Copyright
by
Dogu Arifler
2004

The Dissertation Committee for Dogu Arifler
certifies that this is the approved version of the following dissertation:

Network Tomography Based on Flow Level Measurements

Committee:

Brian L. Evans, Supervisor

Gustavo de Veciana, Supervisor

Ross Baldick

Melba M. Crawford

Theodore S. Rappaport

Sanjay Shakkottai

Network Tomography Based on Flow Level Measurements

by

Dogu Arifler, B.S.E.E., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
May 2004

Dedicated to my family.

Acknowledgments

I wish to thank my parents and my sister for their continuous support throughout my education. Without their encouragement, I would not have accomplished my academic goals.

Very special thanks go to my advisors Prof. Brian L. Evans and Prof. Gustavo de Veciana. I am forever indebted for their guidance and support throughout my doctoral studies. Brian is truly an exceptional advisor. He introduced me to research when I was looking for a senior design project as an undergraduate student at UT Austin. As my advisor, he has taught me a lot, influenced my way of thinking, and supported me for many years. I can say, without overestimation, that I owe more than half of my knowledge to Gustavo. The countless number of classes I took from Gustavo taught me to think everything from the first principles. I have learned the meaning of perfection from him. It is Brian's and Gustavo's invaluable feedback on my research that has made this dissertation possible.

I would also like to thank my committee members Prof. Ross Baldick, Prof. Melba Crawford, Prof. Ted Rappaport, and Prof. Sanjay Shakkottai. It was a privilege to have them on my Ph.D. committee. Their feedback on my research was invaluable. I also wish to acknowledge my Master's advisor Prof. San-Qi Li. He recruited me to UT Austin as a graduate student, influenced and shaped my early research in networking.

Finally, I thank all of the Embedded Signal Processing Laboratory (ESPL) members and the members of Prof. de Veciana's research group. ESPL has provided a pleasant and friendly working environment for me. I am proud to be one of the ESPL alumni.

Dogu Arifler

Austin, Texas
May 2004

Network Tomography Based on Flow Level Measurements

Publication No. ____________________

Dogu Arifler, Ph.D.
The University of Texas at Austin, 2004

Supervisors: Brian L. Evans
Gustavo de Veciana

The primary aim of network tomography is to infer properties of networks from network traffic measurements. Internet traffic mainly consists of flows of packets that belong to World Wide Web accesses, file transfers, and e-mail, whose transmissions are mediated via the Transmission Control Protocol (TCP). TCP flow records, or non-intrusive, flow level measurements, can be collected by the state-of-the-art networking equipment.

In this dissertation, I develop a methodology to process TCP flow records to analyze throughput correlations among TCP flow classes. Throughputs of TCP flows that share resources in the network are correlated. These correlations can be used to infer resource sharing in the Internet. My proposal for using flow level measurements to infer network properties differs significantly from previous network tomography research that has employed packet level measurements for making inferences.

In this work, I develop a sampling strategy for random processes (flow class throughputs) whose samples are taken when the processes are active at the sampling instant. The samples are used to estimate a flow class throughput correlation matrix. Factor analysis is then employed to investigate the correlation structure of TCP flow throughputs and to explore which TCP flow classes might share congested resources. A number of empirical studies are conducted to evaluate the effect of filtering out small or large sized flows on correlation estimates. Bootstrap methods are coupled with exploratory factor analysis to make inferential statements about resource sharing. The applicability of the methods to real datasets is also validated.

Possible applications of the methodology introduced in this dissertation include network monitoring and root cause analysis of poor performance. The methods will have a potential impact on service providers who wish to analyze network performance using flow level measurements. The methodology may also be integrated into the design of future network monitoring equipment and software to perform an off-line evaluation of the congestion status of networks.

Table of Contents

Acknowledgments **v**

Abstract **vii**

List of Tables **xii**

List of Figures **xiii**

Chapter 1. Introduction **1**

- 1.1 Inference of Resource Sharing in the Internet 2
- 1.2 Taxonomy of Network Measurement Methods 3
- 1.3 Related Work 6
- 1.4 Flows, Flow Records, and Flow Classes 8
- 1.5 Acronyms 10
- 1.6 Contributions and Organization of the Dissertation 11

Chapter 2. Methods **14**

- 2.1 Introduction 14
- 2.2 Notation 15
- 2.3 Exploratory Factor Analysis 16
  - 2.3.1 What is factor analysis? 16
  - 2.3.2 The orthogonal factor model 16
  - 2.3.3 Selection of the number of factors 19
  - 2.3.4 Interpretation of factor loadings 20
- 2.4 Bootstrap Methods 21
- 2.5 Conclusion 23

Chapter 3. Flow Level Measures for Capturing Resource Sharing by Flows 25

3.1 Introduction 25
3.2 A Model for Congestion at a Resource 28
3.3 First- and Second-Order Statistics of Flow Class Throughputs 37
3.4 Conditioned Throughputs of Flow Classes 39
3.5 Estimation of the Correlation Matrix from Pairwise Correlations 40
3.6 Simulations Using Fluid Models 42
    3.6.1 The M/GI/1-PS queue 43
    3.6.2 A linear network 44
    3.6.3 Simulation setup 46
    3.6.4 Simulation results and discussion 47
3.7 Conclusion 52

Chapter 4. Inferring Resource Sharing Among TCP Flow Classes: Simulation and Analysis 54

4.1 Introduction 54
4.2 A Brief Review of TCP's Congestion Control 55
4.3 Networks with Tree Topologies 56
    4.3.1 Squared error loss 58
    4.3.2 Single bottleneck: Effect of background traffic 59
    4.3.3 Single bottleneck: Effect of class loads 60
    4.3.4 Single bottleneck: Effect of non-stationary traffic 62
    4.3.5 Three bottlenecks 62
4.4 Interaction of Coupled Flow Classes 64
4.5 Wireless Local Area Networks 67
    4.5.1 Simulation setup 69
    4.5.2 Simulation results 71
    4.5.3 Discussion: Traffic patterns versus spatial access patterns in wireless networks 72
4.6 Conclusion 72

Chapter 5. Case Studies Using Real TCP Flow Measurements 74
5.1 Introduction 74
5.2 Description of Datasets 74
5.3 Methodology 79
5.3.1 Validation of methodology 80
5.3.2 Discussion of results 83
5.4 Conclusion 84

Chapter 6. Conclusion 86
6.1 Summary 86
6.2 Future Work 89

Bibliography 94

Vita 102

List of Tables

1.1 Taxonomy of network measurement methods. 5
1.2 Acronyms. 10

3.1 Key flow attributes. 27

5.1 Description of NetFlow datasets collected at UT Austin's border router. 75
5.2 Mean of bootstrap replications and 95% confidence intervals for eigenvalues of **R** based on Dataset2002. 80
5.3 Mean of bootstrap replications and 95% confidence intervals for eigenvalues of **R** based on Dataset2004. 81
5.4 Mean of bootstrap replications and 95% confidence intervals for factor loadings based on Dataset2002. 82
5.5 Mean of bootstrap replications and 95% confidence intervals for factor loadings based on Dataset2004. 83
5.6 Algorithm to infer resource sharing from flow records. 85

List of Figures

2.1 An exploratory factor analysis model for Y_1, Y_2, Y_3, and Y_4. Y_1 and Y_3 have a common factor F_1. Y_1, Y_2, and Y_4 have a common factor F_2. Each variable also has a factor that is unique to itself given by U_1, U_2, U_3, and U_4. 18

3.1 Sharing of instantaneous available resource capacity by two temporally overlapping flows. In this particular example, both flows have similar packet round trip times and packet loss rates, and the instantaneous bandwidth sharing is roughly fair during the time period over which they overlap. 26

3.2 Temporal overlap between flows f_1 and f_2. The start times, end times, and durations of flows are shown. 30

3.3 Standard deviation of flow throughput as a function of flow duration in (3.2) with $\alpha = 0.3$, and $\sigma_W^2 = 0$, $\sigma_W^2 = 15$. 32

3.4 Standard deviation of flow throughput as a function of flow duration in (3.2) with $\alpha = 0.8$, and $\sigma_W^2 = 0$, $\sigma_W^2 = 15$. 32

3.5 The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 0$. Flow 1 starts at time 0 and ends at time 20. 33

3.6 The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 0$. Flow 1 starts at time 0 and ends at time 30. 33

3.7 The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 15$. Flow 1 starts at time 0 and ends at time 20. 35

3.8 The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 15$. Flow 1 starts at time 0 and ends at time 30. 35

3.9 The effect of noise, $\sigma_W^2 = 15$, on the correlation in (3.1) between throughputs of perfectly overlapping f_1 and f_2 that share a congested resource. Both flows start at time 0. 36

3.10 A collection of flows that belong to two different classes. The flow sojourn times are given by their lengths. The numbers next to the flows (and the widths of flows) indicate the throughputs y_f perceived by flows. The throughput of flow class 1 is (1+2+3)/3=2 at t_1. The throughput of flow class 2 is 2 at t_1. Flow class 1 is inactive at t_2. The throughput of flow class 2 is (2+2)/2=2 at t_2. 38

3.11 Two parallel M/GI/1-PS queues. All four flow classes traverse the measurement point. 45

3.12 A linear network with two resources. All three flow classes traverse the measurement point. 45

3.13 Pairwise correlation between throughputs of flow classes (class 1 and class 2) sharing an M/GI/1-PS queue in Fig. 3.11. . . . 49

3.14 Pairwise correlation between throughputs of flow classes (class 1 and class 3) not sharing an M/GI/1-PS queue in Fig. 3.11. . 49

3.15 Percent variance accounted by significant factors in parallel M/GI/1-PS queues in Fig. 3.11. The points labelled with a 3 on the solid-line plot correspond to the experiments in which three, instead of two, common factors were (incorrectly) identified. 50

3.16 Pairwise correlation between throughputs of classes 2 and 3 coupled by class 1 in the linear network in Fig. 3.12. Correlations are estimated after filtering out flows with sizes larger than 2. 51

4.1 Tree topology used in OPNET TCP simulations. 57

4.2 Percent normalized variance under different background traffic conditions on the single bottleneck. The total load due to classes 1–7 on S1 is kept at 30%. 61

4.3 Squared error loss under different background traffic conditions on the single bottleneck. The total load due to classes 1–7 on S1 is kept at 30%. 61

4.4 Percent normalized variance under different total loads from classes 1–7 on the single bottleneck S1. The utilization of S1 due to background traffic is kept at 50%. 63

4.5 Squared error loss under different total loads from classes 1–7 on the single bottleneck S1. The utilization of S1 due to background traffic is kept at 50%. 63

4.6 Percent normalized variance under different loads offered by each of the classes 1–7 for the three-bottleneck scenario. The total utilization factors of each of the bottlenecks are the same in each offered load case, and are 70%, 80%, and 90%, respectively. 65

4.7 Squared error loss under different loads offered by each of the classes 1–7 for the three-bottleneck scenario. The total utilization factors of each of the bottlenecks are the same in each offered load case, and are 70%, 80%, and 90%, respectively. . . 65

4.8 Linear network topology with two coupled bottleneck links used in OPNET TCP simulations. 66

4.9 A basic service set with twenty wireless users, only four users of interest are shown. All stations can support data rates at 11 Mbps. The link capacity is underprovisioned for the traffic patterns generated by wireless users. Users are perceiving poor quality of service (throughput) due to congestion at the bottleneck link. 70

4.10 A basic service set with twenty wireless users, only four users of interest are shown. The access point is not placed optimally with respect to the spatial distribution of wireless users. Users are perceiving poor quality of service (throughput) due to weak signal strengths at their positions. 70

5.1 Percent distribution of flow sizes in packets for Dataset 2002. . 76

5.2 Percent distribution of flow sizes in packets for Dataset 2004. . 76

5.3 Percent distribution of flow lengths in seconds for Dataset 2002. 77

5.4 Percent distribution of flow lengths in seconds for Dataset 2004. 77

5.5 Cumulative distribution function of flow sizes in kB for Dataset 2002. 78

5.6 Cumulative distribution function of flow sizes in kB for Dataset 2004. 78

6.1 Main steps of the introduced methodology for inferring resource sharing. 87

Chapter 1

Introduction

In today's competitive network service provider market, it is critically important to detect when users experience poor quality of service. Such detection is possible only through effective monitoring of network traffic. Some of the traffic characteristics that are usually monitored include the number of packets dropped at a router over time, the utilization levels of a link, the delay experienced between consecutive packets destined for a given user, and the overall delay experienced by a user when downloading Web documents. Although a number of hardware devices and software tools have been developed to monitor and collect information about network traffic, a well-established methodology for analyzing the voluminous amounts of collected information is not available. Root cause analysis of poor network performance, fault and misconfiguration detection, and postmortem tracing of intrusions or denial of service attacks using the data collected by monitoring tools and devices remain challenging problems for network engineers and researchers.

In addition to the voluminous amounts of collected information, an understanding of the cause for poor network performance is complicated by the fact that network managers generally have information only about their network domain, and have little or no knowledge about the properties of the other domains. *Network tomography* research aims to develop methods for inferring external network characteristics (such as link loss rates, link delays, link utilizations, and routing topologies) using network measurements collected either by actively sending probe packets into the network or by passively monitoring packets at a site. Much of previous network tomography research makes inferences based on packet level characteristics such as the number of packets, packet loss, and packet delay [1–7].

1.1 Inference of Resource Sharing in the Internet

In this dissertation, I address a fundamental network tomography problem that has been identified by Internet service providers (ISPs) and content providers (CPs), and involves determining classes of network flows that share congested resources. Currently, congestion in the Internet might arise due to an overloaded server, an overutilized customer access link, or a link failure or misconfigured routing in the carrier's backbone. Determining which network flows might share congested resources in the Internet is usually difficult without access to the complete routing information for the network.

By inferring which classes of network flows share congested resources using only local measurements and flow attributes (e.g. source/destination addresses of communicating end systems), service providers might perform load balancing of traffic that share a common bottleneck onto disjoint routes. For instance, upon inferring that two customers or customer bases are experiencing poor Web performance due to a bottleneck link serving them, the provider of the Web content might choose to replicate content at a second location to reduce the load on the bottleneck link. From a customer's perspective, determining whether a network provider uses a diverse set of routes (which is an indication of robustness) when carrying different classes of flows to the customer may be valuable, especially since network providers are unwilling to disclose their backbone topology. In this case, inferring that none of the flows share a congested resource might indicate such routing diversity. Identifying a shared congested resource can also be used to determine from where malicious flows are coming (e.g. in postmortem intrusion detection). Even if source addresses were being spoofed, one might still infer with which other flows intruder flows share resources so as to approximate the location of misbehaving hosts.

1.2 Taxonomy of Network Measurement Methods

Network traffic monitoring can be performed by taking two different kinds of measurements, namely *passive* measurement or *active* measurement. I briefly summarize these measurement strategies and their respective advantages and disadvantages.

In passive measurement, the network information is collected based only on the *existing* network traffic. Passive measurements generally involve running an agent, or specialized software, on a network node to collect information about the traffic traversing that node. Sometimes, "in-band" information can be embedded in the traffic so as to measure certain network performance characteristics. Examples of such in-band information are a time stamp and a sequence number in each data packet. Popular tools for collecting passive measurements include Simple Network Management Protocol (SNMP), Remote Monitoring (RMON) [8], and NetFlow [9].

On the other hand, active measurements involve *end systems* injecting *additional* probe traffic into the network towards specified destination end systems in order to be able to measure network performance characteristics. Examples of popular active measurement tools are the commonly available Unix tools: `ping`, `traceroute`, `mtrace`, and `pathchar` [10]. Multicast-based Inference of Network-internal Characteristics (MINC) research [3, 6] also focuses on employing active, end-to-end measurements for network tomography.

An advantage of passive measurements is that no additional bandwidth is wasted by the probe traffic. In general, only privileged users may access the measurements collected by monitoring agents. However, such an approach prevents exposing critical network infrastructure information to the outside world. Passive measurements are usually exported to a data warehouse for further processing.

Active measurements can be performed on-demand by any end system without involvement of internal network nodes. Such measurements are very useful in characterizing user perceived performance such as end-to-end connectivity, round trip packet delay, and packet loss along network routes. Some end-to-end measurement methods may also depend on internal network sup-

Table 1.1: Taxonomy of network measurement methods.

	Advantages	Disadvantages	Examples
Passive	No probe traffic required, secure	Measurements available to privileged users only, internal network nodes perform measurements	SNMP, RMON, NetFlow
Active	Measurements available to end systems on demand	Probe traffic overhead, support from internal network nodes and multicast capability may be necessary	MINC, mtrace, pathchar, ping, traceroute

port. For example, `ping` and `traceroute` require that each router on a probe packet's route respond with an Internet Control Message Protocol (ICMP) [11] reply. The cooperation of a number of receivers and senders is generally necessary in order to be able to collate end-to-end measurements in the analysis of network performance characteristics. The major disadvantage of active measurements is the lack of scalability: As the number of end systems increases, probe traffic grows exponentially.

Table 1.1 summarizes the advantages and disadvantages of each measurement approach. In general, a network monitoring strategy should employ a combination of both approaches whenever possible.

1.3 Related Work

The simplest approach to detecting shared resources is to use a utility such as `traceroute` that tracks the route that a packet follows from its source to its destination. Such utilities, however, require the cooperation of routers in the network on the path of the flow. Owners of the carrier networks are often unwilling to provide information about their networks, and hence, the use of such utilities is not always viable. Savage, Cardwell, and Anderson [12] describe a "locality" based approximation for detecting shared paths by looking at the destination addresses of flows. Their approximation is based on the fact that flows destined to a particular host or network address generally follow the same path, and hence visit the same bottleneck in the network.

Harfoush, Bestavros, and Byers [13] use packet-pair probing for determining whether two flows originating from the same source share a bottleneck. Their technique is based on correlating end-to-end packet loss measurements to identify flows that share "similar network conditions". The main disadvantage of their technique, in addition to being dependent on packet level probing, is the requirement of cooperating senders.

Rubenstein, Kurose, and Towsley [14] develop an end-to-end technique based on packet loss or delay observations to infer whether or not two flows are experiencing congestion on a common set of network resources. Their methodology is based on the observation that losses or delays experienced by two packets passing through the same bottleneck exhibit some degree of positive correlation. A major shortcoming of this approach is the prohibitive computational cost to correlate packet level measurements. Moreover, the technique assumes that the flows share a common endpoint; i.e., either the sources or the destinations of packets are co-located and collaborating, which has limited applicability.

Rabbat, Nowak, and Coates [15] propose sending packet probes from two sources to infer whether a subgraph of a graph formed from the paths connecting two sources to two receivers is shared. The methodology is based on the assumption that probe packets arrive at a receiver in the order in which they reach the node where paths from two sources join. In addition to being based on a packet level approach, the method is limited to two sources and a generalization to more than two sources may not be scalable.

Katabi, Bazzi, and Yang [16] develop iterative techniques that minimize entropy-based cost functions to cluster flows that share a bottleneck into groups. Their method is based on the observation that correct clustering minimizes the entropy of inter-packet spacing within clusters with an empirical distribution measured by an observer. The main advantage of their method is that it does not require sending probe traffic into the network and does not require cooperating senders; i.e., it is passive. However, they also indicate that their technique is robust only when the observer can monitor a large fraction of the traffic from the bottleneck link, and hence is not practical when the observer is an end-receiver.

My work on inferring resource sharing differs significantly from the previous work in that I consider *flow level* instead of *packet level* statistics.

