

THESIS

د 360 د



This is to certify that the dissertation entitled

ESTIMATING AVAILABLE BANDWIDTH FOR REAL-TIME SUPERMEDIA APPLICATIONS

presented by

ALEXANDER CHOBANYAN

has been accepted towards fulfillment of the requirements for the

Doctoral

Computer Science and Engineering

What to. 1h

degree in

Major Professor's Signature

April 30, 2007

Date

MSU is an Affirmative Action/Equal Opportunity Institution

PLACE IN RETURN BOX to remove this checkout from your record. TO AVOID FINES return on or before date due. **MAY BE RECALLED** with earlier due date if requested.

1	DATE DUE	DATE DUE	DATE DUE
1			
1			
			6/07 p:/CIRC/DateDue.indd-p.1

ESTIMATING AVAILABLE BANDWIDTH FOR REAL-TIME SUPERMEDIA APPLICATIONS

By

Alexander Chobanyan

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science and Engineering

2007

ABSTRACT ESTIMATING AVAILABLE BANDWIDTH FOR REAL-TIME SUPERMEDIA APPLICATIONS

By Alexander Chobanyan

The emerging development of remote sensing and control technology gives rise to new types of real-time networking applications that require "pro-active" quality of service guarantees. This dissertation specifics Quality of Service parameters characterizing requirements for successful execution of a given real-time tele-operation over the Internet. The dissertation also provides a statistical model that outputs the likelihood of a failure of remote real-time task execution based on the history of behavior of an end-to-end network path and the task requirements. We provided and compared both parametric and non-parametric approaches, conducted their experimental evaluation and also introduced some mechanisms for improvement of present measurement technologies of an end-to-end quality of a network path.

Available bandwidth (spare capacity) of a network path is a key characteristic of the network path quality. We regard available bandwidth as a random timeseries with an existing auto-correlation structure that is realistic and makes channel quality estimation accurate on real paths over the Internet.

We also consider a set of pro-active measures to save remote real-time task execution from a failure once the likelihood of such a failure exceeds an acceptable threshold. In particular, we give a solution to the Reroute Sequence Planning problem: to choose an optimal sequence of data-streams reallocations across different network paths once the QoS of old paths are unacceptable. Optimal reallocation minimizes extra capacity of a network link that is required for carrying out the reallocation procedure.

TABLE OF CONTENTS

	List o List o	of Tables	v vi
1	Intro	oduction	1
2	Back	rground	9
3	Para 3.1 3.2 3.3 3.4	Imetric Approach Data collection and preprocessing Modeling available bandwidth 3.2.1 "Crossing probability" 3.2.2 ARCH model 3.2.3 Results from extremal values theory Experimental evaluation Conclusion	 19 25 25 26 34 36 39
4	Netv 4.1 4.2 4.3 4.4	work One Way Delay Trend Detection Mechanism Main challenges of IC/CS elimination and trend detection in Pathload IC/CS elimination procedure and t-test Experimental evaluation Conclusion	42 42 46 51 56
5	Non- 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10	-Parametric Approach Moving block bootstrap D-statistic: model verification mechanism D-statistic: theoretical results D-statistic: experimental evaluation Prediction optimization with weights R-statistic: improvement of a verification procedure 5.6.1 Timing R-statistic: theoretical results R-statistic: theoretical results R-statistic: theoretical results R-statistic: experimental evaluation Limits of a verification procedure Conclusion	57 58 62 66 69 74 76 76 76 77 88 84 88 91
5 6	Non- 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 Re-r 6.1 6.2 6.2	-Parametric Approach Moving block bootstrap D-statistic: model verification mechanism D-statistic: theoretical results D-statistic: experimental evaluation Prediction optimization with weights R-statistic: improvement of a verification procedure 5.6.1 Timing Statistic: theoretical results R-statistic: theoretical results R-statistic: theoretical results R-statistic: experimental evaluation Limits of a verification procedure Conclusion Oute Sequence Planning Problem Introduction Setting Of The Problem And Transference Lemma Application of Transference Lemma	57 58 62 66 74 76 76 77 78 84 88 91 93 94 97

	6.5	An alg	for finding σ	. 103
7	Sun	nmary	and Future Research	106
	7.1	Summ	ary	. 106
	7.2	Future	eresearch	. 109
		7.2.1	QoS overlay network infrastructure	. 109
		7.2.2	QoS model improvement	. 109
B	BLI	OGRA	PHY	111

LIST OF TABLES

3.1	Empirically estimated "Crossing" probability for a path MSU-Berkeley	37
3.2	Difference between model-based estimated upper bound and empiri- cally estimated "Crossing" probability for a path MSU-Berkeley	39
4.1	Comparison of PCT/PDT test and t-test performances on nearly empty network and zero receiver's processor utilization	54
4.2	Comparison of PCT/PDT test and t-test performances on nearly empty network and $5 - 7\%$ receiver's processor utilization	54
4.3	Comparison of PCT/PDT test and t-test performances on heavy loaded network and zero receiver's processor utilization	54
4.4	Comparison of PCT/PDT test and t-test performances on average	04
	loaded network (around 50Mbps $\pm 10Mbps$ cross-traffic) and zero receiver's processor utilization	55
5.1	Valid range of history window sizes	73
5.2	P-values for different history window sizes before and after weight adjustment.	85

LIST OF FIGURES

1.1	Estimated available end-to-end bandwidth (Mbps). The time-series is sampled at a rate one observation per minute	4
3.1	Available end-to-end bandwidth (Mbps): Michigan State University (35.9.20.14) - Massachusetts Institute of Technology	20
3.2	Available end-to-end bandwidth (Mbps): Denver (USA) 198.32.154.187 - Canarie, Montreal (Canada) planet1.montreal.canet4.nodes.planet-	
	lab.org	20
3.3	Available end-to-end bandwidth (Mbps): Michigan State University	
	(35.9.20.14) - University of California, Berkeley (169.229.50.12)	21
3.4	Available bandwidth before trend removal and inversion	23
3.5	Available bandwidth after trend removal and inversion	23
3.6	Michigan State University - University of Sydney, Australia (trend	
	example)	24
3.7	Michigan State University - University of Uupsala, Sweden on the	•
	right (routes oscillation example)	24
<u>ა.</u> გ	Measured available end-to-end bandwidth	28
3.9	Simulated ARC $H^{-}(1)$ observations with $p_0 = p_1 = 0.25$.	20
3.10	Fisher Information: standard deviation of β_0	31
3.11	Fisher Information: standard deviation of β_1	31
3.12	Fisher Information: correlation between β_1 and β_0	32
3.13	"Crossing" probability for a threshold $M=25$ Mbps and a number of	05
0.14	observations $N = 50$	35
3.14	"Creasing" probability for a noth MSU Dayleder	40
	Crossing probability for a path MSO-berkeley	40
4.1	Shortcomings of IC/CS elimination procedure in Pathload: before	
	elimination	45
4.2	Shortcomings of IC/CS elimination procedure in Pathload: after elim-	
	ination	45
4.3	OWD pattern at the presence of packet loss. Trains were sent at a	
	rate 40 Mbps higher than available at that time bandwidth.	49
4.4	OWD pattern at the presence of packet loss. Trains were sent at a	
	rate 30 Mbps higher than available at that time bandwidth.	50
4.5	OWD pattern at the presence of packet loss. Trains were sent at a	
	rate 20 Mbps higher than available at that time bandwidth	50
51	Window of size $N = 3$ slides across the history data of size $n = 12$ for	
0.1	prediction of a crossing probability P over coming $N = 3$ observations	59
5.2	Probability density functions for a p-value conditioned by Null and	
	Alternative hypothesis.	63

5.3	Evolution of the crossing probability along with its 95% confidence intervals throughout the day. $M=65 \text{Mbps}, N=5, \tau=0.43 \text{sec.}$	69
5.4	Probability density functions along with p-values collected for the route 35.9.42.90 (Michigan State University) - 137.189.100.235 (Chinese University of Hong Kong).	70
5.5	Probability density functions along with p-values collected for the route 35.9.10.14 (Michigan State University) - 210.72.135.98 (Shenyang	
	Institute of Automation, Chinese Academy of Sciences)	70
5.6	Reducing the time of self-verification procedure when switching from	
	R-statistic to D-statistic	79
5.7	Histograms of acceptable range of lengths of history windows before and after weights adjustment. The black histogram shows the range distribution after adjustment whereas the gray histogram shows the	
	range distribution before adjustment.	87
5.8	Dependency of minimal value of a crossing probability for which ver- ification procedure works well on the future time-window over which	
	prediction is made and on prediction/verification technique	90
6.1	Network graph example	95

Chapter 1

Introduction

Over the past several years the importance of Internet-related technologies has rapidly increased. The growth of the Internet has not only a quantitative nature but also affects the type of information that may be transferred and consequently the methods for real-time interaction between communicating parties. Many emerging Internet-based technologies may benefit not only by transferring traditional mediatypes, such as video and audio, but also by remotely manipulating, touching and feeling a remote object.

Tele-medicine, for example, can potentially allow a doctor to touch and feel a patient thus further expanding opportunities for conducting remote diagnostics. Tele-education opens wider possibilities to learn about different objects via touching and feeling them. Tele-marketing applications can be designed in order to allow a customer to touch the item he wants to purchase before making a financial transaction.

Video, audio, haptic, temperature, control commands and other media used for tele-operations are unified under the term "Supermedia" that can be considered as a significant extension of traditional multimedia information streams.

This dissertation has been written in the frames of a "Supermedia" project that assumes developing and experimentally testing new methods, algorithms, and tools to establish an infrastructure for real-time supermedia transport over the Internet to support tele-operations. Traditionally tele-operations take place over isolated controlled lines. In such a case, the channel has sufficient degree of reliability for conducting important teleoperations that may have high risk-cost associated with channels failure to give necessary quality of service to communicating parties.

On one hand, using the Internet as a transport for tele-operations is much cheaper and therefore more convenient. In addition the Internet is wide-spread and easily accessible medium all over the world. On the other hand, however, the Internet does not provide sufficient quality of service guarantees to communicating parties. A network path over the Internet is by its nature unreliable and uncontrolled. Throughput of such a path at any time-instance always depends on the cross-traffic present in the path. Therefore real-time tele-operation over the Internet may always have unexpected network delays, distortions or even communication failures. One can even argue that the absence of communication reliability represents a principle challenge for wide deployment of tele-operations over the Internet. We believe that if "on average" the cost of tele-operation failure is "sufficiently" small then such a tele-operation still can be performed. Under "average" failure-cost we assume the product of the cost associated with tele-operated task failure multiplied by the likelihood of such a failure to happen.

In order to support tele-operations over the Internet it is vital to develop mechanisms that efficiently estimate for any particular tele-operation the likelihood of the failure of remote real-time task execution and take pro-active measures to minimize this likelihood. Moreover, based on his knowledge of the cost associated with a failure of a particular real-time task the operator should be able to temporarily suspend real-time task execution until better times. For example, before a robot is directed to approach a wall, it is important to report to the operator the probability based on the channel's history that control over the robot is lost due to network problems while the robot approaches the wall. Then the operator makes a decision whether he should begin the wall-approach immediately or to wait until better times.

In this thesis, an available bandwidth (AB) of a network path is considered as a primary QoS characteristic of the path and probabilistic models of tele-operation performance are built around this quantity. Available Bandwidth may be briefly defined as the maximum rate at which a sender can put bits on a wire without affecting existing cross-traffic, or in other words, as a residual capacity of the network path.

Previous work [29] concentrated on obtaining the value of the available bandwidth averaged over a reasonably small time-window without an attempt to provide a broader probabilistic picture characterizing the network path. Figure 1.1 shows an example of how available bandwidth between two nodes located 20 hops apart may behave over time. As depicted in the plot, available bandwidth may instantaneously drop far below its mean value thus causing degradation of service, such as distorted video or audio. Despite the fact that such "spikes" may last just a few seconds, they may be fatal for many real-time remotely-operated tasks. As it follows from figure 1.1, the value of the available bandwidth averaged over an arbitrary large (small) time-window cannot characterize the network path well in terms of its bandwidth QoS properties. The available bandwidth time-series is directly related to the cross-traffic that is present in the path. It has been shown [34] that the network traffic has self-similar properties that imply a non-summable autocorrelation function and long-range dependence between observations. Therefore observations of available bandwidth cannot be considered as independent, which makes statistical inference more challenging.

The principal novelty of approaches introduced in this dissertation is in viewing available bandwidth as a random heavy-tailed time-series with existing autocorrelation structure instead of reporting just one expected value while assuming that it does not change over the measurement time-period. More precisely, this dissertation introduces a new (in the context of available bandwidth time series) QoS quantity called "crossing probability" - the probability that the available bandwidth time-series becomes less than pre-defined threshold over pre-defined time-frame.

This dissertation introduces parametric and non-parametric statistical models that predict "crossing probability" for a pre-defined threshold and future time-frame. Also the dissertation introduces significant improvement of basic block of present



Figure 1.1: Estimated available end-to-end bandwidth (Mbps). The time-series is sampled at a rate one observation per minute.

probe-rate based available bandwidth measurement tools.

The proposed parametric model provides an important *probabilistic* QoS channel specification in terms of coefficients that allow rapid computation of "crossing probability". The aforementioned coefficients are evaluated based on observed available bandwidth behavior over a reasonable period of time and are updated at run-time reflecting the changing dynamics of a network environment. Both, the critical value and the time-frame for which we predict should be defined by the QoS specifications of a particular real-time task.

The accuracy of the "crossing probability" prediction may vary from path to path and over time. However, the notion of "crossing probability" on its own is conceptually novel with respect to the available bandwidth estimation. Real time applications can benefit from this approach by reallocating part of the QoS-sensitive channels over different network paths when the crossing probability exceeds an acceptable threshold. Channel reallocation in this scenario happens before something fatal occurs regarding the real-time task execution but after the probability for this fatal event exceeds a reasonable value.

Despite its own advantages, the parametric approach relies on values of available bandwidth reported by presently known measurement tools and for many network paths these values may not be accurately reported, which significantly limits the range of its applicability. Present measurement tools need around one minute for an accurate reporting of one observation of an available bandwidth time-series. Consequently, in order to collect a statistically significant number of observations one has to consider the history window of at least 1-2 hours. On another hand, our parametric model assumes that the observed available bandwidth time-series is weakly stationary throughout the history time-frame. In order to make this assumption realistic we have to choose a history window size as small as possible. Our experiments showed that for many network paths the aforementioned trade-off for a history window size exists. For such paths the parametric model does its job reasonably well.

However, for many network paths the sampling rate of available bandwidth measurement tools is too small. In addition, when dealing with robotic applications there is a need to make predictions for future windows of the order of seconds whereas a parametric approach works for scales of the order of minutes.

The aforementioned factors pushed us to research on improvement of present available bandwidth measurement tools towards decreasing measurement time while preserving measurement accuracy.

