. This equation comes from the assumption that all links in the network are independent.

To summarize, the *SIHS Network Estimator and Packet Delivery Predictor* is a recursive Bayesian estimator where the measurement output step consists of

$$f_{\boldsymbol{\nu}_k^{k+H-1}}(\tilde{\boldsymbol{\nu}}_0^{H-1}) = \sum_{\tilde{G}} \alpha_k(\tilde{\boldsymbol{\nu}}_0^{H-1}; \tilde{G}) \cdot \beta_k(\tilde{G}) \quad (6.8a)$$

$$\alpha_k(\tilde{\boldsymbol{\nu}}_0^{H-1}; \tilde{G}) = \prod_{h=0}^{H-1} \delta_{k+h}(\tilde{\nu}_h; \tilde{G}) \quad , \quad (6.8b)$$

and the innovation step consists of

$$\beta_k(\tilde{G}) = \frac{\delta_{k-1}(\mathcal{V}_{k-1}; \tilde{G}) \cdot \beta_{k-1}(\tilde{G})}{Z_k} \quad (6.8c)$$

$$\beta_0(\tilde{G}) = \left(\prod_{l \in \tilde{\mathcal{E}}} p_l \right) \left(\prod_{l \in \mathcal{E} \setminus \tilde{\mathcal{E}}} 1 - p_l \right) \quad , \quad (6.8d)$$

where α_k and β_k are functions, Z_k is a normalization constant such that $\sum_{\tilde{G}} \beta_k(\tilde{G}) = 1$, and the functions δ_{k+h} and δ_{k-1} are defined by (6.3).

6.3.2 GEIHS Network Estimator

For compact notation in the probability expressions below, we use \mathbf{V}_0^{k-1} in place of $\nu_0^{k-1} = \mathbf{V}_0^{k-1}$ and only write the random variable and not its value (\cdot).

The derivation of the GEIHS network estimator is similar to the previous derivation, except that the state of the network evolves with every sample time k . Since all the links in the network are independent, the probability that a given topology \tilde{G}' at sample time $k-1$ transitions to a topology \tilde{G} after one sample time is given by

$$\begin{aligned} \Gamma(\tilde{G}; \tilde{G}') &= \mathbb{P}\text{r}(G^{(k)} | G^{(k-1)}) = \\ &= \left(\prod_{l_1 \in \tilde{\mathcal{E}}' \cap \tilde{\mathcal{E}}} 1 - p_{l_1}^d \right) \left(\prod_{l_2 \in \tilde{\mathcal{E}}' \setminus \tilde{\mathcal{E}}} p_{l_2}^d \right) \\ &\quad \times \left(\prod_{l_3 \in \tilde{\mathcal{E}} \setminus \tilde{\mathcal{E}}'} p_{l_3}^u \right) \left(\prod_{l_4 \in \mathcal{E} \setminus (\tilde{\mathcal{E}}' \cup \tilde{\mathcal{E}})} 1 - p_{l_4}^u \right) . \end{aligned} \quad (6.9)$$

First, express $f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})$ as

$$\underbrace{\mathbb{P}\text{r}(\nu_k^{k+H-1} | \mathbf{V}_0^{k-1})}_{f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})} = \sum_{G^{(k+H-1)}} \underbrace{\mathbb{P}\text{r}(\nu_{k+H-1} | G^{(k+H-1)})}_{\delta_{k+H-1}(\tilde{\nu}_{H-1}; \tilde{G}_{H-1})} \cdot \underbrace{\mathbb{P}\text{r}(\nu_k^{k+H-2}, G^{(k+H-1)} | \mathbf{V}_0^{k-1})}_{\alpha_{H-1|k-1}(\tilde{\nu}_0^{H-2}, \tilde{G}_{H-1})} ,$$

where for $h = 2, \dots, H-1$

$$\begin{aligned} \underbrace{\alpha_{h|k-1}(\tilde{\nu}_0^{h-1}, \tilde{G}_h)}_{\mathbb{P}\text{r}(\nu_k^{k+h-1}, G^{(k+h)} | \mathbf{V}_0^{k-1})} &= \\ &= \sum_{G^{(k+h-1)}} \left(\underbrace{\mathbb{P}\text{r}(G^{(k+h)} | G^{(k+h-1)})}_{\Gamma(\tilde{G}_h; \tilde{G}_{h-1})} \cdot \underbrace{\mathbb{P}\text{r}(\nu_{k+h-1} | G^{(k+h-1)})}_{\delta_{k+h-1}(\tilde{\nu}_{h-1}; \tilde{G}_{h-1})} \right. \\ &\quad \left. \times \underbrace{\mathbb{P}\text{r}(\nu_k^{k+h-2}, G^{(k+h-1)} | \mathbf{V}_0^{k-1})}_{\alpha_{h-1|k-1}(\tilde{\nu}_0^{h-2}, \tilde{G}_{h-1})} \right) . \end{aligned} \quad (6.10)$$

When $h = 1$, replace $\mathbb{P}\text{r}(\nu_k^{k+h-2}, G^{(k+h-1)} | \mathbf{V}_0^{k-1})$ in (6.10) with $\mathbb{P}\text{r}(G^{(k)} | \mathbf{V}_0^{k-1}) = \beta_{k|k-1}(\tilde{G})$. The value $\beta_{k|k-1}(\tilde{G})$ comes from

$$\underbrace{\mathbb{P}\text{r}(G^{(k)} | \mathbf{V}_0^{k-1})}_{\beta_{k|k-1}(\tilde{G})} = \sum_{G^{(k-1)}} \underbrace{\mathbb{P}\text{r}(G^{(k)} | G^{(k-1)})}_{\Gamma(\tilde{G}; \tilde{G}')} \cdot \underbrace{\mathbb{P}\text{r}(G^{(k-1)} | \mathbf{V}_0^{k-1})}_{\beta_{k-1|k-1}(\tilde{G}')} .$$

The value $\beta_{k-1|k-1}(\tilde{G}')$ comes from (6.7), with β_k replaced by $\beta_{k-1|k-1}$ and β_{k-1} replaced by $\beta_{k-1|k-2}$. Finally, $\beta_{0|-1}(\tilde{G}) = \mathbb{P}\text{r}(G^{(0)})$, where all links are independent

and have link probabilities equal to their steady-state probability of being in state 1, and is expressed in (6.11f) below.

To summarize, the *GEIHS Network Estimator and Packet Delivery Predictor* is a recursive Bayesian estimator. The measurement output step consists of

$$f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1}) = \sum_{\tilde{G}_{H-1}} \delta_{k+H-1}(\tilde{\nu}_{H-1}; \tilde{G}_{H-1}) \cdot \alpha_{H-1|k-1}(\tilde{\nu}_0^{H-2}, \tilde{G}_{H-1}) \quad , \quad (6.11a)$$

where the function $\alpha_{H-1|k-1}$ is obtained from the following recursive equation for $h = 2, \dots, H - 1$:

$$\alpha_{h|k-1}(\tilde{\nu}_0^{h-1}, \tilde{G}_h) = \sum_{\tilde{G}_{h-1}} \Gamma(\tilde{G}_h; \tilde{G}_{h-1}) \cdot \delta_{k+h-1}(\tilde{\nu}_{h-1}; \tilde{G}_{h-1}) \cdot \alpha_{h-1|k-1}(\tilde{\nu}_0^{h-2}, \tilde{G}_{h-1}) \quad , \quad (6.11b)$$

with initial condition

$$\alpha_{1|k-1}(\tilde{\nu}_0^0, \tilde{G}_1) = \sum_{\tilde{G}} \Gamma(\tilde{G}_1; \tilde{G}) \cdot \delta_k(\tilde{\nu}_0; \tilde{G}) \cdot \beta_{k|k-1}(\tilde{G}) \quad . \quad (6.11c)$$

The prediction and innovation steps consist of

$$\beta_{k|k-1}(\tilde{G}) = \sum_{\tilde{G}'} \Gamma(\tilde{G}; \tilde{G}') \cdot \beta_{k-1|k-1}(\tilde{G}') \quad (6.11d)$$

$$\beta_{k-1|k-1}(\tilde{G}) = \frac{\delta_{k-1}(\mathcal{V}_{k-1}; \tilde{G}) \cdot \beta_{k-1|k-2}(\tilde{G})}{Z_{k-1}} \quad (6.11e)$$

$$\beta_{0|k-1}(\tilde{G}) = \left(\prod_{l \in \tilde{\mathcal{E}}} p_l \right) \left(\prod_{l \in \mathcal{E} \setminus \tilde{\mathcal{E}}} 1 - p_l \right) \quad , \quad (6.11f)$$

where $\alpha_{h|k-1}$, $\beta_{k-1|k-1}$, and $\beta_{k|k-1}$ are functions, Z_{k-1} is a normalization constant such that $\sum_{\tilde{G}} \beta_{k-1|k-1}(\tilde{G}) = 1$, and the functions δ_κ (for the different values of κ above) and Γ are defined by (6.3) and (6.9), respectively.