Packet level Internet Protocol (IP) traffic is data intensive to collect and store for subsequent analysis. In addition, packet level characteristics of IP traffic are complex to analyze due to their extreme variability over a wide range of time scales [17]. Although methods based on packet level measurements are invaluable in inferring packet round trip times and loss rates along network routes, flows may provide a suitable, alternative measurement basis for inferring congested resource sharing. Flows are defined for longer time scales, and hence better capture congestion dynamics in the network and the performance perceived by end-users of “document” traffic, i.e. Web transfers, file transfers, and e-mail. I rely on passive measurements made at a network node (e.g., router, gateway, or server), although it is possible to take an active approach by sending probe flows into the network. Furthermore, while many other previous methods that infer resource sharing are limited to determining whether particular flow class *pairs* share bottlenecks, the method developed in this dissertation considers a set of flow classes *simultaneously.*

1.4 Flows, Flow Records, and Flow Classes

I employ a flow level performance measure, *throughput*, which is directly available from state-of-the-art network monitoring tools, in order to infer which network flow classes share congested resources. Since network flows are the main interest in this work, I first define *flows, flow records*, and *flow classes.* Although there is no standard definition of a *flow*, a commonly accepted definition of an IP flow is a unidirectional sequence of packets, which are close to each other in time and share a common identifier such as a common source and destination address [18]. For instance, packets corresponding to a file download constitute a flow.

The state-of-the-art networking equipment that runs traffic monitoring tools (such as NetFlow [9], sFlow [19], and Argus [20]) is capable of generating *flow records*. A flow record contains the source and destination IP addresses, Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers, IP protocol type, type of service fields in IP headers, start and end times, and the number of packets and bytes in a flow. A major problem in flow measurement is the lack of scalability: At very high speed routers, the number of flows to be measured might easily exceed millions per hour. Therefore, at high link speeds, the flows [21] and/or the packets within a flow [22] may be sampled in order to keep up with the link speeds. The network node, such as a router, performing record generation usually exports these records to a data warehouse for further processing.

I define an IP *flow class* as a collection, or aggregation, of flows that have a common attribute. For example, we can refer to all flows sharing common source and destination IP address prefixes as a flow class. A Web browsing session, in which a user visits a number of pages at a Web site and triggers a number of object downloads at each page, generates flows that may be treated as a flow class.

Table 1.2: Acronyms.

AP	Access Point
BC_a	Bias-Corrected and Accelerated
BSS	Basic Service Set
CP	Content Provider
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CWND	Congestion Window
FTP	File Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISP	Internet Service Provider
M/GI/1-PS	Single-server processor sharing queueing system with an exponential interarrival time distribution and a general, independent service time distribution
MINC	Multicast-based Inference of Network-internal Characteristics
OC	Optical Carrier
RMON	Remote Monitoring
RTT	Round Trip Time
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

1.5 Acronyms

Table 1.2 lists some acronyms, most of which are related to networking. These acronyms will be used throughout this dissertation, and are included here for easy reference.

1.6 Contributions and Organization of the Dissertation

A significant portion of the IP traffic consists of packets from *elastic flows* [23] or "document" traffic, i.e. Web transfers, file transfers (FTP), and e-mail, whose transfers are mediated via TCP (see for example, [24]). In this work, I will exclusively consider TCP flows and all references to flows and flow classes will imply TCP flows and flow classes. TCP uses packet delay and loss as indicators of the available bandwidth to adjust the data transmission window at the sender. Note that capturing this dynamic adjustment of data transmission window from flow records is not possible. Such capture could only be possible by continuously monitoring the data transmission windows of end systems, and would be prohibitively expensive. However, TCP flow throughputs that are available from flow records enable one to infer the congestion status of the network the flows visit.

One key observation in this work is that end users of elastic flows that are temporally overlapping long enough on the same congested resource tend to perceive high quality of service or low quality of service together. In my context, user perceived quality of service is related to the throughput of a flow, i.e., the size of the flow divided by the response time of the flow. This association of quality of service perceived by users directly translates into a *positive correlation* among flow throughputs that share congested resources. While the existence of such correlations is intuitive, their extent needs to be quantified, especially when flow classes visiting multiple resources can introduce throughput correlations among flow classes that do not necessarily share congested resources. I will term the inference of resource sharing based only on measurements collected at one site and the attributes of flows as a "black box" approach [25–27].

In this dissertation, I defend the following thesis statement:

> *The correlation structure of throughputs obtained by flow level measurements for a number of TCP flow classes can often be captured by a fewer number of latent factors that can be used to infer which flow classes share resources in the network.*

The research presented in this dissertation is focused on establishing the validity this statement and exploring its applicability to network performance analysis. The primary contributions of this research are:

1. Description of a methodology to process TCP flow records in order to analyze throughput correlations among TCP flow classes that can be used to infer resource sharing in the Internet.
2. Development of a sampling strategy for flow class throughputs (random processes) whose samples are taken when the classes are active at the sampling instant.
3. Evaluation of the use of *factor analysis* on processed flow records to explore which TCP flow classes might share congested resources. I empirically investigate the effect of filtering out small and large flows on inferences for resource sharing.

4. Validation of the inference methodology using bootstrap methods and non-intrusive, flow level measurements collected at a single network site. I apply my methodology to real TCP flow measurements, and use bootstrap methods to exhibit the statistical accuracy of my inferences.

The outline of the rest of the dissertation is as follows. Chapter 2 reviews the basic theory of *factor analysis* and *bootstrap methods*, and introduces the notation used throughout the dissertation. Chapter 3 describes the method for constructing a flow class throughput correlation matrix using flow level measurements. Known, analytical fluid models are used to explain the causes for positive correlations among throughputs of flow classes that share congested resources. Chapter 4 describes results based on an extensive set of TCP simulations on tree topologies and demonstrates the effectiveness of the proposed methodology. Chapter 5 analyzes real TCP data and uses bootstrap methods to exhibit the statistical accuracy of inferences for resource sharing. Section 6 concludes the dissertation.

Chapter 2

Methods

2.1 Introduction

This chapter reviews the basic theory of *factor analysis* and *bootstrap methods*. Factor analysis will be used extensively to analyze the correlation structure of flow class throughputs, and is the core technique employed for determining which flow classes share congested resources in the network. The bootstrap methods will be used for assessing the accuracy of statistical estimates that result from factor analyzing flow class throughput correlation matrices associated with real data. The bootstrap methods allow one to make inferential statements based on the data at hand.

Section 2.2 introduces the basic notation used. Section 2.3 describes factor analysis based on the principal component method. Section 2.4 discusses the bootstrap, a computer-based method that depends on resampling a given set of data in order to be able to make assessments of the statistical accuracy of an estimate for any statistic, simple or complicated, of data from an unknown probability distribution. Finally, Section 2.5 provides a brief summary of the main concepts discussed in this chapter.

2.2 Notation

Throughout the dissertation, random variables are denoted by uppercase Roman letters, and their realized values (outcomes) by the corresponding lowercase letters. Boldface Roman letters are used for vectors and matrices whose dimensions and randomness will be explicitly stated whenever they are not obvious from the context. A sequence of random variables, say $Y(i)$, is denoted by $\{Y(i)\}$, and their realization by $\{y(i)\}$ for $i = 0, 1, 2, \ldots$. Uppercase Roman letters are also used for constants. Greek letters are generally reserved for parameters (non-random quantities). The estimator of a parameter will have a "hat" (e.g., $\hat{\theta}$). The letters b, i, j, and n are generally used for indexing, and will take integer values. Script letters will be used to denote sets, and $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$.

$\mathbf{Y} \sim f_{\mathbf{Y}}$ means that $\mathbf{Y}$ is a random vector with a (joint) probability distribution function (pdf) $f_{\mathbf{Y}}$. The expectation or mean of a random vector $\mathbf{Y}$ is written as a vector $\boldsymbol{\mu}_{\mathbf{Y}} = \mathbb{E}[\mathbf{Y}]$. The transpose operation for a vector or a matrix is denoted by a superscript T (e.g., $\mathbf{Y}^T$). A $p \times p$ diagonal matrix $\mathbf{D}$ can be written as $\mathrm{diag}(d_{11}, \ldots, d_{pp})$. A $p \times p$ identity matrix $\mathbf{I}_p$ is a diagonal matrix with a 1 in each diagonal position. The trace, $\mathrm{tr}(\cdot)$, of a matrix is the sum of its diagonal elements. The Euclidian norm of a matrix $\mathbf{A}$ is denoted by $\|\mathbf{A}\|$, and is given by $\sqrt{\mathrm{tr}(\mathbf{A}\mathbf{A}^T)}$.

The covariance matrix for a p-dimensional random vector $\mathbf{Y}$ is given by a $p \times p$ matrix $\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{Y}) = \mathbb{E}[(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}})(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}})^T]$. The correlation matrix for a p-dimensional random vector $\mathbf{Y}$ is a $p \times p$ matrix $\mathbf{R} = \mathrm{Corr}(\mathbf{Y}) =$

$\mathbf{D}^{-1}\boldsymbol{\Sigma}\mathbf{D}^{-1}$, where $\mathbf{D} = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, and σ_i is the standard deviation of the ith component of $\mathbf{Y}$ for $i = 1, \ldots, p$. The *total variance*, a measure of variability in data, will be given by $\mathrm{tr}(\boldsymbol{\Sigma})$ (or $\mathrm{tr}(\mathbf{R})$, in which case it may be called *total normalized variance*, and is equal to p.).

2.3 Exploratory Factor Analysis

2.3.1 What is factor analysis?

Although the development of factor analysis may be credited to a number of people, its early uses were mainly in psychometric studies to put forward hypotheses about the organization of mental abilities of individuals based on examining the correlation or covariance structure of scores on a set of cognitive tests [28, 29]. Later, factor analysis was applied in a diverse number of disciplines including econometrics, biometrics, and sociology [30].

In this work, *exploratory* factor analysis is considered. In exploratory factor analysis, there is no theoretical hypothesis about the underlying structure of variables. Instead, one attempts to simplify complex interrelationships among a set of random variables in order to gain insight into their underlying structure. Such simplification amounts to finding a new set of *latent* random variables (factors) that are fewer in number than the original set of variables.

2.3.2 The orthogonal factor model

Suppose that $\mathbf{Y} = (Y_1, \ldots, Y_p)^T$ is a vector of p random variables with a mean vector $\boldsymbol{\mu}_{\mathbf{Y}} = (\mu_1, \ldots, \mu_p)^T$. The idea underlying factor analysis is to consider a representation for $\mathbf{Y}$ in terms of a random vector of m ($m \leq p$) *common* factors $\mathbf{F} = (F_1, F_2, \ldots, F_m)^T$, and a random vector of p *unique* factors $\mathbf{U} = (U_1, \ldots, U_p)^T$. For example, Fig. 2.1 illustrates a case in which four variables are represented in terms of two common factors plus four unique factors. We assume that $\mathbf{Y}$ can be expressed as

$$\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}} = \boldsymbol{\Lambda}\mathbf{F} + \mathbf{U}, \tag{2.1}$$

where $\boldsymbol{\Lambda}$ denotes a deterministic $p \times m$ *loading matrix*. The following additional assumptions are usually made: $\mathbb{E}[\mathbf{F}] = \mathbf{0}$, $\mathrm{Cov}(\mathbf{F}) = \mathbb{E}[\mathbf{F}\mathbf{F}^T] = \mathbf{I}_m$ (orthogonal factors), $\mathbb{E}[\mathbf{U}] = \mathbf{0}$, $\mathrm{Cov}(\mathbf{U}) = \mathbb{E}[\mathbf{U}\mathbf{U}^T] = \boldsymbol{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_p)$ (a diagonal matrix), and $\mathrm{Cov}(\mathbf{U}, \mathbf{F}) = \mathbf{0}$. The assumption that $\boldsymbol{\Psi}$ is diagonal means that all covariances among variables are accounted by the factors. Using (2.1), one can then write

$$\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{Y}) = \mathbb{E}[(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}})(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}})^T] = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}.$$

Alternatively, one can obtain a correlation matrix $\mathbf{R}$, and express it in terms of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$:

$$\mathbf{R} = \mathrm{Corr}(\mathbf{Y}) = (\rho_{ij}) = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}, \quad i, j = 1, \ldots, p \tag{2.2}$$

where $\rho_{ii} = 1$ for $i = 1, \ldots, p$. In this work, I will use the correlation matrix in (2.2) instead of the covariance matrix because the magnitudes of variables of interest can vary greatly, and normalizing such measurements is preferable.

The elements of the loading matrix $\boldsymbol{\Lambda}$, Λ_{ij}, capture the degree of correlation exhibited between a given factor j and variable i. Estimates $\hat{\boldsymbol{\Lambda}}$ and

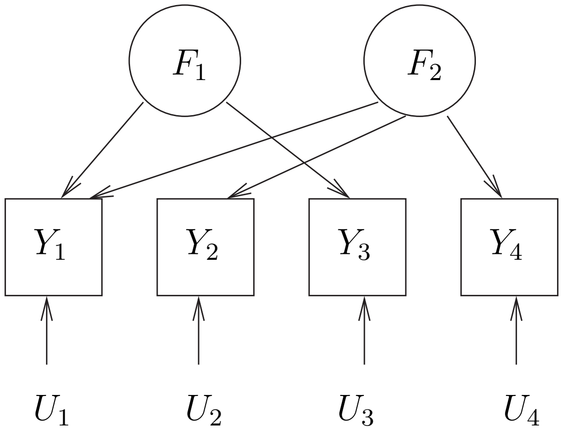

Figure 2.1: An exploratory factor analysis model for Y_1, Y_2, Y_3, and Y_4. Y_1 and Y_3 have a common factor F_1. Y_1, Y_2, and Y_4 have a common factor F_2. Each variable also has a factor that is unique to itself given by U_1, U_2, U_3, and U_4.

$\hat{\mathbf{\Psi}}$ for $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ can be determined by using the *principal component method* as follows (see [31] for more details)[1]. First, the (positive definite) correlation matrix in (2.2) is expressed as

$$\mathbf{R} = e_1 \boldsymbol{\xi}_1 \boldsymbol{\xi}_1^T + e_2 \boldsymbol{\xi}_2 \boldsymbol{\xi}_2^T + \ldots + e_p \boldsymbol{\xi}_p \boldsymbol{\xi}_p^T,$$

where $(e_i, \boldsymbol{\xi}_i)$ are the eigenvalue-eigenvector pairs such that $e_1 \geq e_2 \geq \ldots \geq e_p > 0$. $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ can be determined by taking the largest m eigenvalues, and

---

[1] Unlike the maximum likelihood method, the principal component method does not assume multivariate normality of data when estimating $\mathbf{\Lambda}$ and $\mathbf{\Psi}$. Hence, the principal component method is suitable for analyzing multivariate observations from non-normal distributions such as Internet measurements.

by approximating $\mathbf{R}$ as

$$\begin{aligned}\mathbf{R} &\approx \hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Lambda}}^T + \hat{\boldsymbol{\Psi}} \\ &= (\sqrt{e_1}\boldsymbol{\xi}_1, \ldots, \sqrt{e_m}\boldsymbol{\xi}_m) \times (\sqrt{e_1}\boldsymbol{\xi}_1, \ldots, \sqrt{e_m}\boldsymbol{\xi}_m)^T \\ &\quad + \begin{pmatrix} \hat{\psi}_1 & 0 & \cdots & 0 \\ 0 & \hat{\psi}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\psi}_p \end{pmatrix}, \end{aligned} \tag{2.3}$$

so that $\hat{\Lambda}_{i1}^2 + \hat{\Lambda}_{i2}^2 + \ldots + \hat{\Lambda}_{im}^2 + \hat{\psi}_i = h_i^2 + \hat{\psi}_i = 1$ for $i = 1, \ldots, p$, where h_i^2 is called the *communality*, and $\hat{\psi}_i$ is called the *specific "variance"*. The communality represents the portion of the normalized variance of Y_i that is captured by the m common factors, while $\hat{\psi}_i$ reflects the portion of the normalized variance due to a factor that is unique to Y_i.

2.3.3 Selection of the number of factors

The number of factors m used in the model needs to account for a "reasonable" proportion of the total variance (a measure of overall variability), which is given by the trace of the correlation matrix (see Section 2.2). The proportion of the total normalized variance due to the jth factor is given by e_j/p, i.e., the jth eigenvalue divided by the number of variables. If the proportion of the total normalized variance captured by the common factors is "high", then we say that the factors have a strong or high *explanatory power*.

When using the principal component method to "factor" the correlation matrix *without any assumptions on the distribution of the variables*, one can only use ad hoc heuristics for determining the sufficiency of the number of factors m in the model. In exploratory studies, one common approach to determine m is proposed by Kaiser [32]. Kaiser's rule proposes selecting factors whose normalized variances (e_j) are greater than 1. The intuition behind this rule is that a factor that has a variance less than 1 contains less information than a normalized original variable does.

My experiments with data generated by simulation suggest that Kaiser's rule generally retains fewer factors than expected. Therefore, for simulated datasets, I select m based on the number of eigenvalues that are greater than 0.9. In contrast, I have found that Kaiser's rule produces good results using real TCP data with the following modification: I select m based on the number of eigenvalues whose confidence intervals contain 1 or lie above 1. I will refer to these rules as the modified Kaiser's rule.

2.3.4 Interpretation of factor loadings

The common factors represent shared sources of variation in variables. Among the loadings for a given variable i, i.e., $\hat{\Lambda}_{i1}, \hat{\Lambda}_{i2}, \ldots, \hat{\Lambda}_{im}$, the loading(s) with the largest magnitude(s) are treated as significant loadings. The variables that have the largest loading with a common factor are identified as variables that share a common source of variation.

Note that the loading matrix is determined only up to an orthogonal rotation matrix $\mathbf{\Gamma}$. If $\mathbf{\Lambda}^* = \mathbf{\Lambda\Gamma}$, then

$$\mathbf{R} = \mathbf{\Lambda}^*\mathbf{\Lambda}^{*T} + \mathbf{\Psi} = \mathbf{\Lambda\Gamma\Gamma}^T\mathbf{\Lambda}^T + \mathbf{\Psi} = \mathbf{\Lambda\Lambda}^T + \mathbf{\Psi}.$$

In this work, I apply a rotation to the loading matrix to obtain a better description of the factors by using a common method in factor analysis called *varimax* rotation [31]. Varimax rotation attempts to find a rotation matrix $\mathbf{\Gamma}$ such that the squares of the loadings on each factor are as spread out as possible. More specifically, $\mathbf{\Gamma}$ is chosen to maximize

$$\sum_{j=1}^{m}\left[\sum_{i=1}^{p}\Lambda_{ij}^{*4}-\frac{1}{p}\left(\sum_{i=1}^{p}\Lambda_{ij}^{*2}\right)^2\right].$$

This criterion tends to drive squared loadings towards either zero or one, and away from intermediate values. Hence, deciding which loadings are significant is easier with $\mathbf{\Lambda}^*$.

2.4 Bootstrap Methods

For most statistics, there is no formula for computing the standard error of an estimate. Moreover, most of the time, the distribution of a random sample is unknown. The *bootstrap* [33] was introduced to address these issues. The background material in this section is based largely on [33], wherein a more detailed discussion of the bootstrap can be found.

The bootstrap is a computer-based method that depends on *resampling* a given set of data consisting of N samples B times. A bootstrap sample is a random sample of size N drawn with replacement from the original sample. Corresponding to each bootstrap replication, an estimate for the parameter of interest $\hat{\theta}^*(b)$ is computed for $b = 1, \ldots, B$. The standard error of $\hat{\theta}$ is determined by computing the standard deviation of B independent replications of $\hat{\theta}$. The standard error can be treated as an approximate confidence interval for θ.

