PROBLEMS IN SPATIO–TEMPORAL MODELLING, KRIGING, AND PREDICTION OF COMPUTER NETWORK TRAFFIC

by

Joel M. Vaughan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2012

Doctoral Committee:

Professor George Michailidis, Co-Chair
Associate Professor Stilian A. Stoev, Co-Chair
Professor Vijayan N. Nair
Professor Kerby A. Shedden
Assistant Professor Veronica Berrocal

To Mom, Dad, and Josh.

ACKNOWLEDGEMENTS

Thank you to everyone who made this thesis possible. There are a great many people who helped in many ways, and I appreciate all your help.

I owe a great deal of thanks to my advisers, Prof. Stilian Stoev and Prof. George Michailidis, for their patience, the opportunities they have given me, including this project, and their guidance and support over the last several years. Thank you. Thank you to my committee, Prof. Nair, Prof. Shedden, and Prof. Berrocal, for the time and effort in helping me improve this document, and also for their support and advice on other matters. Thanks to the rest of the faculty in the Statistics Department, who taught me many things, in class, seminars, and off–line conversations. Thank you to Brenda Gunderson for the opportunities and support in learning to be a better teacher, and for always being there with an open ear and candy jar. Thank you to the staff of the department, especially Lu Ann and Mary Ann, who are always a joy to talk to, and have helped me in more ways than I can count.

Thank you to my friends in the program. Eric Laber and Yang Yang were continued sources of inspiration, encouragement, and assistance. Kohinoor Dasgupta and Toshiya Hoshikawa were understanding and helpful friends and officemates. Matt Linn and Carrie Hosman were always looking out for me, and I greatly appreciate it. Thanks to the many others who took the time to answer my questions

or participate in student seminars, reading groups, and study sessions. You enriched the experience and made my time here much better. I especially want to thank Yizao Wang, Ali Shonjaie, Sahar Zangeneh, and Gbenga Olumulade.

Thank you to my family and friends, who provided the support an encouragement I needed. Many thanks to brother, Joshua, for always being there for me and consistently reminding me to stay intellectually curious, and my aunt, Elaine, for her support. Special thanks also go to Greg Buckingham, Matthew Huskey, and David Childers for their timely wisdom, perspective, and help throughout this process. Finally, thank you to my parents, Patricia and Carleton, for their ongoing patience and support. This would not have been possible without you.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Problems in Spatio–Temporal Modelling, Kriging, and Prediction of Computer Network Traffic

by

Joel M. Vaughan

Co-Chairs: George Michailidis and Stilian A. Stoev

In order to maintain consistent quality of service, engineers face the task of monitoring the traffic fluctuations on the individual links making up a computer network. However, due to resource constraints and limited access, it is often not possible to directly measure the traffic on all links. This work explores a statistical framework for simultaneously modeling the traffic levels on links across an entire network and using the model to solve a variety of statistical problems, including prediction of traffic on unobserved links and the detection of statistical anomalies.

We begin by examining some of the important types of network traffic data and features of the traffic. These features present interesting challenges but also provide important structure that is used throughout this work. We next develop a probabilistic spatio–temporal model for large scale computer networks that is based on physical properties of computer networks. This model simultaneously describes the traffic level on all the links of the network, and how these levels fluctuate over time. We next move on to study the so–called kriging and predic-

tion problems, where we use observed traffic measurements on a small subsets of the links of a network to predict the traffic on other (unobserved) links in the network. We then explore an application of this prediction technique to anomaly detection. Finally, we develop an alternative model that more explicitly incorporates the dependence in traffic that arises due to certain mechanisms in the protocols that govern network behavior. We conclude by discussing the strengths and weaknesses of these two approaches.

CHAPTER I

Introduction

This dissertation focuses on the statistical modeling and analysis of traffic in computer networks. The developed methodology can be used to predict traffic and detect anomalies as well as aide practitioners and network traffic engineers in managing modern large–scale communication systems such as backbone computer networks.

The statistical analysis of computer network traffic has received considerable attention in the literature, but many open problems remain.

Understanding the behavior of network traffic is important for assessing network performance [18] and on the potential for development of more suitable protocols [17, 18].

Modern computer networks have become an integral part of people's lives due to their ubiquity and the introduction of sophisticated applications and services, such as packet telephony and television, social and business networking, advanced customer services, etc. (For more details, see [11] and [30].) For networks to operate in a seamless manner and carry out these functions successfully, capacity planning, fault diagnosis, traffic forecasting and provisioning and efficient routing protocol configuration become important issues to be addressed [19]. To

do so, various types of network data can be collected that differ in their granularity, accuracy, volume and delay [14].

The analysis of computer network traffic on a single link has been the focus of a number of studies. It was discovered that traffic exhibits long–range dependence over time, which is closely related to the presence of heavy tails in file sizes, connection durations, and user and application behavior. A number of mechanistic models that take into account individual users' behavior were proposed to account for these empirical facts (see e.g. [18]) While the probabilistic behavior of network traffic on a single link has been well studied, similar models have not been developed to describe the behavior over all the links in the network simultaneously. Thus, an important open problem is the development of a global model for the traffic on an entire computer network. This problem, and applications of its solution, are the focus of this dissertation.

We begin by introducing some important terminology and background on the function of computer networks in Chapter II. We also describe the types of data used to study the behavior of computer network traffic and the Internet2 network, which is used as an example throughout this dissertation. In Chapter III, we develop a global probabilistic models for network traffic. We begin this chapter by reviewing important work describing the temporal behavior of traffic on a single link, and then develop a probabilistic model that simultaneously models the behavior of the traffic on all links in the network. In Chapter IV, we consider the Network Kriging problem, or the problem of using traffic measurements from a small subset of links to predict the traffic on other, unobserved links. After introducing the problem, we show how auxiliary network data may be used to develop a low–dimensional model for important statistical parameters. We con-

clude this chapter by demonstrating how the kriging methodology may be used for detecting anomalous traffic. In Chapter V, we discuss an alternative model that explicitly models the feedback mechanism present in certain types of computer network traffic, and the solution to the Kriging problem under this model. Finally, in Chapter VI, we present results on real network data from Internet2, and compare the two models.

CHAPTER II

Network Preliminaries and Data Description

This chapter provides important background information for the ideas developed in this work. In it, we discuss the basics of computer networks, relevant protocols, and important features of the data we work with. We begin by introducing Internet2, the backbone network we analyze throughout this document. We then describe some important concepts related to computer networks in general, and conclude by describing the types and important properties of computer network traffic data.

2.1 Internet2 Network Description

This section provides details concerning the Internet2 network, the network used as an example throughout this document. Internet2 is a high speed network that forms the backbone for educational and research institutions throughout the United States.

The Internet2 network consists of 26 unidirectional links, which are illustrated in Figure 2.1. In order to simplify notation, each link was assigned an id number. Table 2.1 provides the mapping from the link id numbers to the source and destination of each link, as well as the link capacities. At the time of the data collection, most links had a 10 Gb/s (Gigabits per second) capacity, with the exception of 4

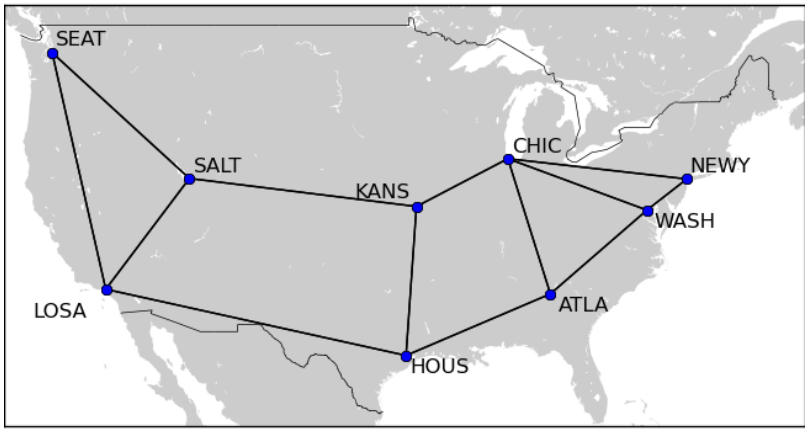


Figure 2.1: Internet2 Topology: The network consists of nine routers and 26 uni–directional links, arranged as displayed in this figure. [10]

Link ID	Source → Destination	Capacity
1,2	Los Angeles → Seattle	10 Gb/s
3,4	Seattle → Salt Lake City	10 Gb/s
5,6	Los Angeles → Salt Lake City	10 Gb/s
7,8	Los Angeles → Houston	10 Gb/s
9, 10	Salt Lake City → Kansas City	10 Gb/s
11, 12	Kansas City → Houston	10 Gb/s
13, 14	Kansas City → Chicago	20 Gb/s
15, 16	Houston → Atlanta	10 Gb/s
17, 18	Chicago → Atlanta	10 Gb/s
19, 20	Chicago → New York	10 Gb/s
21, 22	Chicago → Washington	10 Gb/s
23, 23	Atlanta → Washington	10 Gb/s
25, 26	Washington → New York	20 Gb/s

Table 2.1: ID's of the 26 links of the Internet2 backbone. Odd Link ID's correspond to the forward and the even to the reverse; i.e. Link 15 is the Houston to Atlanta link and Link 16 is the Atlanta to Houston link.

links: Chicago to Kansas City, Kansas City to Chicago, New York to Washington, and Washington to New York. These four links actually were comprised of two 10 Gb/s capacity cables, for a total capacity of 20 Gb/s.

2.2 Basics of Computer Networks and Protocol Mechanisms

Fundamentally, computer networks consist of a set of nodes (typically routers, switches, or end hosts) connected by physical links. Data, in the form of packets, are transmitted from one node (called the *source*) to a second node (*destination*).

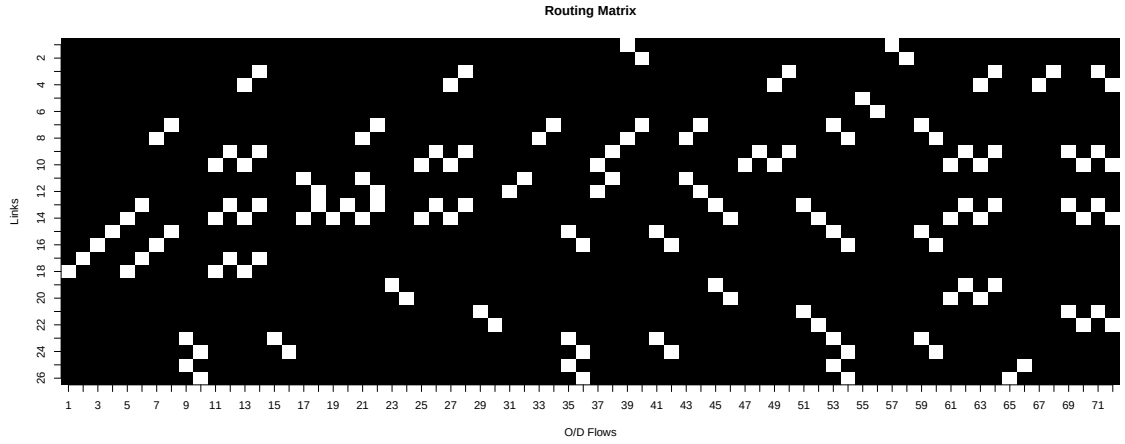


Figure 2.2: A graphical representation of the routing matrix A from Internet2 at the time of analysis. Black indicates a 0 in the matrix, while white indicates 1. The figure is oriented as the matrix, with rows corresponding to links 1–12, and columns to flows 1–72.

This transmission occurs over a pre–specified collection of links, called a *route*. Typically, each node in the network can serve both as a source and a destination, resulting in $\mathcal{J} = n(n-1)$ origin–destination *flows*, although there could be fewer. Each of these flows takes a predetermined path of links across the network. These routes can be summarized by a *routing matrix* $A = (a_{\ell j})_{L \times \mathcal{J}}$, where

$$a_{\ell j} = \begin{cases} 1 & \text{link } \ell \text{ used in route } j \\ 0 & \text{link } \ell \text{ not used in route } j \end{cases} \qquad 1 \le \ell \le L,\ 1 \le j \le \mathcal{J}.$$

We assume that this matrix is known and fixed in time. A graphical depiction of the routing matrix of the Internet2 at the time of the study shown in 2.2.

We are interested in the traffic on each flow in terms of both packets and bytes. Formally, let $X_j^P(t)$ represent the the number of packets transmitted on flow $j \in \{1, \ldots, \mathcal{J}\}$ during the time interval $((t-1)h, th]$, for some fixed period $h > 0$ that may be chosen according to context. Similarly, let $X_j^B(t)$ represent the the number of bytes transmitted on flow $j \in \{1, \ldots, \mathcal{J}\}$ during the time interval $((t-1)h, th]$. We are also interested in the traffic on each of the L links during

this time interval, which we denote $Y_\ell^P(t)$ and $Y_\ell^B(t)$ for packets and bytes respectively. These quantities may be simplified by collecting the flow and link measurements into the vectors $X^B(t) := (X_j^B)_{1\le j\le \mathcal{J}}$,$X^P(t) := (X_j^P)_{1\le j\le \mathcal{J}}$, $Y^B(t) = (Y_\ell^B(t))_{1\le \ell\le L}$, and $Y(t) = (Y_\ell^P(t))_{1\le \ell\le L}$. Additionally, define $X(t) := (X^P(t), X^B(t))^T$ and $Y(t) := (Y^P(t), Y^B(t))^T$. Since the traffic on each origin–destination flow must be carried across links, the traffic on any link ℓ is clearly the sum of the traffic carried by all the flows using the link. This may be more succinctly expressed via the important *routing equations*:

$$Y^B(t) = AX^B(t) \qquad Y^B(t) = AX^B(t) \tag{2.1}$$

or, combining the packet and byte traffic loads in a single vector, we have:

$$Y(t) = \begin{pmatrix} Y^P(t) \\ Y^B(t) \end{pmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{pmatrix} X^P(t) \\ X^B(t) \end{pmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} X(t) \tag{2.2}$$

This relationship between $X(t)$ and $Y(t)$ is crucial to the modeling discussed in upcoming sections. This relationship holds provided that the round trip time (RTT) of the traffic traversing the network is much smaller than the length of the interval over which the traffic is measured (h). This assumption holds for all the data used within this work.

In addition to the relationship between $X(t)$ and $Y(t)$, there is also an important relationship between $X^B(t)$ and $X^P(t)$ that we will exploit in upcoming sections. We will model $X^B(t)$ and $X^P(t)$ as sums of related subflows of specific types of traffic.

As an example network, we will refer to the Internet 2 network [10], the backbone education and research network in the United States. The topology is shown

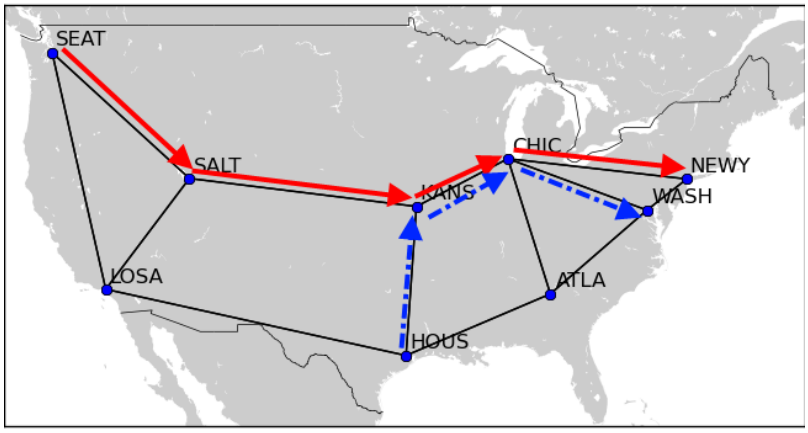


Figure 2.3: Example Origin–Destination Flows depicted on Internet2. Each colored set of arrows represent a different Origin–Destination Flow.

in Figure 2.1. The network consists of nine routers, with $9 \times 8 = 72$ origin–destination flows traveling across 26 physical uni–directional links. The network is described in more detail in Section 2.1.

We now discuss some of the engineered features in computer networks that motivate our models. Computer networks employ a variety of communication protocols, organized into several layers. A complete discussion of the hardware and software involved in heterogeneous computer networks is well beyond the scope of this work; however, we refer the curious reader to [7] or [19] and the numerous references within. One of the most prevalent features of the communication protocol is the layered approach. At each layer, the packets from the previous layer are encased or wrapped into packets at the lower layer. There are four important layers: the *application* layer, which consists of the data the user or end–system interacts with; the *transport* layer, which describes the mechanism by which this data is transmitted; the *network* layer, describing how packets traverse heterogeneous, independent, and interconnected networks; and finally the *link* layer, where the data is packaged to traverse the physical medium.

The two most relevant protocols to this work are transport layer protocols: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP is used when the reliability of the data is essential, such as in file transfer and ssh. This protocol includes an acknowledgment mechanism to ensure accurate delivery. Whenever the destination host receives a packet, it sends a small packet to the source to acknowledge the receipt of the packet. This packet, known as an acknowledgment packet, or ACK, allows the source to ensure that the destination has received all the packets in the correct order. Thus, TCP traffic on a particular origin–destination flow induces traffic on the reverse flow. On the other hand, UDP is used when speed is more important than reliability, such as Voice over IP (VoIP) or streaming video. Data sent via the UDP protocol does not have a built–in reliability mechanism other than the ability of the destination user to request a retransmission if a substantial amount of data is lost.

2.3 Data Description

In this section, we discuss the two types of data available and illustrate the features relevant to the modeling decisions discussed later in this thesis.

Netflow Data: Netflow is a packet sampling technology developed by Cisco Systems [5]. Netflow data is collected at each of the nine routers that comprise Internet2. (Similar technologies exist for hardware from other manufacturers.) Netflow records consist of packet header information for similar packets over a short period of time. This header information includes such things as source and destination IP address and port, time, number of packets, number of bytes, and transmission protocol. An example of the raw data, together with its fields, may be seen in Table 2.2. Pairing this information with some additional auxiliary in-

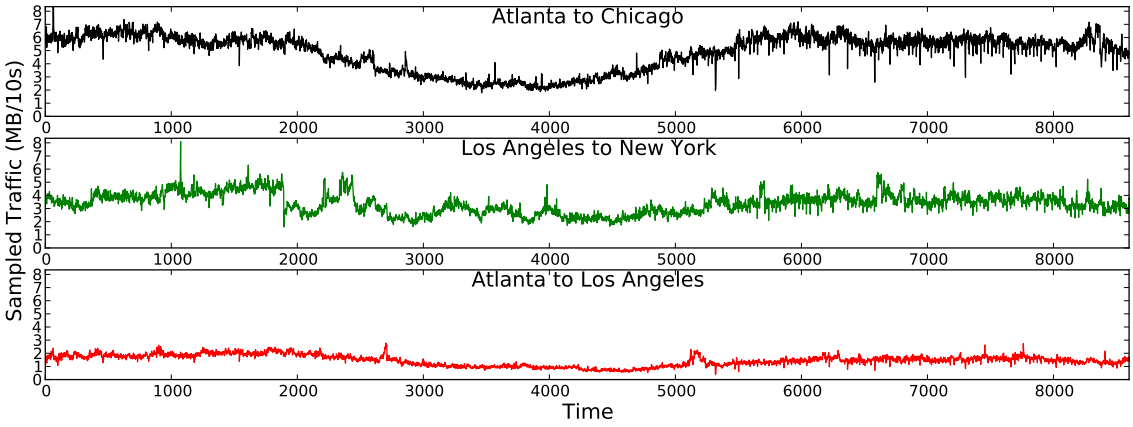


Figure 2.4: Example traffic trace of Origin–Destination Flows reconstructed from Netflow measurements.

formation, it is possible to create a mapping of IP (Internet Protocol) addresses to the nine routers as sources and/or destinations. With the mapping, it is then possible to assign each Netflow record to one of the (72) origin–destination flows. This allows us to examine the data from a variety of perspectives. It allows us to reconstruct time series for the origin–destination flows directly, in terms of either packets or bytes, i.e. to recover $X^B(t)$ or $X^P(t)$. Additionally, it allows us to further divide these flows by protocol, packet size, both of these features, or even other features in the data. It is with this technique that we obtain the data displayed in both Figures 2.6 and 2.9. Examples of reconstructed time series are shown in Figure 2.4.

Netflow data does have some major limitations. The data take non-trivial resources to collect, and as a result, a subset of sample packets are recorded rather than all packets traversing a given node. Once the data have been recorded and transmitted, it takes even more resources to process the data in order to reconstruct the traffic flows as described above. Last but not least, the small sampling rate and implementation issues lead to highly variable and often biased estimates

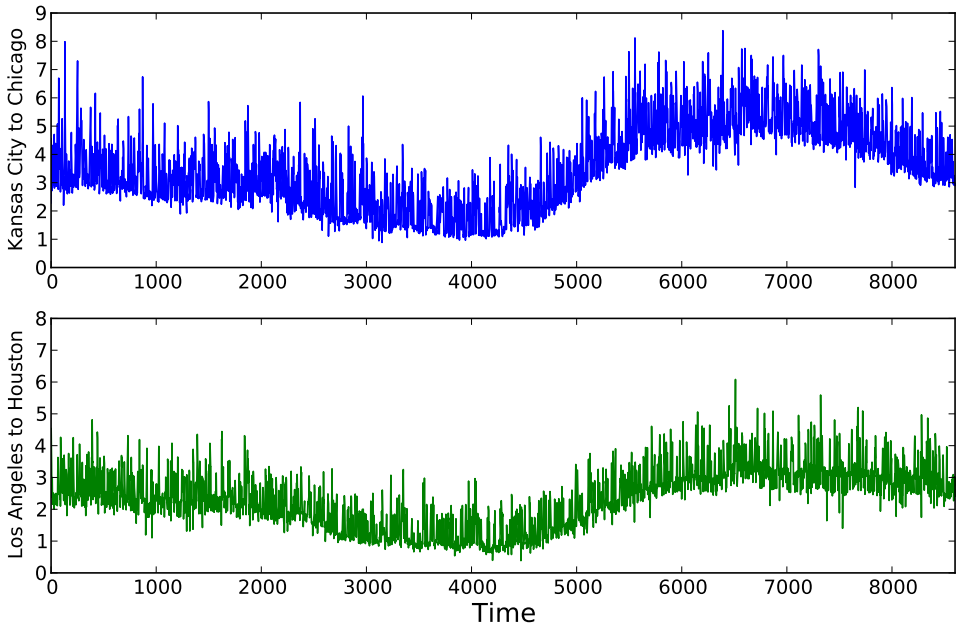


Figure 2.5: Example Origin–Destination Flows depicted on Internet2. Each colored set of arrows represent a different Origin–Destination Flow.

of traffic flows. Thus, these are excellent data to learn the structure of a traffic model, but cannot be used directly for on-line traffic prediction or model estimation.

Link Data: Each physical link in the network may be directly monitored, and counters are maintained for both the number of packets and the amount of data (in bytes) that traverse the link over a given period of time. Compared to the Netflow data, this data is easier to collect, process, store, and transmit without the same strain on network resources. Nonetheless, the resource demand for such data does increase with the size of the network. Figure 2.5 illustrates such traffic traces.

Relevant Protocols: Throughout this work, we focus exclusively on only two transport protocols: TCP and UDP. Table 2.3 shows the distribution of transport protocols used by the traffic, calculated via analysis of the Netflow data. We see

	dpkts	doctets	first	last	srcaddr	dstaddr	nexthop	input	output	srcport	dstport	prot	tos	tcp_flags	src_mask	dst_mask	src_as	dst_as
1	1	48	2515315985.00	2515315985.00	207.46.104.0	131.247.0.0	64.57.28.36	91	95	1863	51692	6	0	24	20	16	8075	5661
2	2	141	2515286360.00	2515328644.00	207.46.104.0	164.58.112.0	156.110.203.5	91	61	1863	7249	6	0	24	20	16	8075	5078
3	1	329	2515313930.00	2515313930.00	207.46.104.0	140.180.0.0	64.57.28.36	91	95	1863	61817	6	0	24	20	20	8075	88
4	1	304	2515311577.00	2515311577.00	207.46.104.0	128.252.16.0	64.57.28.36	91	95	1863	1030	6	0	24	20	23	8075	2552
5	1	149	2515310666.00	2515310666.00	207.46.104.0	64.113.72.0	164.113.238.194	91	63	1863	59977	6	0	24	20	19	8075	2698
6	1	48	2515316912.00	2515316912.00	207.46.104.0	155.246.168.0	64.57.28.36	91	95	1863	49184	6	0	24	20	16	8075	16889
7	1	63	2515332088.00	2515332088.00	164.58.112.0	24.71.120.0	0.0.0.0	59	91	18462	37312	17	0	0	16	0	5078	0
8	1	63	2515321506.00	2515321506.00	164.58.112.0	24.183.48.0	0.0.0.0	59	91	18462	5218	17	0	0	16	0	5078	0
9	1	52	2515302353.00	2515302353.00	164.58.112.0	74.125.88.0	0.0.0.0	59	113	50006	80	6	0	16	16	0	5078	0
10	1	40	2515330196.00	2515330196.00	164.58.112.0	209.85.224.0	0.0.0.0	59	113	1832	80	6	0	20	16	0	5078	0
11	1	40	2515282637.00	2515282637.00	164.58.112.0	209.191.88.0	0.0.0.0	59	113	53339	80	6	0	17	16	0	5078	0
12	1	40	2515297827.00	2515297827.00	164.58.112.0	81.22.32.0	0.0.0.0	59	113	4032	80	6	0	16	16	0	5078	0
13	2	80	2515311143.00	2515311368.00	164.58.112.0	81.22.32.0	0.0.0.0	59	113	4046	80	6	0	16	16	0	5078	0
14	1	40	2515301515.00	2515301515.00	164.58.112.0	209.85.224.0	0.0.0.0	59	113	4020	80	6	0	16	16	0	5078	0
15	5	212	2515285652.00	2515336786.00	164.58.112.0	85.17.88.0	0.0.0.0	59	113	1789	80	6	0	16	16	0	5078	0
16	2	104	2515296749.00	2515297156.00	164.58.112.0	69.63.176.0	0.0.0.0	59	91	50001	80	6	0	16	16	0	5078	0
17	1	48	2515295334.00	2515295334.00	164.58.112.0	24.255.168.0	0.0.0.0	59	113	1591	58703	6	0	24	16	0	5078	0
18	1	1500	2515296670.00	2515296670.00	166.66.112.0	65.55.200.0	64.57.28.24	95	91	50849	443	6	0	16	16	18	22192	8075
19	1	103	2515283553.00	2515283553.00	205.213.112.0	64.4.32.0	64.57.28.24	95	91	11497	1863	6	0	24	16	18	2381	8075
20	1	243	2515295236.00	2515295236.00	137.99.112.0	207.46.216.0	64.57.28.24	95	91	43261	80	6	0	24	16	20	13796	8075
21	1	40	2515300115.00	2515300115.00	137.99.112.0	65.54.224.0	64.57.28.24	95	91	43781	443	6	0	16	16	19	13796	8075
22	1	48	2515292246.00	2515292246.00	137.99.112.0	65.54.224.0	64.57.28.24	95	91	43000	443	6	0	2	16	19	13796	8075
23	3	354	2515278448.00	2515312165.00	158.136.112.0	65.55.40.0	64.57.28.24	95	91	3074	3074	17	0	0	16	23	27405	23468
24	4	4540	2515320624.00	2515323255.00	131.94.128.0	65.55.208.0	64.57.28.24	95	91	80	16514	6	0	16	16	18	3681	8075
25	1	1224	2515322831.00	2515322831.00	131.94.128.0	65.55.208.0	64.57.28.24	95	91	80	16514	6	0	24	16	18	3681	8075
26	2	3000	2515325418.00	2515327463.00	131.94.128.0	65.55.208.0	64.57.28.24	95	91	80	17313	6	0	16	16	18	3681	8075
27	2	3000	2515278214.00	2515278610.00	131.94.128.0	65.55.208.0	64.57.28.24	95	91	80	3605	6	0	16	16	18	3681	8075
28	1	1500	2515280333.00	2515280333.00	131.94.128.0	65.55.208.0	64.57.28.24	95	91	80	3914	6	0	16	16	18	3681	8075
29	1	40	2515278845.00	2515278845.00	35.9.128.0	207.46.144.0	64.57.28.24	95	91	1459	80	6	0	16	8	18	237	8070
30	1	52	2515282025.00	2515282025.00	152.65.128.0	65.55.152.0	64.57.28.24	95	91	55476	443	6	0	16	16	18	46780	8075

Table 2.2: Example Netflow Data from Internet2. (Note: Some uninformative columns are omitted in the interest of space.)

Protocol:	% Packets	% Bytes
TCP:	86	92
UDP:	12	6
Other:	2	2

Table 2.3: Percentage of traffic carried by transport protocols. Note that TCP and UDP combine to carry approximately 98% of the total traffic, as measured in terms of either bytes or packets.

that these two protocols together carry approximately 98% of the total traffic, measured both in terms of packets or bytes. Therefore, we are not losing a substantial amount of traffic by ignoring the other protocols.

Packet Size Distribution: One important feature of the data is the difference in packet size distribution between TCP packets and UDP packets. This difference is shown in by the empirical cumulative distribution functions shown in Figure 2.6. TCP packets tend to have most of the mass near two values, corresponding to the maximum packet size of 1500 bytes (the transmitted payload) and the smallest packet size of 40 bytes (ACKs). The UDP traffic, on the other hand, has a much more uniform distribution across packet sizes. The empirical probability mass functions are shown with more detail in Figure 2.7. Although there are several sizes that have noticeably large probability, there are large regions where there exists a non–trivial mass.

