EFFICIENT DECENTRALIZED COMMUNICATIONS IN SENSOR NETWORKS

THÈSE N^{O} 2952 (2004)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Institut de systèmes de communication

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Razvan CRISTESCU

ingénieur informaticien et électricien, Université Polytechnique de Bucarest, Roumanie
et de nationalité roumaine

acceptée sur proposition du jury:

Prof. M. Vetterli, directeur de thèse
Dr B. Beferull-Lozano, rapporteur
Prof. G. Pottie, rapporteur
Prof. P. Thiran, rapporteur
Prof. R. Wattenhofer, rapporteur

Lausanne, EPFL
2004

Efficient Decentralized Communications in Sensor Networks

Răzvan Cristescu

These N. 2952 (2004)

Thèse presentée à la faculté informatique et communications pour l'obtention du grade de docteur ès sciences.

Prof. Martin Vetterli
Directeur de thèse

Prof. Greg Pottie
Rapporteur

Prof. Patrick Thiran
Rapporteur

Prof. Roger Wattenhofer
Rapporteur

Dr. Baltasar Beferull-Lozano
Rapporteur

Sous la présidence du **Prof. Matt Grossglauser**, *le 27 Février 2004.*

Ecole Polytechnique Federale de Lausanne, 2004.

Acknowledgments

I would like to thank my advisor, Prof. Martin Vetterli, for a great opportunity to pursue my PhD studies under his guidance, in the Audiovisual Communications Laboratory (LCAV) at EPFL. Martin's knowledge, wisdom and support were invaluable at all stages of my research work. His intuition and vision put my efforts on a right track when I was a freshman, and the incredibly broad knowledge that he shares, along with his advice, have lead my steps all the way to the completion of this thesis. I cannot think of a better advisor for my graduate studies than Martin.

My gratitude goes to Dr. Baltasar Beferull-Lozano, whose help in the second part of my thesis was essential. I thank him for fruitful long discussions that resulted in many original ideas, and for his capacity of structuring complex problems that always helped clarify my thoughts.

I would like also to thank Prof. Sergio Servetto for the offer to work in LCAV on a very exciting and original doctoral project, that lead to the first part of my thesis. I also want to thank him for the continuous collaboration we have had.

I would like to thank the members of my thesis committee for their thorough review of my work, and for the discussion during my thesis defense.

I am grateful to all my colleagues that have created a very nice working environment at LCAV, especially to my office mates Irena and Rahul, and to all current and past members of the lab, for the rich scientific environment and for the entertaining discussions. I am also happy I had the chance to interact with the top-level faculty from the Communication Systems department at EPFL.

I would like especially to thank my friends, who made these years in Lausanne so beautiful. Most importantly, I want to thank my romanian friends, most of them at EPFL, for being the dear friends they are, and for making my life after work so rich and so pleasant.

I consider myself very lucky, as I cannot think of a better place to study for a PhD than EPFL, and a better place to live while doing that, than Lausanne.

Last, but not least, I want to thank my parents for their love and support, and for teaching me to always believe in myself.

Abstract

This thesis is concerned with problems in decentralized communication in large networks. Namely, we address the problems of joint rate allocation and transmission of data sources measured at nodes, and of controlling the multiple access of sources to a shared medium. In our study, we consider in particular the important case of a sensor network measuring correlated data.

In the first part of this thesis, we consider the problem of correlated data gathering by a network with a sink node and a tree communication structure, where the goal is to minimize the total transmission cost of transporting the information collected by the nodes, to the sink node. Two coding strategies are analyzed: a Slepian-Wolf model where optimal coding is complex and transmission optimization is simple, and a joint entropy coding model with explicit communication where coding is simple and transmission optimization is difficult. This problem requires a joint optimization of the rate allocation at the nodes and of the transmission structure. For the Slepian-Wolf setting, we derive a closed form solution and an efficient distributed approximation algorithm with a good performance. We generalize our results to the case of multiple sinks. For the explicit communication case, we prove that building an optimal data gathering tree is NP-complete and we propose various distributed approximation algorithms. We compare asymptotically, for dense networks, the total costs associated with Slepian-Wolf coding and explicit communication, by finding their corresponding scaling laws and analyzing the ratio of their respective costs. We argue that, for large networks and under certain conditions on the correlation structure, 'intelligent', but more complex Slepian-Wolf coding provides unbounded gains over the widely used straightforward approach of opportunistic aggregation and compression by explicit communication.

In the second part of this thesis, we consider a queuing problem in which the service rate of a queue is a function of a partially observed Markov chain, and in which the arrivals are controlled based on those partial observations so as to keep the system in a desirable mildly unstable regime. The optimal controller for this problem satisfies a separation property: we first compute a probability measure on the state space of the chain, namely the information state, then use this measure as the new state based on which to make control decisions. We give a formal description of the system considered and of its dynamics, we formalize and solve an optimal control problem, and we show numerical simulations to illustrate with concrete examples properties of the optimal control law. We show how the ergodic behavior of our queuing model is characterized by an invariant measure over all possible information states, and we construct that measure. Our results may be applied for designing efficient and stable algorithms for medium access control in multiple accessed systems, in particular for sensor networks.

Résumé

Cette thèse est concernée par des problèmes de la communication décentralisée dans des grands réseaux. Nous abordons d'une part le problème d'allocation de débit et de transmission de données mesurés aux noeuds, d'autre part le probléme de controle d'accès multiple des sources à un milieu partagé. Dans notre étude, nous considérons en particulier le cas important d'un réseau de senseurs mesurant des données corrélées.

Dans la première partie de cette thèse, nous considérons le problème de récolte de données corrélées par un réseau ayant un noeud évier et une structure de communication en arbre, où le but est de réduire au minimum le coût de transmission pour transporter l'information rassemblé par les noeuds au noeud évier. Deux stratégies de codage sont analysées: un modèle Slepian-Wolf où le codage optimal est complexe et l'optimisation de transmission est simple, et un modèle de codage avec la communication explicite où le codage est simple et l'optimisation de transmission est difficile. Ce problème exige une optimisation commune de l'allocation de débit aux noeuds et de la structure de transmission. Pour le codage Slepian-Wolf, nous dérivons une solution forme-fermée et un algorithme distribué efficace d'approximation. Nous généralisons nos résultats au cas d'éviers multiples. Pour le codage avec communication explicite, nous montrons qu'établir l'arbre de transmission est NP-complet et nous proposons divers algorithmes distribués d'approximation. Nous comparons asymptotiquement, pour des réseaux denses, les coûts liés au codage Slepian-Wolf et au codage avec communication explicite, en trouvant leurs lois correspondantes et en analysant le rapport du leur coûts respectifs. Pour des grands réseaux et dans certaines conditions sur la structure de corrélation, un codage Slepian-Wolf plus complexe mais 'intelligent' fournit des gains illimités au-dessus du l'approche largement répandue de l'agrégation et de la compression par communication explicite.

Dans la deuxième partie de cette thèse, nous considérons un problème dans lequel le taux de service d'une file d'attente est une fonction d'une chaîne de Markov partiellement observée, et où les arrivées sont basées sur des observations partiels, afin de maintenir le système dans un régime modérément instable souhaitable. Le contrôleur optimal pour ce problème satisfait une propriété de séparation: nous calculons d'abord une mesure de probabilité sur l'espace d'état de la chaîne, à savoir l'état de l'information, en employant alors cette mesure comme le nouvel état pour prendre des décisions de commande. Nous donnons une description formelle du système considéré et de sa dynamique, nous formalisons et résolvons un problème de commande optimale, et nous montrons avec des simulations numériques les propriétés concrètes de la loi de commande optimale. Nous montrons comment le comportement ergodique de notre modèle est caractérisé par une mesure invariable au-dessus de tous états possibles de l'information, et nous construisons cette mesure. Nos résultats peuvent être appliqués pour concevoir des algorithmes efficaces et stables pour le contrôle d'accès dans les systèmes d'access, en particulier pour les réseaux de senseurs.

Contents

List of Figures

List of Acronyms

CDMA	Code Division Multiple Access
det	Determinant (of a matrix)
DP	Dynamic Programming
LD	Leaves Deletion algorithm
LP	Linear Programming
MAC	Multiple Access Control
MST	Minimum Spanning Tree
SA	Simulated Annealing algorithm
SPT	Shortest Path Tree
SPT/TSP	Balanced *SPT/TSP* tree
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
TSP	Traveling Salesman Path
WSS	Wide Sense Stationary (process)

Chapter 1

Introduction

1.1 Distributed Communications in Sensor Networks

Sensor networks have recently received a considerable attention from the research community. Such networks are composed of autonomous units (sensors), usually deployed in areas where it is difficult or expensive to use an existing classical communication infrastructure [57], [54], [20], [34]. Sensor networks are used to measure properties of the environment (temperature, seismic data etc.). Data have to reach some specific destination, for further processing, storage or use.

Communication in sensor networks has to be done over an inherently multiple access channel. Moreover, the sensors have limited processing capabilities and must rely on a small amount of energy (battery power). Thus, in most practical cases, communications in such networks have to be done in a distributed manner. The study of various research problems encountered in distributed communications in sensor networks is the main goal of this thesis.

1.2 Communications over Large Networks

In this work, we will analyze various problems in communications in systems with a large number of users. In many practical scenarios, the users have limited processing and transmission abilities. In such systems, communications rarely take place solely in an isolated manner, that is, from some end providing data to another end which needs that data (this kind of communication is usually refereed as point to point). On the contrary, communications in this case rather involve various users participating: one or more users sending data, possibly other users helping the transmission (e.g. by relaying data), and some other set of users receiving the data.

A typical example of such a system is the case of a sensor network, where deployed nodes measure some property of the environment like temperature or seismic data. Data from these nodes is transmitted over the network, using other nodes as relays, to one or more base stations, for storage or control purposes.

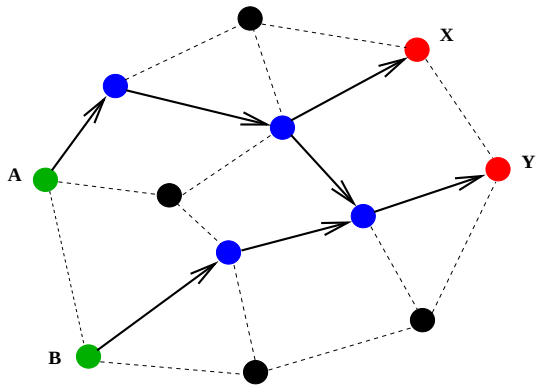


Figure 1.1: An example of a communication network. Source nodes A and B (in green) use the relays (in blue) to transmit their data to the sinks X and Y (in red). The chosen transmission structure is shown in bold arrows; the other links are represented in dashed lines.

An abstraction of a communication system is the *network* entity. Users are generically represented as *nodes*. We call the users from where data originates as *sources*, and we call *sinks* the users where the data is needed. The nodes are able to communicate among them by means of communication *links*. Some of the nodes are supplied with amounts of information (*supplies*). These data must travel across the network, over the links, possibly via some other nodes, to their destinations, or sinks. Sinks have *demands* (negative supplies) assigned. The example in Figure 1.1 shows a network where the sources A and B transmit their data over a set of selected links, highlighted by arrows, via relays, to the sinks X and Y.

Communications over large networks are subject to some important constraints. Like in any network, flow conservation laws are valid: the total amount of data entering a relay node together with its supply must be equal to the amount of data that exits that node.

There are also supplementary constraints, that take into account the specific limitations of such networks. These constraints essentially result from the fact that the resources available at nodes are limited. For instance, in the case of sensor networks, these resources include battery power and processing capabilities. We consider the graph corresponding to the sensor network, that has the sensor sources as nodes, and the connecting links given by the (radio) transmission range of nodes. As battery power is limited, some links may be more expensive to use than others. For instance, if the two ends of a link are at a large distance far apart from each other, the power needed to transmit over that link is very large, and thus that link is not efficient to use. In order to differentiate among links, they are assigned costs, or weights, that account for how expensive it is to use that link. The link weights can be either bulk or, more generally, per unit of use (for instance, per packet/time slot).

Another constraint on communications is associated with the nodes that relay transmission (the relays). Namely, the relay nodes have a maximum bandwidth associated to each of them. This bandwidth is determined for instance by the limited

size of their buffers and the finite rate of processing. This means that such a node cannot forward data simultaneously from more than a limited number of other nodes that use it as a relay. The nodes accessing the relay need usually to content for the access.

All these constraints present in a network give rise to interesting research questions. Such questions are aimed at deciding what amount of supply should each of the nodes provide for complete representation of the data, what links should be used for efficient transmission, and how to deal with the information flow bottlenecks for maximizing the throughput. In this work, we will use a unified approach to address these questions.

One important issue that has to be taken into account in the study of large networks is the fact that it is rather impossible to develop centralized algorithms for tasks such as the design of the transmission structure, rate allocation, or flow control. Any centralized approach, even if realistic in some cases for small networks, is not scalable in general. Thus, the design of *decentralized* algorithms should play an important role in the approach of typical network issues. These issues include:

- coding of the information measured at the nodes,
- the transmission and routing of that information across the network from its source to the destination,
- scheduling of the access to the multiple access (MAC) layer, when more than one source access a limited bandwidth communication channel at the same time.

Consider again the example of a sensor network. In such a case, the nodes are usually deployed randomly, are autonomous (meaning that their battery is limited and not renewable), and communication over long distances is very costly. It is thus impossible in practice to employ a central agent to resolve the tasks enumerated above. Any practical implementation of such a network must resort to decentralized algorithms.

Traditionally, the network aspects mentioned above have been considered as separate problems by the research community. An important goal of this work is to illustrate how a joint treatment of these issues may provide important improvements over approaches that consider each layer separately.

1.3 Motivation

We will consider sensor networks as an example, and as main motivation for our discussion further. Note that our analysis can be easily generalized to other types of networks, where centralized control is unrealistic. We will begin by enumerating some of the important issues that make large scale sensor networks different from classical networks, and thus require new decentralized, efficient, and computationally simple methods to solve them.

First, we consider the primary task of sensing. In these networks there exists a set of privileged end sites (sinks, or base station) where data from the sensors is needed for measure, control or storage purposes. The goal of a sensor network is, in most cases, to transport the data sensed by the nodes to the base stations. An important

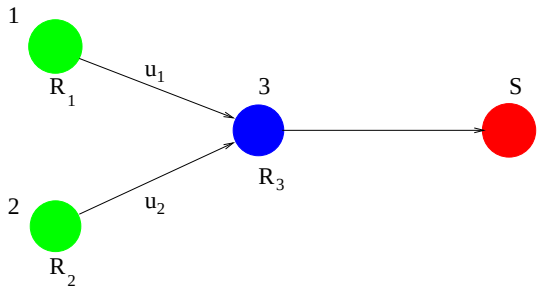


Figure 1.2: The tasks of MAC control and rate allocation/transmission structure optimization for a simple network.

aspect that can be exploited is that, in general, the data measured at different sites on the field is *correlated*. This means that nodes that are spatially close to each other measure data characterized by a strong interdependence.

Second, we consider the task of transmission. Sending the data over the links costs energy which is usually proportional to a function of the length of that link. This function is usually monotonically increasing. Due to this reason, the straight line is in general not the most cost efficient way to transmit between two points. Cost efficient paths rather make use of relay nodes. These can be nodes especially assigned for that task, or nodes that simultaneously perform sensing and relaying.

Third, from the point of view of the shared transmission layer, the multiple access cannot be controlled in a centralized manner by the node that is acting as a relay, since communication with the children is very costly. It is thus needed to design simple decentralized algorithms that regulate the access to the shared medium. These algorithms, implemented by the data sources, should rely only on limited feedback from the routing node.

Finally, in order to perform all these tasks, nodes must have multiple functionalities, while only limited energy and processing abilities are available locally. To summarize, typical functions of a node include:

- Sense the environment.
- Relay data coming from other nodes.
- Control their rate of sending further the measured and/or relayed data.

We illustrate these issues with a simple network example shown in Figure 1.2.

All three nodes $1, 2, 3$ sense the environment. The data measured by the nodes 1, 2 and 3 have to arrive at the base station S. We consider now the various tasks the nodes have to perform.

1.3.1 Rate Allocation and Transmission Structure Optimization

First, nodes have to code their data so that it can be fully reconstructed when it arrives at the base station S. We assume the measured quantities are random variables. This involves a decentralized *rate allocation*. Let us denote the rate allocation at the three

nodes with (R_1, R_2, R_3). This rate allocation has to provide an exact representation of the data measured.

For instance, let us consider a scenario where nodes measure temperature, and the values of the measurements are approximated by integer numbers. Suppose that the reading at a certain time slot is $T_1 = 10^o, T_2 = 9^o, T_3 = 11^o$. If the data measured at nodes were independent, then the rates R_1, R_2 and R_3 can be chosen independently, too. Then, for independent encoding, a possible rate encoding would be $R_1 = R_2 = R_3 = 4$ bits. However, in general, the data measured at nodes *is* correlated. Then, depending on the correlation model, it is very likely that, given the value of $T_1 = 10^o$, the values of T_2 and T_3 would rather have a value in the set $\{9^o, 10^o, 11^o\}$, than outside this set. Then, if the nodes were aware of their inter-correlation, they can use for instance[1] $R_1 = 4$ bits to code the measurement at node 1, and $R_2 = R_3 = 2$ bits to code nodes 2 and 3 relative difference from the measurement of node 1.

We see however that even without a-priori knowledge about the correlation structure, node 3 can assess the interdependence in the measured data by using the data received from nodes 1 and 2, since it is used as a relay by those nodes. Thus, it can adjust its coding rate accordingly (for example, in Figure 1.2, nodes can code with $R_1 = R_2 = 4$ bits, but $R_3 = 2$ bits).

Let us now consider the costs on the links as well. If the links costs were simply given by the Euclidean distance, then the straight line between any node and the sink would be the best path to transmit data from that node. In this case, no node would be used as relay. However, in practical situations, the cost function on a link depends super-linearly on the length of that link. Thus, nodes 1 and 2 may use node 3 as relay, and not waste their energy by sending data directly to the sink over a large distance.

Even in this simple network example, it is not easy to determine the optimal transmission structure. In Figure 1.2, we show a possible transmission structure for gathering data from nodes $1, 2$ and 3. In this scenario, the links used are shown as arrows in the figure, and the rates are $R_1 = R_2 = 4$ bits, $R_3 = 2$ bits. However, suppose the following alternative scenario is employed: node 3 sends its $R_3 = 4$ bits data to node 2, which reduces the rate at node 2 from $R_2 = 4$ to $R_2 = 2$ bits, and then node 2 sends the whole $R_1 + R_2 = 6$ bits to node 3. Then, the trade-off is of whether or not, in terms of total cost, to reduce the rate R_2 at the expense of increasing the total transmission path traveled by the data from node 1 to the sink.

The situation is different should nodes be aware beforehand of the correlation structure, namely the transmission structure and rate allocation can be more easily determined, at the cost of considerably increased complexity in coding. The study of the trade-offs involved in the rate allocation/transmission structure optimization for networks measuring correlated data is what drives the first part of this work.

1.3.2 Multiple Access Control

Suppose now that, as a result of the joint optimization for power efficiency in the previous paragraph, we obtain a transmission structure in which nodes 1 and 2 do indeed use both as relay the node 3. If the relay is aware of the numbers of nodes that access it (in this case, two), it can just allocate some fair proportion of its bandwidth

[1] This can be done by Slepian-Wolf coding, that we present later in this work.

to each of them, avoiding thus collisions. However, such an information is not available in general nor at the relay, neither at the nodes accessing it. Moreover, node 3 has a limited processing/buffering ability, thus it cannot accept, process and forward unconditionally all information received from its children nodes. In order to avoid too many losses, it is thus necessary that nodes 1 and 2 control their transmission rates such that this results in packet losses below a given level, while ensuring a steady throughput. This requires a multiple access control (MAC) protocol that the three nodes involved need to implement.

Suppose each of the two nodes 1 and 2 employs a simple random medium access protocol, defined by two Bernoulli random variables that determine the injection probabilities, respectively. Let us denote by u_1, u_2 the transmission rates with which the nodes 1 and 2 access the relay (router). These random variables may for instance be weighted by the actual data rate that the nodes have to transmit (i.e. $E[u_1] \approx R_1, E[u_2] \approx R_2$). Due to the above mentioned power and communication limitations, the nodes are not able to communicate between them. For the same reasons of minimizing the overhead, they need to control the rate of transmission by using only limited information (feedback) from the relay node. This feedback is usually restricted only to acknowledgments from the router of whether the packet sent has been accepted or not. Most current protocols for data transmission, including Aloha or TCP, use this kind of information for the rate control. Current proposals for medium access protocol in sensor networks make use of randomized controllers. The study of performance and stability of such protocols is thus of obvious importance.

As an example, suppose node 1 uses a probability of injection $u_1 = 0.5$, that is, it will try to inject on average a packet every two time slots. If it sends a packet and this is accepted (there is free place in the buffer of node 3), an adequate policy will consequently increase its rate u_1, since it is probable that node 2 is not active at that particular time. As a result, node 1 accesses the buffer more often. If on the contrary the packet is rejected, then it is probable that node 2 accesses the channel in the same time, too. Then, node 1 will decrease its rate. Note that care must be taken so that none of the nodes takes alone full use of the buffer. This fairness can be achieved, for instance, by drastically reducing the injection probability when losses are experienced. The design and analysis of such control policies is the goal of the second part of this work.

From our discussion so far, it is clear that there is a stringent need for simple, decentralized and efficient algorithms for sensing, transmission and controlling in sensor networks. To summarize, the main issues that motivate our work and constrain the algorithms that we propose along this thesis are:

- Decentralization;
- Power efficiency;
- Joint sensing/transmission/control.

As underlined in the previous paragraphs, our contribution addresses two fundamental tasks for these systems:

- The joint treatment of source coding and transmission structure, as a function of the available knowledge about the correlation structure.

- The control of the access to the shared medium by means of partial observations obtained about the state of the system by simple protocols[2].

1.4 Contributions

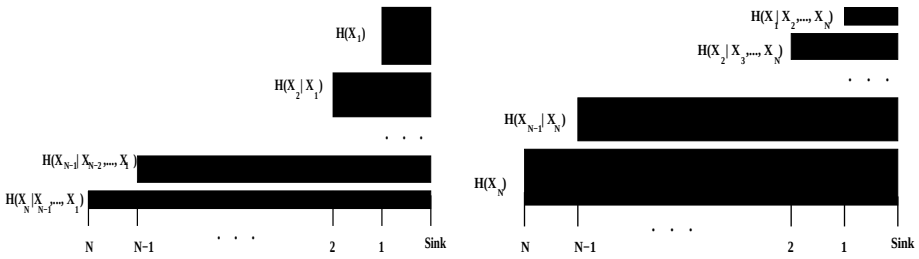
1.4.1 Network Correlated Data Gathering

Consider a network of N nodes. Let $X_1, \ldots, X_N$ be the random variables representing the data measured at nodes $1, \ldots, N$. This data at nodes is correlated, and has to arrive at a sink (or set of sinks), by using the links of the network.

The goal of the first part of this work is to find a rate allocation $R_1, \ldots, R_N$ at the nodes, that fully describe the measured information, while minimizing the cost [function (data size)] $\times$ [link weight]. For example, for a sensor network, this cost is of particular interest, being the total energy utilized by the network. Our main contributions are:

- An original problem statement that jointly addresses the rate allocation and the transmission costs for correlated sources.
- The identification of the two possible coding cases, namely:
  - The case when additional data (side information) is available at a node only when relayed from other nodes (we call this the *explicit communication* case).
  - The case when full information of the correlation structure in the network is known locally at a node, and thus Slepian-Wolf coding of correlated data can be employed.
- The full analysis of Slepian-Wolf coding, namely:
  - An analysis of the complexity in the arbitrary traffic matrix case: we prove that the problems of rate allocation and transmission optimization separate. Further, we show that finding the transmission structure is NP-complete, and finding the optimal rate allocation reduces to Lagrangian minimization under constraints. If the cost function is linear, then the optimization problem is a linear program (LP).
  - A full analysis of the special case of data gathering: we show the optimal transmission structure is the shortest path tree (SPT), and then find the optimal rate allocation in closed form. We are also able to provide performant approximation algorithms, based on only *local information*.
  - We assess the complexity of practical implementation of algorithms for rate allocation based on Slepian-Wolf coding.
- The complexity analysis of the explicit communication case, namely:

[2]The design of such systems is directly related with the Heisenberg principle: to control a system requires to observe/probe the system, and thus to influence the system by the respective observation.



(a) Slepian-Wolf coding. (b) Explicit communication coding.

Figure 1.3: One-dimensional example: the transmission costs are represented by filled rectangles (allocated rate on the vertical, total distance to the sink on the horizontal). If $\lim_{i\to\infty} i \cdot H(X_i|X_{i-1},\ldots,X_1) = 0$, then the transmission cost for Slepian-Wolf coding (left) is asymptotically arbitrarily smaller than the transmission cost for explicit communication (right): $\lim_{N\to\infty} \frac{c_{SW}(N)}{c_{EC}(N)} = 0$.

  - We state an original graph problem corresponding to a simplified setting, and we show that the problem is NP-complete even in this very simple case. There is no longer a separation of the rate allocation optimization and transmission optimization, and we provide simple examples where alternative structures provide asymptotically unbounded better results in terms of total energy than usual structures such as the SPT.
  - We provide performant heuristics based on insights given by simulated annealing.

- We compare asymptotically, for dense networks, the total costs associated with Slepian-Wolf coding and explicit communication, by finding their corresponding scaling laws and analyzing the ratio of their respective costs. We show that, for large networks and under certain conditions on the correlation structure, Slepian-Wolf coding provides unbounded gains over explicit communication (we illustrate this with a simple one-dimensional example, in Figure 1.3).

- We illustrate all our results with numerical examples based on a Gaussian random field example with correlation matrix dependent on the inter-node distances.

1.4.2 Multiple Access Control with Limited Feedback

Now, suppose that in the resulting transmission structure, there are relay nodes used by one or more nodes. We assume these nodes have no information about each other's scheduling of transmissions, because this would require a large overhead, and the relay nodes have a limited bandwidth. Each of the nodes implements a random controller for accessing the shared medium, which essentially is a Bernoulli random variable u_i that represents the probability of packet injection by that node. The goal is to get as many packets through as possible, while limiting the loss rate due to buffering at the router. In practical scenarios, the relay provides limited feedback to the nodes, and

we consider that this information is only in the form of ack/nacks, when the packet was accepted/rejected.

The goal of the second part of this thesis is to design efficient algorithms for determining the access rate u_i of the nodes transmitting data, based on limited feedback (partial information) about the state of the relay's buffer. Namely, this part of the thesis is concerned with the theory and simulation of partially observed systems, in particular of information sources accessing a commonly shared resource. Our main contributions are:

- We state an original problem of multiple access queues with random Bernoulli arrival rates.
- We derive a provably optimal flow control algorithm that maximizes average throughput under loss constraints, and which is based only on partial information available from the relay node.
- We prove convergence results on the long time performance of transmitters employing such a control policy, namely we prove the existence of a limit measure over the space of *information state*[3] vectors. As a result, we are able to prove convergence in probability of significant performance parameters such as throughput and loss rate.
- Furthermore, we prove the uniqueness of this limit measure, and we show a method to construct it.
- We illustrate all our theoretical insights with numerical experiments, and we show how our control policy can be implemented in practice.
- We show how our results may be applied to designing efficient and stable algorithms for MAC access, in particular for sensor networks.
- We derive a provably stable implementation of the slotted Aloha protocol.

1.5 Related Work

1.5.1 Data Gathering in Sensor Networks

Recently, there has been a strong interest in the research community focused on designing power efficient algorithms for several common tasks in sensor networks. These studies are directly related to the problem addressed in the first part of this thesis. Thorough discussions of the main issues specific to sensor networks, and interesting research topics in this area are described in state-of-the-art surveys like [2], [54], [57].

When all nodes need to communicate among each other (like for instance in an ad-hoc network), it is of great interest to solve the general problem of finding minimum energy topologies. Routing in ad-hoc networks has been addressed in [5], [23], [42],

[3]The information state is an entity which comprises all available information about the network that is contained in the sequence of controls and feedback.

[52], however these works do not take the power issue specifically into account. Recent algorithms proposed for power efficient routing can be found in [9], [21], [59], [64]. The work in [62] uses the *wireless multicast advantage*. All nodes within some transmission range receive a transmission without additional power expenditure. This results in Voronoi cells for energy efficient communication. Minimum energy trees are built with distributed algorithms restricted to the topology previously determined. The algorithm needs information on the nodes location and the deployment region. In [68], the deployment region is equivalent with the maximum power, but the power law is assumed to be unknown. The result is a structure where the average node degree is small. Then the authors use the multihop advantage to further reduce the number of neighbors. All these algorithms require connectivity of the network graph. The relation between the wireless range and the connectivity of the graph is studied in [18].

In [58], the authors introduce the cluster based LEACH algorithm. In their model, the base station is located far from the network. Sensors elect themselves randomly as cluster heads (probability depending on the remaining power), and form Voronoi type of clusters. Then the cluster heads compress data arriving from nodes that belong to the respective cluster and send an aggregated equal size packet to the base station. The work in [43] introduces the PEGASIS algorithm, that uses the energy $\times$ delay metric. Their algorithm find chains instead of clusters, and assume equally sized fused data packets. The procedure induces high delays, so the energy $\times$ delay is a good metric to use. They provide solutions for the TDMA and CDMA contention cases by proposing hierarchical versions of their algorithm. The delay constraint in communication trees is formalized in [63], where the problem is shown to be NP-complete and the authors introduce a heuristic for finding the delay-constrained minimum spanning tree.

The work which is most related to the problem we consider in this thesis is the concept of directed diffusion and data aggregation [32]. In that model, sensors measure events characterized by an [attribute, value] set of pairs; this creates gradients of information in the respective neighborhood. The base station requests data by broadcasting *interests*. Also, it is assumed that messages are exchanged only between neighbors. When interests fit gradients, paths of information flow are formed, then the best paths are reinforced. Gradients of events may originate from different locations, when redundant data are measured at different places. In order to reduce communication costs, data may be aggregated on the way. Early aggregation (close to the origin of information) increases the loss probability, since if the packet is lost no information at all will reach the base station. On the other hand, late aggregation (close to the base station) requires more power, as packets originating from different sources will use different paths. The other nodes can be used as relays (this translates into a *Steiner tree* problem). The main question in this setting is from which sensor to take a particular piece of information; this is a version of the computationally hard set covering problem. The goal is to find a good aggregation tree, that gets the data from some of the nodes to the base station. The method implemented is a greedy incremental algorithm: find one shortest path to a particular source node, and incrementally add nodes with minimum cost to the tree. Related work can be found in [37], [26].

In [26] the authors address the problem of data gathering and compression at

relay nodes by using the theory of concave costs applied to single source aggregation. The authors present a thorough background of existing work and develop an elegant algorithm that finds good trees that simultaneously maximize several concave cost functions. Their model is different from ours in the sense that their setting assumes information sources are supplied a constant amount of information regardless of the topology of the gathering tree. Moreover, nodes do not collaborate to exploit the correlation. In our case the amount of aggregated information out of a particular node sent down the tree to the base station depends on the structure of the subtree which has that source as 'parent', due to the correlation structure.

General bounds on the performance of wireless sensor networks have been found in [25], [28]. Progress towards practical implementation of Slepian-Wolf coding has been achieved in [1], [55], [56]. Bounds on the performance of networks measuring correlated data have been derived in [48], [64]. However, none of these works takes into consideration the cost of transmitting the data over the links and the additional constraints that are imposed on the rate allocation by the joint treatment of source coding and transmission.

1.5.2 Design, Stability and Performance of Medium Access Control

Related work to the problem we study in the second part of this thesis includes the various extensive studies of randomized medium access algorithms (e.g. Aloha [7]). Our work on the analysis of feedback controlled queues was, to a large extent, triggered by the work of Anantharam and Verdú, on the Shannon capacity of a simple queuing model [3]. The main question studied in that work is that of how many bits of information can flow across a queue. We see our study as a step towards the study of the capacity of multiple access queues.

The need of a unified theory of control and information in the case of dynamic systems is underlined in the overview of [50], where the author discusses topics related to the control of systems with limited information. These issues are discussed in the context of several examples (stabilizing a single-input LTI unstable system, quantization in a distributed control two stage setting, and LQG), where improvements in the considered cost functions can be obtained by considering information and control together, namely by "measuring information upon its effect on performance". Extensive work along these lines is presented in [67], where the author derived techniques which consider the use of partial information, for capacity optimization of Markov sources and channels, formulated as dynamic programming problems.

There are some important results in the literature dealing with related results on convergence in distribution, in which the state of a system can only be inferred from partial observations. Kaijser proved convergence in distribution of the information state for finite state ergodic Markov chains, for the case when the chain transition matrix and the function which links the partial observation with the original Markov chain (the observation function) satisfy some mild conditions [35]. Kaijser's results were used by Goldsmith and Varaiya, in the context of finite state Markov channels [27]. This convergence result is obtained as a step in computing the Shannon capacity of finite state Markov channels, and it holds under the assumption of i.i.d. inputs; this assumption is crucial to the proof presented there, a key step of that proof is shown in the paper to break down for an example of Markov inputs. This

assumption however is removed in a recent work of Sharma and Singh [65], where it is shown that for convergence in distribution the inputs need not be i.i.d., but in turn the pair [channel input, channel state] should be drawn from an irreducible, aperiodic and ergodic Markov chain. Their convergence result is proved using the more general theory of regenerative processes. However, none of these works address the case of control of queues with feedback based on partial information.

1.6 Thesis Outline

The first part of this thesis is concerned with the study of total flow cost in networks measuring correlated data. We show that, in general, the flow problems for such networks involve both the tasks of supply allocation at nodes, and of finding good transmission structures. We show that a separate treatment of these tasks can lead to highly sub-optimal solutions. Thus, an important goal is to solve the joint problem of rate allocation and transmission structure optimization for such networks. For our analysis, we consider a family of cost function that is very important in various practical scenarios, namely the [F(rate on link)] $\times$ [link weight] metric. We study the two possible paradigms that are met in such cases, namely Slepian-Wolf coding and coding by using explicit communication. The main setting that we analyze is data gathering, namely the case when there is a single sink to which data from all the other nodes needs to arrive. We show also generalizations of our results to arbitrary traffic matrix problems.

In Chapter 2, we study the problem of rate allocation with explicit communication, namely when local compression of data at a node is based only on data received by that node from other nodes that use it as a relay. For such a setting, no a-priori local knowledge at nodes is needed about the correlation structure among nodes, as this knowledge is only estimated from the data that is received from neighbor nodes. Rate allocation is easy to be done in such a scenario, since the compression at a node is directly dependent on the data from nodes that use it as a relay. However, we prove that the rate allocation is directly dependent on the chosen transmission structure, so the two tasks cannot be separated. Furthermore, we prove that finding an optimal transmission structure is NP-complete even for very simplifying assumptions. We also propose some distributed approximation algorithms that perform well.

In Chapter 3, we study the problem of rate allocation with Slepian-Wolf coding, namely when joint compression of data at nodes can be done even without nodes communicating among them. This approach assumes that the correlation structure is known a-priori. In such a case, we prove that a well-studied transmission structure, namely the shortest path tree, is optimal for any rate allocation. Thus, the tasks of rate allocation and transmission structure optimization separate in this case. Namely, first an optimal transmission structure is determined. Second, the rate allocation is found as the solution of an optimization problem that has the weights of the shortest path tree as coefficients. We show how the problem can be solved numerically with the help of Lagrange multipliers, for the case of a general separable cost function, and we concentrate our study on linear cost functions. If the cost function is linear, the optimization problem reduces to an LP problem. We assess the difficulty of the problem in the multiple sink case, and we study in more detail the data gathering problem with a single sink. Namely, we find the solution of this LP problem in a closed form

in the case of data gathering. We present a decentralized approximation algorithm that is based only on local information available at nodes from their neighborhood.

Further, we study the asymptotic behavior of the ratio of costs involved by the two coding approaches, for large networks. We end this chapter with a complexity study of a mixed approach that uses Slepian-Wolf coding on clusters of nodes, and coding by explicit communication among clusters.

The second part of this thesis is dedicated to the study of queues under feedback control. Namely, our goal is to design a decentralized algorithm, based on only partial information about the state of system, that controls the access of several concurrent data sources to a limited buffer queue.

In Chapter 4 we present a system that models a queue in a multiple access regime. Each user accesses the queue with a controllable injection rate that is modeled by a Bernoulli random variable. The users control their rate of access by means of only limited feedback from the queue, namely acknowledgments of whether the injected packet had or had not been accepted in the queue's buffer. This feedback is private, that is the queue sends acknowledgments about a packet only to the source that generated that packet. Moreover, the users have no knowledge about the number of other users accessing the queue simultaneously. We only make the reasonable assumption that the number of active sources forms a Markov chain, and the transition matrix of this Markov chain can be estimated or computed a-priori. The goal is to maximize the throughput of the source, under loss constraints. We design a provably optimal control algorithm based on the information state. The information state can be computed locally in a recursive manner at each node, and it fully contains the information about the network that can be assessed given the sequence of previous controls and acknowledgments from the queue, available at a node. We present a set of numerical simulations that show that our algorithm provides good performance in terms of throughput while maintaining fairness among the set of active sources.

In Chapter 5 we prove that the information state converges in the limit, under some mild conditions. This implies that important quantities related to the performance of the system, like the average throughput and average loss rate, converge in the limit as well. This also shows that the control algorithm that we propose is stable. We base our proof on the theory of Markov chains defined on infinite state-spaces. We first prove the existence and uniqueness of our sought limit distribution of probability over the space of information states. Then, by using discretization techniques, we show how the sought distribution can be constructed and approximated for practical use. We show numerical experiments about the stability of the average throughput and average loss rate that confirm our theory. We also apply our control design policy to the Aloha protocol.

Finally, in Chapter 6 we present our conclusions and ideas of further work.