6.3.3 Packet Predictor Complexity

The network estimators are trying to estimate network parameters using measurements collected at the border of the network, a general problem studied in the field of network tomography (Castro et al., 2004) under various problem setups. One of the greatest challenges in network tomography is getting good estimates with low computational complexity estimators.

Our proposed network estimators are “optimal” with respect to our models in the sense that there is no loss of information, but they are computationally expensive.

Property 6.1. *The SIHS network estimator described by the set of equations (6.8) takes $O(E2^E)$ to initialize and $O(2^H 2^E)$ to update the network state estimate and predictions at each step. The GEIHS network estimator described by the set of equations (6.11) takes $O(E2^{2E})$ to initialize and $O(2^H 2^{2E})$ for each update step.*

Proof. Let $D = \max_k t'_k - t_k$. We assume that converting \tilde{G} to the set of links that are up, $\tilde{\mathcal{E}}$, takes constant time. Also, one can simulate the path of a packet by looking up the scheduled and successful link transmissions instead of multiplying matrices to evaluate $\tilde{F}_{ab}^{(t_\kappa, t'_\kappa; \tilde{G})}$, so computing δ_κ for each graph \tilde{G} only takes $O(D)$. The computational complexities below assume that the pdfs can be represented by matrices, and multiplying an $\ell \times m$ matrix with a $m \times n$ matrix takes $O(\ell mn)$.

SIHS packet delivery predictor complexity:

Computing α_k in (6.8b) takes $O(D2^H 2^E)$ and computing β_k in (6.8c) takes $O(D2^E)$, since there are 2^E graphs and 2^H packet delivery prediction sequences. Computing $f_{\nu_k^{k+H-1}}$ in (6.8a) takes $O(D2^H 2^E)$. The SIHS packet delivery predictor update step is the aggregate of all these computations and takes $O(D2^H 2^E)$.

The initialization step of the SIHS packet delivery predictor is just computing β_0 in (6.8d), which takes $O(E2^E)$.

GEIHS packet delivery predictor complexity:

Computing $\alpha_{h|k-1}$ in (6.11b) takes $O(D2^E + 2^E 2^h + 2^{2E} 2^h)$ and computing $\alpha_{1|k-1}$ in (6.11c) takes $O(D2^E + 2^{E+1} + 2^{2E+1})$, so computing all of them takes $O(DH2^E + 2^H(2^E + 2^{2E}))$, or just $O(DH2^E + 2^H 2^{2E})$. Computing $\beta_{k|k-1}$ in (6.11d) takes $O(2^{2E})$, and computing $\beta_{k-1|k-1}$ in (6.11e) takes $O(D2^E)$. Computing $f_{\nu_k^{k+H-1}}$ in (6.11a) takes $O(D2^E + 2^H 2^E)$. The GEIHS packet delivery predictor update step is the aggregate of all these computations and takes $O(DH2^E + 2^H 2^{2E})$.

Computing Γ in (6.9) takes $O(E2^{2E})$, and computing $\beta_{0|-1}$ in (6.11f) takes $O(E2^E)$. The initialization step of the GEIHS packet delivery predictor is the aggregate of these computations and takes $O(E2^{2E})$.

If we assume that the deadline D is short enough to be considered constant, we get the computational complexities given in Property 6.1. \square

A good direction for future research is to find lower complexity, “suboptimal” network estimators for our problem setup, and compare them to our “optimal” network estimators.

6.3.4 Discussion

Our network estimators can easily be extended to incorporate additional observations besides past packet deliveries, such as the packet delay and packet path traces. The latter can be obtained by recording the state of the links that the packet has tried to traverse in the packet payload. The function δ_{k-1} in (6.8c) and (6.11e) just needs to be replaced with another function that returns 1 if the the received observation is consistent with a network topology \tilde{G} , and 0 otherwise. The advantage of

using more observations than the one bit of information provided by a packet delivery is that it will help the GEIHS network estimator more quickly detect changes in the network state. A more non-trivial extension of the GEIHS network estimator would use additional observations provided by packets from other flows (not from our controller) to help estimate the network state, which could significantly decrease the time for the network estimator to detect a change in the state of the network. This is non-trivial because the network model would now have to account for queuing at nodes in the network, which is inevitable with multiple flows.

Note that the network state probability distribution, given by $\beta_k(\tilde{G})$ in (6.8c) or $\beta_{k-1|k-1}(\tilde{G})$ in (6.11e), does not need to converge to a probability distribution describing one topology realization to yield precise packet predictions $f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})$, where precise means there is one (or very few similar) high probability packet delivery sequence(s) $\tilde{\nu}_0^{H-1}$. Several topology realizations \tilde{G} may result in the same packet delivery sequence.

Also, note that the GEIHS network estimator performs better when the links in the network are more bursty. Long bursts of packet losses from bursty links result in poor control system performance, which is when the network estimator would help the most.

6.4 FPD Controller

In this section, we derive the FPD controller using dynamic programming. Next, we present two controllers for comparison with the FPD controller. These comparative controllers assume particular statistical models (e.g., i.i.d. Bernoulli) for the packet delivery sequence pdf which may not describe the actual pdf, while the FPD controller allows for all packet delivery sequence pdfs. We derive the LQG cost of using these controllers. Finally, we present the computational complexity of the optimal controller.

6.4.1 Derivation of the FPD Controller

We first present the FPD controller and then present its derivation.

Theorem 6.1. *For a plant with state dynamics given by (6.1), the optimal control policy operating on the information set (6.5) which minimizes the cost function (6.6) results in an optimal control signal $u_k = -L_k x_k$, where*

$$L_k = (Q_2 + B^T S_{k+1}(\nu_k=1, \nu_0^{k-1})B)^{-1} B^T S_{k+1}(\nu_k=1, \nu_0^{k-1})A \quad (6.12)$$

and $S_k : \{0, 1\}^k \mapsto \mathbb{S}_+^\ell$ and $s_k : \{0, 1\}^k \mapsto \mathbb{R}_+$ are the solutions to the cost-to-go at

time k , given by

$$\begin{aligned}
S_k(\nu_0^{k-1}) &= Q_1 + A^T \mathbb{E} [S_{k+1}(\nu_0^k) | \nu_0^{k-1}] A \\
&\quad - \mathbb{Pr}(\nu_k = 1 | \nu_0^{k-1}) \left[A^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) B \right. \\
&\quad \left. \times (Q_2 + B^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) B)^{-1} B^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) A \right] \\
s_k(\nu_0^{k-1}) &= \text{tr} \{ \mathbb{E} [S_{k+1}(\nu_0^k) | \nu_0^{k-1}] R_w \} + \mathbb{E} [s_{k+1}(\nu_0^k) | \nu_0^{k-1}] \quad .
\end{aligned}$$

Proof. The classical problem in Åström (1970) is solved by reformulating the original problem as a recursive minimization of the Bellman equation derived for every time instant, beginning with N . At time n , we have the minimization problem

$$\begin{aligned}
&\min_{u_n, \dots, u_{N-1}} \mathbb{E} [x_N^T Q_0 x_N + \sum_{i=n}^{N-1} x_i^T Q_1 x_i + \nu_i u_i^T Q_2 u_i] \\
&= \mathbb{E} \left[\min_{u_n, \dots, u_{N-1}} \mathbb{E} [x_N^T Q_0 x_N + \sum_{i=n}^{N-1} x_i^T Q_1 x_i + \nu_i u_i^T Q_2 u_i | \mathcal{I}_n^c] \right] \\
&= \mathbb{E} \left[\min_{u_n} \mathbb{E} [x_n^T Q_1 x_n + \nu_n u_n^T Q_2 u_n + V_{n+1} | \mathcal{I}_n^c] \right] \quad ,
\end{aligned}$$

where V_n is the Bellman equation at time n . This is given by

$$V_n = \min_{u_n} \mathbb{E} \left[x_n^T Q_1 x_n + \nu_n u_n^T Q_2 u_n + V_{n+1} | \mathcal{I}_n^c \right] \quad .$$