Forward/Reverse Flow Relationship: The TCP protocol discussed in Section 2.2 induces a strong relationship between forward and reverse flows. In particular, the small packets on a given flow j are to a large degree the ACK packets of the reverse flow, denoted j^r. This relationship is clearly seen in the data, and illustrated by Figure 2.9. On this pair of forward/reverse flows (Atlanta to Chicago and Chicago to Atlanta), it is clear that the small size packets (ACKs) follow the same pattern as the large (payload) packets on the corresponding reverse flow. As

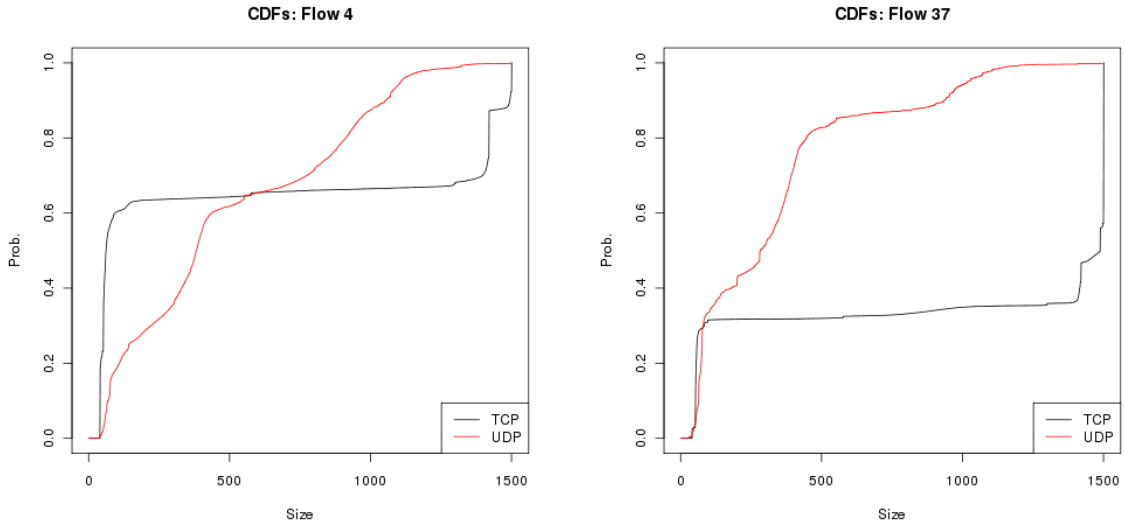


Figure 2.6: Empirical CDF of packet size for TCP and UDP over two flows. Note that the packet size distribution of the TCP subflows is primarily bi–modal, while the UDP subflows have a more spread–out distribution across packet sizes.

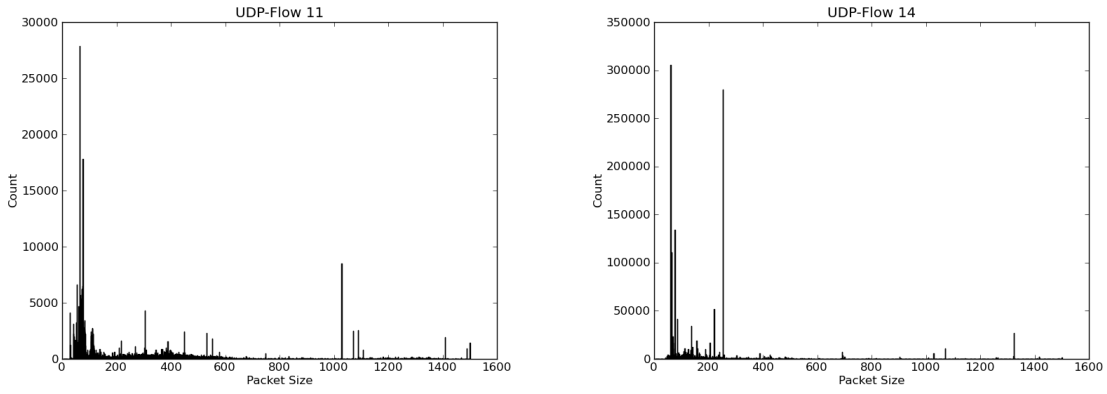


Figure 2.7: Empirical PMF functions for for UDP packets on representative flows (11 and 14). Notice the concentration near 50 bytes, as well as several other spikes at certain sizes.

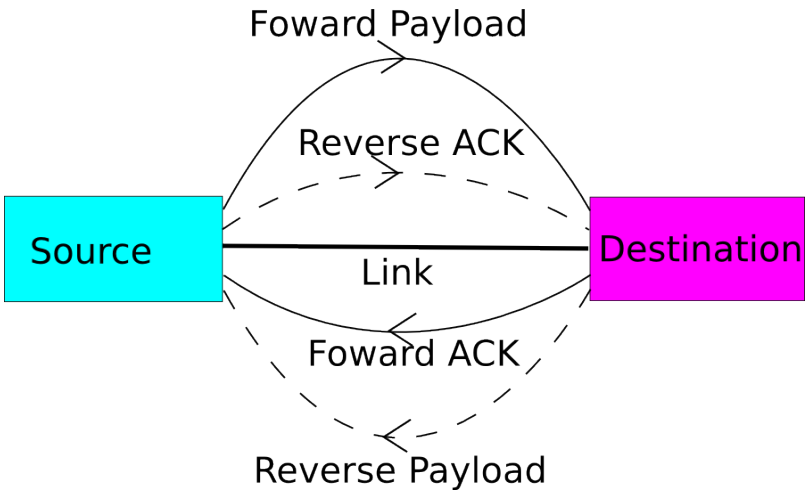


Figure 2.8: The acknowledgment mechanism in TCP traffic. The solid lines show the TCP traffic generated by the Source, with the payload packets traveling from the source to the destination, and the ACK packets being sent back from the destination to the source. The so–called reverse flow sends packets from the Destination to the Source, causing reverse packet to be sent from the Source to the Destination. Thus, the total TCP traffic moving in a given direction consists of the payload data for the forward flow and the acknowledgments for the reverse flow.

suggested by the packet transport mechanism, this pattern is not present in the UDP packets. This is a fundamental mechanism in understanding the statistical behavior of computer network traffic.

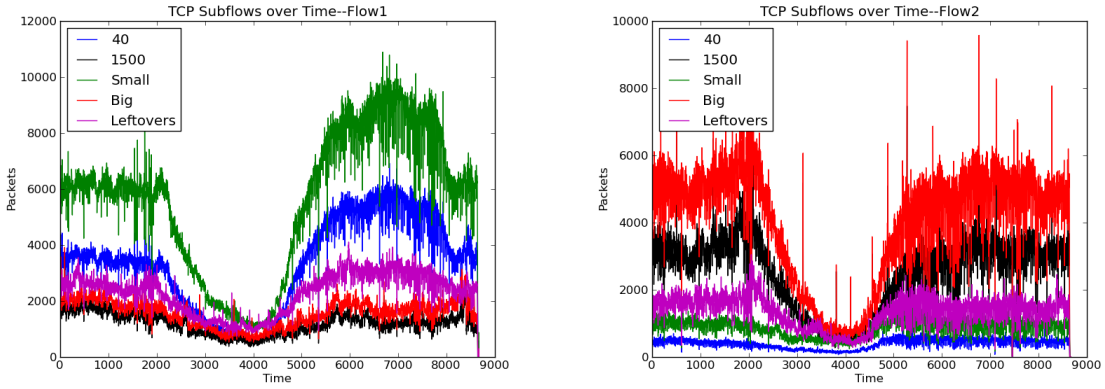


Figure 2.9: Forward/Reverse Relationship for TCP Traffic. These figures depict the TCP subflows for O/D flows 1 and 2. In particular, notice the relationships between the payload subflows (black) with the ACK subflows (blue). The ACK time series tends to follow the same pattern as the payload series on the reverse link.

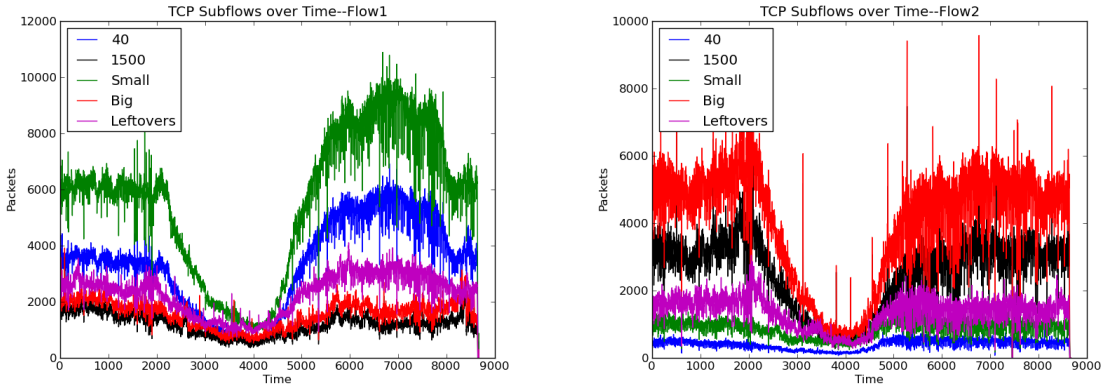


Figure 2.10: Forward/Reverse Relationship for TCP Traffic. These figures depict the TCP subflows for O/D flows . Notice that although the temporal pattern of these flows are different from those displayed in Figure 2.9, the relationship between the subflows remains the same.

CHAPTER III

Global Probabilistic Models for Network Traffic

In this chapter, we first review the existing state–of–the–art probabilistic models of single link/flow traffic. These include the celebrated on/off models leading to the asymptotic approximation of traffic with *fractional Brownian motion*.

We introduce a novel model for the global traffic on the network, which represents the traffic behavior on all links. The model is offered as a scaling limit of on/off sources from all possible origin/destination flows on the network. It faithfully captures the observed temporal behavior of such traffic, but also models the dependence of traffic volumes across different links in the network, which arise from the underlying routing mechanisms of the network.

3.1 Single–Link Models of Computer Network Traffic

In this section we provide basic definitions and properties of Fractional Brownian Motion, Fractional Gaussian Noise, and Long Range Dependence. We discuss the connection between self-similarity and long range dependence, and explore how long range dependence arises in a networking context.

3.1.1 Fractional Brownian Motion and Long Range Dependence

Definition III.1. Fractional Brownian Motion (fBm) is a zero mean Gaussian process $\{X(t)\}_{t\in\mathbb{R}}$ which is self similar and has stationary increments. A process $\{X(t)\}_{t\in\mathbb{R}}$ is considered to have *stationary increments* if

$$\{X(t+h)-X(h)\}_{t\in\mathbb{R}} \stackrel{d}{=} \{X(t)-X(0)\}_{t\in\mathbb{R}}, \forall h \in \mathbb{R}. \tag{3.1}$$

and $\{X(t)\}_{t\in\mathbb{R}}$ is considered *self-similar* with self-similarity exponent $H > 0$ if

$$\{X(ct)\}_{t\in\mathbb{R}} \stackrel{d}{=} \{c^H X(t)\}_{t\in\mathbb{R}}, \forall c > 0. \tag{3.2}$$

Here, $\stackrel{d}{=}$ means equality in finite dimensional distributions (f.d.d.).

Using (3.1) and (3.2) above, one can show that if $\{X(t)\}_{t\in\mathbb{R}}$ is fBm, then

$$\mathrm{Var}(X(t)-X(s)) = \mathrm{Var}(X(t-s)) = \sigma^2|t-s|^{2H}, \text{ where } \sigma^2 = \mathrm{Var}(X(1)).$$

Thus, up to a multiplicative constant σ^2, there is at most one zero-mean Gaussian process that satisfies (3.1) and (3.2) above. It is denoted $B_H(t) = \{B_H(t)\}_{t\in\mathbb{R}}$, and its covariance is

(3.3)
$$\mathrm{Cov}(B_H(t), B_H(s)) = \mathbb{E}B_H(t)B_H(s) = \frac{\sigma^2}{2}(|t|^{2H}+|s|^{2H}-|t-s|^{2H}), \quad \text{where } \sigma^2 = \mathrm{Var}(B_H(1))$$

One can show such processes exist provided $H \in (0,1]$. If $H = 1/2$, the fBm $\{B_{1/2}(t)\}_{t\in\mathbb{R}}$ has independent increments and is the Brownian Motion. If $\sigma^2 = 1$, it is the Standard Brownian Motion. The case $H = 1$ is degenerate since then $B_H(t) = tB_H(1)$, for all $t \in \mathbb{R}$. If, $H \neq 1/2$, the increments of $\{B_H(t)\}_{t\in\mathbb{R}}$ are dependent. Two

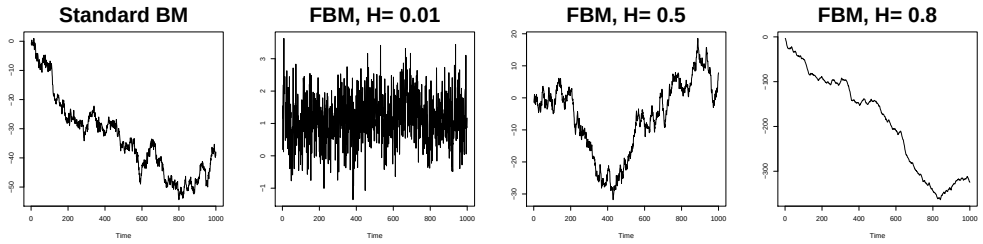


Figure 3.1: Standard Brownian Motion and Fractional Brownian Motion with three different values of H.

distinct regimes are encountered. If $0 < H < 1/2$, the increments of $\{B_H(t)\}_{t\in\mathbb{R}}$ are negatively dependent and the sample paths are rather irregular. However, if $1/2 < H < 1$, then the increments of $\{B_H(t)\}_{t\in\mathbb{R}}$ are positively dependent and the sample paths are smoother. See Figure 3.1 for examples of the behavior of fBm with different values of H.

One can use the Kolmogorov-Čentsov Theorem (e.g. Theorem 2.8 of Karatzas and Shreve [12]) to show that there is a version of $\{B_H(t)\}_{t\in\mathbb{R}}$ with β-Hölder continuous paths for all $\beta \in (0,H)$. This is proved as Proposition 2.4 by Taqqu in [23]. It is not trivial to show that fBms exist. Namely, that the function in (3.3) is positive semi-definite, and therefore it is a valid auto-covariance function. See Proposition 2.2 of Taqqu [23] for a proof.

Alternatively, one can directly construct a process $\{B_H(t)\}_{t\in\mathbb{R}}$ which is H-self similar and has stationary increments by using L^2 stochastic integrals. Namely, let

$$B_H(t) := \int_{\mathbb{R}} \Big((t-s)_+^{H-1/2} - (-s)_+^{H-1/2} \Big) dW(s) \tag{3.4}$$

where $(g(x))_+$ denotes the positive part of $g(x)$, and $\{W(s)\}_{s\in\mathbb{R}}$ represents standard Brownian Motion. It can be shown that the integrand $f_H(t,s) := (t-s)_+^{H-1/2} - (-s)_+^{H-1/2}$ is such that $\int \Big(f_H(t,s)\Big)^2 ds < \infty$ for all $t \in \mathbb{R}$ and $H \in (0,1]$. Thus, the

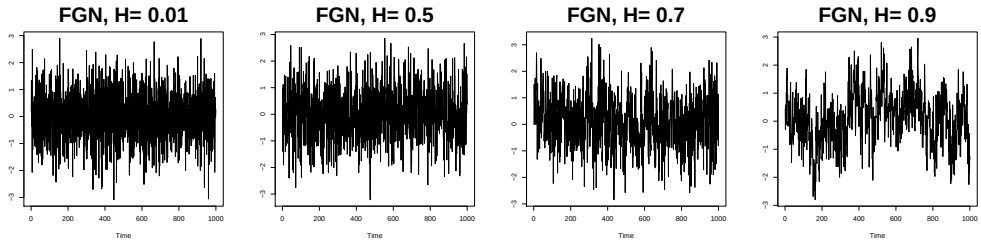


Figure 3.2: Fractional Gaussian Noise with four values of H.

stochastic integral in (3.4) is well defined. One can also show that the resulting zero-mean Gaussian process $B_H(t) = \{B_H(t)\}_{t\in\mathbb{R}}$ has stationary increments and is self-similarity with exponent $H \in (0,1]$. For details, see e.g. Taqqu [23] or Embrechts and Maejima [9].

Definition III.2. Let $\{B_H(t)\}_{t\in\mathbb{R}}$ be fBm. The time series of the increments of $B_H(t)$:

$$Y(k) = B_H(k) - B_H(k-1), \quad k \in \mathbb{Z}, \;\; H \in (0,1]. \tag{3.5}$$

The time series $\{Y(k)\}_{k\in\mathbb{Z}}$ is said to be Fractional Gaussian Noise (FGN). From (3.3), one can show that:

$$\mathrm{Cov}(Y(k+i), Y(i)) = \frac{\sigma^2}{2}\Big(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\Big) = \frac{\sigma^2}{2}\Delta^2(|k|^{2H}) \tag{3.6}$$

where $\Delta^2 = (I - B)^2$ is the second difference operator.

For sample paths of FGN, see Figure 3.2. For further discussion of the properties of FGN, see Taqqu [23].

We now define long range dependence, one of the important properties of FGN.

Definition III.3. A finite-variance stationary time series $Z = \{Z(k)\}_{k\in\mathbb{Z}}$ is said to be Long Range Dependent (LRD) if

$$\sum_{k=1}^{\infty} |\mathrm{Cov}(Z(k), Z(0))| = \infty. \tag{3.7}$$

A Fractional Gaussian Noise time series $\{Y(k)\}_{k\in\mathbb{Z}}$, with $H \in (1/2,1)$ is LRD. Indeed, by using the properties of finite difference and (3.6), one can show that for $H \neq 1/2$,

$$\mathrm{Cov}(Y(k), Y(0)) \sim \sigma^2 H(2H-1)|k|^{2H-2} \text{ as } k \to \infty.$$

Thus, for $1/2 < H < 1$, we have

$$\sum_{k=1}^{\infty} \mathrm{Cov}(Y(k), Y(0)) = \infty \tag{3.8}$$

i.e, the Fractional Gaussian Noise time series is LRD by Definition III.3.

3.1.2 The Connection Between Self Similarity and Long Range Dependence

To illustrate the close connection between LRD and self similarity, suppose that $\{Y(k)\}_{k\in\mathbb{Z}}$ is a zero-mean Gaussian time series which is LRD. In particular, suppose that

$$\gamma(k) = \mathrm{Cov}(Y(k), Y(0)) \sim ck^{2H-2} \text{ as } k \to \infty, \text{ with } H \in (1/2,1). \tag{3.9}$$

Note that $\sum_k |\gamma(k)| = \infty$ Consider the cumulative sum:

$$Y^*(t) = \sum_{k=1}^{[t]} Y(k)$$

We have that

$$\mathrm{Var}(Y^*(Tt)) = [Tt]\sigma_Y^2 + 2\sum_{k=1}^{[Tt]-1} ([Tt]-k)\mathrm{Cov}(Y(k), Y(0)) \tag{3.10}$$

By using (3.9) and the fact that $H \in (1/2,1)$, we have the following scaling limit result.

Proposition III.4. *For all $H \in (1/2,1)$ we have:*

$$\left\{\frac{1}{T^H} Y^*(Tt)\right\}_{t\geq 0} \overset{f.d.d}{\to} \{B_H(t)\}_{t\geq 0} \text{ as } T \to \infty \tag{3.11}$$

where $\{B_H(t)\}_{t\geq 0}$ *is a fractional Brownian motion, as in* (3.3)*, with variance* $\mathrm{Var}(B_H(t)) = \sigma^2 = c/(H-1/2)$*, where* c *is as in* (3.9)*.*

Proof. The process $\left\{T^{-H}Y^*(Tt)\right\}_{t\geq 0}$ is Gaussian and has stationary increments. Therefore, in order to prove (3.11), it is enough to show that

$$\mathrm{Var}\left(\frac{1}{T^H}Y^*(Tt)\right) \to \mathrm{Var}\left(B_H(t)\right) \text{ as } T\to\infty.$$

By (3.10) and (3.9), we have that □

It is important to observe the following with regards to the connection between long range dependence and self similarity.

- LRD time series are asymptotically Self-Similar, in the sense of Proposition III.4.
- The increments of $Y^*(Tt)$ correspond to block-sums of the time series $\{Y(k)\}_{k\in\mathbb{Z}}$. The covariance in (3.11) indicates that

  $$\frac{Y^*(Tk)-Y^*(T(k-1))}{T^H} = \frac{1}{T^H}\sum_{i=[T(k-1)]+1}^{[Tk]} Y(i)$$

  converges to FGN in finite dimensional distributions. Thus, at sufficiently large time scales the LRD series $\{Y(k)\}_{k\in\mathbb{Z}}$ behaves statistically like the FGN. Note that if $\{Y(k) = B_H(k) - B_H(k-1)\}$ itself was an FGN, then by the self-similarity of the fBm for all integer T,

  $$\left\{\frac{1}{T^H}\left(B_H(Tk)-B_H(T(k-1))\right)\right\}_{k\in\mathbb{N}} \equiv \left\{\frac{1}{T^H}\sum_{i=T(k-1)+1}^{Tk} Y(i)\right\}_{k\in\mathbb{N}} \overset{d}{=} \{Y(k)\}_{k\in\mathbb{N}} \ \forall T\in\mathbb{Z}$$

For more details, see [23].

3.1.3 Importance of Long Range Dependence in Networking Context

Initially, work done to analyze network traffic relied on traditional time series methods, often assuming short range dependence. However, much work has been

done to show that many types of high speed Internet traffic exhibits long range dependence. See, for example, [29], [28], and [18].

As an example, consider Figure 3.3. Following the methods in [29], this graph compares actual network traffic and simulated Poisson traffic with the same variance. Here, the traffic time series consists of the average number of packets traversing a link during 1 millisecond intervals. The data was collected from the UNC campus link on April 8, 2002, starting at 9:30 PM. The graph shows 30,000 observations, corresponding to 30 seconds worth of measurements. To simulate Poisson traffic, the variance of these ten thousand measurements was calculated, and then used to generate 30,000 Poisson random variables, having that mean. The mean of each series is then subtracted off, leaving two mean zero series, showing the average traffic over 1 ms intervals. These two series are shown in the the first row of Figure 3.3. In order to be able to compare the right left- and right- hand plots we keep their y axes the same in this and the following plots. Next, block averages of 100 consecutive observations are taken, resulting in series that show the average number of packets over 100 ms intervals. These series are shown in the second row of 3.3. This process is repeated, resulting in series that show the average number of packets over 200 ms and 2 s intervals. Note that the variance of the actual traffic is decreasing noticeably slower than that of the simulated Poisson traffic. Thus, the LRD nature of the measured traffic is apparent.

The On/Off Model of LRD Traffic

Following Willinger, Paxson, Riedi, and Taqqu [28], we now describe the classic On/Off user-level traffic model. In this particular model, we assume that users of a network application switch between periods of activity ("*On* periods") and periods of inactivity ("*Off* periods"). While active (*On*), users are assumed to

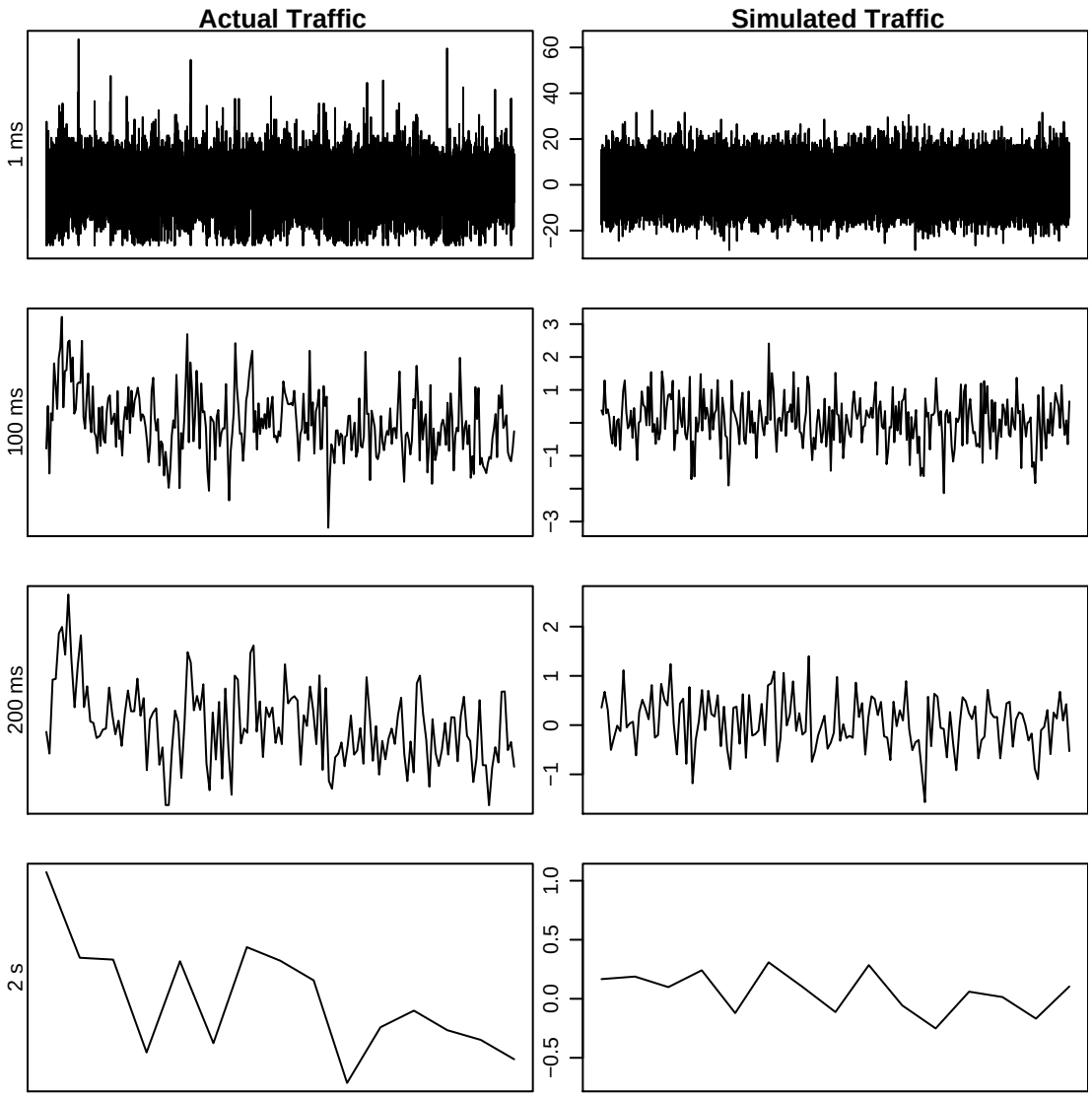


Figure 3.3: Comparison of Actual Traffic from April 8, 2002 at 9:30 PM to simulated Poisson traffic with the same mean at different levels of aggregation.

send packets with a constant rate. Thus, the traffic generated by each user can be represented as a stationary process $\{X(t)\}_{t\geq 0}$, such that :

$$X(t) = \begin{cases} 1 & \text{if the user is ON at time } t \\ 0 & \text{if the user is OFF time } t \end{cases}$$

$X(t)$ can be viewed as the reward at time t, and the process $\{X(t)\}_{t\geq 0}$ is an alternating renewal reward process. Assume that the *On* periods are i.i.d. with cumulative distribution function $F_{on}(x)$, and that the *Off* periods are i.i.d. with cdf F_{off}. Let $\bar{F}(x) = 1 - F(x)$. Furthermore, assume that the lengths of *On* and *Off* periods are independent, and an *Off* period always follows an *On* period. Let σ_{on}^2 and σ_{off}^2 denote the variances of these processes.

In the model, the distribution of the *On* and *Off* times may be either finite variance, or heavy-tailed (finite mean but infinite variance). That is, as $x \to \infty$, either $\bar{F}_{on}(x) \sim c_{on}x^{-\alpha_{on}}$ with $1 < \alpha_{on} < 2$,or $\sigma_{on}^2 < \infty$. Similarly, assume that $\bar{F}_{off}(x) \sim c_{off}x^{-\alpha_{off}}$ as $x \to \infty$ with $1 < \alpha_{off} < 2$,or $\sigma_{off}^2 < \infty$. Here, c_{on} and c_{off} are constants. In the case that $1 < \alpha < 2$, the corresponding *On* or *Off* times have heavy-tailed distributions, with finite mean but infinite variance. (The Pareto distribution would be such an example.) In this case, the corresponding on or off times could be long with high probability. The case $\sigma^2 < \infty$ represents the situations where the on or off times are not likely to last long. Here the distribution might be exponential or Poisson.

The model then considers M independent and identically distributed sources using the same network flow. The aggregate load at time t is then given by $\sum_{k=1}^{M} X_k(t)$. If this load is properly normalized, then by the Central Limit Theorem,

$$\mathcal{L}\lim_{M\to\infty}\frac{1}{\sqrt{M}}\sum_{k=1}^{M}(X_k(t)-\mathbb{E}(X_k(t)))=G(t),t\geq 0. \tag{3.12}$$

where "$\mathcal{L}\lim$" denotes the limit in finite dimensional distributions, and the limit $\{G(t)\}_{t\geq 0}$ is a mean zero stationary Gaussian process.

If one now aggregates over time, then

$$\mathcal{L}\lim_{T\to\infty}\frac{1}{T^H}\int_0^{Tt}G(u)du=\sigma B_H(t),t\geq 0. \tag{3.13}$$

with self-similar parameter

$$H=\frac{3-\min(\alpha_{on},\alpha_{off})}{2} \tag{3.14}$$

where "$\mathcal{L}\lim$" denotes the limit in finite dimensional distributions, and the limit $\{B_H(t)\}_{t\in\mathbb{R}}$ is Fractional Brownian Motion. This result is proved by Taqqu, Willinger, and Sherman as Theorem 1 in [25].

These results show us that, under this model, if we allow the number of users sending traffic over a given network flow first increase to infinity, and then look at the resulting load over increasingly larger time scales, the aggregated total load, centered and rescaled, converges to Fractional Brownian motion in the limit. When $\alpha_{on},\alpha_{off}\in(1,2)$, then $H\in(1/2,1)$, and the fBm $B_H(t)$ has LRD increments. This is in line with the empirical findings that network traffic time series are LRD. Relation (3.14) indicates that the heavier the tails at the on/off periods, the stronger the dependence in the increments of $B_H(t)$.

Let $X^*(T,M)=\int_0^T\sum_{k=1}^{M}\Big(X_k(t)-\mathbb{E}X_k(t)\Big)dt$ denote the cumulative traffic fluctuations over the period $[0,T]$ over a link shared by M independent users. Here, $X^*(T,M)$ corresponds to the cumulative traffic fluctuation, since we subtract the

mean. The limit result in (3.13) is an asymptotic statement about traffic fluctuations. Our focus here is on the study of the statistical properties of the fluctuations of the traffic about its mean.The above result suggests that

$$\frac{1}{T^H\sqrt{M}}X^*(Tt,M) \overset{d}{\approx} B_H(t), \tag{3.15}$$

where " $\overset{d}{\approx}$" means approximate in finite dimensional distributions, and $B_H(t)$ is fBm, as usual. Therefore, if a link is shared by many users, the cumulative fluctuations about the average traffic rate are well modeled by fBm. Thus, provided that one focuses on sufficiently large time scales, fBm is a reasonably good model. In practice, one often studies the stationary time series

$$X(k,\Delta) = X^*(\Delta k, M) - X^*(\Delta(k-1), M) \tag{3.16}$$

representing the traffic fluctuations per time interval Δ. Thus the limit theory suggests that $\{X_\Delta(k)\}_{k\in\mathbb{Z}}$ is well-modeled by FGN for sufficiently large time scales Δ.