Chapter 2

Network Correlated Data Gathering with Explicit Communication

2.1 The Problem and its Motivation

Consider a number of distributed data sources with a certain correlation structure among the sources. A number of links connects sources to each other, establishing a graph where sources are nodes and links are edges. The task is to send all the data to a particular node of the graph, and we refer to this set up as the *correlated data gathering on a graph* problem. The goal is to achieve the data gathering while minimizing a cost functional (e.g. power on links or distance accumulated), possibly under constraints (e.g. some maximal capacities on links). This is of course an instance of a network flow problem, but with an original twist: because the data is correlated, standard solutions may not be optimal, leading to original rate allocation problems and original tree building problems, depending on the source coding model that will be used.

An example is shown in Figure 2.1, where we have N nodes with sources $X_1, \ldots, X_N$, a sink S, and a number of edges that connect the sources. Nodes are supplied amounts of measured data which need to be transmitted to end sites, called base stations or sinks. Intermediate nodes can be also used as relays in addition to measuring data. They aggregate their own data with the data received from other nodes. A very important task in this scenario is to find a transmission structure on the network graph that minimizes a cost of interest (e.g. flow cost [rate] $\times$ [link weight], distance, etc.). In the case of a single sink, the optimal structure is usually a tree. This leads to the question of how to construct efficient data gathering trees.

When the data measured at nodes are statistically independent, the problem separates: the rate allocation becomes trivial, as each node codes its data independently; then next, well developed algorithms can be used to solve various network problems (minimum and shortest path spanning tree, minimum cost flow etc.).

However, in many situations, data at nodes are *not* independent. Due to the cor-

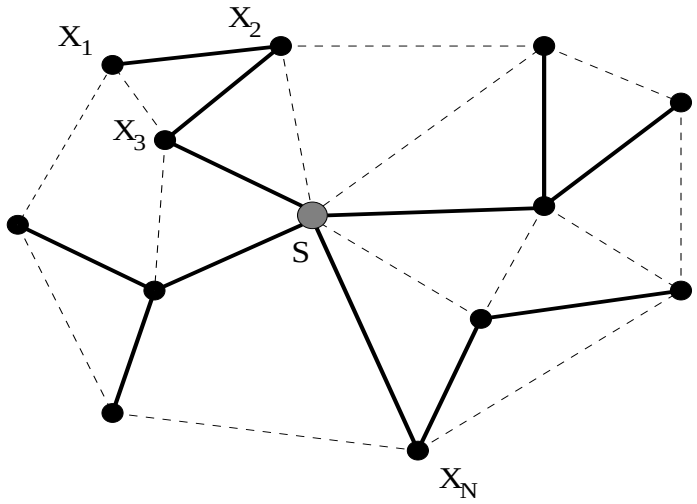


Figure 2.1: In this example, data from nodes $X_1, X_2, \ldots, X_N$ need to arrive at sink S. A rate supply R_i is allocated to each source X_i. In thick solid lines, a chosen transmission structure is shown. In thin dashed lines, the other possible links are shown.

relation present, it is expected that approaches that take into account this correlation, will outperform traditional approaches, for various cost functions (metrics). Moreover, jointly exploiting the data structure and optimizing the transmission topology (structure) in the network can provide substantial further improvements. Therefore, it is worth studying the interaction between the correlation of the data measured at nodes and the transmission structure that is used to transport these data.

A practical instance can be found in sensor networks [2], [51], [54]: a number of sensors acquire measurements which are typically correlated to each other, and these measurements are sent to a base station for evaluation or decision purposes. Collecting images from various sources into a common repository on the internet is another example of correlated data gathering.

In particular, let us consider the case of a network of sensors taking measurements from the environment. Let $\mathbf{X} = (X_1, \ldots, X_N)$ be the vector formed by the random variables measured at the nodes $1, \ldots, N$. We assume that the random variables are continuous and that there is a quantizer in each sensor (with the same resolution for all sensors)[1]. A rate allocation $(R_1, \ldots, R_N)$ (bits) has to be assigned at the nodes so that the quantized measured information samples are described losslessly. That information has to be transmitted through the links of the network to the designated sink (see Figure 2.1).

We make a set of additional assumptions for the case of a sensor network. We assume that the multiple access is controlled by an appropriate ideal protocol, and thus interference and collisions are solved by the upper layers. Since nodes have limited processing capability and/or energy, it is necessary that the rate allocation and

[1]Note that this procedure is sub-optimal; an optimal solution would have to resort to vector quantization, however this is prohibitively complex for the case of sensor networks.

transmission structure optimization are done locally at each node, in a decentralized manner, by using information available only from nodes in the neighborhood. We abstract the wireless case as a graph with point-to-point links, mainly because the full wireless multi-point case is an open problem, and practical schemes transform the problem into a graph with nearest neighbor connectivity.

As battery power is the scarce resource for autonomous sensors, a meaningful metric to minimize is the energy consumption, which is essentially given by the sum of products [rate] $\times$ [link weight], for all the links used in the transmission. Here the weight of the link between two nodes is a function of the distance d of the two nodes (e.g. kd^{α} or $k\exp(\alpha d)$, with $k > 0$, $\alpha > 0$ constants that depend on the medium properties).

We will consider two complementary approaches that may be used in this problem. In the first approach, nodes can exploit the data correlation only by receiving explicit side information from other nodes (for example, when other nodes use a node as relay, their data is locally available at that relaying node). Thus, the correlation structure is exploited through communication and joint aggregate coding/decoding locally at each node. In this case data coding is easy and relies only on locally available data as side information. We call this approach coding by *explicit communication*. However, we will see that optimizing the routing structure becomes complex.

The second approach is to allow nodes to use joint coding of correlated data without explicit communication (this is possible with binning arguments, that is using Slepian-Wolf coding [10], [55], [56], [66]). With this approach, finding a good routing structure turns out to be easy, because routing and coding are separated (decoupled); however data coding becomes complex and global network knowledge is needed for an optimal solution.

The main tradeoff between these two settings is:

- If the correlation structure is not known a-priori, then side information is needed to reduce the entropy. In other words, the correlation structure is learned (explicitly) in a distributed manner through explicit communication. This leads to a simple source coding, but the transmission structure optimization is hard.

- If nodes are assumed to know the correlation structure (or equivalently, the dependence on the distance or other prior dependencies), then they can employ Slepian-Wolf coding. In this case, source coding is complex while the transmission structure can be found with classical polynomial time network flow algorithms.

In this work, we show how the rate allocation is dependent on the chosen setting. In the Slepian-Wolf case, optimal coding allocates most of the load to nodes close to the sink and small rates to nodes at the extremity of the network. In the explicit communication model, data compression is done only when side data is available, and thus large rates are allocated to nodes far from the sink, and much smaller rates at nodes close to the sink. It is therefore expected that a combined third approach will provide not only a complexity tradeoff between the source coding and the transmission structure strategy, but will also distribute more evenly the rate load throughout the network. One possible approach is to consider a combination of Slepian-Wolf coding for clusters of nodes, and explicit communication among clusters.

In this chapter, we study the case when joint coding of correlated data is performed with coding by explicit communication, that is, the reduction in rate by entropy coding due to the correlation is possible only when side information is explicitly available (as relayed data). We prove that this makes the problem NP-complete and propose distributed approximation algorithms with a good performance.

The treatment of the Slepian-Wolf coding approach is presented in the next Chapter 3. In that chapter, we also consider a particular case of the mixed approach, where Slepian-Wolf coding on clusters is used.

The rest of this chapter is organized as follows. In Section 2.2 we introduce the minimum cost network correlated data gathering tree problem. In Section 2.3 we formulate the explicit communication coding approach, we show a simplified system model that preserves the complexity of the original problem, and we discuss some simple network examples that show intuitively that the *SPT* might not be the optimal transmission structure. Further, we assess the difficulty of jointly optimizing the transmission structure and performing the rate allocation, namely we prove that the problem is NP-complete. Next, in Section 2.4 we propose some tree approximation algorithms expected to provide important improvements over *SPT*, and show numerical experiments that show that these gains are indeed obtained for random network instances.

2.2 Problem Formulation

We consider the problem of data gathering with a single sink, to which all the data has to be sent. Let $G = (V, E)$ be a weighted graph with $|V| = N + 1$. We denote by S the particular $(N + 1)$th node called sink. Except the sink, every node in the graph generates a source. Each edge $e = (i, j) \in E$ has a weight w_e. Since the data are correlated, depending on the chosen transmission structure, each node i has to transmit a certain rate R_i through the network to the sink. Let $f(x_e, w_e)$ be an arbitrary cost function of the total rate (flow) x_e going through a particular edge with weight w_e. Then the general *minimum cost data gathering tree* problem is defined as follows: find the spanning tree (ST) of the graph G that minimizes the cost function:

$$c_{ST} = \sum_{e \in ST} f(x_e, w_e), \tag{2.1}$$

under constraints

$$x_{i \to e} - \sum_{e \to i} x_e = R_i, \quad i = 1, \ldots, N; \qquad R_S = \sum_{i=1}^{N} R_i,$$

where we denote by $e \to i$ the set of edges entering node i, and by $i \to e$ the edge from node i to its parent in the tree. We restrict our discussion to functions $f(\cdot, \cdot)$ which are separable as the product of a function that depends only on the rate and another function that depends only on the link weights of the transmission structure[2]. Without loss of generality, we assume $f(x, w) = x \cdot w$. Then, the expression

[2]This corresponds to many practical settings (e.g. the [rate] · [path weight] cost function measures the transmission cost in wired networks, and the [exp(rate)] · [path weight] measures the battery

(2.1) to be minimized can be rewritten as:

$$c_{ST} = \sum_{i \in V} R_i d_{ST}(i, S) \tag{2.2}$$

where $d_{ST}(i, S)$ is the total weight of the path connecting node i to S on the ST tree.

The important new feature that makes this problem different from classical network flow theory is the following: by changing the transmission structure, since we change the inter-node distances, both the set of rates $\{R_i\}_{i=1}^N$, which depends on the inter-node correlation, and the path weights $\{d_{ST}(i)\}_{i=1}^N$ are affected. Thus, the optimization of the set of rates and the path weights has to be done jointly, and it cannot be decoupled. We call this new problem the *minimum cost correlated data gathering tree* problem.

We now particularize the optimization problem (2.2) to the explicit communication based coding setting. In classical network transport theory, the amount of supply (rate in our case) at a node is fixed and independent of the communication links that are chosen to transport the various supplies. In particular, the supply provided by the ith node is independent of the nodes that are connected to the ith node through the chosen edges. In our problem formulation, an important novelty is that the supply at a given node *depends* on the incoming flow from other nodes that use that node as relay, and also on the transmission structure that is used for these nodes.

Consider again the example in Figure 2.1, where nodes have to communicate their correlated data to one sink. To reduce the complexity of local coding, we assume that each relay node forwards received packets without decoding/re-coding received information and they only perform compression by conditional entropy coding of its own measured data, given the received data from the nodes that are using it as intermediate relay node. Denote by $H(X)$ the entropy[3] of a discrete random variable X, and by $H(X|Y)$ the conditional entropy of a random variable X given that the random variable Y is known. If we consider node X_3, then the rate it has to supply depends on whether:

1. Neither X_1 nor X_2 use X_3 as relay. In this case X_1 uses a rate $H(X_1)$, X_2 uses a rate $H(X_2)$, and X_3 uses a rate $H(X_3)$.

2. Node X_1 uses X_2 as relay, and X_2 transmits its aggregate further to X_3 (this case is shown with solid line in Figure 2.1). In this case X_1 uses a rate $H(X_1)$, X_2 uses a rate $H(X_2|X_1)$, and X_3 uses a rate $H(X_3|X_1, X_2)$.

3. Both X_1 and X_2 use X_3 as relay. In this case X_1 uses a rate $H(X_1)$, X_2 uses a rate $H(X_2)$, and X_3 uses a rate $H(X_3|X_1, X_2)$.

In the first case, no side information is available at node X_3 from other nodes. Thus, node X_3 sends its entire amount of data on a path to the sink. In the second case, node X_3 does have side information available from node X_2. The information at these two

consumption in wireless networks, where the [path weight] term is a function of the inter-node distances along a path).

[3]The entropy is a measure of uncertainty of a random variable [10]: $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$, where $\mathcal{X}$ is the discrete alphabet of X.

nodes is not independent. Therefore node X_3 can reduce correspondingly the amount of data it sends further. It jointly codes its data with the data from node X_2 and sends the resulting coded data further. In the third case, the amount of side information available is even larger at node X_3, since two nodes use it as relay. Thus, the data amount reduction at node X_3 is even larger because the conditional coding involves more sources.

It is clear that in either of these three cases, the optimal transmission structure might not be the shortest path tree. We show in the next Section 2.3 how the joint dependence of rates and path weights on the transmission structure actually makes our optimization problem NP-complete.

2.3 Complexity Analysis and Approximation Algorithms

For the sake of simplicity and clarity in our arguments, and without loss of generality in the complexity analysis, we use in this section a simplified model for the data correlation, which allows a clearer analysis of complexity, and for which we develop efficient heuristic approximation algorithms. As we show in this work, this model still completely preserves the original complexity of the optimization problem. Namely, in our model, data at each node are entropy coded with $H(X_i) = R$ bits if no side information is available from other nodes; but only $H(X_i|X_{j_1}, \ldots, X_{j_k}) = r \leq R$ bits, $\forall k$, are needed if the node i has side information available coming from at least another node, which uses node i as relay. Thus, our simplification is that r is constant and does not depend on the number of nodes on which conditioning is done. We denote by $\rho = 1 - r/R$ the correlation coefficient.

2.3.1 The Tradeoff between Shortest Path Tree and Traveling Salesman Path

In the case of uncorrelated data, if the cost for transmitting over an edge was proportional (by a fixed constant) to the Euclidean length of that edge, then the problem is trivial and the optimal communication structure is the edge connecting the node to the sink. However, for an arbitrary weight function on the edge, transmitting via relays may be better than direct transmission (for example, if the edge weight is d^α, $\alpha > 1$). In the case of correlated data, as it is the case in sensor networks, things become even more interesting, even for very simple networks, because the rates $\{R_i\}_{i=1}^N$ are affected by the choice of the transmission structure.

The example in Figure 2.2 shows that even in simple network cases, finding good correlated data gathering structures is not trivial at all. If the data were independent, the shortest path tree (*SPT*) would be optimal (see Figure 2.2 (a)). However, we see that in this example, if $\rho > 1/2$, the *SPT* is no longer optimal, since its cost is larger than the one corresponding to the gathering tree in Figure 2.2 (b).

Figure 2.3 shows one other simple network example, with N nodes equally-spaced on an unit length arc circle at distance D from the sink. It is straightforward to show that $\lim_{N\to\infty} \frac{c_{TSP}}{c_{SPT}} = (1-\rho)\left(\frac{1}{2D} + 1\right)$, where c_{TSP}, c_{SPT} are the total flow costs of the two corresponding trees. Consider the case when the number of nodes is very large and the correlation coefficient is arbitrarily close to unity. This means that a path passing through all the nodes and ending at the sink (a traveling salesman path, *TSP*)

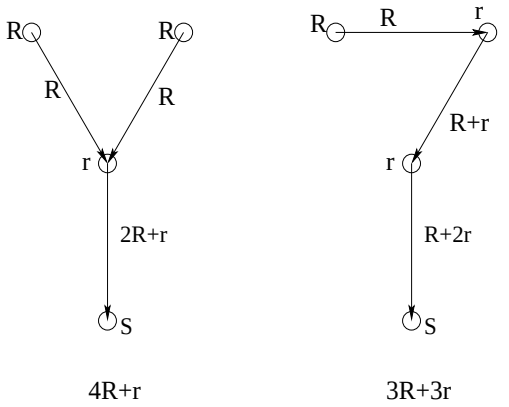


Figure 2.2: All edges have length 1. The TSP (b) outperforms the SPT (a) if $R > 2r$.

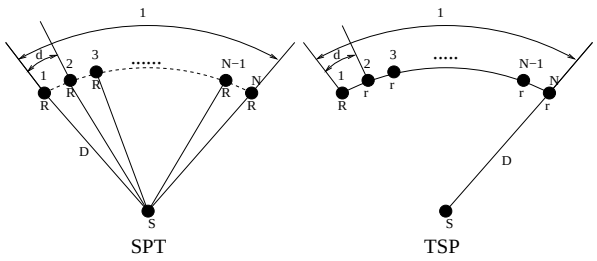


Figure 2.3: SPT vs. TSP.

can be arbitrarily more cost efficient than the direct transmission which corresponds to the SPT in this case.

From these simple examples, it can be seen that the correlated data gathering problem with explicit communication is actually a hard optimization problem, in general. Formally, in terms of graph optimization, we can rewrite the minimization of (2.2) for the case of explicit communication as follows:

- **Given:** graph (V, E).
- **Find:** the spanning tree $ST = \{L, T\}$ with L leaves, T non-terminal nodes, $L \cup T = V$, $L \cap T = \emptyset$.
- **such that:**

$$ST = \arg \min_{\{L,T\}} \left(r \sum_{t \in T} d_{ST}(t, S) + R \sum_{l \in L} d_{ST}(l, S) \right)$$

where $d_{ST}(i, S)$ is the total weight of the path on the ST tree from node i to the sink S.

In terms of the correlation coefficient, $\rho = 1 - r/R$:

$$ST = \arg\min_{L} \left((1-\rho) \sum_{i \in V} d_{ST}(i,S) + \rho \sum_{l \in L} d_{ST}(l,S) \right) \tag{2.3}$$

Let us first look at the two extreme cases, that is $\rho \to 0$ and $\rho \to 1$. When $\rho \to 0$ (independent data), the optimal tree is the *SPT*, which is known to be solvable in polynomial time by e.g. a distributed Bellman-Ford algorithm. At the other extreme, when $\rho \to 1$ (data maximally correlated), the optimal solution is a spanning tree for which the sum of paths from the leaves to the sink is minimum. For this, the core information is taken from the leaf nodes, and passing through all the in-tree nodes only adds an infinitesimally small amount of new information, since data is strongly correlated. It is straightforward to show that solving this problem is equivalent to solving the *multiple traveling salesman* optimization problem $(k - TSP)$[41], which is known to be NP-hard.

To the best of our knowledge, (2.3) is an original spanning tree optimization problem on a graph. In Section 2.3.2, we show that this problem is also NP-hard for the general case $0 < \rho \leq 1$. However, it is possible to design good approximation algorithms and we provide them in Section 2.4.

2.3.2 NP-Completeness

In order to prove the NP-hardness of the optimization problem given in (2.3), we show that the decision version of the problem is NP-complete. The decision version of our optimization problem is:

Definition 2.1 *Network data gathering tree cost decision problem.*

- Instance*: An undirected graph $G = (V, E)$ with weights w_e assigned to the edges $e \in E$, a positive integer M, and a particular node $S \in V$.*
- Question*: Does the graph admit a spanning tree ST such that, when assigning supplies $R_i = R$ to the leaf nodes and $R_i = r < R$ to the in-tree nodes in the spanning tree ST, the total cost of ST given by (2.3) is at most M?*

Theorem 2.1 (NP-completeness) *There is no polynomial time algorithm that solves the network data gathering tree cost problem, unless P=NP.*

Proof: See Appendix 2.A (we use a non-trivial reduction from the min-set cover problem).

Since this problem is a particular version of the general problem given in (2.3), it follows by a trivial reduction that the general problem is also NP-hard.

Corollary 2.1 *Minimizing $\sum_{i \in V} R_i d_{ST}(i,S)$ with $R_i = f(\sum_{e \to i} x_e)$ for an arbitrary monotonic function $f(\cdot)$, is NP-hard.*

Note also that, in general, node i has information from all the nodes in the subtree $sbt(i)$ rooted at node i. Our simplified model is a particular case of this general entropy coding problem, where $H(X_i|\{X_j\}, j \in sbt(i))$ is approximated with $H(X_i|X_j)$, with

j being a child of i. Then it can be shown easily that the NP-complexity of our simplified example extends also to this more general case by means of a trivial further reduction.

Note that the NP-hardness of the problem for a single sink generalizes by a straightforward reduction to the case of multiple sinks. However, the derivation of approximation algorithms for the multiple sinks case is significantly more difficult. For instance, the generalization of the *SPT/TSP* structure to multiple sinks is non-trivial due to the interactions between the approximated structures derived for each single sink in particular. This is because nodes that are leaves for a particular structure can be in-tree nodes for other structures, and thus their corresponding rate allocation cannot be uniquely determined. The study of approximation algorithms for the multiple sink case is a subject of our current research.

Arbitrary function of rate

Note that the NP-completeness result holds for any function of the rate, since an arbitrary function only modifies the values of R and r, but not the multiplicative separable form of the cost function.

2.4 Heuristic Approximation Algorithms

In this section we introduce a set of approximation algorithms for solving problem (2.3).

2.4.1 Shortest path tree

SPT is computed by using the distributed Bellman-Ford algorithm for simultaneously determining the shortest paths from all nodes to the sink. If the data is independent, this is the optimum solution, but it is far from optimal if there are high correlations.

Algorithm 2.1 *Shortest Path Tree:*

- *Initialize* $D_i = \infty$ *and parent nodes with* $par(i) = S, i = 1, \ldots, N$.
- **While** $par(i)$ *changes for some* i*:*
  - $D_i = \min(D_j + d_{ij}), j = \{1 \ldots N\} \backslash \{i\}$.
  - $par(i) = \arg\min_j (D_j + d_{ij}), j = \{1 \ldots N\} \backslash \{i\}$.
- **Endwhile.**

Note that in the distributed version of the algorithm, the search for $par(i)$ can be done only over a neighborhood of node i (given by the transmission range, or the k-nearest nodes neighborhood).

2.4.2 Greedy algorithm

We start from an initial subtree composed only of the sink node. Then, we add successively, to the existing subtree, the node whose addition results in the minimum cost increment.

Algorithm 2.2 *Greedy algorithm:*

- *Let V_{ST} denote the nodes in ST. $V_{ST} = \{S\}$. Let $V_G = V \backslash V_{ST}$.*
- **While:** $V_G \neq \emptyset$
  - **Find:** $\{j_0, i_0\} =$
    $= \arg\min_{\{j \in V_G, i \in V_{ST}\}} (R(d_{j,i} + d_{ST}(i,S)) + (r-R)isLeaf(i)d_{ST}(i,S))$
  - $ST = ST \cup (i_0, j_0), V_{ST} = V_{ST} \cup \{j_0\}, V_G = V_G \backslash \{j_0\}$.
- **Endwhile.**

As expected, given the relationships between the problem in this paper and the TSP problem, greedy algorithms perform suboptimally (as we show experimentally in Section 2.4.6), in the same way as the greedy approximation algorithm for TSP provides a quite suboptimal solution. The reason is that far nodes are being left out, so they need to connect to the sink via a path with large weight.

2.4.3 Simulated annealing

We propose now a computationally heavy method which is known to provide results that are close to optimal for combinatorial problems involving a large number of variables, similar to the problem considered in this paper (e.g. *TSP*). This method was inspired by the *fitness landscape* concept used in evolutionary biology, physics of disordered systems and combinatorial optimization [60]. Its ingredients are: (a) a configuration space Z (finite set of possible *types*), (b) a move set Z (set of adjacencies among types), and (c) a fitness function $f : Z \rightarrow \Re$ (value assigned to each type). The goal is to optimize the fitness over the configuration space.

A very general heuristic optimization method is simulated annealing (SA) [30]. It is based on stochastically simulating the slow cooling of a system and it is closely related to the Gibbs field transition structures. It gives very good results when applied to another NP-hard combinatorial problem in graphs, the traveling salesman problem (TSP) [41, 60].

The fitness landscape formulation [60] of our problem is as follows: (a) the configuration space is the set of all spanning trees (completely defined by the parent relationship), (b) the move set is: one node changes its parent, (c) the fitness function is $g(ST) = R\sum_{l \in L} d_{ST}(l,S) + r\sum_{t \in T} d_{ST}(t,S)$. Our goal is to minimize the fitness over the set of spanning trees.

Algorithm 2.3 *Simulated annealing:*

- *Take a cooling schedule $T[k], k = 1, \ldots, K$.*
- *Initialize parent nodes with $par(i) = S, i = 1, \ldots, N$. Denote by $\mathcal{N}(i)$ the set of one-hop neighbors of i.*

- **While** $k < K$
  - $k = k + 1$, $l = g(ST)$;
  - *choose* $i, j \in \mathcal{N}(i)$, *at random such that deleting edge* $(i, par(i))$, *and adding edge* (i, j) *to the tree, does not form a cycle; let* ST' *be the newly generated spanning tree and let* $l' = g(ST')$ *be its corresponding fitness.*
  - *make the change* $par(i) \leftarrow j$, *and assign* $ST \leftarrow ST'$ *with probability*

$$p = \begin{cases} 1, & \text{if} \quad l' \leq l \\ \exp(-\frac{l'-l}{T[k]}), & \text{if} \quad l' > l \end{cases} .$$

- **Endwhile.**

The main feature of this method is that it avoids getting stuck at a local minimum. With a correctly chosen cooling schedule and if the algorithm runs for a sufficient number of steps, the final version of the spanning tree ST is very close to optimal. The convergence of the method depends on the *ruggedness* and *neutrality* of the fitness landscape [60]. For $\rho = 0$ (SPT), our experiments show that it does provide the exact solution, and convergence is easy to achieve. When ρ is close to 1, the generated landscape is not smooth any longer, so convergence is difficult to obtain in a reasonable number of iterations. We obtained good results (iteration steps vs. ruggedness) with the Lundy and Mees schedule [47]:

$$T_k = \frac{T_{k-1}}{1 + \frac{T_0 - T_K}{K T_0 T_K}}.$$

A provably optimal, but slow schedule for T_k is $c/\log(1+k)$ [30], with c the maximum local minimum depth of the landscape.

However, in general, simulated annealing is usually hard to implement in a decentralized manner, and is computationally expensive. It does however provide a good benchmark close to optimal against which other heuristic algorithms can be tested.

2.4.4 Balanced SPT / TSP tree

We propose a heuristic approximation algorithm consisting of a combination of SPT and $k - TSP$, from the solutions obtained using simulated annealing. The solution provided by this algorithm consists of a SPT structure around the sink that has a certain radius and a set of TSP paths starting from each of the leaves of the SPT. Depending on the amount of correlation, that is the value of ρ, a certain radius for the SPT is more appropriate. We briefly describe the intuition why there is such a value for this radius. Since the leaf nodes contribute most to the cost ($R > r$), then in order to minimize the cost, the flows of R data coming from the leaves of the tree have to travel short paths to the sink (the SPT effect), but in the same time through as many nodes as possible (the TSP effect). On the other hand, when the correlation is large (r is small), the effect of transporting flows of r data through the tree is negligible, so it is essential to have as many in-tree nodes as possible, thus the TSP effect is more important, whereas when the correlation is small (r is large), it is more important that the data from in-tree nodes reach the sink on shortest paths, and thus the SPT effect becomes more pronounced.

Algorithm 2.4 *SPT/TSP balanced tree:*

- *Build the SPT for the nodes that are in a radius $q(\rho)$ from the sink. Denote this SPT by ST. The optimal choice for the radius $q(\rho)$ decreases with the increase of the correlation coefficient ρ.*
- *Let V_{ST} denote the nodes in ST. Let $V_{TS} = V \backslash V_{ST}$.*
- **While** $V_{TS} \neq \emptyset$
  - *Denote by L the set of leaves of ST.*
  - $\{i_0, l_0\} = \arg\min_{\{i \in L, l \in V_{TS}\}} (d(l,i) + d_{ST}(i,S))$.
  - $ST = ST \cup (i_0, l_0), V_{ST} = V_{ST} \cup \{i_0\}, V_{TS} = V_{TS} \backslash \{i_0\}$.

This is actually a suboptimal nearest neighbor approximation of the $k - TSP$, which is easily implementable in a distributed manner.

Square grid network graph: optimal radius for the SPT/TSP algorithm

Since the TSP problem is NP-complete, it is difficult to provide an analytical study of the dependence of the optimal SPT radius on the correlation structure for a general connectivity graph. Therefore, we restrict our attention to a square grid graph and study in detail the structure of our SPT/TSP algorithm in this case. Namely, we study the dependence of the optimal radius on the correlation coefficient ρ for this graph.

Consider a square grid network with $(2n+1) \times (2n+1)$ nodes (see Figure 2.4). The SPT is build on the square area of $(2m+1) \times (2m+1)$ nodes around the sink. Note that the SPT subtree has $8m$ leaves. For the rest of the graph, equal length TSP paths are built. Namely, for each leaf of the SPT subtree, a TSP rooted at that leaf node is constructed, which spans $\texttt{floor}\left((2n+1)^2 - (2m+1)^2/(8m)\right)$ of the nodes left outside the SPT subtree.

We plot in Figure 2.5 the total cost of the SPT/TSP tree as a function of the correlation coefficient ρ. We note that, as expected, the optimal SPT radius m decreases with the increase of the correlation coefficient ρ.

Next, we compute analytically the optimal 'radius' m/n of the SPT subtree as a function of the correlation coefficient $\rho = 1 - r/R$. After some computations, we obtain that the optimal m is a root of the following polynomial: $P(Z) = 3rZ^4 + (8r - 16R)Z^3 + (-r - 4rn + 4R - 4rn^2)Z^2 + (rn^4 + 2rn^3 + rn^2)$. This polynomial has 4 roots, but by solving it numerically, we find that only one of them is in the interval $[0, n]$. We plot this solution for the optimal radius in Figure 2.6. The discontinuity at $\rho = 0$ is due to the properties of the very particular regular grid structure that is analyzed. A particular interesting abrupt phenomenon is observed asymptotically: when n is sufficiently large, there is an optimal normalized radius for the SPT, which does not depend on the correlation coefficient ρ.

2.4.5 Leaves deletion approximation

This algorithm is a simplified version of the TSP/SPT algorithm. Namely, this algorithm constructs first the global SPT, and then uses one-hop TSP paths from the

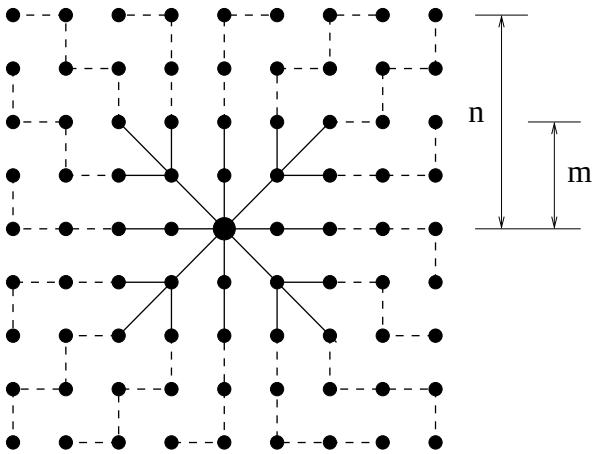


Figure 2.4: Square grid network: the SPT (solid lines) is built on the nodes in the $m \times m$ sub-grid around the sink (larger black dot). The rest of the nodes are spanned by TSPs (dashed lines) rooted in the leaves of the SPT.

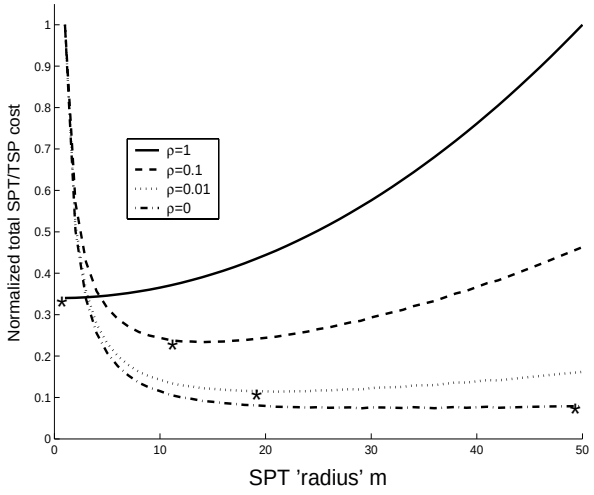


Figure 2.5: Squared grid network: normalized cost of the SPT/TSP tree for a grid network of size $N = 101 \times 101$ nodes ($n = 50$) and several values of the correlation coefficient ρ. Note how the optimum value of the radius m increases from 0 to n as ρ decreases from 1 (high correlation) to 0 (no correlation).

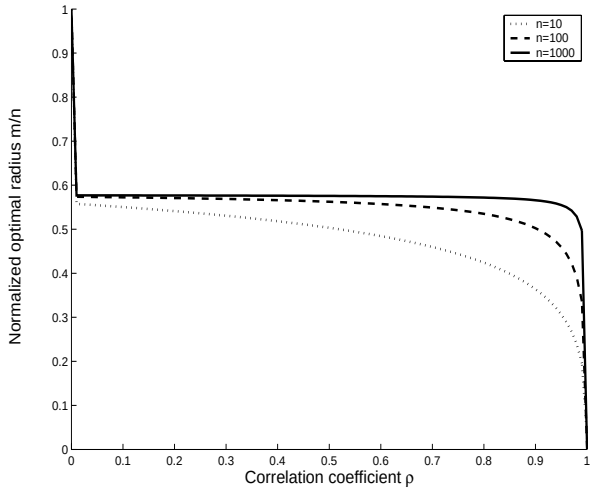


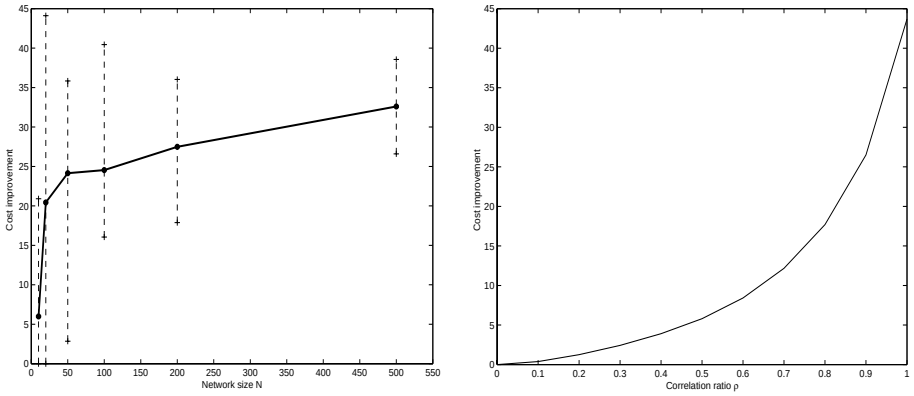
Figure 2.6: Square grid network: optimal radius of the SPT (normalized with respect to the radius of the square grid), as a function of the correlation coefficient ρ, for various sizes $N = (2n+1)^2$ of the network.

outer nodes of the SPT. It is based on the observation that good cost improvements may be obtained mainly by making the leaf nodes change their parent node to some other leaf node in their neighborhood. This operation is done only if it reduces the total cost of the whole tree.

Algorithm 2.5 *Leaves deletion algorithm (LD):*

- *Initialize* $ST \leftarrow SPT$. *Each node* i *maintains its parent, number of children, and total distance* $d_{ST}(i,S)$ *on the current spanning tree to the sink. Let* $par(i)$ *denote the parent node of node* i.
- **While** *there is a decrease in cost:*
  - **For each** *leaf node* i*: Find the leaf node* $j \in \mathcal{N}(i)$ *that maximizes* $R(d_{ST}(i,S)+ d_{ST}(j,S)) - (R(d_{i,j} + d_{ST}(j,S)) + rd_{ST}(j,S)) - I(i)$, *where* $I(i)$ *is an adjustment term indicating the cost lost by transforming single parent nodes into leaves. If the maximizing quantity is positive, then assign* $par(i) \leftarrow j$ *and update the corresponding distances on the tree to the sink, and number of children, for all the three nodes involved* $\{i$, *former* $par(i)$, $j\}$.
- **Endwhile**.

This algorithm involves a small number of iterations after SPT is computed, and is fully distributed. Note that a known good approximation for the geometric TSP is to start from the minimum spanning tree (MST) and eliminate the leaves by successively passing the traveling salesman path through them. Here, we can see that a simplified similar procedure provides good results in our case as well, which confirms the link between our problem and the TSP.



(a) Cost improvement (in %) vs. N. (b) Cost improvement (in %) vs. ρ.

Figure 2.7: Average total cost decrease $(100 \cdot (\frac{c_{SPT}}{c_{LD}} - 1)$, in %) of leaves deletion (LD) over shortest path tree (SPT) for (a) $\rho = 0.9$ and $N = 10, 20, 50, 100, 200, 500$, and (b) $N = 200$ and $\rho = 0, 0.1, \ldots, 1$.

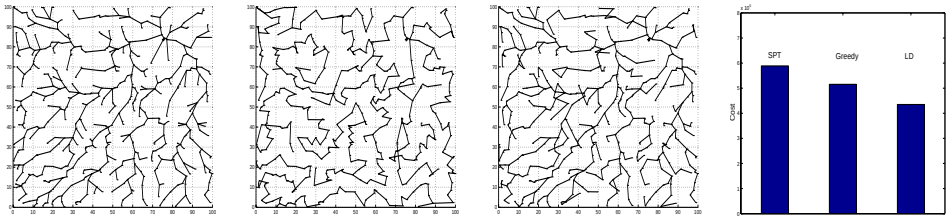
2.4.6 Numerical Simulations

Our simulations were done in MATLAB for a network of nodes randomly distributed on a 100×100 grid, with a value $\alpha = 2$ for the power of the distance. We consider several sizes of the network, from $N = 10$ up to $N = 500$ nodes, and various values for the correlation coefficient ρ among the nodes, within the interval $\rho \in [0, 1]$. As mentioned before, the algorithm that is used for finding the SPT in a distributed manner is a distributed version of the Bellman-Ford algorithm, which runs in $\mathcal{O}(N|E|)$ steps. The actual speed of convergence depends on the degree of each node in the graph, which in turn depends on the range $\mathcal{N}(i)$ over which nodes search for neighbors. For the graph structures we consider, Bellman-Ford runs in an average of 50 steps for a network size of 500 nodes.