To solve the above nested minimization problem, we assume that the solution to the functional is of the form $V_n = x_n^T S_n(\nu_0^{n-1}) x_n + s_n(\nu_0^{n-1})$, where S_n and s_n are functions of the past packet deliveries ν_0^{n-1} that return a positive semidefinite matrix and a scalar, respectively. However, both S_n and s_n are not functions of the applied control sequence \mathbf{u}_0^{n-1} . We prove this supposition using induction. The initial condition at time N is trivially obtained as $V_N = x_N^T Q_0 x_N$, with $S_N = Q_0$ and $s_N = 0$. We now assume that the functional at time $n+1$ has a solution of the desired form, and attempt to derive this at time n . We have

$$\begin{aligned}
V_n &= \min_{u_n} \mathbb{E} \left[x_n^T Q_1 x_n + \nu_n u_n^T Q_2 u_n + x_{n+1}^T S_{n+1}(\nu_0^n) x_{n+1} + s_{n+1}(\nu_0^n) | \mathcal{I}_n^c \right] \\
&= \min_{u_n} \mathbb{E} \left[x_n^T (Q_1 + A^T S_{n+1}(\nu_0^n) A) x_n + \nu_n u_n^T (Q_2 + B^T S_{n+1}(\nu_0^n) B) u_n \right. \\
&\quad \left. + \nu_n x_n^T A^T S_{n+1}(\nu_0^n) B u_n + \nu_n u_n^T B^T S_{n+1}(\nu_0^n) A x_n \right. \\
&\quad \left. + w_n^T S_{n+1}(\nu_0^n) w_n + s_{n+1}(\nu_0^n) | \mathcal{I}_n^c \right]
\end{aligned}$$

This can be further simplified to get

$$\begin{aligned}
V_n = \min_{u_n} & x_n^T \left(Q_1 + A^T \mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{n-1}] A \right) x_n \\
& + \text{tr} \left\{ \mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{n-1}] R_w \right\} + \mathbb{E} [s_{n+1}(\nu_0^n) | \nu_0^{n-1}] \\
& + \mathbb{Pr}(\nu_n = 1 | \nu_0^{n-1}) \left[u_n^T (Q_2 + B^T S_{n+1}(\nu_n=1, \nu_0^{n-1}) B) u_n \right. \\
& \left. + x_n^T A^T S_{n+1}(\nu_n=1, \nu_0^{n-1}) B u_n + u_n^T B^T S_{n+1}(\nu_n=1, \nu_0^{n-1}) A x_n \right] . \quad (6.13)
\end{aligned}$$

In the last equation above, the expectation of the terms preceded by ν_n require the conditional probability $\mathbb{Pr}(\nu_n = 1 | \nu_0^{n-1})$ and an evaluation of S_{n+1} with $\nu_n = 1$. The corresponding terms with $\nu_n = 0$ vanish as they are multiplied by ν_n . The control input at sample time n which minimizes the above expression is found to be $u_n = -L_n x_n$, where the optimal control gain L_n is given by (6.12), with k replaced by n . Substituting for u_n in the functional V_n , we get a solution to the functional of the desired form, with S_n and s_n given by

$$\begin{aligned}
S_n(\nu_0^{n-1}) &= Q_1 + A^T \mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{n-1}] A \\
&\quad - \mathbb{Pr}(\nu_n = 1 | \nu_0^{n-1}) \left[A^T S_{n+1}(\nu_n=1, \nu_0^{n-1}) B \right. \\
&\quad \left. \times (Q_2 + B^T S_{n+1}(\nu_n=1, \nu_0^{n-1}) B)^{-1} B^T S_{n+1}(\nu_n=1, \nu_0^{n-1}) A \right] \quad (6.14a)
\end{aligned}$$

$$s_n(\nu_0^{n-1}) = \text{tr} \left\{ \mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{n-1}] R_w \right\} + \mathbb{E} [s_{n+1}(\nu_0^n) | \nu_0^{n-1}] . \quad (6.14b)$$

Notice that S_n and s_n are functions of the variables ν_0^{n-1} . When $n = k$, the current sample time, these variables are known, and S_n and s_n are not random. But S_{n+i} and s_{n+i} , for values of $i \in \{1, \dots, N - n - 1\}$, are functions of the variables ν_0^{n+i-1} , of which only the variables ν_n^{n+i-1} are random variables since they are unknown to the controller at sample time $n = k$. Since the value of S_{n+1} is required at sample time n , we compute its conditional expectation as

$$\begin{aligned}
\mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{n-1}] &= \mathbb{Pr}(\nu_n = 1 | \nu_0^{n-1}) S_{n+1}(\nu_n=1, \nu_0^{n-1}) \\
&\quad + \mathbb{Pr}(\nu_n = 0 | \nu_0^{n-1}) S_{n+1}(\nu_n=0, \nu_0^{n-1}) . \quad (6.14c)
\end{aligned}$$

The above computation requires an evaluation of $S_{n+i}(\nu_0^{n+i-1})$ through a backward recursion for $i \in \{1, \dots, N - n - 1\}$ for all combinations of ν_{n+i}^{N-2} . More explicitly,

the expression at any time $n + i$, for $i \in \{N - n - 1, \dots, 1\}$, is given by

$$\begin{aligned}
\mathbb{E} [S_{n+i}(\nu_0^{n+i-1})|\nu_0^{n-1}] &= Q_1 + A^T \mathbb{E} [S_{n+i+1}(\nu_0^{n+i})|\nu_0^{n-1}] A \\
&\quad - \sum_{\tilde{\nu}_0^{i-1} \in \{0,1\}^i} \Pr(\nu_{n+i} = 1, \nu_n^{n+i-1} = \tilde{\nu}_0^{i-1} | \nu_0^{n-1}) \\
&\quad \times A^T S_{n+i+1}(\nu_{n+i}=1, \nu_n^{n+i-1}=\tilde{\nu}_0^{i-1}, \nu_0^{n-1}) B \\
&\quad \times (Q_2 + B^T S_{n+i+1}(\nu_{n+i}=1, \nu_n^{n+i-1}=\tilde{\nu}_0^{i-1}, \nu_0^{n-1}) B)^{-1} \\
&\quad \times B^T S_{n+i+1}(\nu_{n+i}=1, \nu_n^{n+i-1}=\tilde{\nu}_0^{i-1}, \nu_0^{n-1}) A \\
\mathbb{E} [s_{n+i}(\nu_0^{n+i-1})|\nu_0^{n-1}] &= \text{tr} \{ \mathbb{E} [S_{n+i+1}(\nu_0^{n+i})|\nu_0^{n-1}] R_w \} \\
&\quad + \mathbb{E} [s_{n+i+1}(\nu_0^{n+i})|\nu_0^{n-1}] \quad .
\end{aligned}$$

Using the above expressions, we obtain the net cost to be

$$J = \text{tr} S_0 R_0 + \sum_{n=0}^{N-1} \text{tr} \{ \mathbb{E} [S_{n+1}(\nu_0^n)] R_w \} \quad . \quad (6.15)$$

Notice that the control inputs u_n are only applied to the plant and do not influence the network or ν_0^{N-1} . Thus, the architecture separates into a network estimator and controller, as shown in Figure 6.2. \square

6.4.2 Comparative controllers

In this section, we compare the performance of the FPD controller to two controllers that assume particular statistical models for the packet delivery sequence pdf, the IID controller and the ON controller.

IID Controller: The IID controller was described in Schenato et al. (Schenato et al., 2007) and assumes that the packet deliveries are i.i.d. Bernoulli with packet delivery probability equal to the a priori probability of delivering a packet through the network.⁷ This is our first comparative controller, where $u_k = -L_k^{\text{IID}} x_k$ and the control gain L_k^{IID} is given by

$$L_k^{\text{IID}} = (Q_2 + B^T S_{k+1}^{\text{IID}} B)^{-1} B^T S_{k+1}^{\text{IID}} A \quad .$$

Here, S_{k+1}^{IID} is the solution to the Riccati equation for the control problem where the packet deliveries are assumed to be i.i.d. Bernoulli. The backward recursion is initialized to $S_N^{\text{IID}} = Q_0$ and is given by

$$S_k^{\text{IID}} = Q_1 + A^T S_{k+1}^{\text{IID}} A - \Pr(\nu_k = 1) A^T S_{k+1}^{\text{IID}} B (Q_2 + B^T S_{k+1}^{\text{IID}} B)^{-1} B^T S_{k+1}^{\text{IID}} A \quad .$$

⁷Using the stationary probability of each link under the G-E link model to calculate the end-to-end probability of delivering a packet through the network.