A bootstrap method that provides confidence intervals that are very close to the exact confidence intervals of θ is the one that estimates the bias-corrected and accelerated (BC_a) confidence intervals. I compute the BC_a confidence intervals whenever an assessment of the statistical accuracy of estimates based on collected samples at hand is needed. Let $\hat{\theta}^{*(\alpha)}$ denote the (100α)th percentile of B bootstrap replications. With an acceleration $\hat{a}$ and bias correction $\hat{z}_0$, the BC_a interval of intended coverage $1-2\alpha$, with a lower confidence bound $\hat{\theta}_{\mathrm{lo}}$ and an upper confidence bound $\hat{\theta}_{\mathrm{up}}$, is given by

$$(\hat{\theta}_{\mathrm{lo}}, \hat{\theta}_{\mathrm{up}}) = (\hat{\theta}^{*(\alpha_1)}, \hat{\theta}^{*(\alpha_2)}),$$

with

$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(\alpha)}}{1-\hat{a}(\hat{z}_0 + z^{(\alpha)})}\right),$$

$$\alpha_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(1-\alpha)}}{1-\hat{a}(\hat{z}_0 + z^{(1-\alpha)})}\right),$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $z^{(\alpha)}$ is the (100α)th percentile point of a standard normal distribution. The bias correction z_0 accounts for possible bias in the estimate of θ, and the acceleration a accounts for possible change in the standard error of $\hat{\theta}$ as θ varies. The estimators of z_0 and a are given by

$$\hat{z}_0 = \Phi^{-1}\left(\frac{|\{\hat{\theta}^*(b) < \hat{\theta}\}|}{B}\right),$$

and

$$\hat{a} = \frac{\sum_{i=1}^{N}(\hat{\theta}_{(\cdot)} - \hat{\theta}_{(i)})^3}{6\{\sum_{i=1}^{N}(\hat{\theta}_{(\cdot)} - \hat{\theta}_{(i)})^2\}^{3/2}}.$$

Here, $\Phi^{-1}(\cdot)$ indicates the inverse function of a standard normal cumulative distribution function. $\hat{\theta}$ is the estimate of θ based on the original data (without resampling), and $\hat{\theta}_{(i)}$ is the estimate of θ computed with the ith value in the data deleted. Finally, $\hat{\theta}_{(\cdot)}$ is given by $\frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_{(i)}$. The recommended number of bootstrap replications to compute BC$_a$ confidence intervals is at least 1000 [33].

2.5 Conclusion

In summary, the factor analysis model assumes that each of p random variables can be expressed as a linear combination of m ($m < p$) (unobserved) common factors and a unique factor. When m is much smaller than p, such a model may be very useful in compactly describing the variability in the original set of variables. The principal component method can be used to estimate the loadings and communalities in the factor model.

The principal component method makes no assumptions on the distributions of the original variables. Instead, the bootstrap method is used to assess the statistical accuracy of estimates of very complicated statistics of data from an unknown distribution. It is a computationally intensive method that has become viable through the availability of high speed computers in recent years.

The methods introduced in this chapter are applied to network flow measurements in subsequent chapters. In Chapters 3–5, factor analysis is used to determine the latent factors of elastic flow class throughputs. The bootstrap is employed in Chapter 5 to assess the statistical accuracy of inference results for real TCP measurements.

Chapter 3

Flow Level Measures for Capturing Resource Sharing by Flows

3.1 Introduction

The quality of service perceived by elastic flows can be characterized by their throughput, i.e., their size in bits, bytes, or packets divided by their delay, or sojourn time, in the network. The throughputs of elastic flows that temporally overlap on congested resources are positively correlated. This premise is in fact very intuitive: e.g., users downloading documents perceive high quality of service or poor quality of service together when there is a common congested resource along the routes from the sources of the documents to users.

Correlated throughputs can be explained by the dynamic bandwidth sharing of rate control mechanisms, such as TCP, that allocate rates or instantaneous bandwidths to resource sharing flows. The capacity or bandwidth available to a flow changes with new flow arrivals and departures over time. Fig. 3.1 illustrates sharing of instantaneous available resource capacity by two temporally overlapping flows with similar packet round trip times and packet loss rates. Due to dynamic bandwidth sharing, the instantaneous bandwidth available to the flows will vary in a correlated manner during the time period

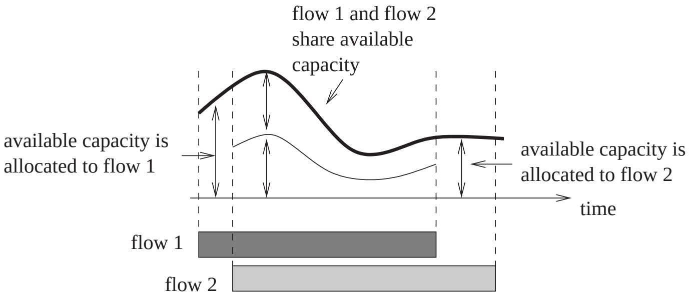

Figure 3.1: Sharing of instantaneous available resource capacity by two temporally overlapping flows. In this particular example, both flows have similar packet round trip times and packet loss rates, and the instantaneous bandwidth sharing is roughly fair during the time period over which they overlap.

over which they overlap. The throughput of a flow is determined by the variable bandwidths allocated to flows during their sojourn in the network. Hence, a key premise in this work is that the throughputs of temporally overlapping flows that share a resource are correlated.

The collection of flows in the network is denoted by a set $\mathcal{F}$. The size, start and end times[1], and duration of a flow f is denoted by v_f, s_f, e_f, and $d_f = e_f - s_f$, respectively. Each flow $f \in \mathcal{F}$ belongs to a flow class $c \in \mathcal{C}$. The function $\phi : \mathcal{F} \rightarrow \mathcal{C}$ associates a flow with a flow class. Key flow attributes are shown in Table 3.1 for easy reference.

Using the notation in Table 3.1, let $\mathcal{F}_c(t) = \{f \in \mathcal{F} : \phi(f) = c \text{ and } s_f \leq t < e_f\}$ denote the set of flows that belong to class c and are *active* at time t. A flow class is active at time t if $|\mathcal{F}_c(t)| > 0$. The perceived *throughput* of a

---

[1] For actual TCP flows, the start time is the time of arrival of the first packet in a flow, and the end time is the time of arrival of the last packet in a flow.

Table 3.1: Key flow attributes.

$\mathcal{F}$	Set of flows, e.g., $f \in \mathcal{F}$
$\mathcal{C}$	Set of classes, e.g., $c, c_i, c_j \in \mathcal{C}$
v_f	Size of flow f in bits, bytes, or packets
s_f	Start time of flow f
e_f	End time of flow f
d_f	Duration, or sojourn time, of flow f
y_f	Throughput of flow f, and is given by v_f/d_f
$b_f(t)$	Bandwidth allocated to flow f at time t
$\phi(\cdot)$	A function that maps a flow to a flow class

flow f is given by $y_f = v_f/d_f$.

This chapter analyzes the extent of throughput correlations that exist between resource sharing elastic flow classes by using known analytical models. Note that the instantaneous rates allocated for flows, which are the primary measurement basis for making inferences in this dissertation, are not available from flow records. Nevertheless, throughput measurements that are directly available from the records can be correlated to make inferences on resource sharing.

Section 3.2 introduces a simple model for the congestion level seen by flows at a resource, and analyzes the degree of correlation between throughputs of flows with different durations and different amounts of temporal overlap. Section 3.3 describes how to process flow records to compute flow class throughputs. Section 3.4 formulates a conditional sampling strategy to analyze time-varying flow class throughputs in the classical factor analysis framework.

Section 3.5 proposes a method to estimate the flow class throughput correlation matrix using pairwise correlations between flow class throughputs. Section 3.6 presents simulation results based on analytical fluid models to verify correlations due to resource sharing and to explore the viability of explaining throughput variability in terms of factors. Section 3.7 concludes the chapter.

3.2 A Model for Congestion at a Resource

First, I formalize the existence of correlation between throughputs of elastic flows that share a resource. In Chapter 1, it was stated that end users of elastic flows that are temporally overlapping long enough on the same congested resource tend to perceive high throughput or low throughput together.

Fluid models, which are the main focus of this chapter, are used to analyze rate control mechanisms. In these models, the rate or bandwidth allocated to a flow is adjusted *instantaneously* when the number of flows in the system changes as a result of flow arrivals and departures. Unlike fluid models, actual rate control mechanisms, such as TCP which will be discussed in Chapter 4, take some time to react to changes in the congestion state of the network. Furthermore, in TCP, the throughputs of very small flows are limited by TCP's Slow Start [11]. Therefore, small flows may not have an opportunity to "learn" the congestion state of the network during their sojourn time. The throughput of large flows (flows that carry a large number of bytes) is largely independent of the arrival and departure dynamics of flows in the system, and is approximately equal to a mean value (see e.g., [34]).

Consider a simple model based on a first-order autoregressive (AR(1)) process for the congestion level seen by a flow[2]. Let $\{B(i)\}$ be an AR(1) process with mean μ_B that represents the instantaneous bandwidth available to each flow sharing a resource at discretized times, and is defined by

$$B(i) - \mu_B = \alpha \left(B(i-1) - \mu_B\right) + Z(i),$$

where $\{Z(i)\} \sim N(0, \sigma_Z^2)$, $|\alpha| < 1$, and $Z(i)$ is uncorrelated with $B(j)$ for each $j < i$. Note that a given flow f carries an amount of data (e.g., bits, bytes, or packets) equal to

$$V_f = \sum_{i=s_f}^{e_f} B(i).$$

For simplicity, consider the throughputs of two flows f_1 and f_2 with given start and end times, and suppose that $s_{f_1} = 0$ and $s_{f_1} \leq s_{f_2}$ without loss of generality (e.g., see Fig. 3.2). The throughputs of f_1 and f_2 will be[3]

$$Y_{f_1} = \frac{1}{d_{f_1}} \sum_{i=0}^{e_{f_1}} B(i) + \frac{W_{f_1}}{d_{f_1}} \quad \text{and} \quad Y_{f_2} = \frac{1}{d_{f_2}} \sum_{j=s_{f_2}}^{e_{f_2}} B(j) + \frac{W_{f_2}}{d_{f_2}},$$

where $W_{f_1}, W_{f_2} \sim N(0, \sigma_W^2)$ model the "noisy" throughputs seen by short flows, and are independent of each other, $\{B(i)\}$, and $\{Z(i)\}$. In this context, a "noisy" throughput means that the throughput perceived by a flow is not a typical one for the class to which the flow belongs. For flows with long sojourn times, the "noise" terms become negligible. The autocorrelation function of

---

[2] I also assume that the congestion process is roughly independent of the flow; i.e., the flow makes only a small contribution to the overall congestion.

[3] Note that in the discrete-time AR(1) model, $d_f = e_f - s_f + 1$.

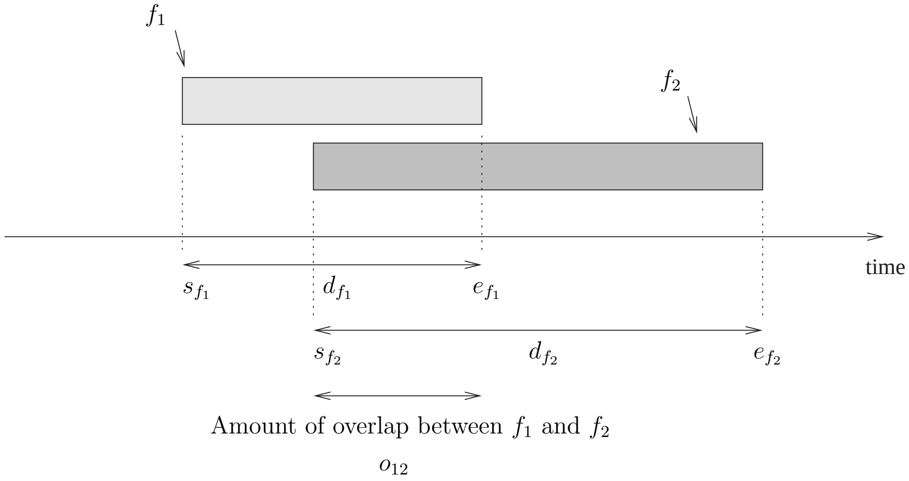

Figure 3.2: Temporal overlap between flows f_1 and f_2. The start times, end times, and durations of flows are shown.

$\{B(i)\}$ is denoted by $\gamma(h)$, and is equal to $\dfrac{\sigma_Z^2\alpha^h}{1-\alpha^2}$ for $h \geq 0$. The covariance and correlation between Y_{f_1} and Y_{f_2} are

$$\mathrm{Cov}(Y_{f_1}, Y_{f_2}) = \frac{1}{d_{f_1}d_{f_2}} \sum_{i=0}^{e_{f_1}} \sum_{j=s_{f_2}}^{e_{f_2}} \gamma(|j-i|),$$

and

$$\mathrm{Corr}(Y_{f_1}, Y_{f_2}) = \frac{1}{d_{f_1}d_{f_2}\sigma_{Y_{f_1}}\sigma_{Y_{f_2}}} \sum_{i=0}^{e_{f_1}} \sum_{j=s_{f_2}}^{e_{f_2}} \gamma(|j-i|), \tag{3.1}$$

where $\sigma_{Y_{f_1}}$ and $\sigma_{Y_{f_2}}$ are the standard deviations of throughputs of f_1 and f_2, respectively. The standard deviation of the throughput of f with $s_f = 0$ is given by

$$\sigma_{Y_f} = \sqrt{\mathrm{Cov}(Y_f, Y_f)} = \sqrt{\frac{1}{d_f^2}\left(\sum_{i=0}^{e_f}\sum_{j=0}^{e_f}\gamma(|j-i|) + \sigma_W^2\right)}. \tag{3.2}$$

I compute the standard deviation of flow throughput as a function of flow duration based on first-order models with $\sigma_Z^2 = 1$, $\sigma_W^2 = 0$ and $\sigma_W^2 = 15$, and $\alpha = 0.3$ and $\alpha = 0.8$. Figs. 3.3 and 3.4 show that throughputs of long flows have smaller standard deviation than those of short flows. Moreover, the increase in throughput standard deviation due to the inability of a flow to react to congestion instantaneously becomes negligible for long flows. These results agree with the observations reported for the throughputs of small and large flows in [34] and [35].

To illustrate the behavior of (3.1) with different flow durations and different amounts of temporal overlap between the two flows, I set $\alpha = 0.5$, $\sigma_Z^2 = 1$, and $\sigma_W^2 = 0$ (no noise), and in Figs. 3.5 and 3.6, exhibit the correlation as a function of s_{f_2} for different d_{f_2} values when $e_{f_1} = 20$ and $e_{f_1} = 30$. One can conclude that the correlation is largely determined by the amount of temporal overlap between flows 1 and 2, o_{12}, relative to the duration of the longer flow, or

$$o_{12}^* = \frac{o_{12}}{\max\{d_{f_1}, d_{f_2}\}},$$

where $o_{12} = \max\{e_{f_1} - s_{f_2}, 0\}$. The figures also show that the location of the overlap of a short flow with a longer flow affects the correlation. For example, in Fig. 3.5 the correlation corresponding to $d_{f_2} = 10$ starts at 0.75 when the start times of the two flows are aligned, increases to 0.79 when $s_{f_2} = 4$ and $s_{f_2} = 6$, and then decreases.

For flows with short sojourn times, the “noise” terms decrease the cor-

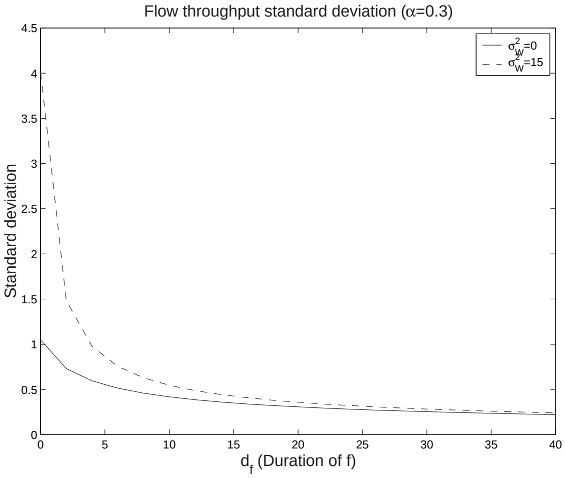

Figure 3.3: Standard deviation of flow throughput as a function of flow duration in (3.2) with $\alpha = 0.3$, and $\sigma_W^2 = 0$, $\sigma_W^2 = 15$.

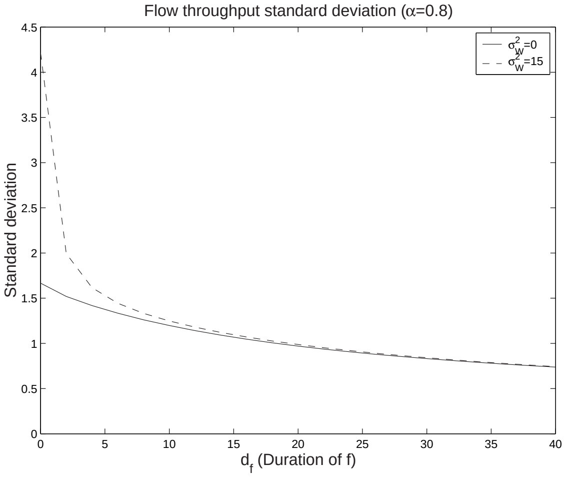

Figure 3.4: Standard deviation of flow throughput as a function of flow duration in (3.2) with $\alpha = 0.8$, and $\sigma_W^2 = 0$, $\sigma_W^2 = 15$.

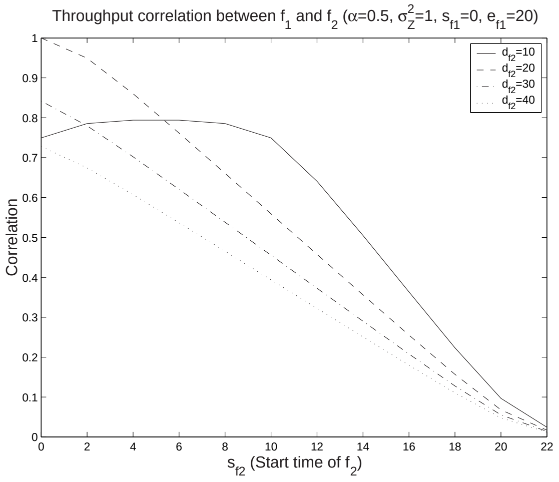

Figure 3.5: The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 0$. Flow 1 starts at time 0 and ends at time 20.