One can let T and M grow to infinity simultaneously. Let $M(T)$ be an integer-valued function of T that is non-decreasing in T such that $M(T) \to \infty$ as $T \to \infty$, and $\alpha = \min(\alpha_{on}, \alpha_{off})$. Provided that

$$\lim_{T\to\infty} M(T)/T^{\alpha-1} = \infty,$$

the LHS of (3.15) still converges to fBm. This case is referred to as the "fast growth regime" in the literature, and will be the focus of much of the following work. There is another case, called the "slow growth regime", in which

$$\lim_{T\to\infty} M(T)/T^{\alpha-1} = 0$$

as T and $M(T)$ grow simultaneously to infinity. In this case, the LHS of (3.15) is

no longer fBm, but is rather a stable Lèvy motion. These results are discussed in Mikosch, Resnick, Rootzén, and Stegeman [15] or in Willinger et al [28].

The On/Off model as described above, and the popular M/G/∞ model have been extensively tested and validated against TCP connection level data on links in networks with various topologies. For more details, see [18] and the related theorems.

3.2 A Spatio–Temporal Model for Network–Wide Traffic

In this section, we use the results concerning the traffic fluctuations on a single link of a network to motivate a similar network-wide model. Unlike previous network level models, this is a physical model on the user level similar in spirit to the single-link models, but extends to the network as a whole. To describe the model, we introduce Functional Fractional Brownian Motion, an abstract probabilistic framework to succinctly represent the model, and discuss some of its properties.

3.2.1 Motivation

Let $X_j(t)$ denote the traffic intensity at time t over route j. Thus, $X_j^*(T) := \int_0^T X_j(t)dt$ represents the cumulative traffic over route j during the time interval $[0,T]$. Although it is natural to view traffic in continuous time, in practice the measurements are taken in discrete time. That is, we observe

$$X_j(k;\Delta) := X_j^*(\Delta k) - X_j^*(\Delta(k-1)) = \int_{\Delta(k-1)}^{\Delta k} X_j(t)dt, \text{ for } k = 1,2,\ldots$$

where $\Delta > 0$ is a particular measurement time scale (e.g. 1 ms, 1 second, etc.). Let now $Y_\ell(t)$ denote the traffic intensity observed on link ℓ at time t. As above, we set

$$Y_\ell^*(t) := \int_0^T Y_\ell(t)dt \tag{3.17}$$

and denote by

$$Y_\ell(k;\Delta) := Y^*_\ell(\Delta k) - Y^*_\ell(\Delta(k-1)) = \int_{\Delta(k-1)}^{\Delta k} Y_\ell(t)dt, \text{ for } k = 1, 2, \ldots \tag{3.18}$$

the time series of link measurements over time scale Δ. As mentioned earlier, the routing matrix provides a natural connection between the typically unobserved traffic over all possible routes $X_j(t), j = 1, \ldots, \mathcal{J}$ and the physical traffic observed on all possible links $Y_\ell(t), \ell = 1, \ldots, L$. From the routing equation (2.2), we have:

$$Y(t) = AX(t), \quad Y^*(t) = AX^*(t), \text{ and } \quad Y(k;\Delta) = AX(k;\Delta). \tag{3.19}$$

for all $t \geq 0$, $k \in \mathbb{N}$, $\Delta > 0$, where $Y(t) = \left(Y_\ell(t)\right)_{1\leq\ell\leq L}$, $X(t) = \left(X_j(t)\right)_{1\leq j\leq\mathcal{J}}$ and similarly $Y^*(t)$, $X^*(t)$, $Y(k;\Delta)$, and $X(k;\Delta$ represent vector quantities for the traffic characteristics measured over all links and all routes, respectively. This representation is valid under the assumption that traffic propagates through the network almost instantaneously. This is not an unreasonable assumption when one focuses on modeling time scales of order greater than the round trip time (RTT) of the network. For example, the round trip time on the Internet2 backbone are low (on the order of microseconds) because of the large bandwidth. [3] On the other hand, the relevant time scales for detecting anomalies and attacks are typically greater than the RTT.

Many statistical models for the traffic on a single link or route are available, such as the previously mentioned On/Off models and the $M/G/\infty$ models. The omnipresent phenomenon of the temporal dependence of network traffic both in these models and in real data is that of Long Range Dependence. Little is known, however, about the interplay between *space* and *time*, that is, about the statistical dependence of traffic on different links seen at the same or different instants of

time. Our goal is to develop a statistical model for the network traffic $(Y_\ell(t))_{\ell=1}^{L}$ that can capture the spatial and temporal dependence structure simultaneously.

A Physical User Level Model

Currently, we make the following assumptions to construct the statistical model:

- $X_j(t)$, $j = 1,\ldots,\mathcal{J}$ are mutually independent. That is, the traffics over different routes are independent even if these routes share some links.
- $X_j(t) = \sum_{i=1}^{M_j} X_{ij}(t)$, the traffic on route j is the sum of the traffic generated by M_j independent users sharing this route.
- The individual user traffic intensity is modeled by an on/off renewal reward process

$$X_{ij}(t) = \begin{cases} 1 & \text{if the } i^{th} \text{ user on route } j \text{ is transmitting at time } t \\ 0 & \text{otherwise} \end{cases}$$

- As in the single-link model, we assume that the on/off periods are independent and that they have heavy tails (infinite variance) but finite means. See the discussion and references in Section 2.3.

By Theorem 1 in [25] (see also (3.12) and (3.13) above), we have the following limit result for the cumulative traffic fluctuation over route j. Namely,

$$\mathcal{L}\lim_{T\to\infty}\frac{1}{T^H}\int_0^{T\tau}\left(\mathcal{L}\lim_{M_j\to\infty}\frac{1}{\sqrt{M_j}}\sum_{i=1}^{M_j}(X_{ij}(t)-\mathbb{E}X_{ij}(t))\right)dt = \{B_H(\tau)\}_{\tau\geq 0} \tag{3.20}$$

or equivalently,

$$\mathcal{L}\lim_{T\to\infty}\mathcal{L}\lim_{M_j\to\infty}\frac{1}{T^H\sqrt{M_j}}\int_0^{T\tau}\Big(X_j(t)-\mathbb{E}X_j(t)\Big)dt = \{B_H(\tau)\}_{\tau\geq 0} \tag{3.21}$$

Thus, over a single route, we have for large T,

$$\frac{1}{T^H}X_j^*(Tt) \overset{d}{\approx} B_H(t) \tag{3.22}$$

or, in terms of the increments over time scale Δ,

$$\left\{\frac{1}{\Delta^H}X_j(k,\Delta)\right\}_{k\mathbb{N}} \overset{d}{\approx} \{B_H(k)-B_H(k-1)\}_{k\in\mathbb{N}} \tag{3.23}$$

for large Δ. Relation (3.21) and the continuous mapping theorem imply the following result.

Theorem III.5. *Assume that all routes are of the same type, or $X_j(t)$ are i.i.d processes in j, and that all routes are balanced, that is, that $M_j = M + o(M)$. Then*

$$\mathcal{L}\lim_{T\to\infty}\mathcal{L}\lim_{M\to\infty}\frac{1}{T^H\sqrt{M}}\Big(Y_\ell^*(Tt)-\mathbb{E}Y_\ell^*(Tt)\Big) = A\Big(B_{H,j}(t)\Big)$$

where $B_{H,j} = \{B_{H,j}(t)\}_{t\geq 0}$, for $j = 1,\ldots,\mathcal{J}$ are independent fBm's with $H = (3-\min(\alpha_{on},\alpha_{off}))/2$.

For more general results, when $M_j = M_j(T)$ may grow at different rates for different routes, see D'Auria and Samorodnitsky [8]. Although their setting is different, the results apply in this case.

In attempting to model the behavior of the networks, a practical issue arises. Although the processes take place in continuous time, measurements are necessarily made in discrete time. Thus, in working with real and simulated data, it becomes necessary to work with these measured time series. It is important that the model reflect this. Thus, we define a discretization of the path level processes. First, define cumulative traffic for a given origin–destination pair as

$$X_j^*(\Delta k) = \int_0^{\Delta k} X_j(t)dt \quad \text{for } k\in\mathbb{Z}^+$$

It is then possible to define the origin–destination flow traffic discretized over an interval as

$$X(k) = X_j^*(\Delta k) - X_j^*(\Delta(k-1)) \quad \text{for } k\in\mathbb{Z}^+$$

Then let

$$Y^*_{\ell_j}(k) = \sum_i a_{ij} X^*_j(k).$$

$Y^*(k)$ then represents the link-level volume over a period of time Δ, which is the measurement that are made.

3.2.2 Functional Fractional Brownian Motion

We now describe an abstract probabilistic framework which provides a succinct representation of the resulting traffic model described above. Consider a measure μ defined on an abstract measurable space $(E, \mathcal{E})$. Let

$$L^{2H}(\mu) := \left\{ f : \int_E |f(x)|^{2H} d\mu(x) < \infty \right\},$$

$H \in (0,1)$ and consider the functional

$$\phi(f,g) \equiv \frac{\sigma^2}{2}(\|f\|^{2H}_{L^{2H}} + \|g\|^{2H}_{L^{2H}} - \|f-g\|^{2H}_{L^{2H}}), \tag{3.24}$$

where $f, g \in L^{2H}(\mu)$ and $0 < H < 1$. The functional $\phi(f,g)$ resembles the autocovariance of the ordinary fBm, although f and g are now functions rather than scalars. In fact, as its counterpart in (3.3), the functional $\phi(f,g)$ as defined above is positive semi-definite (Lemma III.11). This fact allows us to define a Gaussian process with covariance given by (3.24) indexed by functions.

Definition III.6. A zero mean Gaussian process $B = \{B(f)\}_{f \in L^{2H}(\mu)}$, indexed by the functions $f \in L^{2H}(\mu)$ is said to be a *Functional Fractional Brownian Motion (f-fBm)* if it has the covariance given by

$$\mathbb{E}B(f)B(g) = \mathrm{Cov}(B(f), B(g)) = \phi(f,g)$$

The following result shows some basic properties of the f-fBm.

Proposition III.7. *Let $B = \{B(f)\}_{f\in L^{2H}(\mu)}$ be a f-fBm with the measure μ and exponent H. Then, we have:*

(i) For a fixed $f \in L^{2H}(\mu)$, the process $\{B(tf\}_{t\geq 0}$ is a fractional Brownian motion with self-similarity parameter H.

(ii) f-fBm has stationary increments; i.e for all $h \in L^{2H}(\mu)$, we have

$$\{B(f+h) - B(h)\}_{f\in\mathcal{F}} \stackrel{d}{=} \{B(f)\}_{f\in\mathcal{F}} \tag{3.25}$$

(iii) $B(f)$ and $B(g)$ are independent if $f \cdot g = 0$.

Proof. Statement *(i)* follows from (3.24) by noting that

$$\mathrm{Cov}(B(tf), B(sf)) = \frac{\sigma^2\|f\|^{2H}_{L^{2H}}}{2}\Big(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\Big),$$

which is the autocovariance of an fBm.

To prove *(ii)*, it is enough to show that

$$\mathrm{Cov}(B(f+h) - B(h), B(g+h) - B(h)) = \mathrm{Cov}(B(f), B(g)),$$

for all f, g and h in $L^{2H}(\mu)$. In view of (3.24), we have that $\mathrm{Cov}(B(f+h)-B(h), B(g+h) - B(h))$ equals:

$$\begin{aligned}
&\mathrm{Cov}(B(f+h), B(g+h)) + \mathrm{Cov}(B(h), B(h)) - \mathrm{Cov}(B(f+h), B(h)) - \mathrm{Cov}(B(g+h), B(h)) \\
&\quad = \frac{\sigma^2}{2}\Big(\|f+h\|^{2H}_{L^{2H}} + \|g+h\|^{2H}_{L^{2H}} - \|f+h-g-h\|^{2H}_{L^{2H}} + 2\|h\|^{2H}_{L^{2H}} - \|f+h\|^{2H}_{L^{2H}} \\
&\qquad -\|h\|^{2H}_{L^{2H}} + \|f+h-h\|^{2H}_{L^{2H}} - \|g\|^{2H}_{L^{2H}} + \|h\|^{2H}_{L^{2H}} - \|g+h-h\|^{2H}_{L^{2H}}\Big) \\
&\quad = \frac{\sigma^2}{2}(\|f\|^{2H}_{L^{2H}} + \|g\|^{2H}_{L^{2H}} - \|f-g\|^{2H}_{L^{2H}}) \\
&\quad = \mathrm{Cov}(B(f), B(g)).
\end{aligned}$$

(iii): Let $f, g \in L^{2H}(\mu)$. If $f \cdot g = 0$, then

$$\|f-g\|^{2H}_{L^{2H}} = \|f\|^{2H}_{L^{2H}} + \|g\|^{2H}_{L^{2H}}.$$

This, in view of (3.24), yields $\mathrm{Cov}(B(f), B(g)) = 0$, which implies the independence of $B(f)$ and $B(g)$ by multivariate normality. This completes the proof. □

The following result shows that the functional fractional Brownian motion has an integral representation similar to that of the ordinary fBm.

Proposition III.8. *Let $B = \{B(f)\}_{f \in L^{2H}(\mu)}$ be a f–fBm associated with the measure space $(E, \mathcal{E}, \mu)$ and with self–similarity parameter $H \in (0,1]$ as in Definition III.6. Then, we have that*

$$B(f) = \iint_{\mathbb{R}\times E} \left\{ (f(u) - s)_+^{H-1/2} - (-s)_+^{H-1/2} \right\} W(du, ds), \tag{3.26}$$

where $W(du, ds)$ is a Gaussian random measure with control measure $\mu(du)ds$, that is $\mathbb{E}W(du, ds)^2 = \mu(du)ds$, and where the coefficient σ^2 in (3.24) is given as:

$$\sigma^2 = \int_{\mathbb{R}} \left((1 - s)_+^{H-1/2} - (-s)_+^{H-1/2} \right)^2 ds. \tag{3.27}$$

Proof. As for the case of ordinary fBm, by using the Fubini's theorem, one can show that the integrand in (3.26) belongs to $L^2(dsd\mu)$ for all $f \in L^{2H}(\mu)$. Thus, $B = \{B(f)\}_{f \in L^{2H}(\mu)}$ is a well–defined zero mean Gaussian process.

By using the integral representation in (3.26) one can show that B has stationary increments. Thus, to prove that B has the auto–covariance of an f-fBm in (3.24), it is enough to show that

$$\mathrm{Var}(B(f)) = \sigma^2 \|f\|^{2H}_{L^{2H}(\mu)}, \quad \text{for all } f \in L^{2H}(\mu). \tag{3.28}$$

Indeed, this follows from the fact that

$$\mathrm{Cov}(B(f), B(g)) = \frac{1}{2}\Big(\mathrm{Var}(B(f) + \mathrm{Var}(B(g)) - \mathrm{Var}(B(f - g)) \Big),$$

since $B(f) - B(g) \overset{d}{=} B(f - g)$.

We will now complete the proof by showing (3.28). By using a change of variables and the Fubini's theorem, we obtain:

$$\begin{aligned}\mathrm{Var}((B_H(f)) = \mathbb{E}(B_H(f))^2 &= \int_{\mathbb{R}\times E}\left((f(u)-s)_+^{H-1/2} - (-s)_+^{H-1/2}\right)^2 ds\, du \\ &= \int_E f(u)^{2H}\int_{\mathbb{R}}\left((1-s)_+^{H-1/2} - (-s)_+^{H-1/2}\right)^2 ds\, du \\ &= \int_E \sigma^2 f(u)^{2H} du = \sigma^2\|f\|_{L^{2H}}^{2H},\end{aligned}$$

where σ^2 is as in (3.27). □

An Example: Consider now a discrete space $E = \{1,\ldots,\mathcal{J}\}$ where the elements of the space corresponding to routes. Let μ be the counting measure. Let $f_\ell(u) = \mathbf{1}_{A_\ell}$, where A_ℓ is the set of all routes that use link ℓ for $\ell \in \{1,\ldots,L\}$. Observe that now the routing matrix can be represented as $A = \left(f_\ell(j)\right)_{1\le\ell\le L, 1\le j\le\mathcal{J}}$. That is, the rows of the routing matrix A are now viewed as functions. Consider the f-fBm $B = \{B(f)\}_{f\in L^{2H}(\mu)}$ and let $Z^*(t) = \left(B(tf_\ell)\right)_{1\le\ell\le L}$. Because of the on/off scaling limit theorems introduced in the previous section, and the assumptions set forth at the beginning of this section, when we consider the cumulative traffic over a link, we have the following relationship:

$$y^*(t) \overset{d}{\approx} Z^*(t) \tag{3.29}$$

To be more precise,

Theorem III.9. *Consider $\mathcal{J}$ routes, where each route j has M_j on/off sources. The cumulative traffic on route j is given by*

$$X_j^*(t) = \sum_{i=1}^{M_j} X_{ij}^*(t)$$

as before. If we let $M_j = M$ for all j, we have

$$\mathcal{L}\lim_{T\to\infty}\mathcal{L}\lim_{M\to\infty}\frac{1}{T^H}\frac{1}{\sqrt{M}}\left(Y^*(Tt) - \mu_Y(Tt)\right) = Z^*(t). \tag{3.30}$$

where $Z^*(t) = \left(B(tf_\ell)\right)_{1\le\ell\le L}$, $Y^*(t)$ is as defined in (3.17)*, and $\mu_Y(t) = \mathbb{E}Y^*(t) = t\mathbb{E}Y^*(1)$.*

The above model provides a spatial, temporal,and spatio-temporal correlation structure of the fluctuations of traffic over links by means of the f-fBm found in the limit. The next result gives explicit formulas for these correlations.

Proposition III.10. *Let $Y^*_\ell(t) = B(tf_\ell)$, where $B(\cdot)$ is an f–fBm with self-similarity parameter $H \in (0,1)$. The following three expressions give the temporal, spatial, and spatio-temporal correlation structure of the increment time series $Y_\ell(k;\Delta) = Y^*_\ell(k\Delta) - Y^*_\ell((k-1)\Delta), k \in \mathbb{Z}$.*

(i) ***Spatial Correlations:***

$$\mathrm{Cov}(Y_{\ell_1}(k;\Delta), Y_{\ell_2}(k;\Delta)) = \sigma^2\Delta^{2H}|A_{\ell_1} \cap A_{\ell_2}| \tag{3.31}$$

(ii)***Temporal Correlations:***

$$\mathrm{Cov}(Y_{\ell_1}(k;\Delta), Y_{\ell_1}(j;\Delta)) = \frac{\sigma^2|\Delta|}{2}\gamma_H(k-j)|A_{\ell_1}| \tag{3.32}$$

(iii)***Spatio-temporal Correlations:***

$$\mathrm{Cov}(Y_{\ell_1}(k;\Delta), Y_{\ell_2}(j;\Delta)) = \frac{\sigma^2|\Delta|^{2H}}{2}\gamma_H(k-j)|A_{\ell_1} \cap A_{\ell_2}| \tag{3.33}$$

where

$$\gamma_H(k) = |k-1|^{2H} + |k+1|^{2H} - 2|k|^{2H}, \quad k \in \mathbb{Z} \tag{3.34}$$

is the autocovariance of a standard fGn with self-similarity parameter $H \in (0,1)$

Proof. The proof of *(ii)* follows from Proposition III.7, part (i). The temporal correlation for the same link at different times is simply the correlation of an fBm. To prove *(iii)*, first note that:

$$\begin{aligned} \mathbb{E}B_H(tf_{\ell_1})B_H(sf_{\ell_2}) &= \frac{\sigma^2}{2}\left(\|tf_{\ell_1}\|^{2H} + \|sf_{\ell_2}\|^{2H} - \|tf_{\ell_1} - sf_{\ell_2}\|^{2H}\right) \\ &= \frac{\sigma^2}{2}\Big(|t|^{2H}|A_{\ell_1}| + |s|^{2H}|A_{\ell_2}| - |t|^{2H}|A_{\ell_1}\setminus A_{\ell_2}| \\ &\qquad -|s|^{2H}|A_{\ell_2}\setminus A_{\ell_1}| - |t-s|^{2H}|A_{\ell_1} \cap A_{\ell_2}|\Big) \end{aligned} \tag{3.35}$$

We then see that:

$$\begin{aligned}
&\mathrm{Cov}(Y_{\ell_1}(k;\Delta), Y_{\ell_2}(j;\Delta)) \\
&\quad = \mathrm{Cov}(Y^*_{\ell_1}(\Delta k) - Y^*_{\ell_1}(\Delta(k-1)), Y^*_{\ell_2}(\Delta j) - Y^*_{\ell_2}(\Delta(j-1))) \\
&\quad = \mathrm{Cov}(Y^*_{\ell_1}(\Delta k), Y^*_{\ell_2}(\Delta j)) - \mathrm{Cov}(Y^*_{\ell_1}(\Delta(k-1)), Y^*_{\ell_2}(\Delta j)) \\
&\qquad -\mathrm{Cov}(Y^*_{\ell_1}(\Delta k), Y^*_{\ell_2}(\Delta(j-1))) + \mathrm{Cov}(Y^*_{\ell_1}(\Delta(k-1)), Y^*_{\ell_2}(\Delta(j-1)))
\end{aligned}$$

Now, by recalling that $Y^*_\ell(\delta k) = B_H(\delta k f_\ell)$, in view of (3.35), we further have

$$\begin{aligned}
&\mathrm{Cov}(Y_{\ell_1}(k;\Delta), Y_{\ell_2}(j;\Delta)) \\
&= \frac{\sigma^2}{2}\Big[|\Delta k|^{2H}|A_{\ell_1}| + |\Delta j|^{2H}|A_{\ell_2}| - |\Delta k|^{2H}|A_{\ell_1}\setminus A_{\ell_2}| - |\Delta j|^{2H}|A_{\ell_2}\setminus A_{\ell_1}| \\
&\quad -|\Delta(k-j)|^{2H}|A_{\ell_1}\cap A_{\ell_2}| - |\Delta(k-1)|^{2H}|A_{\ell_1}| \\
&\quad -|\Delta j|^{2H}|A_{\ell_2}| + |\Delta(k-1)|^{2H}|A_{\ell_1}\setminus A_{\ell_2}| + |\Delta j|^{2H}|A_{\ell_2}\setminus A_{\ell_1}| \\
&\quad +|\Delta(k-1-j)|^{2H}|A_{\ell_1}\cap A_{\ell_2}| - |\Delta k|^{2H}|A_{\ell_1}| - |\Delta(j-1)|^{2H}|A_{\ell_2}| \\
&\quad +|\Delta k|^{2H}|A_{\ell_1}\setminus A_{\ell_2}| + |\Delta(j-1)|^{2H}|A_{\ell_2}\setminus A_{\ell_1}| + |\Delta(k-j+1)|^{2H}|A_{\ell_1}\cap A_{\ell_2}| \\
&\quad +|\Delta(k-1)|^{2H}|A_{\ell_1}| + |\Delta(j-1)|^{2H}|A_{\ell_2}| \\
&\quad -|\Delta(k-1)|^{2H}|A_{\ell_1}\setminus A_{\ell_2}| - |\Delta(j-1)|^{2H}|A_{\ell_2}\setminus A_{\ell_1}| - |\Delta(k-j)|^{2H}|A_{\ell_1}\cap A_{\ell_2}|\Big] \\
&= \frac{\sigma^2|\Delta|^{2H}}{2}\Big[|k-j-1|^{2H} + |k-j+1|^{2H} - 2|k-j|^{2H}\Big]|A_{\ell_1}\cap A_{\ell_2}| \\
&= \frac{\sigma^2|\Delta|^{2H}}{2}\gamma_H(k-j)|A_{\ell_1}\cap A_{\ell_2}|
\end{aligned}$$

The proof of *(i)* is a special case of *(iii)* where $k = j$. □

The intuition behind this formula is that the spatial-temporal covariance between the two links depends on the time lag and the number of flows that share the two links. As one would expect, larger time lags should reduce the correlation,

and increased number of shared paths should increase the correlation. Also, we observe a two-factor structure in the spatio–temporal covariances in (3.33). The correlations have a factor that corresponds exactly as those of a fGn and a spatial factor that is determined by the routing matrix.

Lemma III.11. *Let $\alpha > 0$ and for any $f, g \in L^\alpha(\mu)$, define*

$$\phi_\alpha(f,g) := \frac{1}{2}\Big(\|f\|_\alpha^\alpha + \|g\|_\alpha^\alpha - \|f-g\|_\alpha^\alpha\Big). \tag{3.36}$$

If $\alpha \in (0,2]$, then the function $\phi_\alpha(\cdot,\cdot)$ is positive definite.

Conversely, if there exist two disjoint sets A_i, $i = 1,2$ with positive finite measures $\mu(A_i) > 0$, then the function $\phi_\alpha(\cdot,\cdot)$ is positive definite only if $\alpha \in (0,2]$.

Proof. We will first show that ϕ_α is positive definite for all $\alpha \in (0,2]$.

Let M_α, $\alpha \in (0,2]$ be an SαS random measure with control measure μ and define

$$\Lambda(f) := \int_E f\, dM_\alpha, \quad \forall f \in L^\alpha(\mu),$$

to be the SαS integral of the deterministic function f. Notice that for all $x_j \in \mathbb{C}$, and $f_j \in L^\alpha(\mu)$, with $1 \le j \le n$, we have

$$\mathbb{E}\Big|\sum_{j=1}^n x_j e^{i\Lambda(f_j)}\Big|^2 = \sum_{j,k=1}^n x_j \overline{x}_k \mathbb{E} e^{i\Lambda(f_j - f_k)} = \sum_{j,k=1}^n x_j \overline{x}_k e^{-\|f_j - f_k\|_\alpha^\alpha}.$$

Since the LHS of the last expression is always non–negative, so is the RHS. This shows that the function $r_\alpha(f,g) := e^{-\|f-g\|_\alpha^\alpha}$, $f, g \in L^\alpha(\mu)$ is positive definite.

Now, the proof proceeds as the proof of the positive definiteness of the auto–covariance function of the fractional Brownian motion (see, e.g. Proposition 2.2 in Taqqu [23]). Indeed, for all $x_j \in \mathbb{C}$, and $f_j \in L^\alpha$, $0 \le j \le n$, and for all $\epsilon > 0$, we

have

$$
\begin{aligned}
(3.37)0 \quad &\leq \sum_{j,k=0}^{n} x_j \overline{x}_k e^{-\epsilon \|f_j - f_k\|_\alpha^\alpha} \\
&= \sum_{j,k=1}^{n} x_j \overline{x}_k e^{-\epsilon \|f_j - f_k\|_\alpha^\alpha} + \sum_{j=1}^{n} x_0 \overline{x}_k e^{-\epsilon \|f_0 - f_k\|_\alpha^\alpha} + \sum_{j=1}^{n} x_j \overline{x}_0 e^{-\epsilon \|f_j - f_0\|_\alpha^\alpha} + x_0 \overline{x}_0 \\
&=: S_1 + S_2 + S_3 + |x_0|^2
\end{aligned}
$$

Since x_0 and f_0 are at our disposal, let $f_0 := 0$ and $x_0 := -\sum_{j=1}^{n} x_j e^{-\epsilon \|f_j\|_\alpha^\alpha}$. Observe that with this choice of x_0 and f_0, we get

$$
S_2 = S_3 = -|x_0|^2 = -\sum_{j,k=1}^{n} x_j \overline{x}_k e^{-\epsilon \|f_j\|_\alpha^\alpha - \epsilon \|f_k\|_\alpha^\alpha},
$$

and therefore

$$
\begin{aligned}
S_1 + S_2 + S_3 + |x_0|^2 &= \sum_{j,k=1}^{n} x_j \overline{x}_k \left(e^{-\epsilon \|f_j - f_k\|_\alpha^\alpha} - e^{-\epsilon \|f_j\|_\alpha^\alpha - \epsilon \|f_k\|_\alpha^\alpha} \right) \\
&= \epsilon \sum_{j,k=1}^{n} x_j \overline{x}_k \left(\|f_j\|_\alpha^\alpha + \|f_k\|_\alpha^\alpha - \|f_j - f_k\|_\alpha^\alpha \right) + o(\epsilon), \qquad (3.38)
\end{aligned}
$$

as $\epsilon \downarrow 0$, where the last relation we used the fact that $e^{-\epsilon a} - e^{-\epsilon b} = \epsilon(b-a) + o(\epsilon)$, as $\epsilon \downarrow 0$. If for some x_j's and f_j's we have $\sum_{j,k=1}^{n} x_j \overline{x}_k (\|f_j\|_\alpha^\alpha + \|f_k\|_\alpha^\alpha - \|f_j - f_k\|_\alpha^\alpha) < 0$, then, for all sufficiently small $\epsilon > 0$, the LHS of (3.38) becomes negative, which in view of (3.37), is impossible. This shows that ϕ_α is positive definite.