Measurement of available bandwidth (even as a single value) is a complicated task since cross-traffic in the network path dynamically changes thus causing change of the available bandwidth. The best that present measurement techniques can do is to measure the *mean* available bandwidth averaged over a certain period of time τ . Decreasing the measurement time τ is an important issue because the assumption that available bandwidth does not change over a smaller period of time is weaker and more realistic rather than the assumption that the bandwidth does not change over larger periods.

Presently two types of approaches to measurement of available bandwidth exist. Probe-gap models [56, 45] send a pair of packets and make further inference about available bandwidth based on the measurement of a gap between packets at the receiver's side. Probe-rate models [29, 46, 42] send sequences or "trains" of packets at different rates assuming that a sequence sent at a rate higher than the available bandwidth causes noticeable positive trend in times that packets in the sequence need for travelling from sender to receiver. These times are often called one-way-delays (OWDs). Then the available bandwidth at that time can be found by a simple binary search. Probe-gap approaches are less intrusive than probe-rate approaches. However, probe-gap approaches suffer from the so called "interrupt coalescence / context switch (IC/CS)" effect [48], which significantly distorts measurements of packets inter-arrival times. Probe-rate models are affected by IC/CS phenomenon as well but not as much as probe-gap models. Probe-rate models send a larger number of packets and therefore an IC/CS effect can at least be clearly observed and to some extent considered by a model deployed in the probe-rate measurement tool.

This dissertation introduces a new metric for OWD trend detection of packets in a train together with a new method for dealing with the IC/CS effect. We compare our results with Pathload [29], which to our knowledge is the only relatively reliably working tool that considers the IC/CS phenomenon. We demonstrate that our metric used for trend detection together with our way of dealing with interrupt coalescence and context switch effects provides much better OWD trend detection accuracy. It can also better tolerate processor utilization of a receiving machine and therefore can serve as a basic block not only in Pathload but also can improve a widerange of probe-rate oriented available bandwidth measurement tools that presently use trend detection algorithm introduced by Pathload's authors. In particular, with our approach in most cases we need to send only up to three trains while achieving the same accuracy as Pathload when it sends 6 trains. We therefore decrease the measurement time without sacrificing accuracy.

The dissertation deploys a robust OWD measurement module (described above) for developing a more general non-parametric approach for computation of a "crossing probability". In contrast to above-mentioned parametric approach that relies on measurements of the AB time-series made by Pathload [29], the non-parametric approach relies directly on measurements of indicators showing whether the AB time-series is smaller or larger than a given value. Let us note that every particular measurement of available bandwidth made by Pathload is obtained by a binary search [29], which itself relies on measurements of a number of indicators mentioned above. Therefore the proposed non-parametric approach is more robust, allows faster computation of a crossing probability than described earlier parametric approach. Also non-parametric approach has a larger range of applicability since it does not rely on any parametric time-series model.

For computation of a "crossing probability" estimate we use modern non-parametric techniques of statistical analysis of a random time-series such as a moving block bootstrap [39]. Moving block bootstrap considers dependence [34] between observations of indicators and helps in providing 95% confidence intervals around the non-parametric estimator of a "crossing probability". Since the main goal of our research is prediction of a "crossing probability," we also introduce two methods for a verification of the accuracy of our prediction based on training and previously unseen testing data-sets obtained by a real-life measurements across multiple Internet paths. Along with real-life experiments, we present theoretical results that show that our verification procedures are probabilistically legitimate.

In addition to "crossing probability" estimation, this dissertation deals with a set of actions that communicating parties might undertake in order to keep teleoperation alive given that "crossing probability" increased above critical threshold. A real-time application may reallocate, for example, some of data streams through different routes that may have better QoS characteristics at given time-period. In this dissertation we provide preliminary results related to improvement of the solution of a Reroute Sequence Planning problem [41] (RSP). The RSP problem chooses an optimal sequence of reallocations of an old set of data-stream paths to a new set of data-stream paths that minimizes extra network-link capacity needed for carrying out of streams reallocation. We provide theoretical results on the bound of the extra needed capacity and also give a sketch of an algorithm for performing reroute sequence planning.

The rest of this dissertation is structured as follows. Chapter 2 provides definitions and describes previous available bandwidth measurement tools. Chapter 2 also gives a brief background on weak stationarity, self-similarity and describes heavytailed long-range dependent models that were used for network traffic modeling. Chapter 3 introduces the notion of a "crossing probability" and describes the parametric model for "crossing probability" estimation. Chapter 4 deals with an improvement of the previous available bandwidth measurement tools with a new One Way Delay trend detection mechanism. Then Chapter 5 introduces a non-parametric approach to "crossing probability" prediction that uses a measurement module introduced in Chapter 4. Also chapter 5 introduces model verification mechanisms that help to check on real data-sets wether predicted values of a "crossing probability" really match on-coming predicted events of a crossing. Chapter 6 introduces Reroute Sequence Planning problem and corresponding mathematical problem of Compact Vector Summation. The chapter gives theoretical results for a link capacity bound and provides an algorithm for finding a sequence of replacements of paths for data streams that complies mentioned above bound for a capacity. In Chapter 7 we summarize results of this dissertation and then talk about our future work.

Chapter 2

Background

We consider a network path as a set of store-and-forward links L_i , i = 1, 2, ..., k. The capacity of the *i*-th link C_i may be defined as a maximum bit rate at which the sender may transfer information through this link when no other traffic is present. Now assume that the link is partially used by another communication. Let T(t) be defined as the number of bits that have been transferred through this link by time t. The spare capacity of link i, or in other words, its available bandwidth $A_i(t)$ at time t can be defined as $A_i(t) \equiv C_i - \lim_{\tau \to 0} \frac{T_i(t+\tau) - T_i(t)}{\tau}$ or: $A_i(t) \equiv C_i - \frac{dT_i}{dt}(t)$. Then the bandwidth A(t) available at time t of a path can be defined as $A(t) \equiv \min_{i=1,...,k} A_i(t)$. From the definition it follows that available bandwidth cannot be measured precisely and therefore needs to be approximated by the mean available bandwidth at time t for the interval τ .

$$A_{\tau}(t) = \min_{i=1,...,l} \{ C_i - \frac{T_i(t+\tau) - T_i(t)}{\tau} \}$$

The period of time τ over which $A_{\tau}(t)$ is averaged depends on the particular network application. If just one value of $A_{\tau}(t)$ is reported, then a smaller τ does not necessarily guarantee better results because the measurement accuracy degrades with decreasing τ . However, with smaller τ , we have more flexibility when describing the probabilistic properties of available bandwidth behavior. Naturally, with a higher sampling rate one can acquire more "behavioral" information while still considering measurement errors within the frames of a chosen probabilistic model. In general, the approximation of available bandwidth is QoS relevant if τ is less than the time that the receiver needs to empty its buffer. Fluctuations of the available bandwidth should not cause degradation of quality at the receiver as long as those fluctuations do not affect the number of bits that the receiver needs to receive to keep its buffer non-empty.

Significant work that has been done in mean available bandwidth estimation is well reflected by Strauss, et al. [56] and Hu, et al. [27]. In short, deployed bandwidth estimation models can be divided into probe gap (or packet dispersion) and probe rate classes.

Probe gap models used in Spruce [56] and Delphi [50] analyze delay between probe packets and make further inference about available bandwidth based on that delay. Probe gap models can be further divided into direct and iterative approach groups.

Tools utilizing the direct approach [50, 56] sample available bandwidth directly analyze either delays between packet arrivals or packet travel times. An iterative approach assumes step-by-step convergence to the tentative range where the available bandwidth resides. Each step represents a comparison of available bandwidth against a specified rate-value.

The classical direct approach assumes fluid cross-traffic as well as an existence of a tight and narrow link with known capacity C. We accept terminology where a "tight" link is the link with smallest available bandwidth among all links in the path whereas the "narrow" link is the link with the smallest among links in the path capacity. Fluid cross-traffic model assumes that at the input of a router during any arbitrarily small time interval θ arrives the cross-traffic of $R_c \theta$ bytes where R_c is the average cross-traffic rate for a time-period θ .

The formula for available bandwidth A can be derived as follows. If R_c exceeds an available bandwidth A then there will be an increase in a queue of the router at the tight and narrow link caused by measurement packets. Let δ_1 be a gap between packets in incoming sequence and δ_2 be a gap between packets after the tight and narrow link. Also let Δq be the queue increase for the time δ_1 and L be the size of a measurement packet. Then Δq can be expressed as $\Delta q = (R_c - A)\delta_1$ where R_c can be viewed as an arrival rate whereas A can be viewed as a service rate for the measurement data. Also one can write: $\delta_1 = L/R_c$ and therefore $\Delta q = \frac{L(R_c - A)}{R_c}$. Then taking into account that capacity C is the rate at which router services input queue (measurement packets + cross-traffic) we can state that increase in the queue for the time δ_1 can be processed for the time

$$\Delta d = \frac{\delta q}{C_t} = \frac{L(R_c - A)}{R_c C_t}$$

Then gap between packets at the output of the slow link can be expressed as

$$\delta_2 = \delta_1 + \Delta d = \delta_1 + \frac{L(R_c - A)}{R_c C_t}$$

Taking into account that $R_c = \frac{L}{\delta_1}$ one concludes that the only unknown parameter in the above equation is the available bandwidth A. Therefore available bandwidth can be expressed as

$$A = C_t - \frac{L}{\delta_1} (\frac{C_t \delta_2}{L} - 1)$$

In the language of rates the above formula can be rewritten as

$$A = C_t - R_c (\frac{C_t}{R_{out}} - 1)$$

where $R_{out} \equiv \frac{L}{\delta_2}$ is the rate at which measurement packets are received. A fluid traffic model approach implies that the number of packets in the sequence sent with the rate R_c does not significantly affect measurement accuracy. One may even conclude that an analysis of a pair of packets demonstrates performance similar to analysis of a large sequence of packets.

Despite seeming advantages of fast sampling of exact values of available bandwidth, the direct approach has several significant drawbacks which according to our point of view makes it inapplicable to available bandwidth measurement. First, there might not exist a revealed tight and narrow link in the network path. Even if such a link exists then its capacity still may not be known. Second, if the packet-size of a cross-traffic is sufficiently large then amount of cross-traffic inserted between measurement packets should be represented by discrete values which significantly falls out of the fluid model. Jain, et al. demonstrated in [31] that measurement error increase from 2% to 40% when cross-traffic packet-size is 1500B and the number of packets in the measurement sequence drops from 100 to 10. However, an increase of number of packets in the measurement sequence also is faulty because of the influence of the long measurement queue onto itself [40].

An iterative probe-gap approach [44, 43, 27, 49] is also based on the assumption that if the rate at which measurement sequence is sent exceeds available bandwidth then gaps between measurement packets increase. The iterative approach, however, does not rely on a known capacity of the tight and narrow link and does not make direct inference about value of available bandwidth based on a single measurement. If gaps between measurement packets increase then the only conclusion made by an iterative approach is that the available bandwidth is lower than the rate at which packets were sent. A variation of this approach estimates rates at which packets were sent and received. Then available bandwidth is believed to be lower than the rate at which packets were sent if the rate at which packets were received is significantly lower than the rate at which they were sent. Various search methods are applied for figuring out what is an approximate value of the available bandwidth. Hu and Steenkiste for example do linear rate variation [27] whereas the Pathchirp measurement tool [49] exponentially increases inter-arrival times between successive packets.

All above-mentioned tools and techniques, however, do not consider the effect of cross-traffic burstiness. Jain and Drovolis note in [31] that basic queueing theory says that queues can increase even before server utilization reaches 100%. And as it was shown by Liu, et al. [40], non-empty fluctuating queues at routers are responsible for constant underestimation of an available bandwidth by all above-mentioned probe-gap based iterative techniques. Such an underestimation happens because if the

rate at which packets are sent begins to increase then gaps between measurement packets begin to increase far before the rate at which packets were sent reaches the point of an available bandwidth.

Probe rate iterative models also utilize sequences or trains of packets sent at different rates. The key difference between probe-rate approaches and probe-gap models is that probe-rate approaches work with packet travel times or so called One Way Delays (OWD) for determining whether the rate at which sequence of packets was sent is higher or lower than the available bandwidth.

For example, the probe rate model used in Pathload [29] uses trains of packets sent at different rates. If the delay between packets at the receiver shows an increasing trend, then this trend may be considered as an indicator that the rate at which packets were sent is higher than the available bandwidth at that moment. If there is no indication of an increasing delay trend then the rate is smaller than the available bandwidth. Pathload then does a binary search to find the rate that is approximately equal to the available bandwidth. As it was stated above, fluctuations of queues at routers can increase gaps between packets. However, a highly revealed OWD positive trend in a sufficiently large sequence of packets unambiguously points to the inability of one of the routers in the network path to serve its queue, which can be interpreted as a lack of available bandwidth of the path with respect to the measurement traffic rate.

Both probe rate and probe gap measurement approaches suffer from known interrupt coalescence / context switch (IC/CS) effects [48]. If interrupt coalescence is enabled at the receiver's network adapter, then the adapter waits for a group of packets to arrive and only after that issues an interrupt request. Then the entire group is processed with one interrupt. The direct IC impact on network measurements is that packets are not time-stamped by a receiver after they really arrived but with a variable delay. Some may wait for an interrupt request for up to hundreds of microseconds, which significantly influences network measurements. Context switch effects have the same impact. If a processor is utilized by another job, then a certain random time-period is passed after a packet's arrival before it is processed and time-stamped by a receiver.

It is fairly complicated to consider the IC/CS influence when only pairs of packets are sent. To our knowledge there are no known probe-gap (packet dispersion) based available bandwidth measurements that consider the IC/CS effect. Pathload is the only tool from the probe-rate approaches that partially addresses the IC/CS issue. We therefore here-and-after focus our attention on Pathoad as the most stable and accurate end-to-end available bandwidth measurement tool to our knowledge.

Pathload checks the time that a system needs for delivering packets from network adapter to upper layers. All packets that are received within this time-frame are assigned to one group or burst. Then in a packet train, Pathload counts only one packet out of each burst and discards the rest of the packets.

After IC/CS related filtering, Pathload divides the remaining N packets into $\Gamma \approx \sqrt{N}$ groups and takes the median from each group. The remaining packets are discarded as well. Finally two metrics are computed based on remaining Γ measurements. The Pairwise Comparison Test (PCT) metric is evaluated as

$$S_{PCT} = \frac{\sum_{k=2}^{\Gamma} I(M_k > M_{k-1})}{\Gamma - 1}$$

where I is an indicator function evaluated to one when the median of the k-th group M_k is larger than the median of the k-1st group and to zero otherwise. As it follows by definition if a train has an increasing trend then the PCT metric approaches 1. If there is no trend, then the PCT metric should be close to zero. The Pairwise Difference Test (PDT) metric is defined as

$$S_{PDT} = \frac{M_{\Gamma} - M_{1}}{\sum_{k=2}^{\Gamma} |M_{k} - M_{k-1}|}$$

If the train does not have an increasing OWD trend then S_{PDT} should vary around zero, whereas in a presence of an increasing trend it should be close to one. The PDT metric was introduced for detection of a trend with a significant start-to-end OWD difference. Finally, two tests together decide what to report for a given train of packets.