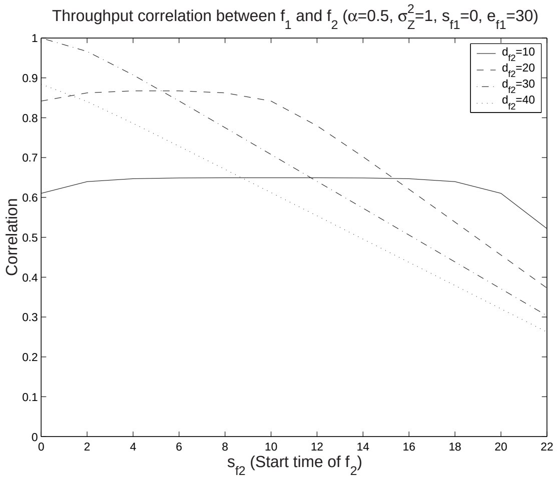

Figure 3.6: The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 0$. Flow 1 starts at time 0 and ends at time 30.

relation in (3.1) even with o^*_{12} close to 100%. By (3.1), for $\sigma^2_W > 0$

$$\frac{\displaystyle\sum_{i=0}^{e_{f_1}} \sum_{j=s_{f_2}}^{e_{f_2}} \gamma(|j-i|)}{\sqrt{\displaystyle\sum_{i=0}^{e_{f_1}} \sum_{j=0}^{e_{f_1}} \gamma(|j-i|) + \sigma^2_W}\sqrt{\displaystyle\sum_{i=s_{f_2}}^{e_{f_2}} \sum_{j=s_{f_2}}^{e_{f_2}} \gamma(|j-i|) + \sigma^2_W}}$$

is less than

$$\frac{\displaystyle\sum_{i=0}^{e_{f'_1}} \sum_{j=s_{f'_2}}^{e_{f'_2}} \gamma(|j-i|)}{\sqrt{\displaystyle\sum_{i=0}^{e_{f'_1}} \sum_{j=0}^{e_{f'_1}} \gamma(|j-i|) + \sigma^2_W}\sqrt{\displaystyle\sum_{i=s_{f'_2}}^{e_{f'_2}} \sum_{j=s_{f'_2}}^{e_{f'_2}} \gamma(|j-i|) + \sigma^2_W}}$$

for two other flows f'_1 and f'_2 with $s_{f'_1} = s_{f_1}$, $s_{f'_2} = s_{f_2}$, $d_{f'_1} = d_{f_1} + \epsilon$ and $d_{f'_2} = d_{f_2} + \epsilon$, with $\epsilon > 0$ and $\epsilon \rightarrow 0$. Figs. 3.7 and 3.8 show the effect of having $\sigma^2_W = 15$. Note that the correlation corresponding to $s_{f_1} = s_{f_2}$ and $d_{f_1} = d_{f_2} = 20$ in Fig. 3.7 is 0.84, whereas the correlation corresponding to $s_{f_1} = s_{f_2}$ and $d_{f_1} = d_{f_2} = 30$ in Fig. 3.8 is 0.89. Fig. 3.9 illustrates the effect of noise, $\sigma^2_W = 15$, on the degree of throughput correlation between perfectly overlapping flows; i.e. $s_{f_1} = s_{f_2}$ and $e_{f_1} = e_{f_2}$. Both flows start at time 0, and their end times correspond to durations of the flows. The degree of positive correlation is low for "noisy" short flows even when they overlap perfectly. As the duration of perfectly overlapping flows increase, throughput correlation approaches to 1.

In summary, the throughput samples associated with long flows that have large amounts of temporal overlap will result in high throughput corre-

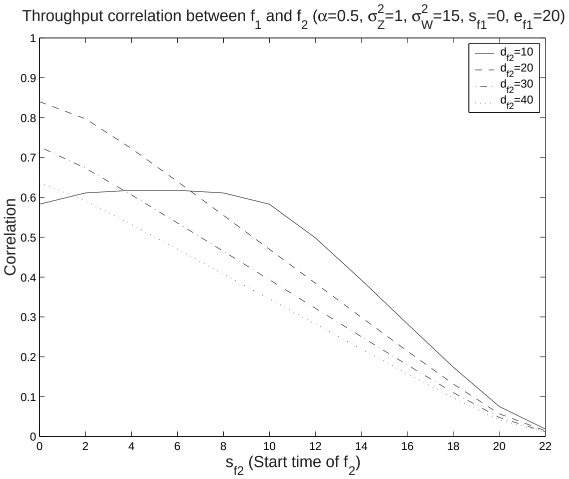

Figure 3.7: The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 15$. Flow 1 starts at time 0 and ends at time 20.

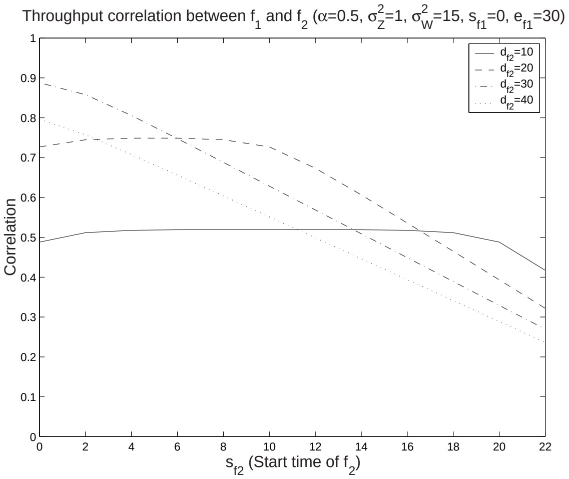

Figure 3.8: The effect of flow duration and temporal overlap on the correlation in (3.1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 15$. Flow 1 starts at time 0 and ends at time 30.

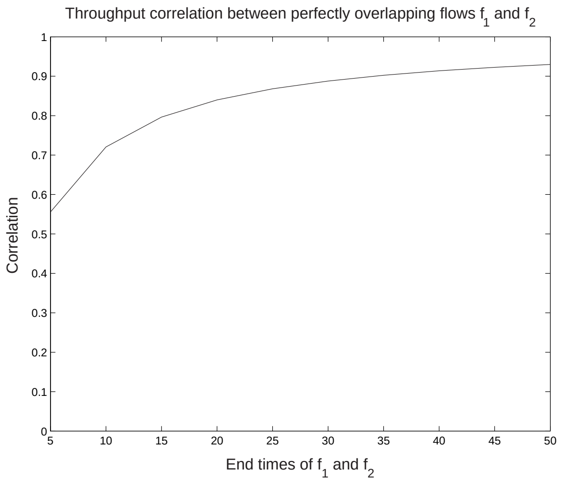

Figure 3.9: The effect of noise, $\sigma_W^2 = 15$, on the correlation in (3.1) between throughputs of perfectly overlapping f_1 and f_2 that share a congested resource. Both flows start at time 0.

lations. However, such samples occur rarely, because there are not too many long flows in the current Internet [36–38]. Moreover, the throughput samples associated with long flows overlapping with short flows will give a lower value for throughput correlation. Hence, leaving out long flows is desirable when estimating throughput correlations. On the other hand, the throughput samples associated with short flows are noisy, and will also give a lower value for throughput correlation even with flow samples that have large amounts of temporal overlap relative to their durations. Since flows with long sojourn times will typically be large (in size), I study the effect of different size thresholds to filter out large flows, and similarly, consider the impact of different size thresholds for omitting small flows. Unlike the duration of a flow, the size of a flow is invariant regardless of the capacity of links. Hence, flow size is the proper flow attribute to consider for filtering out flows.

3.3 First- and Second-Order Statistics of Flow Class Throughputs

Recall that a flow class is defined as a collection of flows that have common attributes. For example, flows that are destined to the same subnet address may be considered as a flow class. In order to estimate throughput correlations due to resource sharing, I consider temporal observations of throughputs of flow classes. In general, there is also a large number of flows from a given class at a given time. As such, I define the *throughput of a flow class* $c \in \mathcal{C}$ as an average over the flows in that class that are active at a time t at a measurement point. The flow class throughput at time t is given by

$$y_c(t) = \begin{cases} \dfrac{1}{|\mathcal{F}_c(t)|} \displaystyle\sum_{f \in \mathcal{F}_c(t)} y_f, & \text{if } |\mathcal{F}_c(t)| > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Fig. 3.10 illustrates how to compute flow class throughputs using an example with two flow classes.

I begin by computing the correlations among flow class throughputs using *temporal flow class throughput observations* at times when *all* of the flow classes are active. Note that the requirement of this *conditional sampling* strategy is a stringent one, especially when there are only a few flows belonging to a flow class under consideration. This requirement will lead to the retention of only a few throughput observations for statistical analysis. However, I choose to impose this stringent condition to guarantee positive definiteness

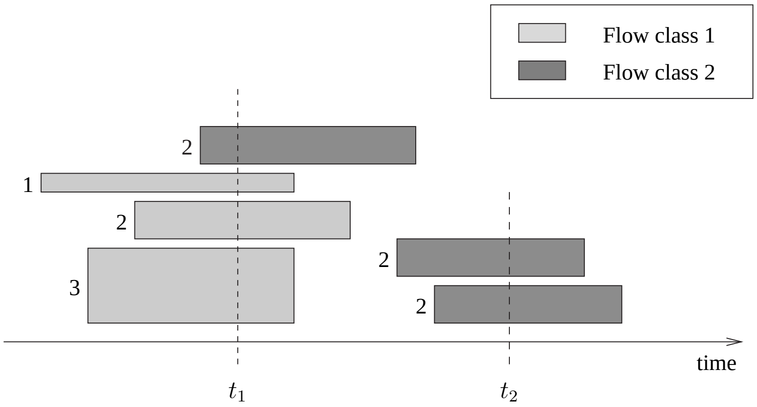

Figure 3.10: A collection of flows that belong to two different classes. The flow sojourn times are given by their lengths. The numbers next to the flows (and the widths of flows) indicate the throughputs y_f perceived by flows. The throughput of flow class 1 is (1+2+3)/3=2 at t_1. The throughput of flow class 2 is 2 at t_1. Flow class 1 is inactive at t_2. The throughput of flow class 2 is (2+2)/2=2 at t_2.

of the flow class throughput correlation matrix that will be used in Section 3.4. Note that the principal component method that was described to estimate factor loadings and specific variances interpreted the eigenvalues of the correlation matrix as variances of factors (see Section 2.3.3). The variances must be positive quantities. I will postpone the discussion of estimation of the correlation matrix of flow class throughputs by using a pairwise correlation strategy to Section 3.5.

The observation time is divided into discrete intervals[4]. Denote the

---

[4] Throughout this dissertation, one-second intervals are used. I find that using finer scale intervals does not affect the reported results. Using intervals that are longer than one second will reduce the number of samples of flow class throughputs, thereby producing less accurate results.

number of discretized time intervals for a measurement period by T and the number of discretized intervals over which all flow classes are active by $N(T)$. I assume that the throughput of a flow at a discretized time interval is equal to its "continuous-time" throughput if the flow is active anytime during that interval. I also assume that $\{y_{c_i}(n)\}$ and $\{y_{c_j}(n)\}$, $n = 1, \ldots, T$, are realizations of ergodic random processes of throughputs of flow classes c_i and c_j, respectively (on discretized intervals). Define the event $S(n) = \{y_{c_i}(n) > 0, \forall c_i \in \mathcal{C}\}$. The *conditional* mean and variance of the throughput for flow class c_i are defined as

$$\mu_{c_i} = \lim_{T \to \infty} \frac{1}{N(T)} \sum_{n=1}^{T} y_{c_i}(n)\, \mathbf{1}_{S(n)}, \tag{3.4}$$

$$\sigma_{c_i}^2 = \lim_{T \to \infty} \frac{1}{N(T)} \sum_{n=1}^{T} (y_{c_i}(n) - \mu_{c_i})^2\, \mathbf{1}_{S(n)}, \tag{3.5}$$

where $\mathbf{1}_E$ is the standard indicator function, which is equal to 1 if E is true and 0, otherwise. The conditional correlation of throughputs of flow classes c_i and c_j is defined as

$$\rho_{c_i c_j} = \lim_{T \to \infty} \sum_{n=1}^{T} \frac{(y_{c_i}(n) - \mu_{c_i})(y_{c_j}(n) - \mu_{c_j})\, \mathbf{1}_{S(n)}}{N(T) \sigma_{c_i} \sigma_{c_j}}. \tag{3.6}$$

3.4 Conditioned Throughputs of Flow Classes

In order to be able to use factor analysis in a classical setting, I introduce vectors of random variables corresponding to flow class throughputs. Let $f_{\mathbf{Y}}$ denote the joint pdf of the random vector, $\mathbf{Y} = (Y_{c_1}, Y_{c_2}, \ldots, Y_{c_p})^T$, of typical class throughputs for p flow classes at a typical time; i.e., $\mathbf{Y} \sim f_{\mathbf{Y}}$. Note that it is possible that $Y_{c_i} = 0$ for any $i = 1, \ldots, p$; i.e., no flow from class c_i is active at a typical time. Next, define $\mathcal{E} = \{Y_{c_i} > 0, \forall c_i \in \mathcal{C}\}$. Also, define a random vector of flow class throughputs conditioned on $\mathcal{E}$, $\mathbf{Y}^* = (Y^*_{c_1}, Y^*_{c_2}, \ldots, Y^*_{c_p})^T$, with a joint pdf $f_{\mathbf{Y}|\mathcal{E}}$, i.e., $\mathbf{Y}^* \sim f_{\mathbf{Y}|\mathcal{E}}$. Denote the mean vector of $\mathbf{Y}^*$, $\mathbb{E}[\mathbf{Y}^*]$, by $\boldsymbol{\mu}_{\mathbf{Y}^*} = (\mu_{c_1}, \mu_{c_2}, \ldots, \mu_{c_p})^T$.

Factor analysis that was described in Section 2.3 can be directly applied to the conditional correlation matrix $\mathbf{R}$ given by

$$\mathbf{R} = \mathrm{Corr}(\mathbf{Y}^*) = (\rho_{c_i c_j}) = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}, \quad i, j = 1, \ldots, p \tag{3.7}$$

where $\rho_{c_i c_i} = 1$, and $\rho_{c_i c_j}$ is given by (3.6). For factor analysis, use of a correlation matrix is preferred to a covariance matrix because the magnitudes of flow class throughputs can vary greatly and normalizing such measurements is required.

3.5 Estimation of the Correlation Matrix from Pairwise Correlations

One drawback of the development until this point is that all of the flows must be active at a given time to contribute an observation of random vector $\mathbf{Y}^*$ to the estimate of the first- and second-order statistics in (3.4)–(3.6). When there are only a few flows belonging to a flow class under consideration, only a few class throughput observations are retained for statistical analysis. To address this problem, I compute pairwise correlations between variables to estimate the correlation matrix for flow class throughputs. In order to compute the pairwise correlations, I use the class throughput observations at times when the flow class pair is active. Pairwise correlations are employed in multivariate statistics when there are “missing values” for one or more variables in a significant number of observation vectors (see for example, [31] and [39]). I adopt this approach to compute correlations since there are a lot of sampling instants when not all of the variables can be manipulated simultaneously: Instead of having missing values, simply, no flow from a given flow class is active at that instant. Since the correlation matrix constructed in this way may not always be positive definite, the matrix can be adjusted to make it positive definite. For example, as proposed in [40], a constant may be added to the non-positive eigenvalues of $\mathbf{R}$ to make them positive.

Define the event $P(n, c_i, c_j) = \{y_{c_i}(n) > 0 \text{ and } y_{c_j}(n) > 0, \text{ for } c_i, c_j \in \mathcal{C}\}$; i.e. both classes c_i and c_j are active at (discretized) time n. Let $N(T)$ now be the number of discretized intervals over which both flow classes c_i and c_j are active, where T denotes the number of discretized time intervals of a measurement period. I estimate the correlation between throughputs of c_i and c_j by replacing $S(n)$ by $P(n, c_i, c_j)$ in (3.4) and (3.5) to obtain the corresponding conditional *pairwise* mean and variance:

$$\mu_{c_i,c_j} = \lim_{T\to\infty} \frac{1}{N(T)} \sum_{n=1}^{T} y_{c_i}(n)\, \mathbf{1}_{P(n,c_i,c_j)},$$

$$\sigma^2_{c_i,c_j} = \lim_{T\to\infty} \frac{1}{N(T)} \sum_{n=1}^{T} (y_{c_i}(n) - \mu_{c_i,c_j})^2\, \mathbf{1}_{P(n,c_i,c_j)},$$

where $\mathbf{1}_E$ is the standard indicator function defined before, and is equal to 1 if E is true and 0, otherwise. The conditional correlation of throughputs of flow classes c_i and c_j is now defined as (cf. the correlation in (3.6))

$$\rho_{c_i c_j} = \lim_{T \to \infty} \sum_{n=1}^{T} \frac{(y_{c_i}(n) - \mu_{c_i,c_j})(y_{c_j}(n) - \mu_{c_j,c_i})\, \mathbf{1}_{P(n,c_i,c_j)}}{N(T)\sigma_{c_i,c_j}\sigma_{c_j,c_i}}. \tag{3.8}$$

Hence, a pairwise correlation matrix that is constructed by using pairwise correlations between variables is given by

$$\mathbf{R} \approx (\rho_{c_i c_j}), \quad i, j = 1, \ldots, p, \tag{3.9}$$

where $\rho_{c_i c_i} = 1$, and $\rho_{c_i c_j}$ is given by (3.8). Henceforth, I will approximate the correlation matrix of flow class throughputs in (3.9) by using the pairwise estimates of correlations.

3.6 Simulations Using Fluid Models

While the claim that flow classes that share congested resources perceive high or poor quality of service (throughput) together is intuitive, it may be helpful to explain such correlations by describing how bandwidth is shared in models that approximate rate control mechanisms such as TCP. Bandwidth sharing mechanisms try to use the available resource capacities to the fullest extent while maintaining a certain fairness criterion (to be made specific later in this section) when making bandwidth allocations to flows. The *rate* of a flow is then the bandwidth share allocated to it at a given time. These sharing mechanisms explain the existence of positively correlated throughputs for classes sharing a congested resource.

In this section, I use known, analytical fluid models to generate flow records by simulation. The fluid models are used to determine the bandwidth shares [34, 35] achieved by flows at a given time. In such models, the rate or bandwidth allocated to a flow is adjusted *instantaneously* when the number of flows in the system changes as a result of flow arrivals and departures. Note that the rate variation of a flow during its sojourn will not be available from its flow record. The dynamic bandwidth sharing model approximates actual rate control mechanisms (such as TCP) well due to the assumption of separation of time scales: The time scale of flow durations is much longer than the time scale on which rate control mechanisms converge to equilibrium. Bandwidth sharing among flows is considered first on a single resource and then in a "linear" network.

3.6.1 The M/GI/1-PS queue

The simplest abstraction in which the number of flows on a single resource (such as a link) can be modelled is an M/GI/1-PS queue, i.e., a single-server processor sharing queueing system with an exponential interarrival time distribution and a general, independent service time distribution [41]. That is, if all flows share similar round trip times and packet loss rates, the resource bandwidth w is shared equally among the active flows. Hence, the instantaneous bandwidth allocated to a flow on the resource is given by

$$b_f(t) = \begin{cases} \dfrac{w}{|\mathcal{A}(t)|}, & \text{if } |\mathcal{A}(t)| > 0 \text{ and } f \in \mathcal{A}(t), \\ 0, & \text{otherwise,} \end{cases} \tag{3.10}$$

where

$$\mathcal{A}(t) = \bigcup_{c \in \mathcal{C}} \mathcal{F}_c(t) \tag{3.11}$$

denotes flows from all classes that are active at a given time t in this single resource system. Then, the *perceived throughput* y_f for a flow f is given by

$$y_f = \frac{1}{d_f} \int_{s_f}^{e_f} b_f(t) dt. \tag{3.12}$$

While (3.10) is not available from flow measurements, y_f in (3.12) can easily be obtained from flow records (see Section 3.1).

A parallel collection of queues, as shown in Fig. 3.11, constitutes one of the simplest test cases to investigate class throughput correlations.

3.6.2 A linear network

In reality, throughputs of flow classes depend on the traffic on all resources. Fig. 3.12 illustrates a linear network model with two resources. Such a network can be used to model flows traversing several links that interact with the cross traffic on these links. The linear network model enables us to investigate the coupling effects between flows following multi-link paths (e.g., class 1, denoted c_1) and cross traffic (e.g., class 2, denoted c_2 and class 3, denoted c_3), and possibly between flow classes not sharing a link (e.g., c_2 and c_3). To determine the character of bandwidth sharing among flows on a linear network, proportionally fair sharing is considered [42, 43][5].

---

[5] More details on various bandwidth sharing models can be found in [42–45].

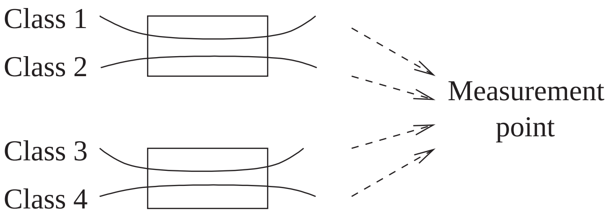

Figure 3.11: Two parallel M/GI/1-PS queues. All four flow classes traverse the measurement point.