Conversely, we will now show that if $\alpha > 2$, then ϕ_α is not positive definite. Indeed, let $f_1 = 1_{A_1}/\mu(A_1)^{1/\alpha}$ and $f_2 = 1_{A_2}/\mu(A_2)^{1/\alpha}$, where $\mu(A_i) > 0$ and $\mu(A_1 \cap A_2) = 0$. Then, one can show that

$$
(\phi_\alpha(f_i, f_j))_{2\times 2} = \begin{pmatrix} 1 & (1 - 2^{\alpha-1}) \\ (1 - 2^{\alpha-1}) & 1 \end{pmatrix}.
$$

Observe that the determinant of the last matrix equals $(1 - (1 - 2^{\alpha-1})^2)$, which is negative for $\alpha > 2$. This shows that the eigen values of the matrix $(\phi_\alpha(f_i, f_j))_{2\times 2}$ have opposite signs and hence the function ϕ_α is not positive definite. □

On/Off Source Simulation

A method for simulation of on/off traffic is outlined here. It is designed to be efficient in that it generates aggregate on/off traffic from several users, but only stores a small amount of information, namely the current state (S_i) and time remaining in that state (R_i) of each user, with S_i and R_i defined below. First, each user is initialized by being randomly assigned a state, $S_i \in \{0, 1\}$ (0 for off and 1 for on), and the time remaining in that state, drawing from the Pareto distribution with an appropriate exponent. For each time step, the time each user is spent in an "on" state is calculated, and these times are totaled. As necessary, the states are updated as the time remaining runs out and new times for the next state are drawn. Figure 3.4 illustrates four possible scenarios. The user represented by line (a) is in an "on" state and remains in that state for the duration of the time step. Thus, the user contributes a activity of length Δ to the aggregate activity for the interval. User (b) on the other had remains inactive for the entire interval and contributes no traffic to the total. User (c) begins the interval in an "on" state. The time remaining in that state is not sufficient to last the entire interval, and so the state is switched, and a new R_i is drawn for the time the user will remain in the off state. Only the time this user spends in the "on" state is added to the total traffic. Finally, user (d) begins the interval in the "off" state. During the course of the interval, the state changes three times, each change resulting in the state, S_i, and the time remaining R_i being replaced each time. Again, the time that user (d) spends in the "on" state is added to the total traffic for that interval. This procedure is used to simulate traffic on any number of independent paths, and then used to generate traffic on links for an network by forward-multiplying the path measurements by the routing matrix.

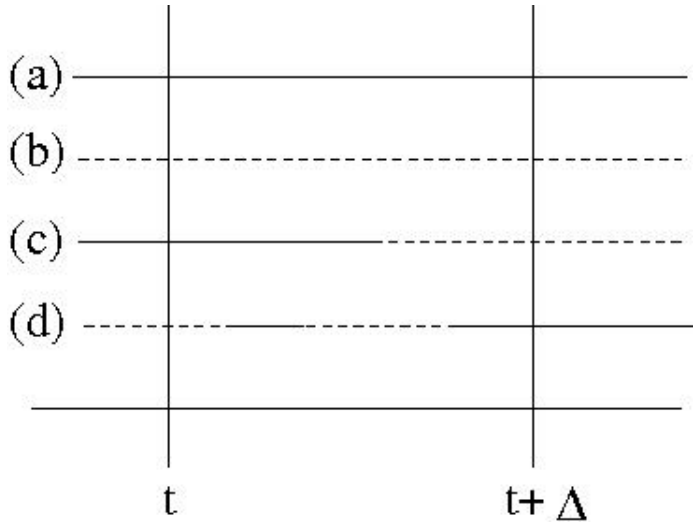


Figure 3.4: Illustration of Simulation Method. The x-axis represents time, while the y-axis represents 4 simulated users. A solid line indicates that that user is currently in an *on* state, while a dashed line indicates that the user is currently in an *off* state.

CHAPTER IV

Network Kriging

In the previous chapter we developed a general probabilistic model for global network traffic. In this chapter, we explore the network kriging problem. One important practical issue will be the fact that the flow means μ_X and variances σ_X^2 are unknown or hard to measure directly. They can, in principle, be estimated from expensive Netflow measurements of the flows, but they are not available in a fast on–line fashion as are direct link measurements. Motivated by this issue, we will refine the theoretical global traffic model and obtain a practical network–specific model by fusing expensive (slow) Netflow and inexpensive (fast) direct link measurements.

4.1 Network Kriging with Known Parameters

We start by discussing a natural application of global network traffic models, namely the network kriging problem. An important motivation behind developing practical global traffic models is to better predict traffic on unobserved network links or detect statistical anomalies, which can help identify structural problems or denial of service attacks, for example.

Network Kriging: *Traffic traces are observed on a subset of links* $\mathcal{O} \subset \{1, \cdots, L\}$*:*

$$\mathcal{D} \equiv \mathcal{D}(t_0, m) := \{Y_\ell(t),\ t_0 - m \le t \le t_0,\ \ell \in \mathcal{O}\}. \tag{4.1}$$

over the time window $t_0 - m + 1 \le t \le t_0$ *of size* m*.*

Obtain estimators $\widehat{Y_\ell}(t_0)$ *for the traffic* $Y_\ell(t_0)$*, for all unobserved links* $\ell \in \mathcal{U} := \{1, \cdots, L\} \setminus \mathcal{O}$ *at time* t_0 *in terms of the data* $\mathcal{D}(t_0, m)$*.*

The term *network kriging* was coined by [4] but used in a different context.

Network Prediction: *Assume that the data* $\mathcal{D}(t_0, m)$ *as in the* Network Kriging *problem is available.*

Obtain estimators $\widehat{Y_\ell}(t_0 + h)$ *for the future traffic loads* $Y_\ell(t_0 + h)$ *for all (observed and unobserved) links* $\ell \in \{1, \cdots, L\}$*, given the data* $\mathcal{D}(t_0, m)$*.*

The network kriging problem is a special case of the network prediction problem ($h = 0$). We state them separately since here we will emphasize on network kriging. A satisfactory practical solution of the general network prediction problem will be the goal of another paper. We review next some theoretical expressions for the optimal predictors. They are valid under the following slightly more general assumptions than given in the global model.

Let $\{Y(t)\}_{t \in \mathbb{Z}},\ Y(t) = (Y_\ell(t))_{\ell=1}^L$ be a stationary multivariate Gaussian time series such that

$$Y(t) \sim \mathcal{N}(\mu_Y, \Sigma_Y), \quad \text{and} \quad \mathrm{Corr}(Y_{\ell_1}(t), Y_{\ell_2}(t+k)) = \rho(k),$$

for all $1 \le \ell_1, \ell_2 \le L$, where $\rho(k),\ k \in \mathbb{Z}$ is an arbitrary positive definite function.

Network Kriging: For known means μ_Y and covariances Σ_Y, the network kriging problem becomes the *simple kriging* problem from spatial statistics. The best linear unbiased predictor (BLUP) for $Y_u = Y_u(t)$ in terms of $Y_o = Y_o(t)$ is then given

by:

$$\widehat{Y}_u(t) = \mu_u + \Sigma_{uo}\Sigma_{oo}^{-1}(Y_o(t) - \mu_o), \tag{4.2}$$

where

$$Y = \begin{pmatrix} Y_o \\ Y_u \end{pmatrix},\ \mu_Y = \begin{pmatrix} \mu_o \\ \mu_u \end{pmatrix},\ \text{and}\ \ \Sigma_Y = \begin{pmatrix} \Sigma_{oo} & \Sigma_{ou} \\ \Sigma_{uo} & \Sigma_{uu} \end{pmatrix}.$$

Here $\mathbb{E}Y(t) = \mu_Y$ and Σ_Y are the mean and the covariance matrix of the vector $Y(t)$, partitioned accordingly into *observed* and *unobserved* components $Y_o(t)$ and $Y_u(t)$. The (conditional) covariance matrix of prediction errors $Y_u(t) - \widehat{Y}_u(t)$ is given by:

$$\text{m.s.e.}(\widehat{Y}_u|\mathcal{D}) \equiv \mathbb{E}\Big((Y_u - \widehat{Y}_u)(Y_u - \widehat{Y}_u)^T|Y_o\Big) = \Sigma_{uu} - \Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou}. \tag{4.3}$$

Network Prediction: Consider first the *h–step-ahead prediction* for directly *observed links* in the past. Proposition 7 in [21] implies that the BLUP $\widehat{Y}_o(t_0+h)$ of $Y_o(t_0+h)$ via $\mathcal{D}(t_0, m)$ is given as follows:

$$\widehat{Y}_o(t_0+h) = \mu_o + \sum_{j=0}^{m-1} c_j(h)(Y_o(t_0-j) - \mu_o), \tag{4.4}$$

where $\vec{c}(h) = (c_j(h))_{j=0}^{m-1} = \Gamma_m^{-1}\vec{\gamma}_m(h)$, with

$$\Gamma_m = (\rho(|i-j|))_{0\le i,j\le m-1} \quad \text{and} \quad \vec{\gamma}_m(h) = (\rho(h+j))_{j=0}^{m-1}. \tag{4.5}$$

(The matrix Γ_m is always invertible, provided $\rho(\tau) \to 0,\ \tau \to \infty$, see e.g. Proposition 5.1.1 in [2].) The prediction error is then

$$\text{m.s.e.}(\widehat{Y}_o(t_0+h)|\mathcal{D}) = \sigma^2(h)\Sigma_{oo}, \quad \text{with} \quad \sigma^2(h) = 1 - \vec{c}(h)^T\Gamma_m\vec{c}(h). \tag{4.6}$$

Now, for the case of *unobserved links* in the past, by Proposition 7 in [21], we have that the BLUP is:

$$\widehat{Y}_u(t_0+h) = \mu_u + \Sigma_{uo}\Sigma_{oo}^{-1}(\widehat{Y}_o(t_0+h) - \mu_o) \tag{4.7}$$

and for the prediction error matrix, we have:

$$\text{m.s.e.}(\widehat{Y}_u(t_0+h)|\mathcal{D}) = \sigma^2(h)\Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou} + \Sigma_{uu} - \Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou}. \tag{4.8}$$

By Gaussianity, the above BLUP's are also the minimum variance unbiased predictors.

Observe that, the predictor $\widehat{Y}_u(t_0+h)$ is obtained as in the case of *simple kriging*, where the future $Y_o(t_0+h)$ in (4.2) are replaced in (4.7) by their optimal predictors $\widehat{Y}_o(t_0+h)$ from (4.4). Consequently, the prediction error matrix in (4.8) is the sum of an error due to predicting the observed links into the future *and* the kriging error. This 'orthogonality phenomenon' is due to the product structure in the spatio–temporal covariances of the time series $\{Y(t)\}_{t\in\mathbb{Z}}$. For more details, see [21].

The performance of the *simple kriging* and the *temporal prediction* is illustrated in Figure 4.1. One link is estimated from *current* and *past* observations of three other links (Scenario 6 in Tables 2.1 & 6.1). Here, μ_Y and Σ_Y are estimated from moving windows of past observations, by using data on all links (described in section 4.2.4). For the temporal correlations $\rho(k)$'s we used the fGn model with Hurst parameter estimated from the data.

In the context of our network kriging and prediction problems, however, not all links are observed, the means and covariances are unknown, and the above predictors in Figure 4.1 are not possible to obtain. Nevertheless, they provide theoretically optimal benchmark (baseline) that will be used to evaluate all methods developed in the sequel.

In the next section, we will develop a model that can be estimated from data on a small subset of links and used to perform kriging and prediction over the entire network. This can be done, however, only by utilizing further information about

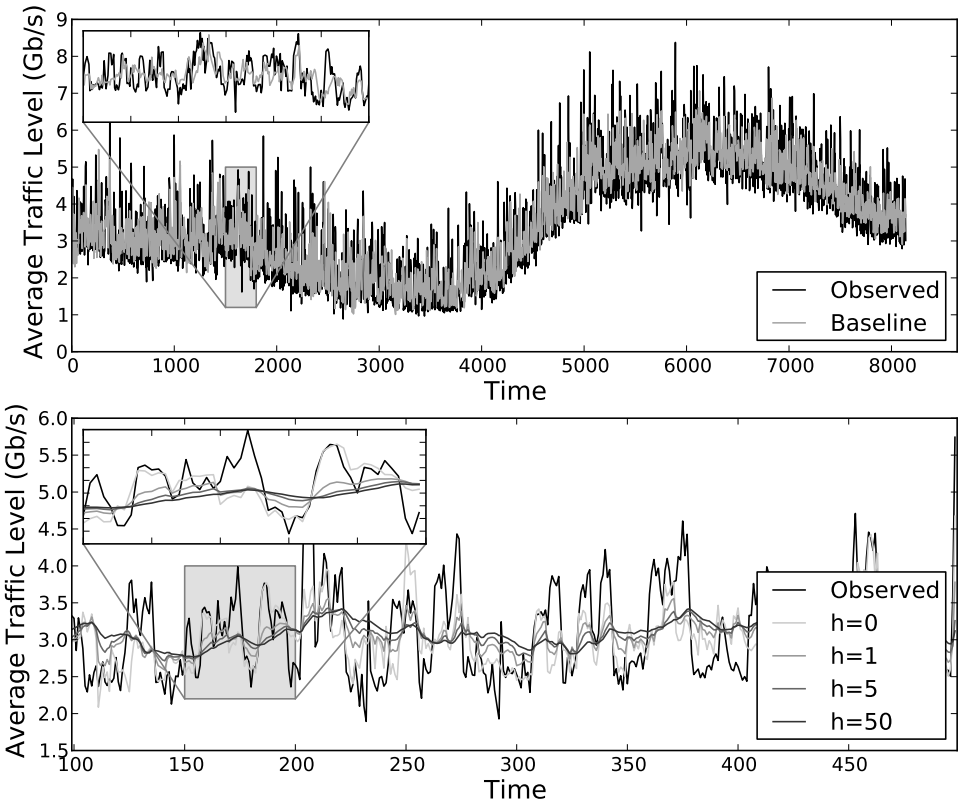


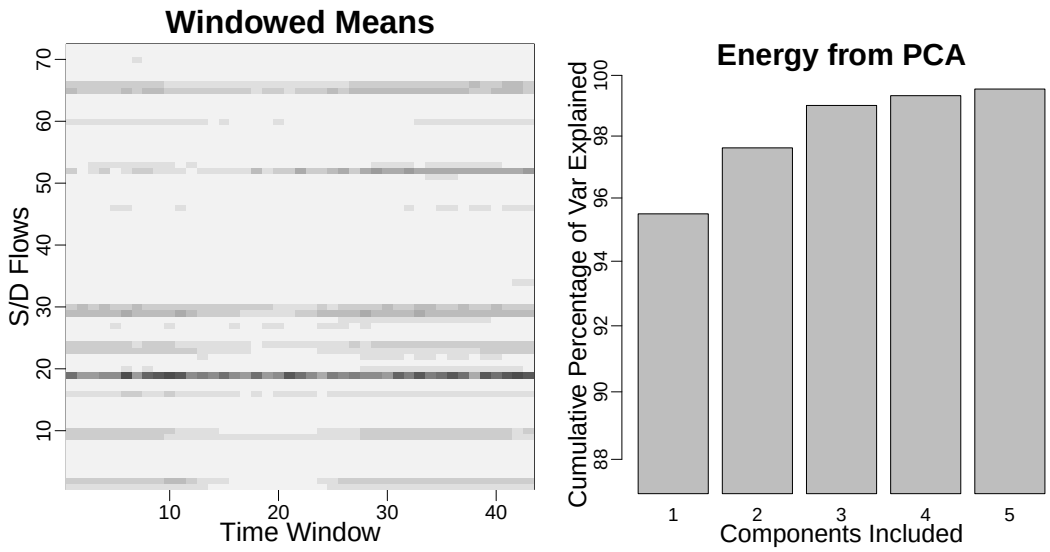
Figure 4.1: Kriging (top) and Prediction (bottom) in Scenario 6 (see Tables 2.1 & 6.1). The top plot shows the link loads over successive 10s windows, beginning at 0h GMT on Feb 19, 2009 and lasting 24 hours. The bottom plot focuses on a period of 500 windows and displays predictors for lags $h = 0, 1, 5$, and 50. Note that $h = 0$ corresponds to *kriging*.

the network.

4.2 Network Specific Modeling via Netflow Data

4.2.1 Modeling Traffic Means

As mentioned in the introduction, direct measurements of the flow–level traffic $X(t)$ are very expensive to obtain because this would involve examining the entire traffic load of the network, i.e. storing and then processing 95–170 Gigabytes of data per day. Modern routers allow for random or systematic sampling of the flow of traversing packets. The routers store important information such as the ports,



(a) Windowed Means (b) Energy

Figure 4.2: *Left:* The columns correspond to local sample means over consecutive windows of 2000 seconds for each of the $\mathcal{J} = 72$ flows. Darker shades indicate higher values. The data were reconstructed from Netflow measurements of the Internet2 network for Feb 19, 2009. *Right:* cumulative energy captured by the F matrix for increasing values of p (see (4.9) and Proposition IV.1 below.)

source and destination IP addresses, etc. from the sampled packet headers. Even though in fast backbone networks (e.g. Internet2) the practical sampling rates are e.g. 1 out of 100 packets, the Netflow mechanism provides unique information about the traffic loads in the network. Using a careful mapping procedure, we assigned the sampled packets to one of the 72 origin–destination flows. We thus constructed an estimate $\{\widetilde{X}(t)\} \approx \{X(t)\}$ of the flow–level traffic. Unfortunately, this method is computationally expensive to implement, which makes it impractical to use repeatedly, and to apply in an on–line fashion. Therefore, the information derived from Netflow can only be viewed as auxiliary data in the context of network kriging. Nevertheless, this information proves useful in building a flexible network-wide model that can be updated online.

Figure 4.2a illustrates the local means of the origin–destination flows as a function of time, where the data were derived from an extensive analysis of Netflow

measurements. This suggests that a linear model for $\mu_X(t)$ with a few constant factors can capture much of the variability in the local means. We therefore posit the model

$$\mu_X(t) = F\beta(t), \tag{4.9}$$

where F is a suitably chosen $\mathcal{J}\times p$ matrix and $\beta = \beta(t) \in \mathbb{R}^p$ is a parameter. Observe that

$$\mu_Y = A\mu_X = AF\beta \quad \text{and also} \quad \mu_{Y_o} = A_oF\beta.$$

Provided p equals $\text{rank}(A_oF)$, the parameter β can be successfully estimated by using linear regression from the available data on the *observed links*. We will see that this essentially means that p is no greater than the number of observed links $|\mathcal{O}|$.

The goal then becomes to choose F *optimally* so that $F\beta(t)$ can approximate best $\mu_X(t)$ (in the Euclidean norm) with a suitable $\beta(t) \in \mathbb{R}^p$. Consider the sample $\widetilde{X}(t)$, $1 \le t \le T$, of the flow–level data derived from the Netflow mapping, where $T = w \times n$. Partition the data into n windows of size w, and let

$$m_X(k) = \frac{1}{w}\sum_{i=1}^{w} \widetilde{X}((k-1)w + i) \tag{4.10}$$

$(1 \le k \le n)$, be the sample mean the $\widetilde{X}(t)$'s in the k–th window.

Consider the set of n vectors $\{m_X(k),\ 1 \le k \le n\}$ in $\mathbb{R}^J$ and observe that the model in (4.9) postulates that $\mu_X(t)$ belongs to range(F) (the linear space spanned by the columns of F). Thus, given the *local means* $m_X(k)$'s, a *least squares optimal* choice of F corresponds to minimizing the sum of the squared distances from the $m_K(k)$'s to the p–dimensional subspace $W := \text{range}(F)$. That is, we want to find

$$W^* = \underset{W\le\mathbb{R}^J,\ \dim(W)=p}{\text{Argmin}} \sum_{k=1}^{n} \|m_X(k) - P_W(m_X(k))\|^2,$$

where P_W denotes the orthogonal projection onto the subspace W. The following result shows that this problem has a simple solution, which corresponds precisely to performing *principal component analysis* (PCA) on a certain matrix.

Proposition IV.1. *Let $m_X(k) \in \mathbb{R}^J$, $1 \le k \le n$. Consider the positive semidefinite $J \times J$ matrix $M = \sum_{k=1}^{n} m_X(k) m_X(k)^T$ and let $M = \sum_{j=1}^{J} \lambda_j b_j b_j^T$, be its spectral decomposition, where b_j, $1 \le j \le J$ are orthonormal and $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_J \ge 0$.*

Set $W^* = \mathrm{span}\{b_1, \ldots, b_p\}$, $1 \le p \le J$. Then, for all $W \le \mathbb{R}^J$ with $\dim(W) = p$, we have that

$$\sum_{k=1}^{n} \|m_X(k) - P_{W^*}(m_X(k))\|^2 \equiv \sum_{j=p+1}^{J} \lambda_j \le \sum_{k=1}^{n} \|m_X(k) - P_W(m_X(k))\|^2, \tag{4.11}$$

where P_W denotes the orthogonal projection onto the subspace W.

The proof is given in Appendix 4.3. This result implies that the $(J \times p)$ matrix F with columns given by the p eigenvectors of M with largest eigenvalues yields the best linear model for the temporal behavior of the local traffic means.

Figure 4.2b shows that just a few PCA vectors p are enough to capture a large percentage of the local variability of the mean vectors $m_X(t)$'s. Figure 4.3 (top) illustrates the prediction performance of this model when the covariance matrix is known, but the *unobserved* means are estimated from the model.

Remark IV.2. Note that we view the local means as slowly changing in time, non–random vector functions with values in $\mathbb{R}^J$. Our model allows us the flexibility to represent the means as locally linear functions and it captures well possible non–stationarity due to time of the day effects. The statistical behavior of the flow–level traffic on finer time scales is of a rather different nature and it can be modeled by stationary long–range dependent time series.

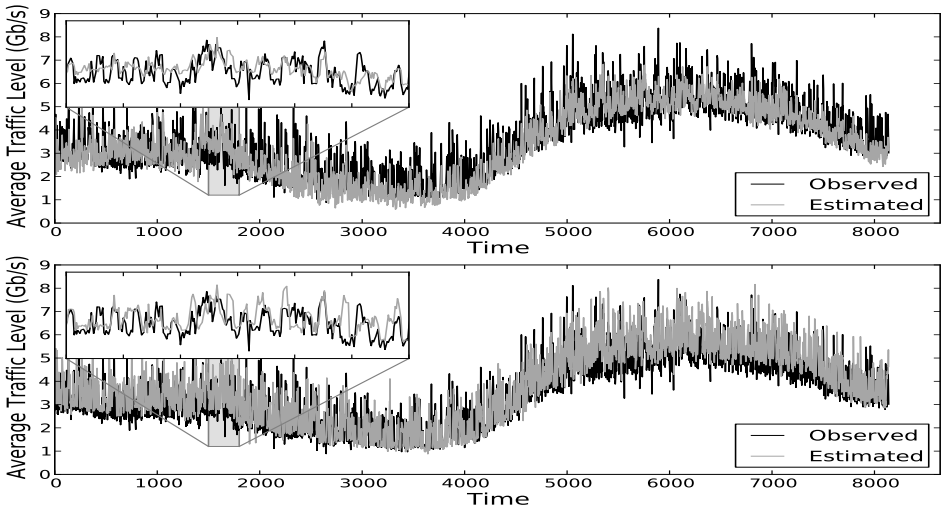


Figure 4.3: *Top:* Kriging in Scenario 6 (Table 6.1) using the PCA–mean model (with $p = 2$) and the *sample covariance* matrix. In reality, the sample covariances for unobserved links are not available, and this plot merely illustrates that the model (4.9) successfully captures the structure of the means. See also Figure 4.1. *Bottom:* Kriging in Scenario 6 using the *complete mean–variance* model with $p = 2$. The time axis shows successive 10s windows, beginning at 0h GMT on Feb 19, 2009 and lasting 24 hours.

Remark IV.3. The method of selecting the mean model $\mu_X = F\beta$ here resembles *principal component* regression, where a lower dimensional model is obtained by focusing on the p eigenvectors of the matrix M.

4.2.2 Modeling the Covariances

The physical nature of network protocols, the mechanisms of transmission, and the user behavior imply strong relationship between the means and the variances of network traffic traces. This relationship was shown to be ubiquitous over different types of computer networks. In the field of network tomography, for example, the mean–variance models have been successfully used to resolve challenging identifiability questions (see [13, 20]). In our context, we also encountered a strong relationship between the means and the variances of *traffic flows*. More precisely, by exploring the sample means $\overline{X}_j(t)$ and standard errors $S_j(t)$, calcu-

lated over a window of traffic data, we observed that

$$S_j(t) \approx C(\overline{X}_j(t))^\gamma, \quad 1 \le j \le \mathcal{J} \tag{4.12}$$

with $\gamma \approx 3/4$. Namely, the standard error of a source–destination flow $X_j(t)$ is proportional to a power of its mean.

We estimated γ (as a function of t) by performing log–linear regression of $S_j(t)$ versus $\overline{X}_j(t)$ *over* j, $1 \le j \le \mathcal{J}$. The resulting estimates remained approximately constant in t and close to 3/4 regardless of the time window used. The power–law relationship is remarkably consistent in time and the regression diagnostics $R^2 \approx 80\%$ indicate strong agreement with the model (see [26]).

Regarding cross-flow correlations, it was shown in [20] that they are rather weak and only *forward* and *reverse* flows, e.g. the Chicago–Los Angeles and Los Angeles–Chicago, exhibit moderate magnitude ones (around 0.3). Such correlations are primarily due to the feedback mechanism built in the TCP protocol. Our experience with Netflow on Internet2 (e.g. Fig. 2 in [21]) and limited NS2–simulations [16] confirm that cross-flow correlations are negligible at the time scales of interest, provided that the network is not congested. The study of heavy traffic scenarios beyond the operating characteristics of the network is interesting but it is outside the scope of the present work. Therefore, in this paper we shall model $\Sigma_X(t)$ as a diagonal matrix of the form:

$$\Sigma_X = \sigma^2 \mathrm{diag}(|F\beta|^{2\gamma}), \tag{4.13}$$

where $|a|^{2\gamma}$ denotes $(|a_i|^{2\gamma})_{i=1}^{\mathcal{J}}$, for $a = (a_i)_{i=1}^{\mathcal{J}} \in \mathbb{R}^{\mathcal{J}}$, and where $\mu_X = F\beta$.

We will show below that the parameter σ can be estimated on–line from link–level data (Y_ℓ's). On the other hand, γ is a structural parameter, obtained from the off–line analysis of Netflow data. (See Section 4.4.2 for more details.)

4.2.3 The Joint Model: Parameter Estimation and Asymptotic Properties

Combining the mean and covariance models from the previous two sections, we obtain the following joint model:

$$Y(t) = AF\beta + \sigma A\mathrm{diag}(|F\beta|^{\gamma})Z(t), \tag{4.14}$$

where $Z(t) \sim \mathcal{N}(0, I_{\mathcal{J}})$ is a standard normal vector in $\mathbb{R}^{\mathcal{J}}$ and where $\beta \in \mathbb{R}^p$ and $\sigma > 0$ are unknown parameters. In this section, we will show how this model can be estimated from *on–line* measurements on a limited set of observed links $\mathcal{O}$. We will also establish asymptotic properties of the proposed estimators.

In the framework of the Network Kriging problem (see Section 4.1), we obtain

$$\overline{Y}_o(t_0) = A_o F\beta + \epsilon_{\overline{Y}_o}(t_0),$$

where

$$\overline{Y}_o(t_0) = \frac{1}{m}\sum_{k=0}^{m-1} Y_o(t_0 - k). \tag{4.15}$$

To establish the covariance structure of the noise $\epsilon_{\overline{Y}_o}$, we introduce the mild assumption that the flow–level traffic is stationary (in practice, traffic is locally stationary on the time scales of interest) and its temporal correlation structure is the same across all routes. Namely, that $\mathrm{Corr}(X_j(t), X_j(t+i)) = \rho(i),\ 1 \le i \le m-1,\ (1 \le j \le \mathcal{J})$. This yields

$$\mathrm{Corr}(Y_\ell(t+i), Y_\ell(t)) = \rho(i),\ 1 \le i \le m-1, \quad \text{for all } 1 \le \ell \le L,$$

and consequently

$$\epsilon_{\overline{Y}_o}(t_0) \sim \mathcal{N}(0, \sigma_m^2 A_o \mathrm{diag}(|F\beta|^{2\gamma}) A_o^T), \tag{4.16}$$

where

$$\sigma_m^2 = \frac{\sigma^2}{m}\Big(1 + 2\sum_{i=1}^{m-1}(1 - i/m)\rho(i)\Big). \tag{4.17}$$

The structure of the noise variance suggests a natural iterated generalized least squares (iGLS) scheme for the estimation of β.

Algorithm: *(Iterated GLS)*

(i) Set $\widehat{\beta_1} = [(A_oF)^T A_oF]^{-1}(A_oF)^T \overline{Y}_o(t_0)$ to be the OLS (ordinary least squares) estimate of β and let $k := 1$.

(ii) Set

$$\widehat{\beta}_{k+1} = [(A_oF)^T G(\widehat{\beta}_k) A_oF]^{-1}(A_oF)^T G(\widehat{\beta}_k)\overline{Y}_o(t_0), \tag{4.18}$$

where

$$G(\beta) := [A_o \mathrm{diag}(|F\beta|^{2\gamma})A_o^T]^{-1}. \tag{4.19}$$

(iii) Set $k := k + 1$ and repeat step *(ii)*. Iterate until $\|\widehat{\beta}_{k+1} - \widehat{\beta}_k\|$ falls below a certain "convergence" threshold.

Observe that the temporal correlation structure does not need to be estimated here since it appears only in the scalar coefficient σ_m^2 of the noise variance, which cancels in (4.18). The above iGLS scheme requires that the matrices involved in steps *(i)* and *(ii)* be invertible, a result established next.

Proposition IV.4. *Suppose that $F\beta > \vec{0}$ and let A_o be of full row–rank. Then:*

(i) The inverse $G(\beta)$ in (4.19) *exists, for all $\gamma > 0$.*

(ii) If A_oF is of full column–rank, then the inverses

$$\Sigma_{GLS}(\beta) := [(A_oF)^T G(\beta) A_oF]^{-1}, \tag{4.20}$$

and $[(A_oF)^T A_oF]^{-1}$ exist and are positive definite.

The proof is given in the Appendix and the assumptions are discussed in the remarks below it.

The following result establishes the *strong* consistency of the iterated GLS estimators $\widehat{\beta_k}$'s, even in the presence of long–range dependence. It also shows that the $\widehat{\beta_k}$'s are asymptotically equivalent to the (unavailable) GLS estimator $\widehat{\beta_{GLS}}$, provided $k \geq 2$.

Theorem IV.5. *Suppose that A_o and A_oF are of full row and column ranks, respectively, and let $F\beta > \vec{0}$. Then:*

(i) For all $k \geq 1$, we have $F\widehat{\beta_k} > \vec{0}$, a.s. as $m \to \infty$. Hence, the estimates $\widehat{\beta_k}$, $k \geq 1$ are well–defined, almost surely, as $m \to \infty$.

(ii) If $\rho(\tau) \to 0$, as $\tau \to \infty$, then for any fixed $k \geq 1$, we have

$$\widehat{\beta_k} \xrightarrow{a.s.} \beta, \quad \text{as } m \to \infty. \tag{4.21}$$

(iii) For all $k \geq 2$, we have that,

$$\widehat{\beta_k} - \widehat{\beta}_{GLS} = o_P(\sigma_m), \ \text{ as } m \to \infty,$$

where $\widehat{\beta}_{GLS}$ is the GLS estimate of β in the model (4.14)*, and σ_m^2 is given in* (4.17)*. Moreover,* $\mathrm{Var}(\widehat{\beta}_{GLS}) = \sigma_m^2 \Sigma_{GLS}(\beta)$ *with Σ_{GLS} as in* (4.20).