The PCT and PDT tests represent a basic mechanism of trend detection in Pathload. In general, the number of packets in a train, the train packet size, and the bandwidth-rate search mechanism may vary among approaches. For example, Man, et al. [42] introduce a light-weight train-based tool that is less intrusive than Pathload on one hand but sacrifices accuracy on the other hand. Nishikawa, et al. [46] propose to evaluate the approximate distribution of available bandwidth as of a random quantity. Basic metrics in these works remained, however, the same as they were introduced first in Pathload.

Pathload is believed to be slightly more intrusive than tools using the probe gap model. To our knowledge Pathload is, however, the most stable and accurate among available bandwidth measurement tools to date. Also Pathload creators propose a way to consider the IC/CS influence and therefore Pathload is to our knowledge the only tool that considers an IC/CS effect.

The accuracy of Pathload is nevertheless far from ideal. Recent research [29] have proved that Pathload's measurements are much more accurate for higher values of available bandwidth than for the lower values. Also our experience with Pathload shows that for many network paths, in a long run, Pathload reports many missing values and sometimes is unable to catch instantaneous rapid drops of available bandwidth that are critical for QoS inference about the channel. Pathload, up to a certain degree, considers IC/CS influence, but still its performance significantly depends on the processor utilization of measurement machines. Therefore for our experiments we had to use machines with a very light processor utilization factor.

From the definition of available bandwidth it follows that it is tightly related with the cross-traffic present at the moment in the network path. Cross-traffic defined as $\frac{dT}{dt}(t)$ represents a random time-series that has been extensively analyzed by both statisticians and network engineers for its statistical properties. If the tight link remains unchanged throughout the measurements (that is usually the case) then the available bandwidth time-series can be evaluated by subtracting the cross-traffic from a tight link's unchanged capacity. It follows that in the aforementioned case the available bandwidth and cross-traffic are exactly the same time-series in terms of their statistical characteristics. Therefore we believe that the model chosen for available bandwidth characterization should incorporate at least basic properties of models that are used presently for Internet traffic modeling. Below we provide some background of such models.

All parametric traffic models assume weak stationarity of the traffic time-series. Weak stationarity can be defined as follows: Let $\{X_t\}$ be a random time-series that takes values $x_1, x_2, x_3, ..., x_n$ at different instants of time. Generally speaking, all these values are observations of different, dependent random variables $X_1, X_2, ..., X_n$. We say that random variables $X_1, X_2, ..., X_t, ...$, observed over time, form a weakly stationary time-series if the following conditions are satisfied: expected values of all X_t 's are the same, and the $\gamma_X(t, h)$ defined as $Cov(X_{t+h}, X_t)$ does not depend on t. $\gamma_X(t, h) = \gamma_X(h)$ is called the autocovariance function of the time-series (ACVF) at lag h. $\rho_X(h)$ defined as $\rho_X(h) \equiv \frac{\gamma_X(h)}{\gamma_X(0)}$ is called an autocorrelation function (ACF) of the time-series at lag h. Weak stationarity means that the correlation picture remains the same under any shift in the time domain. ACF therefore can be treated as a basic characteristic of a weakly-stationary time-series.

Leland, et al. [34] showed that network traffic possesses an important characteristic called self-similarity. Self-similarity can be formally defined as follows: let $\rho_X(h) \sim h^{-\beta}$ as $h \to \infty$, $0 < \beta < 1$. For an integer $m = 1, 2, ..., \infty$ let $\{X_k^{(m)}\}$ be new weakly stationary series defined as $X_k^{(m)} = \frac{1}{m}(X_{km-m+1} + ... + X_{km})$. The time-series $\{X_t\}$ is called second-order self-similar with selfsimilarity parameter $H = 1 - \frac{\beta}{2}$ if for all $m = 1, 2, ..., \infty$ $Var(X(m)) \sim m^{-\beta}$ and $\rho_X(m)(h) = \rho_X(h)$ for all h > 0. Most important implications of self-similarity are "slowly" (slower than 1/h) decreasing autocorrelation implying long-range dependence $(\sum_{h=0}^{\infty} \rho_X(h) = \infty)$, and "slowly" decaying variance of the so called aggregated time-series ($\{X^{(m)}(k)\}$ that was obtained by averaging the initial timeseries over groups of size m).

As a motivation for self-similar characteristics of a network traffic one may consider an efficient modeling of a network traffic by a superposition of several "ON/OFF" sources. Each of such a source can reside in two states: "ON" state when it intensively transfers packets and "OFF" state when it is idle. The model assumption is that the lengths of activity periods are independent identically distributed random variables. If each source "is driven" by the same distributions then it becomes possible to model the behavior of superposition of several such sources. The important achievement made by researchers from Bellcore was to model "ON/OFF" periods with the help of infinite variance distributions (such as Pareto with parameter α in the range from 1 to 2). Superposition of several such "ON/OFF" Pareto sources allows to derive a self-similar traffic model.

There exist many self-similar models by which researchers approximated Internet traffic. Crovella and Bestavros report in [17] that traffic formed by web browsers possesses self-similar characteristics. In their study they analyzed more than 500,000 requests generated by 37 browsers. Each web-browser was modeled as an ON/OFF source. In particular, it was shown that browser data match heavy-tailed Pareto distribution. Similar results were shown in [5] and in [4].

Borella and Brewster [8] use UDP packets for measuring packet delays and showing that delays demonstrate long-range dependent nature. Another study [23] shows that FTP data rate is bursty and the distribution of the number of bytes in bursts has heavy tails.

Several works show that data associated with video traffic also has self-similar characteristics. Video compression algorithms assume frames of variable sizes, which allows to view video traffic as a random time-series. Regular video has scenes with a lot of motion, and scenes with a few of motion which makes distribution of data in frames heavy-tailed. Garrett and Willinger, for example show in [21] that transfer of the "Star Wars" movie coded in JPEG format reveals self-similar nature and that length of the frame follows heavy-tailed Pareto distribution. Authors show, in particular that high degree of variability in frame length is associated with scenes variability.

In another study Beran at al [7] analyzed 20 different Variable Bit Rate video transfers, which were formed with various coding mechanisms. They showed that long-range dependency exists in all 20 transfers and does not depend on type of video-encoding or on used scene-types. Tsybakov and Georganas [57] report similar results.

Zukerman, et al. [60] proposed to use the Poisson Pareto Burst process model for an Internet traffic. In addition to the mean, variance, and self-similarity parameter H, the model has an extra "aggregation" parameter λ . Zukerman, et al, believe that their aggregation parameter significantly improves the accuracy of the model. Li, et al. [36] studied characteristics of streaming media stored in the Internet and came to conclusion that it has long-tailed distribution and therefore may contribute self-similar Internet traffic. Aracil, et al. [3] studied FTP and Telnet data arrival time-series and came to the conclusion that it is self-similar and can be matched to a fractal renewal process.

Traffic modeling studies have constructed models that visually look similar to observed time-series and have autocorrelation structures similar to what has been observed. Also, estimated model parameters based on observed data should remain unchanged or slowly vary over time. Finally, the model should be relatively simple and amenable for further statistical analysis. We will follow the same strategy to unify recent work in Internet traffic analysis, available bandwidth estimation tools, and time-series study with respect to analysis of the QoS characteristics of a network path.

Chapter 3

Parametric Approach

This chapter introduces the notion of a "crossing probability". It emphasizes its importance for tele-operations requiring "pro-active" reaction to QoS degradation, and gives a parametric model for "crossing probability" estimation based on past observations of an available bandwidth time-series.

3.1 Data collection and preprocessing

Figures 3.1, 3.2, 3.3 depict the available bandwidth behavior for the period of 800 minutes related to various network paths collected with the help of a Pathload measurement tool. The distance between nodes varies from 4 to 20 hops. For all examples in figures 3.1, 3.2, 3.3 we observe a small trend that is less than 1 Mbps around the mean value. Also, all examples have bursts reflecting the heavy-tailed self-similar nature of Internet traffic-related series. This work is related to available bandwidth time-series that are stationary. Therefore, we assume that the mean of the time-series does not change or, in another words, the time-series of interest has an insignificant trend.

Sometimes the available bandwidth series may acquire a trend significantly exceeding a reasonable limit (say 5 Mbps). Figures 3.6 shows an example of such a network path (Between nodes located in MSU and the University of Sydney, Australia). The existence of such a trend naturally violates the stationarity assumption



Figure 3.1: Available end-to-end bandwidth (Mbps): Michigan State University (35.9.20.14) - Massachusetts Institute of Technology



Figure 3.2: Available end-to-end bandwidth (Mbps): Denver (USA) 198.32.154.187 - Canarie, Montreal (Canada) planet1.montreal.canet4.nodes.planet-lab.org



Figure 3.3: Available end-to-end bandwidth (Mbps): Michigan State University (35.9.20.14) - University of California, Berkeley (169.229.50.12)

which makes impossible to apply any stationary time-series model for further analysis. We note, however, that if the trend is sufficiently small for the period of time used for the analysis then stationary models still can be applied. For example, if we use historical data of the last hour for prediction of what may happen in coming 30 minutes and if the ongoing trend is insignificant for the period 1.5 hours relevant to prediction then the stationary model still can be applied.

For further fitting of stationary self-similar models, we preprocess the timeseries of interest by subtracting the trend. For all examples considered in figures 3.1, 3.2, 3.3, the trend-subtraction changes each observation for less than 2 Mbps, which is negligible with respect to spikes that are the primary cause for instantaneous unexpected degradation of service. Predicting the probability of such a spike to occur over a certain QoS sensitive time-slice is our primary goal. For the aforementioned purpose we can neglect 2-3 Mbps trend and further consider the "adjusted" time-series. Figures 3.4 and 3.5 show the original time-series representing an available bandwidth between nodes located in MSU and Berkeley along with the corresponding "adjusted" time-series. Figures 3.4 and 3.5 show that trend-removal, inversion and "lining-up" to zero level do not significantly change the behavior pattern of the original time-series of interest. For trend-removal we have used the Spencer 15-point moving average filter described by Brockwell, et al. [11] that skips polynomials of degree three and smoothes everything that has a larger degree.

A Spencer filter can be briefly introduced as follows. Let $\{m_t\}$ be time-series representing the trend whereas $\{X_t\}$ is the original series. Then $\{m_t\}$ obtained by the Spencer filter can be evaluated as follows. $m_t = \sum_j a_j X_{t-j}$, where $a_j = 0$ if |j| > 7, $a_j = a_{-j}$ if $|j| \le 7$ and $[a_0, a_1, ..., a_7] \equiv \frac{1}{320}[74, 67, 46, 21, 3, -5, -6, -3].$

It is also undesirable to have instantaneous spikes affecting trend estimation. Therefore a "trimming" procedure is applied before passing data through the Spencer filter. The whole procedure can be done at run-time. The size of the window that slides across time-series observations is 19. We delete two maximal and two minimal observations, and pass the remaining 15 observations to the Spencer filter that outputs the estimated trend. Then the trend is subtracted from the original time-series, the result is inverted, lined-up to zero, and thus finally the "adjusted" time series is obtained.

The following discussion is related to the analysis of the "adjusted" time-series. Note that in most cases the "adjustment" procedure does not significantly distort data. Even in cases such as the one which is shown in figure 3.6, if the existing trend has a "long-term" nature then we still can apply the inference about spikes made for the "adjusted" time-series to spikes of the original time-series. In such cases the "zero-level" for the "adjusted time-series" during different parts of a day corresponds to different "base-levels" of an available bandwidth. Under the term "base-level" we assume the level of an available bandwidth from which spikes of "adjusted" time-series have to be subtracted in order to obtain an observed value of an available bandwidth. For examples on figures 3.1, 3.2, and 3.3 as well as for majority network paths that we observed, the base-level remains the same during the day. For the case shown in figure 3.6 the base-level changes throughout the day. One, however, still can make predictions based on a smaller time-scale.



Figure 3.4: Available bandwidth before trend removal and inversion



Figure 3.5: Available bandwidth after trend removal and inversion



Figure 3.6: Michigan State University - University of Sydney, Australia (trend example)



Figure 3.7: Michigan State University - University of Uupsala, Sweden on the right (routes oscillation example)

We conclude that if the slope of on-going trend is sufficiently small (with respect to time-series variability) to assume that the "base-level" of an available bandwidth estimated during the history time-frame remains the same during the time-frame for which the prediction is made, then trend subtraction does not affect the accuracy of the prediction.

We do not discuss the rare cases when the routers on the network path frequently change routes for packets, which makes the available bandwidth time-series unstable, such as depicted in figure 3.7.

3.2 Modeling available bandwidth

3.2.1 "Crossing probability"

Available bandwidth is an important QoS characteristic of a network path. As it was mentioned in previous sections, the mean available bandwidth is not enough to reflect the complex nature of bandwidth dynamics. We believe that the key point is to create a statistical model for available bandwidth that efficiently characterizes the network path in terms of its QoS properties.

Assume that the operator is about to decide whether to perform a real-time QoS critical job over a potentially unreliable network path. Then the most important question can be stated as follows. What is the probability of experiencing a QoS critical degradation of service during the time required for a given real-time task execution? An example is the probability that the picture transferred to the operator from a remotely-controlled walking robot distorts to a point when operator is not able to understand where the robot is located with respect to an approaching wall.

Now let us slightly formalize the problem. The natural assumption is that the operator should be aware of the real-time task duration, minimal allowed transmission rate and the receiver's buffer size. These three metrics can be considered as basic QoS characteristics of a given task. The QoS requirements are violated if the

mean available bandwidth time-series averaged over time

 $\tau = \frac{receiver's \ buffer \ size}{minimal \ allowed \ transmission \ rate}$

drops below the threshold of minimal allowed transmission rate during the task's execution time. Note that the task's execution time will correspond to a sequence of observations of the mean available bandwidth time-series of a length

$$N = \frac{task's \ duration}{\frac{receiver's \ buffer \ size}{minimal \ allowed \ transmission \ rate}}$$

Now rephrase the main QoS question as follows: What is the probability that the minimal value over the next N consecutive observations of the available bandwidth (AB) time-series drops below a given threshold where both the threshold and N are specified by the real-time task constraints? For the "adjusted" and inverted mean available bandwidth time-series, the aforementioned probability can be redefined as the probability for the *inverted AB-time-series maximum* over the next consecutive N observations to *exceed* the pre-defined QoS critical threshold. We believe that finding this probability should be an ultimate goal and a statistically correct problem statement for the QoS characteristics of a network path. We hereand-after refer to this probability as to a "crossing probability". Even finding a reasonable upper bound for such a probability is a good contribution to the QoS characterization of a network path. Below we build a model that describes the observed available bandwidth data. It is self-similar and amenable to analysis with respect to the aforementioned "crossing probability".