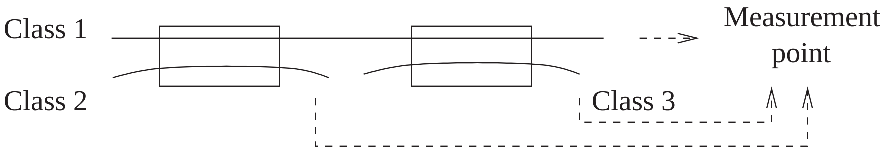

Figure 3.12: A linear network with two resources. All three flow classes traverse the measurement point.

Let $\mathcal{L}$ denote the set of resources, each with capacity w_l, $l \in \mathcal{L}$, in the linear network. At a given time t, the bandwidth allocated to each flow, $b_f(t)$, must be such that the following capacity constraint is satisfied:

$$\sum_{f \ni l, f \in \mathcal{A}(t)} b_f(t) \leq w_l, \forall l \in \mathcal{L}. \tag{3.13}$$

Here, $\mathcal{A}(t)$ denotes flows from all classes in the network that are active at a given time t, and is given by (3.11).

We say that the bandwidth allocations are proportionally fair if they maximize $\sum_{f \ni l, f \in \mathcal{A}(t)} \log b_f(t)$ subject to the capacity constraint in (3.13). In other words, proportionally fair sharing maximizes the overall utility of bandwidth allocations when the utility functions are logarithmic.

Denote the number of flows belonging to class c at time t by $n_c(t)$. For the network in Fig. 3.12, the proportionally fair allocation with unit capacity resources is [34]:

$$b_f(t) = \begin{cases} 0, & \text{if } f \notin \mathcal{A}(t), \\ \dfrac{1}{n_{c_1}(t) + n_{c_2}(t) + n_{c_3}(t)}, & \text{if } f \in \mathcal{F}_{c_1}(t), \\ \dfrac{1}{n_{c_i}(t)} \left(1 - \dfrac{n_{c_1}(t)}{n_{c_1}(t) + n_{c_2}(t) + n_{c_3}(t)}\right), & \\ & \text{if } f \in \mathcal{F}_{c_i}(t), i \geq 2. \end{cases} \tag{3.14}$$

Hence, the instantaneous bandwidth allocation for flows depends on the number of flows on every link. As before, the perceived throughput y_f of a flow is given by (3.12).

3.6.3 Simulation setup

In each case, it is assumed that the flows in class c arrive according to a Poisson process with rate λ_c. The sizes of flows are chosen from a lognormal distribution [37, 38]. A random variable V has a lognormal distribution if the random variable $U = \ln V$ has a normal distribution. Hence, the pdf of the lognormal distribution is given by

$$f(v|\eta, \tau) = \frac{1}{\sqrt{2\pi}\tau v} e^{-(\ln v - \eta)/2\tau^2}, \tag{3.15}$$

where η is the mean, and τ is the standard deviation of the associated normal variable U. Let $\beta = \mathbb{E}[V]$ and $\sigma^2 = \mathrm{Var}(V)$. In the simulations, I set $\beta = 1$ and $\sigma = 10$. Flow sizes selected from the lognormal distribution consist of a large number of small flows and a few very large flows (as may be the case in the current Internet [36–38]).

In the fluid simulations, I choose unit capacity resources; i.e., I set $w = 1$ for each resource. Denote the load offered by a class c_i on a resource by $\nu_{c_i} = \lambda_{c_i}\beta$. The long-term proportion of time the resource is busy, or the utilization factor of the resource, ν, is given by the sum of loads offered by all classes traversing that resource. As usual, assume $\nu < 1$ for stability [46] and $\nu \rightarrow 1$ indicates that a resource is congested and hence is a bottleneck. I vary the utilization factor of a resource by varying the arrival rates of flows. For the parallel, processor sharing queues, I allocate bandwidth to flows according to (3.10), and in the linear network, the bandwidth is allocated to flows according to (3.14). The throughput of each flow is determined by the variable bandwidth allocated to it during its sojourn in the network. After a flow is terminated, its record is created for subsequent processing.

3.6.4 Simulation results and discussion

For two parallel M/GI/1-PS queues, I first determine the correlation between throughputs of classes sharing a queue (class 1 and class 2) and the correlation between throughputs of classes not sharing a queue (class 1 and class 3) as a function of queue utilization. Both queues are kept at the same utilization factors so that any observed correlations can be attributed to resource sharing only.

Figs. 3.13–3.15 illustrate the effect of filtering out large flows on pairwise correlations and on percent variance accounted by the common factors. The plots labelled as “original” correspond to cases in which all flows in the original set are included in class throughput computations. The plots that correspond to retaining flows only in $\{f \in \mathcal{F} : v_f < v_{th}\}$ with $v_{th} = 0.5, 1$, and 2, respectively, are labelled accordingly.

From Fig. 3.13, we observe that the throughputs of classes sharing a congested queue exhibit strong positive correlation, whereas from Fig. 3.14, we can conclude that throughputs of classes not sharing a queue are only very weakly correlated (or uncorrelated). From Fig. 3.13, we observe the benefit of filtering out flows whose sizes are greater than a given threshold. Even at low utilization, the degree of positive correlation is very high after discarding large flows whose throughputs distort the structure of correlation among flows sharing resources. By decreasing the threshold (above which the flows are filtered out), we obtain higher correlations for classes sharing a resource. If we filter out too many flows by further decreasing the threshold, then the number of samples available for statistical analysis will decrease, thereby reducing the accuracy of the estimates. Fig. 3.14 shows that filtering out long flows does not significantly alter the uncorrelated nature of flow throughputs not sharing resources.

Using the modified Kaiser's rule described in Section 2.3.3, I correctly identify two significant factors when queue utilizations are greater than 30%. Fig. 3.15 shows the percentage of variance accounted by significant factors and the number of significant factors for different utilization factors. A "high" percentage indicates a strong explanatory power of factors. The explanatory power increases with the utilization factors: At higher utilizations, flow

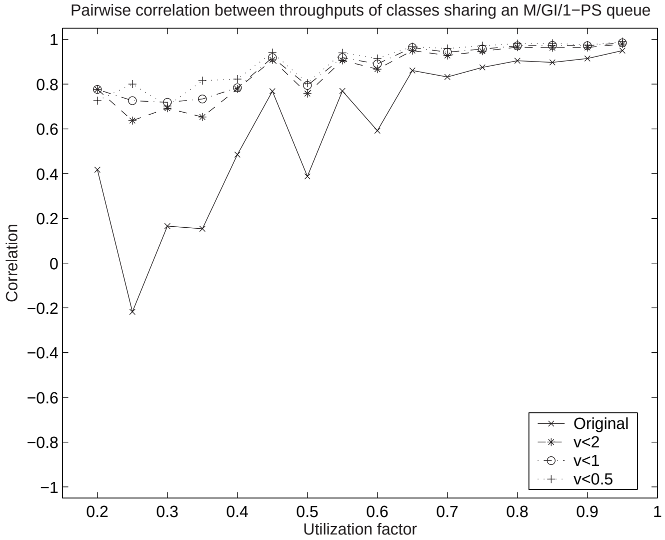

Figure 3.13: Pairwise correlation between throughputs of flow classes (class 1 and class 2) sharing an M/GI/1-PS queue in Fig. 3.11.

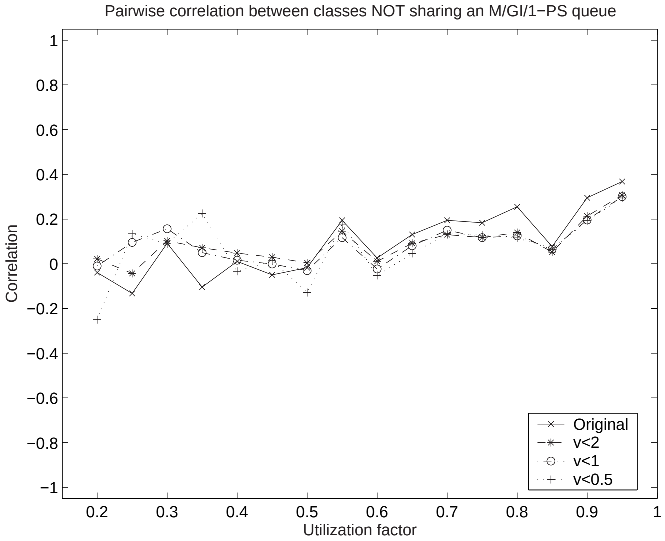

Figure 3.14: Pairwise correlation between throughputs of flow classes (class 1 and class 3) not sharing an M/GI/1-PS queue in Fig. 3.11.

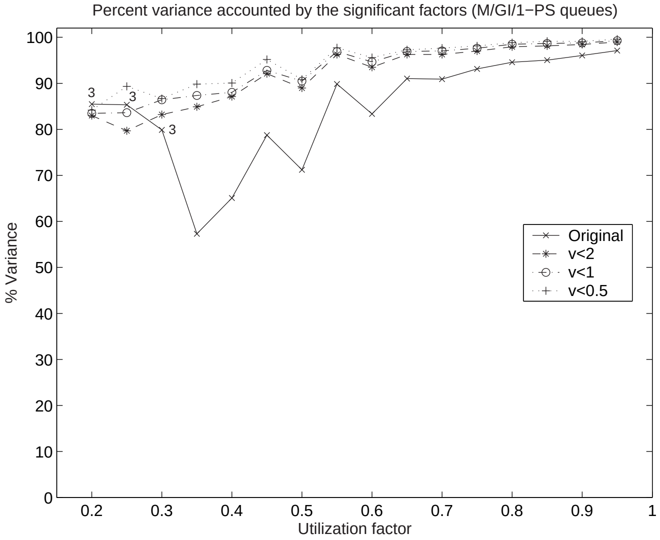

Figure 3.15: Percent variance accounted by significant factors in parallel M/GI/1-PS queues in Fig. 3.11. The points labelled with a 3 on the solid-line plot correspond to the experiments in which three, instead of two, common factors were (incorrectly) identified.

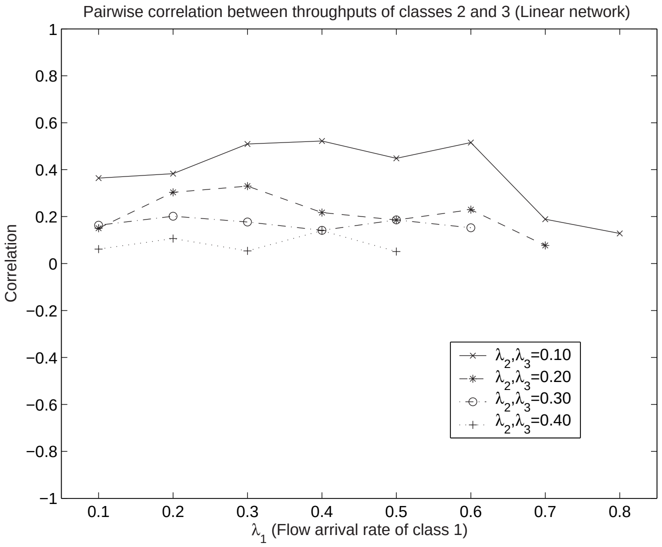

Figure 3.16: Pairwise correlation between throughputs of classes 2 and 3 coupled by class 1 in the linear network in Fig. 3.12. Correlations are estimated after filtering out flows with sizes larger than 2.

throughputs are largely determined by the bottlenecked resources. Without filtering out large flows, the number of significant factors (determined by rules described in Section 2.3.3) for utilization factors 20%, 25%, 30% are incorrectly determined to be 3 instead of 2 (labelled by 3 in Fig. 3.15). Therefore, filtering out large flows also improves our ability to extract significant factors from the correlation matrix.

Fig. 3.16 shows the degree of correlation between classes not sharing a resource but whose throughputs are coupled by a flow class traversing both resources in the linear network in Fig. 3.12. The results shown are based on filtering out flows whose sizes are larger than 2. Each correlation estimate shown in the figure is the average of five simulation runs. The arrival rate of flow class 1 is varied to analyze the effect of the coupling flow class. Different plots in Fig. 3.16 correspond to different arrival rates from classes 2 and 3. The positive correlation between throughputs of classes 2 and 3 is negligible and starts to increase only when the loads offered by classes 2 and 3 are low and when the load offered by class 1 is comparable to those of classes 2 and 3. Such coupling effects can be ignored if we suppose that class 2 corresponds to internal traffic on the first link belonging to one network, class 3 corresponds to internal traffic on the second link belonging to another network, and class 1 corresponds to transit traffic carried by these networks. In general, the proportion of transit traffic that shares a link with internal traffic should be low.

3.7 Conclusion

This chapter introduced a new approach to network tomography problems that involve inferring resource sharing by correlating flow level network traffic measurements. First, by using a model based on an AR(1) process, I demonstrated the degree of correlation between two temporally overlapping flows that share a resource. Based on this first-order model, I explained why filtering out large- and/or small-sized flows helped in capturing throughput correlations due to resource sharing. Then, I described a sampling strategy for multiple random processes, or random functions of time, to extract the correlation structure that exists among flow class throughputs conditioned upon their simultaneous activity in the system. I applied this sampling strategy to simulation data generated by known, analytical fluid models to demonstrate the positive correlation among flow classes that share congested resources in the network.

In the next chapter, factor analysis is applied to flow class throughput correlation matrices obtained via TCP simulations. The results will provide an evaluation of the feasibility of this approach for TCP/IP networks.

Chapter 4

Inferring Resource Sharing Among TCP Flow Classes: Simulation and Analysis

4.1 Introduction

This chapter presents results based on part of an extensive set of OPNET Modeler [47] simulations that incorporate actual characteristics of TCP flows. The primary aim of this chapter is to validate the methods introduced in Chapter 3 for identifying resource sharing flow classes in a controlled environment in which the routes from the sources to destinations are known exactly. The effectiveness of factor analytic methods in identifying such flow classes under different traffic conditions and different network configurations is evaluated.

Section 4.2 briefly summarizes TCP's congestion control mechanism. Section 4.3 describes results of applying factor analysis to infer flow classes that share congested resources in networks with tree topologies. While actual networks rarely look like trees, tree topologies have been frequently used in network tomography research. Trees may provide a good abstraction for logical topologies. For example, each tree branch can represent a link that may potentially become a bottleneck in the actual network, while overprovisioned links are not included in the tree. Section 4.4 revisits the linear network topology that was first introduced in Chapter 3. In a linear network, I show that factor analysis can identify the two bottleneck resources that are visited by a flow class. Section 4.5 describes an application of the developed methodology to wireless local area networks for investigating proper configuration of wireless access point for the traffic and spatial access patterns of wireless users. Section 4.6 concludes the chapter.

4.2 A Brief Review of TCP's Congestion Control

Unlike fluid models discussed in Chapter 3, actual TCP flows are packetized (or, alternatively, the packet sizes are not infinitesimally small as assumed by the fluid models). This means that the notion of "rate" of a flow needs to be redefined. The amount of data in transit from a source at a given time is limited by the size of its congestion window (CWND) [11] to prevent congestion in the network. A source can transmit CWND bytes per round trip time (RTT), which is the time between sending a packet and receiving the corresponding acknowledgement from the destination. We can say that at a given time the "rate" of a TCP flow is approximately $\frac{\text{CWND}}{\text{RTT}}$.

TCP implements an end-to-end congestion control algorithm. Each source determines the available capacity in the network from the acknowledgement packets received from the destination. When congestion due to flow level traffic dynamics causes packet losses inside the network, the destination will not acknowledge the missing packets. The source will then adjust its congestion window, and thus its transmission “rate”, to send fewer packets per RTT. As congestion levels decrease, there will be fewer packet losses, which indicates higher available capacity. The source will then increase its “rate”. For long TCP flows that share a congested resource, congestion control amounts to sharing the bandwidth at that resource.

The throughput of a (small) flow will be limited by TCP’s Slow Start [11] and the throughput of flows, in general, depends on the sizes of sender’s and receiver’s buffers. Moreover, the bandwidth allocated to a given TCP flow does not change instantaneously when the number of flows in the system changes as a result of flow arrivals and departures, as was the case in the fluid models. Nevertheless, I show that such imperfections do not affect our ability to infer resource sharing by correlating flow class throughputs which are computed based only on generated flow records.

4.3 Networks with Tree Topologies

Consider the tree topology shown in Fig. 4.1. Users download files from a server using FTP. Access links are denoted by A1, A2, and A3, and the link connecting the FTP server to the network is denoted by S1. Seven classes of flows are defined according to their local subnet addresses. Each subnet is a 10 Mbps local area network that has 10 workstations. Each simulation corresponds to 2 hours of file download activity. During simulations, I record the request time, size (in bytes) and duration (in seconds) of each file transfer. File transfer requests arrive according to a Poisson distribution, and file sizes

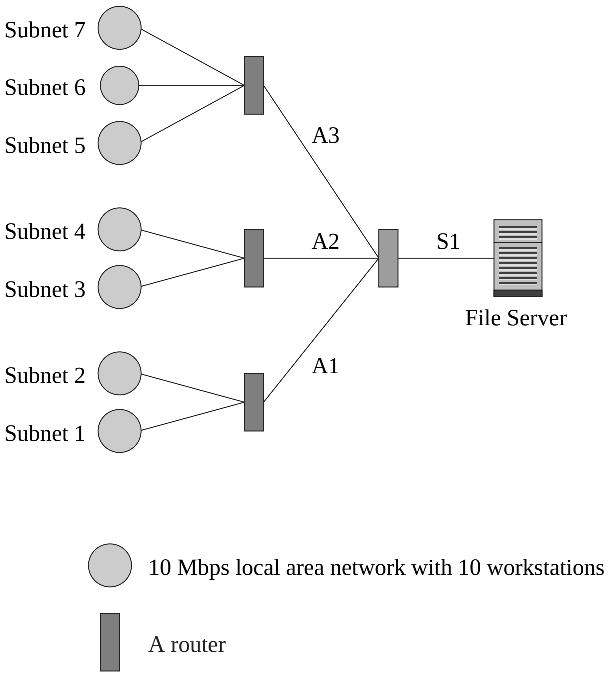

Figure 4.1: Tree topology used in OPNET TCP simulations.

are selected from a lognormal distribution (as was the case in the fluid model simulations) with a mean file size of 16 kB and a standard deviation of 131 kB. On each bottleneck link, I include background traffic to model the effects of additional traffic from other users or applications.

In order to be able to capture the throughput correlation between flow classes successfully, I consider the effect of flow sizes on correlation estimates. The results in Section 3.6.4 showed that removing large flows from flow records under consideration improved our ability to infer positive throughput correlation among resource sharing flow classes. Unlike fluid models, TCP takes some time to react to changes in the congestion state of the network, and the throughputs of very small flows are limited by TCP's Slow Start [11]. Therefore, small flows may not have an opportunity to "learn" the congestion state of the network during their sojourn time. This suggests that when estimating TCP class throughput correlations, small flows and large flows should be filtered out.

In the subsequent discussion, the lower and upper thresholds for filtering out flows are determined empirically. I will show the effects of retaining flows satisfying the following conditions: $\{f \in \mathcal{F} : v_f > 4\text{ kB}\}$, $\{f \in \mathcal{F} : v_f > 8\text{ kB}\}$, $\{f \in \mathcal{F} : v_f < 16\text{ kB}\}$, $\{f \in \mathcal{F} : v_f < 32\text{ kB}\}$, and $\{f \in \mathcal{F} : 4 < v_f < 32\text{ kB}\}$. As before, I refer to the results based on the entire set of flows (without filtering) as the "original".