ON Controller: The ON controller assumes that the packets are always delivered, or that the network is always online. This is our second comparative controller, where $u_k = -L_k^{\text{ON}}x_k$ and the control gain L_k^{ON} is given by

$$L_k^{\text{ON}} = (Q_2 + B^T S_{k+1}^{\text{ON}} B)^{-1} B^T S_{k+1}^{\text{ON}} A \quad .$$

Here, S_{k+1}^{ON} is the solution to the Riccati equation for the classical control problem which assumes no packet losses on the actuation channel. The backward recursion is initialized to $S_N^{\text{ON}} = Q_0$ and is given by

$$S_k^{\text{ON}} = Q_1 + A^T S_{k+1}^{\text{ON}} A - A^T S_{k+1}^{\text{ON}} B (Q_2 + B^T S_{k+1}^{\text{ON}} B)^{-1} B^T S_{k+1}^{\text{ON}} A \quad .$$

Comparative Cost: The FPD controller is the most general form of a causal, optimal LQG controller that takes into account the packet delivery sequence pdf. It does not assume the packet delivery sequence pdf comes from a particular statistical model. Approximating the actual packet delivery sequence pdf with a pdf described by a particular statistical model, and then computing the optimal control policy, will result in a suboptimal controller. However, it may be less computationally expensive to obtain the control gains for such a suboptimal controller. For example, the IID controller and the ON controller are suboptimal controllers for networks like the one described in Section 6.2.1, since they presume a statistical model that is mismatched to the packet delivery sequence pdf obtained from the network model.

Remark The average LQG cost of using a controller with control gain L_n^{comp} is

$$J = \text{tr} S_0^{\text{sopt}} R_0 + \sum_{n=0}^{N-1} \text{tr} \left\{ \mathbb{E}[S_{n+1}^{\text{sopt}}(\nu_0^n)] R_w \right\} \quad , \quad (6.16a)$$

where

$$\begin{aligned} S_n^{\text{sopt}}(\nu_0^{n-1}) &= Q_1 + A^T \mathbb{E} [S_{n+1}^{\text{sopt}}(\nu_0^n) | \nu_0^{n-1}] A + \mathbb{P}\text{r}(\nu_n = 1 | \nu_0^{n-1}) \\ &\times \left[L_n^{\text{comp}T} (Q_2 + B^T S_{n+1}^{\text{sopt}}(\nu_n=1, \nu_0^{n-1}) B) L_n^{\text{comp}} \right. \\ &\left. - A^T S_{n+1}^{\text{sopt}}(\nu_n=1, \nu_0^{n-1}) B L_n^{\text{comp}} - L_n^{\text{comp}T} B^T S_{n+1}^{\text{sopt}}(\nu_n=1, \nu_0^{n-1}) A \right] \quad , \end{aligned} \quad (6.16b)$$

and $\mathbb{E} [S_{n+1}^{\text{sopt}}(\nu_0^n) | \nu_0^{n-1}]$ is computed in a similar manner to (6.14c). The control gain L_n^{comp} can be the gain of a comparative controller (e.g., L_n^{IID} or L_n^{ON}) where the statistical model for the packet delivery sequence is mismatched to the actual model. \square

This can be seen from the proof of Theorem 6.1, if we substitute for the control input with $u_n^{\text{sopt}} = -L_n^{\text{comp}}x_n$ in (6.13), instead of minimizing the expression to

find the optimal u_n . On simplifying, we get the solution to the cost-to-go V_n of the form $x_n^T S_n^{\text{sopt}}(\nu_0^{n-1})x_n + s_n^{\text{sopt}}(\nu_0^{n-1})$, with S_n^{sopt} given by (6.16b) and s_n^{sopt} given by

$$s_n^{\text{sopt}}(\nu_0^{n-1}) = \text{tr} \left\{ \mathbb{E} \left[S_{n+1}^{\text{sopt}}(\nu_0^n) | \nu_0^{n-1} \right] R_w \right\} + \mathbb{E} \left[s_{n+1}^{\text{sopt}}(\nu_0^n) | \nu_0^{n-1} \right] \quad .$$

6.4.3 Algorithm to Compute Optimal Control Gain

At sample time k , we have ν_0^{k-1} . To compute L_k given in (6.12), we need $S_{k+1}(\nu_k=1, \nu_0^{k-1})$, which can only be obtained through a backward recursion from S_N . This requires knowledge of ν_k^{N-1} , which are unavailable at sample time k . Thus, we must evaluate $\{S_{k+1}, \dots, S_N\}$ for every possible sequence of arrivals ν_k^{N-1} . This algorithm is described below.

1. Initialization: $S_N(\nu_k^{N-1}=\tilde{\nu}_0^{N-k-1}, \nu_0^{k-1}) = Q_0, \forall \tilde{\nu}_0^{N-k-1} \in \{0, 1\}^{N-k}$.
2. for $n = N - 1 : -1 : k + 1$
 - a) Using (6.14c), compute $\mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{k-1}, \tilde{\nu}_0^{n-k-1}], \forall \tilde{\nu}_0^{n-k-1} \in \{0, 1\}^{n-k}$.
 - b) Using (6.14a), compute $S_n(\nu_k^{n-1}=\tilde{\nu}_0^{n-k-1}, \nu_0^{k-1}), \forall \tilde{\nu}_0^{n-k-1} \in \{0, 1\}^{n-k}$.
3. Compute L_k using $S_{k+1}(\nu_k=1, \nu_0^{k-1})$.

For $k = 0$, the values $S_0, \mathbb{E}[S_1(\nu_0)]$, and the other values obtained above can be used to evaluate the cost function according to (6.15).

6.4.4 Computational Complexity of Optimal Control Gain

The FPD controller is optimal but computationally expensive, as it requires an enumeration of all possible packet delivery sequences from the current sample time until the end of the control horizon to calculate the optimal control gain (6.12) at every sample time k .

Property 6.2. *The algorithm presented in Section 6.4.3 for computing the optimal control gain for the FPD controller takes $O(q^3(N-k)2^{N-k})$ operations at each sample time k , where $q = \max(\ell, m)$ and ℓ and m are the dimensions of the state and control vectors.*

Proof. The computational complexities below assume that multiplying an $\ell \times m$ matrix with a $m \times n$ matrix takes $O(\ell mn)$, and that inverting an $\ell \times \ell$ matrix takes $O(\ell^3)$.

For the computation of L_k in (6.12), we need to run the algorithm presented in Section 6.4.3. The steps within the for-loop (Step 2) of the algorithm require matrix multiplications and inversions that take $O((2\ell^3 + 6\ell^2 m + 2\ell^2 + 2\ell m^2 + m^3 + m^2)2^{N-k})$ operations, or $O(q^3 2^{N-k})$ operations if we let $q = \max(\ell, m)$. This must be repeated $N - k - 1$ times in the for-loop, so Step 2 takes $O(q^3(N-k-1)2^{N-k})$.

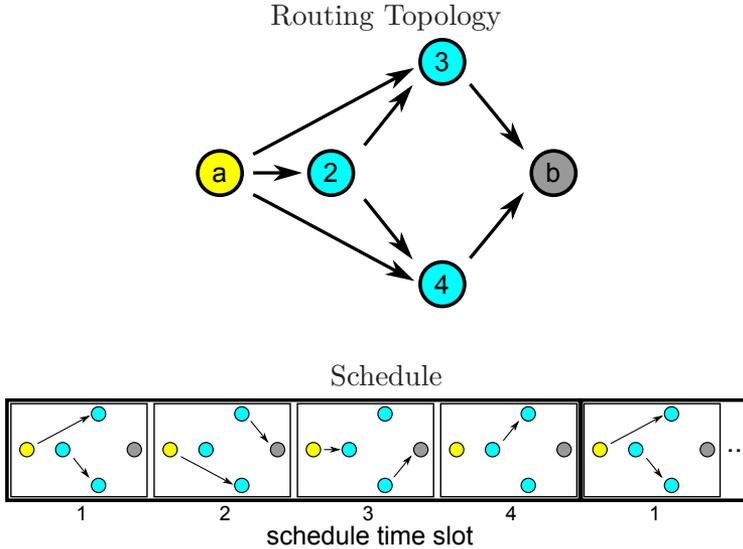


Figure 6.6: Example of a simple mesh network for network estimation.

Finally, Step 3 takes $O(4l^2m + lm^2 + m^3 + m^2)$ operations for the matrix multiplications and inversions, which simplifies to $O(q^3)$. Combining these results and simplifying yields the computational complexity given in Property 6.2. \square

For the SIHS network model, once the network state estimates from the SIHS network estimator converge, the conditional probabilities $f_{\nu_k^{k+H-1}}$ will not change and the computations can be reused. But, for a network that evolves over time, like the GEIHS network model, the computations cannot be reused, and the computational cost remains high.