3.2.2 ARCH model

In this work we propose a parametric model that allows quick computation of the "crossing probability" for any given number of future observations N and threshold M based on estimated model parameters. We used the " $ARCH^2(1)$ " model [20] for the description of the mean available bandwidth data. Therefore we first give below
some background on ARCH models and also show reasons why this model has been chosen as a candidate for the behavior of mean available bandwidth description.

The ARCH(p) (Auto-Regressive Conditional Heteroscedasticity) time-series model was introduced by Engle [20] in applications to finance events. The ARCH(p) model can be defined as an *autocorrelated* Gaussian noise with dependency between observations described as follows.

$$X_t = (\sqrt{\beta_0 + \sum_{i=1}^p \beta_i X_{t-i}^2}) Z_t$$

where Z_t are independent observations following a standard normal distribution. The simplest partial case of the ARCH(1) time-series is defined as

$$X_t = (\sqrt{\beta_0 + \beta_1 X_{t-1}^2}) Z_t$$

Finally, $ARCH^2(1)$ time-series will be defined as

$$Y_t = (\beta_0 + \beta_1 Y_{t-1}) Z_t^2$$

From the definition of $ARCH^2(1)$, it follows that the model is completely defined by a pair of coefficients $[\beta_0, \beta_1]$. Both ARCH(1) and $ARCH^2(1)$ models were extensively applied to finance-related time-series, which at least visually have the similarity to "adjusted" mean available bandwidth time-series pattern. Before going any further we provide a picture with "adjusted" mean available bandwidth on the right and simulated $ARCH^2(1)$ time-series on the left. As shown in figures 3.8, 3.9 the pattern of both time-series is the same. Another encouraging factor, as shown by Embrechts, et al. [19], is the $ARCH^2(1)$ time-series possesses self-similar characteristics. The $ARCH^2(1)$ model allows further computation of a "crossing probability" based on the estimated model coefficients. The mathematical analysis of $ARCH^2(p)$ models, where p > 1, is much more complex and there are no known mathematical results about "crossing probability" computation for such models. The model's amenability to further mathematical analysis was the crucial argument in favor of



Figure 3.9: Simulated $ARCH^2(1)$ observations with $\beta_0 = \beta_1 = 0.25$.

considering the case when p = 1. However, we conducted simple statistical data analysis aiming to show that collected data do not give any significant evidence for choosing ARCH models with p > 1. As it is known from time-series studies [11] the following quantity called AICC is considered for choosing an optimal order p:

$$AICC = -2L(\beta_0, \beta_1, ..., \beta_p) + \frac{2(p+1)n}{n-p-2}$$

where L is Quasi Log Likelihood Function evaluated as:

$$\begin{split} L(\beta_0, \beta_1, ..., \beta_p) &= -\sum_{i=1}^{n-p} [\log(\beta_0 + \beta_1 X_{i+p-1}^2 + ... + \beta_p X_i^2) + \\ &\frac{X_{i+p}^2}{\beta_0 + \beta_1 X_{i+p-1}^2 + ... + \beta_p X_i^2}] \end{split}$$

and *n* is the number of observations. The AICC criterion is a function of $p, \beta_0, ..., \beta_p$. The "best-fit" order *p* is the value that minimizes the AICC criterion for $\beta_0, ..., \beta_p$, which further minimize *AICC* for a given *p*. We computed the *AICC* criterion for 30 non-intersecting blocks of observations belonging to a real available bandwidth time-series. Each block contains 200 observations that corresponds to 3.3 hours given that the available bandwidth time-series is sampled at a rate 1 observation per minute. The size of the block was chosen large enough to be representative and small enough for assuming that model coefficients remain unchanged throughout block. For each data-block, the *AICC* criterion was evaluated for p = 1, 2, and 3. Below is shown an *AICC* minimizing *p* value for each block.

- 1, 1, 1, 3, 1, 2, 2, 3, 3, 1, 3, 1, 1, 2, 2,
- 2, 2, 3, 3, 1, 3, 1, 2, 1, 1, 2, 3, 2, 3, 1

Out of 30 trials we have 12 cases when p = 1, 9 cases when p = 2, and 9 cases when p = 3. From above values it can be noticed that the data do not give us sufficient evidence to believe that ARCH models with p > 1 work better than the one with

p = 1. Then following the well-known paradigm (Occam's razor) telling that among models with relatively the same performance, the best model is the simplest one. We conclude that our decision to choose for our model p = 1 is reasonable.

We, finally, conducted a test for checking an ARCH model applicability to reallife measurements of available bandwidth.

Real-life measurements of an available bandwidth time-series were first preprocessed (as it was described in previous section) and then statistical test checking data for heterascidascicity [20] has been performed. The Null hypothesis for such a test was an assumption that data follows ARCH model whereas an alternative hypothesis was an assumption that ARCH model is not applicable. After trend removal and inversion for real available bandwidth measurements taken across Internet paths we obtained high p-values which clearly suggest to accept the Null hypothesis. In particular, for the path MSU-Australia the reported p-value was 0.237, which does not allow us to reject Null hypothesis at a reasonable significance level.

£

We conclude that we have sufficient preliminary information to apply the ARCH model to an available bandwidth time-series and check how well it predicts real network paths in the Internet.

In order to apply a parametric model to "crossing probability" estimation, one needs to ensure that the model parameters can be estimated at run-time sufficiently fast. Then the model needs to be mathematically analyzed in order to find an efficient method to compute the "crossing probability" based on estimated model parameters. Also one needs to be able to build confidence intervals around the estimated probability in order to be aware of the model's accuracy. Then the model needs to be empirically verified. We therefore provide below our work on unification of results in $ARCH^2(1)$ parameters estimation and "crossing probability" computation together with the scheme of confidence intervals construction.

ARCH(1) model has been studied extensively with regard to the description of time-series in finance. The basic work in ARCH parameter estimation has been done by Weiss [59], who proposed to estimate ARCH parameters by maximizing the



Figure 3.10: Fisher Information: standard deviation of $\hat{\beta_0}$



Figure 3.11: Fisher Information: standard deviation of $\hat{\beta_1}$



Figure 3.12: Fisher Information: correlation between $\hat{\beta}_1$ and $\hat{\beta}_0$

Quasi Log Likelihood Function evaluated as:

$$L(\beta_0,\beta_1) = -\sum_{i=1}^{n-1} [\log(\beta_0 + \beta_1 X_i^2) + \frac{X_{i+1}^2}{\beta_0 + \beta_1 X_i^2}]$$

Here n is the number of observations in the window that is used for parameter estimation. The parameter n is very important and we will talk separately about its proper adjustment. The advantage of the QMLE (Quasi Maximum Likelihood) estimator of $[\beta_0, \beta_1]$ is that it is asymptotically unbiased. Also it is known [51] that $[\hat{\beta}_0, \hat{\beta}_1]$ is normally distributed with mean $[\beta_{0true}, \beta_{1true}]$ and covariance matrix called the Fisher Information. The normality of estimators gives background for computing a confidence interval of any level around the estimated values of the model parameters. The biggest challenge for the QMLE estimator is, however, that it is not given in a closed form. Therefore finding estimators for $[\beta_0, \beta_1]$ reduces to a maximization of a function of two variables. The function maximization problem has been intensively studied and many efficient algorithms exist. However their performance significantly depends on the initial point from which they have to converge and sometimes this cannot be done sufficiently fast for an on-line implementation. Bose, et al. [9] proposed another estimator for $\vec{\beta} = [\beta_0, \beta_1]'$.

$$\vec{\hat{\beta}} = \frac{\left[\sum_{i=1}^{n} \{\frac{S_{i-1}X_{i}^{2}}{\sigma_{i-1}^{4}(\hat{\beta}pr)}\}\right]}{\left[\sum_{i=1}^{n} \{\frac{S_{i-1}S_{i-1}'}{\sigma_{i-1}^{4}(\hat{\beta}pr)}\}\right]}$$

where $S_{i-1} = [1, X_{i-1}^2]'$, $\sigma_{i-1}^2(\beta) = S_{i-1}\beta$, and $\hat{\beta} = (\vec{S}'\vec{S})^{-1}\vec{S}'\vec{Y}$. Although Bose, et al. [9] give estimators of the $[\beta_0, \beta_1]$ in a closed form, we run a verification procedure in order to check how the precision and computation time of this approach may be compared with QMLE. Our simulation included modeling of 8,000,000 different ARCH(1) time-series with varying known $[\beta_{0true}, \beta_{1true}]$. We came to a conclusion that for many tested values of $[\beta_{0true}, \beta_{1true}]$, the estimator proposed by Bose, et al. [9] is biased whereas the QMLE estimator is unbiased for all tested values of $[\beta_{0true}, \beta_{1true}]$. Therefore we decided to join two approaches in the following way. We first find preliminary estimators of $[\beta_0, \beta_1]$ by the Bose, et al. [9] approach that takes less than a second. Then we put this preliminary estimator as an initial point of convergence for a QMLE algorithm. In this way the QMLE algorithm takes an average of approximately 3 seconds to converge.

Since estimators of $[\beta_0, \beta_1]$ depend on past observations and therefore have a random nature, we also provide a way for quick computation at run-time of the confidence region of any level (by default 95% CR). The 95% confidence region for $[\beta_0, \beta_1]$ is defined as a random region in \mathbb{R}^2 that contains the point $[\beta_0, \beta_1]$ with probability ≥ 0.95 . This region for the QMLE estimator will have an elliptical form defined by the equation obtained by Hotelling [26]

$$(\vec{\beta} - \vec{\beta})'\hat{\Sigma}^{-1}(\vec{\beta} - \vec{\beta})(n-2)/2 = F_{2,n-2}(0.95)$$

where $F_{2,n-2}(0.95)$ is 95% quantile of F-distribution with 2 and n-2 degrees of freedom, and $\hat{\Sigma}$ is estimated covariance matrix. In order to compute a confidence

region quickly at run-time we tabulated $\hat{\Sigma}$ for a fixed n = 720 and a grid of β_0 and β_1 . Figures 3.10, 3.11, 3.12 plot standard errors for $\hat{\beta}_0$ and $\hat{\beta}_1$ as well as the correlation between $\hat{\beta}_0$ and $\hat{\beta}_1$ as functions of true values of β_0 and β_1 . Standard errors and correlation were calculated by simulating 1000 independent $ARCH^2(1)$ time-series for each fixed pair of β_0, β_1 and finding the difference between estimated and true values of the aforementioned vector. β_0 and β_1 then were varied within the region (0.5, 3.5) and (0, 0.6), respectively. This region covers a reasonable region of values that β_0 and β_1 might take while describing the available bandwidth behavior within the range of 200 Mbps. It is known that the estimated covariance matrix is proportional to n^{-1} . Therefore, when having available tables visualized in figures 3.10, 3.11, 3.12 for n = 720, one can recompute $\hat{\Sigma}$ for an arbitrary n in less than a second.

3.2.3 Results from extremal values theory

Below we show some mathematical results that allow us to compute the "crossing probability" for any specified threshold M, number of observations N, and estimated model parameters β_0, β_1 . These results also help us to draw a 95% confidence interval for a "crossing probability" by considering the fact that the "crossing probability" is a monotonically increasing function of both β_0 and β_1 and finding "crossing probabilities" for sufficiently many points on the 95% ellipse in the β_0, β_1 space. Embrechts, et al. [19] summarize recent work in extreme values providing formulas for finding "crossing probability" for various time-series models. In particular for the $ARCH^2(1)$ model, Embrechts, et al. have

$$\lim_{N \to \infty} P(N^{-\frac{1}{\kappa}} \max\{X_1, ..., X_N\} \le M_{in}) = \exp^{-C_1 M_{in}^{-\kappa}}$$

that allows us to rewrite this result for "crossing probability" ${\cal P}$ and large N as follows

$$P = 1 - \exp^{-C_1 M_{in}^{-\kappa} N}$$



Figure 3.13: "Crossing" probability for a threshold M=25 Mbps and a number of observations ${\rm N}=50$

where κ is a unique root of the equation $\frac{(2\beta_1)^u}{\sqrt{\pi}}\Gamma(u+\frac{1}{2})=1$. M_{in} stands for a threshold for an *inverted* available bandwidth time-series. It can be related to a threshold M by a simple formula $M_{in} = constant - M$, where the *constant* is the "base-level" around which the original available bandwidth time-series was inverted (or in other words "mirrored"). Although Embrechts et al. [19] give an expression for C_1 , it is more efficient to estimate it empirically off-line for different β_0 and β_1 . For that purpose we simulated 800,000 observations for each pair of β_0 and β_1 coming from a grid with range (0.5, 3.5) and (0, 0.6), respectively. Figure 3.13 shows this probability for a fixed N = 50 and M = 25 Mbps. Then, based on the probability expression above, this tabulated result can be quickly generalized on-line to arbitrary N and M.

Calculating the estimated probability together with its confidence interval gives

an opportunity to judge the size of the window n used for prediction of β_0 and β_1 coefficients. On one hand we want n to be as small as possible because we assume that the model parameters remain unchanged for the time-period related to n. On the other hand, with larger n, accuracy of the prediction increases with corresponding shrinking of the 95% confidence region. In our experiments, for example, we put the following condition to the 95% confidence interval for the probability: the difference between the 95% upper bound and the predicted value should not exceed 0.2 for an arbitrary predicted probability value from the range [0;0.5]. Then for $N \in \{1, ..., 100\}$ and $M \in [60; 90]$, the lower boundary for n will be evaluated to 120. Note that the dependence on N is natural: with increasing of the period of time for which the prediction is made (duration of real-time task execution), the period of time used for prediction also needs to be correspondingly increased.

3.3 Experimental evaluation

In this section we describe our method for testing the accuracy of our approach to available bandwidth modeling. In previous sections we introduced the "crossing probability" as a characteristic of a network path with respect to available bandwidth and the given real-time task that needs to be remotely executed. "Crossing probability" can be viewed as a characteristic of a current state of a network path expressed in the form of a prediction of the bandwidth behavior for the time-period equivalent to the next N observations of the available bandwidth time-series. The "Crossing probability" approach tells that if a network path resides in the present state (evaluated with the help of the past n observations) for the period of the next N observations, then the probability of crossing of a predefined threshold can be quickly evaluated by a proposed parametric model.