4.3.1 Squared error loss

In order to assess the ability of factor loadings to distinguish which factors have the most effect on throughputs, I define *squared error loss* as

$$L := \|\mathbf{\Lambda}^0 - \mathrm{abs}(\hat{\mathbf{\Lambda}})\|^2 = \sum_{i=1}^{p}\sum_{j=1}^{m}(\Lambda_{ij}^0 - |\hat{\Lambda}_{ij}|)^2, \tag{4.1}$$

where $\Lambda_{ij}^0 = 1$ if the flow class c_i shares the factor j, and $\Lambda_{ij}^0 = 0$ otherwise, which correspond to "ideal" loadings in a matrix $\mathbf{\Lambda}^0$. When there is only one factor ($m = 1$), I set $\Lambda_{i1}^0 = 1$ for all i. The notation $\|\cdot\|$ denotes the Euclidian norm, and for a matrix, it is given by the square root of the sum of squares of each element in the matrix. The function $\mathrm{abs}(\cdot)$ returns a matrix whose elements are the absolute values of the corresponding elements in the input matrix. The primary motivation behind introducing the squared error loss is to assess the accuracy of our inference and to investigate various schemes to improve the accuracy[1]. I will say that the accuracy of the inference is high if the squared error loss is small. Note that squared error loss penalizes large deviations from the ideal more than small deviations. Next, I will analyze scenarios with different number of bottlenecks under various load conditions.

4.3.2 Single bottleneck: Effect of background traffic

Consider the case in which link S1 (1.544 Mbps) in Fig. 4.1 is the bottleneck and access links A1, A2, and A3 are overprovisioned (44.736 Mbps). The users belonging to classes 1–7 generate a total load of 30% on the bottleneck link and the bottleneck's background traffic utilization is varied from 40% to 65% to demonstrate the ability to identify this bottleneck under different background traffic conditions.

Using the modified Kaiser's rule described in Section 2.3.3, I correctly determine that there is one significant factor for each utilization factor considered. Fig. 4.2 shows that the explanatory power of this single factor increases as congestion increases on the bottleneck. Fig. 4.3 illustrates that at higher utilization levels of the bottleneck link, the accuracy of inference is higher. Both Figs. 4.2 and 4.3 show the effect of filtering out small flows, large flows, and small and large flows simultaneously on the percentage of normalized variance

---

[1] If comparisons *across* different bottleneck configurations are desired, one can divide squared error loss by $p \times m$, the number of elements in the loading matrix.

accounted by the significant factor and on the squared error loss. We see that filtering out small flows or large flows improves the explanatory power of the factor and decreases the squared error loss. Note that increasing (decreasing) the lower (higher) filtering threshold of flow sizes has significant benefits on these measures. However, increasing (decreasing) the lower (higher) threshold retains fewer flows and decreases the statistical accuracy of estimates. An important observation is that omitting *both* small and large flows simultaneously significantly improves the explanatory power of the factor and decreases the squared error loss. Retaining flows whose sizes are between 4 kB and 32 kB is a compromise between reliability of inference for resource sharing and statistical accuracy of estimates.

4.3.3 Single bottleneck: Effect of class loads

Consider again the case in which link S1 (1.544 Mbps) is the bottleneck and access links A1, A2, and A3 are overprovisioned (44.736 Mbps). I investigate the ability to identify this bottleneck for different total loads (20%, 30%, 40%) generated by users belonging to classes 1–7 on the bottleneck link S1. Background traffic utilizes 50% of link S1.

Using the modified Kaiser's rule, I successfully determine that there is one significant factor when total class loads on the bottleneck link are greater than or equal to 20%. Fig. 4.4 shows that the explanatory power of this factor increases as congestion increases on the bottleneck. Fig. 4.5 illustrates that at higher utilization levels, the accuracy of inference is higher. Again,

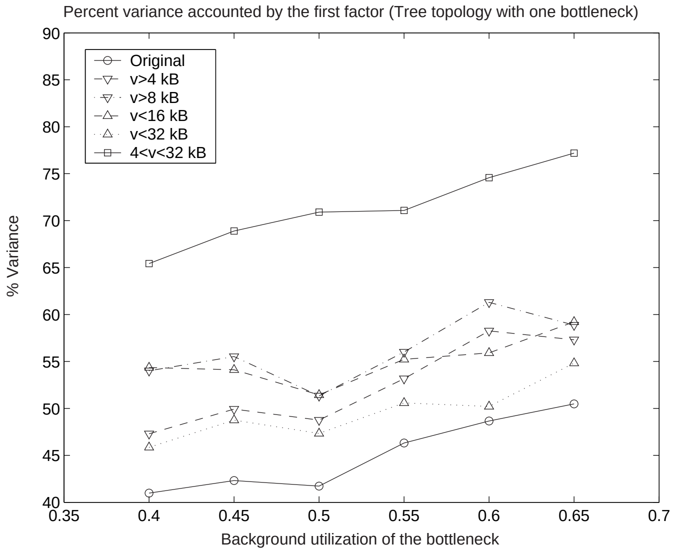

Figure 4.2: Percent normalized variance under different background traffic conditions on the single bottleneck. The total load due to classes 1–7 on S1 is kept at 30%.

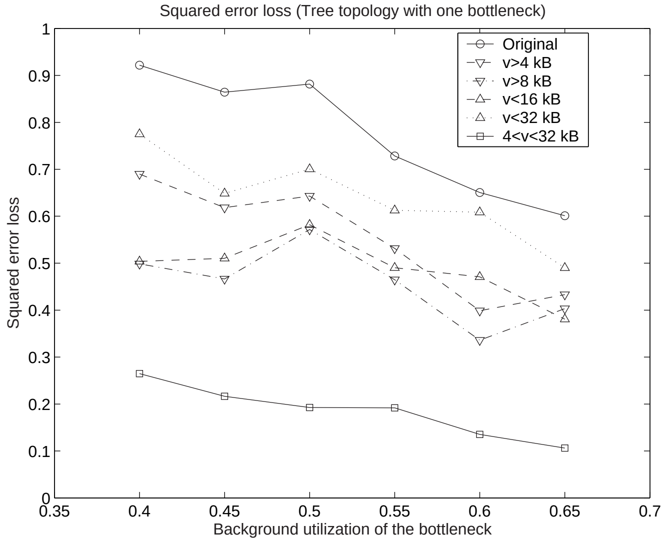

Figure 4.3: Squared error loss under different background traffic conditions on the single bottleneck. The total load due to classes 1–7 on S1 is kept at 30%.

omitting both small and large flows simultaneously significantly improves the explanatory power of the factor and decreases the squared error loss.

4.3.4 Single bottleneck: Effect of non-stationary traffic

I also investigate the effect of having non-stationary background traffic for the single bottleneck case described in the previous subsections. The background traffic utilization of the bottleneck link changes between 60% and 40% every 20 minutes over the period of 2 hours. Using the modified Kaiser's rule, I successfully determine that there is one significant factor with explanatory power 71%. For this particular scenario, non-stationarity of the network traffic does not seem to affect determination of resource sharing. Further analysis may be required to assess the impact of non-stationarity of network traffic on inference results.

4.3.5 Three bottlenecks

Consider the case in which links A1, A2, and A3 (each 1.544 Mbps) are bottlenecks and link S1 is overprovisioned (44.736 Mbps). I investigate the ability to identify these bottlenecks and *associate* each flow class with a bottleneck for different loads (10%, 15%, 20%) generated by *each* class.

The background utilization on A1 and A2 (each serving two subnets) is set to 50%. The background utilization on A3 (serving three subnets) is adjusted so as to keep the total utilization on each bottleneck (A1, A2, and A3) the same; i.e., the background utilization on A3 is 40%, 35%, and 30%

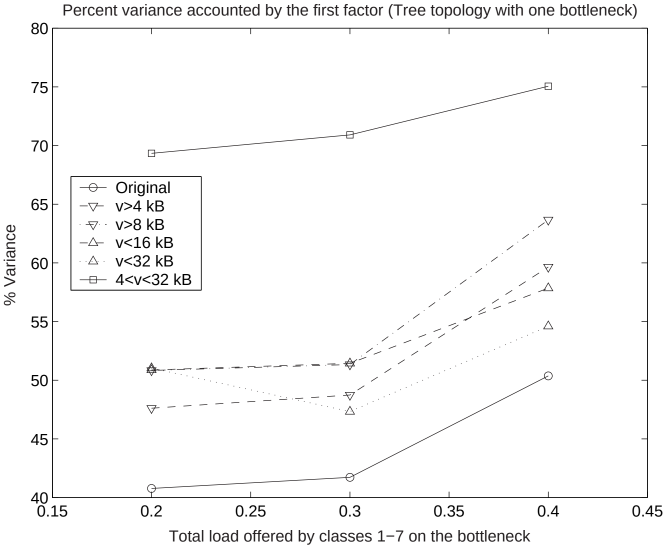

Figure 4.4: Percent normalized variance under different total loads from classes 1–7 on the single bottleneck S1. The utilization of S1 due to background traffic is kept at 50%.

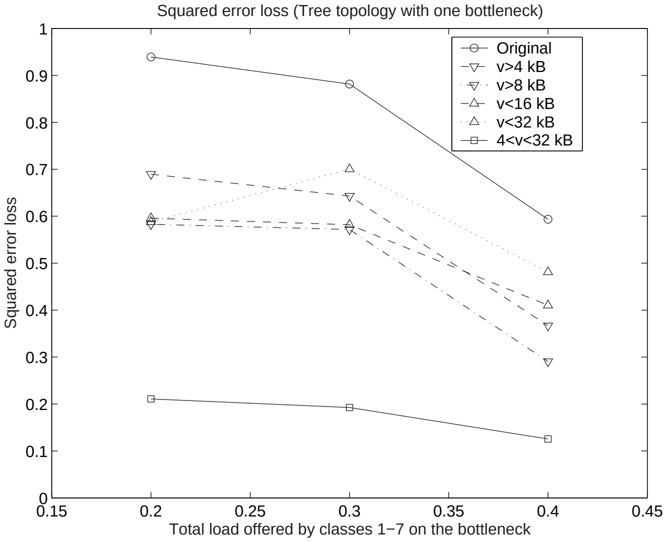

Figure 4.5: Squared error loss under different total loads from classes 1–7 on the single bottleneck S1. The utilization of S1 due to background traffic is kept at 50%.

corresponding to 10%, 15%, and 20% loads offered by each class, respectively. From the generated records, I successfully determine that there are three significant factors when loads generated by each class are greater than or equal to 10%. Furthermore, I correctly identify which flow classes share congested resources: I find that the throughputs of flow classes from subnets 1 and 2 have the largest loading with factor 1, the throughputs of flow classes from subnets 3 and 4 have the largest loading with factor 2, and the throughputs of flow classes from subnets 5, 6 and 7 have the largest loading with factor 3. These factors are interpreted as the access links A1, A2, and A3, respectively. Fig. 4.6 shows that the explanatory power of the three factors increases as congestion increases on the bottlenecks. Fig. 4.7 illustrates that at higher utilization factors, the factor loadings distinguish which factor a flow class is most associated with more easily. As in the previous cases, omitting both small and large flows simultaneously significantly improves the explanatory power of the factors and decreases the squared error loss.

4.4 Interaction of Coupled Flow Classes

Recall that Section 3.6.2 described a linear network topology. I argued that when the load offered by a flow class traversing multiple bottlenecks and interacting with the cross traffic in both resources was low, either no or only weak correlation was introduced between flow classes not sharing a resource due to the coupled bottlenecks. In this section, I show how factor analysis identifies bottlenecks on an example scenario given in Fig. 4.8. Users belonging

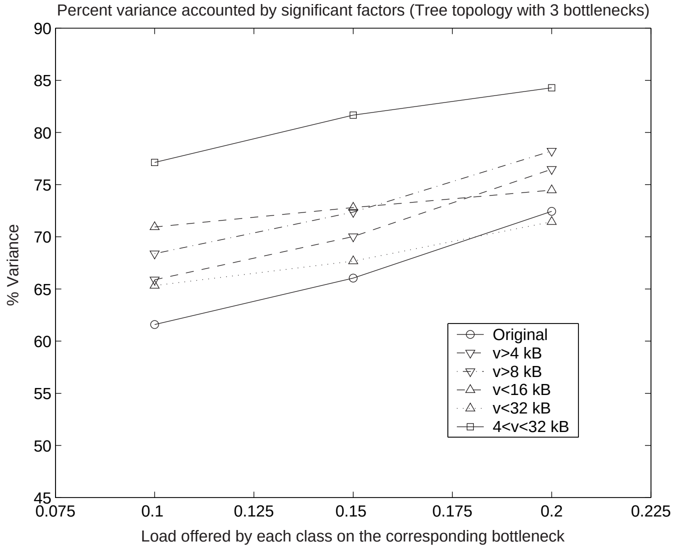

Figure 4.6: Percent normalized variance under different loads offered by each of the classes 1–7 for the three-bottleneck scenario. The total utilization factors of each of the bottlenecks are the same in each offered load case, and are 70%, 80%, and 90%, respectively.

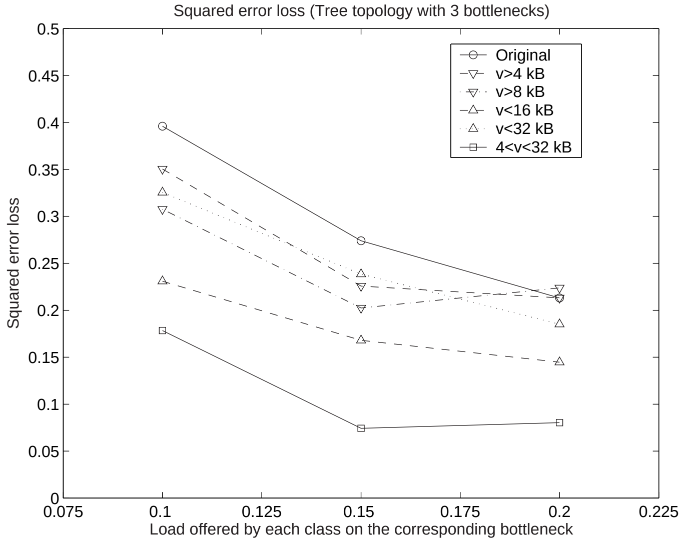

Figure 4.7: Squared error loss under different loads offered by each of the classes 1–7 for the three-bottleneck scenario. The total utilization factors of each of the bottlenecks are the same in each offered load case, and are 70%, 80%, and 90%, respectively.

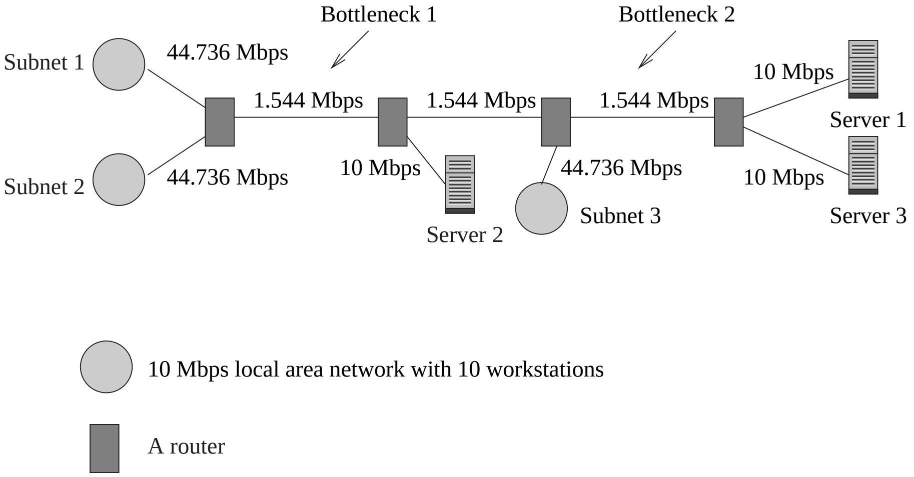

Figure 4.8: Linear network topology with two coupled bottleneck links used in OPNET TCP simulations.

to subnet 1 (class 1) download files from server 1, users belonging to subnet 2 (class 2) download files from server 2, and users belonging to subnet 3 (class 3) download files from server 3. Class 1 offers a load of 20% on the bottleneck link 1 and 2. Class 2 offers a load of 40% on the bottleneck link 1. Class 3 offers a load of 40% on the bottleneck link 2. The load due to background traffic on the bottlenecks 1 and 2 is set to 20%.

After filtering out flows whose sizes are smaller than 4 kB or greater than 32 kB, I find that there are two significant factors. Then, I estimate factor loadings of four class throughputs based on two significant factors. I first estimate factor loadings and specific factors based on (2.3), and then use varimax rotation [31] on $\hat{\mathbf{\Lambda}}$:

$$\hat{\mathbf{\Lambda}}^* = \begin{pmatrix} 0.5011 & 0.7272 \\ 0.9570 & -0.0041 \\ -0.1143 & 0.9214 \end{pmatrix},$$

and

$$\hat{\mathbf{\Psi}} = \text{diag}\,(0.2200, 0.0841, 0.1380)\,.$$

The explanatory power of the two factors is 85%. From the results, one can see that the throughputs of classes 2 and 3 are captured by only one factor, i.e. the bottleneck link that each traverses. For class 1, one can argue that both loadings are significant, and hence the throughputs of class 1 can be explained by two factors, i.e. the two bottlenecks the flows belonging to class 1 visit. This example scenario shows the effectiveness of factor analysis in identifying *multiple* bottlenecks in a linear network.

4.5 Wireless Local Area Networks

Determination of the cause of poor performance in wireless local area networks (WLANs) by using *only* flow level measurements is a challenging problem. A set of wireless stations belonging to a basic service set (BSS) may experience poor network service quality due to a number of reasons. Here, the number of wireless users, frequency and mean size of file download requests will be termed as the *traffic patterns* of wireless users. In wireless networks, the *spatial location* of a user is also very important in determining the perceived throughput performance. Often, the users experience poor performance not because of the lower capacity of the wireless network compared to high-bandwidth wired technologies, but either because of an underprovisioned backhaul link to the infrastructure network or because of bad coverage, or misplacement, of the access point (AP) for a given spatial distribution of users in the WLAN.

I apply the proposed factor analysis framework to determine the cause of poor performance for users of FTP traffic in an IEEE 802.11b WLAN [48–50], which is currently the most widely deployed WLAN technology. IEEE 802.11b WLANs support four different data rates, 11 Mbps, 5.5 Mbps, 2 Mbps, and 1 Mbps, to compensate for the loss of signal strength when decoding data packets. The wireless stations constantly detect signal strength and select the best data rate at which they can operate accordingly. The signal strength between an AP and a wireless station depends on the distance between them and is significantly affected by obstructions along the signal's propagation path. IEEE 802.11b WLANs use carrier sense multiple access with collision avoidance (CSMA/CA) mechanism to mediate access to the wireless channel.

This section considers two cases to demonstrate the power and applicability of flow level measurements and factor analysis in WLAN network management. In the first case, a link in the provider's network will be underprovisioned for the traffic patterns of wireless users, and is the source of congestion as shown in Fig. 4.9. I will suppose that the wireless network managers do not have access to the utilization levels of the bottleneck link in the provider's network. In the second case, all links in the provider's network will be overprovisioned. However, the AP will not be optimally located with respect to wireless users as shown in Fig. 4.10. In this case, the AP will be the bottleneck operating at a low data rate due to weak signal strength to the users. In both cases, users generate equal load on the bottlenecked resource (underprovisioned link or congested AP), and hence wireless users will perceive the same (poor) average flow throughput performance.