Our experiments show important average improvements of the LD algorithm over the SPT for nodes randomly distributed on a 100×100 grid (see Figures 2.7–2.10). The computational load of LD is small, namely at most 4 iteration steps after the SPT are required for its implementation, while the algorithm is still distributed. It also outperforms the greedy algorithm (see Figure 2.8). In this experiments, the sink is located at the center of the grid. Similar performances are obtained when the sink is situated outside the network area.

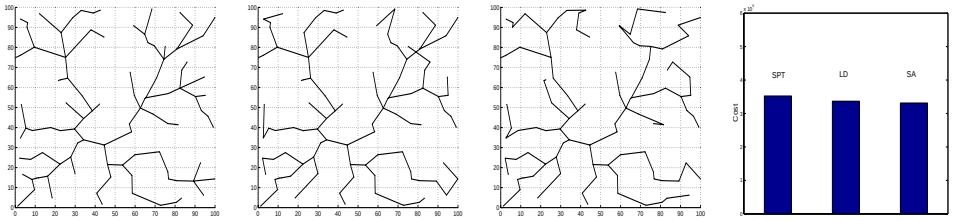
It can be seen that some clearly non-optimal patterns appear on the gathering tree solutions obtained using our heuristic algorithms. This is due to the fact that there are cases when leaf nodes do not have other close leaves on the graph, unless they choose leaves for which the corresponding connecting edge crosses over some already existing edges.

When comparing the various heuristic algorithms with the simulated annealing solution, which is expected to provide results close to optimal, we notice that our simple heuristic algorithms perform relatively well, while being completely distributed, scal-



(a) SPT algorithm (b) Greedy algorithm (c) Leaves deletion algorithm (d) Total cost

Figure 2.8: Data gathering tree algorithms on a network instance: $N = 500$, $\rho = 0.8$: (a) Shortest path tree (SPT), (b) Greedy algorithm, (c) Leaves deletion (LD), (d) Total cost. The total cost of having all the nodes transmitting their data directly to the sink is one order of magnitude larger.



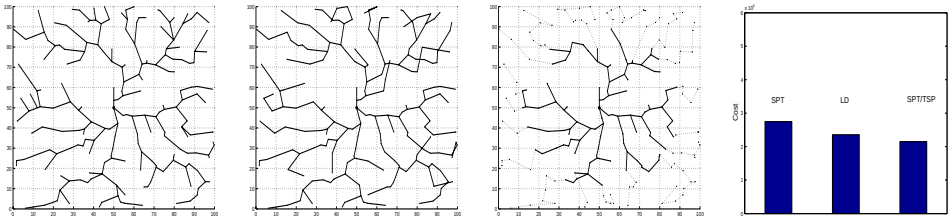
(a) SPT algorithm (b) Leaves deletion algorithm (c) Simulated annealing (d) Total cost

Figure 2.9: Data gathering tree algorithms on a network instance: $N = 100$, $\rho = 0.5$: (a) Shortest path tree (SPT), (b) Leaves deletion (LD), (c) Simulated annealing, (d) Total flow cost. Costs for this instance: SPT: 3.52e+6; LD: 3.36e+6; SA: 3.31e+5.

able and efficient from a complexity point of view (see Figure 2.9). Note that it is possible that simulated annealing did not provide the optimal solution either, but it is expected to do so with the right scheduling policy and with a long enough running time.

We show in Figure 2.10 some simulation results for the SPT/TSP algorithm. For networks with $\rho = 0.2$ and $N = 200$, the improvements are of the order of 10% over the LD algorithm. As the simulated annealing results suggest, a good tree solution has a small number of leaves, but at the same time short paths to the sink. Solutions for a network instance are shown in Figures 2.9–2.10. In Figure 2.10(c) we plot the branches in the SPT subtree in solid lines, and the branches added in the step involving TSP paths are shown in dashed lines.

Our experiments show important improvements of the LD and the SPT/TSP algorithms over SPT, in terms of average performance over randomly generated networks (see Figure 2.11).



(a) SPT algorithm (b) Leaves deletion algorithm (c) SPT/TSP algorithm (d) Total cost

Figure 2.10: Approximated gathering trees on a network instance: $N = 200$, $\rho = 0.2$: (a) Shortest path tree (SPT), (b) Leaves deletion (LD), (c) SPT/TSP algorithm. Costs for this instance: SPT: 2.74e+5; LD: 2.36e+5; SPT/TSP: 2.15e+5.

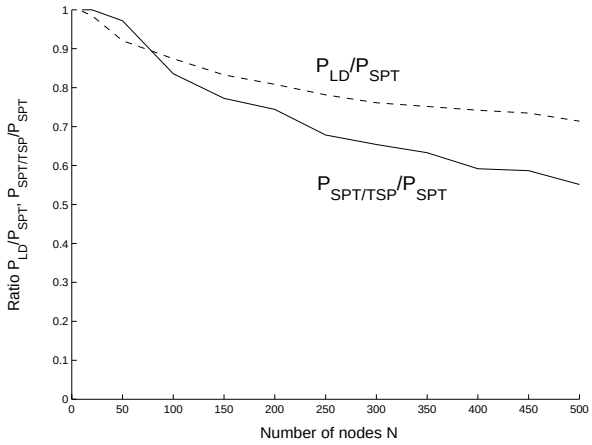


Figure 2.11: Average ratios of total costs between leaves deletion (LD) and SPT, and between balanced SPT/TSP and SPT: $\rho = 0.9$.

2.5 Conclusions

In this chapter, we formulated the network correlated data gathering problem with coding by explicit communication. Namely, we addressed an optimization problem that jointly considers rate allocation and transmission structure optimization in such networks. We showed that the two tasks cannot be separated in this case. Moreover, we proved that the problem is NP-complete even for simplified assumptions. We proposed approximation algorithms for the transmission structure that provide significant gains over the shortest path tree. Moreover, our algorithms provide solutions close to a provably optimal, but computational hard optimization method, namely simulated annealing.

Note that the NP-completeness of the problem for a single sink presented in this chapter generalizes by a straightforward reduction to the case of multiple sinks. However, the derivation of approximation algorithms for the multiple sinks case is significantly more difficult. For instance, the generalization of the *SPT/TSP* structure to multiple sinks is non-trivial due to the interactions between the approximated structures derived for each single sink in particular. This is because nodes that are leaves for a particular structure can be in-tree nodes for other structures, and thus their corresponding rate allocation cannot be determined. The study of approximation algorithms for the multiple sink case is a subject of our current research.

Appendix 2.A Proof of Theorem 2.1

Proof:

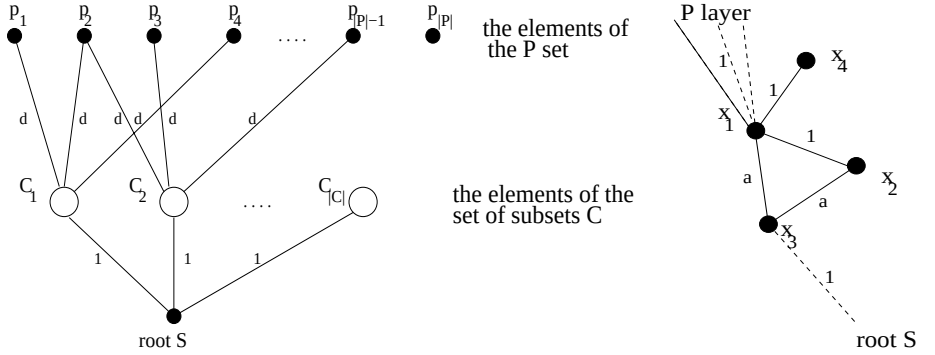
First, the decision version of our problem is in NP: a nondeterministic algorithm needs to guess the parent relationship (that is, specify the parent node for each of the nodes), and then find in polynomial time the nodes that are not parent nodes, assign to all nodes the number of bits corresponding to either leaf or in-tree node, and test that its total cost is less than the given value M.

Next, to prove the NP-hardness, we perform a reduction from the set cover problem [24], whose decision version is defined as follows:

Definition 2.2 *Set cover.*

- Instance*: A collection C of subsets of a finite set P and an integer $0 < K \leq |C|$, with $|C|$ the cardinality of C.*
- Question*: Does C contain a subset $C' \subseteq C$ with $|C'| \leq K$, such that every element of P belongs to at least one of the subsets in C' (this is called a set cover for P)?*

For any instance of the set cover problem we build an instance of our decision problem. Figure 2.12(a) illustrates the construction of the graph instance for our problem. The resulting graph is formed of three layers: a sink node S, a layer corresponding to the subsets $C_i \in C$, and a layer corresponding to the elements $\{p_j\}$ of the set P. For each element $C_i \in C$ we build a structure formed by 4 nodes x_1, x_2, x_3, x_4, as in Figure 2.12(b) (there are four different nodes for each subset C_i, but we drop the superscript C_i of the nodes x for the sake of simplicity). This structure originates



(a) A graph instance

(b) Inner structure of each subset C_i

Figure 2.12: Reduction from the min-set cover problem.

from our toy example in Section 2.3.1 and has properties linked with the tradeoffs observed there. The node x_3 is linked to the sink S, node x_4 is connected only to node x_1, and x_1, x_2, x_3 are all interconnected. Furthermore, we connect each structure $C_i \in C$ (namely the node x_1 from that structure) to only the nodes in the P layer that correspond to elements contained in C_i (example: in the instance in Figure 2.12(a), subset $C_1 = \{p_1, p_2, p_4\}, C_2 = \{p_2, p_3, p_{|P|-1}\}$ etc.) All the edges connecting the P layer to the C layer have a weight $d > 0$; for all C_i, the edges of type (x_1, x_3) and (x_2, x_3) have weight $a \geq 1$; the rest of the edges shown in Figure 2.12(a) have all weight 1. All other edges are assumed of infinite weight and are not plotted. Without loss of generality, we consider that in-tree nodes use $r = 1$ bits for coding their data, while leaf nodes use $R > 1$ bits.

The goal is to find a spanning tree for this graph, for which the cost in (2.3) is at most M. We now show that if $M = |P|(d + a + 1)R + K(2aR + 3R + a + 2) + (|C| - K)(aR + 3R + 2a + 4)$, for the positive integer $K \leq |C|$, then finding a spanning tree with cost at most M is equivalent to finding a set cover of cardinality K or less for the set P. Notice that the construction of our graph instance from the set cover instance can be performed in polynomial time.

With a large enough value chosen for d (i.e. $d > |C|(2aR + 3R + a + 2)/R$), a tree with cost at most M will contain exactly $|P|$ links between the layers P and C. That means that no p_j node is used as relay, so all $p_j \in P$ are necessarily leaf nodes. If some p_j node was used as relay, then the cost of the tree would contain R bits passing through more than $|P|$ such links, which would result in a cost larger than M. This also implies that the only way the C_i structures can connect to the sink S is via their corresponding x_3 node, so all x_3's must be in-tree nodes. Furthermore, all x_4's nodes need to be connected to their corresponding x_1 node in order to belong to the tree, so necessarily all x_4-s are leaf nodes and all x_1's nodes are in-tree nodes. The only degrees of freedom are the choices of two out of the three edges interconnecting the nodes x_1, x_2, x_3, for each structure C_i.

The key idea of our proof is that, for properly chosen values for d and a, finding a tree with cost at most M means connecting the nodes in layer P to *at most* K nodes

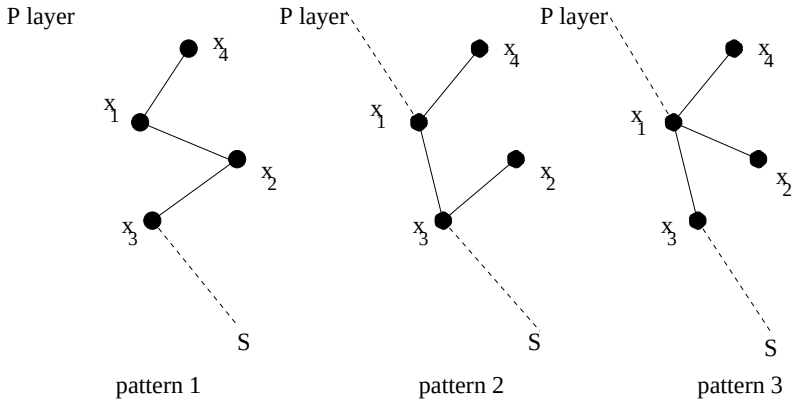


Figure 2.13: The three possible gathering patterns for the substructure C_i.

of layer C. If the tree needs to connect the layer P to more than K nodes in layer C, then the cost of the tree will necessarily be higher than M. The intuition is that 'detours' via the (x_1, x_2) edges are worthy from the point of view of cost reduction only if the flow that goes through node x_1 comes exclusively from node x_4 and no flow from the P layer goes through x_1. If some flow from the P layer joins as well, then the optimal path would use the edge (x_1, x_3) instead. In this latter case, we see now that for optimality, the edge (x_1, x_2) should not be used.

We choose a value of $a \geq 1$ such that $(a+2)/a < R < (a+2)/(a-1)$. Note that, for a given $R > 1$, it is always possible to choose a value for a that fulfills this condition.

With the given weights on the edges, if no p_j node is connected to a C_i structure, then since $R > (a+2)/a$, the optimal pattern (pattern 1, see Figure 2.13) for this structure contains the links (x_4, x_1), (x_1, x_2), (x_2, x_3), (x_3, S), with cost $(a+3)R+(a+2)+(a+1)+1$. The other possible structures contain either links (x_4, x_1), (x_1, x_3), (x_2, x_3), (x_3, S) (pattern 2) with cost $(a+2)R+(a+1)R+(a+1)+1$, or links $(x_4, x_1), (x_1, x_3), (x_2, x_1), (x_3, S)$ (pattern 3) with cost $(a+2)R+(a+2)R+(a+1)+1$. They both are sub-optimal if $R > (a+2)/a$ (since pattern 2 is always better than pattern 3, we will consider only pattern 2 for the rest of our proof).

However, when $m \geq 1$ nodes $\{p_j\}_{j=1}^{m}$ from the P layer connect to x_1, for any of C_i's, the pattern 1 is no longer optimal, because it has a cost $m(d+a+2)R+(a+3)R+(a+2)+(a+1)+1$. The alternative structure (pattern 2) has cost $m(d+a+1)R+(a+2)R+(a+1)R+(a+1)+1$, which is more efficient if $m \geq 1$, and $R < (a+2)/(a-1)$. We notice that in an optimal tree the cost to transmit data from each p_j to the sink S is the same for all p_j's nodes (and equal to $(d+a+1)R$). Therefore the goal is to keep minimal the part of the total cost corresponding to the rest of the nodes (i.e. nodes in layer C).

That means that to find a tree with cost less or equal to $|P|(d+a+1)R+K(2aR+3R+a+2)+(|C|-K)(aR+3R+2a+4)$ is equivalent to finding a set of K elements or less from the C layer to which all nodes in the set P connect. This is actually achieved by having at most K nodes of type x_1 used to connect to the p_j's nodes,

which turns out to be equivalent to finding a set cover for the set P of size K or less, that is to solving the set cover problem.

Thus our decision problem is NP-complete and our optimization problem NP-hard. □

Chapter 3

Networked Slepian-Wolf

3.1 Network Flows with Slepian-Wolf Coding

Consider again the setting of networks that transport supplies among nodes, introduced in the previous chapter. In this chapter, besides the single sink data gathering scenario, we will also consider the more general case of multiple sinks. An example is shown in Figure 3.1, where there are N nodes with sources $X_1, \ldots, X_N$, two of them being the sinks denoted by S_1 and S_2, and a graph of connectivity with edges connecting certain nodes. Sources corresponding to nodes in the sets V^1 and V^2 need to transmit their data, possibly using other nodes as relays, to sinks S_1 and S_2 respectively.

The source coding approach that takes maximum advantage of the data correlation, at the price of coding complexity, is based on Slepian-Wolf coding [66]. In that work, it was shown that when nodes measure correlated data, these data can be coded with a total rate not exceeding the joint entropy, even *without* nodes explicitly communicating with each other (under some constraints on the minimal rates given by the Slepian-Wolf region). Their result provides the whole achievable *rate region* for the rate allocation, that is *any* rate in that region is achievable. We describe in more detail the Slepian-Wolf coding in Section 3.2.2.

In addition to encoding the data, these data need to be transmitted over the network from the sources to the sinks. As in the previous chapter, we consider a joint treatment of the rate allocation and the chosen transmission structure, by means of cost functions that are functions of both. Cost functions usually met in practice separate in multiplicative terms the rate term from the link weight term (e.g. [function(rate)] $\times$ [link weight] metrics, like for example $R \cdot d$, $R \cdot d^{\alpha}$, or $e^R \cdot d$, where r is the bit rate and d is the distance). We show that if Slepian-Wolf coding is used in such scenarios, then the optimization separates: first an optimal transmission structure needs to be determined, and second the optimal rate allocation is found on this transmission structure. The optimal rate allocation is in general unique, except in some degenerate cases.

We will again first consider the case of a single sink, to where the data from all the nodes has to arrive. A rate allocation $(R_1, \ldots, R_N)$ (bits) has to be assigned at the nodes so that the quantized measured information samples are described losslessly.

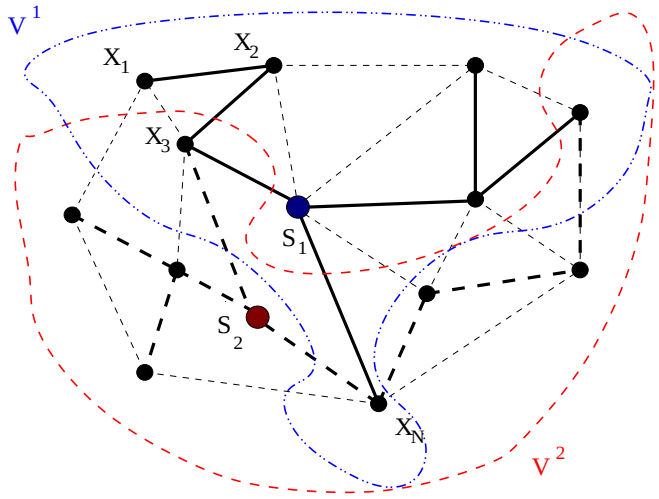


Figure 3.1: An example of a network. Sources transmit their data to the sinks. Data from the sets V^1 and V^2 need to arrive at sink S_1 and S_2, respectively. A rate supply R_i is allocated to each node X_i. In thick solid and dashed lines, a chosen transmission structure is shown. In thin dashed lines, the other possible links are shown.

The model for a single sink can be extended to the case when there is a number of sinks to where data from different subsets of nodes have to be sent. Notice that in this case it is also possible to allocate different rates at each node, depending on which sink it sends its data to, but this involves important additional coding overhead, which might not be always feasible. We consider both cases in this work, but for the sake of simplicity we concentrate our discussion on the unique rate allocation case. The sink-dependent rate allocation is a straightforward generalization of the unique rate allocation.

To the best of our knowledge, this is the first research work that addresses jointly Slepian-Wolf lossless source coding and network flow cost optimization.

The rest of this chapter is organized as follows. In Section 3.2, we state the optimization problem and describe the Slepian-Wolf source coding approach and the optimal region of rate allocations. In Section 3.3, we fully solve an important particular case, namely the correlated data gathering problem with Slepian-Wolf source coding, in the single sink case. In Section 3.4 we study the complexity for the case of a general traffic matrix problem and we prove that finding the optimal transmission structure is NP-complete. We show that, if centralized algorithms were allowed, finding the optimal rate allocation is simple; however, in our sensor network setting, the goal is to find distributed algorithms, and we show that in order to have a decentralized algorithm, we need a substantially large communication overhead in the network. In Section 3.5 we apply the results obtained in Section 3.3 to other particular cases of interests. In Section 3.6, we study the performance of Slepian-Wolf coding in large networks, in comparison with opportunistic compression. In Section 3.7 we present an efficient decentralized algorithm for approximating the optimal rate allocation in the data gathering case, and discuss how this algorithm can be used in the scenarios

presented in Section 3.5. Finally, we present some numerical simulations in Section 3.8.

3.2 Problem Formulation

3.2.1 Optimization Problem

Consider a graph $G = (V, E)$, with $|V| = N$ vertices. Each edge $e \in E$ is assigned a weight w_e. Nodes on the graph are sources of data. Some of the nodes are also sinks. Data has to be transported over the network from sources to sinks. Denote by $S_1, S_2, \ldots, S_M$ the set of sinks and by $V^1, V^2, \ldots, V^M$ the set of subsets $V^j \subseteq V$ of sources; data measured at nodes V^j have to be sent to sink S_j. Denote by S^i the set of sinks to which data from node i have to be sent. Denote by $E^i \subseteq E$ the subset of edges used to transmit data from node i to sinks S^i, which determines the transmission structure corresponding to node i.

Definition 3.1 (Traffic matrix) *We call the* traffic matrix *of a graph G the $N \times N$ square matrix T that has elements given by:*

$$\begin{aligned} T_{ij} &= 1, \text{ if source } i \text{ is needed at sink } j, \\ T_{ij} &= 0, \text{ else.} \end{aligned}$$

With this notation, $V^j = \{i : T_{ij} = 1\}$ and $S^i = \{j : T_{ij} = 1\}$.

The overall task we consider is to assign an optimal rate allocation $R_i^*, i = 1, \ldots, N$ for the N sources and to find the optimal transmission structure on the graph G that minimizes the total flow cost [F(rate)] $\times$ [link weight], where $F(\cdot)$ is a function of the rate. In most settings, the function $F(\cdot)$ is monotonically increasing, either linearly or super-linearly. Thus, the optimization problem is:

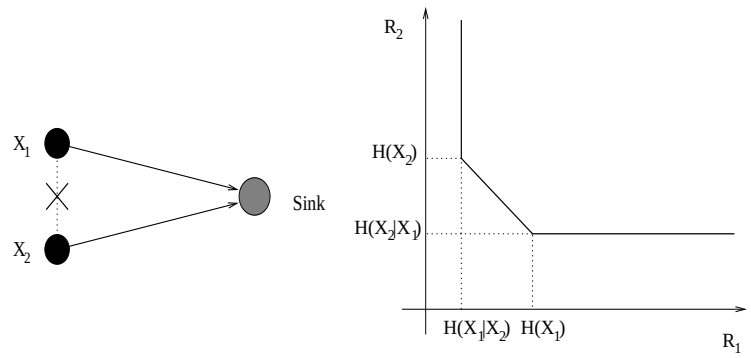
$$\{R_i^*, d_i^*\}_{i=1}^N = \arg \min_{\{R_i, d_i\}_{i=1}^N} \sum_{i=1}^N F(R_i) d_i \tag{3.1}$$

where d_i is the total weight of the transmission structure chosen to transmit data from source i to the set of sinks S^i:

$$d_i = \sum_{e \in E^i} w_e.$$

Notice that finding the optimal $\{d_i^*\}_{i=1}^N$ is equivalent to finding the optimal transmission structure.

In the next subsection we show that, when Slepian-Wolf coding is used, the task of finding the optimal rate allocation, or $\{R_i^*\}_{i=1}^N$, and the task of finding the optimal transmission structure, or the set $\{d_i^*\}_{i=1}^N$ can be split into two optimization problems. That is, one can first find the optimal transmission structure, which can be shown to always be a tree, and then find the optimal rate allocation. As a consequence, after finding the optimal transmission structure, the problem (3.1) can be posed as a Lagrangian optimization problem for finding the optimal rate allocation.



(a) Two correlated sources X_1, X_2 send their data to one sink.

(b) The Slepian-Wolf region shows the achievable pairs of rates that can be allocated to sources X_1 and X_2 for lossless data coding.

Figure 3.2: Two correlated sources, and the Slepian-Wolf region for their rate allocation. (a) Two correlated sources X_1, X_2 send their data to one sink. (b) The Slepian-Wolf region for two correlated sources.

3.2.2 Slepian Wolf Coding

Consider the case of two random sources X_1 and X_2 that are correlated (see Figure 3.2(a)). Intuitively, each of the sources can code their data at a rate greater or equal to their respective entropies, $R_1 = H(X_1)$, $R_2 = H(X_2)$, respectively. If they are able to communicate, then they could coordinate their coding and use together a total rate equal to the joint entropy $R_1 + R_2 = H(X_1, X_2)$. Slepian and Wolf [66] showed that two correlated sources can be coded with a total rate $H(X_1, X_2)$ even if they are *not* able to communicate with each other. Figure 3.2(b) shows the Slepian-Wolf rate region for the case of two sources. This can be generalized to the N-dimensional case.

Consider again the example shown in Figure 3.1. Assume that the set of sources that send their data to sink j, that is the set of nodes denoted $\{X_{j1}, \ldots, X_{j|V^j|}\} \in V^j, j = 1, 2$, know in advance the correlation structure in that set V^j. Then, nodes in V^j can code their data jointly, without communicating with each other, with a total rate of $H(X_{j1}, X_{j2}, \ldots, X_{j|V^j|})$ bits, as long as their individual rates obey the Slepian-Wolf constraints related to the different conditional entropies [10], [66]. As a consequence of the possibility of joint source coding without sources communicating among them, we can state the following [13], [12]:

Proposition 3.1 – *Separation of source coding and transmission structure optimization:*

When the [rate] and [link weight] terms of the joint cost function separate, and Slepian-Wolf coding is used, then, for any traffic matrix, the transmission structure optimization separates from the rate allocation optimization, in terms of the overall minimization of (3.1).

Proof: Once the rate allocation is *fixed*, the best way to transport any amount of data

from a given node i to the set of sinks S^i does not depend on the value of the rate. This is true because we consider separable flow cost functions, and the rate supplied at each node does not depend on the incoming flow at that node. Since this holds for any rate allocation, it is true for the minimizing rate allocation and the result follows. □

For each node i, the optimal transmission structure is in fact a Steiner tree with root at node i spanning the sinks S^i to which its data are sent [7]. Thus, the entire optimization problem can be separated into a spanning tree optimization for each source, and the rate allocation optimization. Then, after the optimal tree structure is formed, (3.1) becomes a problem of rate allocation that can be posed as an optimization problem under the usual Slepian-Wolf linear constraints. The optimal solution is therefore:

1. Find the optimal weights $\{d_i^*\}_{i=1}^N$, which determine uniquely the optimal transmission structure.
2. Given the found weights $\{d_i^*\}_{i=1}^N$, find the optimal rate allocation $\{R_i^*\}_{i=1}^N$ such that:

$$\{R_i^*\}_{i=1}^N = \arg \min_{\{R_i\}_{i=1}^N} \sum_{i=1}^N F(R_i) d_i^* \tag{3.2}$$

under constraints:

$$\sum_{l \in \mathcal{Y}} R_l \geq H(\mathcal{Y}^j | V^j - \mathcal{Y}^j), (\forall) V^j, \mathcal{Y}^j \subseteq V^j. \tag{3.3}$$

Note that there is one set of constraints for each set V^j.

Moreover, note that (3.2) is an optimization problem under linear constraints, so if the weights $\{d_i^*\}_{i=1}^N$ can be determined, the optimal allocation $\{R_i\}_{i=1}^N$ can be found easily with a *centralized* algorithm, by using Lagrange multipliers [46]. However, in Section 3.4 we will show that for a general traffic matrix T, finding the optimal coefficients $\{d_i^*\}_{i=1}^N$ is NP-complete. Moreover, in general, even if the optimal structure is found, it is hard to *decentralize* the algorithm that finds the optimal solution $\{R_i^*\}_{i=1}^N$ of (3.2), as this requires a substantial amount of global knowledge in the network.

Note that if the rate allocation at a source X_i can take different values R_{ij} depending to which sink j source X_i sends its data, then it is straightforward to find this multiple rate assignment. The rate allocation for each cluster will be independent from the rate allocations in the other clusters (see Section 3.3 for the closed-form solution for the rate allocation for each set V^j, and Section 3.7 for a decentralized algorithm for finding the optimal solution). However, this involves even more additional complexity in coding, so in many situations it might be desirable to assign a unique rate to each node, regardless of which sink the data is sent to. Thus, without loss of generality, we will restrict our further discussion to the study of the case where a single rate allocation is done at each node regardless of the number of sinks to where data measured at that node has to arrive.

In the following sections, for various problem settings, we first show how the transmission structure can be found (i.e. the values of $\{d_i^*\}_{i=1}^N$), and then we discuss the complexity of finding the rate allocation (3.2) in a decentralized manner.

We begin our analysis with a particular case widely encountered in practice, namely the *data gathering* problem. In this case, there is only one sink to which data from all the other nodes have to arrive. We fully solve this case in Section 3.3, and this result will provide useful insight into the structure of the general problem.

3.3 Data Gathering

In data gathering, the entire set of N sources ($V^j = V$) are sent to a single sink $S = j$. An example is shown in Figure 2.1.

First, we state the following corollary of Proposition 3.1, valid for data gathering scenarios.

Corollary 3.1 – *Separation of source coding and tree building in data gathering: When there is a single sink and Slepian-Wolf coding is used, the shortest path tree (SPT) is optimal, in terms of minimizing (3.2), for any rate allocation.*

Proof: Once the rate allocation is *fixed*, the best way to transport any amount of data from any node to the sink is to use the shortest path. Minimizing the sum of costs in (3.2) becomes equivalent to minimizing the cost corresponding to each node in part. Since the shortest path tree is a superposition of all individual shortest paths, it is optimal for any rate allocation that does not depend on the transmission structure, which is the case here. □

We consider two important functions of the rate term in the cost function, that are widely met in practice. Namely, we study in Section 3.3.1 the case when $F(R) = R$, that corresponds to wired networks, and in Section 3.3.2 the case when $F(R) = e^R$, that corresponds to wireless networks.

3.3.1 Linear Cost Function: Optimal Solution

In the case of a linear cost functional $F(R) = R$, the expression (3.2) to be minimized can be rewritten as:

$$\sum_{i \in V} R_i d_{ST}(i, S) \tag{3.4}$$

where $d_{ST}(i, S)$ is the total weight of the path connecting node i to S on the spanning tree ST, and under the constraints:

$$\sum_{i \in \mathcal{Y}} R_i \geq H(\mathcal{Y}|\mathcal{Y}^C) \tag{3.5}$$

for any of the $2^N - 1$ subsets $\mathcal{Y} \subseteq V$ (see Figure 3.2(b) for the rate region of two sources).

The minimization of (3.4) becomes now a linear programming (LP) problem:

$$\{R_i^*\}_{i=1}^N = \arg \min_{\{R_i\}_{i=1}^N} \sum_{i \in V} R_i d_{SPT}(i, S) \tag{3.6}$$

under constraints (3.5).

Suppose without loss of generality that nodes are numbered in increasing order of the total weight of their path to the sink on the SPT: $(X_1, X_2, \ldots, X_N)$ with $d_{SPT}(X_1, S) \leq d_{SPT}(X_2, S) \leq \cdots \leq d_{SPT}(X_N, S)$. That is, node X_1 is the closest node to the sink and node X_N is the furthest one from the sink on the SPT.

Theorem 3.1 (LP solution) *The solution of the optimization problem in (3.6) is [13], [11]:*

$$\begin{array}{rcl} R_1^* & = & H(X_1), \\ R_2^* & = & H(X_2|X_1), \\ \ldots & \ldots & \ldots \\ R_N^* & = & H(X_N|X_{N-1}, X_{N-2}, \ldots, X_1). \end{array} \tag{3.7}$$

In words, the solution of this problem is given by the innermost corner of the Slepian-Wolf region that is tangent to the cost function (see Figure 3.3 for an example with two sources). The closest node to the sink is coded with a rate equal to its unconditioned entropy. Each of the other nodes is coded with a rate equal to its respective entropy conditioned on all other nodes which are closer to the sink than itself.

Proof: First, we prove that (3.7) is indeed a feasible solution for (3.6), that is, it satisfies all the constraints given by (3.5). Consider any constraint from (3.5), for some subset $\mathcal{Y} \in V$. Denote by $M = |\mathcal{Y}|$ the number of elements in $\mathcal{Y}$. Order the indices of $X_i \in Y$ as $i_1, i_2, i_3, \ldots i_M$, with i_1 closest and i_M furthest from the sink on the SPT.

We rewrite the left-hand-side in terms of the solutions we give in the theorem:

$$\begin{array}{rcl} \sum_{i \in \mathcal{Y}} R_i & = & H(X_{i_M}|X_{i_M-1}, \ldots, X_1) + \\ & & H(X_{i_{M-1}}|X_{i_{M-1}-1}, \ldots, X_1) + \\ & & \cdots + \cdots + \ldots \\ & & H(X_{i_1}|X_{i_1-1}, \ldots, X_1) \end{array} \tag{3.8}$$

and expand the right-hand-side terms with the chain law for conditional entropies:

$$\begin{array}{rcl} H(\mathcal{Y}|\mathcal{Y}^C) & = & H(X_{i_M}|\mathcal{Y}^C \cup \{\mathcal{Y} - \{X_{i_M}\}\}) + \\ & & H(X_{i_{M-1}}|Y^C \cup \{\mathcal{Y} - \{X_{i_M}, X_{i_{M-1}}\}\}) + \\ & & \cdots + \cdots + \ldots \\ & & H(X_{i_1}|\mathcal{Y}^C \cup \{\mathcal{Y} - \{\mathcal{X}_{i_M}, \ldots, X_{i_1}\}\}) \\ & = & H(X_{i_M}|V - \{X_{i_M}\}) + \\ & & H(X_{i_{M-1}}|V - \{X_{i_M}, X_{i_{M-1}}\}) + \\ & & \cdots + \cdots + \ldots \\ & & H(X_{i_1}|V - \{X_{i_M}, X_{i_{M-1}}, \ldots, X_{i_1}\}) \end{array} \tag{3.9}$$

Consider the terms on the right-hand-side in expressions (3.8) and (3.9). It is clear that for any $i_k \in \mathcal{Y}$, the term corresponding to X_{i_k} in (3.9) is at most equal to its counterpart in (3.8). This is because the set of nodes on which the entropy

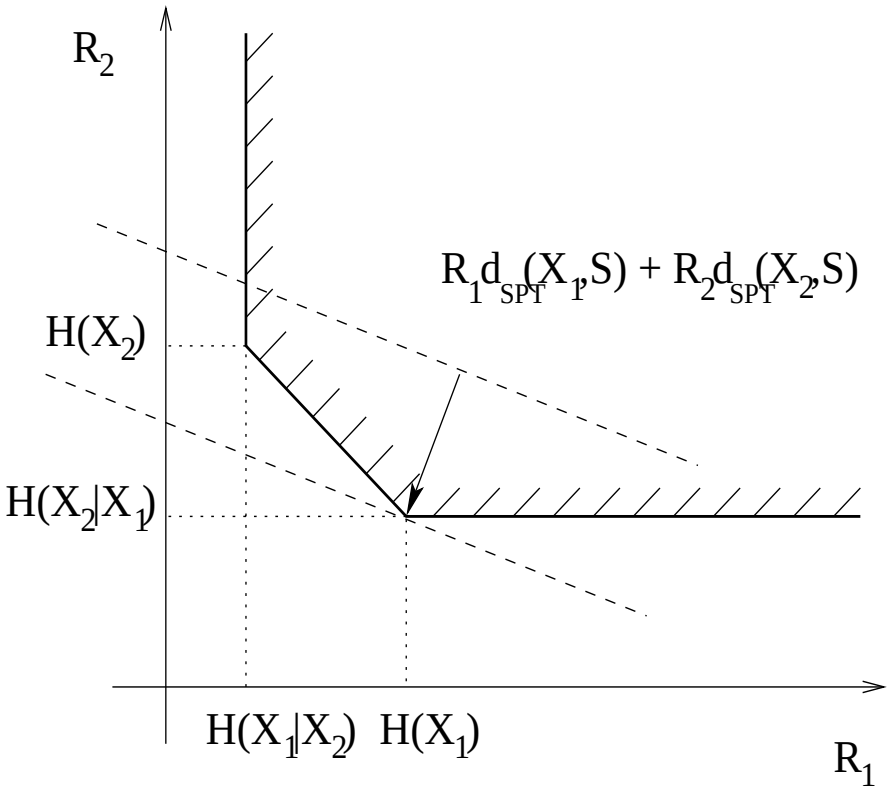


Figure 3.3: A simple example with two nodes. The total weights from sources X_1, X_2 to the sinks, are respectively $d_{SPT}(X_1, S), d_{SPT}(X_2, S)$, $d_{SPT}(X_1, S) < d_{SPT}(X_2, S)$, in this particular case. In order to achieve the minimization, the cost line $R_1 d_{SPT}(X_1, S) + R_2 d_{SPT}(X_2, S)$ has to be tangent to the most interior point of the Slepian-Wolf rate region, given by $(R_1, R_2) = (H(X_1), H(X_2|X_1))$.

conditioning is done for each term in (3.8) is a subset of its counterpart in (3.9). Since the choice of $\mathcal{Y}$ was arbitrary, then any constraint in (3.5) is satisfied by the assignment (3.7).

On the other hand, note also that the rate allocation in (3.7) satisfies with equality the constraint on the total sum of rates:

$$\sum_{i \in V} R_i \geq H(X_1, \ldots, X_N). \tag{3.10}$$

So we have proven that (3.7) is a valid rate allocation. We have to prove now that the assignment in (3.7) makes the expression to be minimized in (3.6) smaller than any other valid assignment. Figure 3.3 gives an example involving only two nodes, showing how the cost function is indeed minimized with such a rate allocation. The assignment (3.7) corresponds in this case to the point $(R_1, R_2) = (H(X_1|X_2), H(X_2))$.

We prove this by recursion. Note first that the rate allocation to node N is minimal. That is, we cannot allocate to X_N less than $H(X_N|X_{N-1}, X_{N-2}, \ldots, X_1)$ bits, due to the Slepian-Wolf constraint corresponding to $\mathcal{Y} = \{X_N\}$. Assume now that a solution that assigns $H(X_N|X_{N-1}, X_{N-2}, \ldots, X_1)$ bits to X_N is not optimal, and X_N is assigned $H(X_N|X_{N-1}, \ldots, X_1) + b$ bits. Due to (3.10), at most b bits in total can be extracted from the rates assigned to some of the other nodes. But since $d_{SPT}(X_N, S)$ is the largest coefficient in the optimization problem (3.6), it is straightforward to see that any such change in rate allocation increases the cost function in (3.6). Thus assigning $R_N = H(X_N|X_{N-1}, \ldots, X_1)$ bits to X_N is indeed optimal.