The proof is given in Appendix 4.3. In view of Slutsky's theorem and part *(iii)* of the above result, we obtain that for all $k \geq 2$,

$$\frac{1}{\sigma_m}(\widehat{\beta_k} - \beta) \Rightarrow \mathcal{N}\Big(0, \Sigma_{GLS}(\beta)\Big), \quad \text{as } m \to \infty. \tag{4.22}$$

Thus, the $\widehat{\beta_k}$'s ($k \geq 2$) are asymptotically optimal. Further, given σ_m, one can use (4.22) to obtain consistent confidence intervals for β.

As mentioned above, the scale parameter σ of the covariance structure in (4.14) is not involved in the formula for the predictors $\widehat{Y_u} = f(\widehat{\beta_k}, Y_o)$ (see e.g. (4.18)). The parameter σ is involved, however, in the expression of the prediction error

(4.30). Therefore, to gauge the accuracy of prediction, and to be able to use our estimators for detection of anomalies (see Sections 4.2.5 and 4.5 below), one needs an estimate of σ.

As in the case of the ordinary kriging estimator (see (4.25) below), a natural estimate of σ is obtained as follows:

$$\widehat{\sigma}^2 := \mathrm{vec}(\widehat{\Sigma}_{Y_o})^T \mathrm{vec}(\Sigma_{oo}(\widehat{\beta}))/\mathrm{vec}(\Sigma_{oo}(\widehat{\beta}))^T \mathrm{vec}(\Sigma_{oo}(\widehat{\beta})), \tag{4.23}$$

where $\widehat{\Sigma}_{oo}$ is the partition of the modeled covariance matrix corresponding the observed links, as in (4.26), $\widehat{\Sigma}_{Y_o} = \widehat{\Sigma}_{Y_o}(t_0)$ is the sample covariance matrix of the vector Y_o, calculated from past m observations $\{Y_o(t_0 - k),\ 0 \le k \le m-1\}$, and $\widehat{\beta}$ is an estimate of β.

Proposition IV.6. *Assume the conditions of Theorem IV.5(ii). Then, with* $\widehat{\beta} = \beta_k$, $k \ge 1$, *for* $\widehat{\sigma}^2$ *as in* (4.23), *we have* $\widehat{\sigma}^2 \overset{a.s.}{\to} \sigma^2$, *as* $m \to \infty$.

The proof is given in Appendix 4.3. We conclude this section with a few technical comments.

Remarks:

1. Here we do not estimate the temporal correlation structure $\rho(\cdot)$. In practice, this can be done, but the presence of LRD makes the precise asymptotic analysis cumbersome. This will be pursued in another work. Practical approaches are discussed in Section 4.2.5.
2. The model in (4.14) is realistic only if $F\beta > \vec{0}$. In our experience, the estimates $\widehat{\beta}_k$ obtained from real network data always satisfy $F\widehat{\beta}_k > \vec{0}$. This is perhaps due to the careful (optimal) choice of the matrix F discussed in Section 4.2.1.
3. The assumption that A_o is of full row–rank is natural since for prediction purposes, one need not include in the set of observed links ones that are per-

fect linear combination of other observed links. In practice, such a redundant scenario can arise only in the trivial case when some nodes do not generate traffic.

4. The full column–rank condition on $A_o F$ is required for the identifiability of β. If the dimension of β is greater than the number of observed links, then the model parameters cannot be identified (see also Section 4.4.2 and Figure 4.6a). In practice, we implement the iGLS procedure by using the Moore–Penrose generalized inverse.

4.2.4 On the implementation of simple and ordinary kriging

The baseline estimator in Section 4.1 is essentially the *simple kriging* predictor, which assumes knowledge of the mean and covariance of Y ($\mu_Y(t)$ and $\Sigma_Y(t)$). In practice, we estimate these quantities from moving windows of past data: $\mu_Y(t) \approx \widehat{\mu}_Y(t) := \frac{1}{m}\sum_{j=1}^{m} Y(t-j)$ and

$$\Sigma_Y(t) \approx \widehat{\Sigma}_Y(t) := \frac{1}{m-1}\sum_{j=1}^{m}(Y(t-j)-\widehat{\mu}_Y(t))(Y(t-j)-\widehat{\mu}_Y(t))^T.$$

The *ordinary kriging* methodology is used in our first solution of the prediction problem in Section 4.1. In this case, Σ_Y is modeled by $\sigma_X^2 AA^T$, where the scale σ_X is unknown. The means $\mathbb{E}Y_\ell = \mu_Y$ are unknown but assumed to be constant across the links $1 \le \ell \le L$. Here σ_X and μ_Y are allowed to vary slowly with time t. In contract, to the baseline estimator, we can no longer use $\widehat{\mu}_Y(t)$ and $\widehat{\Sigma}_Y(t)$ above since only some links are observed. Under these assumptions, the least squares optimal linear predictor of a link $Y_u(t)$ from $Y_o(t)$ becomes

$$\widehat{Y}_u(t) = \Lambda Y_o(t), \text{ where } \Lambda = \begin{pmatrix} \Gamma_{oo} & \vec{1} \\ \vec{1}^T & 0 \end{pmatrix}^{-1} \vec{\gamma}_{ou}, \tag{4.24}$$

where $\Gamma_{oo} = (\mathbb{E}(Y_{\ell_i} - Y_{\ell_j})^2)_{\ell_i,\ell_j \in \mathcal{O}}$ is a matrix of the variograms for the set of observed links $\mathcal{O}$ and $\vec{\gamma}_{ou} = (\mathbb{E}(Y_u - Y_\ell)^2)_{\ell \in \mathcal{O}}$ is a vector of cross–variograms between the unobserved link and observed links. For more details, see [6]. The resulting ordinary kriging coefficients are such that $\vec{1}^T \Lambda = 1$ so that the predictor is unbiased. In our application, we calculated the variograms by estimating the unknown parameter σ_X from a window of past data from the *observed links* $Y_o(t)$. Namely, since $\Sigma_{Y_o} = \sigma_X^2 A_o A_o^T$, we obtain the linear regression estimate

$$\widehat{\sigma}_X^2 = [\mathrm{vec}(A_o A_o^T)^T \mathrm{vec}(A_o A_o^T)]^{-1} \mathrm{vec}(\widehat{\Sigma}_{Y_o})^T \mathrm{vec}(A_o A_o^T), \tag{4.25}$$

where $\mathrm{vec}(B)$ stands for the vectorized matrix B. The estimator $\widehat{\sigma}_X^2$ corresponds to minimizing $\|\mathrm{vec}(\widehat{\Sigma}_{Y_o}) - \sigma^2 \mathrm{vec}(A_o A_o^T)\|$ with respect to σ^2.

4.2.5 Model–based Kriging and Prediction: Implementation

In this section, we describe how to estimate the model, and use it to perform kriging, and prediction of link loads in practice. The performance of this methodology is evaluated in the next section.

1. Learning the Model: The first step involves using the *expensive* Netflow data to 'learn the model', i.e., construct the matrix F and estimate the parameter γ as discussed in Section 4.2.

2. Estimating Model Parameters: After learning the model, we use the *inexpensive* link loads data to estimate the parameters β and σ from (4.14). Namely, β is estimated with iterated GLS from the sample means of observed links in (4.15) as in Section 4.2.3. Then, σ is estimated by using (4.23).

In order to carry out *temporal prediction*, it is also necessary to obtain an estimate $\widehat{H}$ of the Hurst exponent H. This is a well–studied and yet difficult problem. Here, we use the robust wavelet method of [27].

3. Estimating Traffic: Given F and γ, along with the parameter estimates $\widehat{\beta}$, $\widehat{\sigma}^2$, and $\widehat{H}$, one can obtain instantaneous estimates of the traffic loads on unobserved links (kriging) and/or estimates of the future traffic loads (prediction).

Let

$$\widehat{\mu}_Y := AF\widehat{\beta} = \begin{pmatrix} \widehat{\mu}_o \\ \widehat{\mu}_u \end{pmatrix} \qquad \text{and} \qquad \widehat{\Sigma}_Y := \widehat{\sigma}^2 \mathrm{diag}\left(|F\widehat{\beta}|^{2\gamma}\right) = \begin{pmatrix} \widehat{\Sigma}_{oo} & \widehat{\Sigma}_{ou} \\ \widehat{\Sigma}_{uo} & \widehat{\Sigma}_{uu} \end{pmatrix}. \tag{4.26}$$

The *kriging estimator*, obtained from (4.2) is:

$$\widehat{Y}_u(t) = \widehat{\mu}_u + \widehat{\Sigma}_{uo}\widehat{\Sigma}_{oo}^{-1}\left(Y_o(t) - \widehat{\mu}_o\right). \tag{4.27}$$

A plug–in estimate of the mean squared error obtained from (4.3) is given by:

$$\text{m.s.e.}(\widehat{Y}_u) = \widehat{\Sigma}_{uu} - \widehat{\Sigma}_{uo}\widehat{\Sigma}_{oo}^{-1}\widehat{\Sigma}_{ou}. \tag{4.28}$$

One can show that the resulting plug–in estimates are asymptotically optimal under the model. Indeed, let $\widetilde{Y}_u$ be the kriging estimator based on the *unavailable in practice* GLS estimate of β. By using Theorem IV.5, one can show that $\widehat{Y}_u - \widetilde{Y}_u \overset{a.s.}{\to} 0$, as $m \to \infty$, i.e. the estimates $\widehat{Y}_u$ converge to optimal GLS–based estimate $\widetilde{Y}_u$.

By using the estimated Hurst exponent, $\widehat{H}$, we also obtain expressions for the *temporal predictors*. Assume that the temporal correlation structure is given by the standard fractional Gaussian noise (see Section 3.1.1). By plugging $\widehat{H}$ for H in (4.5) and (4.6), we we obtain the corresponding estimated quantities: $\widehat{\Gamma}_m$, $\widehat{\gamma}_m(h)$, $\widehat{c}(h)$, and $\widehat{\sigma}^2(h)$, for a given time lag h. By using these quantities and (4.4), we obtain predictors of the traffic load on an *observed* link h steps into the future:

$$\widehat{Y}_o(t_0 + h) = \widehat{\mu}_o + \sum_{j=0}^{m} \widehat{c}_j(h)(Y_o(t_0 - j) - \hat{\mu}_o). \tag{4.29}$$

The mean squared error estimate is:

$$\text{m.s.e.}(\widehat{Y}_o(t_0 + h)) = \widehat{\sigma}^2(h)\widehat{\Sigma}_{oo} \tag{4.30}$$

Similarly, by (4.7) we obtain predictors for *unobserved* links h steps ahead into the future:

$$\widehat{Y}_u(t_0+h) = \widehat{\mu}_u + \widehat{\Sigma}_{uo}\widehat{\Sigma}_{oo}^{-1}(\widehat{Y}_o(t_0+h) - \widehat{\mu}_o). \tag{4.31}$$

In this case, the mean squared error estimate is:

$$\text{m.s.e.}\Big(\widehat{Y}_u(t_0+h)\Big) = \widehat{\sigma}^2(h)\widehat{\Sigma}_{uo}\widehat{\Sigma}_{oo}^{-1}\widehat{\Sigma}_{ou} + \widehat{\Sigma}_{uu} - \widehat{\Sigma}_{uo}\widehat{\Sigma}_{oo}^{-1}\widehat{\Sigma}_{ou}. \tag{4.32}$$

Remarks:

1. Note that $\widehat{\sigma}^2$ is a consistent estimate of σ^2, as $m \to \infty$, provided $\rho(\tau) \to 0,\ \tau \to \infty$. In view of (4.27), the temporal correlation structure does not play a role in the kriging estimate. On the other hand, by (4.28), we see that the variance of this predictor does depend on ρ.

2. As discussed in Section 4.2.3, the expression for σ_m^2 in (4.17) is not tractable under the general assumptions of long–range dependence. However, if the correlation structure ρ is that of fGn (3.6), then by using the self–similarity of fBm one obtains:

$$\sigma_m^2 = \frac{\sigma^2}{m^{2(1-H)}}.$$

   In Section 4.4, we use this fact and point estimates of H in order to obtain prediction intervals.

4.3 Proofs of Sections 4.2 and 4.5

Proof of Proposition IV.1. Let $W = \text{span}\{\vec{f}_1, \cdots, \vec{f}_p\}$, where $\{\vec{f}_1, \cdots, \vec{f}_J\}$ is an orthonormal basis of $\mathbb{R}^J$. Observe that, for all $1 \le k \le n$:

$$\|m_X(k) - P_W(m_X(k))\|^2 = \sum_{j=p+1}^{J} \langle m_X(k), \vec{f}_j\rangle^2 = \sum_{j=p+1}^{J} \vec{f}_j^{\,T}(m_X(k)m_X(k)^T)\vec{f}_j.$$

Now, by summing over k, the right–hand side of (4.11) becomes:

$$\sum_{j=p+1}^{J} \vec{f_j}^T \Big(\sum_{k=1}^{n} m_X(k) m_X(k)^T \Big) \vec{f_j} = \sum_{j=p+1}^{J} \vec{f_j}^T M \vec{f_j}.$$

Clearly, the last sum is minimized when $\text{span}\{\vec{f}_{p+1}, \cdots, \vec{f}_J\} = (W^*)^{\perp} \equiv \text{span}\{\vec{b}_{p+1}, \cdots, \vec{b}_J\}$. In this case, this sum equals $\sum_{j=p+1}^{J} \lambda_j$. □

Proof of Proposition IV.4. Since $F\beta > \vec{0}$, the matrix $\text{diag}(|F\beta|^{2\gamma})$ is positive definite. Thus, the fact that A_o is of full *row–rank*, implies that the square matrix $A_o \text{diag}(|F\beta|^{2\gamma}) A_o^T$ is of full row–rank and hence invertible. This proves *(i)*.

To show *(ii)*, let $x \in \mathbb{R}^p$ and suppose that $x^T (A_o F)^T G(\beta) A_o F x = 0$. Thus, for the vector $y = A_o F x$, we have $y^T G(\beta) y = 0$. This, since $G(\beta)$ is positive definite, implies that $y = A_o F x = \vec{0}$, which in turn yields $x = \vec{0}$, because $A_o F$ has a trivial null–space. We have thus shown that $(A_o F)^T G(\beta) A_o F$ is a positive definite matrix. □

Proof of Theorem IV.5. Note that

$$\overline{Y}_o(t_0) = A_o F\beta + \sigma A_o \text{diag}(|F\beta|^{\gamma}) \overline{Z}(t_0),$$

where $\overline{Z}(t_0) = \frac{1}{m} \sum_{t=t_0-m+1}^{t_0} Z(t)$. Since $\rho(\tau) \to 0,\ \tau \to \infty$, the Maruyama's Theorem implies that the Gaussian process $Z = \{Z(t)\}_{t \in \mathbb{Z}}$ is mixing. Therefore, $\overline{Z}(t_0) \xrightarrow{a.s.} 0$, and hence $\overline{Y}_o(t_0) \xrightarrow{a.s.} A_o F\beta$, as $m \to \infty$.

Note that the OLS estimator $\widehat{\beta_1}$ is well–defined since $A_o F$ is of full column rank. Observe also that, since $\overline{Y}_o(t_0) \xrightarrow{a.s.} A_o F\beta$, we have

$$\widehat{\beta_1} - \beta = [(A_o F)^T A_o F]^{-1} (A_o F)^T (\overline{Y}_o(t_0) - A_o F\beta) \xrightarrow{\text{a.s.}} 0, \text{ as } m \to \infty.$$

We proceed by induction. Let $k \geq 2$ and $\widehat{\beta}_{k-1} \xrightarrow{a.s.} \beta,\ m \to \infty$. Then, since $F\beta > \vec{0}$, by continuity, $F\widehat{\beta}_{k-1} > \vec{0}$, almost surely, as $m \to \infty$, and $\widehat{\beta}_k$ is well–defined, as $w \to \infty$ (Proposition IV.4). Further, for all $k \geq 2$, by (4.18),

$$\widehat{\beta}_k = C(\widehat{\beta}_{k-1}) \overline{Y}_o(t_0), \tag{4.33}$$

with the matrix

$$C(\widetilde{\beta}) := [(A_o F)^T G(\widetilde{\beta}) A_o F]^{-1} (A_o F)^T G(\widetilde{\beta}). \tag{4.34}$$

Note that $C(\widetilde{\beta})$ is well–defined and continuous for $F\widetilde{\beta} > \vec{0}$. Since, also $C(\beta)A_o F\beta = \beta$, the convergences $\overline{Y}_o(t_0) \overset{a.s.}{\to} A_o F\beta$, and $\widehat{\beta}_{k-1} \overset{a.s.}{\to} \beta$, imply that $\widehat{\beta}_k \overset{a.s.}{\to} \beta$, as $m \to \infty$. We have thus shown parts *(i)* and *(ii)*.

We shall now prove *(iii)*. As in (4.33), for all $k \geq 2$, we have

$$\widehat{\beta}_k = C(\widehat{\beta}_{k-1})\overline{Y}_o(t_0) \text{ and also } \widehat{\beta}_{GLS} = C(\beta)\overline{Y}_o(t_0).$$

Note also that $C(\beta)A_o F\beta = \beta = C(\widehat{\beta}_k)A_o F\beta$, and therefore,

$$\widehat{\beta}_k - \widehat{\beta}_{GLS} = (C(\widehat{\beta}_{k-1}) - C(\beta))(\overline{Y}_o(t_0) - A_o F\beta). \tag{4.35}$$

Now, by (4.16) and (4.19), we have

$$\mathrm{Var}(\overline{Y}_o(t_0)) = \sigma_m^2 G(\beta)^{-1}. \tag{4.36}$$

Thus, Relation (4.35), the convergence $\widehat{\beta}_{k-1} \overset{a.s.}{\to} \beta$, $m \to \infty$, and the continuity of $C(\cdot)$ imply that $\widehat{\beta}_k = \widehat{\beta}_{GLS} = o_P(\sigma_m)$, $m \to \infty$. Note also that by (4.34) and (4.36), we readily have $\mathrm{Var}(\widehat{\beta}_{GLS}) = \sigma_m^2 \Sigma_{GLS}(\beta)$. □

Proof of Proposition IV.6. As in the proof of Theorem IV.5, the Maruyama's theorem implies that $\widehat{\Sigma}_{Y_o} \overset{a.s.}{\to} \Sigma_{Y_o}$, as $m \to \infty$. By Theorem IV.5 *(ii)* we also have $\widehat{\beta}_k \overset{a.s.}{\to} \beta$, as $m \to \infty$. Note that the right–hand side of (4.23) is a continuous function of $\widehat{\beta}$ and $\widehat{\Sigma}_{Y_o}$, which by (4.14), equals σ^2 when $\widehat{\beta}$ and $\widehat{\Sigma}_{Y_o}$ are replaced by β and Σ_{Y_o}, respectively. This implies the strong consistency of $\widehat{\sigma}^2$. □

Proof of Proposition IV.7. The variance of $\widetilde{Z}_t$ is then given by:

$$\mathbb{E}[\widetilde{Z}_t^2] = (1-\phi)^2 \sum_{j=0}^{\infty}\sum_{k=0}^{\infty} \phi^{j+k}\gamma(j-k) = (1-\phi)^2 \sum_{j=0}^{\infty}\sum_{k=0}^{\infty} \phi^{j+k} \int_{-\pi}^{\pi} e^{i\theta(k-j)} f(\theta)d\theta,$$

where $f(\theta)$ stands for the spectral density of $\{Z_k\}$. By using the expression of f given in equation 9.12 on p.34 of [24] , we obtain that $\mathbb{E}[\widetilde{Z}_t^2]$ equals

$$\lambda^2 \int_{-\pi}^{\pi} \Big| \sum_{k=0}^{\infty} \phi^k e^{i\theta k} \Big|^2 f(\theta) d\theta = \lambda^2 C_2(H)^{-2} \int_{-\infty}^{\infty} \frac{2(1-\cos(\theta))|\theta|^{-(2H+1)}}{\phi^2 + 1 - 2\phi\cos\theta} d\theta$$
$$= \lambda^2 C_2(H)^{-2} \int_{-\infty}^{\infty} \frac{2(1-\cos(\theta))|\theta|^{-(2H+1)}}{\lambda^2 - 2\lambda(1-\cos\theta) + 2(1-\cos\theta)} d\theta.$$

□

4.4 Model Validation and Calibration

In this section, we evaluate our model in the context of traffic prediction. We focus on 9 representative scenarios described in Tables 2.1, 6.1 and Appendix 2.1.

4.4.1 Performance and Model Robustness

Table 4.1 provides relative mean squared errors (ReMSE) for the optimal baseline estimator, the ordinary kriging estimator, and the proposed model (with $p = 2$). The reported ReMSE's of the estimates are computed as follows:

$$\text{ReMSE}(\widehat{Y}) = \sum_{t=1}^{T} \|\widehat{Y}(t) - Y(t)\|^2 / \sum_{t=1}^{T} \|Y(t)\|^2.$$

Here $\widehat{Y}(t)$ is an estimator (obtained by either kriging or temporal prediction) for the true value $Y(t)$ and $\|\cdot\|$ stands for the Euclidean norm. The ReMSE's quantify empirically the prediction error relative to the energy of the true 'signal' $Y(t)$, over the duration T. In a controlled setting where $Y(t)$ is available, the ReMSE's allow us to objectively compare the performance of various estimators.

Scenarios 1–7 represent situations where the observed links share sufficiently many flows with the unobserved ones to make relatively accurate prediction. In *all* these cases, the *network–specific model* outperforms the naïve *ordinary kriging* method, with an average improvement of the ReMSE by 0.2072 points or 20.72%.

Scen.	Baseline	Ordinary Kriging	Model	02-18	02-20	02-26	03-12
1	0.0305	0.4052	0.2476	0.2342	0.2363	0.2629	0.2209
2	0.0287	0.1266	0.0517	0.0461	0.0550	0.0424	0.0746
3	0.0288	0.3279	0.0514	0.0459	0.0549	0.0425	0.0750
4	0.0314	0.1209	0.0512	0.0696	0.0658	0.0694	0.0596
5	0.0285	1.0241	0.2414	0.2651	0.2864	0.3344	0.2722
6	0.0262	0.1129	0.0468	0.0619	0.0587	0.0684	0.0462
7	0.0216	0.0614	0.0388	0.0501	0.0495	0.0564	0.0384
8	0.0727	0.8423	1.0322	1.1060	1.0687	1.4335	1.7803
9	0.0723	0.2649	0.6277	0.7618	0.6875	0.7792	0.7449

Table 4.1: Evaluation of Model Performance. *Columns 2 and 3*: ReMSE's of the baseline (simple kriging) and network–specific model for Feb 19, 2009. *Columns 4 to 7:* The matrix F was obtained from Feb 19, 2009 Netflow data (X_j's), and then used to fit the model and perform kriging based on link data (Y_ℓ's) for four additional days.

Ordinary kriging estimates are obtained by using the sample covariance matrix of all links (as in the *baseline*) and assuming equal but unknown flow–means (see e.g. [6]). The latter is a reasonable approximation when no Netflow data is available. The difference is as high as 78% and as low as 2.2% in favor of the network specific model. In most cases and across different days our model yields useful predictions with ReMSE's of about 5%. In Scenarios 8 and 9, however, fewer flows are shared by the observed and unobserved links and hence the accurate prediction is objectively more difficult. In these scenarios, the baseline predictor has over 2–3 times the ReMSE's as compared to Scenarios 1–7. This initial comparison shows that the joint model improves significantly upon the naïve ordinary kriging approach and comes close to the optimal ReMSE lower bound in the cases where the prediction problem is well–posed (Scenarios 1–7). Scenarios 8 and 9 illustrate that the accuracy of prediction has natural limitations, inherent to the routing of the network, that neither model can overcome.

One apparent limitation of the network specific approach is that it relies on expensive flow–level data (X_j's) to build the matrix F. Surprisingly, it turns out that once the matrix F is obtained from flow–level measurements during a single

day, it can be successfully used to model the link–level traffic for many days in the future. That is, even though the model requires the extensive off–line analysis of Netflow data, once it is built, it can be readily estimated *on–line* using only link–level data and used for several days before it has to be updated. It is remarkable that all results in Table 4.1 are based on a model (i.e. a matrix F) learned from Feb 19, 2009 flow–level data. Then, the *same* model was used to predict Y_ℓ's in all 9 scenarios for 5 different days. Even a month later, this model continues to outperform the ordinary kriging in the first seven scenarios. In only one of these scenarios therein we have an appreciable increase in the prediction ReMSE's due perhaps to an outdated model. These results may be attributed to the fact that the structure of the traffic means across all flows in the network, although complex, is relatively constant, and is therefore well–captured by the principal components involved in the matrix F. The model must be updated should structural changes in the network occur.

The performance of the model under *temporal prediction* may be seen in Table 4.3 and Figure 4.4. Table 4.3 shows that the prediction is better for links that were directly observed in the past in comparison with unobserved ones, as might be naturally expected (see also (4.30) & (4.32)). More importantly, the model–based approach remains close to the unattainable baseline. The model–based predictor in Figure 4.4 may be compared to the baseline predictor in Figure 4.1 (bottom). Although the model–based predictor does not track the average traffic level as precisely as the baseline, it has similar behavior. For small time lags (h), the predictor follows many of the features. As the time lag increases, the predictor becomes smoother and closer to the traffic mean.

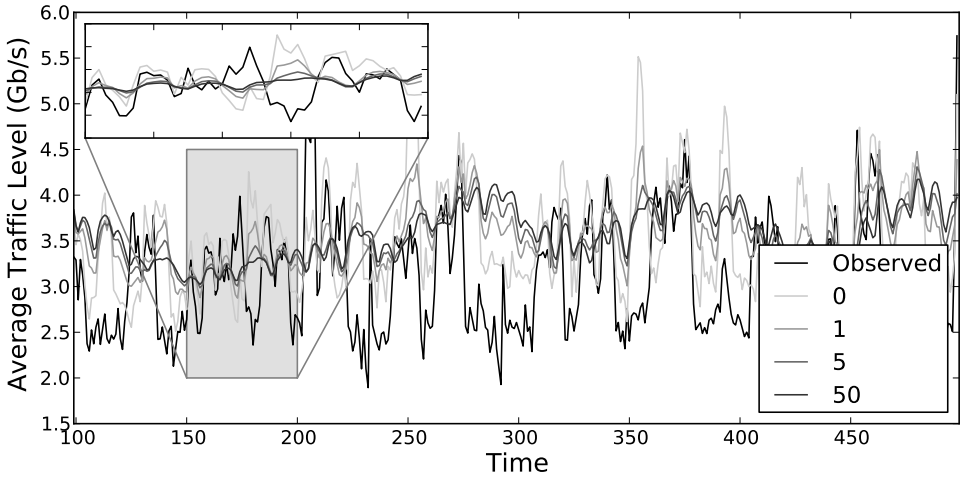


Figure 4.4: Temporal Prediction for various time horizons using the joint model.

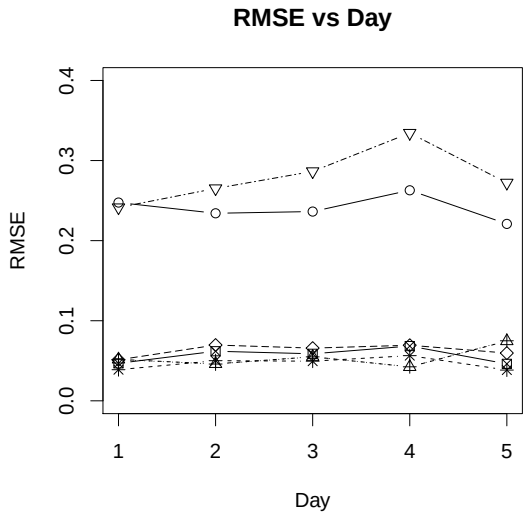


Figure 4.5: ReMSE of network–specific model over time. The model was learned on Feb 19, 2009 (1). The matrix is then used to predict the previous day (2), the next day (3), a day one week later (4), and a day 4 weeks later (5). Each line corresponds to one of the first seven scenarios described in Table 6.1.

	Baseline		Model	
h	Observed Link (3)	Unobserved Link (13)	Observed Link (3)	Unobserved Link (13)
1	0.0111	0.0347	0.0111	0.0444
5	0.0367	0.0329	0.0366	0.0452
10	0.0389	0.0322	0.0388	0.0402
25	0.0418	0.0426	0.0417	0.0522
50	0.0367	0.0402	0.0366	0.0499

Table 4.2: Performance of Network Prediction, in terms of ReMSE, for observed links and unobserved links. Here we compare the Baseline situation (with known mean and covariance) to the model, for both an observed link (3) and an unobserved link (13). The link is predicted according to Scenario 6.

Time Step	1	3	5	10	15	20
Observed Link, Baseline	0.01	0.04	0.04	0.04	0.04	0.04
Unobserved Link, Baseline	0.01	0.04	0.04	0.04	0.04	0.05
Observed Link, Model	0.03	0.04	0.03	0.03	0.03	0.04
Unobserved Link, Model	0.03	0.04	0.03	0.03	0.03	0.04

Table 4.3: Performance of Network Prediction, in terms of ReMSE, for observed links and unobserved links. Here we compare the Baseline situation (with known mean and covariance) to the model, for both an observed link (3) and an unobserved link (13). The link is predicted according to Scenario 6.

4.4.2 Calibration

Applying the model to real data relies on the choice of several parameters, such as p, γ, and m, as described in Section 4.2. The prediction performance is remarkably robust to the choice of these parameters, as discussed in detail below.