4.5.1 Simulation setup

In the two cases considered in Figs. 4.9 and 4.10, the users download files from a server using FTP. In each case, there are twenty wireless users in the BSS. The users generate a load of 85% on the bottlenecked resource. I will consider flows destined to wireless stations 1–4. In the first case, the access link has a capacity of 1 Mbps. The average throughput received by each of these stations is approximately 60 kbps. In the second case, the AP operates at a data rate of 1 Mbps, which is the rate selected by the transceivers of wireless stations due to weak signal strength. The average throughput received by each of these stations is again approximately 60 kbps. Hence, in both cases, users experience the same poor flow throughput performance.

Each simulation corresponds to 2 hours of file download activity. During the simulations, I record the request time, size (in bytes) and duration (in seconds) of each file transfer. File transfer requests arrive according to a Poisson distribution, and file sizes are selected from a lognormal distribution with a mean file size of 16 kB and a standard deviation of 131 kB as before. The conditional correlation matrix is estimated for the throughputs of four

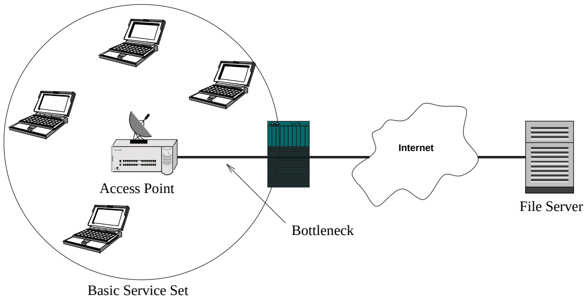

Figure 4.9: A basic service set with twenty wireless users, only four users of interest are shown. All stations can support data rates at 11 Mbps. The link capacity is underprovisioned for the traffic patterns generated by wireless users. Users are perceiving poor quality of service (throughput) due to congestion at the bottleneck link.

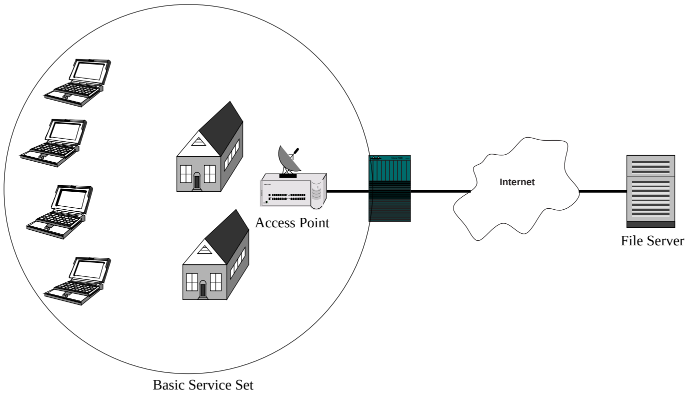

Figure 4.10: A basic service set with twenty wireless users, only four users of interest are shown. The access point is not placed optimally with respect to the spatial distribution of wireless users. Users are perceiving poor quality of service (throughput) due to weak signal strengths at their positions.

wireless stations (classes 1–4), and factor analysis is performed on the matrix.

4.5.2 Simulation results

In the first case, when the poor quality of service is due to a bottleneck link in the infrastructure network, the eigenvalues of the conditional correlation matrix of flow class throughputs are

$$\{3.0254, 0.6139, 0.2066, 0.1541\}.$$

Based on the modified Kaiser's rule in Section 2.3.3, there is a single significant factor that accounts for most of the variability in throughputs. The explanatory power of this single factor is 76%. It can be concluded that the variability in flow class throughputs is mainly due to the way in which congested link bandwidth is allocated to active flows.

In the second case, when the poor quality of service is due to a poorly located AP, the eigenvalues of the conditional correlation matrix of flow class throughputs are

$$\{1.2571, 0.9530, 0.9416, 0.8484\}.$$

Once again, based on the modified Kaiser's rule, there are three significant factors. The explanatory power of the three factors is 79%. Existence of more than one common factor can be attributed to the congested AP that divides its capacity among active users. As such, the variability of flow class throughput of each user is mainly due to the way bandwidth is allocated to its active flows at its own congested "path". Therefore, existence of more than one common factor indicates that the poor performance is due to low data rates at which wireless stations operate because of their poor locations with respect to the AP.

4.5.3 Discussion: Traffic patterns versus spatial access patterns in wireless networks

One can employ the methodology described in this dissertation to investigate the causes of poor performance perceived by wireless users when downloading documents from a server. For instance, given that the flow source addresses are available from records, upon collecting flow level measurements at an AP of a BSS where poor throughputs are being reported, one can determine that the AP is not placed properly if multiple factors are determined to be the source of variation in flow class throughputs. In this case, the owners of the AP may relocate it to a better location. If there is one factor underlying the variations in flow class throughputs, the network managers may then request an increase of the provisioned capacity of the access links or deploy additional APs to support the traffic patterns of wireless users.

4.6 Conclusion

This chapter presented an evaluation of using factor analysis to infer which flow classes share congested resources through extensive TCP simulations. The results showed that the methodology can be a very effective tool in inferring path sharing by TCP flows and diagnosing problems that arise from bottleneck sharing. The benefit of filtering out small and large flows was investigated for flow records generated by simulation. Flow filtering thresholds were determined empirically. The methodology was also validated for wireless local area networks through simulations. A potential application involves assessing wireless network performance for given traffic and spatial access patterns based only on flow level measurements collected at an AP.

The next chapter will apply the inference methodology to real TCP measurements. The results are expected to provide a realistic assessment of the potential use of this methodology in performance analysis of real networks.

Chapter 5

Case Studies Using Real TCP Flow Measurements

5.1 Introduction

In this chapter, I apply factor analysis to actual TCP flow class throughput correlation matrices. Flow class throughputs are obtained from TCP flow records that are collected by networking equipment. A validation of the resource sharing results with real flow measurements is extremely hard, if not impossible, since routing information about all the domains that flows visit and the congestion status of the servers that provide the incoming traffic are not available. However, bootstrap confidence intervals can be used to demonstrate the statistical accuracy of the inference methodology.

Section 5.2 describes the datasets that are used in this chapter. Section 5.3 applies the inference methodology to the real datasets. Section 5.4 concludes the chapter.

5.2 Description of Datasets

I use NetFlow [9] records collected at the border router of The University of Texas at Austin (UT Austin) on November 6, 2002, between 12:58 PM

Table 5.1: Description of NetFlow datasets collected at UT Austin's border router.

	Date	Period	TCP records
Dataset2002	11/6/2002	12:58 PM - 2:07 PM	5,173,385
Dataset2004	1/21/2004	12:58 PM - 1:26 PM	4,440,697

and 2:07 PM CST, and on January 21, 2004, between 12:58 PM and 1:26 PM CST. The records that are collected in 2002 are referred to as Dataset2002, and those that are collected in 2004 are referred to as DataSet2004. Dataset2002 consists of 5,173,385 TCP flow records out of a total of 5,866,602 flow records. Dataset2004 consists of 4,440,697 TCP flow records out of a total of 6,556,674 flow records. The records contain both the incoming and outgoing traffic from UT Austin. The IP addresses belonging to UT Austin were made anonymous to protect privacy. Table 5.1 summarizes these datasets. Figs. 5.1–5.6 provide some descriptive statistics for the TCP flows in Dataset2002 and Dataset2004. The pie charts in Figs. 5.1 and 5.2 show the percent distribution of flow sizes in packets. Percent distribution of flow lengths in seconds are shown in Figs. 5.3 and 5.4. The cumulative distribution functions in Figs. 5.5 and 5.6 provide some insight into the flow size distributions in the datasets.

I assume that over a one-hour period, flow class throughputs can be modelled as stationary processes. Furthermore, I assume that the packets from a given TCP flow follow the same route[1]. Such assumptions, although

---

[1] This assumption is supported by the empirical measurements in [51].

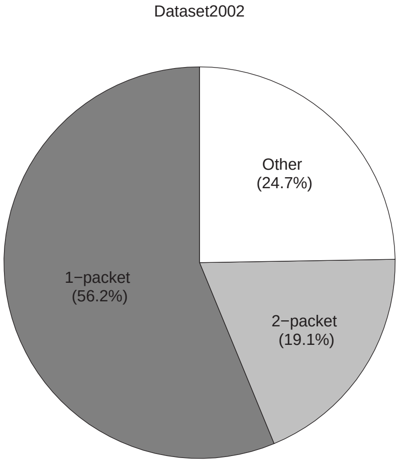

Figure 5.1: Percent distribution of flow sizes in packets for Dataset 2002.

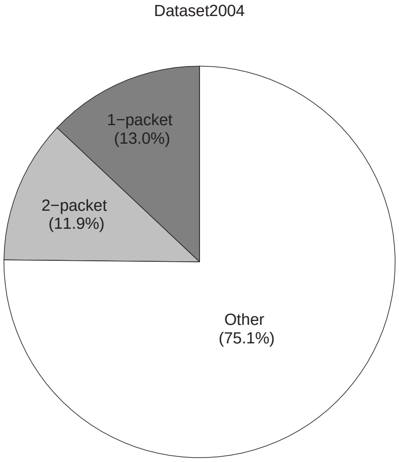

Figure 5.2: Percent distribution of flow sizes in packets for Dataset 2004.

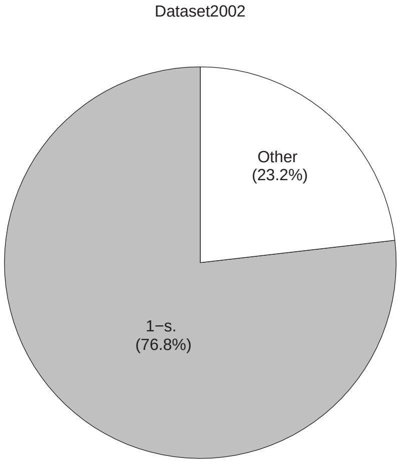

Figure 5.3: Percent distribution of flow lengths in seconds for Dataset 2002.

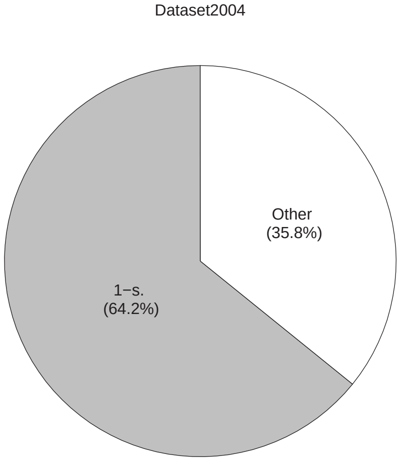

Figure 5.4: Percent distribution of flow lengths in seconds for Dataset 2004.

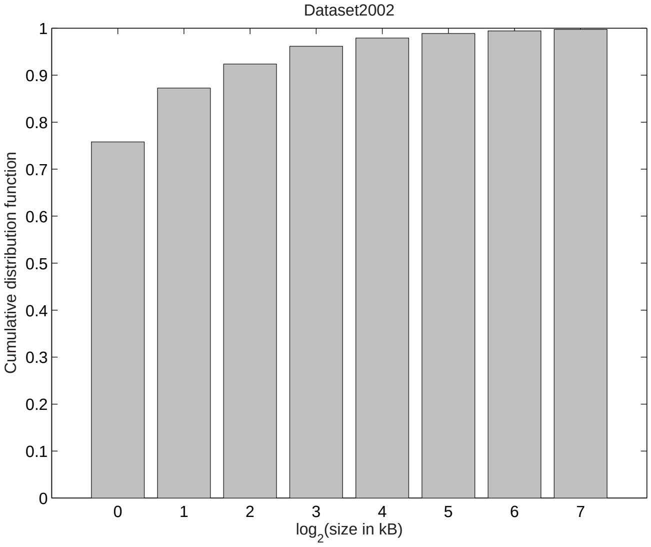

Figure 5.5: Cumulative distribution function of flow sizes in kB for Dataset 2002.

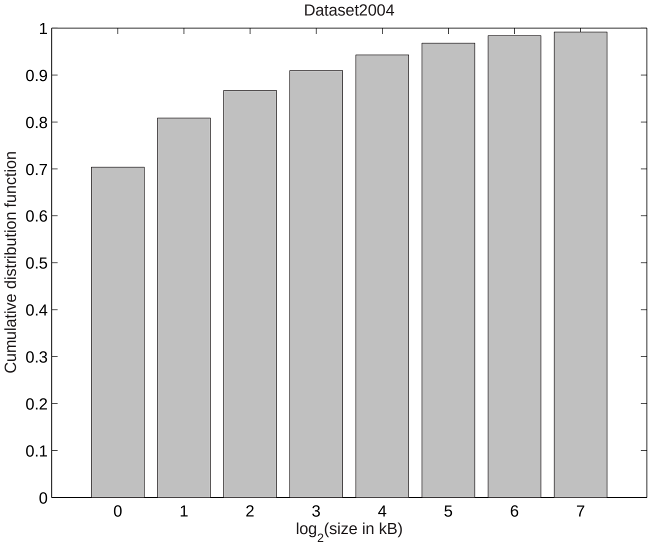

Figure 5.6: Cumulative distribution function of flow sizes in kB for Dataset 2004.

idealized, are not completely unrealistic for our one-hour long flow measurements.

5.3 Methodology

In NetFlow records, the start time of a flow is the time of arrival of the first packet in the flow, and the end time is the time of arrival of the last packet in the flow. Since the time between the first and the last packet is zero, flow throughput is not defined for flows consisting of one packet. Hence, one-packet flows will be omitted. Based on the claims that are validated by performed simulations in Chapters 3 and 4, I filter out all flow records whose sizes are smaller than one threshold or larger than another threshold in order to better capture the throughput correlations among flow classes. Based on extensive empirical investigations, I find that flows whose sizes are between 8 kB and 64 kB can represent the dynamics in the network well for the datasets at hand. In addition, in the Internet, packets belonging to flows that consist of only a few packets can sometimes arrive back to back (or with a very small inter-packet spacing). In this case, it is unreasonable to assume that such large flow throughputs are typical for that flow class. Hence, I will also omit all flows whose durations are shorter than one second.

I choose to analyze incoming traffic associated with AOL and HotMail, since one can reasonably assume that traffic belonging to these CPs potentially experience congestion at their source due to high demand for their content. I define two flow classes for traffic from each provider: AOL1 and AOL2 (class

Table 5.2: Mean of bootstrap replications and 95% confidence intervals for eigenvalues of **R** based on Dataset2002.

Eigenvalue	Mean	Interval estimate
e_1	1.7274	(1.5457, 1.7900)
e_2	1.1562	(1.0861, 1.3206)
e_3	0.8344	(0.7058, 0.9150)
e_4	0.2785	(0.2194, 0.4458)

1 and class 2) from AOL, and HotMail1 and HotMail2 (class 3 and class 4) from Microsoft Corporation. Assignment of flows into AOL1 or AOL2 (and similarly for HotMail1 and HotMail2) is performed by randomly splitting all flows from AOL (and HotMail) into two sets. That way, I can hypothesize which classes share infrastructure on their path to UT Austin with reasonable certainty.

5.3.1 Validation of methodology

First, I describe how to choose the number of significant factors. I estimate 95% BC_a confidence intervals for four eigenvalues of the class throughput correlation matrix **R**. The results are shown in Table 5.2 for Dataset2002, and in Table 5.3 for Dataset2004.

Once the confidence intervals for the eigenvalues are estimated, the modified Kaiser's rule for real data can be used to choose the significant eigenvalues: the eigenvalues whose confidence intervals lie below 1 are designated as *insignificant*. Therefore, from Tables 5.2 and 5.3, there are two significant

Table 5.3: Mean of bootstrap replications and 95% confidence intervals for eigenvalues of $\mathbf{R}$ based on Dataset2004.

Eigenvalue	Mean	Interval estimate
e_1	1.4287	(1.3646, 1.4786)
e_2	1.0780	(1.0237, 1.1603)
e_3	0.9094	(0.8230, 0.9690)
e_4	0.5856	(0.5413, 0.6379)

factors; i.e., four classes share two different network infrastructures. The explanatory power of the two factors is 72% in the case of Dataset2002 and 63% in the case of Dataset2004.

After establishing the number of significant factors, I estimate factor loadings of four class throughputs based on two significant factors. I first estimate factor loadings and specific factors based on (2.3), and then use varimax rotation on $\hat{\mathbf{\Lambda}}$. For DataSet2002,

$$\hat{\mathbf{\Lambda}}^* = \begin{pmatrix} 0.7933 & 0.0711 \\ 0.7289 & -0.1315 \\ -0.0842 & 0.9088 \\ 0.0501 & 0.9240 \end{pmatrix}$$

and

$$\hat{\mathbf{\Psi}} = \mathrm{diag}\left(0.3656, 0.4514, 0.1669, 0.1437\right).$$

For Dataset2004,

$$\hat{\mathbf{\Lambda}}^* = \begin{pmatrix} 0.8378 & -0.0451 \\ 0.8411 & 0.0044 \\ 0.0200 & -0.7415 \\ 0.0260 & -0.7351 \end{pmatrix}$$

and

$$\hat{\mathbf{\Psi}} = \mathrm{diag}\left(0.2961, 0.2926, 0.4497, 0.4589\right).$$

Table 5.4: Mean of bootstrap replications and 95% confidence intervals for factor loadings based on Dataset2002.

Loading	Mean	Interval estimate
$\lvert\Lambda^*_{11}\rvert$	0.7944	(0.7567, 0.8252)
$\lvert\Lambda^*_{12}\rvert$	0.0761	(0.0036, 0.1688)
$\lvert\Lambda^*_{21}\rvert$	0.7250	(0.6360, 0.7884)
$\lvert\Lambda^*_{22}\rvert$	0.1331	(0.0235, 0.2401)
$\lvert\Lambda^*_{31}\rvert$	0.0836	(0.0241, 0.1436)
$\lvert\Lambda^*_{32}\rvert$	0.9110	(0.8564, 0.9362)
$\lvert\Lambda^*_{41}\rvert$	0.0535	(0.0042, 0.1294)
$\lvert\Lambda^*_{42}\rvert$	0.9250	(0.8806, 0.9483)

Next, 95% BC_a confidence intervals for absolute values of eight rotated factor loadings are computed. When computing confidence intervals for factor loadings, one needs to take into account sign reversals of loadings and changes in the order of factors across bootstrap samples. As such, I compute the confidence intervals of the absolute values of loadings. I rearrange the order of factors if such reordering results in a smaller $\|\hat{\mathbf{\Lambda}}^* - \hat{\mathbf{\Lambda}}^*(b)\|$, where $\hat{\mathbf{\Lambda}}^*$ is estimated using (2.3) and varimax rotation, and $\hat{\mathbf{\Lambda}}^*(b)$ is the estimate for $\hat{\mathbf{\Lambda}}^*$ using the bth bootstrap replication. The results are given in Table 5.4 for DataSet2002 and in Table 5.5 for DataSet2004.

By inspecting the significant loadings on the loading matrix, we can conclude that classes 1 and 2 (flows belonging to AOL) share factor 1, and classes 3 and 4 (flows belonging to HotMail) share factor 2 with 95% confidence. In this case, factor 1 would be interpreted as the networking infrastructure belonging to AOL, and factor 2 would be the networking infrastructure

Table 5.5: Mean of bootstrap replications and 95% confidence intervals for factor loadings based on Dataset2004.

Loading	Mean	Interval estimate
$\lvert\Lambda^*_{11}\rvert$	0.8370	(0.8223, 0.8544)
$\lvert\Lambda^*_{12}\rvert$	0.0478	(0.0034, 0.1312)
$\lvert\Lambda^*_{21}\rvert$	0.8402	(0.8254, 0.8580)
$\lvert\Lambda^*_{22}\rvert$	0.0303	(0.0000, 0.1296)
$\lvert\Lambda^*_{31}\rvert$	0.0458	(0.0000, 0.0731)
$\lvert\Lambda^*_{32}\rvert$	0.7395	(0.6314, 0.7879)
$\lvert\Lambda^*_{41}\rvert$	0.0512	(0.0004, 0.0969)
$\lvert\Lambda^*_{42}\rvert$	0.7316	(0.6207, 0.7737)

belonging to Microsoft Corporation.