Consider now the rate assigned to X_{N-1}. From the rate constraint corresponding to $\mathcal{Y} = \{X_{N-1}, X_N\}$, it follows that:

$$\begin{aligned} R_N + R_{N-1} &\geq H(X_N, X_{N-1}|X_{N-2}, \ldots, X_1) \\ &= H(X_N|X_{N-1}, X_{N-2}, \ldots, X_1) + \\ &\quad + H(X_{N-1}|X_{N-2}, \ldots, X_1) \end{aligned}$$

Since for optimality R_N must be given by $R_N = H(X_N|X_{N-1}, X_{N-2}, \ldots, X_1)$, it follows that $R_{N-1} \geq H(X_{N-1}|X_{N-2}, \ldots, X_1)$. Following a similar argument as for X_N, we can show in the same way that the optimal solution allocates $R_{N-1} = H(X_{N-1}|X_{N-2}, \ldots, X_1)$. The rest of the proof follows similarly by considering successively the constraints corresponding to the subsets $\mathcal{Y} = \{X_i, X_{i+1}, \ldots, X_N\}$, with $i = N-2, N-3, \ldots 1$. □

Even if the solution can be provided in a closed form as (3.7), a distributed implementation of the optimal algorithm at each node implies knowledge of the overall structure of the network (distances between nodes and distances to the sink). This knowledge is needed for:

1. Ordering the distances on the *SPT* from the nodes to the sink: each node needs its index in the ordered sequence of nodes so as to determine on which other nodes to condition when computing its rate assignment. For instance, it may happen that the distance on the graph between nodes X_N and X_{N-1} is large. Thus, closeness in the ordering does not mean necessarily closeness in distance.

2. Computation of the rate assignment:

$$\begin{aligned} R_i &= H(X_i|X_{i-1},\ldots,X_1) \\ &= H(X_1,\ldots,X_i) - H(X_1,\ldots,X_{i-1}) \end{aligned}$$

Note that *all* distances among nodes $(X_1,\ldots,X_i)$ are needed locally at node X_i for computing this rate assignment.

But this implies that for a distributed algorithm, global knowledge should be available at nodes, which might not be the case in practical settings.

Therefore, in Section 3.7, we propose a fully distributed approximation algorithm, which avoids the need for a node to have global knowledge of the network, and which provides solutions very close to the optimum. The approximation algorithm we propose is based on the observation that nodes that are outside this neighborhood count very little, in terms of rate, in the local entropy conditioning, under the assumption that local correlation is dominant. For instance, in sensor networks, this is a natural assumption, since usually the correlation decreases with the distance between nodes.

3.3.2 Cost Function with Exponential Rate Term

We consider now the metric exp[rate] $\times$ [link weight]. This metric is useful in the case of wireless networks. Note that the [link weight] in this case is a power k, $k \in [2,4]$ of the corresponding Euclidean distance.

Then the optimization problem is now:

$$\{R_i^*\}_{i=1}^N = \arg \min_{\{R_i\}_{i=1}^N} \sum_{i\in V} e^{R_i} d_{SPT}(i,S) \tag{3.11}$$

under the Slepian-Wolf constraints (3.5).

We will make the following change of variables:

$$P_i = e^{R_i}, \text{for all } i = 1,\ldots,N,$$

Then (3.11) becomes equivalent to:

$$\{P_i^*\}_{i=1}^N = \arg \min_{\{P_i\}_{i=1}^N} \sum_{i\in V} P_i d_{SPT}(i,S)$$

under the constraints:

$$\sum_{i\in Y} \log P_i \geq H(\mathcal{Y}|\mathcal{Y}^C)$$

for all $\mathcal{Y} \subseteq V$.

Let us study (3.11) for the simple case of two sources. The problem in this case is:

$$\{P_1,P_2\}^* = \arg \min_{P_1,P_2} \left(P_1 d_{SPT}(1,S) + P_2 d_{SPT}(2,S)\right)$$

under constraints

$$\begin{aligned} P_1 &\geq e^{H(X_1|X_2)}, \\ P_2 &\geq e^{H(X_2|X_1)}, \\ P_1P_2 &\geq e^{H(X_1,X_2)}. \end{aligned}$$

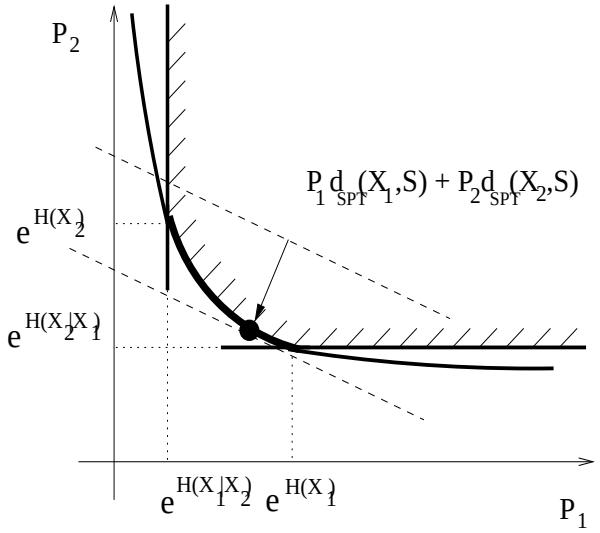


Figure 3.4: The same example with two nodes. In order to achieve the minimization, the cost line $P_1 d_{SPT}(X_1, S) + P_2 d_{SPT}(X_2, S)$ has to be tangent to the most interior point of the transformed Slepian-Wolf rate region.

The solution of this minimization problem is illustrated in Figure 3.4. Note that, with the change of variables, the cost function remains linear, while the constraints turn from additive to multiplicative (due to the log-transformation).

We observe that, even in this simple example, the rate allocation depends on the slope of the cost function (and thus implicitly on the weights $d_{SPT}(i, S)$). Depending on these weights, the innermost tangent of the cost function to the rate region curve can be either on the joint constraint curve, or on one of the corner points. It is thus difficult in general to provide a direct closed form for the optimal rate allocation. However, since all the constraints are multiplicative, the solution is given by the point where, starting from infinite and going towards the origin, the cost function touches the bottom of the first curve/corner point corresponding to the strongest constraint. But in order to find this constraint, one essentially has to look over *all* the possible subsets of nodes.

We have studied so far the case when there is a single sink. For simplicity, we will limit our discussion further to the case of linear separable cost function. The case of non-linear separable cost functions can be solved by using Lagrange multipliers. Namely, the Slepian-Wolf constraints are linear, and the optimal rate allocation can be found by putting to zero all partial derivatives of the Lagrangian with respect to each of the unknowns $\{R_i\}_{i=1}^N$, and solving the resulting system.

Let us now discuss in the next section the arbitrary traffic matrix case.

3.4 Arbitrary Traffic Matrix

We begin the analysis with the most general case, that is when the traffic matrix T is arbitrary, by showing the following proposition [12]:

Proposition 3.2 – *The optimal transmission structure is a superposition of Steiner trees:*

Given an arbitrary traffic matrix T, for any i, the optimal value d_i^* in (3.2) is given by the minimum weight tree rooted in node i and which spans the nodes in S^i; this is exactly the minimum Steiner tree that has node i as root and which spans S^i, which is an NP-complete problem.

Proof: The proof is straightforward: data from node i has to be send over the minimum weight structure to the nodes in S^i, possibly via nodes in $V - \{i, S^i\}$. This is a minimum Steiner tree problem for the graph G with weights w_e, thus it is NP-complete. □

The approximation ratio of an algorithm that finds a solution for an optimization problem is defined as the guaranteed ratio between the cost of the found solution and the optimal one. If the weights of the graph are the Euclidean distances ($w_e = l_e$ for all $e \in E$), then the problem becomes the Euclidean Steiner tree problem, and it admits a polynomial time approximation scheme (PTAS) [4] (that is, for any $\epsilon > 0$, there is a polynomial time approximation algorithm with an approximation ratio of $1 + \epsilon$). However, in general, the link weights are not the Euclidean distances (e.g. if $w_e = l_e^2$ etc.). Then finding the optimal Steiner tree is APX-complete (that is, there is a hard lower bound on the approximation ratio), and is only approximable (with polynomial time in the input instance size) within a constant factor $(1 + \ln 3)/2$ [6], [61].

The approximation ratios of the algorithms for solving the Steiner tree translate into bounds of approximation for our problem. By using the respective approximation algorithms for determining the weights d_i, the cost of the approximated solution for the joint optimization problem will be within the Steiner approximation ratio away from the optimal one.

Once the optimal weights $\{d_i^*\}_{i=1}^N$'s are found (i.e. approximated by some approximation algorithm for solving the Steiner tree), then, as we mentioned above, (3.2) becomes a Linear Programming (LP) problem. The solution of this problem is given by the innermost corner of the Slepian-Wolf region that is tangent to the plane of the cost function (see Figure 3.3). Consequently, it can be readily solved with a centralized program. If global knowledge of the network is allowed, then this problem can be solved computationally in a simple way. However, it is not possible in general to find in closed-form the optimal solution determined by the corner that minimizes the cost function. Consequently, the derivation of a decentralized algorithm for the rate allocation is difficult, as this involves exchange of network knowledge among the clusters.

Example 3.1 *Figure 3.5 shows a simple example (but sufficiently complete) which illustrates the difficulty of this problem. Suppose that the optimal total weights $\{d_i^*\}_{i=1}^3$*

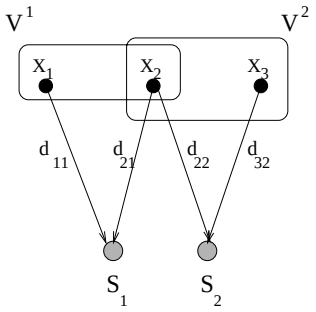


Figure 3.5: Two sets of sources transmit their correlated data to two sinks.

in (3.2) have been approximated by some algorithm. Then the cost function to be minimized is:

$$R_1 w_{11} + R_2(w_{21} + w_{22}) + R_3 w_{32}$$

with $d_1^* = w_{11}, d_2^* = w_{21} + w_{22}, d_3^* = w_{32}$*, and the Slepian-Wolf constraints are given by:*

$$\begin{aligned} R_1 + R_2 &\geq H(X_1, X_2) \\ R_1 &\geq H(X_1|X_2), \text{ for set } V^1 = \{X_1, X_2\} \\ R_2 &\geq H(X_2|X_1) \end{aligned}$$

and respectively,

$$\begin{aligned} R_2 + R_3 &\geq H(X_2, X_3) \\ R_2 &\geq H(X_2|X_3), \text{ for set } V^2 = \{X_2, X_3\} \\ R_3 &\geq H(X_3|X_2). \end{aligned}$$

Suppose the weights are such that $w_{11} < w_{21} + w_{22} < w_{32}$*. A decentralized algorithm has to use only local information, that is, information only available in a certain local transmission range or cluster neighborhood). We assume that only local Slepian-Wolf constraints are considered in each set* V^j *for the rate allocation, and no knowledge about the total weights* d_i *from nodes in the other subsets is available. Then, as we have seen in Section 3.3, it readily follows that the optimal rate allocations in each of the two subsets are:*

$$\begin{aligned} R_1' &= H(X_1) \\ R_2' &= H(X_2|X_1), \text{ for set } V^1 \end{aligned}$$

and respectively,

$$\begin{aligned} R_2' &= H(X_2) \\ R_3' &= H(X_3|X_2), \text{ for set } V^2. \end{aligned}$$

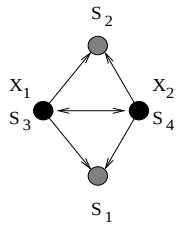


Figure 3.6: Data from X_1, X_2 need to be transmitted to all nodes S_1, S_2, S_3, S_4.

We can see from this simple example that we cannot assign the correct unique optimal rate R_2 to source X_2, unless node 2 has global knowledge of the whole distance structure from nodes $1, 2, 3$ to the sinks S_1 and S_2. This rate allocation depends on all the set of weights $w_{11}, w_{21}, w_{22}, w_{32}$. It is clear that such global sharing of cluster information over the network is not scalable, because the amount of necessary global knowledge grows exponentially.

There are however some other important special cases of interest where the problem is more tractable, and we treat them in the following section.

3.5 Other Cases of Interest

3.5.1 Broadcast of Correlated Data

This case corresponds to the scenario where some sources are sent to all nodes ($S^i = V$). A simple example is shown in Figure 3.6. In this example, the traffic matrix has $T_{ij} = 1, (\forall)j$, for some arbitrary L nodes $\{i_1, \ldots, i_L\} \subset V$.

In this case, for any node i, the value d_i^* in (3.2) is given by the tree of minimum weight which spans V; this is the minimum spanning tree (MST), and thus, by definition, it does not depend on i. Thus, in this case all weights $\{d_i^*\}_{i=1}^N$ are equal. Notice that this case trivially includes the typical broadcast scenario where one node transmits its source to all the nodes in the network.

3.5.2 Multiple Sink Data Gathering

This case corresponds to the scenario where all sources ($V^j = V$) are sent to some set S^a of sinks. In this case (see Figure 3.7), finding the optimal weights $\{d_i^*\}_{i=1}^N$ is as difficult as in the arbitrary matrix case, presented in Section 3.4. For every i, the optimal weight d_i^* is equal to the weight of the minimum Steiner tree rooted at i and spanning the nodes in the set S^a.

However, given the optimal transmission structure, the optimal rate allocation can be easily found in a similar manner as in Section 3.3.1. First, we order the nodes by increasing distance $d_1^* < d_2^* < \cdots < d_N^*$, and then the optimal rate allocation is given as (3.7).

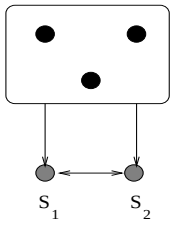


Figure 3.7: Data from all nodes has to be transmitted to the set of sinks $S^a = \{S_1, S_2\}$. Each sink has to receive data from *all* the sources.

3.5.3 Localized Data Gathering

This case corresponds to the scenario where disjoint sets $\{V^1, V^2, \ldots, V^L\}$ are sent to some sinks $\{S_1, S_2, \ldots, S_L\}$. In this case, for each i, the solution for the optimal weight d_i^* is again the corresponding Steiner tree rooted at i and that spans S^i. If d_i^* can be found, then the rate allocation can be solved for each set $\{V^j\}_{j=1}^L$, in the same way as in Section 3.3.1, that is, we solve L LP programs independently (decentralization up to cluster level).

Algorithm 3.1 *Disjoint sets:*

- *For each set V^j, order nodes $\{i, i \in V^j\}$ as a function of the total weight d_i.*
- *Assign rates in each V^j as in (3.7), taking into account this order.*

3.6 Scaling Laws: Slepian-Wolf vs. Explicit Communication

The alternative to Slepian-Wolf coding, namely explicit communication, was presented in the previous chapter. The advantages that coding by explicit communication has over Slepian-Wolf coding are (i) no a-priori knowledge of the correlation structure is needed, and (ii) the compression is trivial at nodes relaying data. However, even for the simple one dimensional setting presented in this section, our analysis shows that in large networks, Slepian-Wolf coding can provide very important gains over opportunistic aggregation, in terms of total flow cost.

Let N nodes be placed uniformly on a line (see Figure 3.8). The distance between two consecutive nodes is d. The nodes need to send their correlated data to the sink S.

For this scenario, the SPT is clearly the optimal data gathering structure for both coding approaches. Thus, the overall optimization problem simplifies, and we can compare the two different rate allocation strategies in terms of how they influence the total cost.

Within the one-dimensional model, we consider [14] two important cases of network scalability, namely, the *refinement network*, where the total distance from node N to the sink S is kept constant $Nd = 1$ (that is, nodes are uniformly placed on an interval of length 1, and hence, by adding nodes, the inter-node distance goes to

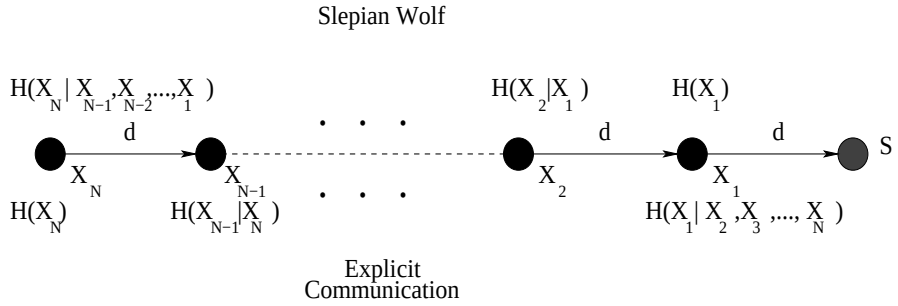


Figure 3.8: A one dimensional example: The rate allocations for Slepian-Wolf (above the line) and explicit communication (below the line).

zero), and the *expanding network*, where the inter-node distance is kept constant and equal to $d = 1$ (that is, by adding nodes we increase the distance between node N and the sink S).

In this work, we consider continuous-space processes of different types, including regular and singular continuous-space random processes [40]. In particular, we consider two classes of random processes:

(a) In the first model, we assume that the nodes of the network are sampling a Gaussian continuous-space non-bandlimited wide-sense-stationary (WSS) random process $X_c(s)$, where s denotes the position (this kind of correlation structure was introduced in Section 3.8.1). Thus, we have a vector of correlated sources $\mathbf{X} = (X_1, \ldots, X_N)$ where $X_i = X_c(id)$ and where the correlation structure for the vector $\mathbf{X}$ is inherited from the correlation present in the original process $X_c(s)$. As N goes to infinity, the set of correlated sources represents a discrete-space random process denoted by $X_d(i)$, with the index set given by the node positions. Thus the spatial data vector $\mathbf{X}$ measured at the nodes has an N-dimensional multivariate normal distribution $G_N(\boldsymbol{\mu}, \mathbf{K})$. We consider two correlation models, namely (a.1): $K_{ij} = \sigma_{ij}^2 \exp(-c|d_{i,j}|)$, which corresponds to a regular continuous-space process [40], and (a.2): $K_{ij} = \sigma_{ij}^2 \exp(-c|d_{i,j}|^2)$, which corresponds to a singular continuous-space process [40], where $c > 0$.

(b) In the second model, we assume that the nodes are sampling a Gaussian continuous-space WSS bandlimited process. This process can also be shown to be a singular continuous-space process.

Let us denote the conditional entropies by $a_i = H(X_i|X_{i-1}, \ldots, X_1)$. Note that for any correlation structure, the sequence a_i is monotonically decreasing (because conditioning cannot increase entropy), and is bounded from below by zero (because the entropy cannot be negative). Since the nodes are equally spaced, and the correlation function of a WSS process is symmetric, it is clear that $H(X_I|X_{I-1}, X_{I-2}, \ldots, X_{I-i}) = H(X_I|X_{I+1}, X_{I+2}, \ldots, X_{I+i})$, for any $I, 0 \leq i \leq I-1$.

Let us denote by γ_N the ratio between the total cost associated to Slepian-Wolf coding ($\text{cost}_{SW}(N)$) and the total cost corresponding to coding by explicit communi-

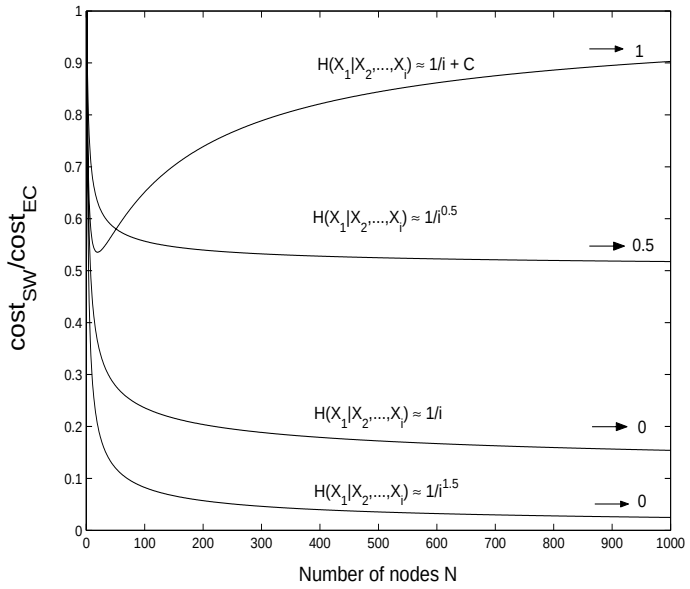


Figure 3.9: Typical behavior of the ratio of the total costs $\text{cost}_{SW}(N)/\text{cost}_{EC}(N)$, for various speeds of decay of the conditional entropy.

cation ($\text{cost}_{EC}(N)$), that is:

$$\gamma_N = \frac{\text{cost}_{SW}(N)}{\text{cost}_{EC}(N)} = \frac{\sum_{i=1}^{N} i a_i}{\sum_{i=1}^{N} (N-i+1) a_i} \tag{3.12}$$

Then, the following theorem holds [14], [11]:

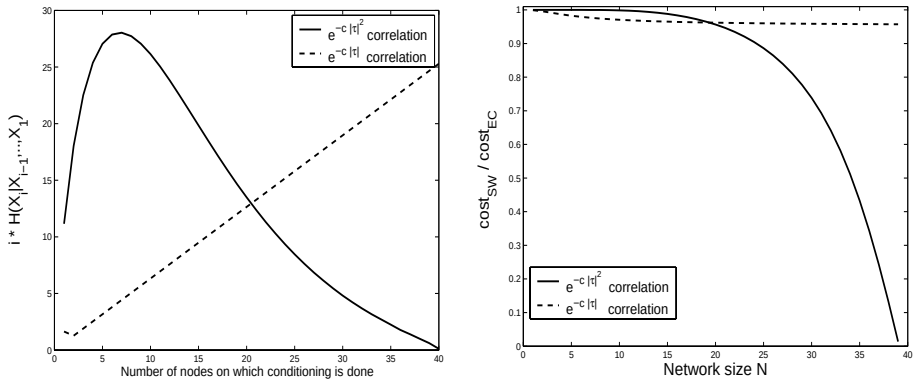
Theorem 3.2 (Scaling laws.) *When $N \to \infty$, we have:*

(i) If $\lim_{i\to\infty} a_i = C > 0$, then $\lim_{N\to\infty} \gamma_N = 1$. If $\lim_{i\to\infty} a_i = 0$ and a_i decreases as fast as $1/i^p, p \in (0,1)$, then $\lim_{N\to\infty} \gamma_N = 1-p$. In both cases, $cost_{SW}(N) = \Theta(cost_{EC}(N))$.

(ii) If $\lim_{i\to\infty} a_i = 0$ and a_i decreases at least as fast as $1/i$, then $\lim_{N\to\infty} \gamma_N = 0$, that is, $cost_{SW}(N) = o(cost_{EC}(N))$. Moreover, for a_i decreasing as fast as $1/i^p, p \geq 1$, we have the following different scaling laws depending on the value of p: (ii-1) if $p = 1$, then $\gamma_N = \Theta(1/\log N)$; (ii-2) if $p \in (1,2)$, then $\gamma_N = \Theta(1/N^{p-1})$; (ii-3) if $p = 2$, then $\gamma_N = \Theta(\log N/N)$; (ii-4) if $p > 2$, then $\gamma_N = \Theta(1/N)$.

We prove Theorem 3.2 in Appendix 3.A. Note that when $\lim_{i\to\infty} i a_i = 0$, the rate of decay of the ratio γ_N of costs is determined by the rate of increase with N of the partial sum $s^N(q) = \sum_{i=1}^{N} \frac{1}{i^q}$ for $q \in (0,1]$. In Figure 3.9, we show typical behaviors of the ratio of total flow costs for the two approaches.

We apply now Theorem 3.2 to the correlation models we consider in this work:



(a) The product between the conditional entropy and number of nodes on which conditioning is done.

(b) Ratio of total costs.

Figure 3.10: We consider a Gaussian random process, where the correlation dependence on the inter-node distance d is $\exp(-c|\tau|^{\beta}), \beta \in \{1, 2\}$: (a) The conditional entropy $H(X_i|X_{i-1}, \ldots, X_1)$ decreases faster than $1/i$ for $\beta = 2$, but is constant for $\beta = 1$ (after $i \geq 2$); (b) The behavior of the ratio of total $\text{cost}_{SW}(N)/\text{cost}_{EC}(N)$ with increasing size of the network.

- For an expanding network: In cases (a.1) and (a.2), the result of sampling is a discrete-space regular process [53], thus $\lim_{i\to\infty} a_i = C > 0$, and it follows that $\lim_{N\to\infty} \gamma_N = 1$. In case (b), if the spatial sampling period d is smaller than the Nyquist sampling rate of the corresponding original continuous-space process, then $\lim_{i\to\infty} a_i = 0$. The speed of convergence of a_i depends on the sampling period and the specific (bandlimited) power-spectrum density function of the process; the exact theoretical analysis of this dependence is a topic of further research.

- For a refinement network: In case (a.1), we show in Appendix 3.B that $a_i = H(X_i|X_{i-1}, \ldots, X_1) = H(X_i|X_{i-1})$, for any $i \geq 2$. Then for any finite N, $a_N > 0$. Since ia_i does not converge to zero (see Figure 3.10), then it follows from Theorem 3.2 that in the limit the ratio of total costs is $\lim_{N\to\infty} \gamma_N = 1$. In case (a.2), a closed-form expression for the conditional entropy is difficult to derive. However, we show numerically in Figure 3.10(a) that in this case a_i decreases faster[1] than $1/i$. Thus, from Theorem 3.2, $\lim_{N\to\infty} \gamma_N = 0$. In Figure 3.10(b), we plot also the ratio of total costs for this correlation model. Finally, in case (b), it is clear that a_i goes to zero very fast, as for the case (a.2), because of the singularity of the original bandlimited process, thus $\lim_{N\to\infty} \gamma_N = 0$.

[1]Since these two processes are both non-bandlimited, sampling them results in discrete-space regular processes [53]. However, the sampled model (a.2) inherits a 'superior predictability' than (a.1), which makes a_i decrease faster than $1/i$.

3.7 Approximation Algorithms

In Section 3.3.1, we found the optimal solution of the linear programming rate assignment under Slepian-Wolf constraints. We consider now the problem of designing a distributed algorithm [13].

Suppose that the correlation in the network depends on the distance between the nodes. In many practical situations, the correlation between data sources decreases when the distance between the sources increases. For such a scenario, suppose each node i has complete information (distances between nodes and distances to the sink) only about a local vicinity $\mathcal{N}(i)$ formed by its closest neighbors. This information can be computed by running for example a distributed algorithm for finding the *SPT* (e.g. Bellman-Ford).

Algorithm 3.2 *Approximated Slepian-Wolf coding:*

- *Find the SPT.*
- *For each node i:*
  - *Find in the neighborhood $\mathcal{N}(i)$ the set $\mathcal{C}_i$ of nodes that are closer to the sink, on the SPT, than node i.*
  - *Transmit at rate $R_i^\dagger = H(X_i|\mathcal{C}_i)$.*

This means that data are coded locally at the node with a rate equal to the conditional entropy, where the conditioning is performed *only* on the subset formed by the neighbor nodes which are closer to the sink than the respective node.

The proposed algorithm needs only local information, so it is completely distributed. Still, it will give a solution very close to the optimum since the neglected conditioning is small in terms of rate for a correlation function that is sufficiently decaying with distance (see Section 3.8).

Similar techniques can be used to derive decentralized approximation algorithms for the other particular cases of interests that we discussed in Section 3.5.

3.8 Numerical Simulations

3.8.1 Gaussian Random Fields

A model frequently encountered in practice is the Gaussian random field. This model has also the nice property that the dependence in data at different nodes is fully expressed by the covariance matrix, which makes it more suitable for analysis. Thus, we assume a *jointly Gaussian model* for the spatial data X measured at nodes, with an N-dimensional multivariate normal distribution $G_N(\boldsymbol{\mu}, \mathbf{K})$:

$$f(X) = \frac{1}{\sqrt{2\pi}\det(\mathbf{K})^{1/2}} e^{-\left(\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu})^T\mathbf{K}^{-1}(\mathbf{X}-\boldsymbol{\mu})\right)}$$

where $\mathbf{K}$ is the covariance matrix (positive definite) of $\mathbf{X}$, and $\boldsymbol{\mu}$ the mean vector. The diagonal elements of $\mathbf{K}$ are the variances $K_{ii} = \sigma_i^2$. The rest of the K_{ij}'s depend on the distance between the corresponding nodes (e.g. $K_{ij} = \sigma^2 \exp(-cd_{i,j}^2)$). Then,

for any index combination $I = \{i_1, \ldots, i_k\} \in \{1, \ldots, N\}$, $k \leq N$, $(X_{i_1}, \ldots, X_{i_k})$ is k-dimensional normal distributed. Its covariance matrix is the submatrix $\mathbf{K}[I]$ selected from $\mathbf{K}$, with rows and columns corresponding to $\{i_1, \ldots, i_k\}$.

We use here without loss of generality differential entropy instead of entropy, since we assume that data at all nodes is quantized with the same quantization step, and differential entropy differs from entropy by a constant for uniformly quantized variables [10]. The entropy of a k-dimensional multivariate normal distribution $G_k(\boldsymbol{\mu}, \mathbf{K})$ is:

$$h(G_k(\boldsymbol{\mu}, \mathbf{K})) = \frac{1}{2} \log(2\pi e)^k \det \mathbf{K}.$$

The Slepian-Wolf constraints can readily be expressed as:

$$\begin{aligned} h(\mathcal{Y}|\mathcal{Y}^C) &= h(\mathcal{Y}, \mathcal{Y}^{\mathbf{C}}) - h(\mathcal{Y}^C) \\ &= \frac{1}{2} \log \left((2\pi e)^{N-|\mathcal{Y}^C|} \frac{\det \mathbf{K}}{\det \mathbf{K}[\mathcal{Y}^C]} \right) \end{aligned}$$

where $\mathbf{K}[\mathcal{Y}^C]$ is the selected matrix out of $\mathbf{K}$, with indices corresponding to $\mathcal{Y}^C$ elements respectively.

This natural correlation model is useful for us because our approximation algorithm can be easily tested. We see that in this case, the correlation decays exponentially with the distance. Then, the performance of our approximation algorithm will be close to optimal even for small neighborhoods $\mathcal{C}_i$.

3.8.2 Experiments

We present numerical simulations that show the performance of the approximation algorithm introduced in Section 3.7, for the case of data gathering. We consider the stochastic data model introduced in Section 3.8.1, given by a multi-variate Gaussian random field, and a correlation model where the inter-node correlation decays exponentially with the distance between the nodes.

We use an exponential model of the covariance $K_{ij} = \exp(-cd_{i,j}^2)$, for varying neighborhood range radius and several values for the correlation exponent a. The weight of an edge (i, j) is $w_{i,j} = d_{i,j}^2$ and the total cost is given by the expression (3.6). Figure 3.11 presents the average ratio between the approximated solution and the optimal one. As can be seen, for an exponentially decaying correlation structure, the approximated rate allocation involves only minor increases in terms of general total flow cost as compared to the optimal solution.

In Figure 3.12, we show a comparison of the rate allocations with our different approaches for rate allocation, for a typical random network instance, as a function of the distances from the nodes to the sink. For the explicit communication approach, we used a first-order correlation model, in which in-tree nodes condition only on the closest child, and the value of ρ is computed a-priori as an average. Our experiments have shown that either full conditioning on all children, or use of a distance dependent correlation coefficient between pairs of nodes, do not improve much on the total cost (2.2). The main reason for which the approximated Slepian-Wolf approach outperforms the explicit communication approach is that the Slepian-Wolf approach allocates much smaller values of rates for nodes far from the sink. However, this is

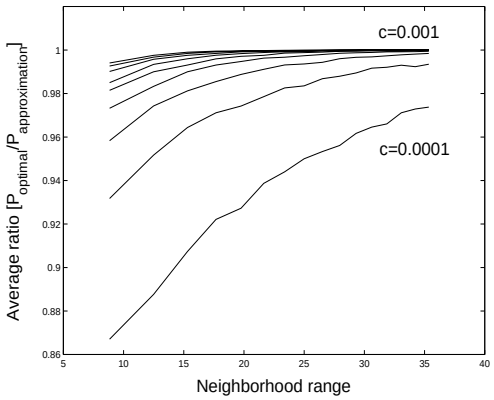


Figure 3.11: Slepian-Wolf coding: average value of the ratio between the optimal and the approximated solution, in terms of total cost, vs. the neighborhood range. The network instances have 50 nodes uniformly distributed on a square area of size 100×100, and the correlation exponent varies from $c = 0.001$ (high correlation) to $c = 0.01$ (low correlation). The average has been computed over 20 instances for each (c, radius) pair.

achieved at the cost of increased network knowledge. We discuss further this issues in the next section, for the case of large networks.

In Figure 3.13 we show a comparison in *average* of our different approaches for rate allocation, as a function of the distances from the nodes to the sink. Note that the slight increase in rate allocation for Slepian-Wolf coding is a boundary effect, namely nodes that are at the extremity of the square grid area have a smaller number of close neighbors on which to condition as compared to nodes which are situated at an intermediate distance from the sink.

3.9 Clustered Slepian-Wolf

Performing Slepian-Wolf coding of the data jointly *at all nodes* in the network is difficult as it involves complex source coding and global knowledge of the network. In this section we consider the case where Slepian-Wolf coding is done locally, on clusters of nodes. This simplifies significantly the coding task, since only knowledge of the local network structure is needed. However, as we will see, the actual task of optimal clustering becomes also complex, thus, showing the inherent complexity that is present in our problem.

Suppose first that a node is part of two clusters. It can use the same random codes, but it needs to use different binning for the two clusters since two independent Slepian-Wolf codings have to be performed. Then, it has to transmit the maximum rate allowable by its local cluster rate allocations. In order to solve the optimizations, the node needs to locally express the inequalities in (3.5). But this means it knows all the weights involved in these constraints, so clearly, the rate allocation can be improved by using the union of the two clusters. This is the reason why we consider only the case of *disjoint* Slepian-Wolf clusters for the rest of this section.

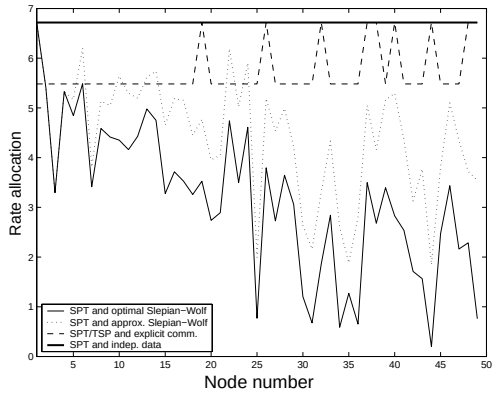


Figure 3.12: Typical rate allocation for a network instance of 50 nodes, and correlation exponent $c = 0.005$. On the x-axis, nodes are numbered in order as the distance from S increases, on the corresponding spanning tree. Total costs for this instance: SPT with optimal Slepian-Wolf: $0.84e + 5$; SPT with approximated Slepian-Wolf: $1.30e + 05$; SPT/TSP with explicit communication (with conditioning on first order children): $1.94e + 5$; SPT with independently coded data: $2.24e + 5$.

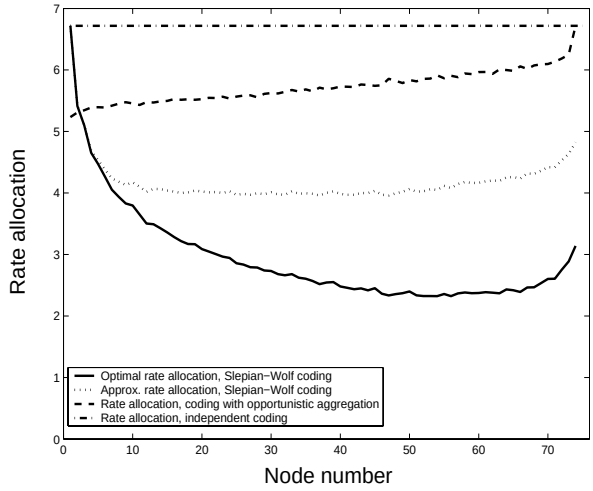


Figure 3.13: Average rate allocation for 1000 network instances of 75 nodes, and correlation exponent $c = 0.0008$ (strong correlation). On the x-axis, nodes are numbered in order as the distance from the sink increases, on the corresponding spanning tree.

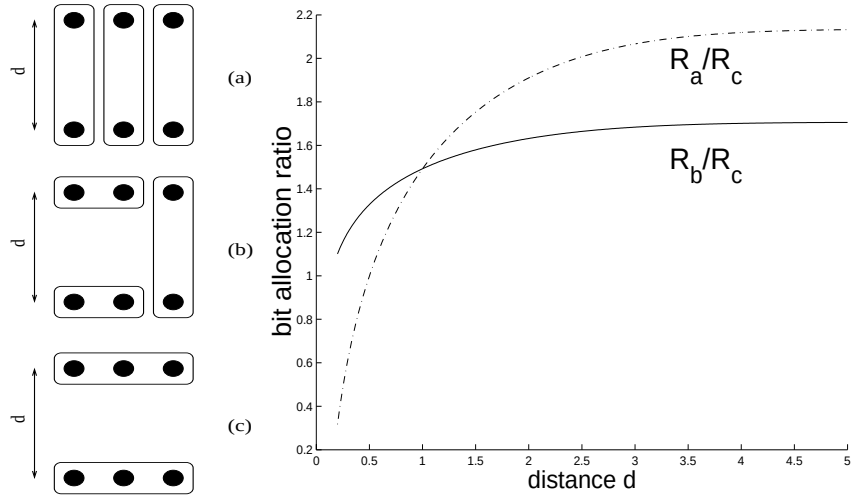


Figure 3.14: Three possible clustering modes for a network of 6 jointly Gaussian nodes (left). The ratio between the optimal bit allocations between the examples in (a),(b) and (c) respectively, as function of the distance d.