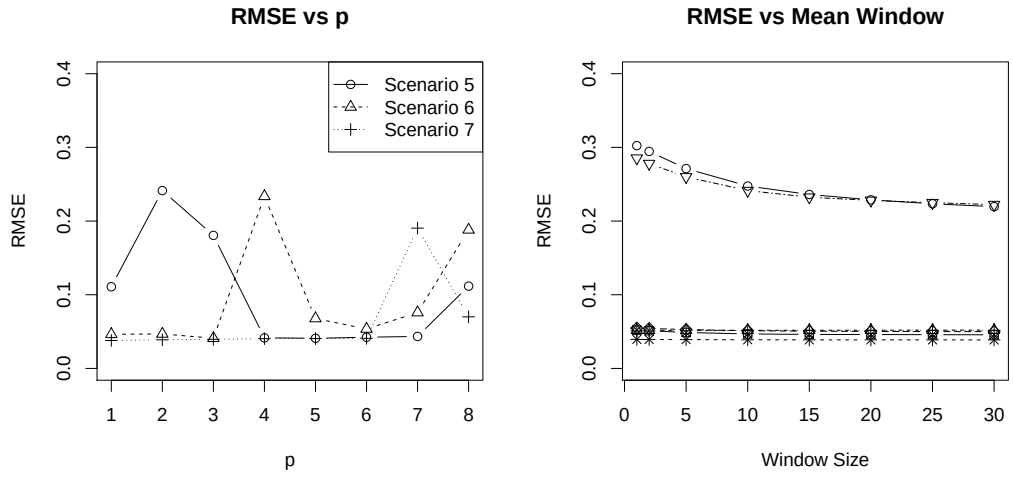
• *The role of p*: The parameter p equals the number of principal components (columns of the matrix F) used to model the traffic means in (4.9). The prediction performance is robust to the choice of p, provided that p is less than the number of *observed links* used in prediction. Figure 4.6a shows the ReMSE's for 3 prediction scenarios as a function of p. In Scenarios 5, 6, and 7 the *same* link is predicted via two, three, and seven other links, respectively (see Table 6.1). If p exceeds the number of observed links, then the parameter β in (4.14) is not identifiable, potentially resulting in poor performance. This explains the peaks in the ReMSE's at $p = 2,\ 4$, and 7 in Scenarios 5–7. Surprisingly, in the first two cases the

ReMSE's recover as p grows, even in the presence of non–identifiability. Similar patterns are seen in the other six prediction scenarios (omitted, for simplicity). The performance of the model remains stable for all choices of p less than the number of predictors.

In light of these results, we advocate using a relatively small value of p (e.g. $p = 2$). While a larger value of p can slightly improve prediction errors when many links are observed, having small value of p allows one to fit the model in a wide variety of prediction scenarios, without sacrificing the overall performance. Recall also Figure 4.2b.

• *The role of* γ: This parameter controls the mean/variance relationship in the model (see (4.9) and (4.13)). We observed the relationship (4.12), between the sample means $\overline{X}_j(t)$ and standard deviations $S_j(t)$'s obtained from windows of the flow–level data. The parameter γ was estimated by using a log–linear regression of $\overline{X}_j(t)$ versus $S_j(t)$, over $j,\ 1 \le j \le \mathcal{J}$. This was done for a range of window sizes and times t, and the estimates were found to be stable and $\widehat{\gamma} \approx 3/4$. Independently, in [26], we explored the sensitivity of the model to the choice of γ and found that the ReMSE's are robust to all choices $\gamma \in [0.5, 1]$. Small values of $\gamma \approx 0.5$ lead generally to slightly better ReMSE as compared to larger γ's. This may be due to the fact that the small powers γ lead to a "smoother" covariance matrix and hence have a regularizing effect. In practice, however, we need not only accurate prediction but also adequate models for the variance, in order to have reliable estimates of the prediction error. Therefore, we recommend using $\gamma = 3/4$ as inferred from the data.

• *The role of the window size m:* In practice, at each time point t, the model (4.14) is estimated from a window of past m data $\{Y_o(t-k),\ 0 \le k \le m-1\}$. Namely,



(a) Role of p

(b) Mean Window

Figure 4.6: *Left:* ReMSE of scenarios 5, 6, and 7 as a function of p. Performance suffers when p exceeds the number of observed links, but is otherwise robust to the choice of p. *Right:* ReMSE of scenarios 1–7, as a function of window size m used to estimate the means. We used $p = 2$ in these cases.

β is obtained by using the iGLS algorithm and σ from (4.23) (see Section 4.2.3). Figure 4.6b illustrates the effect of the window size m on the quality of prediction. Note that in all scenarios therein the prediction performance is rather robust to the choice of m, provided that $m \geq 10$. It is remarkable that with the exception of 2 out of the 7 shown prediction scenarios the model works well even when m is less than 10.

Recall that the scalar σ does not affect the prediction and the ReMSE's in Figure 4.6b depend only on the quality of estimation of β. The parameter σ is involved in the prediction error. Our experiments with simulated data (not shown here, for simplicity) show that the estimates of σ are also robust to the window size m.

• *Convergence of* $\widehat{\beta}_{iGLS}$: In practice, the estimates of $\widehat{\beta}_{GLS}$ stabilize after a few iterations. Here we assume the convergence criterion $\|\widehat{\beta}_k - \widehat{\beta}_{k+1}\| < \epsilon$, with $\epsilon = 0.001$, for example. In the figures and tables, however, the algorithm was allowed to run

for at least 20 iterations to be conservative.

4.4.3 Model Misspecification

We use the joint model (4.14) with simulated data that violate the assumption of stationarity. Our goal is to understand the limitations of the model, when applied to network traffic with slowly changing trend. Network flows are simulated using independent fractional Gaussian noise (fGn) time series with self–similarity parameter $H = 0.8$ (see Section 4.5.1, below). This value of H is typical for many real network flows that we examined.

We compare the mean squared prediction errors in the stationary and non–stationary regimes. In the stationary case, constant means are added to the simulated fGn's to produce realistic traffic volumes. Non–stationary traffic traces were obtained by adding a sinusoidal trend to all simulated stationary flows. In both cases (stationary and non–stationary), link–level data was obtained from the simulated flow–level data through the routing equation (2.2).

We focus on prediction Scenario 6 in Table 6.1. We computed the baseline simple kriging predictor $\widetilde{Y}$ by using the known means and covariances of the simulated data. We also estimated our model and used it to obtain a predictor $\widehat{Y}$ of the unobserved link.

Table 4.4 shows the resulting prediction errors as a function of the window size used to estimate the parameter β. The empirical error of our estimators is comparable to the optimal MSE's for the baseline estimator. This is so even in the presence of non–stationarity. The major exception is when in the non–stationary case the window size (m) becomes close to the half–period (50) of the sinusoidal trend. This limited experiment shows that our model adapts well and it is essentially robust to non–stationary trends provided that relatively small window sizes

m are used.

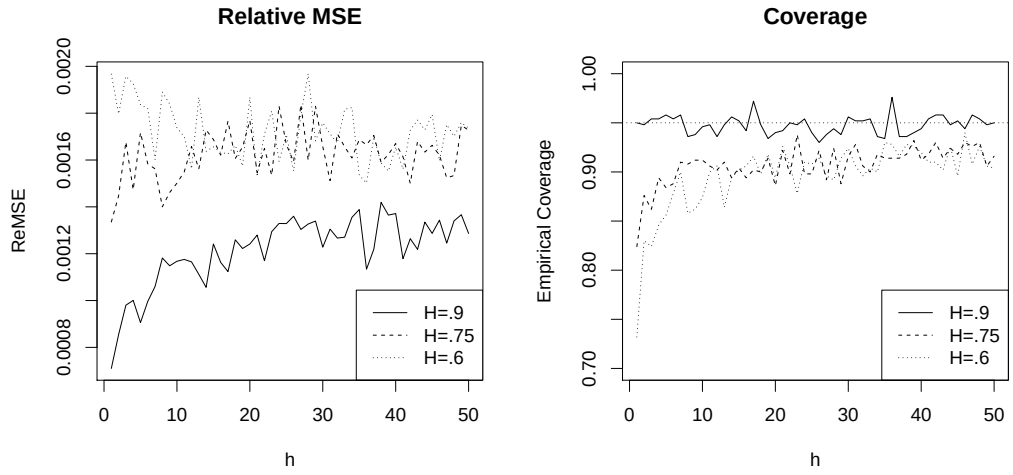
	Baseline	Network Specific Model							
Window Size (m)		5	10	25	30	50	75	100	200
Stationary	2.61	2.93	2.88	2.83	2.82	2.80	2.78	2.77	2.75
Non–Stationary	2.61	4.15	4.15	4.50	4.71	5.42	6.19	4.99	4.98

Table 4.4: Empirical mean squared errors for the baseline (simple kriging) predictor $\widetilde{Y}$ and the predictor $\widehat{Y}$, based on our network–specific model with $p = 2$. Time series of 20,000 observations were used.

We also examine the effect of an incorrect estimate of the Hurst exponent H on temporal prediction. We simulated traffic from fGn with three different values of H: 0.9, 0.75, and 0.6. For each simulation, a plug–in of $\widehat{H} = 0.9$ was used in conjunction with known mean and covariance to construct temporal predictors and prediction intervals (in Scenario 6). The results are summarized in Figure 4.7. We see that the ReMSE is higher in both cases, and the prediction intervals undercover when H is over–estimated. Overall, overestimating H results in assuming a stronger temporal dependence than is present in the traffic. This results in stronger weight being placed on recent observations in calculating the predictors and leading to narrow prediction intervals, which explains the lower than nominal coverage in Figure 4.7.

4.5 Statistical Detection of Anomalies under Long Range Dependence

In this section, we present an application of the above methodology to the case when all links on the network are observed. In this case, for each link ℓ, one can compare the observed $Y_\ell(t)$ and the predicted $\widehat{Y}_\ell(t)$ traffic (computed by using all links with the exception of ℓ). If statistically significant deviations are encountered, then this can serve as a flag of an anomaly or some structural change in the network traffic.



(a) Performance

(b) Coverage

Figure 4.7: The effect of mis–specified Hurst exponent H on temporal prediction. We simulated 500 independent fGn traces for each of three different Hurst exponents: 0.9, 0.75, and 0.6. In each case, the predictors were obtained assuming that $H = 0.9$. Plotted are the resulting empirical ReMSE's and coverages for 95%–prediction intervals.

To illustrate and detect such differences, we use a modified exponentially weighted moving average (EWMA) control chart on the differences $Y_\ell - \widehat{Y_\ell}$. The latter have zero means and variances equal to the prediction error, which can be estimated from (4.30) and (4.23).

Although EWMA control charts are widely used and well–studied (see, e.g, [1]), they rely on an assumption of independent or weakly dependent (in t) observations. Computer network traffic is long–range dependent (LRD) and the usual variance formula used in the EWMA charts does not apply. We show next how the control limits of these charts can be adjusted to account for the presence such dependence.

4.5.1 Control charts for long–range dependent data

Consider the EWMA with discount factor $\phi \in (0,1)$ of the time series $\{Z_k\}_{k\geq 0}$:

$$\widetilde{Z}_t := (1-\phi)(Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \cdots) \tag{4.37}$$

	Link Type	$H=$	Time Step:	1	3	5	10	15	20
Baseline	Observed	.9	Coverage	0.85	0.82	0.81	0.84	0.85	0.85
		.9	Width	4.51	5.31	5.54	5.78	5.91	5.99
	Observed	.6	Coverage	0.78	0.82	0.83	0.87	0.91	0.90
		.6	Width	6.00	7.07	7.36	7.69	7.86	7.97
	Unobserved	.9	Coverage	0.91	0.91	0.85	0.89	0.86	0.86
		.9	Width	8.89	9.45	9.62	9.80	9.90	9.96
	Unobserved	.6	Coverage	0.85	0.93	0.92	0.90	0.93	0.89
		.6	Width	12.22	12.89	13.09	13.32	13.43	13.51
Model	Observed	.9	Coverage	0.78	0.74	0.75	0.81	0.77	0.83
		.9	Width	5.02	5.92	6.17	6.44	6.58	6.67
	Observed	.6	Coverage	0.28	0.32	0.35	0.42	0.39	0.34
		.6	Width	2.00	2.36	2.46	2.57	2.62	2.66
	Unobserved	.9	Coverage	0.76	0.73	0.73	0.79	0.75	0.80
		.9	Width	10.37	10.95	11.12	11.31	11.41	11.47
	Unobserved	.6	Coverage	0.39	0.38	0.38	0.36	0.46	0.36
		.6	Width	4.20	4.43	4.50	4.58	4.62	4.65

Table 4.5: Coverage and Width for Simulated Traffic. In each case, a plug–in estimate of $H = .9$ was used, while the actual value of the Hurst parameter differed.

Letting $\lambda = 1-\phi$, this moving average may be efficiently updated via $\widetilde{Z}_t = \lambda Z_t + (1-\lambda)\widetilde{Z}_{t-1}$. For independent Z_t's, for $\mathrm{Var}(\widetilde{Z}_t) = \sigma^2_{\widetilde{Z}}$, we have $\sigma^2_{\widetilde{Z}} = \frac{\lambda}{2-\lambda}\sigma^2_Z$. When the Z_t's are long–range dependent (such as in the case of network traffic), however, the latter formula underestimates the variance $\sigma^2_{\widetilde{Z}}$, which can lead to frequent false positive alarms.

As discussed in Chapter III, Internet traffic traces exhibit long–range dependence, which can be well modeled by using fractional Gaussian noise, a stationary time series with auto–covariance given by (3.6). The following result provides an expression for the variance $\sigma^2_{\widetilde{Z}}$ of the EWMA control chart corresponding to LRD fGn data.

Proposition IV.7. *For $\widetilde{Z}_t$ as in* (4.37) *with Z_t's an fGn with self–similarity parameter $H \in (0,1)$ and variance σ^2, we have:*

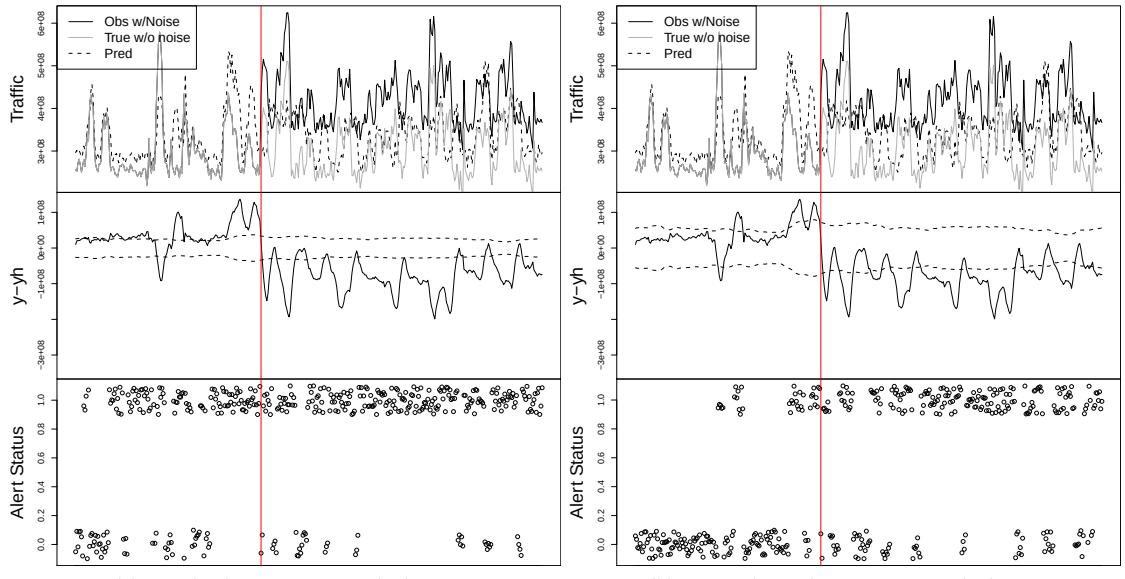
$$\mathrm{Var}(\widetilde{Z}_t) = \frac{\lambda^2\sigma^2}{C_2(H)^2}\int_{-\infty}^{\infty} \frac{2(1-\cos(\theta))|\theta|^{-2H-1}}{\lambda^2 - 2\lambda(1-\cos\theta) + 2(1-\cos\theta)}d\theta, \tag{4.38}$$

where $C_2(H)^2 := \pi/(H\Gamma(2H)\sin(H\pi))$.

The proof is given in Appendix 4.3. In practice, the expression in (4.38) is readily evaluated by using numerical integration.

4.5.2 Simulated Anomalies in Observed Network Traffic

We now present some examples of using the adjusted EWMA control chart for real network data, applied to the differences $Y_\ell(t) - \widehat{Y}_\ell(t)$. The mean is taken to be 0 for the duration of the control chart, since the predictor $\widehat{Y}_\ell(t)$ is unbiased under the model. The variance of the control chart is calculated using (4.38), with σ^2 replaced by $\widehat{\sigma}_Y^2$ (estimated in the prediction procedure). The Hurst LRD parameter H of the traffic is obtained by using the wavelet–based methods described in [22]. In each of the examples, a simple mean–shift anomaly is added to one source–destination flow. Each figure in this section shows a plot of the observed, predicted, and true traffic (top panels); the control chart of $|Y_\ell - \widehat{Y}_\ell|$ (middle panels), and an indicator of whether the process is identified as out of control (bottom panels). The vertical line indicates the onset of the simulated anomaly.



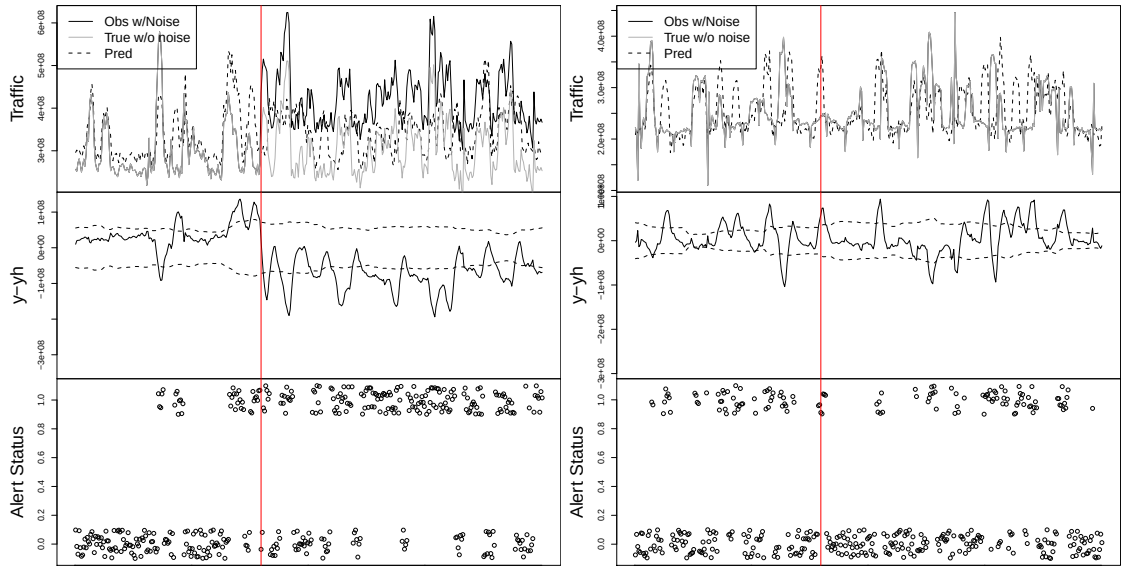
(a) Standard EWMA Control Chart (b) LRD–adjusted EWMA Control Chart

Figure 4.8: Performance of the standard EWMA control chart for i.i.d. data (left plot) and that of the LRD–adjusted chart (right plot).

Figure 4.8 demonstrates the importance of the LRD–adjustment for the control limits of the EWMA charts. Here, we examine link 13 (Kansas City to Chicago), which is predicted using all links sharing at least one flow with it (Scenario 7 in Table 6.1). A simulated anomaly is added to flow 20, which traverses only link 13. The standard chart results in far too many false positives and one cannot distinguish between anomalous and non–anomalous traffic. While the adjusted chart still has several false positives (due to high traffic variability), the onset of the anomaly is essentially detected.

The second example shows how one can use the chart to determine which flow is behaving anomalously. The mean shift was added to flow 6 (Kansas City to Atlanta), which traverses two links: 13 (Kansas City to Chicago) and 17 (Chicago-Atlanta) (see Figure 2.1). The LRD–adjusted control charts for Links 13 (Figure 4.9a) and 17 (not shown), clearly indicate the onset of the anomaly. The charts of the other links, not carrying the anomalous flow, (e.g. link 7 in Figure 4.9b) involve just a few false alarms and detect no anomaly. This suggests that the flow using links 13 and 17 (that is, flow 6) is experiencing the anomaly.

In the last example, we illustrate a case where the anomaly detection is inherently more challenging. We add a mean shift to the relatively long flow 14 (Seattle to Atlanta), which traverses four links: 3 (Seattle to Salt Lake City), 9 (Salt Lake City to Kansas City), 13 (Kansas City to Chicago), and 17 (Chicago to Atlanta) (see Figure 2.1). In Figure 4.10a, the control chart is based on predicting Link 17 by using all links that do not carry the anomalous flow. Unfortunately, these links do not provide sufficient information to predict Link 17, hence the predictor is a relatively 'smooth curve' as compared to the true traffic trace, and the error is relatively large. Nevertheless, we can pick up the anomaly. The segment with

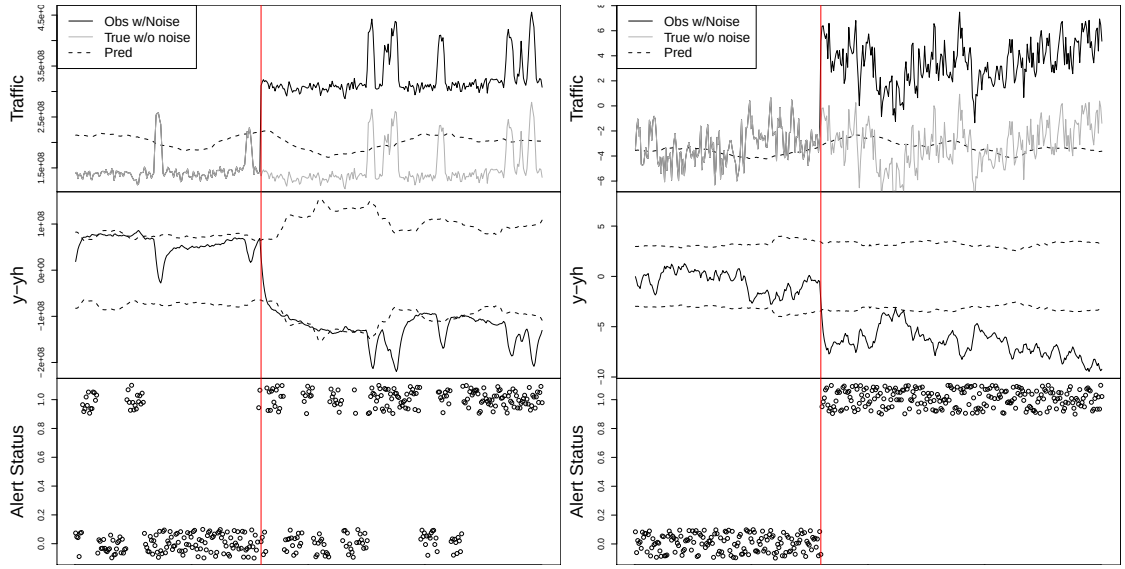


(a) Link Carrying Anomalous Flow (13)

(b) Link Not Carrying Anomalous Flow (7)

Figure 4.9: Detecting an anomalous multi–link flow. The simulated anomaly is added to a two–link flow. The links carrying the flow show anomalous behavior (left plot), while the rest behave as in the figure on right.

false positive alerts can be explained by the presence of small bias in our model, since $F\beta$ is not capturing perfectly the fine dynamics of the means. Indeed, if we repeat the exercise using simulated traffic (Figure 4.10b), it shows that again the predictor is not particularly useful i.e. it yields a smooth curve that tracks only the local means. In this situation, however, there is no bias and the control chart has no false positive alerts.



(a) Real Traffic (b) Simulated Traffic

Figure 4.10: An anomaly is added to flow 14 (Seattle to Atlanta), and a control chart is constructed on the Chicago–Atlanta Link (17), where all 'non–anomalous' links are used in the prediction. These links, however, do not provide enough information and the predictor is a relatively smooth curve.

CHAPTER V

Multi–Modal Network Kriging

In the previous chapters, we developed and studied a global model for computer network traffic. Although that model has many attractive theoretical properties and behaves well in practice, it does not account for certain small but real statistical dependencies present in such traffic. In this chapter, we introduce and explore an alternate model that explicitly models the TCP feedback mechanism, the underlying cause of such dependence.

In addition to explicitly modeling the dependence between forward and reverse flows, the model we develop in this section also explicitly describes the relationship between the packet and byte flows. This relationship can be leveraged in predicting either packet or byte traffic levels on unobserved links. In practice, however, predictions of the byte mode is most useful for network administration purposes. Using the two traffic modes (bytes and packets) is similar in spirit to the idea of *co–kriging* in spatial statistics, as described in [6].

We first introduce the new model, called the subflow model. We then discuss the practical aspects of using this model for kriging, and finish by discussing some of the details of the model. The evaluation of its performance is reserved for the following chapter, where it is compared to the model from the previous chapter.

5.1 A Mechanistic Byte–Packet Subflow Model

In this section, we build a model for the joint behavior of the packet and byte flows across the entire network. In this model, we incorporate several of the important features discussed in Chapter II. Specifically, we try to capture the *TCP acknowledgment behavior*, and the *packet size distribution* of the protocols described in Section 2.3. When combined with the fact that TCP and UDP carry roughly 98% of the data, these two features allow us to achieve a mean/variance relationship in a principled way, based on the engineered behavior of the network, rather than using an empirical relationship as we did in Chapter IV.

We assume a subflow model for each of the source/destination flows. For each flow j, denote by j^r the *reverse* flow. We assume the TCP mechanism exists; namely that when TCP packets are sent on the forward flow j, ACK packets are sent along the reverse flow j^r. This creates dependence between the forward and reverse flows. The dependence is modeled in the covariance implied by the following model. As in Chapter IV, we again assume that the time scale of measurement is small as compared to the round trip time of the network.

We further assume (without loss of generality) that the $\mathcal{J}$ flows are *ordered* so that the forward and reverse flows are consecutive entries in the appropriate $X(t)$ vector. We assume that each flow is composed of a set of *subflows*, and the total traffic carried by a flow at a given time is simply the sum of the traffic carried by the individual subflows. Each subflow caries packets of a *constant* size and is utilized by a single protocol (UDP or TCP). In particular, assume that there are two TCP subflows (the data subflow and the ACK subflow) and K UDP subflows. Let c_A denote the (constant) size of ACK packets, c_T be the constant packet size of

TCP packets, and $c_{U,k}, \forall k \in \{1,\ldots,K\}$ be the constant packet sizes of the K different UDP subflows. Note that k should be relatively small as compared to the total number of packet sizes for parsimony. In our analysis, we take $K = 3$ to approximate ≈ 1500 possible packet sizes. We assume these constants are all known, although we discuss the process of choosing them appropriately via the auxiliary data in Section 5.3.1 below. This then allows us to express the number of packets on flow j as:

$$X_j^P(t) = X_j^T(t) + X_{j^r}^T(t) + \sum_{k=1}^{K} X_{j,k}^U(t), \quad j = 1,\ldots,\mathcal{J} \tag{5.1}$$

Similarly, we may express the bytes on flow j as

$$X_j^B(t) = c_T X_j^T(t) + c_A X_{j^r}^T(t) + \sum_{k=1}^{K} c_{U,k} X_{j,k}^U(t), \quad j = 1,\ldots,\mathcal{J} \tag{5.2}$$

These expressions match our intuition of the situation. Each subflow contributes a certain number of packets to the overall packet and a certain number of bytes to the byte total, where the contribution is equal to the packet size times the number of packets. Furthermore, this model incorporates two novel and important features. By including the TCP acknowledgment mechanism, the dependence between forward and reverse flows is accounted for. Secondly, since both modes are functions of the number of packets in each subflow, the relationship of the two modes is described by these equations.