The final target of the modeling is a prediction of a probability. The only way, therefore, to empirically verify the model is to apply the ergodic theorem, i.e., to have B groups with N observations in each and then count the number L of groups where the maximum over N observations within the group crossed a pre-defined

N\Mbps	60	62	64	66	68	70	72	74	76	78	80	82	84	86	88	90
15	0.09	0.09	0.08	0.07	0.07	0.07	0.07	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.04
20	0.12	0.12	0.10	0.09	0.09	0.09	0.09	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.05
25	0.14	0.14	0.13	0.11	0.11	0.11	0.11	0.09	0.09	0.09	0.09	0.09	0.09	0.08	0.08	0.06
30	0.18	0.18	0.16	0.14	0.14	0.14	0.14	0.11	0.11	0.11	0.11	0.11	0.11	0.10	0.10	0.07
35	0.20	0.20	0.18	0.16	0.16	0.16	0.16	0.13	0.13	0.13	0.13	0.13	0.13	0.11	0.11	0.08
40	0.22	0.22	0.20	0.18	0.18	0.18	0.18	0.15	0.15	0.15	0.15	0.15	0.15	0.13	0.13	0.10
45	0.27	0.27	0.24	0.22	0.22	0.22	0.22	0.17	0.17	0.17	0.17	0.17	0.17	0.15	0.15	0.11
50	0.28	0.28	0.25	0.23	0.23	0.23	0.23	0.19	0.19	0.19	0.19	0.19	0.19	0.17	0.17	0.12
55	0.32	0.32	0.29	0.26	0.26	0.26	0.26	0.21	0.21	0.21	0.21	0.21	0.21	0.19	0.19	0.14
60	0.32	0.32	0.30	0.26	0.26	0.26	0.26	0.24	0.24	0.24	0.24	0.24	0.24	0.21	0.21	0.15
65	0.33	0.33	0.29	0.26	0.26	0.26	0.26	0.22	0.22	0.22	0.22	0.22	0.22	0.19	0.19	0.14
70	0.34	0.34	0.30	0.26	0.26	0.26	0.26	0.24	0.24	0.24	0.24	0.24	0.24	0.20	0.20	0.14
75	0.35	0.35	0.31	0.28	0.28	0.28	0.28	0.25	0.25	0.25	0.25	0.25	0.25	0.22	0.22	0.16
80	0.34	0.34	0.32	0.28	0.28	0.28	0.28	0.23	0.23	0.23	0.23	0.23	0.23	0.19	0.19	0.16
85	0.40	0.40	0.35	0.31	0.31	0.31	0.31	0.27	0.27	0.27	0.27	0.27	0.27	0.22	0.22	0.18
90	0.39	0.39	0.32	0.28	0.28	0.28	0.28	0.24	0.24	0.24	0.24	0.24	0.24	0.19	0.19	0.15
95	0.39	0.39	0.35	0.29	0.29	0.29	0.29	0.25	0.25	0.25	0.25	0.25	0.25	0.21	0.21	0.17
100	0.41	0.41	0.35	0.29	0.29	0.29	0.29	0.25	0.25	0.25	0.25	0.25	0.25	0.21	0.21	0.17

Table 3.1: Empirically estimated "Crossing" probability for a path MSU-Berkeley

threshold. Then the ratio L/B can be considered as an empirically computed value of a "crossing probability". We considered as our design accuracy the following criteria: within the frames of the model, the difference between the predicted probability and its 95% upper confidence limit should not exceed 0.2 at any circumstances. Our simulations of $ARCH^2(1)$ series show that such "within-the-model" accuracy on a probability scale for $N \leq 90$ dictates the lower bound on number of observations needed for model parameters estimation as n = 120. Higher accuracy leads to higher values of a lower bound for n.

In real-life, however, it is impossible to achieve such an ideal design of the experiment as it was described in the previous paragraph because of the following factors. First, we obtain B groups of measurements in a real scenario that belong to the same time-series (instead of having multiple independently generated series in the simulation scenario) and therefore they are somehow dependent. Second, even if we assume that the model parameters remain the same over the estimation window of length n (that is reasonable if n correspond to 2 hours) it is too naive to assume that these parameters will remain the same over the whole time-period needed for collection of B groups of size N. Recall that present available bandwidth measurement tools allow to make consistently stable measurements at a rate 1 sample per minute. Therefore, B = 100 reasonably separated groups of N = 100 observations in a group correspond to time needed for collection of 20,000 consecutive observations. With presently available sampling rate this time is evaluated to 20,000 minutes or 2 weeks of consistent measurements.

In order to address the above discussed challenges we propose the following scheme for estimating efficiency of the upper bound on "crossing probability". We consider many network paths aiming at those that have visually the same daily behavior range. In this section we show an experimental evaluation of one such path between nodes in MSU and Berkeley that are 20 hops apart. These nodes were chosen because their hardware parameters and stable working allowed us to take 15,000 continuous observations with the help of a Pathload measurement tool. For the MSU-Berkeley network path we chose n = 120 and for a fixed N = 50 monitored how estimated 95% upper bound for a "crossing probability" evolved over the period of one day. Therefore we separated our data-set into a training-validation part that constituted 24 hours or 1440 observations and testing samples. For the first day we evaluated the width of the strip where the estimated probability resided and then computed the empirical probability for the rest of the data. By doing this we computed the "crossing probability" averaged over the period of 1350 observations. Given that the behavior of the path did not have any significant trend or oscillations over all measurement periods, we were expecting an average probability estimated by an ergodic theorem to reside in a strip defined by a first day behavior pattern. Shown in the table 3.1 are estimated empirically values of the "crossing probability" for different N and threshold M varying within the range 15-100 observations and 60-90 Mbps, respectively. As was expected, it turned out that for all thresholds Mand sizes N empirically computed probability resides below the estimated bound. Table 3.2 shows the difference between the empirically computed probability and the estimated upper bound, whereas figure 3.14 shows two surfaces where the upper surface represents an upper bound estimated by the model and the lower surface represents an empirically computed probability. Figure 3.14 shows, in particular,

N\Mbps	60	62	64	66	68	70	72	74	76	78	80	82	84	86	88	90
15	0.06	0.06	0.06	0.06	0.06	0.05	0.04	0.05	0.05	0.04	0.04	0.03	0.03	0.03	0.03	0.04
20	0.08	0.07	0.07	0.08	0.07	0.06	0.06	0.07	0.06	0.05	0.05	0.04	0.04	0.04	0.04	0.05
25	0.10	0.09	0.09	0.10	0.09	0.08	0.07	0.08	0.07	0.06	0.06	0.05	0.05	0.05	0.04	0.06
30	0.11	0.09	0.09	0.10	0.09	0.08	0.07	0.09	0.08	0.07	0.06	0.05	0.05	0.05	0.05	0.07
35	0.13	0.11	0.11	0.11	0.10	0.09	0.08	0.10	0.09	0.08	0.07	0.06	0.05	0.06	0.05	0.08
40	0.14	0.12	0.12	0.13	0.11	0.10	0.08	0.11	0.10	0.09	0.08	0.07	0.06	0.07	0.06	0.09
45	0.13	0.11	0.12	0.12	0.11	0.09	0.08	0.11	0.10	0.09	0.08	0.07	0.06	0.07	0.06	0.09
50	0.14	0.12	0.14	0.14	0.12	0.11	0.09	0.12	0.11	0.10	0.08	0.07	0.06	0.07	0.06	0.10
55	0.14	0.12	0.13	0.14	0.12	0.10	0.09	0.12	0.11	0.10	0.09	0.07	0.06	0.08	0.07	0.10
60	0.16	0.14	0.15	0.16	0.14	0.13	0.11	0.12	0.11	0.09	0.08	0.07	0.06	0.08	0.07	0.11
65	0.18	0.16	0.19	0.19	0.18	0.16	0.14	0.16	0.14	0.13	0.12	0.10	0.09	0.11	0.10	0.14
70	0.20	0.18	0.19	0.21	0.19	0.17	0.16	0.17	0.15	0.14	0.12	0.11	0.10	0.12	0.11	0.16
75	0.21	0.19	0.21	0.22	0.20	0.18	0.16	0.18	0.16	0.14	0.13	0.11	0.10	0.12	0.11	0.16
80	0.25	0.22	0.22	0.25	0.23	0.21	0.19	0.21	0.20	0.18	0.16	0.15	0.14	0.16	0.15	0.17
85	0.21	0.19	0.22	0.24	0.22	0.20	0.18	0.20	0.18	0.16	0.15	0.13	0.12	0.15	0.14	0.17
90	0.24	0.22	0.27	0.29	0.26	0.24	0.22	0.25	0.23	0.21	0.20	0.18	0.17	0.20	0.19	0.21
95	0.26	0.24	0.26	0.29	0.27	0.25	0.23	0.25	0.24	0.22	0.20	0.19	0.17	0.20	0.19	0.21
100	0.26	0.24	0.28	0.31	0.29	0.27	0.25	0.27	0.25	0.24	0.22	0.20	0.19	0.22	0.20	0.23

Table 3.2: Difference between model-based estimated upper bound and empirically estimated "Crossing" probability for a path MSU-Berkeley

that two surfaces are nearly parallel to each other, i.e., that the model reflects scales of both N and M in reasonable agreement with what was empirically observed. Also figure 3.14 shows that estimation becomes less accurate when N approaches to n, which is understandable.

We conclude that for a given channel the model produced verifiable and a reasonably accurate prediction. We also note that the model is supposed to do its job for any pair of network paths and real-time tasks for which parameters β_0 and β_1 do not fall out of pre-defined regions of variability and also significant trend or routes oscillations do not occur for the period of time required for a real-time task execution.

3.4 Conclusion

In this chapter we presented a "probabilistic" parametric approach to a description of the available bandwidth behavior. We showed that after trend subtraction and inversion the available bandwidth can be described by the $ARCH^2(1)$ parametric model. We have intensively studied and unified known results in $ARCH^2(1)$, tab-



Figure 3.14: Empirically estimated 95% upper bound over empirically computed "Crossing" probability for a path MSU-Berkeley

ulated relevant characteristics and provided an efficient method for computation of a "crossing probability". Then we tested the accuracy of our model on a channel with relatively stable long-term behavior and proved that the approach has a certain range of applicability. The most significant shortcoming of our experiments is the sampling rate of Pathload and its unstable output over significant number of network paths. The best Pathload's sampling rate during our experiments was 1 observation per minute. Recall that τ should be of the order of receiver's buffer time in order to allow observations to be relevant with respect to a particular real-time task execution. Also as our simulations show we need at least 120 observations for accurate enough estimation of model's parameters. At the same time the smaller is the size of the history window the more realistic is the assumption that behavior of available bandwidth time-series is stationary over history time-frame used for prediction. Taking into account Pathload's sampling rate we therefore were restricted with paths with relatively stationary behavior for the period of at least 2 hours.

If in the future there are better available bandwidth measurement tools that are

able to sample at higher rates, then the range of applicability of the parametric approach will significantly increase. The next chapter introduces work directed towards increasing the accuracy of probe-rate based available bandwidth measurement tools and consequently available bandwidth sampling rate.

Chapter 4

Network One Way Delay Trend Detection Mechanism

In this chapter we demonstrate an approach that improves a whole range of "proberate" AB measurement tools that send sequences of measurement packets (called "trains") across the network path and capture a positive trend in their travel times or One Way Delays (OWD). In particular, we show an algorithm for efficient OWD trend detection and compare it to widely used OWD trend detection tests.

4.1 Main challenges of IC/CS elimination and trend detection in Pathload

Jain and Dovrolis have shown in [30] that Pathload demonstrates reasonably good performance on paths that have tight and narrow link in the middle of the path. For such cases Pathload's error was around 5% of available bandwidth value when this value belongs to the range 70-80Mbps and around 10% of available bandwidth value when the value is smaller than 10Mbps.

In such cases, however, the slow link, which lies in the middle of the path, stretches packets, whereas fast links contract packets. In other words, suppose that packets are sent with a certain fixed delay. When they pass from a faster link to a slower link the delay between packet arrivals becomes smaller or is eliminated. In contrast, when packets pass from a slow link to a fast link, then the delay between packets increases. Therefore by having a slow link in the middle of the path the authors are guaranteed that there will be a significant delay between packet arrivals at the end which first, makes trend detection easier and second, makes measurements less vulnerable to IC/CS effect.

Another "friendly" factor in the experimental setup was good match of a length of a packet train to the path across which train was sent. More precisely, when a train was sent at rates higher than available at that time bandwidth, routers in the middle of the path were experiencing instantaneous increase of packet queues but at the same time were not dropping any significant portion of packets. That experimental design allows one to avoid addressing the problem of packet loss. It is unreasonable, however, to assume that an algorithm of an ideal match of length of a train to queue length on intermediate routers can be developed.

Ubik, et al. [58] performed an extensive analysis of Pathloads operation for various networks, values of available bandwidth and values of a cross-traffic. Tables shown in their work clearly show that performance of a Pathload is far from ideal.

We set up an experiment where queues on routers/switches were small enough for dropping significant portion of packets sent by a Pathload at rate higher than the available bandwidth. In some of our experiments Pathload's PCT/PDT test falsely reported trend for approximately 25% of analyzed trains sent at the rate lower than available bandwidth while not reporting a trend for 14% of trains sent at the rate higher than available at that time bandwidth.

Even when experiencing packet losses Pathload still can report available bandwidth from a correct range despite the high error rate of trend detection at the level of a particular train. Pathload creators recognized the challenge of packet losses and introduced a very artificial way of partially avoiding this problem. They have set up two levels of packet losses. If the percentage of losses in at least one train in a set of trains sent at some fixed rate is higher than HIGH LOSS RATE than the rate is automatically believed to be higher than the available bandwidth. If losses in more than 50% of trains sent at fixed rate exceed MEDIUM LOSS RATE then the rate is believed to be higher than the available bandwidth as well. These two levels of percentages are initially set in a Pathload as 15% and 7% respectively.

The artificial way described above of dealing with packet losses has its own shortcomings. First, HIGH and MEDIUM loss rate parameters have to be nicely tuned. Second, packets may be lost because of a processor utilization at receiving computer. As a consequence of implemented in Pathload packet-loss dealing mechanism, Pathload fails to report correct bandwidth even when processor utilization at receiving machine reaches 5%. This can be considered already as a significant limitation of a tool usage especially for a long-lasting measurements given that a receiver machine is running under any of wide-spread non-real-time operating system.

The way how Pathload deals with IC/CS effect is not perfect as well. The time that the operating system needs for delivering packets to upper layer cannot be precisely estimated. Pathload's PCT/PDT metrics also leave a place for improvement. Figures 4.1 and 4.2 illustrate a typical example of a train misclassified by the PCT/PDT test. The train was sent at the rate which was 10Mbps higher than available at that time bandwidth. On the left hand-side are plotted one-way delays of 100 packets belonging to the train. We see that the plot has parts with negative trend and these parts are strictly "linear", which is a direct consequence of IC/CSeffect. We can think of IC/CS as of an additional buffer at which packets spend certain time and which is periodically emptied. The last packet in the group before the "buffer" is emptied waits the least time whereas the one that came straight after previous emptying waits the longest period of time. Also, it is known that all packets were released by the sender with some fixed unchanged delay. It follows therefore that all packets related to strictly linear decreasing part of the plot belong to the same group. Now consider a subset of packets constituted by the last packet taken out of each strictly-linear decreasing part. On the left hand-side plot one can clearly observe that aforementioned set constitutes subsequence with clearly-revealed positive trend.

In figure 4.2 is depicted a subsequence of points chosen by a Pathload IC/CS elimination procedure. One can see on that plot that at least some points of the



Figure 4.1: Shortcomings of IC/CS elimination procedure in Pathload: before elimination



Figure 4.2: Shortcomings of IC/CS elimination procedure in Pathload: after elimination \mathbf{P}

chosen subsequence do not characterize packets that are least delayed by the IC/CS effect. Moreover the trend picture that is clearly observed even by a naked eye on figure 4.1 is not so evident on figure 4.2. The information about the index of packets finally chosen by IC/CS elimination procedure is also not considered by a Pathload.