6.5 Examples and Simulations

Using the system architecture depicted in Figure 6.2, we will demonstrate the GEIHS network estimator on a small mesh network and use the packet delivery predictions in our FPD controller. Figure 6.6 depicts the routing topology and short repeating schedule of the network. Packets are generated at the source every 409 time slots,⁸ and the packet delivery deadline is $t'_k - t_k = 9, \forall k$. The network estimator assumes all links have $p^u = 0.0135$ and $p^d = 0.0015$.

The packet delivery predictions from the network estimator are shown in Figure 6.7. Although the network estimator provides $f_{\nu_k^{k+H-1}}(\tilde{\mathcal{V}}_0^{H-1})$, at each sample

⁸Effectively, the packets are generated every $9 + 4K$ time slots, where K is a very large integer, so we can assume slow system dynamics with respect to time slots and ignore the delay introduced by the network.

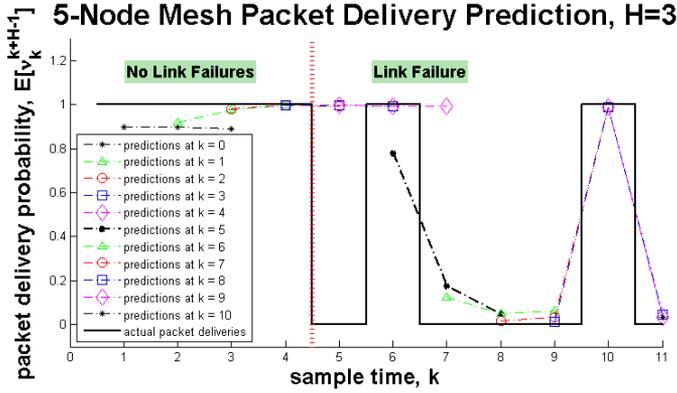


Figure 6.7: Packet delivery predictions when network in Figure 6.6 has all links up and then link $(3, b)$ fails.

time k we plot the average prediction $\mathbb{E}[\nu_k^{k+H-1}]$. In this example, all the links are up for $k \in \{1, \dots, 4\}$ and then link $(3, b)$ fails from $k = 5$ onwards. After seeing a packet loss at $k = 5$, the network estimator revises its packet delivery predictions and now thinks there will be a packet loss at $k = 7$. The average prediction for the packet delivery at a particular sample time tends toward 1 or 0 as the network estimator receives more information (in the form of packet deliveries) about the new state of the network.

The prediction for $k = 7$ (packet generated at schedule time slot 3) at $k = 5$ is influenced by the packet delivery at $k = 5$ (packet generated at schedule time slot 1) because hop-by-hop routing allows the packets to traverse the same links under some realizations of the underlying routing topology G . Mesh networks with many interleaved paths allow packets generated at different schedule time slots to provide information about each others' deliveries, provided the links in the network have some memory. As discussed in Section 6.3.4, since a packet delivery provides only one bit of information about the network state, it may take several packet deliveries to get good predictions after the network changes.

Now, consider a linear plant with the following parameters

$$A = \begin{bmatrix} 0 & 1.5 \\ 1.5 & 0 \end{bmatrix}, B = \begin{bmatrix} 5 & 0 \\ 0 & 0.2 \end{bmatrix}, R_w = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, R_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

$$Q_1 = Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q_n = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} .$$

The transfer matrix A flips and expands the components of the state at every sampling instant. The input matrix B requires the second component of the control input to be larger in magnitude than the first component to have the same effect on the respective component of the state. Also, the final state is weighted more than

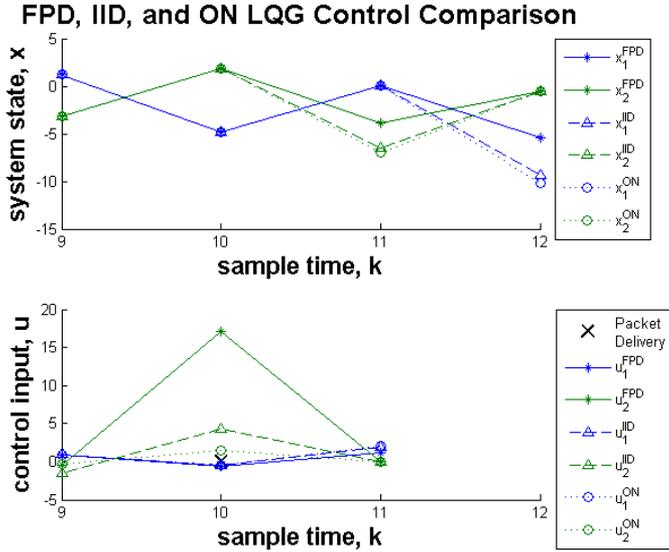


Figure 6.8: Plot of the different control signals and state vectors when using the FPD controller, an IID controller, and an ON controller (see text for details).

the other states in the cost criterion. We compare the three finite horizon LQG controllers discussed in Section 6.4, namely the FPD controller, the IID controller, and the ON controller with their costs (6.15) and (6.16a).

The controllers are connected to the plant at sample times $k = 9, 10, 11$ through the network example given in Figure 6.7. Figure 6.8 shows the control signals computed by the different controllers and the plant states when the control signals are applied following the actual packet delivery sequence. From the predictions at $k = 8, 9, 10$ in Figure 6.7, we see that the FPD controller has better knowledge of the packet delivery sequence than the other two controllers. The FPD controller uses this knowledge to compute an optimal control signal that outputs a large magnitude for the second component of u_{10} , despite the high cost of this signal. The IID and ON controllers believe the control packet is likely to be delivered at $k = 11$ and choose, instead, to output a smaller increase in the first component of u_{11} , since this will have the same effect on the final state if the control packet at $k = 11$ is successfully delivered.

The FPD controller is better than the other controllers at driving the first component of the state close to zero at the end of the control horizon, $k = 12$. Thus, the packet delivery predictions from the network estimator help the FPD controller significantly lower its LQG cost, as shown in Table 6.9. The costs reported here are obtained from Monte-Carlo simulations of the system, averaged over 10,000 runs, but with the network state set to the one described above.

Table 6.9: Simulated LQG Costs (10,000 runs) for Example Described in Section 6.5

FPD Controller	IID Controller	ON Controller
681.68	1,008.2	1,158.9

6.6 Discussion on Network Model Selection

The ability of the network estimator to accurately predict packet deliveries is dependent on the network model. A natural objection to the GEIHS network model is that it assumes links are independent and does not capture the full behavior of a lossy and bursty wireless link through the G-E link model (Willig et al., 2002). Why not use one of the more sophisticated link models mentioned by Willig et al. (Willig et al., 2002)? Why not use a network model that can capture correlation between the links in the network? A good network model must be rich enough to capture the relevant behavior of the actual network, but not have too many parameters that are difficult to obtain.

In our problem setup, the relevant behavior is the packet delivery sequence of the network. As mentioned in Section 6.3.4, the network state probability distribution does not need to identify the exact network topology realization to get precise packet delivery predictions. In this regard, the GEIHS network model has too many states (2^E states) and may be overmodeling the actual network. However, the more relevant question is: Does the GEIHS network model yield accurate packet delivery predictions, predictions that are close to the actual future packet delivery sequence? Do the simplifications from assuming link independence and using a G-E link model result in inaccurate packet delivery predictions? These questions need further investigation, involving real-world experiments.

Our GEIHS network model has as parameters the routing topology G , the schedule $\mathbf{F}^{(T)}$, the G-E link transition probabilities $\{p_l^u, p_l^d\}_{l \in \mathcal{E}}$, the source a , the sink b , the packet generation times t_k , and the deadlines t'_k . The most difficult parameters to obtain are the link transition probabilities, which must be estimated by link estimators running on the nodes and relayed to the GEIHS network estimator. Furthermore, on a real network these parameters will change over time, albeit at a slower time scale than the link state changes. The issue of how to obtain these parameters is not addressed in this chapter.

Despite its limitations, the GEIHS network model is a good basis for comparisons when other network models for our problem setup are proposed in the future. It also raises several related research questions and issues.

Are there classes of routing topologies where packet delivery statistics are less sensitive to the parameters in our G-E link model p_l^u and p_l^d ? How do we build networks (e.g., select routing topologies and schedules) that are “robust” to link modeling error and provide good packet delivery statistics (e.g., low packet loss, low delay) for NCSs? The latter half of the question, building networks with good packet

delivery statistics, is partially addressed by other works in the literature like Soldati et al. (Soldati et al., 2010), which studies the problem of scheduling a network to optimize reliability given a routing topology and packet delivery deadline.

Another issue arises when we use a controller that reacts to estimates of the network's state. In our problem setup, if the network estimator gives wrong (inaccurate) packet delivery predictions, the FPD controller can actually perform *worse* than the ON controller. How do we design FPD controllers that are robust to inaccurate packet delivery predictions?