Thus far, we have considered the mechanistic aspects of theses subflows. We now consider an appropriate statistical model for these subflows. In contrast to the assumptions made earlier in this work, the individual subflows are allowed to have a relatively small number packets, a Poisson model is more appropriate

than the normal approximation, as used above. However, empirical evidence, discussed below in Section 5.3.2, suggests that network data is *overdispersed*, and thus requires a model for count data that reflects this important property. We therefore assume that each subflow has a **Poisson** distribution, conditional on a certain parameter $\lambda_j^k(t)$, which is allowed to vary across each subflow. That is,

$$\begin{aligned} X_j^T|\lambda_j^T &\sim Pois(\lambda_j^T) \qquad \forall j = \{1,\dots,\mathcal{J}\} \\ X_{j,k}^U|\lambda_{j,k}^U &\sim Pois(\lambda_{j,k}^U) \qquad \forall k = \{1,\dots,K\}, \quad j = \{1,\dots,\mathcal{J}\} \end{aligned}$$

We assume that the λ's have a gamma distribution, with a shape parameter α_j that depends on each subflow, and the scale parameter θ which is the same across all subflows.

$$\begin{aligned} \lambda_j^T &\sim \Gamma(\alpha_j^T, \theta) \qquad \forall j = \{1,\dots,\mathcal{J}\} \\ \lambda_{j,k}^U &\sim \Gamma(\alpha_{j,k}^U, \theta) \qquad \forall k = \{1,\dots,K\}, \quad j = \{1,\dots,\mathcal{J}\} \end{aligned}$$

We can then see that for a given subflow, the mean and variance may be given by:

$$\mathbb{E}(X_j^T) = \mathbb{E}\Big(\mathbb{E}(X_j^T|\lambda_j^T)\Big) = \alpha_j^T\theta \tag{5.3}$$

and

$$\begin{aligned} \text{Var}(X_j^T) &= \mathbb{E}\Big(\text{Var}(X_j^T|\lambda_j^T)\Big) + \text{Var}\Big(\mathbb{E}(X_j^T|\lambda_j^T)\Big) = \mathbb{E}(\lambda_j^T) + \text{Var}(\lambda_j^T) \\ &= \alpha_j^T\theta + \alpha_j^T\theta^2 = \alpha_j^T\theta(1+\theta) \end{aligned} \tag{5.4}$$

This allows us to calculate the mean and variance of each flow, and the covariance between any pair of flows:

$$\mathbb{E}X_j^P(t) = \theta\left(\alpha_j^T + \alpha_{j^r}^T + \sum_{k=1}^{K}\alpha_{j,k}^U\right) \tag{5.5}$$

$$\mathbb{E}X_j^B(t) = \theta\left(c_T\alpha_j^T + c_A\alpha_{j^r}^T + \sum_{k=1}^{K}c_{U,k}\alpha_{j,k}^U\right) \tag{5.6}$$

$$\mathrm{Var}(X_j^P(t)) = \theta(1+\theta)\left(\alpha_j^T + \alpha_{j^r}^T + \sum_{k=1}^{K}\alpha_{j,k}^U\right) \tag{5.7}$$

$$\mathrm{Var}(X_j^B(t)) = \theta(1+\theta)\left(c_T^2\alpha_j^T + c_A^2\alpha_{j^r}^T + \sum_{k=1}^{K}c_{U,k}^2\alpha_{j,k}^U\right) \tag{5.8}$$

$$\mathrm{Cov}(X_j^P(t), X_j^B(t)) = \theta(1+\theta)\left(c_T\alpha_j^T + c_A\alpha_{j^r}^T + \sum_{k=1}^{K}c_{U,k}\alpha_{j,k}^U\right) \tag{5.9}$$

$$\mathrm{Cov}(X_j^P(t), X_{j^r}^P(t)) = \theta(1+\theta)\left(\alpha_{j^r}^T + \alpha_j^T\right) \tag{5.10}$$

$$\mathrm{Cov}(X_j^B(t), X_{j^r}^B(t)) = \theta(1+\theta)\left(c_Ac_T(\alpha_{j^r}^T + \alpha_j^T)\right) \tag{5.11}$$

$$\mathrm{Cov}(X_j^P(t), X_{j^r}^B(t)) = \theta(1+\theta)\left(c_T\alpha_{j^r}^T + c_A\alpha_j^T\right) \tag{5.12}$$

$$\mathrm{Cov}(X_j^B(t), X_{j^r}^P(t)) = \theta(1+\theta)\left(c_A\alpha_{j^r}^T + c_T\alpha_j^T\right) \tag{5.13}$$

with all other covariance terms zero due to independence

For convenience, this may be re–parametrized via:

$$\vec{\nu} = \vec{\alpha}\theta \qquad \text{and} \qquad \phi = 1+\theta \tag{5.14}$$

which allows for an interpretation of the parameters, with $\vec{\nu}$ representing the means of the subflows and ϕ representing the variance scaling factor. Using this alternative representation, the relationships above may be re–written as:

$$\mathbb{E}X_j^P(t) = \nu_j^T + \nu_{j^r}^T + \sum_{k=1}^{K} \nu_{j,k}^U \tag{5.15}$$

$$\mathbb{E}X_j^B(t) = c_T \nu_j^T + c_A \nu_{j^r}^T + \sum_{k=1}^{K} c_{U,k} \nu_{j,k}^U \tag{5.16}$$

$$\mathrm{Cov}(X_j^P(t), X_j^P(t)) = \nu_j^T + \nu_{j^r}^T + \sum_{k=1}^{K} \nu_{j,k}^U \tag{5.17}$$

$$\mathrm{Cov}(X_j^B(t), X_j^B(t)) = \phi\left(c_T^2 \nu_j^T + c_A^2 \nu_{j^r}^T + \sum_{k=1}^{K} c_{U,k}^2 \nu_{j,k}^U \right) \tag{5.18}$$

$$\mathrm{Cov}(X_j^P(t), X_j^B(t)) = \phi\left(c_T \nu_j^T + c_A \nu_{j^r}^T + \sum_{k=1}^{K} c_{U,k} \nu_{j,k}^U \right) \tag{5.19}$$

$$\mathrm{Cov}(X_j^P(t), X_{j^r}^P(t)) = \phi\left(\nu_{j^r}^T + \nu_j^T \right) \tag{5.20}$$

$$\mathrm{Cov}(X_j^B(t), X_{j^r}^B(t)) = \phi\left(c_A c_T (\nu_{j^r}^T + \nu_j^T) \right) \tag{5.21}$$

$$\mathrm{Cov}(X_j^P(t), X_{j^r}^B(t)) = \phi\left(c_T \nu_{j^r}^T + c_A \nu_j^T \right) \tag{5.22}$$

$$\mathrm{Cov}(X_j^B(t), X_{j^r}^P(t)) = \phi\left(c_A \nu_{j^r}^T + c_T \nu_j^T r \right) \tag{5.23}$$

While somewhat cumbersome to repeat, this representation is instructive in that these relationships are as they would have been if we had chosen a simple Poisson model for each flow, with one major exception: each entry of the covariance matrix is multiplied by a scaling factor ϕ, allowing overdispersion.

Thus the joint distribution of $\vec{X}$ and consequently $\vec{Y}$ (through the routing equation (2.2)) are determined by $\vec{c}$, $\vec{\alpha}$, θ (or, alternatively, $\vec{c}$, $\vec{\alpha}$, θ). While these parameters are, in principle, unknown, we consider a framework similar to that in Chapter IV, where we differentiate between the so–called *model parameters* which are learned via auxiliary data, and the parameters that are estimated using the observed link level data. In this case, we will consider $\vec{c}$ to be a be a parameter learned offline, θ or ϕ to be learned online, and $\vec{\alpha}$ or $\vec{\nu}$ to be modeled in the spirit

of the means model in Chapter IV, as a linear combination of a small number of basis vectors, where the basis vectors are learned offline, and the combination fit via observed data. We discuss this process in the next Section 5.2.

5.2 Multi–Modal Kriging

The model defined in Section 5.1 states that $X(t)$ and $Y(t)$ are conditionally Poisson random variables. In backbone networks, the entries of $\vec{\nu}(t)$ are sufficiently large so that $X(t)$ may be well approximated by a multivariate Normal distribution:

$$X(t) \sim N\left(\mu_X(\vec{\nu}(t), \phi), \Sigma_X(\vec{\nu}(t), \phi)\right) \tag{5.24}$$

where μ_X and Σ_X as functions of $\vec{\nu}(t)$ and ϕ are given in (5.15)–(5.16) and (5.17)–(5.23), respectively. Observe that μ_X and Σ_X are linear functions of $\nu(t)$. With this approximation, we have:

$$Y(t) \sim N\left(\mu_Y, \Sigma_Y\right) \tag{5.25}$$

where $\mu_Y = \tilde{A}\mu_X(\vec{\nu}(t), \phi)$ and $\Sigma_Y = \tilde{A}\Sigma_X(\vec{\nu}(t), \phi)\tilde{A}^T$, with

$$\tilde{A} := \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix},$$

where A denotes the routing matrix.

Using the above approximation, we consider the *Kriging Problem* from Section 4.1. With the above approximation, the Kriging Problem may be solved as before, if $\vec{\nu}(t)$ and ϕ are *known*. Given $\vec{\nu}(t)$ and ϕ, the best linear unbiased predictor for $Y_u = Y_u(t)$ in terms of $Y_o = Y_o(t)$ is then given by:

$$\widehat{Y_u}(t) = \mu_u + \Sigma_{uo}\Sigma_{oo}^{-1}(Y_o(t) - \mu_o), \tag{5.26}$$

where

$$Y(t) = \begin{pmatrix} Y_o \\ \\ Y_u \end{pmatrix},\ \mu_Y(\vec{v}(t), \phi) = \begin{pmatrix} \mu_o \\ \\ \mu_u \end{pmatrix},\ \text{and}\ \ \Sigma_Y(\vec{v}(t), \phi) = \begin{pmatrix} \Sigma_{oo} & \Sigma_{ou} \\ \\ \Sigma_{uo} & \Sigma_{uu} \end{pmatrix}.$$

are partitions of the traffic vector, mean vector, and covariance matrix, respectively.

Remark V.1. In contrast with the approach proposed in Chapter IV, we have arrived at the model in (5.26) in a more mechanistic way by incorporating the packet–byte relationship as well as the TCP feedback mechanism.

As before, the difficulties lie in estimating the unknown parameters of the distribution of $Y(t)$. Under the subflow model, both μ_Y and Σ_Y are more structured than in the previous chapter, however, both depend on the unknown $\vec{v}(t)$ and ϕ. The vector of subflow means, $\vec{v}(t)$, is of length $((K+1) \times \mathcal{J})$, which is larger than the number of observed links, so we attempt a dimension reduction technique as before to estimate $\vec{v}$. This parallels our construction of basis vectors to model traffic means in Chapter IV.

First, we construct locally smoothed estimates of the subflow means $\vec{v}$ via the Netflow data. This includes a series of estimates for each subflow of each flow, or $((K+1) \times \mathcal{J})$ smoothed estimates of $\vec{v}$. We then let the matrix V be the first p principle components of the matrix formed by joining the estimates of $\vec{v}$. We then model $\vec{v}(t)$ as a linear combination of these basis vectors:

$$\vec{v}(t) = V\vec{\beta}(t) \tag{5.27}$$

Without loss of generality, assume that $K = 3$. Then we see that:

$$\mu_X^P = \left(\mathbb{I}_{J/2} \otimes \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \right) \vec{\nu} =: \Psi \vec{\nu} \tag{5.28}$$

where $\otimes$ denotes the Kronecker product. Similarly,

$$\mu_X^B = \left(\mathbb{I}_{J/2} \otimes \begin{bmatrix} c_T & c_{U,1} & c_{U,2} & c_{U,3} & c_A & 0 & 0 & 0 \\ c_A & 0 & 0 & 0 & c_T & c_{U,1} & c_{U,2} & c_{U,3} \end{bmatrix} \right) \vec{\nu} =: \Phi \vec{\nu} \tag{5.29}$$

We can then write:

$$\mu_X^P = \Psi V \beta \quad \text{and} \quad \mu_X^B = \Phi V \beta$$

We see that under the subflow model,

$$\vec{\mu}_X := \begin{pmatrix} \mu_X^P \\ \mu_X^B \end{pmatrix} = \begin{pmatrix} \Psi V \\ \Phi V \end{pmatrix} \beta \tag{5.30}$$

In view of the routing equation (2.2) we have:

$$\vec{\mu}_Y := \begin{pmatrix} \mu_Y^P \\ \mu_Y^B \end{pmatrix} = \begin{pmatrix} A\Psi V \\ A\Phi V \end{pmatrix} \beta \tag{5.31}$$

Defining the matrix

$$M := \begin{bmatrix} A\Psi V \\ A\Phi V \end{bmatrix}, \tag{5.32}$$

then we can write $\vec{\mu}_Y = M\beta$. This then allows one to estimate β using the observed set of links (whether bytes or packets), since

$$\overline{Y}_O = M_O \beta$$

In principle, we can then estimate β via OLS. If OLS is insufficient, we may use it as an initial estimate of β to estimate $\vec{v}$ which in turn allows us to calculate the covariance matrix, and proceed with iterated GLS. In practice, this algorithm has the potential to converge to a value of β that results in *negative* values for some of the entries of $\vec{v}$. This is not consistent with the model, since both Poisson and traffic means must be positive. This can be avoided with an iGLS procedure where at each step the resulting $\widehat{\beta}$ is constrained via quadratic programming to satisfy $V\widehat{\beta} = \vec{v} \geq 0$.

Once an estimate of β has been generated, it is straightforward to estimate the overdispersion parameter ϕ. This is accomplished by defining the quantity

$$\tilde{\Sigma}_Y(\nu) := \frac{1}{\phi}\Sigma_Y(\nu(t), \phi) \tag{5.33}$$

where $\Sigma_Y(\nu(t), \phi)$ is the covariance matrix as defined by above. Note that $\tilde{\Sigma}_Y$ does not depend on ϕ, and that:

$$\Sigma_Y(\nu, \phi) = \phi\tilde{\Sigma}_Y(\nu)$$

We then vectorize the sample covariance matrix of Y_o and the observed part of $\tilde{\Sigma}_Y(\nu)$. It is then straightforward to estimate ϕ via OLS.

Once estimates for β and ϕ are obtained, they define estimates of μ_Y and Σ_Y, which may then be used in the kriging equations. The results of kriging under this model are presented in Chapter VI. The remainder of this chapter further explores some details of this model.

5.3 Model Design and Validation

In this section, we discuss the problem of appropriately choosing the constant packet sizes in the subflow model, and the need for a model that accounts for the

overdispersion present in the data.

5.3.1 Model Design

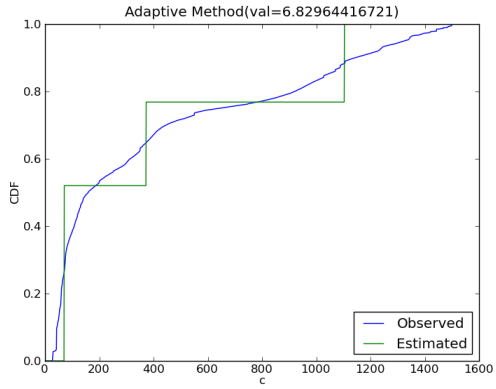
In Section 5.1, we assume that we know the value of $\vec{c}$. In practice, it is easy to choose values of c_A and c_T as the smallest and largest possible packet sizes, respectively. However, we need to choose a small number of packet sizes to closely approximate the packet–size distribution of the UDP subflows. In fact, we wish to approximate *optimally* a cumulative density function, denoted $F(x)$, with a discrete distribution having mass at k pre–specified values of x: $c_1,\ldots,c_k$. We might assume $F(x)$ is continuous, or we might simply assume a discrete distribution where the number of values having non–zero probability is much larger than k. In practice, one does not observe the cdf, but rather the empirical cdf, denoted $\widehat{F}(x)$. However, we study the problem of approximating the empirical cdf with k points.

We will approximate the cdf, $F(x)$ via the function:

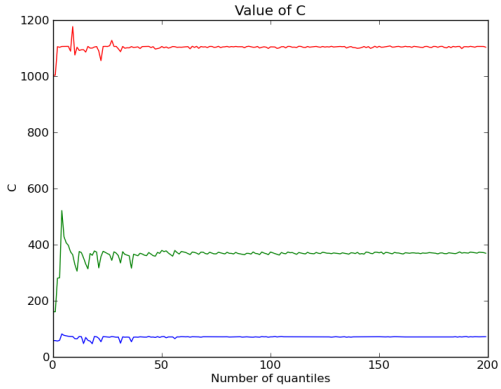
$$\tilde{F}(x) := \sum_{k=1}^{K} p_k \mathbb{I}_{\{c_k<x\}} \tag{5.34}$$

We approach this problem via a two–stage process:

1. For a fixed $\vec{c} := c_1,\ldots,c_k$, use a method to estimate the vector $\vec{p} := p_1,\ldots,p_k$. Some different approaches are discussed in the following.
2. Use an optimization routine, such as the Nelder-Mead algorithm, to search over the space of possible values of $\vec{c}$. For each $\vec{c}$, we calculate the value of $\vec{p}$ and then calculate the overall error between the empirical cdf and the estimated cdf in (5.34). This overall error is minimized to give the appropriate choice of $\vec{c}$.



(a) Estimated CDF



(b) Estimates of $\vec{c}$ as a function of number of quantiles.

In the first step, we assume the vector $\vec{c}$ is given, and are trying to find the best choice of $\vec{p}$. This is done using the quantiles of the empirical cdf, and a linear model. We chose n observed quantiles, and solve the following for $\vec{p}$.

$$\begin{pmatrix} \widehat{F}(x_1) \\ \widehat{F}(x_2) \\ \vdots \\ \widehat{F}(x_n) \end{pmatrix} = \begin{bmatrix} \mathbb{I}_{\{c_1 \le x_1\}} & \mathbb{I}_{\{c_2 \le x_1\}} & \cdots & \mathbb{I}_{\{c_k \le x_1\}} \\ \mathbb{I}_{\{c_1 \le x_2\}} & \mathbb{I}_{\{c_2 \le x_2\}} & \cdots & \mathbb{I}_{\{c_k \le x_2\}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{I}_{\{c_1 \le x_n\}} & \mathbb{I}_{\{c_2 \le x_n\}} & \cdots & \mathbb{I}_{\{c_k \le x_n\}} \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{pmatrix}$$

Solving the above system via OLS is not guaranteed to arrive at a valid solution, since we have the $0 \le p_i \le 1 \forall i \in 1,\ldots,k$ and $\sum p_i = 1$. We can enforce the constraints by solving this system via a quadratic program.

In the second step, we need to define an objective function. Some candidates include the sum of the L1 and L2 differences between $\widehat{F}(x)$ and $\tilde{F}(x)$. In practice, we choose the L2 difference:

$$\int |\widehat{F}(x) - \tilde{F}(x)|^2 dx \tag{5.35}$$

We then minimize this function via the Nelder–Mead algorithm to arrive at a choice of $\vec{c}$ that is close to the empirical cdf.

Many different applications utilize the UDP protocol. The three values of c allow us to split the traffic into different application usage: small packets for short messages, large packets representing mostly data transmission, and medium sized packets, representing other applications. However, we must remember that, in principle, whatever model we choose must be fit with a small number of observed link measurements, so we must choose a number of c values that walks the fine line between flexibility to describe the data and parsimony in the model.

5.3.2 Overdispersion of Data

In this section, we briefly illustrate the need of an overdispersed model for the individual subflows. Since the parameter ϕ cancels out in the prediction formula, it does not directly effect the prediction error. In order to see the necessity for the overdispersed model, it is necessary to examine the prediction intervals. Figure 5.1 illustrates the Poisson model, while Figure 5.2 shows the results of the prediction intervals with the overdispersed model. These results illustrate that the variance present in the data is too high to be adequately described by a Poisson model.

5.4 Calibration and Model Misspecification

In this section we evaluate the effect of model misspecification on the performance of the subflow model. In particular, we examine the way in which the number of UDP subflows and the misspecification of packet sizes affect the quality of the prediction. We also explore the utility of a model developed using auxiliary data recorded on a particular day in carrying out prediction on other days.

Throughout this section, we evaluate the performance of the model via what we refer to as the *single–station network monitoring* problem. For a given router, we

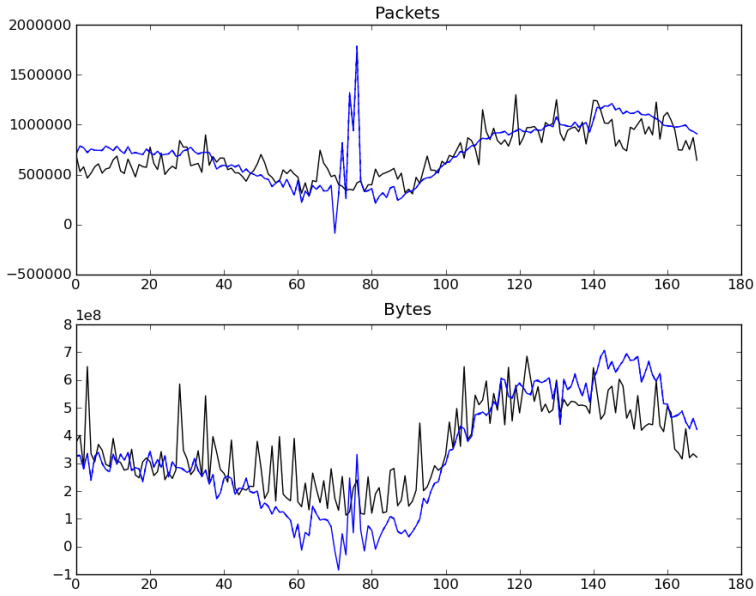


Figure 5.1: Prediction of traffic with prediction intervals for a pure Poisson model,which is equivalent to the subflow model with $\phi = 1$. The prediction intervals are too small to be distinguished from the prediction itself. The variability in the data, is clearly much larger.

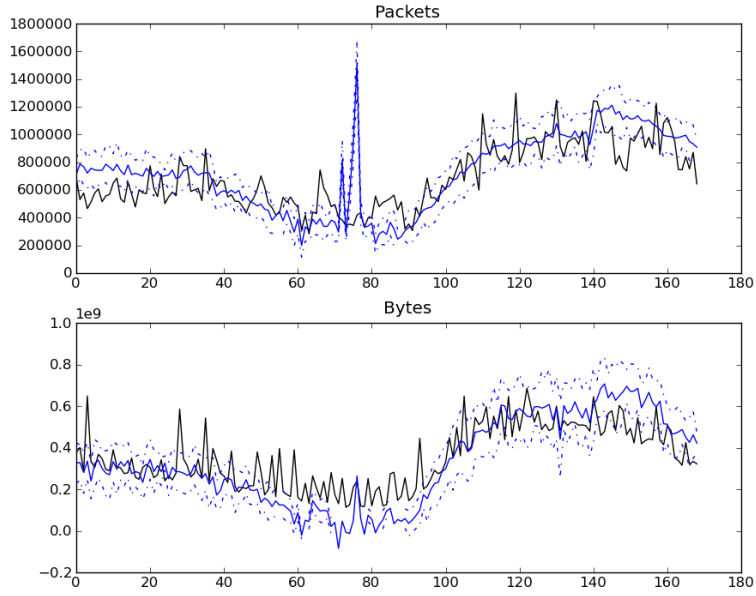


Figure 5.2: Prediction of traffic with prediction intervals for the overdispersed model, with ϕ estimated from the data. Although the coverage is not perfect, the coverage is much more reasonable.

take as observed all the links that connect directly to the router, and predict the traffic levels on the remaining links in the network. These scenarios correspond to monitoring the traffic on the *entire* network using a combination of direct observation and multi–modal kriging. We evaluate the performance at each router via:

$$\mathrm{ReMSE}_2(\widehat{Y}) := \frac{1}{T}\sum_{t=1}^{T}\frac{\|\hat{Y}(t) - Y(t)\|^2}{\|Y(t)\|^2}$$

For further details on the single–station monitoring problem, see the discussion in Chapter VI.

5.4.1 Number of UDP Subflows

In the above derivation, we have assumed that the model consists of one TCP subflow and $K = 3$ UDP subflows, as well as the ACK subflow induced by the TCP mechanism on the reverse flow. In principle, however, we may choose any number K of UDP subflows. In Table 5.1, we explore other possible values of K, namely, $K = 1$ and $K = 5$. In the table, we see that the use of a single UDP subflow is not particularly robust, as there are a few routers for which the prediction is quite poor. With these exceptions, increasing the number of flows sometimes results in marginally better prediction, although in some situations a smaller number of UDP subflows performs better. Choosing $K = 3$ appears to be a reasonable compromise between model flexibility and parsimony.

5.4.2 Misspecification of Subflow Packet Size

The model we propose uses known constant packet sizes for each of the subflows in the model. In Section 5.3.1, we discuss a means of selecting the packet sizes for the UDP subflows using auxiliary data. The size of the TCP payload and acknowledgment packets, on the other hand, are known based on the design of

	Packets:			Bytes:		
Number of UDP Subflows:	1	3	5	1	3	5
ATLA	0.174	0.154	0.155	0.149	0.166	0.162
CHIC	0.320	0.308	0.319	0.190	0.217	0.228
HOUS	0.166	0.154	0.156	0.122	0.122	0.121
KANS	0.226	0.225	0.230	0.132	0.143	0.147
LOSA	3750.969	0.166	0.170	2643.996	0.131	0.131
NEWY	0.182	0.201	0.194	0.162	0.217	0.212
SALT	73383.537	0.230	0.239	106791.485	0.151	0.158
SEAT	0.252	0.257	0.257	0.153	0.173	0.175
WASH	0.152	0.140	0.139	0.188	0.176	0.173

Table 5.1: The effect of Number of UDP subflows used in the mode, reported in ReMSE_2, for predicting both packets and bytes. Note that the performance of a single subflow is somewhat unstable. Furthermore, the change from 3 to 5 subflows does not improve prediction substantially. In fact, in some cases it is marginally worse.

the protocol. In this section, we estimate the effect of misspecifying the packet sizes for the acknowledgment, TCP payload, and UDP packet sizes.

In Table 5.2, we investigate specification of the ACK packet size. We see that prediction of the packet mode is generally robust, although there is an increase for large values. Predictions of the byte mode, on the other hand, are far less robust, and become notably worse as the packet size is larger.

Table 5.4 examines the misspecification of the TCP payload size. As with the ACK packet size, the effect is more noticeable when predicting byte traffic.

Finally, we examine the effect of misspecifying the size of UDP in Table 5.4. We misspecify the size of UDP subflows. The model seems more robust to this type of misspecification, which agrees with our intuition, since the UDP protocol accounts for a much smaller fraction of traffic than TCP.

5.4.3 Robustness of Model on Other Days

In Chapter IV, we demonstrated that the model proposed therein was robust to different days, in the sense that a model (the matrix F) learned on one day could be used to predict traffic on other days without increasing the error substantially.

Packets:						
Packet Size:	40	0	1	80	400	1500
ATLA	0.191	0.202	0.201	0.182	0.156	0.168
CHIC	0.166	0.143	0.143	0.198	0.512	0.209
HOUS	0.126	0.128	0.128	0.124	0.140	0.138
KANS	0.133	0.126	0.126	0.141	0.239	0.152
LOSA	0.135	0.138	0.138	0.133	0.142	0.136
NEWY	0.260	0.277	0.276	0.244	0.170	0.203
SALT	0.131	0.129	0.129	0.134	0.198	0.191
SEAT	0.163	0.158	0.158	0.169	0.228	0.199
WASH	0.191	0.210	0.210	0.174	0.122	0.120
Bytes:						
Packet Size:	40	0	1	80	400	1500
ATLA	0.153	0.151	0.151	0.155	0.196	0.840
CHIC	0.244	0.220	0.221	0.273	0.530	0.198
HOUS	0.150	0.150	0.150	0.150	0.169	0.369
KANS	0.203	0.198	0.198	0.209	0.269	0.266
LOSA	0.151	0.151	0.151	0.151	0.176	0.546
NEWY	0.174	0.173	0.173	0.175	0.205	1.108
SALT	0.174	0.173	0.173	0.177	0.229	0.275
SEAT	0.242	0.244	0.244	0.241	0.255	0.290
WASH	0.140	0.140	0.140	0.141	0.154	0.460

Table 5.2: Misspecification of ACK packet size. Comparing the correct packet size of 40 to various misspecified packet sizes. Notice that misspecification much more seriously impacts the prediction of the byte traffic.

We attempt a similar analysis with the subflow model.

Table 5.5 shows the results of this analysis. It is clear that this model is less robust to changes in the underlying structure than the model explored in Chapter IV. However, with the exception of certain routers (New York and Washington), the model seems to achieve a consistent bound in most cases that is useful for the coarser level network tasks.

Intuitively, we expect the subflow model to be more sensitive to changes in network structure, since here we include finer structural detail in the model, considering the protocol level subflows making up the origin–destination flows rather than considering the aggregate flows themselves. Furthermore, this sensitivity to changes in the network is potentially useful, since it could, in principle, allow anomaly detection methods based on the subflow model to detect more subtle

	Packets:					Bytes:				
Packet Size	1500	50	200	700	1200	1500	50	200	700	1200
ATLA	0.191	0.339	0.375	0.511	0.348	0.153	0.803	0.636	0.223	0.138
CHIC	0.166	0.255	0.315	0.471	0.287	0.244	0.827	0.666	0.239	0.125
HOUS	0.126	0.250	0.263	0.297	0.186	0.150	0.816	0.663	0.285	0.157
KANS	0.133	0.189	0.222	0.299	0.166	0.203	0.837	0.689	0.286	0.151
LOSA	0.135	0.227	0.248	0.314	0.197	0.151	0.839	0.690	0.297	0.159
NEWY	0.260	0.575	0.614	0.793	0.469	0.174	0.771	0.582	0.177	0.152
SALT	0.131	0.134	0.139	0.149	0.132	0.174	0.857	0.736	0.384	0.193
SEAT	0.163	0.150	0.151	0.147	0.139	0.242	0.866	0.757	0.441	0.278
WASH	0.191	0.408	0.435	0.591	0.442	0.140	0.789	0.614	0.202	0.129

Table 5.3: Misspecification of TCP payload packet size. The correct size of 1500 is compared to a range of misspecified values. As with the ACK packet size, the misspecification most seriously impacts the prediction of byte traffic.

	Packets:				Bytes:			
Packet Size:	Opt.	700	20	1200	Opt.	700	20	1200
ATLA	0.191	0.169	0.216	0.163	0.153	0.152	0.160	0.154
CHIC	0.166	0.233	0.136	0.331	0.244	0.304	0.223	0.373
HOUS	0.126	0.124	0.138	0.134	0.150	0.155	0.155	0.165
KANS	0.133	0.144	0.124	0.166	0.203	0.203	0.207	0.209
LOSA	0.135	0.132	0.145	0.132	0.151	0.147	0.160	0.141
NEWY	0.260	0.199	0.300	0.167	0.174	0.158	0.185	0.152
SALT	0.131	0.135	0.131	0.142	0.174	0.172	0.176	0.165
SEAT	0.163	0.167	0.159	0.169	0.242	0.231	0.259	0.212
WASH	0.191	0.144	0.231	0.132	0.140	0.141	0.150	0.158

Table 5.4: Misspecification of UDP Packet Size. The "Opt" column indicates the error when we assign packet sizes based on the results from Section 5.3.1, while the other columns indicate the misspecified size of all three subflows. We see that the model is more robust to misspecification of this type than it is to the

changes in the network structure, or detect anomalies faster.

Packets:						
	2-19-2009	2-18-2009	2-26-2009	3-12-2009	3-17-2009	4-19-2009
ATLA	0.191	0.237	0.303	0.564	0.456	0.436
CHIC	0.166	0.172	0.235	0.303	0.293	0.301
HOUS	0.126	0.156	0.186	0.481	0.367	0.329
KANS	0.133	0.138	0.180	0.275	0.281	0.308
LOSA	0.135	0.161	0.224	0.425	0.374	0.328
NEWY	0.260	0.374	0.420	0.809	0.919	1.181
SALT	0.131	0.143	0.184	0.248	0.259	0.282
SEAT	0.163	0.162	0.220	0.301	0.284	0.339
WASH	0.191	0.206	0.243	0.393	0.406	0.501
Bytes:						
	2-19-2009	2-18-2009	2-26-2009	3-12-2009	3-17-2009	4-19-2009
ATLA	0.153	0.216	0.288	0.497	0.392	0.424
CHIC	0.244	0.261	0.349	0.306	0.338	0.383
HOUS	0.150	0.184	0.247	0.432	0.344	0.377
KANS	0.203	0.211	0.283	0.272	0.319	0.391
LOSA	0.151	0.190	0.263	0.389	0.350	0.387
NEWY	0.174	0.286	0.350	0.806	0.776	0.878
SALT	0.174	0.186	0.257	0.242	0.291	0.354
SEAT	0.242	0.245	0.333	0.312	0.330	0.443
WASH	0.140	0.191	0.267	0.351	0.348	0.443

Table 5.5: Examining the robustness of the subflow model to different days. The subflow model was used to learn the model via auxiliary data collected on 02-19-2009, and used to predict traffic on several other days. Although there is an increase in error as we move from the original day, most routers seem to reach a reasonable steady state.