For a Gaussian random field, the optimization problem can be posed in a closed form. We allow a maximum cluster size of B. Then the problem is to find an optimal clustering of V that minimizes the weighted sum of rates $\sum_i R_i d_{SPT}(i, S)$ (see Figure 3.14, for $B = 2$). Consider first the particular case when the distances on the SPT are all unity (e.g. nodes placed on an arc of circle). We show that even in this simple case, the problem of optimal clustering becomes complex for $3 \leq B < N$. Denote by $I_1, \ldots, I_C$ the disjoint sets of indices such that $\bigcup I_i = \{1, \ldots, N\}$, with $C \leq N$ the (unknown) total number of clusters, and each cluster can be encoded with $H(X_{I_i})$ bits using Slepian-Wolf in that cluster. The optimization problem is then to minimize over $C, \{I_i\}_{i=1}^{C}$:

$$\sum_i h(G_{|I_i|}(0, \mathbf{K}[I_i])) = \sum_i \frac{1}{2} \log(2\pi e)^{|I_i|} \det \mathbf{K}[I_i]$$

So finally, we need to solve

$$\{I_i^*\} = \arg \min_{C, \{I_i\}_{i=1}^{C}} \left(\prod_{i=1}^{C} \det \mathbf{K}[I_i] \right), \tag{3.13}$$

with $|I_i^*| \leq B, i = 1, \ldots, C$.

Case $B = 1$

Trivial, data are coded separately.

Case $B = 2$

If $B = 2$, the problem (3.4) is equivalent to finding a *minimum weighted perfect matching*, and it can be solved in polynomial time with the Edmonds 'blossom' perfect matching algorithm [19]:

Algorithm 3.3 *Rate allocation for* $B = 2$*:*

- *For each edge* (i, j)*:*
  - *assign* (R_i, R_j) *under* local *constraints:*
    - ∗ **if** $d_{SPT}(i, S) < d_{SPT}(j, S)$
    - ∗ **then** $(R_i, R_j) = (H(X_i), H(X_j|X_i))$,
    - ∗ **else** $(R_i, R_j) = (H(X_i|X_j), H(X_j))$

    (see Section 3.3.1).
  - *let* $d'(i, j) = R_i d_{SPT}(i, S) + R_j d_{SPT}(j, S)$.
- *Run Edmonds algorithm for* $d'(i, j)$.

Case $B = N$

If $B = N$, the optimal cluster division is the cluster formed by the whole set of nodes $(1, \ldots, N)$, which follows from the fact that $\mathbf{K}$ is positive definite, and applying Fischer's inequality [31].

Case $3 \leq B < N$

We show how this becomes very complex, being a particular case of an NP-complete problem, minimum 3-dimensional (3-D) matching. We conjecture the optimization of (3.13) is also NP-complete.

Definition 3.2 *Minimum 3-D matching.*

- Instance*: Three sets* X*,* Y*, and* W *and a cost function* $c : X \times Y \times W \to N$.
- Question*: An assignment* A*, i.e., a subset* $A \subseteq X \times Y \times W$ *such that every element of* $X \cup Y \cup W$ *belongs to exactly one triple in* A.
- Measure*: The cost of the assignment, i.e.,* $\sum_{(x,y,w)\in A} c(x, y, w)$.

The minimum 3-D matching problem is NP-complete, and this extends to the B-dimensional problem. The sets X, Y, W are replaced by V, and the cost $c(\cdot)$ is replaced by the joint entropy of the corresponding nodes.

Suppose there exists a polynomial time algorithm to solve the B-dimensional problem, for $3 \leq B < N$. Then (3.4) could be solved in polynomial time similarly to the 2-dimensional matching case. Unfortunately, such a polynomial time algorithm is unlikely to exist.

Denote by C_M the set of possible M-tuples of nodes (note that a node may appear more than once in a tuple; this allows for clusters of size *at most* M). For each $c \in C_M$, solve the local Slepian-Wolf optimal rate allocation problem R_i. Assign to the tuple

c the cost $\sum_{i \in c} d_{SPT}(i,S)R_i$. Finally, find in polynomial time the set of tuples that minimizes the cost.

From Fischer's inequality, it follows that any refinement[2] of a given clustering increases the total entropy of the cluster set. If global Slepian-Wolf (one cluster of size N) was allowed, it would be optimal, as expected. However, as discussed, this requires knowledge of all the network and the binning process is prohibitively complex.

3.10 Conclusions

In this chapter, we studied the problem of network correlated gathering with Slepian-Wolf coding. In this case, we showed that the two tasks of rate allocation and transmission structure optimization separate. We fully solved the data gathering case by deriving a closed-form solution for the rate allocation. We used our insights for the case of arbitrary traffic matrix and multiple sinks, and we showed that in this case the optimization problem is in general NP-complete. We presented an approximation algorithm for the rate allocation that relies only on local information, and we showed numerical simulations for the Gaussian random field.

Further, we compared the ratio of communication costs involved by the two coding approaches. We provided asymptotic scaling laws for a one-dimensional example. Our results show that if certain conditions on the correlation structure are fulfilled, Slepian-Wolf coding is arbitrarily better in the limit than explicit communication.

Finally, we considered the case when Slepian-Wolf coding is performed on clusters of nodes, namely we showed that the problem of choosing the optimal clustering is a particular instance of an NP-complete problem. This study is a subject of our current research.

Appendix 3.A Proof of Proposition 3.3

Denote $\gamma = \lim_{N \to \infty} \gamma_N$.

3.A.1 Case (a): $a_i \to C$, $C > 0$

Lemma 3.1 *If $g_i \to 0$, and g_i is monotonically decreasing, then:*

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} g_i = 0$$

Proof:

[2]Here by refinement we mean further division of a given clustering into clusters of smaller size.

Since $g_i \to 0$ and g_i is monotonically decreasing, it results that for any $\epsilon > 0$, there exists an integer N_ϵ such that for any $i \geq N_\epsilon$, $g_i < \epsilon$. Then, for any $\epsilon > 0$,

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} g_i &\leq \lim_{N\to\infty} \frac{1}{N}\left[N_\epsilon d_0 + (N - N_\epsilon)\epsilon\right] \\ &= 0 + \lim_{N\to\infty} \frac{(N - N_\epsilon)\epsilon}{N} \\ &= \epsilon\end{aligned}$$

Since this happens for any $\epsilon > 0$, the result follows. □

Lemma 3.2 *Let $a_i \to C$, $C > 0$. Then $\gamma = 1$:*

$$\lim_{N\to\infty} \frac{\sum_{i=1}^{N} i a_i}{\sum_{i=1}^{N}(N - i + 1)a_i} = 1 \tag{3.14}$$

Proof: We can write:

$$\begin{aligned}\lim_{N\to\infty} \frac{\sum_{i=1}^{N} i a_i}{\sum_{i=1}^{N}(N - i + 1)a_i} &= \lim_{N\to\infty} \frac{\sum_{i=1}^{N} i a_i}{(N+1)\sum_{i=1}^{N} a_i - \sum_{i=1}^{N} i a_i} \\ &= \lim_{N\to\infty} \frac{1}{(N+1)\frac{\sum_{i=1}^{N} a_i}{\sum_{i=1}^{N} i a_i} - 1} \qquad (3.15)\end{aligned}$$

Now, since $a_i \to C$, $C > 0$, then we can write:

$$a_i = g_i + C, C > 0,$$

where $g_i \to 0$ and g_i is monotonically decreasing.

Then

$$\begin{aligned}\lim_{N\to\infty} (N+1)\frac{\sum_{i=1}^{N} a_i}{\sum_{i=1}^{N} i a_i} &= \\ &= \lim_{N\to\infty} \frac{(N+1)\sum_{i=1}^{N}(g_i + C)}{\sum_{i=1}^{N} i(g_i + C)} \\ &= \lim_{N\to\infty} \frac{(N+1)\sum_{i=1}^{N} g_i + N(N+1)C}{\sum_{i=1}^{N} i g_i + C\frac{N(N+1)}{2}} \\ &= \lim_{N\to\infty} \frac{\frac{1}{N}\sum_{i=1}^{N} g_i + C}{\frac{1}{N(N+1)}\sum_{i=1}^{N} i g_i + C/2}\end{aligned}$$

We can easily prove that $0 \leq \lim_{N\to\infty} \frac{1}{N(N+1)}\sum_{i=1}^{N} i g_i \leq \frac{1}{N}\lim_{N\to\infty}\sum_{i=1}^{N} g_i$, thus, it is enough to apply Lemma 3.1 and obtain that

$$\gamma = \frac{1}{2-1} = 1.$$

□

3.A.2 Case (b): $a_i \to 0$

Lemma 3.3 *Let $a_i \to 0$. If a_i decreases faster than $1/i$, then $\gamma = 0$. If a_i decreases as fast as $1/i^p, 0 < p < 1$, then $\gamma = 1 - p$.*

Proof:

From (3.15), we see that we have that the limit $\gamma = 0$ iff the ratio

$$\frac{\sum_{i=1}^{N} i a_i}{(N+1)\sum_{i=1}^{N} a_i} \tag{3.16}$$

goes to zero as $N \to \infty$.

- If a_i decreases faster than $1/i$, then $ia_i \to 0$. Then we can use directly Lemma 3.1 for ia_i:

$$\lim_{N\to\infty} \sum_{i=1}^{N} \frac{i a_i}{N+1} = 0$$

Since $\sum_{i=1}^{N} a_i$ is monotonically increasing, the result $\gamma = 0$ follows.

If a_i decreases as fast as $1/i$, then, without loss of generality, take $a_i = 1/i$. Then

$$\gamma = \lim_{N\to\infty} \frac{1}{(N+1)\frac{\sum_{i=1}^{N} 1/i}{N} - 1}$$

But, since $\sum_{i=1}^{N} 1/i$ is divergent, it follows again $\gamma = 0$.

- If a_i decreases slower than $1/i$, suppose without loss of generality that $a_i = 1/i^p$, with $p \in (0,1)$.

Then, by using the integral test, we obtain

$$\begin{aligned}
\lim_{N\to\infty} \frac{\sum_{i=1}^{N} i^{1-p}}{(N+1)\sum_{i=1}^{N} i^p} &= \lim_{T\to\infty} \frac{\int_0^T t^{1-p} dt}{(T+1)\int_0^T t^p dt} \\
&= \lim_{T\to\infty} \frac{\frac{t^{2-p}}{2-p}|_{t=0}^{t=T}}{(T+1)\frac{t^{1-p}}{1-p}|_{t=0}^{t=T}} \\
&= \lim_{T\to\infty} \frac{\frac{T^{2-p}}{2-p}}{(T+1)\frac{T^{1-p}}{1-p}} \\
&= \frac{1-p}{2-p}.
\end{aligned}$$

This means that the sought limit in the case $p \in (0,1)$ is:

$$\gamma = \frac{1}{\frac{2-p}{1-p} - 1} = 1 - p.$$

So far, we have seen that

$$\gamma_N = \frac{1}{(N+1)\frac{\sum_{i=1}^N a_i}{\sum_{i=1}^N i a_i} - 1}$$

converges to zero if ia_i goes to zero in the limit $i \to \infty$. Suppose without loss of generality $a_i = 1/i^p$, so the condition for $\lim_{N\to\infty} \gamma_N = 0$ is $p \geq 1$.

We note that the rate of decay of the ratio of costs is directly related with the rate of increase with N of the partial sum $s^N(q) = \sum_{i=1}^N \frac{1}{i^q}$ for $q > 0$, namely:

- $s^N(q) = \Theta(N^{1-q})$, if $q \in (0,1)$;
- $s^N(q) = \Theta(\log N)$, if $q = 1$;
- $s^N(q) = \Theta(1)$, if $q > 1$.

We thus obtain:

Case $p = 1$:

In this case, $\gamma_N = \Theta(\frac{1}{N\frac{\log N}{N} - 1}) = \Theta(\frac{1}{\log N})$.

Case $p \in (1,2)$:

In this case, $\gamma_N = \Theta(\frac{1}{N\frac{1}{N^{2-p}} - 1}) = \Theta(\frac{1}{N^{p-1}})$.

Case $p = 2$:

In this case, $\gamma_N = \Theta(\frac{1}{N\frac{1}{\log N} - 1}) = \Theta(\frac{\log N}{N})$.

Case $p > 2$:

In this case, $\gamma_N = \Theta(\frac{1}{N-1}) = \Theta(\frac{1}{N})$.

□

Appendix 3.B Conditional Entropy for Correlation Law $\exp(-c\tau)$

Consider the one-dimensional example in Figure 3.8. For the sake of simplicity, assume that the variance $\sigma^2 = 1$. The correlation between nodes l and j is thus $K_{lj} = \exp(-c|l-j|d)$. Denote by $\rho = \exp(-cd)$. Then we can write the covariance matrix of any i consecutive nodes on the line as:

$$K^i = \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{i-1} \\ \rho & 1 & \rho & \dots & \rho^{i-2} \\ \dots & \dots & \dots & \dots & \dots \\ \rho^{i-2} & \rho^{i-1} & \dots & 1 & \rho \\ \rho^{i-1} & \dots & \rho^2 & \rho & 1 \end{pmatrix}$$

and thus their joint differential entropy is:

$$h(X_1, \ldots, X_i) = \log(2\pi e)^i \det K^i = \log(2\pi e)^i (1-\rho^2)^{i-1}$$

It follows that we can write the conditional differential entropy as:

$$\begin{aligned} h(X_i|X_{i-1}, \ldots, X_1) &= h(X_i, X_{i-1}, \ldots, X_1) - h(X_{i-1}, \ldots, X_1) \\ &= \log(2\pi e)^i (1-\rho^2)^{i-1} - \log(2\pi e)^{i-1}(1-\rho^2)^{i-2} \\ &= \log \frac{(2\pi e)^i (1-\rho^2)^{i-1}}{(2\pi e)^{i-1}(1-\rho^2)^{i-2}} \\ &= \log(2\pi e)(1-\rho^2) \\ &= h(X_i|X_{i-1}) \end{aligned}$$

which depends on the inter-node distance d, but not on the number of nodes i.

Since the conditional entropy $a_i = H(X_i|X_{i-1}, \ldots, X_1)$ differs in approximation by only a constant from the conditional differential entropy, it follows $a_i \approx H(X_i|X_{i-1})$, for any i.

Chapter 4

Flow Control for Multiple Access Queues

4.1 Preliminaries

4.1.1 Bits Through Multiple Access Queues

Let $S^{(1)}...S^{(N)}$ denote N information sources. Each $S^{(i)}$ can be in one of two possible states: when in the ON state, symbols (drawn from a finite alphabet) are generated according to some unspecified distribution; when in the OFF state, no symbols at all are generated. Source state transitions between ON/OFF states are independent over time. The symbols generated by each $S^{(i)}$ are placed in the buffer of a single-server queue, which serves them using some predetermined scheduling algorithm (typically, first-in first-out). In this proposed scenario, two important questions arise:

- *What is a fair split of the service rate of the queue among the different sources?* This is essentially equivalent to the classical problem of control in networks. In a network with several users and interconnecting nodes, the need for control (i.e., for controlling the amount of data that each source is allowed to inject into the network) arises because of the limited available network resources. The purpose of flow control is to allocate these resources efficiently, while keeping this allocation *fair* among users. Classical papers on this subject are [22, 33], among others.

- *What is the Shannon capacity of the channel available to each source?* In [3], this question is answered in a special case of the above described setup: a single source (i.e., $N = 1$), always in an ON state.

In this work we formulate and present some first steps towards solving a multiuser version of the Bits Through Queues problem of [3], which we refer to as the *Bits Through Multiple-Access Queues* problem. Note that although certainly related, our problem is significantly more complex than that considered in [3]. To start with, under the assumptions of [3] the first question is meaningless, since there is only one source. In the general case, the mechanism used to split the queue resources among

information sources will essentially determine the channel available to each one of them, and hence its capacity. Further complications arise from the fact that, in the general case, the number of active sources changes over time. A study of flow control techniques suitable for use in the context of the multiuser version of the problem considered in [3] is the main focus of this work.

4.1.2 The Shannon Capacity of Queuing Models

An important result of [3] is that, assuming 1-bit packets, the Shannon capacity of a single-server queue with exponential service is $\frac{2}{e}\log_2(e)\mu \approx 1.0615\mu$, where μ is the average service rate. Note that loading the queue at a rate $\lambda > \mu$ will clearly not achieve this goal, since this results in the system becoming unstable, and still, the output rate would be not more than μ. The answer to this apparent paradox stands in the *timing* information.

Consider the following simple example. Assume a queue as illustrated in Figure 4.1, with service times s having an arbitrary mean μ. Assume first that the variance of s is zero. Choose a real number $S > \mu$, inject one packet at time 0, another at time S. Then, the first packet departs at time μ, the second at time $\mu + S$, and the receiver can recover S exactly, by subtracting the first departure time from the second. But since S is a real number that was communicated without distortion, we immediately see that this queue has *infinite* capacity, even if packets are of finite size and serving them requires nonzero time.

Figure 4.1: The queueing model considered in *Bits Through Queues* [3]: an infinite buffer, a process of arrivals with mean λ, service times with mean μ, and packets of fixed length.

In the example above, when $s > \mu$ and if the zero variance condition is fulfilled, then the receiver can recover s without distortion. However, in realistic scenarios, departures are random, meaning that the variance of μ is different from zero. Thus, the rate at which information can flow through the system will be lower. This uncertainty is the source of noise for communication based on timing codes. Therefore, we can formulate the basic tradeoff between transmission of information based on timing codes and based on classical data codes:

- Communication via packet contents resulting in a heavily loaded queue, and no packet information, vs.
- Communication via idle periods resulting in frequent idle periods, and no timing information.

From [3], it follows that in the context of queueing models, a system that is efficient in an information theoretic sense must be operated in an 'almost' unstable regime (in the sense of stochastic stability, in this case corresponding to $\lambda \approx \mu$).

4.1.3 Bits Through Shared Queues

There is a "natural" multiuser generalization of the setup considered in [3], consisting of multiple sources sharing the same queue, as illustrated in Figure 4.2, which shows N sources sharing a single finite buffer. When each source can observe the state of the entire network, this problem degenerates to the single source case. The interesting case however occurs when sources only have partial information about the state of the system. In this case, they must base decisions about when to access the channel only on that partial data.

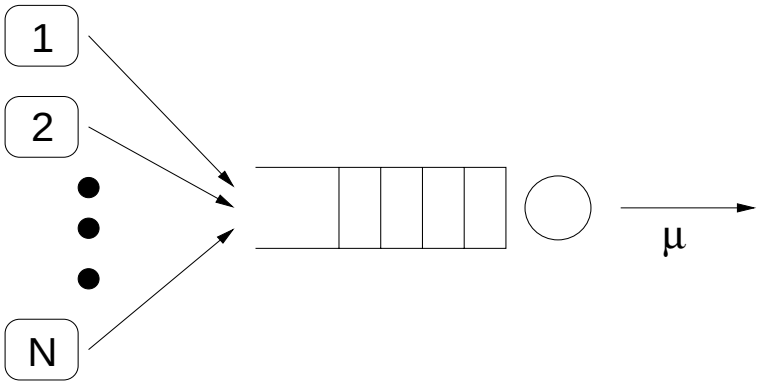


Figure 4.2: The problem of N sources sharing a single finite buffer.

New interesting questions arise in this setup, questions that make no sense in the original formulation of [3]. For example: how do sources gain access to the shared queue? What information do sources have available to make these decisions? What are relevant performance bounds, taking multiple users into account? To provide answers to some of these questions, we must deal with a number of questions which are not only relevant and interesting, but also difficult:

- The problem of how different sources gain access to the shared queue is an abstraction of the thoroughly studied congestion problem in networks. Many practical and well debugged algorithms have been developed over the years [22, 33], and more recently formulations of this problem have taken more analytical approaches, based on game theoretic, optimization, and flows-as-fluids concepts [36, 39, 44, 45]. Queueing models however have not found application yet in this context, and we believe valuable insights could be obtained from taking a queueing and stochastic control view of this problem.

- Even if centrally designed (in the sense that all nodes are assumed to execute the same control algorithm), decentralized control problems are much less understood than their centralized counterpart. These are problems deemed in general both relevant and difficult.

Given all these considerations, our main goal in this work is to set up a "toy" problem, which is analytically tractable, and which captures in a clean manner some of these issues.

To the best of our knowledge, the literature on *controlled* queues is limited. The main problem in analyzing the behavior of queueing models incorporating control is that the feedback loop results in the introduction of dependencies in the arrivals and/or service processes. However, most queueing models that have been successfully analyzed to date involve processes with independent increments, an assumption clearly violated in the presence of feedback control: decisions made about current packets affect decisions to be made about future packets. Our main contribution presented in this work lies in having been able to formulate, analyze, and simulate a queueing model which explicitly accounts for the type of dependencies introduced by feedback control.

In this chapter we set up a model of a queueing system in which multiple sources compete for access to a shared buffer. We describe its dynamics, and we formulate and solve an appropriate stochastic control problem. We also present results obtained in numerical simulations, to illustrate with concrete examples properties of these control boxes.

Then, in Chapter 5, we study ergodic properties of the queueing model that results from operating the system described in this chapter under closed-loop control. There, we show how long-term averages are described in terms of a suitable invariant measure. We first prove the existence of that measure, and then show how to effectively construct it.

4.2 The Control Problem

4.2.1 Distributed Flow Control with Partial Information

In the design of the desired control modules, there is a wide range of options in terms of information available to the controller, to the sources, to the queue itself, etc. Two extreme examples correspond to cases when (a) there is a unique, global, central controller which can observe exactly the state of the queue and of all the sources at any point in time; and (b) a decentralized, local controller which can only see the state of the individual source it controls, as well as some feedback information that it obtains about the state of the queue. This situation is illustrated in Figure 4.3.

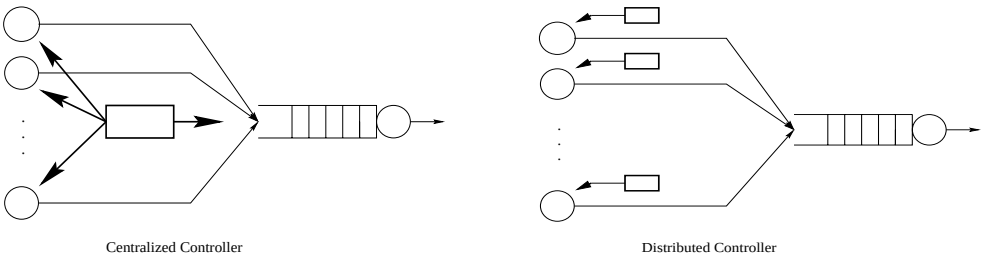


Figure 4.3: Two extreme cases in which control may be necessary. Left: a unique centralized global controller; right: local decentralized controllers. In general, global controllers are desirable since they will give the best network performance. The problem is that their communication complexity often renders them prohibitively complex in practice, thus there is interest in local decentralized control devices.

We see in Figure 4.3 that, for the centralized controller, the solution to any reasonable formulation of the flow control problem is trivial: knowing all the information about the state of individual sources and of the buffer state, it is enough to allocate to each user a service share equal to the ratio between the service rate and the number of sources active at a given moment. The second type of controller, more realistic but harder to implement, is what we focus on in this work. Our goal is to design a flow control mechanism, simple enough to yield to mathematical analysis in the context of the *Bits Through Multiple-Access Queues* problem, yet also rich enough to be a good model for real-life situations.

4.2.2 System Model and Dynamics

Consider the following discrete time model [15], [17]:

- N sources feed data into the network, switching between ON/OFF states in time. While ON, source $S^{(i)}$ generates a symbol at time k with probability $u_k^{(i)}$, and remains silent with probability $1 - u_k^{(i)}$; while OFF, the source remains silent with probability 1. Given the intensity value u, the switch between states is independent of everything else.
- The queue has a finite buffer. When a source generates a symbol to put in this buffer, if the buffer is full then the symbol is dropped and the source is notified of this event; if there is room left in the buffer the symbol is accepted, and the source is notified of this event as well. Note: feedback is sent *only* to the source that generates a symbol, and not to all of them.
- The control task consists of choosing values for all $u^{(i)}$'s, at all times. A basic assumption we make is that sources are *not* allowed to coordinate their efforts in order to choose an appropriate set of control actions $u^{(i)}$ $(i = 1...N)$: instead, the only cooperation we allow is in the form of having all sources implement the same control technique, based on feedback they receive from the queue.
- The service rate of the queue is deterministic. Note however that, unlike in the motivating example from [3] presented in Section 4.1.2, in our model this does not lead to a system with infinite capacity. Interference from other users (in the form of random numbers of packets in the queue in between consecutive packets of any individual source) leads to uncertainty in the departure times, and therefore to finite capacity.

An illustration of this proposed model is shown in Figure 4.4.

The dynamics of this system are modeled as follows:

- $x_k \in \mathcal{S} = \{1, \ldots, N\}$: number of ON-sources at time k, modeled as a finite state Markov chain with known matrix of transition probabilities P_{ij} given by $\Pr(x_k = j | x_{k-1} = i)$ (independent of the source intensities $u_k^{(i)}$ and of the time index k), and known $\Pr(x_0)$ (the initial distribution over states).
- $r_k^{(i)} \in \mathcal{O} = \{-1, 0, 1\}$: ternary feedback from the queue to the source. The convention we use is that -1 denotes losses, 0 denotes idle periods, and 1 denotes positive acknowledgments.

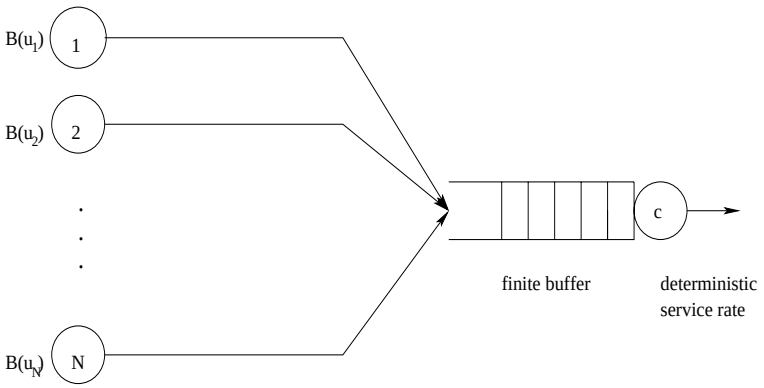


Figure 4.4: The proposed model. N sources switch between ON/OFF states, and generate symbols with a (controllable) probability $u_k^{(i)}$. The only information a source has about the network is a sequence of 3-valued observations: acknowledgments, if the symbol was accepted by the buffer, losses if it is rejected due to overflow, and nothing if the decision was not to transmit at the current moment (denoted 1, -1, 0, respectively).

- $u_k^{(i)} \in \mathcal{U}$, where $\mathcal{U} = (0, 1]$ source intensities, controllable (as defined above).
- $p(r|x, u)$: the probability of occurrence of an observation $r \in \mathcal{O}$, when x sources are active, and when symbols are generated by all active sources at an average rate u.

These dynamics are illustrated in Figure 4.5.

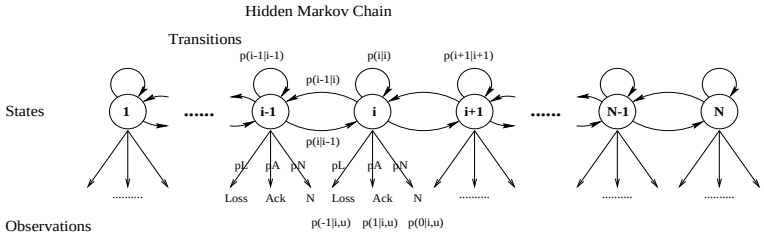


Figure 4.5: An illustration of the model from the point of view of a single source, based on a simple birth-and-death chain for the evolution of the number of active sources.

There are two important observations to make about how we have chosen to set up our model. Describing the probabilities of observations $p(r|x, u)$ only in terms of the number of active sources x and the *average* injection rate u of all the active sources does require some justification: how can we assume that all sources inject the same amount of data, when the data based on which these decisions are made (feedback from the queue) is not shared, that is each source gets its own private feedback? Although this might seem unjustified, that is not the case. Once we study in some detail the control problem we are setting up here, we will find that the optimal control

action u_k at time k is given by a memoryless function $u_k = g(\pi_k)$ of a random vector π that has the same distribution for *all* sources, and with well defined ergodic properties. The study of these ergodic properties is the subject of Chapter 5. Therefore, even though at any point in time there will likely be some sources getting more and some other sources getting less than their fair share, on average all sources get the same. This issue is further discussed below, both analytically (in Chapter 5) and in terms of numerical results (in Figure 4.10).

Another important point to note is that there are strong similarities between our model and the formalization of multiaccess communication that led to the development of the Aloha protocol [8]. However, the fact that feedback is not broadcast to all active sources in our model is a major difference between our formulation and that one. In fact, we conceived our model as an analytically tractable "hybrid" between Aloha and the standard Transport Control Protocol (TCP). Like in slotted Aloha, time is discrete, feedback is instantaneous, and the state follows a Markovian evolution; but like in TCP, feedback is private only to the source that generated a transmitted packet, and the buffer is finite.

As an application for some of the results and proof techniques developed in this work, we are looking in Section 5.5 at their implementation in the context of Aloha with private feedback and random arrivals. Hajek [29] reviews a series of results for the two usual models for Aloha (finite user, one packet at a time, and infinite number of users [8]). Decentralized policies for the injection probabilities, that maintain stability in the case of private acknowledgment feedback, are hard to be derived for the infinite nodes case with Poisson arrivals. There is however important work [29] about stability in the finite nodes study of Aloha. The theory in [29] is applied, as an example, to finding conditions of stability for multiplicative policies for sources that are supplied with Poisson arrivals. The theory we develop in this work provides a useful background for an Aloha model with random arrivals (not necessarily Poisson), with a finite number of backlogged packets, and its extension to the infinite user model.

4.2.3 Formal Problem Statement

Intuitively, what we would like to do is maximize the rate at which information flows across this queue, subject to the constraint of not losing too many packets. Since each time we attempt to put a packet into the shared buffer there is a chance that this packet may be lost, it seems intuitively clear that without accepting the possibility of losing packets, the throughput that can be achieved will be low. At the same time, we do not want a high packet loss rate, as this would correspond to a highly unstable mode of operation for our system.

This intuition is formalized as follows. Our goal is to find a policy g that solves

$$\max_{g} \ \limsup_{K\to\infty} \ \frac{1}{K}\sum_{k=1}^{K} \mathtt{Pr}(r_k = 1|x_k, u_k) \tag{4.1}$$

$$\text{subject to:} \quad \mathtt{Pr}(r_k = -1|x_k, u_k) \le T, \ \forall k,$$

where $T \in (0, 1]$ is a parameter that specifies the maximum acceptable rate of packet losses, and K is the time horizon. If there was no loss threshold T, then all sources could inject packets with probability one, get on average their fair share, but in the

same time this would involve a very large amount of losses (losses will happen with probability $p(r = -1|x, u = 1) = 1-1/x$).In Figure 4.9, on numerical simulations, we illustrate how this parameter affects the behavior of the controller. Note that we use a lim sup in the definition of our utility function (instead of a regular limit) because we don't know yet that the limit actually exists. This limit does exist, as we will show in Chapter 5.

4.2.4 Finite Horizon and Observed State

We start with the solution to an easier version of our control problem: one in which the state of the chain (i.e., the number of active sources at any time) is known to all the sources. Although this is *not* a reasonable assumption to make (it does trivialize the problem), we find that looking at the solution to the general problem in this specific case is actually quite instructive. We start here as a step towards the solution of the case of true interest (hidden state).

The problem formulated above is a textbook example of a problem of optimal control for controlled Markov chains, and its solution is given by an appropriate set of Dynamic Programming equations [38]. Define $\mathbf{c}(u) = [\mathtt{Pr}(1|i, u)...\mathtt{Pr}(1|N, u)]^\top$, and then:

$$\begin{aligned} V_K(i) &= 0 && (4.2)\\ V_k(i) &= \sup_{u\,:\,\mathtt{Pr}(-1|i,u)\le T} \{\mathbf{c}(u) + P(i,u)V_{k+1}\} \\ &= \sup_{u\,:\,\mathtt{Pr}(-1|i,u)\le T} \{\mathbf{c}(u) + C\} \ \ (C \text{ independent of } u). && (4.3)\end{aligned}$$

Equation (4.2) is set to 0 because this is only a finite horizon approximation, but we are interested in the infinite horizon case, and in this case the boundary condition given by $V_K = 0$ has a vanishing effect as we let $K \to \infty$.

What is more interesting is that from (4.3) it follows that a *greedy* controller is optimal: this is not at all unexpected, since in our model the transition probabilities P are not affected by control, only observations are. The interplay among control and the different probabilities of observations is illustrated in Figure 4.6 (see also Appendix 4.A). Consider a fixed (observed) state i, and for simplicity assume a large shared buffer (these curves can be replaced by curves derived from large deviations estimates such as given by the Chernoff bound). Then, our assumptions are as follows. The probability of a packet loss is zero until the injection rate hits the fairness point $1/i$, beyond which it increases linearly. The probability of a packet finding available space in the shared buffer increases linearly up until the fairness point $1/i$, beyond which it remains constant. Note that $u^* > \frac{1}{i}$ is the largest $u \in (0, 1]$ such that $\mathtt{Pr}(-1|i, u) \le T$. The gap between $1/i$ and u^* is the "margin of freedom" we will have to risk the loss of packets, in the case when i cannot be observed.

4.2.5 Partial Information

The case of partial information (i.e., when the underlying Markov chain cannot be observed directly) poses new challenges. The problem in this case is that Markovian control policies based on state estimates are not necessarily optimal. Instead, optimal policies satisfy a "separation" property, illustrated in Figure 4.7 and extensively

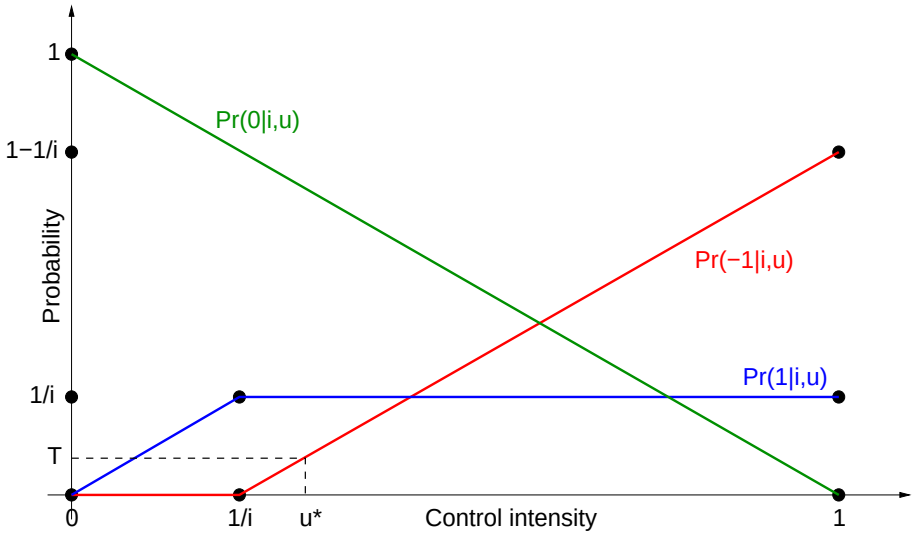


Figure 4.6: Probabilities of observation as a function of the control and the number of states.

discussed in [38]. Suppose we have a controlled system, which produces certain observable quantities related to its unobserved state. Based on these observations, we compute an *information state*. The information state is a quantity that somehow must capture all inference about the state of the system, given all the information we have seen so far (this concept will be made rigorous later). This information state is fed into a control law, that uses it to make a decision of what control action to choose, and this action is fed back into the system.

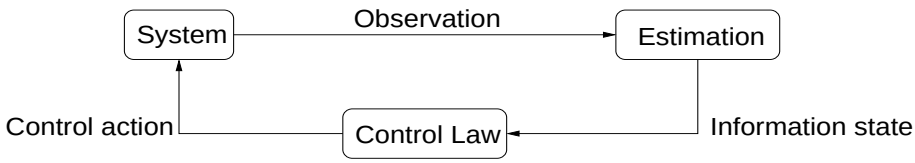


Figure 4.7: The separation of estimation and control.

Formally, an information state π_k is a function of the entire history of observations and controls $r_0 ... r_{k-1} u_0 ... u_{k-1}$, with the extra requirement that π_{k+1} can be computed from π_k, r_k, u_k. Note that this is a reasonable requirement to make about a quantity that captures some notion of state for our system. A typical choice is to let π_k be $\Pr(x_k | r^{k-1}, u^{k-1})$, the conditional probability of x_k given all the past observations and applied controls.

Definition 4.1 *Denote the simplex of N-dimensional probability vectors by* $\Pi =$

$\{(p_1, \ldots, p_N) \in \mathcal{R}^N : p_i \geq 0, \sum_{i=1}^N p_i = 1\}$.

Then, an optimal controller for partially observed Markov chains also satisfies a set of Dynamic Programming equations, but instead of this set of equations being over the states of the chain (a finite number), these equations are defined over information states [38] (i.e., over all points in the simplex of Π probabilities over N points):

$$\begin{aligned} V_K(\pi) &= 0 \\ V_k(\pi) &= \sup_{u \,:\, E_\pi \mathtt{Pr}(-1|i,u) \leq T} \mathtt{E}_\pi \left\{ \mathbf{c}(i,u) + V_{k+1}(F[\pi,u,r]) \right\}, \end{aligned} \tag{4.4}$$

where F denotes the recursive updates of π, and where the notation $\mathtt{E}_\pi$ denotes expectation relative to the measure π. A straightforward derivation gives the information state transition function F (see Appendix 4.B):

$$\begin{aligned} \pi_{k+1} &= F[\pi_k, u_k, r_k] \\ &= C_{\pi_k} \cdot \pi_k \cdot D(u_k, r_k) \cdot P \end{aligned}$$

with C_{π_k} a normalizing constant, P the transition probability matrix of the underlying chain, and $D(u,r) = \mathrm{diag}\left[\mathtt{Pr}(r|1,u) \ldots \mathtt{Pr}(r|N,u)\right]$ a diagonal matrix. This is essentially the same set of Dynamic Programming equations as before, but where dependence on states is removed by averaging with respect to the current information state π_k. As before, the optimal controller is chosen by recording for each π the value of u that achieves the supremum in the right hand side of eqn. (4.4). The optimal control will thus be a function of only the information state, $u = g(\pi)$.