6.7 Summary

This chapter proposes two network estimators based on simple network models to characterize wireless mesh networks for NCSs. The goal is to obtain a better abstraction of the network, and interface to the network, to present to the controller and (future work) network manager. To get better performance in a NCS, the network manager needs to *control and reconfigure* the network to reduce outages and the controller needs to *react to or compensate for* the network when there are unavoidable outages. We studied a specific NCS architecture where the actuation channel was over a lossy wireless mesh network and a network estimator provided packet delivery predictions for a finite horizon, Future-Packet-Delivery-optimized LQG controller.

There are several directions for extending the basic problem setup in this chapter, including those mentioned in Sections 6.3.3, 6.3.4, and 6.6. For instance, placing the network estimator(s) on the actuators in the general system architecture depicted in Figure 6.1 is a more realistic setup but will introduce a lossy channel between the network estimator(s) and the controller(s). Also, one can study the use of packet delivery predictions in a receding horizon controller rather than a finite horizon controller.

Conclusions and Future Work

This thesis deals with contention-based multiple access architectures for NCSs. We model a multiple access network on the sensor link of a closed loop system, and design the access mechanism. We propose a state-aware MAC, which can be used to influence the probability of a successful transmission through the contention-based network. We consider two realizations of such a MAC, and present an analysis of each realization. In a separate study, we design the optimal controller, which compensates for lost packets on the controller link, due to the presence of a mesh network between the controller and the actuator. We present a short summary of the contributions of this thesis, and directions for future work below.

7.1 Conclusions

In this thesis, we model a closed loop system with a multiple access network on its sensor link, and identify three multiple access architectures. The first two architectures are the static and dynamic MACs, which can be found in use today. We analyze their effect on the performance of a closed loop system in the network. We also propose a new architecture, called the adaptive MAC, where the plant state is used to determine access to the shared medium. Thus, the adaptive MAC is state-aware, and we provide two realizations of this MAC.

The state-based scheduler regulates the flow of data from the plant, to obtain a better control cost given the current traffic in the network. We analyze the impact of a state-based scheduler on the design of the closed loop system, and find that there is a dual effect when the scheduler uses the state of the plant to initiate a transmission in the network. The optimal controller is difficult to find. However, with the imposition of two conditions on the scheduling criterion, we find that the design of the closed loop system is significantly simplified. A symmetric scheduling criterion, which is not a function of the applied control signals, results in a separation of the scheduler, observer and controller designs. The optimal controller in this setup is certainty equivalent. These results form the basis of the dual predictor architecture that we propose for NCSs. The scheduler in this architecture results

in a network-aware event-triggering mechanism.

We also look at Attention-based tournaments, which is another realization of a state-aware MAC more suited for delay sensitive systems. This MAC provides a mechanism to evaluate adaptive state-based priorities, called Attentions, in a contention-based setting. We use the innovations in the current measurement to determine the Attention required for a given data packet from across the network. This permits a generic formulation for a MAC in a heterogenous network, where all the nodes need not belong to closed loop systems. A bit-dominance strategy called the tournament is used to evaluate the priorities, and determine access to the shared medium. We present a performance analysis for this MAC, and compute the probability of a successful transmission for a node in this network. This is required to quantify the estimation error across the network, or the control cost for a closed loop system using this MAC.

This thesis also presents a study on mesh networks for NCSs, with the network on the actuating link. We propose two network estimators, which are used by the optimal controller to compensate for packets dropped in this network.

7.2 Future Work

There are several directions to further develop the work presented in this thesis. We list some of these below, as a reference for future work.

7.2.1 A MAC for the Actuating Link

This thesis explores the design of a MAC for the sensing link. A similar study must be performed for the actuating link as well. There are some dualities in the problem setup which can be exploited, but there are significant differences as well. The change in the problem formulation would affect the design of the closed loop system.

7.2.2 Optimal Controller Design

The results in this thesis establish the sub-optimality of certainty equivalent controllers for a closed loop system with a state-based scheduler, when the input arguments to the scheduling criterion include the applied controls. The optimal controller can be found by solving the expression in (4.21) for a 2-step horizon LQG cost. This could provide a starting point for investigation into the design of the optimal controller.

7.2.3 Flow Control with State-based Schedulers

State-based schedulers make it possible to regulate the flow of traffic from a node in a network of control loops, through appropriate choice of the scheduling threshold. In the examples presented in this thesis, the scheduling threshold was chosen

through simulations, not analysis. What is needed is an algorithm to choose the scheduling threshold, possibly in a socially optimal manner. This would provide a strategy to perform flow control for NCS. To accomplish this task, two other studies must be completed:

- Performance Analysis of a MAC with state-based schedulers: Tuning parameters in the state-based scheduler in accordance with the current network traffic could result in a scheduling law that guarantees a probabilistic performance. This is currently not easy to show, as the performance analysis of a closed loop system with a state-based scheduler in a multiple access network has not been accomplished. It is not trivial as independent plant states become correlated through interactions in a state-based MAC for feedback systems. However, it might be possible to analyze the performance at steady state.
- Performance Guarantees for a closed loop system: The state-based schedulers presented in this thesis do not provide any performance guarantees. An analysis of the link between parameters of the scheduler, and closed loop properties such as stability, would be useful in defining multiple access architectures for NCSs. These conditions can be used as constraints from the MAC in a flow control formulation.

7.2.4 Flow Control with Tournaments

The Attention-based tournaments offer a simpler setup to develop a flow control mechanism for NCSs. The performance analysis of this MAC has already been presented in this thesis. However, a detailed study linking the effect of parameters of this MAC to the traffic outcome for the network is still required.

7.2.5 A Network Interface for NCSs

A large part of the work in this thesis has been derived from the model of multiple access for NCSs presented in Chapter 1. This model implicitly assumes a single link between the sensor and the controller. If we assume that there is a more realistic mesh network between the sensor and the controller, the channel access parameters in our model become symbolic parameters. Any tuning of these parameters must be realized across distributed, and possibly correlated links. A better solution would be to use a more realistic model for the network and the MAC. However, very realistic models can be quite complicated, and difficult to analyze or gain insight from. Also, many of the parameters in such a model may not directly affect the closed loop system. Thus, a simplification of the interface between networks and closed loop systems is required to build a simple and effective model for NCSs.

7.2.6 Implementation of Prioritized MACs

The work in this thesis indicates that adaptive MACs offer an advantage over other MACs. Protocols such as tournaments, which can realize such a MAC by allowing wireless nodes to communicate between each other in a more direct manner, are hard to implement in wireless nodes. This is because tournaments require time synchronization, the ability to switch from transmission mode to reception mode, etc. An investigation into the feasibility of implementing prioritized medium access protocols, such as tournaments on a wireless sensor node, are required to advance the use of such a MAC.

There is a trade off between the design of an ideal mechanism for evaluating priorities in a MAC and the throughput achievable from its realization on a real radio transceiver. One aspect of the study could identify the most suitable design of a prioritized MAC compatible with the current generation of radio technology. Prioritized MACs are designed as modifications that are effected in software, in the MAC Layer of the protocol stack. Thus, another aspect of the study could identify a hardware-specific design that would lead to a significant increase in throughput from a prioritized MAC.

A.1 Appendix to State-based Schedulers

Derivation of the 2-Step Horizon Example Here are the complete derivations for the estimates and the Bellman equations for the optimal controller for Example 4.6.2.