Finally, let us apply the PCT test to what is left on the right plot. There are left only seven points and only half of those are bigger than their direct predecessors. As a final result for this real-life typical example Pathload received a value of 0.43 for PCT metric and 0.17 for PDT metric which are deep in the region classifying train as the one without trend.

The conclusion is that Pathload throws away significant part of a measurement information at the IC/CS elimination stage as well as when dividing the rest of data into groups and discarding everything except medians in each group. Also there is an absence of statistical motivation of the efficiency of the PCT/PDT test as well as absence of reasoning why the chosen thresholds for PCT and PDT metrics should do a reasonably good job for an arbitrarily taken network path. The example which we plotted at figures 4.1 and 4.1 was very frequent in data that we collected at our network segment and therefore explains significant inaccuracy of the PCT/PDT test when the experimental setup is not as friendly as it was with a highly-revealed tight and narrow link in the middle of the path.

In next section we explain our way to fix deficiencies of a Pathload listed above and later give comparative experimental results demonstrating effectiveness of our approach.

4.2 IC/CS elimination procedure and t-test

In this section we give a detailed analysis of our way of dealing with interrupt coalescence / context switch effect as well as describe the metric that we choose for efficient trend detection.

The idea of proper IC/CS elimination is very simple. We do not consider the time which the system needs to bring packets from the network adapter to memory since it cannot be estimated accurately. Instead we detect cases when packets constitute a strictly linear decreasing trend. If difference in one-way delays of a pair of packets with neighboring indices is negative and is very close to the same difference of neighboring packet pairs then all such packets are assigned to the same group and only the last packet from that group is considered for the further analysis. In other words, all packets with one way delays constituting a clearly revealed straight line with a negative slope are discarded except the very last one. We require such a group to have at least three packets in order to discard all packets in the group except the very last packet. At this stage it is important that we keep information about indices of packets left for a further statistical analysis. This way of dealing with the IC/CS effect is simpler and more effective than the one proposed in Pathload. We do not assume that the very last packet in the group is processed immediately after it arrives. We only say that it waits the smallest possible time for an interrupt among all packets in its group. Also we assume that the time which the last packet in a group waits for an interrupt is normally distributed and therefore methods of linear regression analysis can be further applied to chosen subsequence of packets for trend detection.

The basic idea of a t-test in linear regression analysis is simple. In our case so-called "response variable" is one way delay of a packet whereas "predictor" is the packet index that incorporates time flow. The main question to which t-test for a slope-coefficient in linear regression is supposed to answer can be stated as follows. Do the train data give sufficient evidence to believe that one-way delay of a packet depends on its index? Linear regression procedure plots a best-fit line across data points corresponding to one-way packet delays in a way that minimizes squared distances from data points to the plotted line. Slope coefficient of the line can be found according to the formula

$$\hat{\beta}_1 = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

In this formula y_i represents one way delay of a packet, x_i represents packet's

index in the train whereas n is total number of finally left packets belonging to the train. Since $\hat{\beta}_1$ depends on random measurement errors it also has a random nature. It is known [51] that under linear regression model assumptions coefficient $\hat{\beta}_1$ follows t-distribution with n-1 degrees of freedom centered around true value of this coefficient. The null hypothesis in our t-test is an assumption that the train does not contain any OWD trend or in other words the slope coefficient of a regression line is zero. Therefore the distribution of a random quantity $\hat{\beta}_1$ under null hypothesis is t-distribution centered around zero with n-1 degrees of freedom.

If we reject the null hypothesis (i.e., decide that there is a trend in a train) then the probability of an error can be calculated as an area under the centered around zero t-distribution density, which remains on the right hand-side of the rejection threshold. The error is an error of classifying the train with no-trend as a train with a trend and is called a type I error. The probability of a type I error naturally depends on the placement of a rejection threshold. The smallest value of type I error probability given that we still reject the null hypothesis can be achieved if we place the rejection threshold straight to the value of $\hat{\beta}_1$. The area of t-distribution density function which is left on the right hand-side of the observed value of $\hat{\beta}_1$ is called p-value of the test.

We decided to choose the obtained p-value of the test as our final quantity for judging trend presence instead the value of $\hat{\beta}_1$ itself. As it follows from previous paragraphs, the low p-value values should be considered as an indicator of a trend presence, whereas high values should be considered in favor of accepting of the null hypothesis. We choose the threshold for a p-value to be equal to 0.01. The threshold set in the p-value space has a statistical motivation behind it. It means that we want to have the misclassification probability for trains which really do not have OWD trend (i.e., trains which were sent at the rate lower than available at that time bandwidth) to be approximately equal to 0.01.

We have to accept that there is not much that one can say about type II error probability (probability of classifying a train with trend as a train without trend) unless it is estimated by an experiment for a fixed value of a decision threshold.



Figure 4.3: OWD pattern at the presence of packet loss. Trains were sent at a rate 40 Mbps higher than available at that time bandwidth.

However, we now have at least some statistical reasoning under choosing a particular level of a p-value as a decision threshold instead of choosing rejection regions for PCT and PDT metrics on absolutely empirical basis.

A consideration of how much of information is still available after IC/CS elimination for a further analysis can be viewed as one more argument towards choosing p-value as a final statistic of interest. With a decreasing of number of points remaining after IC/CS elimination, and corresponding decrease of degrees of freedom n t-distribution becomes more and more "stretched-out". This means that for a fixed value of $\hat{\beta}_1$ the area on its right-hand side will be larger and larger with decreasing of amount of available information (i.e., available data-points or degrees of freedom). Correspondingly, in order to have a sufficient basis for rejecting the null hypothesis we will need a much higher value of $\hat{\beta}_1$ compared to the case when more data points are available for the analysis.

Another advantage of our scheme of IC/CS elimination and trend detection is that it deals with packet losses better. The influence of packet loss to trend detection is very significant. When routers in the path discard significant number of packets they may "recover" from "instantaneous" overflowing of their queues. Therefore trend detection becomes available only when considering each piece of contiguous



Figure 4.4: OWD pattern at the presence of packet loss. Trains were sent at a rate 30 Mbps higher than available at that time bandwidth.



Figure 4.5: OWD pattern at the presence of packet loss. Trains were sent at a rate 20 Mbps higher than available at that time bandwidth.

chunks of received packets separately. The effect of packet-loss is well-illustrated by figures 4.3, 4.4, 4.5. We can clearly see there that even after elimination of IC/CS parts the whole picture does not give an impression that there is a trend whereas every contiguously received chunk does give such an impression. In such cases Pathload reports a high percentage of "uncleared" trains because Pathload throws away too much information. Pathload frequently does not have any reserve left for further splitting a train into a number of sub-trains. Our scheme can afford such splitting and therefore we report much lower "unclear" trains, which results in significant reducing of number of trains that are required to be sent in order to achieve the desired detection accuracy. We report train as an "unclear" only when we have finally less than 4 data-points left. If we have a number of reported p-values related to a number of sub-trains then we decide about the trend in the whole train based on the majority-vote rule in sub-trains. Note that if the IC/CSpacket group is directly followed by lost packets then we throw away all that group of packets including the last one. With decreasing a number of trains required to be sent, measurement time also is significantly reduced. This gives our approach a higher degree of flexibility in available-bandwidth related planning, which is of a high importance for present real-time network applications.

In the next section we describe our experimental setup and give comparison of our scheme performance versus the Pathload.

4.3 Experimental evaluation

We experimented within our laboratory network, which consists of three 100Mbps capacity segments. The sender and receiver were placed in the first and third segments, while the path from the first to the third segment was lying through the second segment. We wanted to have both Pathload's PCT/PDT test and our approach to have absolutely equal conditions. Therefore we modified an original version of a Pathload by adding a number of printouts that were outputting to the separate file the OWD times measured by Pathload. Also for each train we outputted values of

the PCT/PDT metric together with Pathload's final decision about the train. Then we processed the same train data by our code, which eliminated the IC/CS effect, conducted t-test, outputted p-values, and finally made a decision about the trend presence.

During the first part of the experiment it was known with a high degree of confidence that the background traffic in all three segments is very low (around 1 Mbps) and does not exceed 2-3Mbps at any period of time of the length of 200 milliseconds. For that part of the experiments we knew therefore that all trains sent at the rate lower than 95Mbps should be reported as trains without trend. Also we knew that all three links capacities were 100Mbps and therefore all trains that were sent at rates 98 Mbps and higher should be reported as trains with a trend. This experimental setup did not contain any tight/narrow link in the middle of the path. We run Pathload continuously for the period of approximately 7 minutes and analyzed in total 4650 trains. Then we imposed a slight utilization to the processor of a receiver and repeated the experiment. Table 4.1 shows how the performance of the PCT/PDT test deployed in a Pathload is compared with the performance of our approach when no processor utilization was imposed at the measurement machine responsible for receiving packets. Table 4.2 shows detection performance comparison when processor utilization was 5-7%. Tables are structured as follows. In the second bold column are shown type II errors for PCT/PDT test and t-test, respectively. In particular, there is shown the portion of trains that were incorrectly classified as trains without a trend when the trend was really present. Note that this portion has been computed "conditionally" i.e., as a portion out of total number of trains that were sent in the specified rate-range and were not assigned to "uncleared" train set. In the third bold column is shown type I error for multiple rate ranges. Let us remember that we have set a decision threshold for p-value equal to 0.01. We see in all tables that experimentally evaluated type I error is not that far from "theoretically-predicted" value 0.01, which can be considered to some extent as a verification of model's applicability. "Unclear trains" column is followed by a column showing the number of trains that the t-test based tool needs to send in order to achieve accuracy of PCT/PDT test sending 6 trains at a given rate. Naturally, if a number in this column is greater than 6 then PCT/PDT test demonstrates better performance for a corresponding rate and vice-versa otherwise.

For the case when there was no any extra processor utilization, t-test performed better for all trains sent at the rate higher than 98Mbps as well in the range 15-55Mbps. Although we have slightly higher error rate for a range 55-95Mbps than PCT/PDT test has, still both approaches need to send three trains for achieving 99% accuracy. Difference in performances for the mentioned range therefore can be considered as not significant whereas for other rates our trend detection mechanism significantly outperforms the PCT/PDT test.

Table 4.2 clearly shows that the PCT/PDT performs worse when operating even under very small on-going processor utilization. In particular, the PCT/PDT test fails to correctly classify trains sent at the rate-range 15-55Mbps. Its error approaches 0.5 and sometimes even exceeds this value, which makes the rate classification procedure either unacceptably long or not applicable at all. At the same time in the global scale Pathload was unable to report even once available bandwidth from a correct range when processor utilization was imposed. For example, the maximal reported value of available bandwidth out of twenty runs was 70Mbps when the channel was almost free and processor utilization was 3.5%. The responsibility of this failure is Pathload's method to deal with packet losses.

During the second part of our experiment it was known with a high degree of confidence that the total cross-traffic was always exceeding the level of 65Mbps. In other words, we were guaranteed that if trains that were sent at rate 35Mbps or higher are classified as those with trend, then this classification is correct. There was no processor utilization imposed to a measurement machine. The goal of the experiment was to check how efficient t-test and PCT/PDT test detect trend in trains that are sent at the rate lower than the capacity but higher than the available bandwidth of the network path.

Table 4.3 shows results of t-test and PCT/PDT test accuracy for trains sent at the rate higher than the available bandwidth. Instead of type I error in the third bolded

Rate at which trains were sent	Classified 95 Mbps (trend	d as 0- without d)	Classified than 98 (with tr	as more Mbps end)	Unclear	Clear trains out of 6 trains that t-test needs to clear trains thet t-test needs to clear trains train train train trains that t-test needs to clear trains train trains trains trains trains trains trains train trains train train train train train train train train train train train train train train train trains trai trains trai trai trains trai trains trai trai trai trains tra		Unclear trains		required to least 0.99 iracy	Total number of analyzed trains	
	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT test	PCT/PDT	t-test	
15-35			0.000	0.014	0.500	0.014	0.985	1.000	3	7	3	72
35-55			0.560	0.023	0.233	0.000	0.312	1.000	1	>100	3	564
55-75			0.002	0.040	0.093	0.000	1.000	0.999	9	3	3	420
75-95			0.002	0.054	0.098	0.006	1.000	0.997	11	3	3	1296
> 98	0.083	0.026			0.296	0.000	0.971	1.000	1	9	3	2226

Table 4.1: Comparison of PCT/PDT test and t-test performances on nearly empty network and zero receiver's processor utilization

Rate at which trains were sent	Classified 95 Mbps (trend	l as 0- without J)	Classified than 98 (with tr	as more Mbps end)	Unclear	trains	Trend det accuracy fleet cons out of 6 t	for a sisting trains	N of trains that t-test needs to achieve 6- train accuracy of	N of trains achieve at accu	N of trains required to achieve at least 0.99 accuracy	
	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT test	PCT/PDT	t-test	
15-35			0.441	0.010	0.190	0.055	0.508	1	1	>100	3	199
35-55			0.493	0.008	0.152	0.063	0.412	1	1	>100	3	1186
55-75			0.063	0.010	0.077	0.074	0.994	1	3	5	3	1070
75-95			0.033	0.011	0.053	0.069	0.999	1	5	4	3	1926
> 98	0.081	0.061			0.302	0.046	0.973	0.996	3	9	5	710

Table 4.2: Comparison of PCT/PDT test and t-test performances on nearly empty network and 5-7% receiver's processor utilization

Rate at which trains were sent	Classified than 35 (withou	d as less 5 Mbps t trend)	Classif more th Mbps (wit	ied as nan 35 th trend)	Unclear	trains	Trend de accuracy consisting trai	etection for a fleet g out of 6 ns	N of trains achieve at accu	Total number of analyzed trains	
	PCI/PDI	t-test	PCT/PDT	t-test	PCI/PDI	t-test	PCI/PDT	t-test	PC1/PD1	t-test	
35-55			0.005	0.922	0.453	0.000	0.000	0.992	>100	5	592
55-75			0.021	0.761	0.958	0.000	0.004	0.853	>100	17	426
75-95			0.325	0.958	0.394	0.000	0.157	0.999	>100	3	335
> 98	0.276	0.082			0.305	0.001	0.766	0.991	38	5	1601

Table 4.3: Comparison of PCT/PDT test and t-test performances on heavy loaded network and zero receiver's processor utilization

Difference between rate at which trains were sent and	Classification error		Unclear	r trains	N of trains that t-test needs to achieve 6- train accuracy	N of train to achiev 0.99 ac	Total number of analyzed	
available bandwidth			PCT/PDT	t-test	of PCT/PDT	PCT/PDT	t-test	
15-35	0.122	0.008	0.464	0.000	1	16	1	1023
35-55	0.054	0.000	0.299	0.002	1	7	1	711
55-75	0.020	0.003	0.344	0.001	1	6	1	751
75-95	0.443	0.292	0.397	0.001	1	>100	29	741
> 98	0.476	0.065	0.346	0.001	1	>100	5	664

Table 4.4: Comparison of PCT/PDT test and t-test performances on average loaded network (around 50Mbps $\pm 10Mbps$ cross-traffic) and zero receiver's processor utilization

column we show detection accuracy for a specified range of rates. Table 4.3 shows that the PCT/PDT test completely fails to detect trend when suffering packet losses and non-friendly experimental setup. We also note that despite this failure at the level of trains, Pathload still always reported values of the available bandwidth from the correct range because of the artificially implemented packet-loss mechanism. We believe, however, that an available bandwidth measurement tool can be significantly improved after excluding an artificial mechanism of dealing with packet losses and respectively relying solely on a train-level reported statistic.