4.2.6 Infinite Horizon

In the previous subsections we derived the solution for the optimal control in the case of partial observations when the time horizon is finite. We can get back now to the infinite horizon problem stated in (4.1). The dynamic programming algorithm becomes a fixed-point system of equations with the unknowns spanning the simplex Π. Indeed, let us start from the finite horizon case:

$$V_K(\pi) = \sup_{u \,:\, E_\pi \mathtt{Pr}(-1|i,u) \leq T} E_\pi \left[c(i,u) + V_{K-1}(F[\pi,u,r_k]) \right] \tag{4.5}$$

We rewrite (4.5) as [38]:

$$\frac{V_K(\pi)}{K} = \sup_{u \,:\, E_\pi \mathtt{Pr}(-1|i,u) \leq T} E_\pi \left[c(i,u) + V_{K-1}(F[\pi,u,r_k]) - V_K(\pi) + \frac{V_K(\pi)}{K} \right] \tag{4.6}$$

Assume the following limits exist for all $\pi \in \Pi$ and some J^*:

$$\lim_{K \to \infty} \left(V_K(\pi) - K J^* \right) = V_\infty(\pi)$$

Then by taking the limit $K \to \infty$ in (4.6) we finally get[38]:

$$J^* + V_\infty(\pi) = \sup_{u \,:\, E_\pi \mathtt{Pr}(-1|i,u) \leq T} E_\pi \left[c(i,u) + V_\infty(F[\pi,u,r_k]) \right] \tag{4.7}$$

The DP equation in (4.7) holds actually under more general conditions that are easy to verify for our model [38]. The transition probability matrix P does not depend in our model on the control policy. Further, the Markov chain given by the number of active sources is irreducible in normal circumstances. Then it is shown in [38] that if these conditions are fulfilled, then the dynamic programming equation system for the average cost criterion is as in (4.7) and there exists $V(\pi), \pi \in \Pi$ and J^* that solves it. Also, J^* is the minimum average cost and a policy g is optimal if $g(\pi)$ attains the minimum in (4.7).

One might attempt to solve the fixed-point system in (4.7) with an iterated algorithm on a discretized version of the equations system. However, there are practical difficulties to implement and simulate the optimal controller in the partial information case as defined above, having to do with the fact that our state space is the whole simplex of probability distributions Π. Our approach to find an approximate solution for the optimization problem (4.1) is to solve the dynamic programming system for the finite horizon case (finite K), and study the properties of the obtained control policy by numerical simulations.

4.3 Numerical Simulations

To help develop some intuition for what kind of properties result from the optimal control laws developed in previous sections, in this section we present results obtained in numerical simulations. Our approximation consists in choosing the maximum control at time k that still obeys the loss constraint, since this will also maximize the throughput.

In Figure 4.8 we present a typical evolution over time of the information state. This plot corresponds to a symmetric birth-and-death chain as shown in Figure 4.5, with probability of switching to a different state $p = 0.001$, $N = 10$ sources, and loss threshold $T = 0.04$. At time 0, the initial π_0 is taken to be $\pi^s(i) = \frac{1}{N}$, the stationary distribution of the underlying birth-and-death chain. While there are no communication attempts (up until time $k = 6$), π_k remains at π^s. Then at time 6 a packet is injected into the network and it is accepted, and as a result, there is a shift in the probability mass towards the region in which there is a small number of active sources. Then at time 19 another communication attempt takes place but this time the packet is rejected, and as a result, now the probability mass shifts to the region of a large number of active sources. This type of oscillations have been observed repeatedly, and gives a pleasing intuitive interpretation of what the optimal controller does: keep pushing the probability mass to the left (because that is the region where more frequent communication attempts occur, and thus leads to maximization of throughput), but dealing with the fact that losses push the mass back to the right. Similar oscillations are also typical of linear-increase multiplicative-decrease flow control algorithms such as the one used in TCP.

In Figure 4.9, we illustrate how different values of the threshold T influence the behavior of the controller. In this case we consider the same birth-and-death model considered in Figure 4.8, with three different values for T: top, $T = 0.1$; center, $T = 0.02$; bottom, $T = 0.05$. In all plots, the horizontal axis is time, the vertical axis is control intensity, and two controllers are shown: the thick black line corresponds to our optimal control law, the thin dotted line corresponds to a genie-aided controller

that can observe the hidden state. And we observe a number of interesting properties: (a) when T is large (top plot), our optimal control stays most of the time above the fair share point determined by the actions of the genie-aided controller; (b) also when T is large, we see that sudden increases in bandwidth are quickly discovered by our optimal law; (c) when T (middle plot) is small, the gap between the control actions of our optimal law and the genie-aided law is smaller, but our law has a hard time tracking a sudden increase in available bandwidth; (d) for intermediate values of T (bottom plot), both the size of the gap and the speed with which changes in available bandwidth can be tracked is in between the previous two cases. These plots also suggest another intuitively pleasing interpretation: T is a measure of how "aggressive" our optimal control law is.

In Figure 4.10 we address the fairness issue raised at the end of Section 4.2.2. In this case we also consider a birth-and-death chain model as in previous examples, but now with only two sources ($N = 2$). In the top figure, we show the maximum and the minimum control value chosen by either one of the sources over time: thick black line shows the minimum, thin solid lines shows the maximum (for reference, the genie-aided controller is also shown); in the bottom figure, the thick line corresponds to the control actions of only one of the sources, all the time. Observe how, around time steps 150-250, the source shown at the bottom is the one that achieves the *maximum* at the top; but around time steps 500-600, the same source achieves the *minimum* of those injection rates. This is yet another intuitively pleasing pattern that we have observed repeatedly in many simulations: the control law is essentially fair in the sense that, whereas we do not have enough information to make sure that at any time instant all controllers will use the same injection rate, at least over time the different controllers "take turns" to go above and below each other.

4.4 Conclusions

In this chapter we studied an optimal controller that can be used by multiple sources to gain access to a shared queue. The sources are independent and cannot communicate with each other. We provided the structure of an optimal controller that determines the injection rate of each source. Our controller relies only on the sequence of private actions and feedback obtained from the router; this information is gathered in the *information state*, namely the vector of probabilities of the system being in a certain state given the sequence of all past controls and observations. We showed how the information state can be computed recursively using only the instantaneous pair (control, observation). Our extensive numerical simulations show that our control policy achieves also fairness for the sharing of the bandwidth among the sources.

Appendix 4.A Probabilities of Observation

We consider a simplified model for the probability of observation: we suppose the buffer is sufficiently large so that losses appear only when the source attempts to transmit with a rate above the fair share, and not because of occasional collisions with other sources transmitting also below the fair share, due to the random behavior of the controller. Numerical experiments showed that a finite buffer size is enough

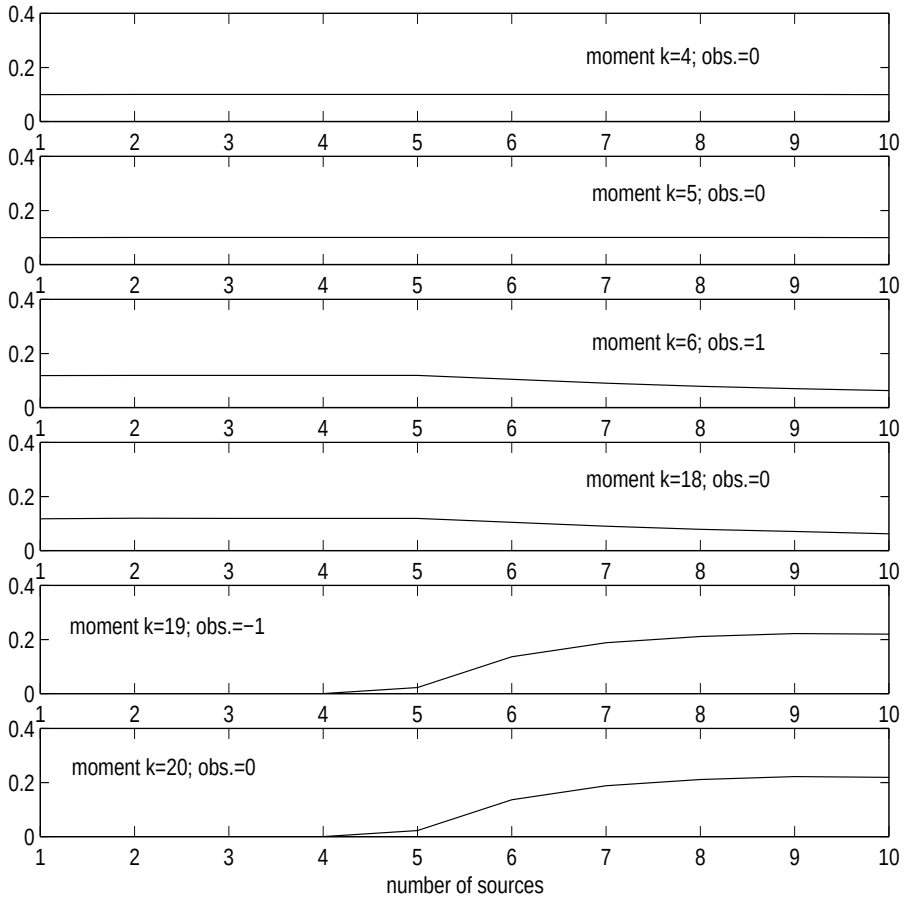


Figure 4.8: Typical dynamics of π. There are $N = 10$ sources, and this plot illustrates the evolution of the information state vector, as being updated by the source of interest. At time $k = 6$ a packet is sent by the source and accepted in the buffer, thus the result of the positive acknowledgment is a shift in probability mass towards the region corresponding to a small number of active sources. The reverse effect is observed at time $k = 19$, when a loss happens.

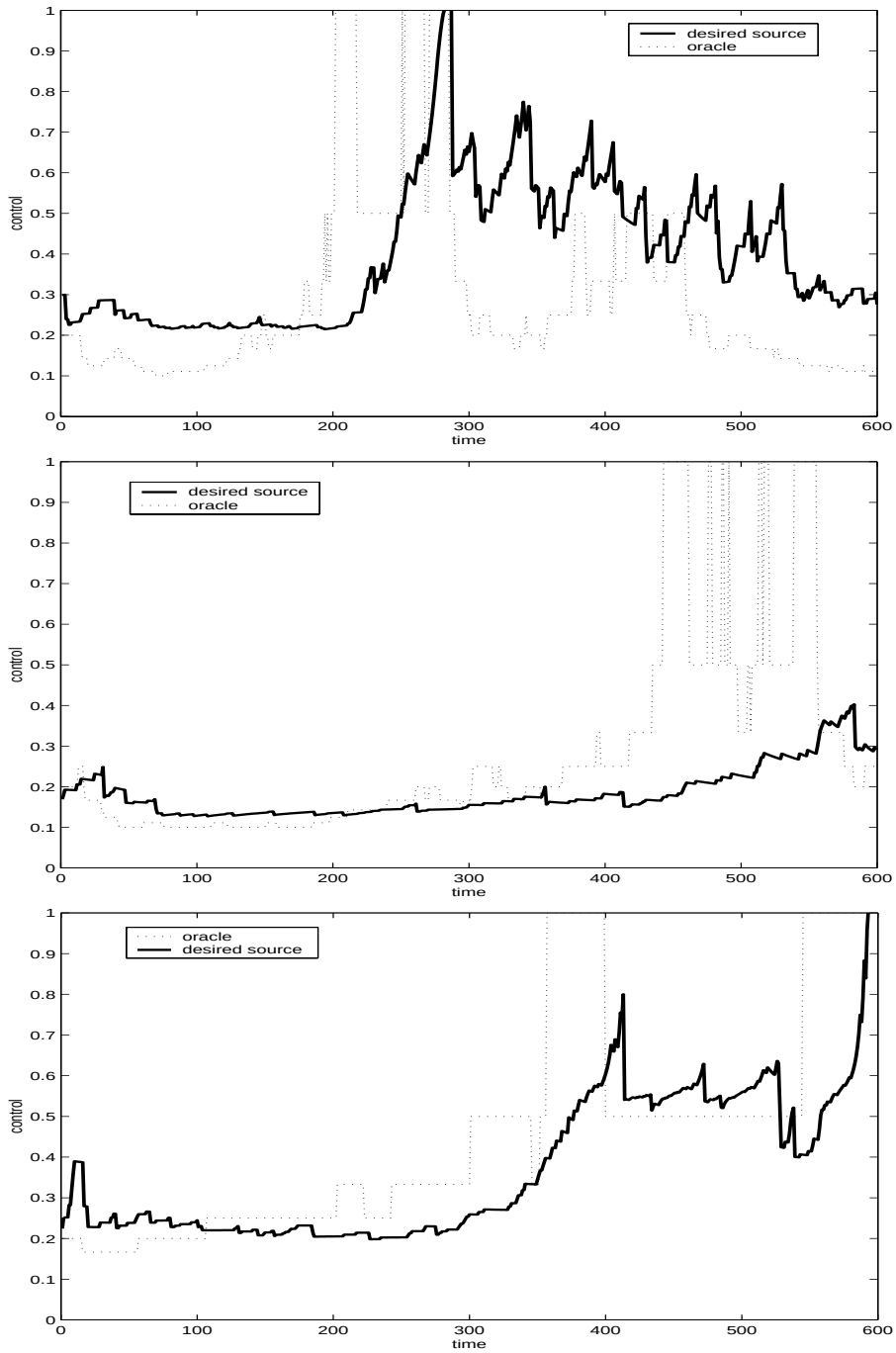


Figure 4.9: Illustration of how the value of the loss threshold T affects the optimal control law. A large value of the loss threshold $T = 0.1$ (up) results in quick reaction of the control to changes in the environment conditions, at the expense of a large number of losses, due to the operation above the fair share. A small value of the loss threshold $T = 0.02$ (middle) results in slow response to positive environment conditions, but in the same time the control operates close to the fair share. The plot (down) illustrates the behavior of the control for a medium threshold value $T = 0.05$.

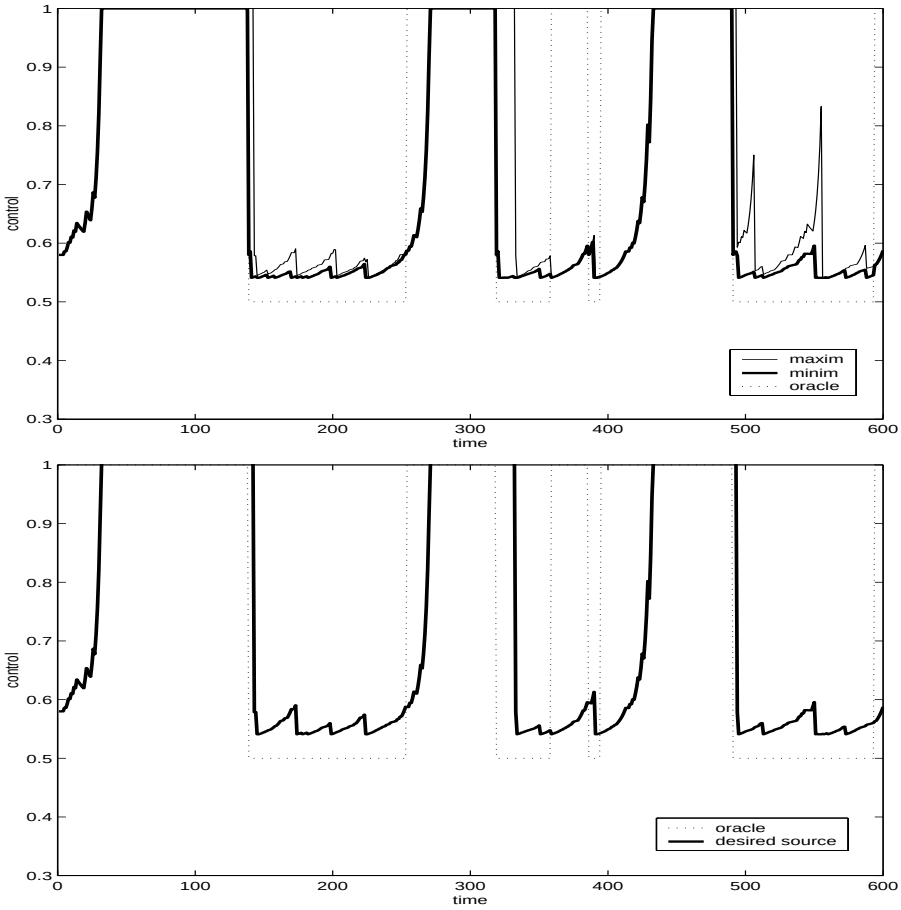


Figure 4.10: The fairness issue raised at the end of Section 4.2.2: maximum and minimum value of the two controls (up); the control implemented by the source of interest (down). We notice that both sources take turns in using a larger bandwidth than the other.

for this condition to be met approximately. For instance, for a maximum number of $N = 10$ active sources, a buffer size $B = 2$ gives numerical results which are close to the ideal case.

In the ideal case, we would have:

$$p(1|x,u) = \begin{cases} u, u < 1/x \\ 1/x, u \geq 1/x \end{cases}$$

$$p(0|x,u) = 1 - u,$$

and

$$p(-1|x,u) = \begin{cases} 0, u < 1/x \\ u - 1/x, u \geq 1/x \end{cases}$$

Exact formulae for the observation probabilities can be derived using the Chernoff bound or by numerical simulations, for the finite buffer case, or even in the case of a more general network topology.

Appendix 4.B Information State Recurrence

Denote $u^k = (u_0, u_1, \ldots, u_k), r^k = (r_0, r_1, \ldots, r_k)$. In this section we derive a recursive expression for $p(x_k|r^{k-1}, u^{k-1})$.

First, we apply the total probability law:

$$\begin{aligned} p(x_k|r^{k-1}, u^{k-1}) & \\ &= \sum_{x_{k-1}} p(x_k|x_{k-1}, r^{k-1}, u^{k-1}) p(x_{k-1}|r^{k-1}, u^{k-1}) \\ &= \sum_{x_{k-1}} p(x_k|x_{k-1}) p(x_{k-1}|r^{k-1}, u^{k-1}). \end{aligned}$$

Now we express $p(x_{k-1}|r^{k-1}, u^{k-1})$ as a function of $p(x_{k-1}|r^{k-2}, u^{k-2})$. By Bayes rule, and noting that $(r^{k-1}, u^{k-1}) = (r^{k-2}, u^{k-2}, u_{k-1}, r_{k-1})$, it results:

$$\begin{aligned} p(x_{k-1}|r^{k-1}, u^{k-1}) & \\ &= \frac{p(x_{k-1}, r^{k-1}, u^{k-1})}{p(r^{k-1}, u^{k-1})} \\ &= \frac{p(x_{k-1}, u_{k-1}, r_{k-1}, r^{k-2}, u^{k-2})}{p(r^{k-1}, u^{k-1})} \\ &= \frac{p(r_{k-1}|x_{k-1}, u_{k-1}, r^{k-2}, u^{k-2}) p(x_{k-1}, u_{k-1}, r^{k-2}, u^{k-2})}{p(r^{k-1}, u^{k-1})} \\ &= \frac{p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}, u_{k-1}, r^{k-2}, u^{k-2})}{p(r^{k-1}, u^{k-1})}, \end{aligned} \tag{4.8}$$

where the last equality holds because the present observation depends only on the previous state and control.

Now we express the denominator by using again the law of total probability:

$$\begin{aligned} p(r^{k-1}, u^{k-1}) & \\ &= \sum_{x_{k-1}} p(x_{k-1}, r^{k-1}, u^{k-1}) \\ &= \sum_{x_{k-1}} p(r_{k-1}|x_{k-1}, u_{k-1}, r^{k-2}, u^{k-2}) p(x_{k-1}, u_{k-1}, r^{k-2}, u^{k-2}) \end{aligned}$$

But

$$\begin{aligned} p(x_{k-1}, u_{k-1}, r^{k-2}, u^{k-2}) & \\ &= p(x_{k-1}|u_{k-1}, r^{k-2}, u^{k-2}) p(u_{k-1}, r^{k-2}, u^{k-2}) \\ &= p(x_{k-1}|r^{k-2}, u^{k-2}) p(u_{k-1}, r^{k-2}, u^{k-2}), \end{aligned}$$

since x_{k-1} does not depend on u_{k-1}. So we can rewrite equation (4.8) as:

$$p(x_{k-1}|r^{k-1}, u^{k-1}) = \frac{p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}|r^{k-2}, u^{k-2})}{\sum_{x_{k-1}} p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}|r^{k-2}, u^{k-2})}.$$

As a result, we modeled the time/state dependence of the information state by the following equations:

$$p(x_k|r^{k-1}, u^{k-1}) = \sum_{x_{k-1}} p(x_k|x_{k-1}) p(x_{k-1}|r^{k-1}, u^{k-1}) \tag{4.9}$$

$$p(x_{k-1}|r^{k-1}, u^{k-1}) = \frac{p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}|r^{k-2}, u^{k-2})}{\sum_{x_{k-1}} p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}|r^{k-2}, u^{k-2})}, \tag{4.10}$$

with the initial condition/notation

$$p(x_0|r^{-1}, u^{-1}) = p(x_0). \tag{4.11}$$

The denominator in equation (4.10) is just a normalizing factor to guarantee that $p(x_{k-1}|r^{k-1}, u^{k-1})$ is a probability mass function. If this factor is omitted, then we can combine equations (4.9), (4.10) and obtain

$$p(x_k|r^{k-1}, u^{k-1}) = \sum_{x_{k-1}} p(x_k|x_{k-1}) p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}|r^{k-2}, u^{k-2}).$$

In words, this means that:

- Up to time $k-1$, we know the initial condition $p(x_0)$ and

$$(u_0, \ldots, u_{k-2}, r_0, \ldots, r_{k-2}) \to p(x_{k-1}|u_0, \ldots, u_{k-2}, r_0, \ldots, r_{k-2})$$

- Given the extra-knowledge u_{k-1}, r_{k-1}, we can update:

$$\begin{aligned} & p(x_k|r_0, \ldots, r_{k-1}, u_0, \ldots, u_{k-1}) = \\ &= \sum_{x_{k-1}} p(x_k|x_{k-1}) p(r_{k-1}|x_{k-1}, u_{k-1}) p(x_{k-1}|r_0, \ldots, r_{k-2}, u_0, \ldots, u_{k-2}) \end{aligned}$$

- summarizing,

$$p(x_k|r^{k-1}, u^{k-1}) = F[r_{k-1}, u_{k-1}, p(x_{k-1}|r^{k-2}, u^{k-2})]$$

that is, we have proven that π_k is an information state (π_k is a function of r^{k-1}, u^{k-1} and it can be computed if $\pi_{k-1}, r_{k-1}, u_{k-1}$ are obtained).

Now, define the $N \times N$ diagonal matrix

$$D(r_k, u_k) = \begin{pmatrix} p(r_k|x_k = 1, u_k) & 0 & \dots & 0 \\ \dots & & & \\ 0 & & \dots & 0 \quad p(r_k|x_k = N, u_k) \end{pmatrix}$$

Remember that the row vector of the state variable is the information state:

$$\pi_k(r^{k-1}, u^{k-1}) = Pr(x_k|r^{k-1}, u^{k-1})$$

Then equation (4.12) may be written in compact form

$$\pi_k(r^{k-1}, u^{k-1}) = \frac{\pi_{k-1}(r^{k-2}, u^{k-2})D(r_{k-1}, u_{k-1})P}{\mathrm{C}}, \tag{4.12}$$

where C is a normalization constant. The transition matrix P is known and constant, for any k:

$$P = [p(x_k = i|x_{k-1} = j)]_{ji}.$$

If the constant C is neglected then the system is linear in the state variable π_k. To conclude, we see that we did obtain an information state that can be computed by a linear recursion.

Chapter 5

Weak Convergence of Feedback Control for Multiple Access Queues

5.1 Overview

5.1.1 Problem Formulation

In Chapter 4, we introduced a model for a multiple access system, we described its dynamics, and we derived an optimal control policy for the shared access in this system. Now, once we have that optimal control algorithm, each source gets to operate the queue based on its local controller, thus resulting in a "decoupling" of the problem, as illustrated in Figure 5.1. Since all the nodes execute exactly the same control algorithm, the distribution of π is the same for all nodes. But other than through this statistical constraint, all decisions are taken locally by each node, based on private data that is not available to any other node, and therefore completely independent.

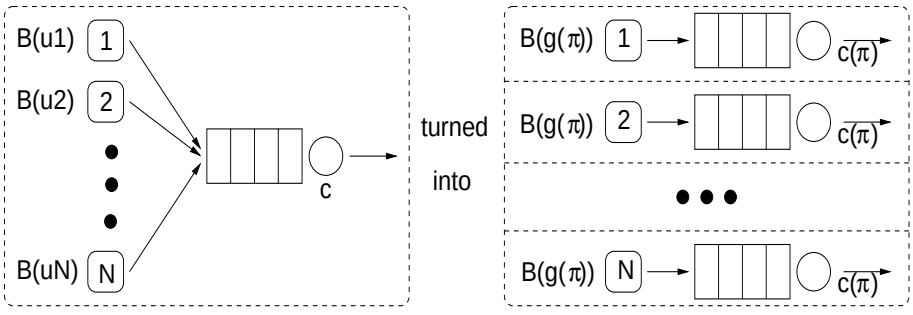


Figure 5.1: The original problem is broken into N independent identical subproblems.

Perhaps the first question that comes to mind once we formulate the picture shown in Figure 5.1 is about ergodic properties of the resulting controlled queues. Specifically, we will be interested in two quantities:

- Average throughput:

$$J(g) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathtt{Pr}(1|x_k, g(\pi_k)) \stackrel{?}{=} \int_{\{x,\pi\}} \mathtt{Pr}(1|x, g(\pi))\ \mathrm{d}\nu(x,\pi).$$

- Average loss rate:

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathtt{Pr}(-1|x_k, g(\pi_k)) \stackrel{?}{=} \int_{\{x,\pi\}} \mathtt{Pr}(-1|x, g(\pi))\ \mathrm{d}\nu(x,\pi).$$

Therefore we see that, in both cases, the questions of interest are formulated in terms of a suitable invariant measure. Since we have assumed the underlying finite state Markov chain to be irreducible and aperiodic, this chain does admit a stationary distribution. Therefore, a sufficient condition for the existence of the sought measure ν is the weak convergence of the sequence of information states π_k to some limit distribution over the simplex Π of probability distributions on N points. And to start developing some intuition on what to expect in terms of the sought convergence result, it is quite instructive to look at typical trajectories of the information state, as shown in Figure 5.2. For a system with three sources, π is a point in the 2D simplex as shown in this picture. And after letting the system run for some time, we find that there are regions of space visited fairly often (bottom right), regions visited less often (bottom left), and regions never visited (top right). Yet each point on this simplex determines a choice of an injection rate, and therefore the frequency with which each point is visited is clearly a fundamental performance analysis tool.

In view of the previous results, a seemingly feasible approach to establishing the sought convergence for our system would have been considering the control action $u \in \mathcal{U}$ to play the role of a channel input in the setup of [35], [27], [65], while the observations $r \in \mathcal{O}$ could have played the role of a channel output (thus making the control u and the observation r the available partial observations). However, this approach does not yield the sought result: in our system, the control u is a function of the information state, i.e., it depends on the state of the system, whereas in those previous works, inputs are independent of the state of the system. So it is this dependence due to feedback-control that makes the main difference between our setup and previous work.

5.1.2 Weak Convergence of the Information State: Steps of the Proof

The proof of weak convergence [16], [17] of π involves five steps:

1. First, we show that the sequence of information states π_k has the Markov property itself. This is a Markov chain taking values in an uncountable space (the simplex Π).

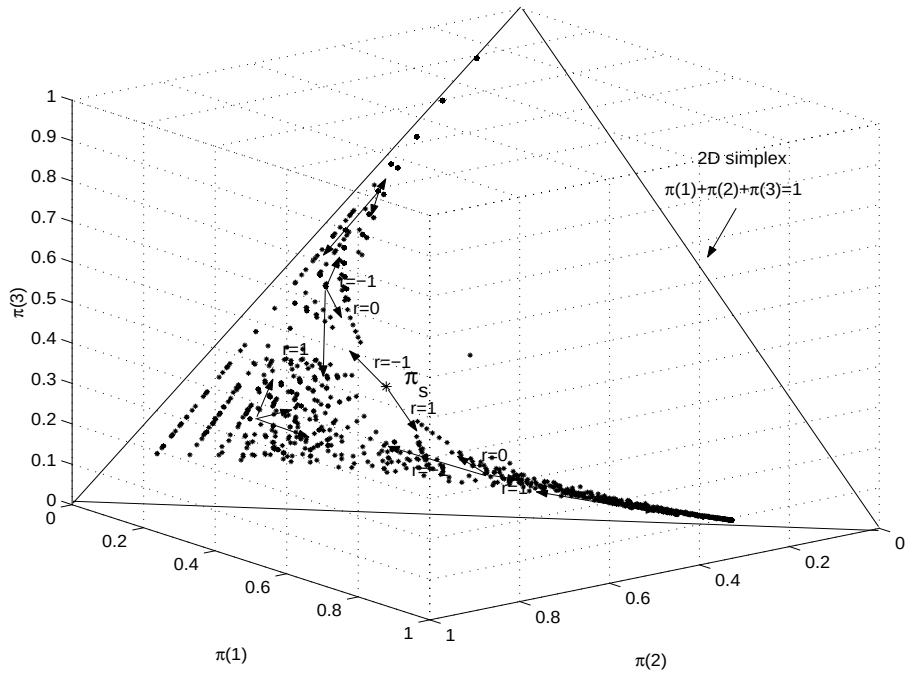


Figure 5.2: The evolution in time of the information state.

2. Then we discretize the simplex Π. We show that for all "small enough" discretizations, there is at least one observation taking π_k out of any cell with positive probability. With this we make sure that there are no absorbing cells, namely we prove that once the chain hits that cell it does not get stuck there forever.

3. Then we show that the stationary distribution π^s of the underlying (finite state) Markov chain is a point reachable from anywhere in the simplex. With this we make sure that there is at least one cell which can be reached from any initial point in Π, and hence that the set of recurrent cells is not empty.

4. Consider next any "small enough" discretization of the space, and define a new process whose values are the cells of this discretization, based on whether π_k hits a particular cell. Then, this new process is (finite state) Markov, and positive recurrent on a non-empty subset of the cells, and therefore it admits an invariant measure itself.

5. Finally we construct a measure as the limit of the "simple" measures from step 4 (as we let the size of the discretization vanish), and we show that this limit is invariant over Π. This requires some further steps, largely based on the elegant framework of [49]:

   5.1. We show that the limit exists and is well defined (it is independent of the

particular sequence of discretizations considered).

5.2. We construct a simple φ-irreducibility measure on Π, and from there we conclude the existence of a *unique* maximal ψ-irreducibility measure.

5.3. We construct a family of accessible atoms in Π, and show that π_k is positive recurrent. From this and from 5.2, using a theorem from [49], we conclude that there exists a unique invariant measure on Π.

5.4. We show that the limit measure of 5.1 is indeed invariant, and therefore conclude that it must be the unique measure of 5.3.

Although steps 2-4 can be dealt with using classical finite state Markov chain theory, steps 1 and 5 cannot: this is because π_k is a Markov chain defined on an (uncountable) metric space, and therefore to analyze its properties we need to resort to a more general theory of Markov processes. Meyn and Tweedie provide an excellent coverage of the problem of Markov chains on general spaces [49], which we found to be an invaluable tool in our work.

We continue now with the formal proofs for the steps of the proof above.

5.2 Weak Convergence—Steps 1-4

5.2.1 Step 1: π is Markov

Although involving a chain defined over a metric space, this proof is elementary, since all we need to invoke is the standard definition of the Markov property and the total probability law:

$$\begin{aligned} p(\pi_{k+1}|\pi_k,\ldots,\pi_0) &\overset{(a)}{=} \sum_{r_k} p(\pi_{k+1}|\pi_k,\ldots,\pi_0,r_k)p(r_k|\pi_k,\ldots,\pi_0) \\ &\overset{(b)}{=} \sum_{r_k} p(\pi_{k+1}|\pi_k,r_k,u_k) \\ &\quad \sum_{x_k} p(r_k|x_k,\pi_k,\ldots,\pi_0)p(x_k|\pi_k,\ldots,\pi_0) \\ &\overset{(c)}{=} \sum_{r_k} p(\pi_{k+1}|\pi_k,g(\pi_k),r_k)\sum_{x_k} p(r_k|x_k,g(\pi_k))\pi_k(x_k) \\ &= p(\pi_{k+1}|\pi_k), \end{aligned}$$

where (a) results from the total probability law, (b) is because when conditioning on π_k, we can add in the conditioning u_k, since $u_k = g(\pi_k)$, and the total probability law, and (c) because conditioned on anything else, r_k depends only on x_k, u_k, and π_k contains all information about x_k given the past. So we see that when conditioning on the past values, π_{k+1} depends only on π_k, and hence π is Markov.

An interesting observation to make here—which gives some insight into structural properties of our model that will allow us to prove the sought weak convergence result—is that the intensity of the arrivals process is a *memoryless* function of π. Although we have not attempted to prove this, it seems at least intuitively clear to us that if instead of the optimal controller we used a suboptimal one (typically based on

the formation of an estimate of the current state), then the optimal decision would not be a memoryless function of the state estimate, but would actually require past state estimates as well.

5.2.2 Step 2: Non-Absorbing Small Discretization Cells

The next step is to show that there is a constant $C > 0$ such that, for any information state $\pi \in \Pi$, there exists an observation $r \in \mathcal{O}$ for which the distance between π_k and the next-step information state π_{k+1} corresponding to r is larger than C. This allows us to *quantize* the simplex Π and make sure that, provided the size of a quantization cell is small enough, at least one observation will take the current information state to a different cell.

Lemma 5.1 *There exists a constant C such that for any $\pi \in \Pi$, there is an observation r for which $||\pi - F[\pi, g(\pi), r]|| \geq \epsilon$, for all $0 < \epsilon \leq C$, and for any norm $||\cdot||$.*

Proof: This basically means that for any state there is at least one observation that moves the chain at a finite non-zero distance away from that given state. We prove this by contradiction: we show that if all jumps are infinitesimally small, then the only information state that can satisfy this condition is the stationary distribution π^s of the original chain. But for this particular information state, any observation different from $r = 0$ does allow jumps of finite size away from π^s.

Suppose that for any $C > 0$, there exists a point $\pi \in \Pi$ such that for any observation $r \in \mathcal{O}$, $||\pi - F[\pi, g(\pi), r]|| < C$. Denote by Q_C the set of points π verifying the above condition for a given C. Note that if $C_1 > C_2$, then $Q_{C_2} \subseteq Q_{C_1}$. Denote by Q_0 the intersection of all Q_C sets. Then the supposition that we want to contradict is equivalent to $Q_0 \neq \emptyset$.

Consider now any $\pi \in Q_0$. Then for any C arbitrarily close to 0, and for any observation r, $||\pi - F[\pi, g(\pi), r]|| < C$ (all jumps are arbitrarily close to zero).

In what follows $k_{(\cdot)}$ are normalizing constants. If $r = 0$, it results π is arbitrarily close to πP. This means π is arbitrarily close to π^s (the stationary distribution of P). Also, for $r = -1$, or $r = 1$, consider the respective $D(g(\pi), r)$ diagonal matrices: it results π is arbitrarily close to $1/k_{\pi,r}\pi D(g(\pi), r)P$. But π is arbitrarily close to π^s as well, so it results π^s arbitrarily close to $1/k_{\pi^s,r}\pi^s D(g(\pi^s), r)P$. In the limit, $\pi^s = 1/k_{\pi^s,r}\pi^s D(g(\pi^s), r)P$. But this cannot be true because D is not the identity matrix. Actually, D is a diagonal matrix with increasing or decreasing diagonal elements $(d_1, \ldots, d_N)$, for $r = 1$ respectively $r = -1$. If, e.g., $r = 1$, then $\pi^s = 1/k_{\pi^s,1}\pi^s D(g(\pi^s), 1)P$. This would mean that there exists $\pi_1 = 1/k_{\pi^s,1}\pi^s D(g(\pi^s), 1)$ with $\pi^s = \pi_1 P$. We know that $\pi^s = \pi^s P$, so if the chain admits only one stationary distribution, it results $\pi_1 = \pi^s$. However this is not possible, since $\pi^s D(g(\pi^s), 1)$ moves towards $(1, 0, \ldots, 0)$ the mass function of the new probability vector away from π^s. □

5.2.3 Step 3: π^s is Reachable from Anywhere

Lemma 5.2 *For any $\pi \in \Pi$, there is a non-zero probability that in the limit the chain gets arbitrarily close to the state π^s, when starting in state π.*

Proof: We illustrate in Figure 5.3 the intuition on which we base our proof. The proof relies on the observation that finite length sequences of $r = 0$ observations move the state arbitrarily closer to π^s. If the observation at time k is $r_k = 0$, then the matrix $D(u_k, r_k)$ becomes diagonal with elements $d_{ii} = 1 - u_k$, so it equals the identity matrix multiplied with a constant; then the recursion for the information state, whenever the source decides not to transmit, can be expressed as $\pi_{k+1} = \pi_k P$. This vector equation has as solution the stationary distribution π^s. It follows that for any $\pi \in \Pi$ as initial state of the chain, there is a path by which the chain reaches in the limit the stationary distribution state π^s, via for example a sequence of successive $r_k = 0$ observations. But any arbitrary length finite sequence of $r_k = 0$ observations may happen with non-zero probability, so for any $\epsilon^s > 0$, there is a finite time K with $r_k = 0, k \leq K$ in which the chain can reach with non-zero probability a state π^{ϵ^s} such that $\|\pi^{\epsilon^s} - \pi^s\| \leq \epsilon^s$. □

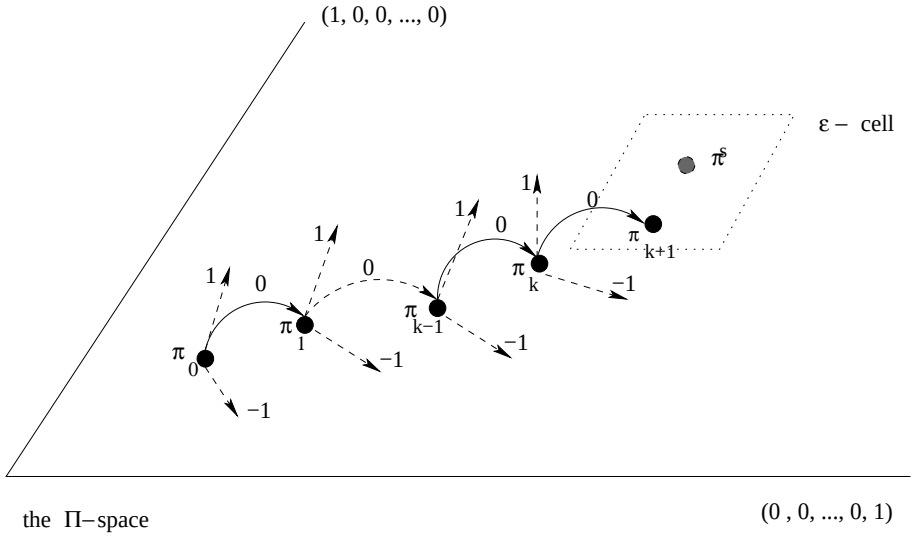


Figure 5.3: A sequence of $r = 0$ observations leads the chain arbitrarily close to π^s.