A.1.1 Derivation of Estimates

From (4.26), we get

$$\hat{x}_{0|0} = \begin{cases} x_0 & \delta_0 = 1 \\ \mathbb{E}[x_0|x_0 < 0.5] & \delta_0 = 0 \end{cases} . \quad (\text{A.1})$$

As $x_0 \sim \mathcal{N}(0, 1)$, we can find the expected value

$$\bar{x}_{\delta_0} := \mathbb{E}[x_0|x_0 < 0.5] = \int_{-\infty}^{0.5} x \phi_{x_{\delta_0}}(x) dx ,$$

where $\phi_{x_{\delta_0}}$ is the conditional probability distribution function (pdf) of x_0 , conditioned on $x_0 < 0.5$. Thus, $\phi_{x_{\delta_0}}(x) = \phi_{x_0}(x)/Pr(x_0 < 0.5)$, where ϕ_{x_0} is the pdf of x_0 . The probability of a non-transmission, $Pr(x_0 < 0.5)$, is given by

$$Pr(x_0 < 0.5) = \int_{-\infty}^{0.5} \phi_{x_0}(x) dx .$$

The estimation error $\tilde{x}_{0|0}$ is thus given by

$$\tilde{x}_{0|0} = \begin{cases} 0 & \delta_0 = 1 \\ x_0 - \bar{x}_{\delta_0} & \delta_0 = 0 \end{cases} . \quad (\text{A.2})$$

The pdf of $\tilde{x}_{0|0}$ is $\phi_{\tilde{x}_0}(x) = \phi_{x_{\delta_0}}(x + \bar{x}_{\delta_0})$. The estimation error variance is given by

$$P_{0|0} = \begin{cases} 0 & \delta_0 = 1 \\ R_{\tilde{x}_0} & \delta_0 = 0 \end{cases}, \quad \text{where,} \quad R_{\tilde{x}_0} = \mathbb{E}[(x_0 - \bar{x}_{\delta_0})^2 | x_0 < 0.5] = \int_{-\infty}^{0.5 - \bar{x}_{\delta_0}} x^2 \phi_{x_{\delta_0}}(x + \bar{x}_{\delta_0}) dx. \quad (\text{A.3})$$

Let us denote e_1 as the unknown part of x_1 before y_1 is received:

$$e_1 = \begin{cases} w_0 & \delta_0 = 1 \\ ax_0 + w_0 & \delta_0 = 0 \end{cases}, \quad \text{and} \quad \phi_e(\epsilon) = \begin{cases} \phi_{w_0}(\epsilon) & \delta_0 = 1 \\ \phi_{e_{\delta_0}}(\epsilon) & \delta_0 = 0 \end{cases},$$

where, ϕ_e is the pdf of e_1 . The variable e_1 is the sum of two random variables if $\delta_0 = 0$, and its pdf is denoted $\phi_{e_{\delta_0}}$, and given by

$$\begin{aligned} \phi_{e_{\delta_0}}(\epsilon) &= \int_{-\infty}^{0.5} \phi_{x_{\delta_0}}(x) \phi_{w_0}(\epsilon - ax) dx \\ &= \frac{e^{-\epsilon^2/2(1+a^2)}}{\sqrt{2\pi(1+a^2)}} \left[\frac{Pr(t < \frac{1+a^2-2a\epsilon}{2\sqrt{1+a^2}})}{Pr(x_0 < 0.5)} \right], \end{aligned}$$

where, $t \sim \mathcal{N}(0, 1)$. Then, at the next time instant, we get

$$\hat{x}_{1|1} = \begin{cases} x_1 & \delta_1 = 1 \\ \begin{cases} ax_0 + bu_0 + \bar{w}_0 & \delta_0 = 1 \\ bu_0 + \bar{e}_{\delta_0} & \delta_0 = 0 \end{cases} & \delta_1 = 0 \end{cases}. \quad (\text{A.4})$$

As $w_0 \sim \mathcal{N}(0, 1)$, we can find the expected value

$$\bar{w}_0 = \mathbb{E}[w_0 | w_0 < 0.5 - ax_0 - bu_0] = \int_{-\infty}^{0.5 - ax_0 - bu_0} w \phi_{w_0}(w) dw,$$

where ϕ_{w_0} is the pdf of w_0 . Similarly, using the expression for $\phi_{e_{\delta_0}}$, we can derive

$$\begin{aligned} \bar{e}_{\delta_0} &= \mathbb{E}[ax_0 + w_0 | x_0 < 0.5, ax_0 + w_0 < 0.5 - bu_0] \\ &= \frac{1}{Pr(e_1 < 0.5 - bu_0)} \int_{-\infty}^{0.5 - bu_0} \epsilon \phi_{e_{\delta_0}}(\epsilon) d\epsilon, \end{aligned}$$

where $Pr(e_1 < 0.5 - bu_0)$ is the probability of no transmission at time $k = 1$. We know that

$$Pr(e_1 < 0.5 - bu_0) = \int_{-\infty}^{0.5 - bu_0} \phi_{e_{\delta_0}}(\epsilon) d\epsilon.$$

We now define \tilde{e}_1 as the error in estimating the term e_1 after y_1 arrives, with pdf $\phi_{\tilde{e}}$, so that

$$\tilde{e}_1 = \begin{cases} w_0 - \bar{w}_0 & \delta_0 = 1 \\ ax_0 + w_0 - \bar{e}_{\delta_0} & \delta_0 = 0 \end{cases}, \quad (\text{A.5})$$

and, $\phi_{\tilde{e}}(\epsilon) = \begin{cases} \phi_{\bar{w}}(\epsilon + \bar{w}_0 | w_0 < 0.5 - ax_0 - bu_0) & \delta_0 = 1 \\ \phi_{e_{\delta_0}}(\epsilon + \bar{e}_{\delta_0} | e_1 < 0.5 - bu_0) & \delta_0 = 0 \end{cases}$.

Now, we can define the estimation error variance $P_{1|1}$ by

$$P_{1|1} = \begin{cases} 0 & \delta_1 = 1 \\ R_{e_1} & \delta_1 = 0 \end{cases}, \quad (\text{A.6})$$

where $R_{e_1} = \mathbb{E}[\tilde{e}_1^2 | \delta_1 = 0]$ is given by

$$R_{e_1} = \begin{cases} \int_{-\infty}^{0.5 - ax_0 - bu_0 - \bar{w}_0} w^2 \frac{\phi_{w_0}(w + \bar{w}_0)}{\Pr(w_0 < 0.5 - ax_0 - bu_0)} dw & \delta_0 = 1 \\ \int_{-\infty}^{0.5 - bu_0 - \bar{e}_{\delta_0}} \epsilon^2 \frac{\phi_{\delta_0}(\epsilon + \bar{e}_{\delta_0})}{\Pr(e_1 < 0.5 - bu_0)} d\epsilon & \delta_0 = 0 \end{cases}.$$

Note that increasing u_0 will decrease R_{e_1} .

A.1.2 Derivation of V_1 and V_0

We use dynamic programming to find the Bellman equations V_1 and V_0 , which must be minimized to get u_1 and u_0 . Using (4.17), we write

$$\begin{aligned} V_1 &= \min_{u_1} \mathbb{E}[x_1^2 Q_1 + u_1^2 Q_2 + x_2^2 Q_0 | \mathbb{I}_1^C] \\ &= \min_{u_1} \mathbb{E}[x_1^2 (Q_1 + a^2 Q_0) | \mathbb{I}_1^C] + \text{tr}\{Q_0 R_w\} + u_1^2 (Q_2 + b^2 Q_0) + 2abQ_0 \hat{x}_{1|1} u_1. \end{aligned}$$

Minimizing the above expression with respect to u_1 , we get (4.36). Substituting for u_1 in the above expression for V_1 , we get

$$V_1 = \mathbb{E}[x_1^2 S_1 + \text{tr}\{\frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0} P_{1|1}\} | \mathbb{I}_1^C] + \text{tr}\{Q_0 R_w\},$$

where $S_1 = Q_1 + a^2 Q_0 - \frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0}$. To derive V_0 , we need to find the expected value $\mathbb{E}[P_{1|1} | \mathbb{I}_0^C]$. From the definition of $P_{1|1}$, we find that

$$\mathbb{E}[P_{1|1} | \mathbb{I}_0^C] = \Pr(\delta_1 = 0 | \mathbb{I}_0^C) \mathbb{E}[R_{e_1} | \mathbb{I}_0^C].$$

Then, we can find the u_0 that minimizes V_0 . We have

$$\begin{aligned}
 V_0 &= \min_{u_0} \mathbb{E}[x_0^2 Q_1 + u_0^2 Q_2 + V_1 | \mathbb{I}_0^C] \\
 &= \min_{u_0} \mathbb{E}[x_0^2 (Q_1 + a^2 S_1) | \mathbb{I}_0^C] + \text{tr}\{S_1 R_w\} + \text{tr}\{Q_0 R_w\} \\
 &\quad + u_0^2 (Q_2 + b^2 S_1) + 2\hat{x}_{0|0} ab S_1 u_0 + \frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0} \mathbb{E}[P_{1|1} | \mathbb{I}_0^C].
 \end{aligned}$$

Bibliography

- N. Abramson. The ALOHA system: another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference, AFIPS '70 (Fall)*, pages 281–285, New York, NY, USA, November 1970. ACM. URL <http://doi.acm.org/10.1145/1478462.1478502>.
- A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010.
- K. J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, 1970. Republished by Dover Publications, 2006.
- K. J. Åström and B. M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 2011–2016 vol.2, 2002.
- K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1995.
- M. Athans. The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *IEEE Transactions on Automatic Control*, 16(6):529–552, December 1971.
- R. Bansal and T. Basar. Simultaneous design of measurement and control strategies for stochastic systems with feedback. *Automatica*, 25(5):679–694, 1989. URL <http://www.sciencedirect.com/science/article/B6V21-47WTG8D-R9/2/4fa8dc03b45b2c63169e8f2fdad5c7a1>.
- L. Bao, M. Skoglund, and K. H. Johansson. Iterative encoder-controller design for feedback control over noisy channels. *IEEE Transactions on Automatic Control*, PP(99):1–14, 2010.
- Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500, October 1974.
- G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