5.3.2 Discussion of results

The potential power of this inference technique in root cause analysis may be illustrated by considering the results in Tables 5.4 and 5.5. For example, suppose that the users belonging to classes AOL1 and AOL2 at UT Austin were experiencing poor performance (long download times), and UT Austin's network managers were capable of verifying that utilization of the local network was low. Treating the external network as a "black box" (i.e., no knowledge about the utilization factors of access links or routing information of outside network), network managers could infer that poor performance was not due to the access links connecting UT Austin to the Internet, because the flow classes did not have one common factor that would indicate a bottleneck shared by all classes. The network managers could then hypothesize that the cause for poor performance was either at the CP's server or a bottleneck link visited by both flow classes in the Internet.

5.4 Conclusion

This chapter analyzed real TCP flow records collected at the border router of UT Austin by using factor analysis. The algorithm developed to infer resource sharing from flow records is summarized in Table 5.6. The methods identify TCP flow classes that share network infrastructure with 95% confidence. The applicability of the methodology to real data has a potential impact on designing network monitoring tools for investigating root cause of poor network performance. The application of factor analysis to real network measurements to infer network properties is, to the best of my knowledge, a novel idea.

Although the results presented in this chapter are quite encouraging, future research is required to further investigate the validity of the methodology in the constantly changing and evolving Internet. Some suggestions for possible future research directions are included in Chapter 6.

Table 5.6: Algorithm to infer resource sharing from flow records.

```
1: Define the flow classes of interest, $\mathcal{C}$
2: Set the flow filtering thresholds:
       $T_{\text{packets}}$, $T_{\text{duration}}$, $T_{\text{bytes}}^{\text{lower}}$, and $T_{\text{bytes}}^{\text{upper}}$
3: Determine the flows $\mathcal{F}$ that belong to $\mathcal{C}$ and satisfy
       $\textit{number of packets in flow} > T_{\text{packets}}$
       $\textit{flow duration} > T_{\text{duration}}$
       $T_{\text{bytes}}^{\text{lower}} < \textit{number of bytes in flow} < T_{\text{bytes}}^{\text{upper}}$
4: Compute flow class throughputs using (3.3)
5: Discretize time and estimate pairwise correlations using (3.8)
6: Find the number of factors $m$ using the eigenvalues of
   the correlation matrix in (3.9) and the modified Kaiser's rule
7: Perform factor analysis based on $m$ factors
8: Rotate factor loadings using varimax rotation
9: Determine which flow classes have the largest loading on
   a given factor: These classes are likely to share a congested resource
```

Chapter 6

Conclusion

6.1 Summary

This dissertation introduces a new flow level approach to network tomography that infers which flow classes share congested resources. I develop a methodology to analyze correlations of flow class throughputs from temporal observations. This new methodology for flow level network tomography is based on my thesis statement:

> *The correlation structure of throughputs obtained by flow level measurements for a number of TCP flow classes can often be captured by a fewer number of latent factors that can be used to infer which flow classes share resources in the network.*

The research presented in this dissertation validates this statement through the use of exploratory factor analysis on conditionally sampled flow class throughputs. Filtering out flows based on their sizes has also been shown to be a necessary step in preprocessing of flow records. Fig. 6.1 summarizes the main steps of the inference methodology described. The figure also illustrates how the bootstrap can be used in conjunction with factor analysis to make inferential statements about resource sharing.

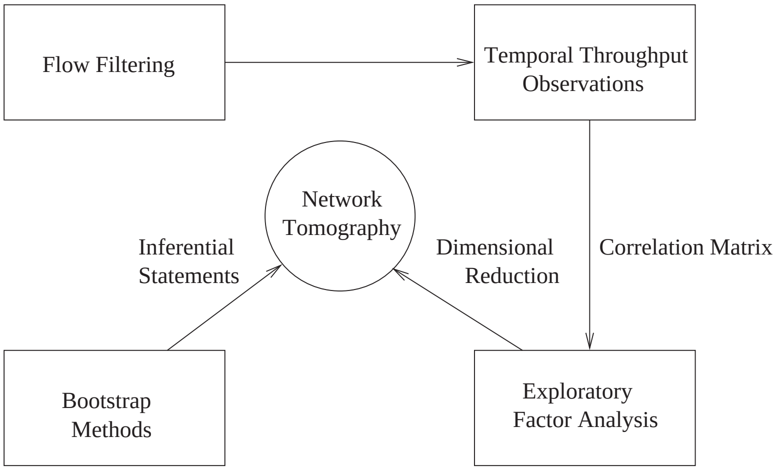

Figure 6.1: Main steps of the introduced methodology for inferring resource sharing.

A brief summary of the dissertation follows. Chapter 1 presented the main motivations for ISPs and CPs for inferring resource sharing. Related work was summarized and contrasted with the methods introduced in this dissertation.

Chapter 2 provided a review of factor analysis and the bootstrap. Both methods were the core techniques employed in this work.

Chapter 3 formalized the existence of correlation between throughputs of flows that share a congested resource. A conditional sampling technique was described to capture flow class throughput correlations. Factor analysis was used on the correlation matrix to explore which flow classes might share resources in the network. Flow filtering criteria were established to better capture resource sharing in the network.

Chapter 4 validated the developed methods by using extensive TCP simulations. A number of traffic conditions and network topologies were used to evaluate the effectiveness of factor analysis in identifying flows that shared resources. The applicability of the methodology in wireless local area networks was also considered. Factor analysis successfully determined whether the cause of poor performance in a wireless local area network was due to traffic patterns of wireless users or due to users' spatial locations.

Chapter 5 coupled exploratory factor analysis with bootstrap methods to make inferential statements about resource sharing based on the data at hand. The methodology was validated using real TCP data. The applicability of the methodology to real data may potentially impact the design of future network monitoring tools.

I summarize the primary contributions of this dissertation:

- **Methodology.** I believe that the use of factor analysis in analyzing network properties is a novel idea [25–27]. A distinctive feature of my work is the consideration of the correlation structure of conditionally sampled random processes (flow class throughputs) whose samples are taken when the processes are active at the sampling instant. A very recent work described in [52] also considers applying structural analysis techniques, principal component analysis in particular, to network traffic measurements for examining the intrinsic dimensionality of origin-destination flow time series in a network.

- **Empirical findings.** Due to the constantly changing nature of user behavior and the emergence of new applications, a thorough study of distribution of flow sizes in the network and its impact on throughput correlations is almost impossible. Based on an extensive set of empirical studies, I present several flow filtering criteria based on flow sizes that improve inference results for resource sharing. The methods are validated with two distinct real datasets from UT Austin's border router.

- **Applications.** The methodologies described in this dissertation for inferring path sharing based on flow records can serve as a tool for network monitoring and root cause analysis of poor performance.

6.2 Future Work

In this dissertation, I employed a series of simulations and two real datasets to establish the validity of the presented methodology. However, future work should further investigate the validity of the proposed methods in the constantly changing and evolving Internet, perhaps through analysis of more extensive datasets.

Future research on inferring network properties using flow level measurements could focus on a variety of topics that are oriented towards addressing the open issues in this work. Examples include a further investigation of filtering thresholds, the feasibility of an "overlap" based flow filtering strategy, the use of active probe flows to infer network properties, and the effect of non-stationary network traffic on inferences.

Filtering out flows based on their sizes in bytes or packets, and on their duration requires further research. In this work, I established flow filtering criteria empirically. I believe that the nature of flow sizes and durations in the Internet [53] is too complex and too dynamic, and determination of filtering thresholds analytically is extremely difficult, if not impossible. However, one could perform more experiments to devise “good” engineering rules for selecting flow size or duration thresholds using datasets from several major ISPs.

One can also rethink the flow filtering strategy. I argued that filtering out flows based on their sizes was an effective strategy to capture throughput correlations. As demonstrated in Chapter 3, the amount of overlap relative to the duration of the longer flow is the key factor in determining the correlation between flow throughputs. Although the implementation of such a strategy may not be as straightforward as the size or duration based filtering, it may be worthwhile to analyze the benefit of resorting to “overlap” based filtering of flows.

A very interesting research problem involves investigating the use of active *probe flows* to “learn” network properties. The techniques employed in this dissertation are based on passively collected flow records. Extending the methodology to an active measurement framework could be useful for end systems that wish to perform on demand performance analysis of the network. For example, in an active framework, the service provider could probe two or more users by sending a number of temporally overlapping flows to capture the congestion state of the network routes between the provider and users. Establishing bounds on the number and size of probe flows that are required to achieve accurate inferences without generating prohibitive probe traffic could potentially be a significant contribution to network tomography research. Recall that filtering out small and large flows better captures throughput correlations, but reduces the number of flows that are available for statistical analysis. With an active probing strategy, flows with "proper" sizes can be used to address the dilemma between discarding small and large flows and retaining enough samples for statistical analysis. Note that active measurements can always be employed in conjunction with passive measurements for more effective network monitoring.

In this work, I addressed the impact of non-stationarity of network traffic on inferences for one particular topology. Network traffic is stationary only over a few hours. One could investigate the effect of non-stationarity on correlation estimates, and hence, on factor analysis for datasets that contain measurements for longer periods of time. One approach to dealing with non-stationarity may involve dividing the measurement period into segments [30, 54] over which the network traffic is stationary, and estimating correlations among flow class throughputs over these segments separately.

In addition to further research that involves analysis, one could explore new application areas for the introduced methodology. For example, proper configuration and placement of access points in wireless local area networks poses a challenging problem, and is a topic of active research. Further evaluation of the use of factor analysis should be performed to investigate how wireless users can be classified according to their traffic patterns and spatial distributions based on flow level measurements.

In addition to the wireless local area networks, cellular networks are increasingly carrying "document" traffic [55]. However, the major challenge in cellular networks is the high mobility of users: Users might receive poor performance due to their location for a brief period of time, and later, start receiving acceptable performance as they move to a different location. In this case, the throughput perceived by the users is mainly driven by their mobility rather than the congestion control mechanisms. Conducting research in applications of factor analysis in highly mobile cellular networks to analyze flow level performance characteristics of users could be rewarding for cellular service providers.

There is also an emerging interest in analyzing the performance of sensor networks [56]. Sensor networks consist of a very large number of low-power, low-cost devices, or sensors, that are interconnected with low-capacity links. The sensors that are dispersed in an area collect various measurements of their environment. The measurements are then propagated to a data processing center. Propagating voluminous amounts of measurements towards a data processing center ("measurement implosion") is very likely to create hot spots in the sensor network. By using factor analysis, one could identify which sensor flows share hot spots. An investigation of the application of the introduced methods may provide useful guidelines in design and implementation of sensor networks.

Finally, current network service billing mechanisms usually charge users based on the amount of data they send and receive, the duration of connection to the network, or the size of the access ports, e.g. OC-3 (155.250 Mbps), OC-12 (622.080 Mbps), etc. A usage based charging scheme based on flows is described in [57]. I believe that throughput must also be a key factor in determining the price that a user has to pay to the service provider. If the users are experiencing poor quality of service (low throughput), and the cause of poor performance is determined to be due to a bottleneck in the provider's network, users could get a discount in their bill to compensate for their dissatisfaction with the service. The methods presented in this dissertation may provide the first step in determining the root causes of poor performance of TCP flows, and should be easily integrable to user pricing mechanisms.

Bibliography

[1] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, Mar. 1996.

[2] M. Allman and V. Paxson, "On estimating end-to-end network path properties," in *Proc. ACM Conf. on Appl., Tech., Arch., and Protocols for Computer Communications*, Aug. 1999, pp. 263–274.

[3] R. Cáceres, N. G. Duffield, J. Horowitz, and D. F. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. on Info. Theory*, vol. 45, no. 7, pp. 2462–2480, Nov. 1999.

[4] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *Proc. IEEE Conf. on Computer Communications*, vol. 1, 1999, pp. 353–360.

[5] J. Liu and M. Crovella, "Active measurements: Using loss pairs to discover network properties," in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2001, pp. 127–138.

[6] F. L. Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE/ACM Trans. on Networking*, vol. 10, no. 6, pp. 761–775, Dec. 2002.

[7] Y. Tsang, M. Coates, and R. D. Nowak, "Network delay tomography," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2125–2136, Aug. 2003.

[8] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, 3rd ed. Addison-Wesley, 1998.

[9] *NetFlow.* http://www.cisco.com: Cisco Systems, Inc.

[10] *Internet Measurement Tool Taxonomy.* http://www.caida.org/tools: The Cooperative Association for Internet Data Analysis (CAIDA).

[11] D. E. Comer, *Internetworking with TCP/IP*, 3rd ed. Prentice Hall, 1995, vol. 1.

[12] S. Savage, N. Cardwell, and T. Anderson, "The case for informed transport protocols," in *Proc. IEEE Workshop on Hot Topics in Operating Systems*, Mar. 1999, pp. 58–63.

[13] K. Harfoush, A. Bestavros, and J. Byers, "Robust identification of shared losses using end-to-end unicast probes," in *Proc. IEEE Int. Conf. on Network Protocols*, Nov. 2000, pp. 22–36.

[14] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Trans. on Networking*, vol. 10, no. 3, pp. 381–395, June 2002.

[15] M. Rabbat, R. Nowak, and M. Coates, "Network tomography and the identification of shared infrastructure," in *Proc. IEEE Asilomar Conf. on Signals, Systems and Computers*, Nov. 2002, pp. 34–38.

[16] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Proc. IEEE Int. Conf. on Computer Communications and Networks*, 2001, pp. 174–181.

[17] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger, "Dynamics of IP traffic: a study of the role of variability and the impact of control," in *Proc. ACM Conf. on Appl., Tech., Arch., and Protocols for Computer Communications*, Aug. 1999, pp. 301–313.

[18] N. Brownlee, C. Mills, and G. Ruth, "Traffic flow measurement: Architecture," *IETF Request for Comments 2722*, Oct. 1999.

[19] *sFlow*. http://www.sflow.org: sFlow.org.

[20] *Argus*. http://www.qosient.com/argus: QoSient, llc.

[21] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2001, pp. 75–80.

[22] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2002, pp. 159–171.

[23] S. Shenker, “Fundamental design issues for the future Internet,” *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sept. 1995.

[24] *IP Monitoring Project.* http://ipmon.sprintlabs.com: Sprint Corp.

[25] D. Arifler, G. de Veciana, and B. L. Evans, “Network tomography based on flow level measurements,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Montreal, Canada, May 2004, to appear.

[26] ——, “Inferring path sharing based on flow level TCP measurements,” in *Proc. IEEE Int. Conf. on Communications*, Paris, France, June 2004, to appear.

[27] ——, “Network tomography based on flow level measurements,” *IEEE/ACM Trans. on Networking*, submitted Feb. 2004.

[28] C. Spearman, “General intelligence, objectively determined and measured,” *American Journal of Psychology*, vol. 15, pp. 201–293, 1904.

[29] D. N. Lawley and A. E. Maxwell, *Factor Analysis as a Statistical Method*, 2nd ed. American Elsevier, New York, 1971.

[30] A. Basilevsky, *Statistical Factor Analysis and Related Methods: Theory and Application.* Wiley, 1994.

[31] A. C. Rencher, *Multivariate Statistical Inference and Applications.* Wiley, 1998.

[32] H. F. Kaiser, "The application of electronic computers to factor analysis," *Educational and Psychological Measurement*, vol. 20, pp. 141–151, 1960.

[33] D. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap.* Chapman & Hall, Inc., 1993.

[34] L. Massoulié and J. W. Roberts, "Bandwidth sharing and admission control for elastic traffic," *Telecommunication Systems*, vol. 15, pp. 185–201, June 2000.

[35] S. B. Fred, T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts, "Statistical bandwidth sharing: a study of congestion at flow level," in *Proc. ACM Conf. on Appl., Tech., Arch., and Protocols for Computer Communications*, Aug. 2001, pp. 111–122.

[36] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. on Networking*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[37] A. B. Downey, "The structural causes of file size distributions," in *Proc. IEEE Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Aug. 2001, pp. 361–370.

[38] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," in *Proc. Allerton Conf. on Communications, Control, and Computing*, Oct. 2001, pp. 182–191.

[39] S. J. Devlin, R. Gnanadesikan, and J. R. Kettenring, “Robust estimation of dispersion matrices and principal components,” *Journal of the American Statistical Association*, vol. 76, no. 374, pp. 354–362, June 1981.

[40] ——, “Robust estimation and outlier detection with correlation coefficients,” *Biometrika*, vol. 62, no. 3, pp. 531–545, 1975.

[41] L. Kleinrock, *Queueing Systems: Computer Applications.* Wiley-Interscience, 1976, vol. 2.

[42] F. P. Kelly, “Charging and rate control for elastic traffic,” *European Trans. on Telecommunications*, vol. 8, pp. 33–37, 1997.

[43] F. P. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: Shadow prices, proportional fairness and stability,” *Journal of the Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[44] D. Bertsekas and R. Gallager, *Data Networks*, 1992.

[45] L. Massoulié and J. W. Roberts, “Bandwidth sharing: objectives and algorithms,” *IEEE/ACM Trans. on Networking*, vol. 10, no. 3, pp. 320–328, June 2002.

[46] G. de Veciana, T.-J. Lee, and T. P. Konstantopoulos, “Stability and performance analysis of networks supporting elastic services,” *IEEE/ACM Trans. on Networking*, vol. 9, no. 1, pp. 2–14, Feb. 2001.

[47] *OPNET Modeler 9.0.* http://www.opnet.com: OPNET Technologies, Inc.

[48] *IEEE Std 802.11-1997*, IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Nov. 1997.

[49] *IEEE Std 802.11b-1999, 2000*, Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 2000.

[50] *IEEE Std 802.11b-1999/Cor 1-2001*, IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 2: Higher-Speed Physical Layer (PHY) Extension in the 2.4 GHz Band - Corrigendum 1, Nov. 2001.

[51] V. Paxson, “End-to-end routing behavior in the Internet,” *IEEE/ACM Trans. on Networking*, vol. 5, no. 5, pp. 601–615, Oct. 1997.

[52] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems*, New York, USA, June 2004, to appear.

[53] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of Internet flow rates," in *Proc. ACM Conf. on Appl., Tech., Arch., and Protocols for Computer Communications*, vol. 31, no. 4, 2002, pp. 309–322.

[54] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed. McGraw-Hill, 2000.

[55] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and S. Viterbi, "CDMA/HDR: A bandwidth efficient high speed data service for nomadic users," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 70–77, July 2000.

[56] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[57] N. Duffield, C. Lund, and M. Thorup, "Charging from sampled network usage," in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2001, pp. 245–256.

Vita

Dogu Arifler, the son of Gunay and Soner Arifler, was born in Nicosia, Cyprus, on March 9, 1975. In 1993, he was awarded a full scholarship through the Cyprus-America Scholarship Program to study Electrical and Computer Engineering at The University of Texas at Austin. He received his B.S.E.E. (Highest Honors) in May 1997. In August 1997, he entered the Graduate School at The University of Texas at Austin where he received his M.S. degree in Electrical Engineering in May 1999. He worked as a full-time software engineer at National Instruments in Austin, Texas, from July 1999 until December 2000. In January 2001, he went back to the Graduate School at The University of Texas at Austin to pursue his Ph.D.E.E. degree under the supervision of Prof. Brian L. Evans and Prof. Gustavo de Veciana. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and Sigma Xi (The Scientific Research Society).

Permanent address: Malazgirt Sokak
Yetkili 2 Apt., No: 4
Kyrenia, Cyprus

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.