5.2.4 Step 4: Positive Recurrent Discretization on a Non-Empty Subset

We consider now quantizations of the Markov chain formed by the sequence of information states, with quantization cells of size $\epsilon \leq C$. If the cell size is small enough, then from Lemma 5.1 it follows that for any π inside a discretization cell, there is at

least one observation happening with non-zero probability for which the chain jumps outside the cell. This ensures that there is no state of the chain in which the system stays forever, so the recurrent irreducible subset of discretized cells has more than one element. With this procedure we define a family of quantizations of Π, with members of the family of the form $q^\epsilon = \{q_1^\epsilon, \ldots, q_{N_\epsilon}^\epsilon\}$, where q_i^ϵ are the N_ϵ compact sets contained in q^ϵ and $\bigcup_i q_i^\epsilon = \Pi, \bigcap_i q_i^\epsilon = \emptyset$. For simplicity we will denote $q^\epsilon(\pi)$ the cell to which the instantaneous information state π belongs. We note that

$$\begin{aligned} p(q_{k+1}^\epsilon | q_k^\epsilon, q_{k-1}^\epsilon, \ldots) &= \int_{\pi_{k+1} \in q_{k+1}^\epsilon, \pi_k \in q_k^\epsilon, \pi_{k-1} \in q_{k-1}^\epsilon, \ldots} \times \\ &\quad \times p(\pi_{k+1} | \pi_k, \pi_{k-1}, \ldots) d\pi_{k+1} d\pi_k d\pi_{k-1} \ldots \\ &= \int_{\pi_{k+1} \in q_{k+1}^\epsilon, \pi_k \in q_k^\epsilon} p(\pi_{k+1} | \pi_k) d\pi_{k+1} d\pi_k \\ &= p(q_{k+1}^\epsilon | q_k^\epsilon) \end{aligned}$$

since the process π_k is Markov. The measure with respect to which we are integrating is Lebegue measure over the Π space: we just count how often the continuous chain falls in a given cell. Thus the process $q^\epsilon(\pi_k)$ forms a *finite-state* chain, also having the Markov property (inherited from the continuous chain).

Lemma 5.3 *For any $\epsilon \leq C$, there is a subset $\mathcal{P}^\epsilon \subseteq \Pi$, which contains the stationary distribution π^s of the x_k original chain, and on which the discretized chain $q^\epsilon(\pi_k)$ is positive recurrent.*

Proof: We show in Figure 5.4 a typical behavior of the chain, that shows the existence of a recursive subset $\mathcal{P}_\epsilon \subseteq \Pi$. We base our proof on the fact that π^s is recurrent so its properties will be induced on a recurrent closure of the discretized version of the simplex Π. As we showed in Lemma 5.2, the information state π^s can be reached in the limit with non zero probability from any π_0 initial state of the Markov chain. For any initial π_0 there is a sequence $\pi_0, \ldots, \pi_k, \ldots$ such that $\pi_k \to \pi^s$ when $k \to \infty$, and the size of the quantization cell is strictly positive. Then the time in which the discretized chain $q^\epsilon(\pi_k)$ reaches the cell containing state π^s is finite, so without loss of generality we may consider our limit results with π^s as initial value for the information state.

Denote by $\mathcal{P}_\epsilon$ the set of reachable quantization cells $q^\epsilon(\pi)$ if the chain starts in π^s. We already proved that the cell containing π^s is accessible in a finite number of steps from any other information state π, so implicitly from any cell $q_i^\epsilon \in \mathcal{P}_\epsilon$ as well. Moreover, by construction, any cell $q_i^\epsilon \in \mathcal{P}_\epsilon$ is reachable from $q^\epsilon(\pi^s)$. Since $\epsilon \leq C$, then for any $q^\epsilon(\pi_k)$ there is at least one observation r for which the transition from π_k to π_{k+1} leads to $q^\epsilon(\pi_{k+1}) \neq q^\epsilon(\pi_k)$. It follows that the chain with states in $\mathcal{P}_\epsilon$ is irreducible (and aperiodic as well, since the cell containing π^s is one-step reachable from itself, via a $r = 0$ observation). The state space is finite, so the chain is positive recurrent, and thus it has a stationary distribution. □

Denote by p_ϵ this limit distribution over the $\mathcal{P}_\epsilon$ state space. If $q_i^\epsilon \notin \mathcal{P}_\epsilon$, then $p_\epsilon(q_i^\epsilon) = 0$. We will prove now that there is a limit probability measure on Π to which p_ϵ converges in the limit $\epsilon \to 0$, and study the properties of that measure.

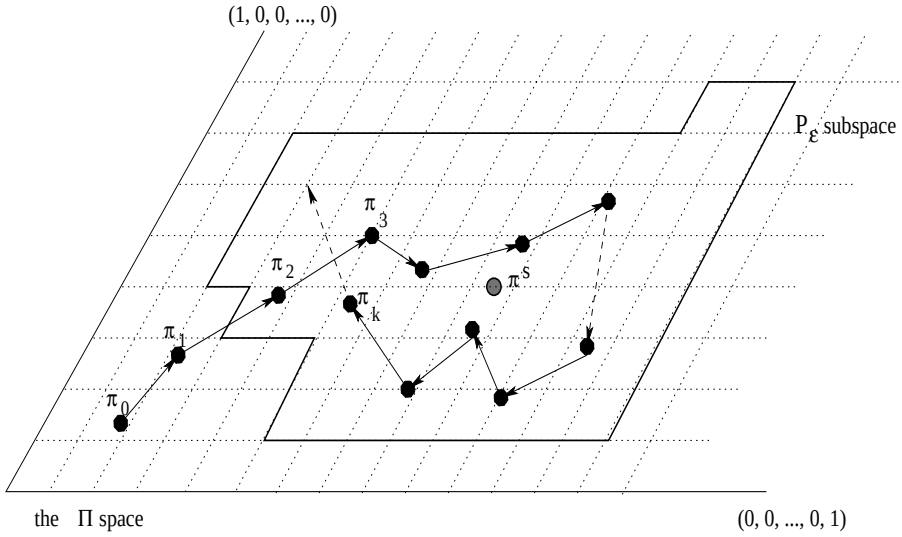


Figure 5.4: After passing through a sequence of transient states, the chain reaches a recursive subset of the discretized simplex Π.

5.3 Weak Convergence—Step 5

There exists a unique limit invariant measure over Π, $\nu_\epsilon \to \nu$, when $\epsilon \to 0$.

5.3.1 Step 5.1: Existence of the limit measure

We will show that the limit measure exists by considering, for any subset A of Π, sequences of measures on subsets of the discretized simplex that cover, and respectively intersect A. We show that they converge to the same limit.

Definition 5.1 *Define the* inner *and* outer *sequences of measures over the simplex* Π*, corresponding to the set of* ϵ*-discretizations:*

$$\nu_\epsilon^I(A) = \sum_{S \in \mathcal{P}_\epsilon : S \subseteq A} p_\epsilon(S) \tag{5.1}$$

$$\nu_\epsilon^O(A) = \sum_{S \in \mathcal{P}_\epsilon : S \cap A \neq \emptyset} p_\epsilon(S), \tag{5.2}$$

where A *is any subset in the* σ*-algebra of* Π.

We want to prove that, for any given A, both $\nu_\epsilon^I(A)$ and $\nu_\epsilon^I(A)$ converge to the same limit, as $\epsilon \to 0$. That limit will be our limit invariant measure $\nu(A)$.

We will prove first convergence of each of the limits. Consider ν_ϵ^I; we will prove that the sequence is Cauchy for any set A, and it trivially has a convergent subsequence, which will mean the whole sequence is convergent.

For a given set A, denote $A_n = \{\cup S \in \mathcal{P}_{\epsilon_n} : S \subseteq A\}$ the inner cover of the set A corresponding to discretization step ϵ_n. We will prove first that the normalized

volume of the difference set between two inner covers of the set A tends to the empty set, and consequently the probability measure over that difference set tends to zero. Define the metric $d(X,Y) = \mu^{\text{Leb}}((X-Y)\cup(Y-X))/\mu^{\text{Leb}}(\Pi)$, on the σ-algebra $\mathcal{B}$ of Π, $X, Y \in \mathcal{B}$ (this represents the normalized volume of the set where the two subsets X, Y differ from each other—μ^{Leb} is Lebesgue measure). It is easy to verify that $d(\cdot,\cdot)$ is indeed a valid metric.

Let ϵ_n be a decreasing sequence of discretization steps, with $\lim_{n\to\infty}\epsilon_n = 0$. Then due to the fact that $\mathcal{P}_{\epsilon_n}$ is a sequence of subsequent discretizations of the space Π when $n \to \infty$, it follows $\lim_{n\to\infty} A_n = A$. Since A_n is convergent, it is also Cauchy in the metric space $(\mathcal{B}, d)$. This means that for any $\delta > 0$, there exists n_δ such that $d(A_n, A_m) < \delta$, for any $m > n \geq n_\delta$. So the normalized volume of the set difference between two set elements of the sequence becomes arbitrarily small. That also means that if $\epsilon_n, \epsilon_m \to 0$, then $\nu^I_{\epsilon_n\epsilon_m} d(A_n, A_m)) \to 0$, as $\nu_{\epsilon_n\epsilon_m}$ is a stationary distribution over finite spaces with decreasing cell size. Then for any $\delta_\nu > 0$, there is n_{δ_ν} such that $\nu^I_{\epsilon_n\epsilon_m}((A_n - A_m)\cup(A_m - A_n)) < \delta_\nu$, for any $m > n \geq n_{\delta_\nu}$. Note that $\nu^I_{\epsilon_n\epsilon_m}((A_n - A_m)\cup(A_m - A_n)) \geq |\nu^I_{\epsilon_n\epsilon_m}(A_n) - \nu^I_{\epsilon_n\epsilon_m}(A_m)|$.

Finally, we note that $|\nu^I_{\epsilon_n\epsilon_m}(A_n) - \nu^I_{\epsilon_n\epsilon_m}(A_m)| = |\nu^I_{\epsilon_n}(A_n) - \nu^I_{\epsilon_m}(A_m)|$, due to the property of inclusion for the sequence of measures (sum of probabilities of ϵ'-discretization cells that cover exactly a ϵ-discretization cell is equal to the probability of that ϵ-discretization cell), and the way A_n, A_m are constructed (cells corresponding to multiples of ϵ are all included in the cells corresponding to ϵ).

We conclude that for any $\delta_\nu > 0$, there is n_{δ_ν} such that $|\nu^I_{\epsilon_n}(A_n) - \nu^I_{\epsilon_m}(A_m)| < \delta_\nu$, for any $m > n \geq n_{\delta_\nu}$. This means that the sequence $\nu^I_{\epsilon_n}(A_n)$ is Cauchy as well. It is trivial to show that there is a convergent subsequence of $\nu^I_{\epsilon_n}(A_n)$: pick, e.g., $\epsilon_n = \epsilon_0/2^n$, then the corresponding subsequence is bounded from above by 1, and monotonically increasing; it follows the subsequence is convergent. But a Cauchy sequence with a convergent subsequence is convergent, which proves that $\nu^I_\epsilon(A)$ is convergent for any set A.

The proof for convergence of ν^O_ϵ is similar and we will omit it. Both limits exist, and it is obvious that they fulfill the inequality:

$$\lim_{\epsilon\to 0}\nu^I_\epsilon(A) \leq \lim_{\epsilon\to 0}\nu^O_\epsilon(A), \text{ for any } A \subset \Pi.$$

We want to prove that the inequation above holds in fact with equality. Assume that the inequality is strict; then let $\delta = \nu^O_0(A) - \nu^I_0(A) > 0$. But this would mean that there exists at least a cell in any partition $\mathcal{P}_\epsilon$ of size $\delta > 0$, for all $\epsilon_n \to 0$. However, in the limit the two sets of sumation become equal (with union A), so a contradiction results.

Definition 5.2 *Define the measure ν over the simplex Π as the common limit of the two sequences of measures:*

$$\nu(A) = \lim_{\epsilon\to 0}\nu^I_\epsilon(A) = \lim_{\epsilon\to 0}\nu^O_\epsilon(A) \tag{5.3}$$

for a given $A \subseteq \Pi$.

For the proofs in the next two sections we will use definitions and notations also found in [49].

5.3.2 Step 5.2: Existence of a unique maximal ψ-irreducibility measure

Definition 5.3 *Denote by* $B(\pi_0, \delta) = \{\pi \in \Pi : \|\pi - \pi_0\| < \delta\}$ *the open ball, with* $\delta > 0$.

Definition 5.4 *Denote by* $\mathcal{B}(\Pi)$ *the* σ*-field generated by the open balls in* Π.

Definition 5.5 *Denote, for any state* $\pi \in \Pi$ *and subset* $A \in \mathcal{B}(\Pi)$*, the probability that, when starting in state* π*, the chain ever reaches subset* A*:*

$$L(\pi, A) = P_\pi(\tau_A < \infty)$$

Lemma 5.4 *Let* $\pi_n \in \Pi, n = 0, 1, \ldots$ *be a sequence of information states. Then* π_n *is* ϕ*-irreducible on* $\mathcal{B}(\Pi)$.

Proof: Let π^s be the stationary distribution of the underlying chain. Define the measure ϕ on $\mathcal{B}(\Pi)$ as:

$$\begin{aligned} \phi(B(\pi^s, \delta)) &= \mu^{\text{Leb}}(B(\pi^s, \delta)) \\ \phi(A) &= 0, \text{otherwise} \end{aligned} \tag{5.4}$$

In the step 3 of the proof, we proved π^s is reachable from anywhere. Hence, for all $\pi \in \Pi$, we have $L(\pi, B(\pi^s, \delta)) > 0$, and ϕ is an irreducibility measure. □

Note: If a ϕ-irreducibility measure exists, then there is a part of the space reachable from anywhere, so one might expect independence of the chain from the initial conditions, by analogy with finite chains.

Proposition 5.1 *If* π_n *is* ϕ*-irreducible, then there exists a unique 'maximal' measure* ψ *on* $\mathcal{B}(\Pi)$ *such that* π_n *is* ψ*-irreducible and* $\phi \leq \psi$*. We denote by* $\mathcal{B}^+(\Pi)$ *the* σ*-algebra of* Π *with sets on which* ψ *is positive.*

Proof: The proof is standard for chains fulfilling the previous conditions, and it can be found in [49]. □

5.3.3 Step 5.3: Uniqueness of the invariant measure on Π

Definition 5.6 *We call* $\alpha \in \mathcal{B}(\Pi)$ *an* atom *for a sequence* π_n *if there exists a measure* μ *on* $\mathcal{B}(\Pi)$*, such that, for any* $\pi \in \alpha$*,* $P(\pi, A) = \mu(A)$ *(for any* $A \in \mathcal{B}(\Pi)$*).*

Definition 5.7 *We call* α *an* accessible atom *for a sequence* π_n*, if* π_n *is* ψ*-irreducible and* $\psi(\alpha) > 0$.

Note: Atoms behave like states in finite chains. From the development in [49], it turns out that the reason why so many results about finite chains carry over to more general settings is precisely the fact that it is always possible to construct atoms.

Proposition 5.2 *All balls $B(\pi^s, \delta)$, with $\delta > 0$, are accessible atoms for any sequence π_n.*

Proof: Let α be a set in $\mathcal{B}(\Pi)$, and let $\pi \in \alpha$. Then, depending on the current observation r, there are three possible transitions from π, via the recursion function $F[\pi, g(\pi), r]$. Then for any $A \in \Pi$ we can consider the measure:

$$\begin{aligned} \mu(A) &= p(r), \text{ if } F[\pi, g(\pi), r] \in A \\ \mu(A) &= 0, \text{ otherwise.} \end{aligned}$$

Then any $\alpha = B(\pi^s, \delta)$ is an accessible atom. □

Definition 5.8 *Denote by $E_\pi[\eta_A]$ the expected number of returns of the chain to subset $A \in \Pi$ when starting in state π.*

Definition 5.9 *A set $A \in \mathcal{B}(\Pi)$ is called* recurrent *if $E_\pi[\eta_A] = \infty$ for all $\pi \in A$ (when starting in A, the expected number of returns to A is infinite).*

Lemma 5.5 *If π_n is ψ-irreducible and admits a recurrent atom α, then every set in $\mathcal{B}^+(\Pi)$ is recurrent.*

Proof: If $A \in \mathcal{B}^+(\Pi)$ then for any π, there exist r, s such that $P^r(\pi, \alpha) > 0$, $P^s(\alpha, A) > 0$ and we can write, by considering the paths of the chain that go from π to A via the atom α:

$$\sum_n P^{r+s+n}(\pi, A) \geq P^r(\pi, \alpha)[\sum_n P^n(\alpha, \alpha)]P^s(\alpha, A) = \infty.$$

since α being an atom implies that $\sum_n P^n(\alpha, \alpha)$ diverges. □

Note: Observe again the analogy between atoms and states of a finite chain.

Definition 5.10 *A sequence π_n is called* recurrent *if and only if it is ψ-irreducible, and $E_\pi[\eta_A] = \infty$ for any $\pi \in \Pi$ and $A \in \mathcal{B}^+(\Pi)$.*

Lemma 5.6 *Any sequence of information states drawn from the Markov chain π_n is recurrent.*

Proof: From Lemma 5.4 and Proposition 5.1 it results that π_n is ψ-irreducible. Furthermore, from step 3 of the proof it results that all the balls $B(\pi^s, \delta)$, with $\delta > 0$, are recurrent atoms. Then every $A \in \mathcal{B}^+(\Pi)$ is recurrent, and from Definition 5.9, it results that $E_\pi[\eta_A] = \infty$ for all $\pi \in A$. We still need to prove that even if $\pi \notin A$, we still have $L(\pi, A) = 1$. By definition, $L(\pi, \bigcup \mathcal{B}^+(\Pi)) = 1$. Suppose the sequence π_n hits at some time a set $B \in \mathcal{B}^+(\Pi)$. If $A = B$, then the sought result follows. Otherwise, we will have $L(y, A) > 0$ for all $y \in B$, because of the ψ-irreducibility over $\mathcal{B}^+$. But $B \in \mathcal{B}^+(\Pi)$ and $E_y[\eta_B] = \infty$, so it results $L(y, A) = 1$. So finally $L(\pi, A) = 1$ and this case is reduced to the previous one (where $\pi \in A$). Hence, π_n is recurrent. □

Definition 5.11 *A sequence π_n is called* positive *if and only if it is ψ-irreducible and admits an invariant measure γ.*

Lemma 5.7 (Kac's theorem) *If a sequence π_n is recurrent and admits an atom $\alpha \in \mathcal{B}^+(\Pi)$, then π_n is positive if and only if $E_\alpha[\tau_\alpha] < \infty$.*

Proof: If $E_\alpha[\tau_\alpha] < \infty$ then obviously $L(\alpha, \alpha) = 1$, so it results π_n is recurrent. It also results from the structure of γ (see [49]) that γ is finite, so positive as well. The converse results from the structure of γ as well. □

Lemma 5.8 *The sequence of information states π_n is positive.*

Proof: From Lemma 5.6 it results that π_n is recurrent. Also, from step 3 of the proof it results that every ball $\alpha = B(\pi^s) \in \mathcal{B}^+(\Pi)$ is an atom, and $E_\alpha(\tau_\alpha) < \infty$. Then it results π_n is positive. □

Theorem 5.1 *There exists an unique invariant* probability *measure of π_n.*

Proof: The proof for this theorem is valid for chains having the properties we have analyzed until now, and can be found in [49]. □

5.3.4 Step 5.4: Invariance of ν

We will state now the main theorem of this chapter.

Theorem 5.2 *The measure ν (as constructed in step 5.1) is the unique invariant probability measure on* Π.

Proof: For invariance of ν, we need to prove that

$$\nu(A) = \int_\Pi \nu(dy) P(y, A). \tag{5.5}$$

From the definition of ν we have that for any $\epsilon > 0$,

$$\nu_\epsilon^I(A) \leq \nu(A) \leq \nu_\epsilon^O. \tag{5.6}$$

If we denote by $P_\epsilon(\cdot,\cdot)$ the transition probability kernel for the ϵ-discretization, then we can rewrite the rightmost term of the inequality (5.6) as:

$$\begin{aligned}\nu_\epsilon^I(A) &= \sum_{S\in\mathcal{P}_\epsilon: S\subseteq A} p_\epsilon(S) \\ &= \sum_{S\in\mathcal{P}_\epsilon: S\subseteq A}\sum_{T\in\mathcal{P}_\epsilon} p_\epsilon(T)P_\epsilon(T,S) \\ &= \sum_{T\in\mathcal{P}_\epsilon}\sum_{S\in P_\epsilon: S\subseteq A} p_\epsilon(T)P_\epsilon(T,S) \\ &= \sum_{T\in\mathcal{P}_\epsilon} p_\epsilon(T)P_\epsilon(T,\cup\{S\in P_\epsilon : S\subseteq A\})\end{aligned}$$

In a similar manner we can rewrite the expression for $\nu_\epsilon^O(A)$:

$$\nu_\epsilon^O(A) = \sum_{T\in\mathcal{P}_\epsilon} p_\epsilon(T)P_\epsilon(T,\cup\{S\in P_\epsilon : S\cap A\neq\emptyset\})$$

By taking now the limit in expression (5.6), we know that both left and right limits exist and are equal, so it results

$$\begin{aligned}\nu(A) &= \lim_{\epsilon\to 0}\sum_{T\in\mathcal{P}_\epsilon} p_\epsilon(T)P_\epsilon(T,\cup\{S\in P_\epsilon : S\subseteq A\}) \\ &= \lim_{\epsilon\to 0}\sum_{T\in\mathcal{P}_\epsilon} p_\epsilon(T)P_\epsilon(T,\cup\{S\in P_\epsilon : S\cap A\neq\emptyset\}) \\ &\overset{(a)}{=} \int_\Pi \nu(dy)P(y,A), \qquad (5.7)\end{aligned}$$

where equality (a) holds because, under some continuity conditions, in the limit $\epsilon\to 0$: the sum becomes integral; the probability limit ν *exists*; the quantization cell $T\in\mathcal{P}_\epsilon$ becomes the infinitesimal integration variable $T\to dy$; the transition probability kernel $P_\epsilon(\cdot,\cdot)\to P(\cdot,\cdot)$; both the reunions of cells included in A and respectively intersecting A cover whole set A.

From (5.7), it results that ν is invariant, and thus it is the unique invariant measure on Π. □

5.4 Numerical Simulations

In this section, we show results of numerically evaluating the integrals above. We simulated a system with $N=2$, $N=4$, and $N=8$ sources, and with different values for the loss threshold $T=0.02$, $T=0.05$, and $T=0.1$. The chain is birth-and-death with probability p. We let the system run for $t=100000$ time steps. We plot the average throughput and loss as a function of the transition probability p. The resulting plots are shown in Figures 5.5, 5.6. We see that the plots do not depend significantly on p: the dependence is essentially on the stationary probability π_s of

the original chain P, which is the same for any symmetric birth-and-death chain. As expected, large values for T imply larger throughput, as the controller is allowed to probe more often the environment; this is on the expense of increased losses.

5.5 Example: Slotted Aloha

There have been important research efforts recently focused on access protocols for sensor networks. Algorithms designed for classical networks do not always work in this context, due to the special properties of the wireless access medium. Randomized access protocols have emerged to be the most appropriate in this setting (e.g. IEEE 802.11), because they cope well with the randomness of the access medium and with the changing of environment conditions.

We focus on slotted Aloha as MAC protocol [8]. In the model we have studied in the previous two chapters, each source has feedback only from the router, about the successful transmission of its own packets. On the contrary, in Aloha, feedback is shared, so all information sources know at each time moment whether the router successfully sent a packet. In this section, our goal is to design an algorithm for slotted Aloha to estimate in the best possible manner the instantaneous number of backlogged packets and the corresponding optimal injection probability.

Consider a number of sources accessing a bottleneck router. The overall process formed by the number of packets that arrives from all the information sources together is modeled with an arbitrary arrival distribution: a new packets arrive at time k with probability $p(a = a_k)$ at the bottleneck. We focus on the *infinite set of nodes* approach [8]. That is, each new packet newly arrived is regarded as a separate node, no matter which source it belongs to. The packet is either transmitted with probability q_k, or backlogged. We further assume that if two or more nodes try to transmit at the same time instant, then a collision occurs, and consequently they are all backlogged. A correct transmission does happen only if *exactly* one of the nodes attempts transmission in a given time slot.

Our goal is to design a policy for controlling the injection attempt probability q_k that the sources need to use at time k, in order to maximize their throughput. We know from the slotted Aloha theory [8] that this happens when $q_k = 1/b_k$, where b_k is the number of backlogged packets. However, the problem is that no source knows the overall number of backlogged packets, so the best it can do is to make an estimation of it. It is crucial to have a good estimation of this number to ensure good performance and stability of the system [8]. We propose an algorithm based on a hidden Markov chain model for this system, and derive a control for the transmission probability based only on feedback information from the router.

5.5.1 System Model and Dynamics

We consider a number of sources accessing a single shared bottleneck router. The actual number of sources is not essential in the 'infinite set of nodes' approach. Only the *overall* number of backlogged packets is important, no matter from which particular source they originated.

The system parameters we consider for our model are:

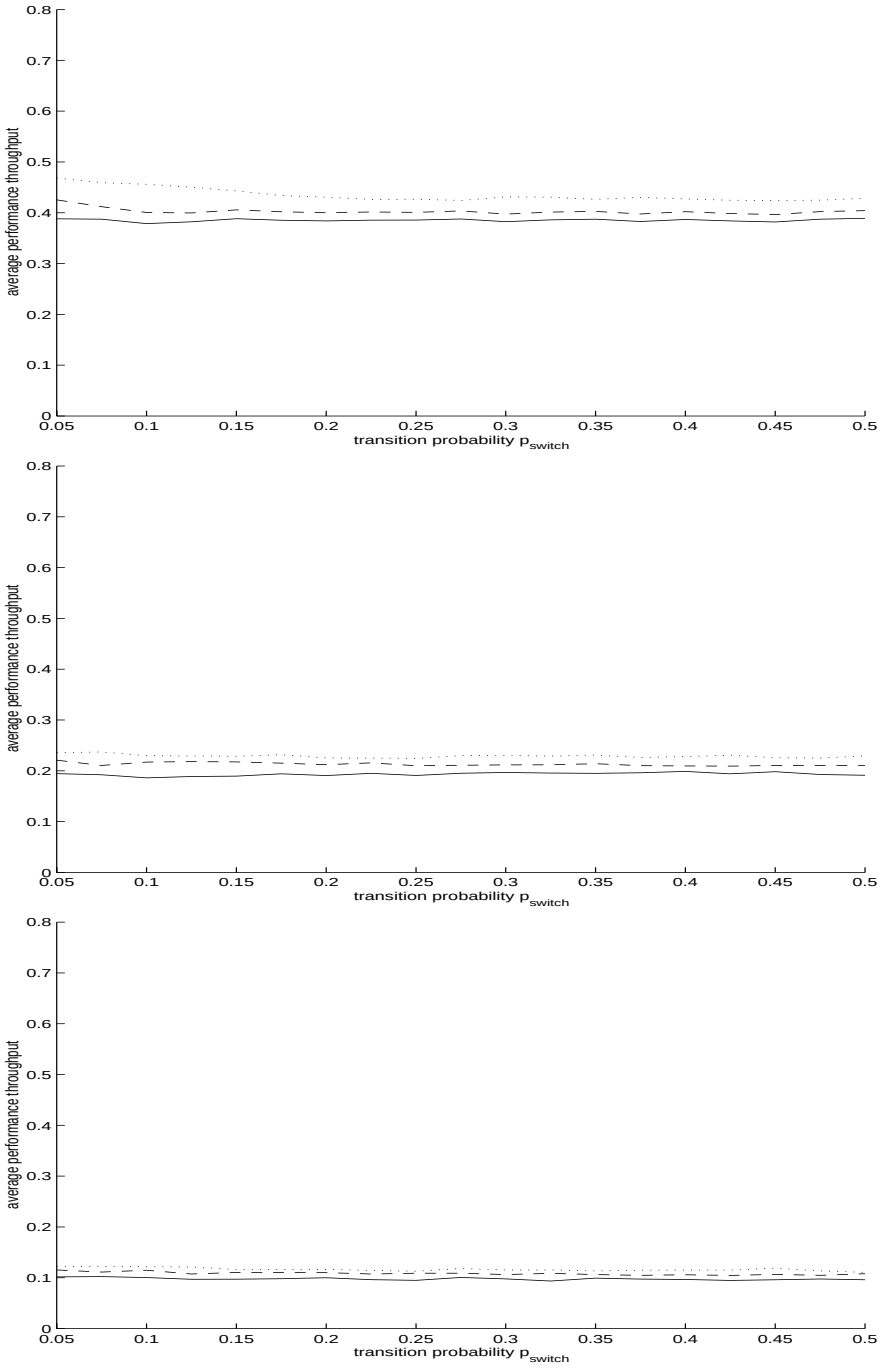


Figure 5.5: The plots of average throughput from top to bottom: $N = 2, 4, 8$ sources. Legend: dotted plot, $T = 0.1$; dashed plot, $T = 0.05$; solid plot, $T = 0.01$.

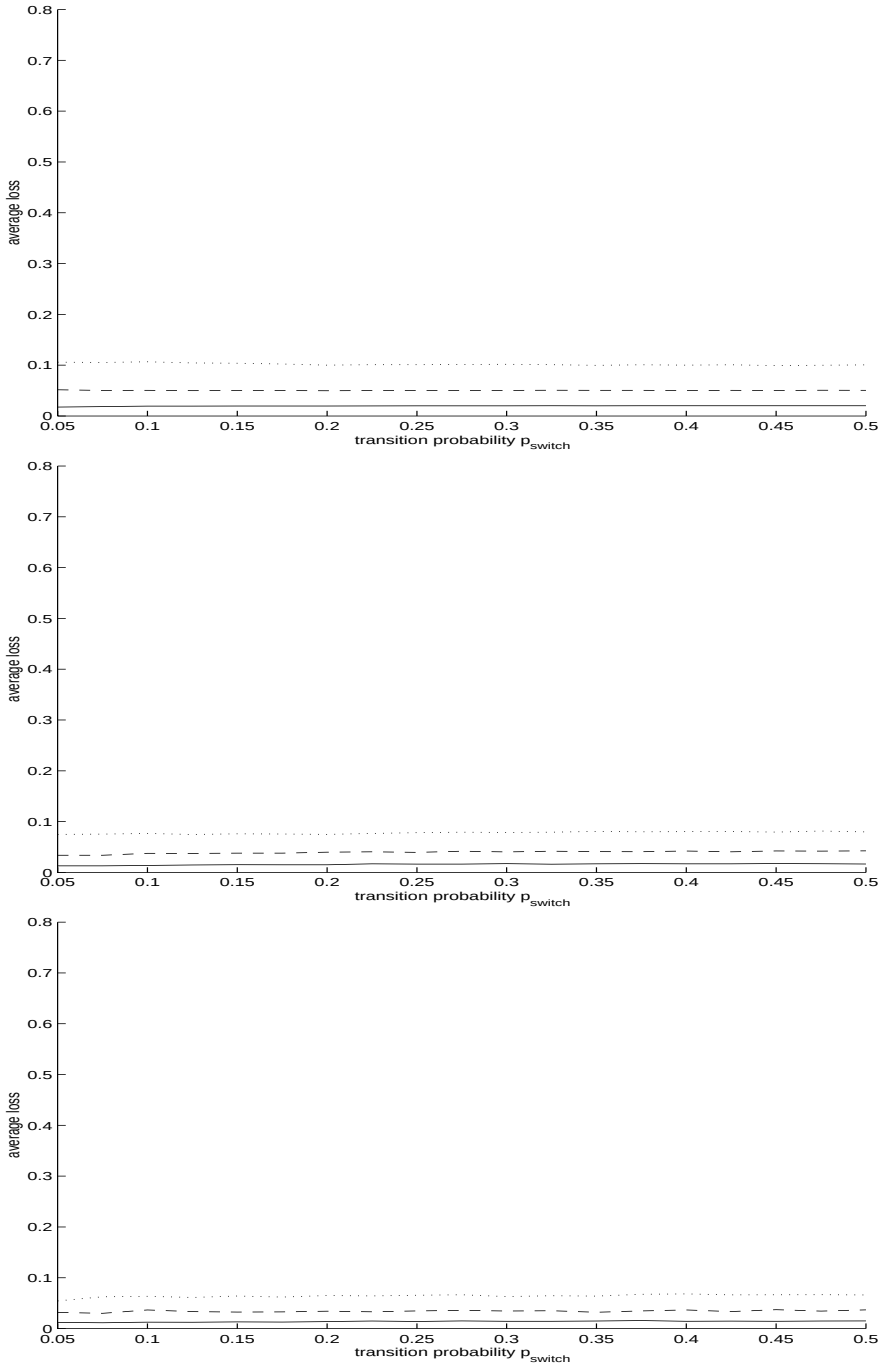


Figure 5.6: The plots of average loss from top to bottom: $N = 2, 4, 8$ sources. Legend: dotted plot, $T = 0.1$; dashed plot, $T = 0.05$; solid plot, $T = 0.01$.

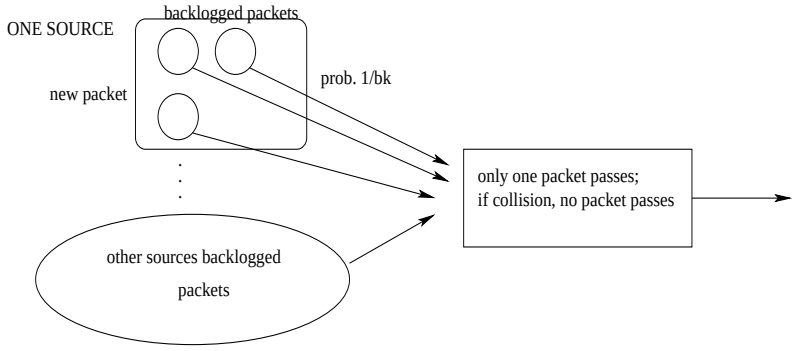


Figure 5.7: A Slotted Aloha access medium.

- The arrival process, which is determined by the probability distribution of the random process a_k of newly arrived packets at time k, $p(a_k)$. This is considered as a 'constant' of the system, and assumed to be known.

- The unknown state of the system, which is given by the total number of backlogged nodes, $b_k \in \{0, \ldots, B_{max}\}$.

- The number of backlogged nodes, which increases by one unit for each new arrival, and decreases by one unit for each successful sending.

- The injection probability q_k (the parameter to be controlled). The optimal value [8] for q_k in order to maximize the throughput is $1/b_k$.

- The three-valued feedback $r_k \in \{-1, 0, 1\}$ for collision/no packet sent/successful transmission, observed at the router is available to all the backlogged nodes.

The Aloha model assumes that unsuccessfully sent packets are again backlogged. Backlogged packets behave like separate nodes. That is, even if a source has more than one packet backlogged, it will try to transmit each of them separately with probability q_k. This also insures some sort of fairness, that is sources that have many packets backlogged will have more chances to transmit one of them than other sources that have few packets backlogged. We will show that the number of backlogged packets b_k forms a partially observed (hidden) controlled Markov process, that is, given all past states and controls, b_{k+1} depends only on b_k and q_k. The partial observation is the feedback r_k resulted by applying the control q_k. We denote the corresponding transition matrix of the controlled chain by P_{q_k} (see below). For a complete theory of partially observed control Markov chains, see [38].