- G. Bianchi, I. Tinnirello, and L. Scalia. Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations. *IEEE Network*, 19(4):28 – 34, 2005.
- R. Blind and F. Allgöwer. Analysis of networked event-based control with a shared communication medium: Part I - Pure ALOHA. *Submitted to IFAC World Congress*, 2011a. URL http://www.ist.uni-stuttgart.de/~blind/blind_IFAC11_I.pdf.
- R. Blind and F. Allgöwer. Analysis of networked event-based control with a shared communication medium: Part II - Slotted ALOHA. *Submitted to IFAC World Congress*, 2011b. URL http://www.ist.uni-stuttgart.de/~blind/blind_IFAC11_II.pdf.
- V. Borkar and S. Mitter. LQG control with communication constraints. *Communications, Computation, Control, and Signal Processing: a Tribute to Thomas Kailath.*, 1997. Norwell, MA, Kluwer Academic Publishers.
- R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3):123 – 131, 2005.
- R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: Recent developments. *Statistical Science*, 19(3):499–517, Aug. 2004.
- A. Cervin and T. Henningsson. Scheduling of event-triggered controllers on a shared network. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3601 – 3606, 2008.
- I. Chlamtac, M. Conti, and J. J. N. Liu. Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, 2003.
- M. C. F. Donkers and W. P. M. H. Heemels. Output-based event-triggered control with guaranteed L_∞ -gain and improved event-triggering. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
- N. Elia. Remote stabilization over fading channels. *Systems & Control Letters*, 54(3):237 – 249, 2005. URL <http://www.sciencedirect.com/science/article/B6V4X-4DVBWGH-1/2/1fd849197215a24e8cffbf2ca4454951>.
- A. A. Feldbaum. The theory of dual control. IV. *Automation and Remote Control*, 22:109 – 121, 1961.
- N. M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. *IEE Proceedings - Control Theory and Applications*, 147(1):118 – 128, January 2000.
- V. Gupta, B. Hassibi, and R. M. Murray. Optimal LQG control across packet-dropping links. *Systems & Control Letters*, 56(6):439 – 446, 2007. URL <http://www.sciencedirect.com/science/article/B6V4X-4MV1NT9-1/2/f15a97e79373553b875b5908f9ce6491>.

- V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi. Data transmission over networks for estimation and control. *IEEE Transactions on Automatic Control*, 54(8):1807–1819, 2009.
- Y. Halevi and A. Ray. Integrated communication and control systems. I - analysis. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control*, 110:367–373, December 1988a.
- Y. Halevi and A. Ray. Integrated communication and control systems. II - design considerations. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control*, 110:367–373, December 1988b.
- HART Communication Foundation. *WirelessHART Data Sheet*, 2007. Datasheet.
- T. Henningsson and A. Cervin. A simple model for the interference between event-based control loops using a shared medium. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
- J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *Proceedings of the IEEE*, volume 95, pages 138–162, 2007.
- S. H. Hong. Scheduling algorithm of data sampling times in the integrated communication and control systems. *IEEE Transactions on Control Systems Technology*, 3(2):225–230, June 1995.
- IEEE. *IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, 2006. URL <http://www.ieee802.org/15/pub/TG4.html>.
- International Society of Automation. ISA-SP100 wireless systems for automation website. <http://www.isa.org/isa100>, 2010.
- H. Ishii. H_∞ control with limited communication and message losses. *Systems & Control Letters*, 57(4):322–331, 2008. URL <http://www.sciencedirect.com/science/article/B6V4X-4R05J15-3/2/1316b8e02b9809c2ab67f7ac756c686b>.
- T. Kailath, A. H. Sayed, and B. Hassibi. *Linear estimation*. Prentice-Hall information and system sciences series. Prentice Hall, 2000. ISBN 9780130224644. URL http://books.google.com/books?id=gX_FQwAACAAJ.
- L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, December 1975.
- A. S. Krishnamoorthy and M. Parthasarathy. A multivariate gamma-type distribution. *The Annals of Mathematical Statistics*, 22:549–557, 1951.

- F. L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems Magazine*, 21(1):66–83, February 2001.
- X. Liu and A. Goldsmith. Wireless medium access control in networked control systems. In *Proceedings of the American Control Conference*, volume 4, pages 3605–3610 vol.4, 2004.
- A. S. Matveev and A. V. Savkin. The problem of state estimation via asynchronous communication channels with irregular transmission times. *IEEE Transactions on Automatic Control*, 48(4):670–676, 2003.
- A. Molin and S. Hirche. On LQG joint optimal scheduling and control under communication constraints. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 5832–5838, 2009.
- A. Molin and S. Hirche. Structural characterization of optimal event-based controllers for linear stochastic systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
- R. M. Murray et al. *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems*. Society for Industrial and Applied Mathematics, 2003. URL <http://www.cds.caltech.edu/~murray/cdspanel>.
- G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, 95(1):108–137, 2007.
- J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Lund Institute of Technology, Department of Automatic Control, Sweden, January 1998.
- R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- P. G. Otanez, J. R. Moyne, and D. M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proceedings of the American Control Conference*, volume 4, pages 3015–3020 vol.4, 2002.
- N. Pereira, B. Andersson, and E. Tovar. WiDom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics*, 3(2):120–130, May 2007.
- K. S. J. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In *Proceedings of the IASTED International Symposium on Distributed Sensor Networks (DSN)*, Nov. 2008.
- M. Rabi. *Packet based Inference and Control*. PhD thesis, Institute for Systems Research, University of Maryland, 2006. URL <http://hdl.handle.net/1903/6612>.

- M. Rabi and K. H. Johansson. Scheduling packets for event-triggered control. *Proceedings of 10th European Control Conference*, pages 3779 – 3784, 2009.
- C. Ramesh, H. Sandberg, and K. H. Johansson. LQG and medium access control. *Preprints of the 1st IFAC Workshop on Estimation and Control of Networked Systems*, pages 328 – 333, September 2009.
- A. Ray. Performance evaluation of medium access control protocols for distributed digital avionics. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control.*, 109:370 – 377, December 1987.
- Robert Bosch GmbH. *Bosch CAN Specification, ver. 2*. Stuttgart, 1991. URL <http://www.semiconductors.bosch.de/pdf/can2spec.pdf>.
- C. L. Robinson and P. R. Kumar. Control over networks of unreliable links: Controller location and performance bounds. In *5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops*, pages 1 –8, 2007.
- R. Rom and M. Sidi. *Multiple access protocols: performance and analysis*. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97253-6.
- L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1): 163 –187, 2007.
- P. Seiler and R. Sengupta. An H_∞ approach to networked control. *IEEE Transactions on Automatic Control*, 50(3):356 – 364, 2005.
- S. C. Smith and P. Seiler. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Transactions on Automatic Control*, 48(12):2163 – 2171, 2003.
- P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269, 1997.
- P. Soldati, H. Zhang, Z. Zou, and M. Johansson. Optimal routing and scheduling of deadline-constrained traffic over lossy networks. In *IEEE Global Telecommunications Conference*, pages 1 –6, 2010.
- P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680 –1685, 2007.
- S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. *IEEE Transactions on Automatic Control*, 49(9):1549 – 1561, 2004.

- D. Varagnolo, P. Chen, L. Schenato, and S. S. Sastry. Performance analysis of different routing protocols in wireless sensor networks for real-time estimation. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4904–4909, 2008.
- 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. *IEEE Transactions on Industrial Electronics*, 49(6):1265 – 1282, December 2002.
- Wireless Industrial Networking Alliance. WINA website. <http://www.wina.org>, 2010.
- H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):131–147, 1968. URL <http://link.aip.org/link/?SJC/6/131/1>.
- H. S. Witsenhausen. Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, 59(11):1557 – 1566, 1971.
- Y. Xu and J. P. Hespanha. Optimal communication logics in networked control systems. In *Proceedings of the 43rd Conference on Decision and Control*, volume 4, pages 3527 – 3532 Vol.4, 2004.
- Y. Xu, J. Winter, and W.-C. Lee. Dual prediction-based reporting for object tracking sensor networks. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 154 – 163, 2004.
- J. K. Yook, D. M. Tilbury, and N. R. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *IEEE Transactions on Control Systems Technology*, 10(4):503 – 518, July 2002.
- W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84 – 99, February 2001.