In the third part of our experiment we artificially generated and monitored crosstraffic in order to obtain a clearer pattern of error percentage as a function of difference between the rate at which a train was sent and available bandwidth. Unfortunately we were unable to capture all traffic on the network and were able to monitor traffic only with a certain error. In our setup generated traffic was fluctuating around 50 Mbps. Since there was no 100% confidence in cross-traffic measurements and in order to be on a safe side we monitored trains which were sent at rates either at least 15Mbps higher or 15Mbps lower than tentatively reported available bandwidth. Based on the setup of our experiment we were able with sufficiently high degree of confidence to assume that these train rates were either higher or lower than the available bandwidth. Table 4.4 shows an output for this part of the test. Now we have only one column showing classification errors. For lines where difference between train rate and available bandwidth is negative, the error represents type I error. For lines with positive difference between rate and available bandwidth, the error represents type II error. Table 4.4 shows that our approach again outperforms PCT/PDT test for all shown ranges.

4.4 Conclusion

In this chapter we proposed a simple algorithm of dealing with the IC/CS effect together with a trend detection algorithm based on a simple linear regression model and p-value reported by t-test. We showed that our algorithm is more robust with respect to the IC/CS effect and packet losses. Our real-life experiments clearly show better performance of our approach in comparison to PCT/PDT test deployed first in Pathload and later in other probe-rate measurement tools [42, 46].

As a result, with our scheme either the accuracy of any probe-rate based available bandwidth measurement tool can be significantly improved or the measurement time can be reduced at least twice while preserving the same measurement accuracy.

However, despite the algorithm introduced in this chapter allows an increase of at least twice the available bandwidth sampling rate, the range of applicability of such a measurement tool for "crossing probability" estimation is still limited. We know that for most real-time applications, the averaging time τ is of the order of seconds rather than minutes.

In the next chapter we introduce a non-parametric model for "crossing probability" estimation, which is based directly on the OWD trend detection technique. Taking into account that the OWD at a given rate can be detected for the period of the order of tens of a second, our non-parametric model can predict crossing probability over the next second based on the behavioral history of last ten minutes. The next chapter gives details about our non-parametric approach.

Chapter 5

Non-Parametric Approach

In this chapter we provide a more general non-parametric approach for computation of a "crossing probability". In contrast to Chapter 3 that relies on measurements of the available bandwidth (AB) time-series, a non-parametric approach in this chapter relies directly on measurements of indicators showing whether the AB time-series is smaller or larger than a given value. Therefore a non-parametric approach is robust, allows faster computation of a crossing probability than the parametric approach in Chapter 3, and has a larger range of applicability since it does not rely on any parametric time-series model. Moreover, for computation of indicator values that compare available bandwidth against a specified value, we do not rely on the Pairwise Comparison Test / Pairwise Difference Test (PCT/PDT) metrics [29], but use our own way of trend detection introduced in Chapter 4.

We suggest to further optimize estimation of a "crossing probability" by "tuning" it to a specific network path and even the day of the week at which measurements are made in order to make estimation more efficient. First, different network paths have different time-frames at which available bandwidth may be treated as a stationary time-series. Second, different paths may "forget" about their history at different rates. In other words, even while assuming that the available bandwidth time-series is stationary, it might be applicable to assign larger weights to the recent past and smaller weights to the older history. The rate at which weights decay is network-path specific and needs to be adjusted as well. Finally, since the main goal of our research is a prediction of a "crossing probability," we introduce and compare methods for a verification of the accuracy of our prediction based on training and previously unseen testing data-sets obtained by real-life measurements across multiple Internet paths. Along with real-life experiments, we present theoretical results that show that our verification procedures are probabilistically legitimate.

5.1 Moving block bootstrap

Let us consider a stationary sequence $X_1, X_2, ..., X_n$ of observations of zeros and ones where each observation is related to the period of time of approximately 0.5 second. Let the value zero correspond to an event when the available bandwidth timeseries is larger than a given threshold whereas one is related to the case when the available bandwidth is smaller than the threshold. Each observation can be obtained with the help of the probe-rate approach. "One" is reported when there exists a positive trend in travel times of packets in a sequence sent at a given threshold rate. "Zero" is reported when there is no indication of such a trend. For obtaining observations of zeros and ones related to a specified threshold rate, sequences or trains of packets were sent exactly at that rate. Since each train was sent within the same period of time, the "averaging time" τ does not vary from train to train. Each observation is related to one train of packets and therefore τ can be defined as the time that one train of packets requires to be delivered from sender to receiver.

The problem of finding the crossing probability is reduced to finding a probability that among N consecutive observations of "zeroes" and "ones" there exists at least one "one". Note, that observations of zeros and ones are tightly related to cross-traffic existing in the path and therefore we have to assume that these observations depend on each other. More precisely, they form a stationary sequence that possesses long-range dependency and self-similar properties [34].

Based on aforesaid, the natural way of non-parametric estimation of a crossing probability is as follows. Observe n values $X_1, X_2, ..., X_n, X_i = \{0, 1\}, i \in$



Figure 5.1: Window of size N = 3 slides across the history data of size n = 12 for prediction of a crossing probability P over coming N = 3 observations

 $\{1, 2, ..., n\}$ of the time-series of interest. Then establish a window of size N sliding across the sequence of observations of size n by one observation at a time (see figure 5.1). At each cycle increase the counter C when among N observations in a sliding window there exists at least one "one". When the sliding window of size N gets to the end of the sequence of size n, divide the counter C by total number of cycles (n - N + 1) and consider this ratio as the estimator of the crossing probability for the next N observations. If n - N + 1 is large and at the same time the time-series is stationary during the period of time that corresponds to n observations, then the ergodicity theorem says that the aforementioned ratio is a legitimate estimator of the crossing probability (i.e., it converges to the crossing probability as n tends to infinity).

The estimator of the crossing probability C/(n-N+1) depends on observations of a random time-series and therefore is a random quantity. It is important to report not only the value of an estimator but also its 95% confidence interval. This is a random interval around the estimator such that the true value of the crossing probability falls into the interval with a likelihood of 95%.

The only presently known way of non-parametric estimation of a distribution of an estimator is the bootstrap technique. However, the bootstrap in its basic formulation helps to find a distribution of a random quantity T(data) that is a function of *independent* identically distributed random observations. Given that initially there are given k independent random observations, the bootstrap proposes to consider B groups of k observations in each, where each group is sampled with replacement. Therefore in some group there may exist multiple copies of a particular observation and some observations may not exist in a certain group as well. The bootstrap proposes to compute B values of quantity T, where each value is observed only at a particular group taken with replacement. Then distribution of a random quantity T can be estimated based on its B bootstrap values. In particular, the 95% confidence interval can be found with the help of 2.5% and 97.5% quantiles of obtained dataset of size B.

Our quantity of interest T can be defined as

$$T = \frac{\sum_{i=1}^{n-N+1} C_i}{n-N+1}$$

. In our case, however, n-N+1 observations of an indicator

$$C_i \equiv 1_{\max\{X_i, X_{i+1}, \dots, X_{i+N-1}\}=1}$$

depend on each other. Therefore application of a bootstrap in its original formulation is not legitimate. Liu and Singh [39] introduce the notion of Moving Block Bootstrap (MBB) re-sampling for the case of m-dependent random sequences. They believe however that their procedure works well on more general stationary sequences. Therefore we believe that it is legitimate to use the Moving Block Bootstrap procedure in our work.

We next describe the procedure. Let b be the size of the block of consecutive observations of time-series $X_i, X_{i+1}, ..., X_{i+b-1}$. If this block "moves" across the whole dataset of size n then n - b + 1 such blocks can be obtained. Note that length of the block b should be sufficiently large for capturing all correlation structure of the underlying time-series. More precisely, b should exceed the lag H at which the autocorrelation function $\rho(h)$ is negligibly small. Then $[\frac{n}{b}]$ blocks are chosen with replacement out of n - b + 1 existing blocks. These blocks are grouped together, one after another. Thus the new n "re-sampled" time-series observations are obtained. The first bootstrap value of a crossing probability estimator $\hat{p_1}^*$ is obtained based on these n observations. We repeat the above procedure 1000 times and obtain 1000 different bootstrap values of \hat{p}_B^* , B = 1, ..., 1000. Then an empirical distribution of a crossing probability estimator can be obtained from \hat{p}_B^* bootstrap values. In particular, the 95% confidence interval can be constructed by 2.5% and 97.5% quantiles of 1000 observations of \hat{p}_B^* .

The size of the block b as well as total number of available bootstrap blocks n - b + 1 are important parameters. With an increase of the period of time defined by the number N over which the prediction is made, the "history" period of time that is used for computation of a predictor has to be increased as well. On the other hand we tried to keep the "history" period as small as we could because the "stationarity" assumption of a "zero-one" time-series is more realistic for smaller "history" time-periods.

We believe that the following rule achieves the best tradeoff between keeping the size n of the history-window small and still having a representative statistic. Each block in the MBB approach has to have at least 20 non-overlapping windows of the length N and the "history" window should contain at least 10 non-overlapping blocks. Therefore the size n of the window that is used for prediction should be approximately 200 times more than the size N of the window over which prediction is made.

For example, if the rate threshold for which we predict the crossing probability is 80 Mbps, then one train of 120 packets can be sent for approximately 0.3 of a second. Suppose then that the real-time task takes 3 seconds to execute with $\tau = 0.3$ seconds, which means that N = 10. Then *n* should be of the order of 2000, which corresponds to time-period of 600 seconds or 10 minutes. Therefore we conclude that prediction of crossing probability for the period of 3 seconds can be made based on channels behavior over recent 10 minutes that is reasonable for assuming stationarity of the available bandwidth time-series.

5.2 D-statistic: model verification mechanism

The moving block bootstrap technique described above helps to report a nonparametric estimate of the probability (along with its confidence interval) that among on-coming N observations there is at least one "one". The problem now is to verify efficiently how accurate non-parametric prediction technique works for a real-life time-series of indicators. We predict the probability of the realization of a certain event and the natural way to verify accurateness of our prediction is to observe whether on real unseen data the predicted event has been realized. Then from many independent predictions and realizations/non-realizations of these predictions, one might make an inference about the accurateness of the prediction.

Multiple predictions, however need to be independent and therefore for the same network channel these predictions have to be significantly separated from each other in time scale. These predictions separated in time-scale may report different values of crossing probability because the stationarity assumption for available bandwidth time-series does not hold for large periods of time. Below we show the design of a statistically correct verification test that clearly states the Null and Alternative hypothesis. We also provide a theoretical background that proves legitimacy of our test.

Assume that we have the set of predictors \hat{p}_i , $i \in 1, ..., l$ of success probabilities of independent Bernoulli random variables along with actual observations of these variables Y_i . Let the Null hypothesis be an assumption that predictors \hat{p}_i match actual success probabilities of Y_i . Let the alternative hypothesis be an assumption that \hat{p}_i do not have any relationship with realizations Y_i . Now consider the following sum

$$D = \frac{\sum_{i=1}^{l} (Y_i - \hat{p}_i)}{\sqrt{\sum_{i=1}^{l} \hat{p}_i (1 - \hat{p}_i)}}$$

Below in the separate theoretical section we prove that despite Y_i s are not identically distributed random variables, the aforementioned sum under the Null hypothesis converges in the distribution to standard normal random variable (i.e., the Central


Figure 5.2: Probability density functions for a p-value conditioned by Null and Alternative hypothesis.

Limit Theorem does hold).

If D statistic follows the standard normal distribution (N(0, 1)) under the Null hypothesis, then we can reduce the verification problem to a classical Kolmogorov-Smirnov normality test in the following way. Given that one can observe the network channel for an arbitrarily large period of time, it is possible to collect many realizations of a random quantity D. Then, once we have at least 100 such observations we conduct Kolmogorov-Smirnov normality test and consider p-value of the test as an indicator of whether our predictions were correctly made. Null hypothesis H_0 for the normality test is an assumption that $D \sim N(0, 1)$. An alternative hypothesis says that D does not follow N(0, 1). Under the Null hypothesis, the p-value is known to be uniformly distributed on the interval [0;1]. At the same time the p-value for data with distribution other than N(0, 1) take small values. Therefore low p-values (less than 0.05) can be considered as a sign that the normality test failed and that prediction did not work correctly. High p-values are indicators that the data do not give us reasons to claim that the approach has failed to predict correctly.

For a better explanation and visualization of the meaning of a p-value in the Kolmogorov-Smirnov normality test we did the following simulation. We generated groups of size 30 of observations of Bernoulli random variables. In each group were present realizations of variables with different success probabilities. Directly from the form of D, it is clear that if the approach either constantly overestimates or underestimates the success probability, then the distribution of D will be significantly shifted from zero and therefore by definition will not be considered even close to standard normal. In our simulations we considered the scenario where probabilities p_i participating in construction of the D statistic had nothing to do with realizations of simulated Y_i . Values p_i were simply randomly taken values from the interval [0;1]. Once we had a group of reported values of the D statistic of size 200, we conducted a Kolmogorov-Smirnov normality test and reported its p-value. The whole aforementioned procedure was repeated 10000 times and 10000 p-values were obtained for the scenario where the prediction was fake. Figure 5.2 shows two probability density functions on the interval [0;1]. The solid line shows the empirical density of p-value of the Kolmogorov-Smirnov test for data where predicted success probabilities were not related to realizations. The dashed line shows the density of the distribution of the p-value under the Null hypothesis, namely the uniform distribution on [0, 1]. We provide figure 5.2 in order to give a clear impression that if reported for a real dataset that the p-value of the Kolmogorov-Smirnov normality test is sufficiently high then this can be considered as a reasonable verification of matching of reported predictions to on-coming events. Low p-values of the test may be considered as an indicator that the predictions are inaccurate.

Remember that the history window size, n, used for prediction of a crossing probability is an important parameter. On one hand it should be large enough to have representative history data. On the other hand, with the increase of the history window, the assumption of stationarity of a time-series $\{X_i\}$ becomes less realistic. Different network paths have different time-periods during which one can treat an available bandwidth related time-series as stationary. We refer here-and-after to such a time-period as a stationary time-period. Our goal therefore is to detect the stationary time-frame for a particular network path.