The probability of getting a certain value for the observation r_k depends only on the number of backlogged packets b_k and on the applied control/ injection probability q_k. A packet is accepted if only one of the backlogged nodes decides to transmit. No packet is sent when none of the nodes attempts transmission. This can be expressed

with the following equations:

$$\begin{aligned} p(r_k = 1|q_k, b_k) &= b_k q_k (1-q_k)^{b_k - 1} \\ p(r_k = 0|q_k, b_k) &= (1-q_k)^{b_k} \\ p(r_k = -1|q_k, b_k) &= 1 - (b_k q_k (1-q_k)^{b_k-1} + (1-q_k)^{b_k}) \end{aligned}$$

All the information available about the number of backlogged packets b_k can be gathered the information state. As information state, we will use, again, the conditional probability vector $\pi_{k+1} = [p(b_{k+1} = 0|q^k, r^k), \ldots, p(b_{k+1} = B_{max}|q^k, r^k)]$ where, for a random variable v, we denote $v^k = \{v_0, \ldots, v_k\}$.

This quantity can be computed recursively. We first state the formula for updating the information state when the new information r_k, q_k is available [38]:

$$\pi_{k+1} = \pi_k P_{q_k} D(r_k, q_k)$$

where P_{q_k} is the transition matrix of the controlled Markov chain when the control is q_k, and $D(r_k, q_k)$ is a diagonal matrix having as elements $p(r_k|q_k, b)$, for $b = 0, \ldots, B_{max}$.

If the injection rate is q, then the corresponding elements of the matrix P_q can be computed as:

$$P_q(n, n+i) = \begin{cases} p(a_k = 0) + p(a_k = 1)q, \\ \text{ if } n = 0, i = 0 \\ p(a_k = i)(1 - \binom{1}{i} q(1-q)^{i-1} + \\ + p(a_k = i+1)\binom{1}{i+1} q(1-q)^i), \\ \text{ if } n = 0, i \neq 0 \\ p(a_k = 0)\binom{1}{n} q(1-q)^{n-1}, \\ \text{ if } n > 0, i = -1 \\ p(a_k = i)(1 - \binom{1}{n+i} q(1-q)^{n+i-1}) + \\ + p(a_k = i+1)\binom{1}{n+i+1} q(1-q)^{n+i}, \\ \text{ if } n > 0, i \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

So far, we have described the main ingredients of our system. We proceed now with a description of the dynamics of the system. At each step k, the transmitting sources keep track of the information state π_k and compute π_{k+1} using the new available information, control q_k and observation r_k. Then the optimal injection rate that each backlogged node should employ can be derived using a dynamic programming algorithm [38], to maximize on the long term cost function. A suboptimal yet very simple and intuitive algorithm is to just compute the estimated number of backlogged packets $b^{est}_{k+1} = \texttt{ceil}(E_{\pi_{k+1}} b_{k+1})$, and then use at time $k+1$ an injection probability $q_{k+1} = 1/b^{est}_{k+1}$. Thus, at time $k+1$, with probability $p(a_{k+1})$ a total of a_{k+1} new packets arrive, each backlogged or newly arrived packet attempts to get through with probability q_{k+1}, and an observation r_{k+1} is consequently generated by the router. The process repeats for every time slot k.

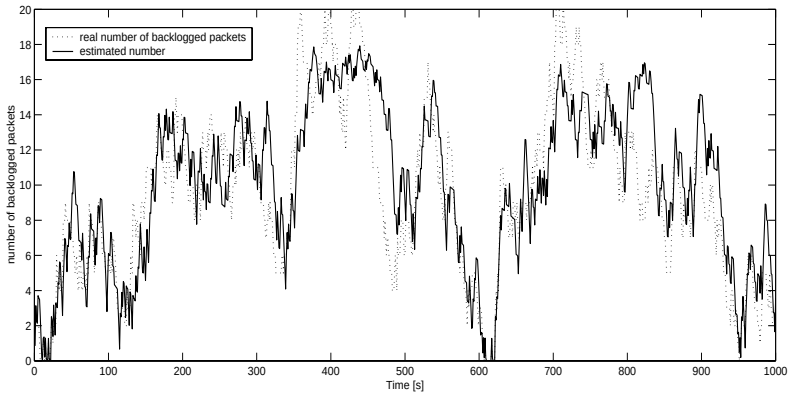


Figure 5.8: An illustration of our algorithm for Aloha access: the estimated number of backlog packets keeps track of the real value.

5.5.2 Numerical Results

We use as arrival process an i.i.d. process with $p(a_k) = (0.76, 0.16, 0.05, 0.02, 0.01)$ where $a_k = 0, 1, \ldots, A_{max}$ is the number of newly arrived packets. As an example[1]: probability of 0 new arrivals is $p(a_k = 0) = 0.76$, probability of 1 new arrival is $p(a_k = 1) = 0.16$ etc. Any arrival process can be considered (though it has to be estimated beforehand and assumed stationary along the experiment), as long as it fulfills the stability condition $E(a_k) < 1/e = 0.3679$ [8], for the case of constant service capacity 1.

Our performance evaluation is to see how close the estimation of the number of backlogged packets is to the real value, and of course the ultimate goal would be to keep it as close to zero as possible. Figure 5.8 shows a $T = 1000$ plot. We see that the estimated value of the number of backlogged packets does indeed keep a reasonably good track of the real value. A complete analysis of this algorithms is a subject of our further research.

5.6 Conclusions

In this chapter we have proved that the information state introduced in Chapter 4 converges in the limit. This shows that our system is provably stable. Since the optimal control is a function of the information state, it follows that various performance metrics of interest, like for example the average throughput and loss, converge weakly too.

Our proof is based on the theory of Markov chains over general spaces. We first

[1]This example is chosen so as to ensure the stability of the queue. We considered this particular distribution because its mean is 0.36, or very close to the 'saturation' threshold. Note that an obvious choice would be a Poisson arrival process, however we argue that our algorithm is valid also for arbitrary arrival distributions.

showed that the limit measure exists, and then we showed a method to construct it by using successive discretizations over the information state spaces. Finally, we applied our results to derive a provably stable estimation of the injection probability in the Aloha protocol.

Chapter 6

Conclusions and Future Work

6.1 Summary

6.1.1 Network Correlated Data Gathering

The first part of this thesis was concerned with the optimization of the total flow cost in networks measuring correlated data. We showed that, in general, the flow optimization problems for networks with correlated data at nodes involve both the task of supply allocation at nodes, and the task of finding good transmission structures. We showed how a separate treatment of these tasks can lead to highly sub-optimal solutions. Thus, an important goal is to solve the joint problem of rate allocation and transmission structure optimization for such networks. Our setting led to original rate allocation problems and original tree building problems, depending on the source coding model.

We considered a family of cost functions that is widely met in various practical scenarios, namely the class of separable metrics $[F(\text{rate})] \times [\text{link weight}]$. We studied the two possible coding paradigms that are met in such problems, namely coding by using explicit communication, and Slepian-Wolf coding. The main analyzed scenario was the data gathering problem, namely the case when there is a single sink to which data from all the other nodes needs to arrive. We also studied generalizations of our results to arbitrary traffic matrix problems.

For the explicit communication approach, we showed that rate allocation is easy to be done, since the compression at a node is directly dependent on the data from nodes that use it as a relay. However, we proved that the rate allocation and the chosen transmission structure are inter-related, so the two tasks cannot be separated. Moreover, we proved that finding an optimal transmission structure is NP-complete even for very simplifying assumptions. Our main result for the coding by explicit communication case states that there is no polynomial time algorithm that solves the network data gathering tree cost problem, unless P=NP. Our proof is a non-trivial reduction from the min-set cover problem, that is a problem known to be NP-complete.

We also proposed distributed approximation algorithms that are expected to perform well for arbitrary settings. Namely, we propose an original tree structure for

the case when single hop opportunistic aggregation coding is performed, namely the SPT/TSP tree (shortest path tree/traveling salesman path), which builds the SPT for nodes in a certain radius away from the sink, and TSPs for the rest of the network. Our algorithm is inspired from insights obtained by using simulated annealing to find structures close to the optimal one, for our computational-hard problem.

The second approach is to allow nodes to use joint coding of correlated data without explicit communication, by using Slepian-Wolf coding. With this approach, finding a good routing structure turns out to be easy, because routing and coding are separated (decoupled); however data coding becomes complex and global network knowledge is needed for an optimal solution. In the second approach, namely coding by explicit communication, nodes can exploit the data correlation only by receiving explicit side information from other nodes (for example, when other nodes use a node as relay, their data is locally available at that relaying node). Thus, the correlation structure is exploited through communication and joint aggregate coding/decoding locally at each node. In this case, data coding is easy and relies only on locally available data as side information. However, optimizing the routing structure becomes complex.

For the Slepian-Wolf approach, we proved that a well-studied transmission structure, namely the shortest path tree, is optimal for any rate allocation. Thus, the tasks of rate allocation and transmission structure optimization separate in this case. We showed how the rate allocation problem can be solved numerically with the help of Lagrange multipliers, for the case of a general separable cost function. We focused our study further on linear cost functions, for which the optimization problem reduces to a linear programming (LP) problem. We assessed the difficulty of the problem in the multiple sink case, and studied in more detail the data gathering problem with a single sink. Namely, we found the rate allocation solution of this LP problem in a closed form.

Next, we have shown a comparison in terms of scaling laws between the two coding approaches. For a one-dimensional network, we provided scaling laws for asymptotic behavior and limits of the ratio of total costs associated to the two coding approaches. In particular, we showed that for some conditions on the correlation structure, the use of Slepian-Wolf coding techniques can result in unbounded gains in terms of total flow cost over simple coding by explicit communication. We also provided examples to support our results with two widely used correlation models for the Gaussian random field.

Furthermore, we presented a decentralized approximation algorithm for the rate allocation in the single sink case, that is based only on local information available at nodes from their neighborhood. This algorithm gives a solution very close to the optimum since the neglected conditioning is small in terms of rate for a correlation function that is sufficiently decaying with distance.

We also presented numerical results for a particular widely used data model, namely the Gaussian random field; our simulations show the possible benefits, in terms of total flow cost optimization, that can be obtained from a practical implementation of the algorithms we propose.

Summarizing, the main tradeoff between the two settings for rate allocation and transmission of correlated data across a network is

- If nodes are assumed to know the correlation structure (or equivalently, the

dependence on the distance), then they can employ Slepian-Wolf coding. In this case, source coding is complex while the transmission structure can be found with classical polynomial time network flow algorithms.

- If the correlation structure is not known a-priori, then side information is needed to reduce the entropy. In other words, the correlation structure is learned (explicitly) in a distributed manner through explicit communication. This leads to a simple source coding, but the transmission structure optimization is hard.

Further, we analyzed the complexity of the scenario when a combination of the the two coding approaches is used, namely Slepian-Wolf coding is done on clusters of nodes, and coding by explicit communication is done between clusters; namely, we conjecture that the problem of optimal clustering is NP-complete.

6.1.2 Queues Under Feedback Control

In the second part of this thesis we considered the modeling and analysis of the control of multiple access queues with partial information. We formulated a queueing problem in which the process of arrivals is *not* independent of the (partially observed) state of the queue. We studied the structure, properties, and long term behavior of control devices for accessing such queuing systems. We were inspired by the mechanism of TCP's flow control in our attempt to model the general problem of controlling systems for which the state is approximated locally in the transmitters by means of control actions and respective observations obtained about the state of the system. These control actions affect the state of the commonly shared resource.

In particular, we considered a bottleneck router with finite buffer and constant service rate, accessed by a variable number of sources that turn between on/off transmitting states independently of each other. The injection rates (control actions) are modeled as Bernoulli random variables: an active source decides between transmitting a packet or staying idle with a certain probability. The goal was to find the optimal control strategy a source should use, under the assumption that the only locally available information is the instantaneous feedback received from the router about the successful acceptance of the respective packet by the buffer.

First, we solved the corresponding optimal control problem. We have found an optimal control policy for our setting, based on the classical theory of control optimization by dynamic programming for partially observed Markov systems. Under this framework, the optimal value for the control is uniquely determined by a quantity that characterizes the system, based on the previous evolution, and which contains all the information about the state of the network that is available locally at the source, namely an *information state*. We showed how a good choice for this quantity is the conditional probability of the true state of the chain of active sources, given the history of controls and corresponding observations available to the source until the current time; the information state can be easily computed locally by means of a recursion formula. Our algorithm essentially provides a memoryless function that provides the optimal control as a function of the instantaneous value of the information state. This function depends on the performance criterion set for the algorithm. In our case, we chose as relevant cost function the expected throughput of the source, under constraints on the losses.

Our simulations showed that if the speed of variation of the number of active sources is inside some limits, and given an appropriate level for the loss threshold, then the control algorithm we propose is able to sense in an adaptive manner variations in the available bandwidth, and thus to use the channel efficiently, while still maintaining fairness with respect to the other users.

Our study focused next on performance criteria for our system, namely average throughput and average loss. We argued that an important tool to give analytical expressions (even if not in a closed form) for such performance criteria is the limit distribution of the information state. Thus, we further studied the existence, construction and properties of this invariant measure. We proved what we believe is an important theorem regarding the ergodic behavior of our system, namely the existence of a suitable invariant measure. Our main insight to tackle these problems was that by conditioning on information states, the arrivals process does become a process with independent increments. Since this conditioning term is Markov itself, this is how the model is rendered analytically tractable.

In short, we proved the theorem on the convergence in distribution as follows. We showed that the information state process forms a Markov chain over the simplex of N-dimensional probability vectors. For this chain, jumps outside cells of size smaller than a fixed constant happen with non-zero probability, no matter what is the state. This helps constructing a family of discretizations for the chain, which induces finite state Markov chains with recurrent classes. The corresponding recurrent restrictions have in the limit stationary distributions. Next, we proved that when the size of the cell becomes infinitely small, there exists an invariant probability measure over the simplex that is positive on the recurrent subset of the simplex, and null outside, which is the limit probability distribution we are looking for. For this last step of the proof we used results on the theory of Markov chains over uncountable metric spaces (Meyn and Tweedie provide an excellent coverage of the problem of Markov chains on general spaces [49]).

An interesting conclusion we draw from our results is that to make a system efficient in an information theoretic sense requires driving the system very close to the instability point. Consistent with this observation, for our model to be efficient the queue needs to be driven *past* the stability point and into the instability region. This is because, without being able to observe the state of the system but being able to observe when the system becomes unstable, this instability is the only indication we have that we are operating at peak efficiency. Note that this is a key idea behind the implementation of TCP (increase window size while packets get acknowledged, decrease when packets get lost). Also in our model, without forcing "TCP compatibility", we observe the exact same type of behavior (small increases in rate at positive ACKs, and drastic reductions at NACKs), thus providing some analytical evidence (on top of the unquestionable abundant empirical evidence) of the soundness of TCP's design.

6.2 Future Research

6.2.1 Rate Allocation and Transmission Structure Optimization in Networks with Correlated Data

The main direction of future work is the study of the combined approach of the two coding techniques. We have seen that Slepian-Wolf coding may provide important gains in terms of flow cost optimization, however it is more complex to perform than coding by explicit communication. It makes sense then to combine Slepian-Wolf coding and explicit communication, namely clusters of nodes to code data jointly with a Slepian-Wolf procedure, while the further reduction in data size due to communication between clusters to be based on explicit communication. For instance, since nodes that are close to the base station relay data from the rest of the network, they will have an important amount of side information available and thus, they do not need to employ Slepian-Wolf coding, reducing the complexity in terms of coding.

One other interesting issue is the power consumption equalization across the network in the case of sensor networks. The cost considered in this thesis seeks to minimize the total power consumption, however another interesting metric is the lifetime of the network. The lifetime of the network is usually measured in terms of the time of the first node failure (or depletion). The goal of this direction of research is the derivation of algorithms for evenly distributing the power load among loads. In the current approach, nodes close to the base station have to relay data from the whole network, and thus are more prone to quick depletion of resources.

A complete characterization of our problem would also have to include constraints concerning the maximum rate that can be supported by links. These constraints are given for example by the capacity of those links. We expect however the constraint problem to be significantly more difficult than the current setting.

6.2.2 Multiple Access Control with Partial Information

There are some more issues that still need to be dealt with about our queuing model. One of them is related to the existence of alternative characterizations of the optimal control law (other than the standard one given by Dynamic Programming, which is what we used so far). Namely, our conjecture is that if a controller forces average losses $= T$, then it is optimal. Note that in the case of complete state observations, this condition is sufficient for optimality. This problem is relevant because it may allow us to derive control laws—optimal and/or (theoretically sound) heuristics—of far lower complexity than the DP law, especially for the case of partial information.

Knowing that there exist ergodic limits for the behavior of our model raises the question of rates of convergence. How long does it take for the system to reach the ergodic regime, over what time scales is the system well described by its ergodic properties? Thus, it appears necessary to give some kind of analytical expression for the integral $\int pd\mu$, as a function of parameters the underlying Markov chain. We have explored this issue to some extent by means of numerical simulations, in Section 5.4. A direction of further research is the analytical study of this integral.

The concern about the complexity of the optimal DP control law is a recurrent one for us, as can be seen from the work that we express interest in above. The question of rates of convergence to ergodic limits is certainly interesting in its own

right, not only from a purely theoretical point of view. Knowing the exact conditions under which we can approximate the dynamics of our model with its ergodic behavior would enable us to develop greatly simplified control laws that still would have a sound theoretical basis. We believe both (a) that there is enough structure in this problem to develop remarkably simpler control algorithms than that based on implementing the DP equations, and (b) that this is a meaningful thing to do, since the benefit for applications would be substantial.

Bibliography

[1] A. Aaron and B. Girod. Compression with Side Information using Turbo Codes. In *Proc. of IEEE Data Compression Conference, DCC 2002*, pages 252–261, 2002.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communications Magazine*, 40(8):102–116, 2002.

[3] V. Ananthram and S. Verdú. Bits Through Queues. *IEEE Trans. Inform. Theory*, 42(1):4–18, 1996.

[4] S. Arora. Polynomial Time Approximation Scheme for Euclidean TSP and Other Geometric Problems. In *Proc. 37th Ann. IEEE Symp. on Foundations of Comput. Sci, IEEE Computer Society*, 1996.

[5] D.J. Baker and A. Ephremides. The Architectural Organization of a Mobile Radio Network via Distributed Algorithm. *IEEE Trans. on Comm.*, COM-29:56–73, 1981.

[6] M. Bern and P. Plassmann. The Steiner Problem with Edge Lengths 1 and 2. *Inform. Process. Lett.*, (32), 1989.

[7] D. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[8] D. Bertsekas and R. Gallager. *Data Networks (2nd ed)*. Prentice Hall, 1992.

[9] J.-H. Chang and L. Tassiulas. Fast Approximate Algorithms for Maximum Lifetime Routing in Wireless Ad-Hoc Networks. *Lecture Notes in Computer Science, Springer*, 1815:702–713, 2000.

[10] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley and Sons, Inc., 1991.

[11] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Networked Slepian-Wolf: Theory, Algorithms and Scaling Laws. *submitted to IEEE Trans. on Inf. Th.*, 2003.

[12] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Networked Slepian-Wolf: Theory and Algorithms. In *Proc. EWSN*, 2004.

[13] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. On Network Correlated Data Gathering. In *Proc. INFOCOM*, 2004.

[14] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Scaling Laws for Correlated Data Gathering. In *Proc. ISIT*, 2004.

[15] R. Cristescu and S.D. Servetto. Flow Control for Multiple-Access Queues. In *Proc. CISS*, 2001.

[16] R. Cristescu and S.D. Servetto. A Convergence Theorem for Controlled Queues with Partial Observations. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2002.

[17] R. Cristescu and S.D. Servetto. Queues under Feedback Control. *submitted*, 2003.

[18] O. Dousse, P. Thiran, and M. Hasler. Connectivity in Ad-Hoc and Hybrid Networks. In *IEEE Infocom*, pages 1079–1088, June 2002.

[19] J. Edmonds. Maximum Matching and a Polyhedron with 0,1-Vertices. *J. Res. Nat. Bur. Standards 69B*, pages 125–130, 1965.

[20] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the World with Wireless Sensor Networks. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[21] A. Farago, I. Chlamtac, and S. Bassagni. Virtual Path Network Topology Optimization using Random Graphs. In *IEEE INFOCOM*, 1999.

[22] S. Floyd and V. Jacobson. Random Early Detection for Congestion Avoidance. *IEEE/ACM Trans. Networking*, 1(4):397–413, 1993.

[23] J.J. Garcia-Luna-Aceves and E.L. Madruga. A Multicast Routing Protocol for Ad-Hoc Networks. In *IEEE INFOCOM*, 1999.

[24] M.R. Garey and D.S Johnson. *Computers and Intractability.* W.H. Freeman, 1979.

[25] M. Gastpar and M. Vetterli. On the Capacity of Wireless Networks: the Relay Case. In *Proc. of IEEE INFOCOM*, 2002.

[26] A. Goel and D. Estrin. Simultaneous Optimization for Concave Costs: Single Sink Aggregation or Single Source Buy-at-Bulk. In *ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[27] A. J. Goldsmith and P. Varaiya. Capacity, Mutual Information, and Coding for Finite-State Markov Channels. *IEEE Trans. Inform. Theory*, 42(3):868–886, 1996.

[28] P. Gupta and P.R. Kumar. The Capacity of Wireless Networks. *IEEE Transaction on Information Theory*, IT-46(2):388–404, 2000.

[29] B. Hajek. Stochastic Approximation Methods for Decentralized Control of Multiaccess Communications. *IEEE Trans. on Inf. Theory*, IT-31(2), 1985.

[30] B. Hajek. Cooling Schedules for Optimal Annealing. *Math. Oper. Res.*, (13):311–329, 1988.

[31] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge Univ. Press, 1991.

[32] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed Diffusion for Wireless Sensor Networking. *IEEE/ACM Trans. on Networking*, 11(1), February 2003.

[33] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, 1988.

[34] J.M. Kahn, R.H. Katz, and K.S.J. Pister. Next Century Challenges: Mobile Networking for 'Smart Dust'. In *Proc. MOBICOM, pp:271-278*, 1999.

[35] T. Kaijser. A Limit Theorem for Partially Observed Markov chains. *The Annals of Probability*, 3(4):667–696, 1975.

[36] F. Kelly. Mathematical Modeling of the Internet. In "Mathematics Unlimited - 2001 and Beyond" (Editors B. Engquist and W. Schmid). Springer-Verlag, Berlin, 2001. Available from `http://www.statslab.cam.ac.uk/ frank/PAPERS/`.

[37] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling Data Centric Routing in Wireless Sensor Networks. Technical Report 02-14, USC Computer Engineering, 2002.

[38] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice Hall, Inc., 1986.

[39] R. La and V. Ananthram. Utility Based Rate Control in the Internet for Elastic Traffic. In *To appear in the IEEE/ACM Transactions on Networking. Available from* `http://www.eecs.berkeley.edu/ ananth/`.

[40] H. Larson and B. Shubert. *Probabilistic Models in Engineering Sciences*. John Wiley and Sons, Inc., 1979.

[41] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys(ed.). *The Traveling Salesman Problem*. Wiley, 1990.

[42] S.-J. Lee. Routing and Multicast Strategies in Wireless Mobile Networks. Technical report, Dept. of Computer Sc., UCLA, Los Angeles, 2000.

[43] S. Lindsey, C. S. Raghavendra, and K. Sivalingam. Data Gathering in Sensor Networks using the Energy*Delay Metric. In *Proc. of IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, April 2001.

[44] S.H. Low and D.E. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence. *IEEE/ACM Trans. Networking*, 7(6):861–875, 1999.

[45] S.H. Low, F. Paganini, and J.C. Doyle. Internet Congestion Control. *IEEE Control Syst. Mag.*, 22(1):28–43, 2002.

[46] D. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[47] M. Lundy and A. Mees. Convergence of an Annealing Algorithm. *Mathematical Programming*, (34), 1986.

[48] D. Marco, E. Duarte-Melo, M. Liu, and D.L. Neuhoff. On the Many-to-One Transport Capacity of a Dense Wireless Sensor Network and the Compressibility of its Data. In *Proc. 2nd International Workshop on Information Processing (IPSN'03). Lecture Notes in Computer Science vol. 2634, Springer: New York*, pages 1–16, 2003.

[49] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Springer Verlag, 1993.

[50] S. Mitter. Control with Limited Information. Plenary talk ISIT 2002, and *European Journal of Control Vol. 7, no.1, 2001.*

[51] C. Perkins. *Ad Hoc Networking*. Adison-Wesley Pub Co, 2000.

[52] C. Perkins and E.M. Royer. Ad Hoc on Demand Distance Vector (AODV) Routing. *Internet draft*, Aug. 1998.

[53] B. Porat. *Digital Processing of Random Signals*. Prentice Hall, 1994.

[54] G.J. Pottie and W.J. Kaiser. Wireless Integrated Sensor Networks. *Communications of the ACM*, 43(5):51–58, 2000.

[55] S. Pradhan. *Distributed Source Coding Using Syndromes (DISCUS)*. PhD thesis, U.C. Berkeley, 2001.

[56] S. Pradhan and K. Ramchandran. Distributed Source Coding Using Syndromes (DISCUS): Design and Construction. In *Proc. DCC'99 Data Compression Conference, IEEE Comput. Soc.*, pages 414–23, March 1999.

[57] J. Rabaey, M.J. Ammer, J.L. da Silva, D. Patel, and S. Roundy. PicoRadio Supports Ad-Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7):42–48, 2000.

[58] W. Rabiner-Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proc. of the 33rd International Conference on System Sciences (HICSS '00)*, January 2000.

[59] R. Ramanathan and R. Rosales-Hain. Topology Control of Multihop Wireless Networks using Transmit Power Adjustments. In *IEEE INFOCOM*, 2000.

[60] C. Reidys and P. Stadler. Combinatorial Landscapes. *SIAM Review*, 44:3–54, 2002.

[61] G. Robins and A. Zelikovsky. Improved Steiner Tree Approximation in Graphs. In *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms, ACM-SIAM*, 2000.

[62] V. Rodoplu and T. H. Meng. Minimum Energy Mobile Wireless Networks. *IEEE Jour. Selected Areas Comm.*, 17(8):1333–1344, August 1999.

[63] H.F. Salama, D.S. Reeves, and Y. Viniotis. A Distributed Algorithm for Delay-Constrained Unicast Routing. In *IEEE INFOCOM*, 1997.

[64] A. Scaglione and S. Servetto. On the Interdependence of Routing and Data Compression in multi-hop sensor networks. In *ACM MOBICOM*, 2002.

[65] V. Sharma and S.K. Singh. Entropy and Channel Capacity in the Regenerative Setup with Applications to Markov Channels. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT) 2001*, 2001.

[66] D. Slepian and J.K. Wolf. Noiseless Coding of Correlated Information Sources. *IEEE Trans. Information Theory*, (IT-19):471–480, 1973.

[67] S. Tatikonda. *Control Under Communication Constraints*. PhD thesis, MIT, September 2000.

[68] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed Topology Control for Wireless Multihop Ad-Hoc Networks. In *Proc. of IEEE INFOCOM*, pages 1388–1397, 2001.

Curriculum Vitae

Răzvan Cristescu

- **PERSONAL DATA:**
  - Date of birth: 19 Dec. 1974
  - Place: Ploiesti, Romania
- **EDUCATION:**
  - *Oct. 2000 - Mar. 2004* **Ph.D.** student, Laboratory of Audio-Visual Communications, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
  - *Oct. 1999 - July 2000* Graduate School in Communication Systems, SSC, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
  - *Oct. 1998 - Dec. 2000* **Lic.Tech.** degree, Laboratory of Computer and Information Science, Helsinki University of Technology, Finland.
  - *Oct. 1998 - Sept. 1999* **DEA** in Systems Modeling and Simulation, Polytechnic University of Bucharest, Romania.
  - *Oct. 1993 - Sept. 1998* **M.Sc.** degree, Department of Engineering Sciences, Electrical Engineering and Computer Science, Polytechnic University of Bucharest, Romania.
    (graduated as 'first of promotion' '98 with highest GPA).
- **JOURNAL PAPERS:**

[1.] R. Cristescu, S.D. Servetto, *Queues under Feedback Control*, submitted, 2003.

[2.] R. Cristescu, B. Beferull-Lozano, M. Vetterli, *Networked Slepian-Wolf: Theory, Algorithms and Scaling Laws*, submitted, 2003.

[3.] R. Cristescu, B. Beferull-Lozano, M. Vetterli, R. Wattenhofer, *Network Correlated Data Gathering with Explicit Communication: NP-Completeness and Algorithms*, submitted, 2004.

- **CONFERENCE PAPERS:**

At EPFL:

[1.] R. Cristescu, B. Beferull-Lozano and M. Vetterli, *Scaling Laws for Correlated Data Gathering*, to appear in Proc. IEEE ISIT, 2004, Chicago, IL.

[2.] D. Ganesan, R. Cristescu and B. Beferull-Lozano, *Power-Efficient Sensor Placement and Transmission Structure for Data Gathering under Distortion Constraints*, to appear in Proc. IPSN, 2004, Berkeley, CA.

[3.] R. Cristescu, B. Beferull-Lozano and M. Vetterli, *On Network Correlated Data Gathering*, to appear in Proc. IEEE INFOCOM 2004, Hong Kong.

[4.] R. Cristescu, B. Beferull-Lozano and M. Vetterli, *Networked Slepian-Wolf, Theory and Algorithms*, in Proc. IEEE EWSN 2004, Berlin, Germany.

[5.] T. Ajdler, R. Cristescu, P. L. Dragotti, M. Gastpar, I. Maravic and M. Vetterli, *Distributed Signal Processing and Communications: on the Interaction of Sources and Channels*, in Proc. IEEE ICASSP, 2003, Hong Kong, (invited paper).

[6.] R. Cristescu and M. Vetterli, *Power Efficient Gathering of Correlated Data: Optimization, NP-Completeness and Heuristics*, in Proc. ACM MobiHoc 2003 (poster session), Annapolis, MD.

[7.] R. Cristescu, S. D. Servetto, *A Convergence Theorem for Controlled Queues with Partial Observations*, IEEE ISIT '02, Lausanne, Switzerland.

[8.] R. Cristescu, S. D. Servetto, *Flow Control for Multiple-Access Queues*, Proc. CISS '01, Baltimore, MD.

At HUT:

[1.] R. Cristescu, T. Ristaniemi, J. Joutsensalo, and J. Karhunen, *CDMA Delay Estimation using Fast ICA Algorithm*, in Proc. PIMRC2000, London, UK.

[2.] R. Cristescu, T. Ristaniemi, J. Joutsensalo, and J. Karhunen, *Delay Estimation in CDMA Communications Using A FastICA Algorithm*, in Proc. ICA2000, Helsinki, Finland.

[3.] R. Cristescu, J. Joutsensalo, J. Karhunen, and E. Oja, *A Complexity Minimization Approach for Estimating Fading Gaussian Channel in CDMA Communications*, Proc. ICA2000, Helsinki, Finland.

[4.] R. Cristescu, T. Ristaniemi, J. Joutsensalo, and J. Karhunen, *Blind Separation of Convolved Mixtures for CDMA Systems*, in Proc. EUSIPCO2000, Tampere, Finland.

[5.] R. Cristescu, J. Joutsensalo, and T. Ristaniemi, *Fading Channel Estimation by Mutual Information Minimization for Gaussian Stochastic Processes*, in Proc. IEEE ICC2000, New Orleans, LA.

[6.] R. Cristescu, J. Karhunen, J. Joutsensalo, and T. Ristaniemi, *Blind Separation Methods for CDMA Communications*, in Proc. The Learning Workshop 2000, Snowbird, UT (invited paper).

[7.] A. Hyvärinen, R. Cristescu and E. Oja, *A fast algorithm for estimating overcomplete ICA bases for image windows*, in Proc. IEEE IJCNN99, Washington, DC.

- **WORK EXPERIENCE:**
  - *Sept. '00 - present*, researcher, teaching assistant, Laboratory of Audio-Visual Communications (LCAV), Swiss Federal Institute of Technology (EPFL), Lausanne. Member of the Mobile Information and Communication Systems (MICS) National Centre of Competence in Research (NCCR). Project title: 'Distributed Communications'.
    - Co-teaching a course in Advanced Signal Processing.
    - TA for courses in Signal Processing and Advanced Signal Processing.
    - Supervised 3 student semester projects.
  - *Oct. '99 - July '00*, researcher, Communication Systems, Swiss Federal Institute of Technology (EPFL), Lausanne, Project title: 'Distributed Flow Control with Partial Information'.
  - *Oct. '98 - Sept. '99*, researcher, Laboratory of Computer and Information Science , Helsinki University of Technology, ICA research group Project title: 'Independent Component Analysis Applications in Telecommunications'. Funded by the Academy of Finland.
  - Reviewer for: IEEE Trans. on Signal Processing, IEEE INFOCOM, ACM Sensys, IEEE ICC, IEEE EWSN, IPSN.

- **AWARDS:**
  - Licentiate of Technology graduation award, Helsinki Univ. of Technology.
  - ACM student travel grant to attend SIGMETRICS2001 (Boston, MA, US).
  - IEEE student travel grant to present a paper in ICC'2000 (International Conference on Communications, New Orleans, LA, US).
  - Graduated 'first of promotion '98' from EE and CS department, Polytechnic University of Bucharest.
  - Romanian Government Merit Fellowship (offered to less than 5% of the students nationwide each semester), 1993-1998.
  - Mathematics National Contest -'89 -mentioned, '88-participant.
  - Informatics National Contest - '93 - participant.

THE LITTLE COUSIN SERIES IN MATHEMATICAL SIGNAL PROCESSING
Editor: Martin Vetterli

The Columbia series

[1.] Karlsson, Gunnar David. *Subband Coding for Packet Video.* CU/CTR/TR 137-89-16, May 1989.

[2.] Linzer, Elliot Neil. *Arithmetic Complexity and Numerical Properties of Algorithms involving Toeplitz Matrices.* October 1990.

[3.] Kovačević, Jelena. *Filter Banks and Wavelets: Extensions and Applications.* CU/CTR/TR 257-91-38, September 1991.

[4.] Uz, Kamil Metin. *Multiresolution Systems for Video Coding.* CU/CTR/TR 313-92-23, May 1992.

[5.] Radha, Hayder M. Sadik. *Efficient Image Representation using Binary Space Partitioning Trees.* CU/CTR/TR 343-93-23, December 1992.

[6.] Nguyen, Truong-Thao. *Deterministic Analysis of Oversampled A/D Conversion and Sigma/Delta Modulation, and Decoding Improvements using Consistent Estimates.* CU/CTR/TR 327-93-06, February 1993.

[7.] Herley, Cormac. *Wavelets and Filter Banks.* CU/CTR/TR 339-93-19, April 1993.

[8.] Garrett, Mark William. *Contributions toward Real-Time Services on Packet Switched Networks.* CU/CTR/TR 340-93-20, April 1993.

[9.] Ramchandran, Kannan. *Joint Optimization Techniques in Image and Video Coding with Applications to Multiresolution Digital Broadcast.* June 1993.

[10.] Shah, Imran Ali. *Theory, Design and Structures for Multidimensional Filter Banks and Applications in Coding of Interlaced Video.* CU/CTR/TR 367-94-14, December 1993.

[11.] Hong, Jonathan Jen-I. *Discrete Fourier, Hartley, and Cosine Transforms in Signal Processing.* CU/CTR/TR 366-94-13, December 1993.

[12.] Ortega, Antonio. *Optimization Techniques for Adaptive Quantization of Image and Video under Delay Constraints.* CU/CTR/TR 374-94-21, June 1994.

The Berkeley years

[13.] Park, Hyung-Ju. *A Computational Theory of Laurent Polynomial Rings and Multidimensional FIR Systems.* Coadv. with Tsit-Yuen Lam, Mathematics, U.C. Berkeley. UCB/ERL M95/39, May 1995.

[14.] Cvetković, Zoran. *Overcomplete Expansions for Digital Signal Processing.* UCB/ERL M95/114, December 1995.

[15.] McCanne, Steven Ray. *Scalable Compression and Transmission of Internet Multicast Video.* Coadv. with Van Jacobson, Lawrence Berkeley National Laboratory. UCB/CSD 96/928, December 1996.

[16.] Goodwin, Michael Mark. *Adaptive Signal Models: Theory, Algorithms, and Audio Applications.* Coadv. with Edward A. Lee, EECS, U.C. Berkeley. UCB/ERL M97/91, December 1997.

[17.] Goyal, Vivek K. *Beyond Traditional Transform Coding.* UCB/ERL M99/2, September 1998.

[18.] Chang, Sai-Hsueh Grace. *Image Denoising and Interpolation based on Compression and Edge Models.* Coadv. with Bin Yu, Statistics, U.C. Berkeley. UCB/ERL M99/57, Fall 1998.

The Lausanne time

[19.] Prandoni, Paolo. *Optimal Segmentation Techniques for Piecewise Stationary Signals.* EPFL 1993(1999), June 1999.

[20.] Lebrun, Jérôme. *Balancing MultiWavelets.* EPFL 2192(2000), May 2000.

[21.] Weidmann, Claudio. *Oligoquantization in Low-Rate Lossy Source Coding.* EPFL 2234(2000), July 2000.

[22.] Balmelli, Laurent. *Rate-Distortion Optimal Mesh Simplification for Communication.* EPFL 2260(2000), September 2000.

[23.] Marziliano, Pina. *Sampling Innovations.* EPFL 2369(2001), April 2001.

[24.] Horbelt, Stefan. *Splines and Wavelets for Image Warping and Projection.* EPFL 2397(2001), May 2001.

[25.] Hasler, David. *Perspectives on Panoramic Photography.* EPFL 2419(2001), July 2001.

[26.] Do, Minh N. *Directional Multiresolution Image Representations.* EPFL 2500(2001), November 2001.

[27.] Dragotti, Pier Luigi. *Wavelet Footprints and Frames for Signal Processing and Communications.* EPFL 2559(2002), April 2002.

[28.] Pecenovic, Zoran. *Integrating Visual and Semantic Descriptions for Effective, Flexible and User-Friendly Image Retrieval.* EPFL 2679(2002), October 2002.

[29.] Hu, Xiao-Yu. *Low-Delay Low-Complexity Error-Correcting Codes on Sparse Graphs.* EPFL 2681(2002), November 2002.

[30.] Gastpar, Michael. *To Code Or Not To Code.* EPFL 2687(2002), December 2002.