CHAPTER VI

Discussion, Comparison of Methods, and Analysis of Real Data

In this chapter, we examine the performance of the subflow model introduced in the previous chapter, and compare its relative strengths and weaknesses to the model described in earlier chapters. We first introduce the idea of prediction scenarios, and then evaluate the two models based on these scenarios.

6.1 Evaluation Criteria

To compare the performance of the two models, we use two different criteria. In the first, we predict a single link using a subset of observed links.

Prediction Scenarios: In this case, estimators are compared for 9 different prediction scenarios (see Table 6.1). In each case one link is treated as unobserved, and a subset of the remaining links are used as predictors. The choice of the unobserved/observed links is not arbitrary. Recall that the traffic on any single link is equal to the sum of the traffic of the *O/D flows* that utilize the link. The three unobserved links were chosen to represent a range in the utilization levels (in terms of number of flows): Link 7, predicted in scenarios 1-3, is used by a *medium* number of flows; Link 13, in cases 5-7, is used by 14 flows, the *most* of any link; finally, Link 19 is utilized by a *small* number of flows.

The links used as predictors were also chosen based on the number of origin–

Scenario	Predicted	Observed Links
1	7	2,12
2	7	2,12,13,15
3	7	2,12,13,15,23,25
4	13	3,7
5	13	3,9
6	13	3,9,12
7	13	3,7,9,12,17,19,21
8	19	3,9,13
9	19	2,3,9,12,15,21,23,25

Table 6.1: Description of nine Kriging scenarios used to evaluate the model. The choice of predictors is based on the number of shared traffic flows. The link id's are given in Table 2.1.

destination flows shared with the unobserved link. Namely, for each unobserved link, the predictors were selected so that they share at least one origin–destination flow with the unobserved link. However, within a prediction scenario, multiple observed links may carry the same OD flow.

While using the prediction scenario criteria, we are looking at how well a single link is estimated, and we use the following natural measure to determine the quality of a prediction. This measure can be interpreted as average relative error over the network.

$$\mathrm{ReMSE}_1(\widehat{Y}) = \sum_{t=1}^{T} \|\widehat{Y}(t) - Y(t)\|^2 / \sum_{t=1}^{T} \|Y(t)\|^2. \tag{6.1}$$

Single Router Network Monitoring: In addition to the set of scenarios described above, we also consider the idea of monitoring the entire network from a single router. In this scenario, all the links connected to a given router are directly observed, while all other links are predicted. This is a natural problem, as it allows the monitoring of the entire network without expending any bandwidth to transmit information about network usage. In our analysis, we do this for all nine routers making up Internet2.

Scenario	Packet Traffic		Byte Traffic	
	SF Mod	PCA Mod	SF Mod	PCA Mod
1	0.3708	0.1488	0.08	0.4692
2	0.1201	0.0219	0.0752	0.0692
3	0.1473	0.0348	0.0682	0.1113
4	0.0433	0.1623	0.0888	0.147
5	0.037	0.085	0.0312	0.5011
6	0.0423	0.0749	0.0308	0.0992
7	0.037	0.0328	0.0349	0.0439
8	0.11	0.2543	0.1365	7.5838
9	0.1111	0.2063	0.1382	2.0336

Table 6.2: Results comparing the results of the Subflow model with the previous model across the 9 prediction scenarios introduced in Table 6.1. The lowest value is highlighted. While the two methods are comparable in terms of predicting the packet mode, the Subflow model has a clear advantage in terms of predicting the byte mode.

Since this goal is more global in nature, we must consider a different measure of success. We use the second version of relative mean square error as a global measure of prediction success.

$$\text{ReMSE}_2(\widehat{Y}) := \frac{1}{T}\sum_{t=1}^{T}\frac{\|\hat{Y}(t) - Y(t)\|^2}{\|Y(t)\|^2} \tag{6.2}$$

These two measures are used as appropriate in the remainder of this chapter to compare the performance of network kriging under these two models.

6.2 Comparison

Here, we compare the performance of the two models we have developed by evaluating their performance in terms of both the prediction scenarios and the network monitoring scenarios.

Scenario Prediction Perspective: Results for all the prediction scenarios are shown in Table 6.2. Further details for some of the prediction scenarios are depicted in Figure 6.3, which shows the observed and predicted traffic traces under different models for one of the prediction scenarios. It is clear that in general, the

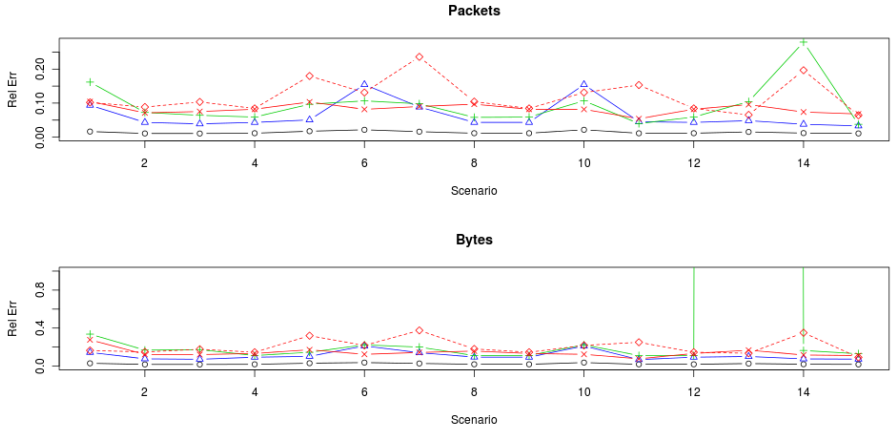


Figure 6.1: Comparing the Subflow model to the PCA model for the original scenarios, in terms of predicting packets or bytes.

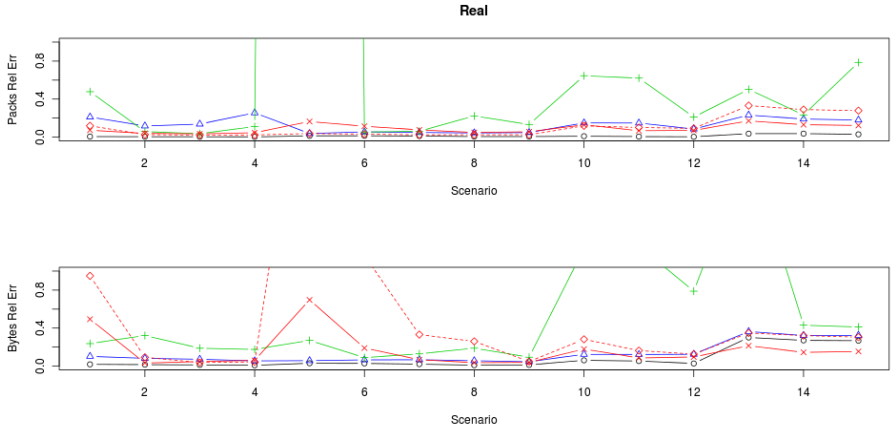


Figure 6.2: Comparing the Subflow model to the PCA model for the *reverse* of the observed links from the original scenarios, in terms of predicting packets or bytes. Because the Subflow models explicitly models the forward/reverse relationship, we expect it to perform better.

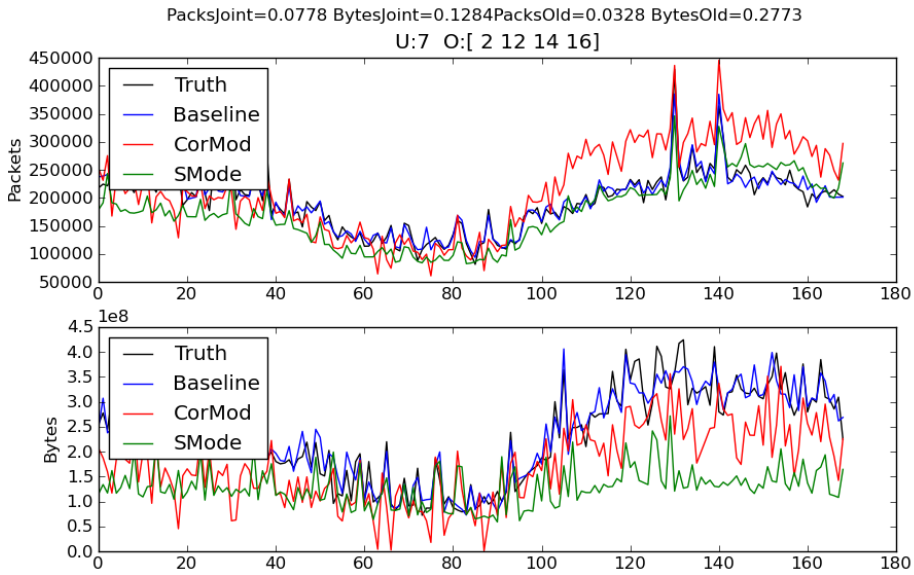


Figure 6.3: Comparison of models for Scenario 4, with all forward links observed.

subflow model outperforms the PCA model in most of the studied scenarios. The subflow model has the most notable advantage in the case of predicting the byte traffic as opposed to the packet mode, which is most useful for administration of a network.

Network Monitoring Perspective: It is clear that the subflow model is superior for monitoring the network. As shown in Figure , the subflow model outperforms the PCA model in terms of ReMSE_2. This is undoubtedly due to the fact that the subflow model makes more use of the information contained in the reverse flows than the PCA model, a concept that is explored in more detail in the following section.

6.3 Advantages of Subflow Model

In this section, we discuss the intuitive advantages of the subflow model over the PCA model, and show these advantage hold true in practice when imple-

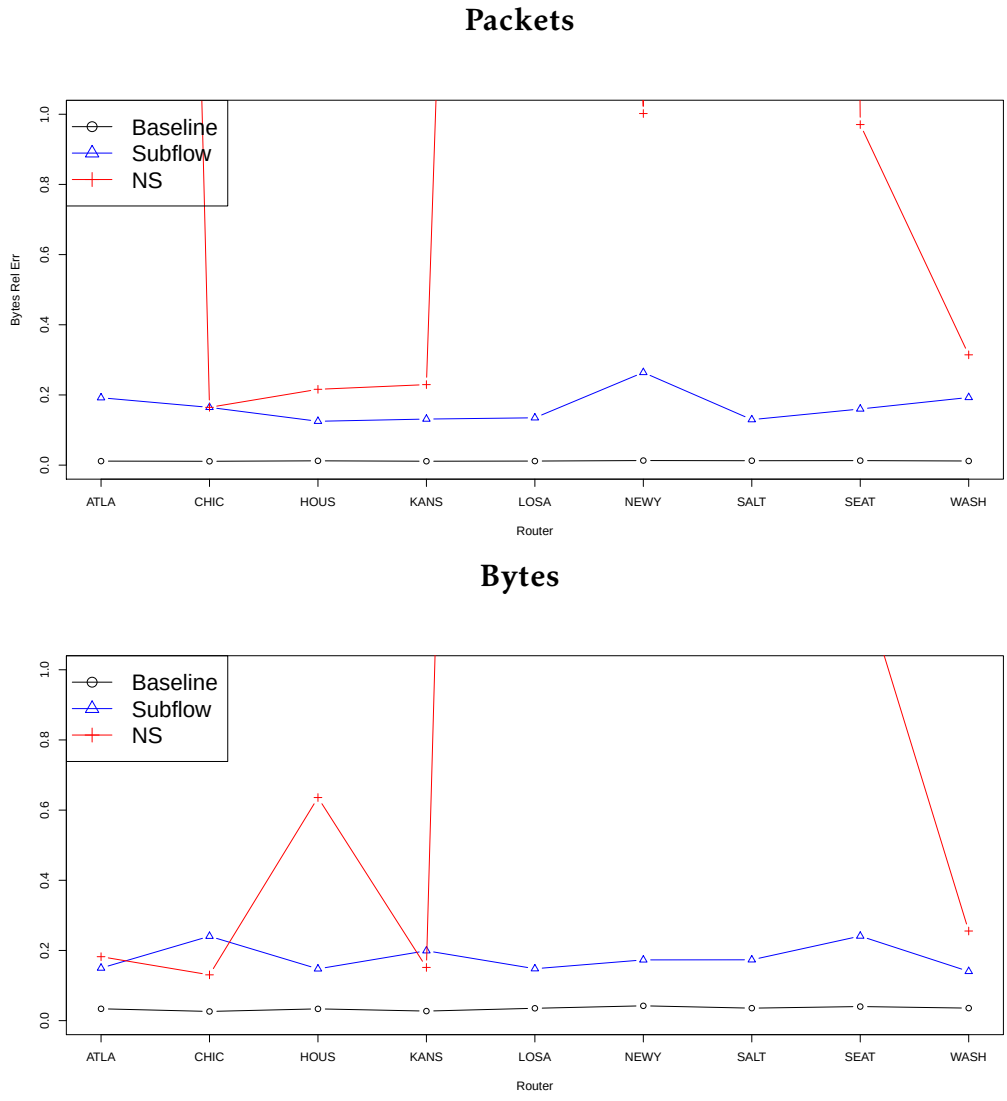


Figure 6.4: Total Network Error: Router Perspective. Traffic is predicted on the entire network, observing only the links directly connected to each of the nine routers of the Internet2 backbone. Notice that we achieve prediction error as low as 15% even though we observe links at a single node.

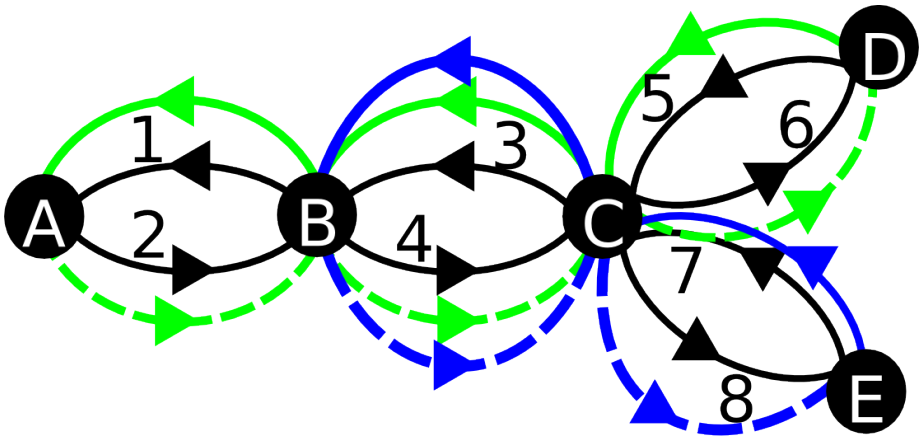


Figure 6.5: Example network illustrating intuition behind the subflow model. The network consists of five routers, labeled A–E, and eight uni–directional links, labeled 1–8. Also depicted are four origin–destination flows, D→A (green solid), E→B (blue solid), A→D (green dashed), and B→E (blue dashed). Other flows are present, but not displayed. If we consider predicting the traffic level on link 3, we wish to learn about the two solid flows by observing other links. Since the odd numbered links carry these flows, these links provide information about the traffic on link 3 under both models. However, under the subflow model, the relationship between the *forward* and *reverse* flows (solid and dotted flows of the same color) is explicitly modeled, and so the even numbered links provide information about the traffic on link 3.

mented with real data from Internet2.

6.3.1 Intuition of Subflow Model

There are two primary advantages of the subflow model. The first is that by using the two modes of network traffic (bytes and packets), the model has more information to use in predicting unobserved traffic. This advantage is somewhat intuitive, and is related to the idea of co–kriging from spatial statistics. It is, at its most basic level, an application of the fundamental idea that additional information may be gained by observing a separate, but correlated quantity. In our case, there is a fundamental relationship between byte and packet level traffic loads which may be exploited to improve prediction. The second advantage, the utility of reverse flows, is more subtle and explored below.

The subflow model, as discussed in Chapter V, explicitly models the relationship between forward and reverse flows through the TCP subflows. These results

in a non–diagonal covariance matrix of the vector of flows ($\vec{X}$), and, consequently, more non–zero entries in the covariance matrix of the link level traffic vector ($\vec{Y}$). In turn, this means that a greater number of links provide information about any unobserved link under this model.

Figure 6.5 shows an illustration of the idea on a simple network. In the figure, the physical structure of the network is depicted in black. Four of the possible origin/destination flows are depicted in color, with forward/reverse pairs depicted in the same color. From the perspective of predicting the traffic on Link 3, the solid links represent "forward" flows, while the dashed lines represent the corresponding "reverse" flows. The model from Chapter IV models the dependence between traffic on links through shared flows, so the other odd–numbered links (1,5,7) provide the most useful information for predicting Link 3. Under this model, however, the flows are considered independent, so the even numbered links are not useful beyond determining an estimate of the mean vector. The subflow model (introduced in Chapter V), explicitly models this dependence, making the links that carry the reverse flows more useful in prediction. Thus, in the network depicted, the even numbered links are more useful in predicting the traffic on Link 3 under the subflow model than they are under the PCA model. This effect is shown in Figure 6.6, where only links carrying "reverse" flows are used in predicting the traffic on an unobserved link. This relationship is further explored in the case of real data in the next section.

6.3.2 Selecting Observed Links

The subflow model introduced in Chapter V raises several interesting questions regarding the choice of which links should be observed in order to predict

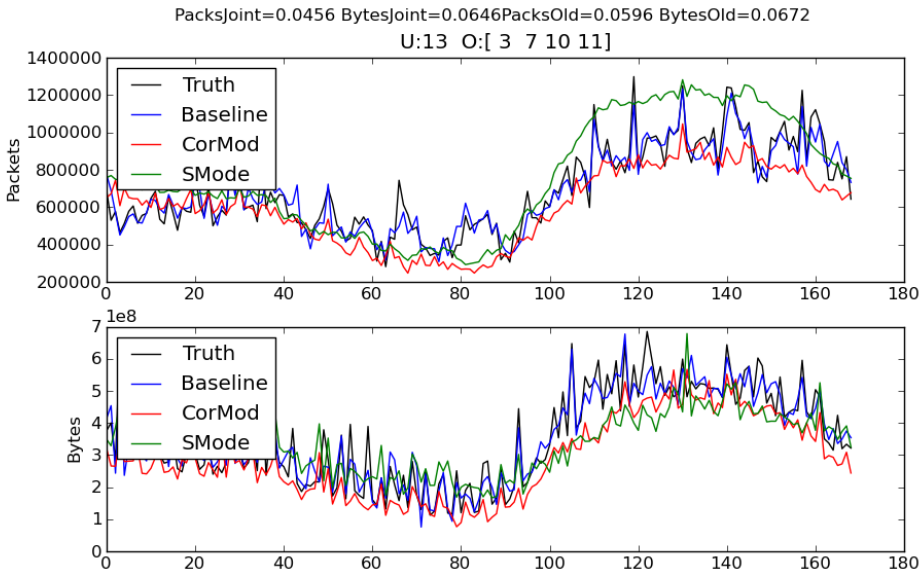


Figure 6.6: Comparison of models for Scenario 4, with all **reverse** links observed.

the unobserved links. In this context, the subflow model raises other interesting questions, which we illustrate in this section. This is an interesting problem, of interest in ongoing and future research.

In this section, we investigate the usefulness of various links in the network to the task of predicting the traffic on a particular link. We conduct a series of experiments, with examples shown in Figures 6.7 and 6.8. In the experiments, we consider predicting the traffic on one link from four others. The four links are reasonable for prediction, in the sense that they carry at least one flow with the predicted link. (This is way the prediction scenarios described in Section 6.1 were chosen.) We gradually switch these links from those that would be useful to the link between the same two routers, but traveling in the reverse directions. Thus, the useful links are switched with those that do not carry any flows in common with the predicted link. They do, however, carry flows that are related to flows carried by the predicted link, via the TCP subflows. As we see in the figures, the

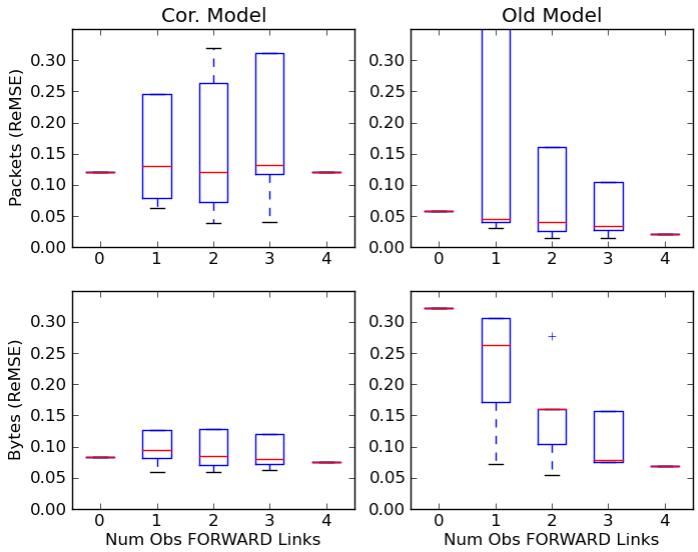


Figure 6.7: Predicting Link 7 via 4 links, chosen to be either forward or reverse. Note the performance decrease for the PCA model versus the more stable performance of the subflow model.

subflow model is robust to these changes, while the PCA model is not, showing that the Subflow model is able to utilize the information in the reverse flows to a much greater degree than the PCA model.

6.4 Summary of Main Contributions

The preceding work provides new methodology to better understand and utilize the complicated data associated with computer network traffic. The main contribution of this work is the development and study of a global spatio–temporal model for computer network traffic. In addition to its methodological contributions to the field, this work also introduces useful applications to network engineers and important themes to guide future work.

The primary focus of this work is the modeling and prediction of computer network traffic. Although the temporal behavior of computer network traffic had been well–studied, the global, network–wide traffic behavior is largely an open

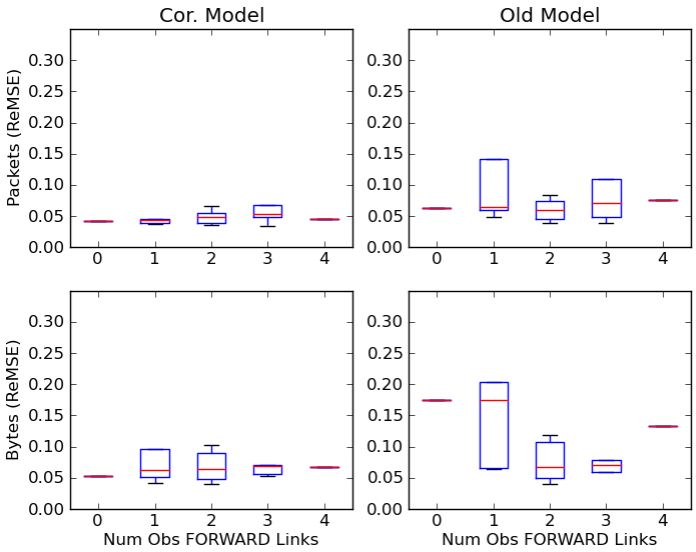


Figure 6.8: Predicting Link 13 via 4 links, chosen to be either forward or reverse. Note the performance decrease for the PCA model versus the more stable performance of the subflow model.

challenge. In this Chapter III, we introduced and developed a spatio–temporal model. This model specifies not only the temporal relationships, but also those of any set of links on the same network. While in principle this model can be used to solve the kriging problem, the values of key parameters are unknown and should be estimated from data. Chapter IV provides a practical method of estimating these parameters by using auxiliary data to develop a structural model which may be fit using a small number of observed links. In Chapter V, we provide an explicit model for the dependence between flows via the behavior of network protocols. The model also describes the relationship between two modes of network traffic: number of packets and amount of data. This chapter introduces multi–modal network kriging. In this framework, we can improve the prediction and kriging error by measuring traffic levels in terms of both number of packets and amount of data. This is because the mechanistic dependence between packets and byes induces correlation between those modalities, which allows us to "bor-

row strength" and thus improve the performance of network kriging.

Two important applications of the proposed methodologies are developed. First, in Chapter IV, we developed a modified control–chart methodology that can be useful to detect network anomalies by comparing the observed traffic levels to those predicted by our model. Because the model implicitly represents the structural relationships of the underling network flows, substantial differences between the observed and predicted traffic indicate a deviation from this structure, and hence an anomaly. Then, in Chapter V, we introduce the idea of single–point network monitoring. This application of the modeling and kriging methodologies allow network administrators to monitor traffic on the entire network while using substantially less network bandwidth for monitoring. If when monitoring with a single node, the prediction error is not sufficiently small for a particular application, then the technique could be extended to the case of monitoring at a small number of nodes, rather than the entire network. This has the potential to balance the bandwidth usage with the quality of the monitoring. The choice of the optimal nodes to monitor is an interesting open question.

Two important themes appear throughout this work. All the methodological developments were scientifically informed, in that they incorporated knowledge about the physical world and the designed features of computer networks. The global model from Chapter III is, at its core, based on the physical behavior of users of the network, and the actual distributions of file sizes. Similarly, the subflow model explored in Chapter V is directly based on the physical relationship between the packet and byte modalities as well as the engineered TCP feedback mechanism. Secondly, this work is an example of the growing field of data–fusion. In both Chapters IV and V, we use two different sources of data about the net-

work traffic. One is accurate but aggregate and computationally inexpensive to work with, while the other is both much noisier and much more challenging computationally, but much more detailed in the information it provides. We show how these fundamentally different sources of information may be combined in a way that acknowledges the limits of each, but produces better results than could be obtained by either on its own. The developed methodologies and principles extend to other types of networks and traffic, such as transportation and social networks. We believe, however, that understanding both the underlying scientific principles and the relationships between disparate types of data are crucial to extending these methods to other types of networks.

BIBLIOGRAPHY

Bibliography

[1] G. E.P. Box, A. Luceno, and M. d. C. Paniagua-Quinones. *Statistical Control by Monitoring and Adjustment*. Wiley, second edition, 2009.

[2] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer-Verlag, New York, 2nd edition, 1991.

[3] J. Cao, W. Cleveland, D. Lin, and D. Sun. Internet traffic tends toward poisson and independent as the load increases. In C. Holmes, D. Denison, M. Hansen, B. Yu, and B. Mallick, editors, *Nonlinear Estimation and Classification*, New York, 2002. Springer.

[4] D.B. Chua, E.D. Kolaczyk, and M. Crovella. Network kriging. *Selected Areas in Communications, IEEE Journal on*, 24(12):2263–2272, Dec. 2006.

[5] Cisco. Introduction to cisco ios netflow. `http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.pdf`, 2007.

[6] N. Cressie. *Statistics for Spatial Data: revised ed*. John Wiley, New York, 1993.

[7] M. Crovella and B. Krishnamurthy. *Internet Measurement: Infrastructure, Traffic, & Applications*. John Wiley & Sons, Ltd, 2006.

[8] B. D'Auria and G. Samorodnitsky. Limit behavior of fluid queues and networks. *Oper. Res.*, 53(6):933–945, 2005.

[9] P. Embrechts and M. Maejima. *Selfsimilar processes*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2002.

[10] I2. About internet2. `http://www.internet2.edu/about`.

[11] M. G. Kallitsis, R. D. Callaway, M. Devetsikiotis, and G. Michailidis. Distributed and dynamic resource allocation for delay sensitive network services. In *2008 IEEE Global Telecommunications Conference, GLOBECOM 2008*, pages 1432–1437. Institute of Electrical and Electronics Engineers, 2008.

[12] I. Karatzas and S. E. Shreve. *Brownian motion and stochastic calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1991.

[13] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi. Network tomography: a review and recent developments. In *Frontiers in statistics*, pages 345–366. Imp. Coll. Press, London, 2006.

[14] Y. Liu, D. Towsley, T. Ye, and J. C. Bolot. An information-theoretic approach to network monitoring and measurement. In *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 14–14, Berkeley, CA, USA, 2005. USENIX Association.

[15] T. Mikosch, S. Resnick, H. Rootzén, and A. Stegeman. Is network traffic approximated by stable Lévy motion or fractional Brownian motion? *The Annals of Applied Probability*, 12(1):23–68, 2002.

[16] NS2. The network simulator – NS2. `http://www.isi.edu/nsnam/ns/`.

[17] K. Park, G. Kim, and M. E. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of the Fourth International Conference on Networks Protocols (ICNP'96)*, October 1996.

[18] K. Park and W. Willinger. Self-similar network traffic and performance evaluation. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic: An Overview*. J. Wiley & Sons, Inc., New York, 2000.

[19] L. L. Peterson and B. S. Davie. *Computer Networks*. Morgan Kaufmann Publishers, San Francisco, CA, 2000.

[20] H. Singhal and G. Michailidis. Identifiability of flow distributions from link measurements with applications to computer networks. *Inverse Problems*, 23(5):1821–1850, 2007.

[21] S. Stoev, G. Michailidis, and J. Vaughan. Global modeling of backbone network traffic. In *INFOCOM: The 29th Conference on Computer Communications*. San Diego, CA, 2010.

[22] S. Stoev and M. S. Taqqu. Wavelet estimation for the Hurst parameter in stable processes. In Govindan Rangarajan and Mingzhou Ding, editors, *Processes with Long-Range Correlations: Theory and Applications*, pages 61–87, Berlin, 2003. Springer Verlag. Lecture Notes in Physics 621.

[23] M. S. Taqqu. Fractional brownian motion and long range dependence. In Murad S. Taqqu Paul Doukhan, George Oppenheim, editor, *Theory and Applications of Long-Range Dependence*. Birkhauser, 2003.

[24] M. S. Taqqu. Fractional Brownian motion and long-range dependence. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory and Applications of Long-range Dependence*, pages 5–38. Birkhäuser, 2003.

[25] M. S. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. *Computer Communications Review*, 27(2):5–23, 1997.

[26] J. Vaughan, S. Stoev, and G. Michailidis. Network-wide statistical modeling and prediction of computer traffic. Technical Report Number 504, 2010.

[27] D. Veitch and P. Abry. `LDestimate`: a Matlab code for wavelet–based estimation of the Hurst long–range dependence parameter. `http://www.cubinlab.ee.unimelb.edu.au/~darryl/LDestimate_code.tar.gz`, 1998.

[28] W. Willinger, V. Paxson, R. H. Riedi, and M. S. Taqqu. Long-range dependence and data network traffic. In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, *Theory and Applications of Long-range Dependence*, pages 373–407. Birkhäuser, 2003.

[29] W. Willinger, M. S. Taqqu, W. E. Leland, and D. V. Wilson. Self-similarity in high-speed packet traffic: Analysis and modeling of ethernet traffic measurements. *Statistical Science*, 10(1), 1995.

[30] B. Yu. Embracing statistical challenges in the information technology age. *Technometrics*, 49(3):237–248, 2007.