The "self-verification procedure" described above can be considered as a natural way to detect a suitable range of history window sizes. We repeat the same procedure and obtain p-values for the same history data and different history window sizes. Dependency of reported p-values on the history window size is believed to be unimodal. Naturally, for large window sizes we are supposed to obtain small p-values because the stationarity assumption does not hold and predictions are inaccurate. For small windows, p-values should also be small because there are too few zeroes and ones used for predicting crossing probability \hat{p} and therefore the prediction is not accurate. Therefore, the dependency of a p-value on the history window size should have one mode defining the range of acceptable sizes of history-windows.

The approximate time-scale of a prediction/self-verification is therefore as follows. For predicting of a crossing probability for the period of the next 2 seconds we need around 8 minutes of history frame. However, for doing self-verification and reporting a p-value we need 1500 independent realizations/non-realizations of an event of a crossing. Therefore self-verification may require the time of 15 hours of data-collection. Let us emphasize that actual prediction of a crossing probability for the next 2 seconds is based on the history of the order of 8 minutes and stationarity is assumed only within the frames of these 8 minutes of a history window size. We can start outputting crossing probability immediately after observing a network path for the period of 8 minutes. In addition, after observing a channel for 10-15 hours we are capable also to specify the range of sizes of history windows that are suitable for a prediction and use the window from that range for making our prediction. Finally, if for some network path it turns out that stationarity does not hold for the required time-period and the range of acceptable history window-sizes does not exist, then we can detect this phenomenon and report a temporary inability to make accurate predictions. Decreasing the time-period required for self-verification is important and we address this issue in detail later in this work.

5.3 D-statistic: theoretical results

In this section we give a theoretical background to the fact that D statistic has standard normal distribution given that predictors $\hat{p_i}$ match the actual success probabilities of corresponding Bernoulli random variables Y_i .

We will start with a basic result that establish the Central Limit Theorem when success probabilities p_i are known.

Proposition. Let $Y_1, ..., Y_n, ...$ be independent with distribution $Y_i \sim Bernoulli(p_i)$. Assume that

$$\sum_{i=1}^{\infty} p_i (1-p_i) \to \infty$$

. Then

$$\frac{\sum_{i=1}^{l} (Y_i - p_i)}{\sqrt{\sum_{i=1}^{l} p_i (1 - p_i)}} \Rightarrow N(0, 1)$$

Remark. A simple condition that implies

$$\sum_{i=1}^{\infty} p_i (1-p_i) = \infty$$

is the existence of c>0 such that $c < p_i < 1-c, \, i=1,2,3,\ldots$

Proof. Denote

$$Y_{i,l} = \frac{Y_i - p_i}{\sqrt{\sum_{i=1}^{l} p_i (1 - p_i)}}, i = 1, ..., l$$

Since $E(Y_i) = p_i$ for any *i* we have

a. $E(Y_{i,l}) = 0$, and b. $\sum_{i=1}^{l} E(Y_{i,l}^2) = 1$

The result follows by a standard application of Lindberg-Feller condition. For every $\varepsilon > 0$

$$\sum_{i=1}^{l} E(Y_{i,l}^2; \mid Y_{i,l} \mid > \varepsilon) \to 0 \text{ as } n \to \infty$$

Observe that

$$\sup_{1 \le i \le l} \{ |Y_{i,l}| \} \le \frac{1}{\sqrt{\sum_{i=1}^{l} p_i (1-p_i)}} \to 0 \text{ as } l \to \infty$$

So Lindberg-Feller condition holds. \Diamond

Our next task is to convert the Proposition into a result that matches the setup we are dealing with. Now we assume that p_i are not known, so we replace them by their estimators \hat{p}_i .

Let n be a fixed positive integer. We remark that n appears actually as n - N + 1in other sections. Let $Y_k^{(i)} \sim Bernoulli(p_i)$ be a triangular array of random variables where each row $\{Y_k^{(i)}, k = 0, 1, ..., n\}$ represents a stationary sequence, i = 1, 2, ...Further, we assume that the stationary sequences represented by different rows are independent. We now estimate each p_i by

$$\hat{p_i} = rac{\sum_{k=0}^{n-1} Y_k^{(i)}}{n}$$

To tie our notations with those used in other sections we will denote $Y_i = Y_n^{(i)}$. The following can be proved similarly to the Proposition. Therefore we skip the proof.

Theorem. Assume

$$\sum_{i=1}^{\infty} Var(Y_i - \hat{p}_i) = \infty$$

Then

$$\frac{\sum_{i=1}^{l} (Y_i - \hat{p}_i)}{\sqrt{\sum_{i=1}^{l} Var(Y_i - \hat{p}_i)}} \Rightarrow N(0, 1)$$

The quantity $Var(Y_i - \hat{p}_i)$ can be calculated precisely using $\rho_m^{(i)} = P(Y_k^{(i)} = 1, Y_{k+m}^i = 1)$, a quantity that due to the stationarity assumption does not depend on k. Observe that $\rho_0^{(i)} = p_i$.

Lemma.

$$Var(Y_i - \hat{p_i}) = (1 + \frac{1}{n})p_i - \sum_{m=1}^n (\frac{2m}{n^2})\rho_m^{(i)}$$

Proof of Lemma. We will drop the index i for ease of notation.

$$Var(Y - \hat{p}) = \frac{Var(\sum_{k=0}^{n-1} Y - Y_k)}{n^2}$$
$$= \frac{\sum_{k,j=0}^{n-1} E[(Y - Y_k)(Y - Y_j)]}{n^2} =$$
$$\frac{\sum_{k,j=1}^{n} \rho_0 - \rho_k - \rho_j + \rho_{|k-j|}}{n^2} = (1 + \frac{1}{n})\rho_0 - \sum_{m=1}^{n} (\frac{2m}{n^2})\rho_m$$

Using the Lemma we can rewrite the convergence in the Theorem as

$$\frac{\sum_{i=1}^{l} (Y_i - \hat{p}_i)}{\sqrt{(1 + \frac{1}{n}) \sum_{i=1}^{l} p_i - \sum_{i=1}^{l} \sum_{m=1}^{n} (\frac{2m}{n^2}) \rho_m^{(i)}}} \Rightarrow N(0, 1)$$

A natural estimate of $\rho_m^{(i)}$ is given by

$$\hat{\rho}_m^{(i)} = \frac{\sum_{k=0}^{n-1-m} Y_k^{(i)} Y_{k+m}^{(i)}}{n-m}$$

We conclude that for large n we get that

$$\frac{\sum_{i=1}^{l} (Y_i - \hat{p}_i)}{\sqrt{\sum_{i=1}^{l} \hat{p}_i - \sum_{i=1}^{l} \sum_{m=1}^{n} (\frac{2m}{n^2}) \hat{\rho}_m^{(i)}}} \Rightarrow N(0, 1)$$

In the extreme case where $\{Y_k^{(i)}\}, k=0,1,...,n$ are independent then $\rho_m^{(i)}=p_i^2, m\geq 1$ and we get:

$$Var(Y_i - \hat{p}_i) = (1 + \frac{1}{n})(p_i - p_i^2) \sim p_i(1 - p_i)$$

In general, we do not have independence but only stationarity. However, if the stationary sequence has enough ergodic (mixing) properties then we do get $\rho_m^{(i)} \sim p_i^2$,



Figure 5.3: Evolution of the crossing probability along with its 95% confidence intervals throughout the day. M = 65Mbps, N = 5, $\tau = 0.43$ sec.

at least for large m. Now we look at the expression $\sum_{m=1}^{n} \left(\frac{2m}{n^2}\right) \rho_m^{(i)}$ and we see that it represents a weighted average of $\{\rho_m^{(i)}\}$ where most of the weight is on large value of m. So an assumption of strong ergodicity will lead to

$$\sum_{m=1}^{n} (\frac{2m}{n^2}) \rho_m^{(i)} \sim p_i^2$$

In that case we can write that approximately

$$D = \frac{\sum_{i=1}^{l} (Y_i - \hat{p}_i)}{\sqrt{\sum_{i=1}^{l} \hat{p}_i (1 - \hat{p}_i)}} \sim N(0, 1)$$

The formula above is used in the following section.

5.4 D-statistic: experimental evaluation

Figure 5.3 shows how crossing probability evolved throughout the day for the network path between nodes at Michigan State University (35.9.42.90) and the Shenyang Institute of Automation of the Chinese Academy of Sciences (210.72.135.98). The accuracy of prediction is clearly shown on the figure by means of the reported 95% confidence interval.



Figure 5.4: Probability density functions along with p-values collected for the route 35.9.42.90 (Michigan State University) - 137.189.100.235 (Chinese University of Hong Kong).



Figure 5.5: Probability density functions along with p-values collected for the route 35.9.10.14 (Michigan State University) - 210.72.135.98 (Shenyang Institute of Automation, Chinese Academy of Sciences).

Experimental verification of correctness of crossing probability prediction has been performed on multiple network paths where experimental nodes were located in the United States, mainland China, and Hong Kong. On all paths we observed the same pattern which is well-illustrated on two examples below.

The verification procedure described in the previous sections has been applied to real measurements for verifying whether prediction of crossing probability made by our module is correct. Verification has been performed for different values of aggregation time τ , bandwidth threshold M, and number N of observations over which we predict. Note that the machines used for measurements do not belong to the Planetlab network because there have been criticism that Planetlab machines do not reflect typical Internet patterns of available bandwidth between nodes and the processor utilization on Planetlab machines are often high. Processor utilization of measurement machines is an important factor, which needs to be low in order not to affect measurements. Therefore the machines that were used in the experiments were completely dedicated to measurement.

Verification for a particular path and values of τ , N, and M was performed according to the following scenario. First, we collected around 400,000 continuously taken observations of indicators. Given that τ varied up to 0.5 sec for collecting data for one set of parameters, we observed the channel for the period of two days. Measurement of indicators was performed with the help of modified Pathload's code. We enabled logging of one way delay times in Pathload, and used our own module for processing and analyzing these times. Also we removed Pathload's binary search part of the code and forced the Pathload to send packets at the user-defined rate.

After obtaining 400,000 continuous observations of indicators, we used the window of size N * 200 for making our first prediction of the crossing probability for next N observations. After the prediction was made we checked whether the bandwidth threshold M was really crossed for next N observations thus obtaining the first realization of Bernoulli random variable Y_1 considered in the previous sections. Then the window used for prediction of the length 200 * N was shifted 100 observations ahead and new prediction was made for next N observations of indicators. Therefore adjacent groups of size N were separated by 100 observations. We considered such a separation sufficient for claiming that two adjacent groups are independent on each other. Then the same procedure of shifting and predicting has been repeated 30 times allowing us to collect 30 observations of crossing probabilities p_i and indicators of actual realization of the predicted event Y_i . With the help of aforementioned observations we construct the first value of the D statistic. Then we continue with another portion of 30 cycles of predicting crossing probability, checking the realization of predicted event and then shifting the window ahead. We therefore obtained 130 values of the D statistic and then applied the Kolmogorov-Smirnov normality test to these 130 values. Finally we report p-value of this test.

The whole procedure of data collection and analysis has been repeated for values of bandwidth threshold $M \in 55, 60, 65$ Mbps and $N \in 5, 10, 15$. Our verification test can be viewed as a two class categorization problem. The first "matched" class may be defined as the class of input instances of success probabilities of Bernoulli random variables together with their realizations when realizations match probabilities. The second "unmatched" class is the class of instances where success probabilities are not related to realizations. The p-value is the only feature based on which we decide to which class given instance belongs. Figure 5.2 show class-conditional densities of the p-value. In other words, for any new unseen instance p-value, Figure 5.2 suggests to assign it to the class which has higher probability density at the point which corresponds to that p-value. Therefore for each given network path we have to check that at least 95% of p-values relate to real-life data, which clearly suggest to us to assign those instances to the "matched" class. Figure 5.4 and Figure 5.5 show p-values for two paths that we monitored for a significant period of time. The p-values for different runs are shown as "+" signs. Note that the y-coordinate of observed p-values (shown on plots by "+" signs) was assigned a random value just for better display of the spread of data along the x-direction. The x coordinate of "+" signs is meaningful and represents the observed p-value. For clarity purposes, we copied two class-conditional densities to the result plot. With the densities on the plot, it is clear which class-density takes higher value at points corresponding

	400	600	800	1000	1200	1400	Valid Range of Size	
						1400	Obs	Seconds
1	0.004	0.035	0.838	0.475	0.0135	0.0045	800-1000	400-500
2	0.041	0.583	0.236	0.846	0.2456	0.0094	600-1200	300-600
3	0.003	0	0.004	0.242	0.5723	0.0389	1000-1200	500-600
4	0.467	0.683	0.007	0.005	0.0023	0.0003	400-600	200-300
5	0.097	0.344	0.272	0.012	0.0241	0.003	600-800	150-200
6	0.572	0.296	0.973	0.386	0.742	0.067	400-1200	100-300
7	0.042	0.634	0.437	0.027	0.0024	0	600-800	150-200
8	0.003	0.048	0.473	0.489	0.0081	Ő	800-1000	200-250
9	0	0	0.346	0.39	0.6723	0.0038	800-1200	200-300

Table 5.1: Valid range of history window sizes

to p-values of tests conducted against real prediction measurements. Figure 5.4 and Figure 5.5 show that all instances of our verification procedure are clearly suggested to be assigned to the "match" class. Also one can note that the whole pattern of p-values on both figures is similar to the pattern of realizations of a uniform random variable.

Table 5.1 shows dependency of reported p-values on the chosen history-window size. Each row in a table represents a different network path. Each column represents a different size of a history-window used for prediction. All predictions were made over windows of size $N \in \{2, 3, 4, 5\}$. The two right columns represent the range of acceptable history window sizes in observations and in seconds, respectively. As it follows from the table, some experiments were made when observations were sampled at the rate 2 observations per second and some at the rate 4 observations per second. As figure 5.2 suggests we assume that the particular history window-size works well if reported p-value is larger than 0.1. Table 5.1 clearly shows that the trade-off between complying stationarity assumption on one hand and having representative statistic on another hand can be achieved in practice for all analyzed cases.

We conclude this section with a note that our prediction without an exclusion passed the accuracy verification test at all paths that we monitored.

5.5 Prediction optimization with weights

In addition to choosing an optimal history window-size described in section 5.2, we can use the same data that were used for that purpose for further increasing prediction accuracy by consideration of different weights applied to the recent and older past. The predictor $\hat{p} = \frac{C}{n-N+1}$ shown in previous sections can be viewed as a partial case of applying weights. We can view the counter C as a sum of indicators I_k , $k \in \{1, 2, ..., n - N + 1\}$ where $I_k = 1$ when at least one $X_i = 1$, $i \in \{k, k+1, ..., k+N-1\}$ and $I_k = 0$ otherwise. We therefore can see that the same weight $\frac{1}{n-N+1}$ is applied to each of n - N + 1 indicators within the frames of a history window. Naturally, all weights should add up to one. In this work we propose to use weights of the form $\frac{1}{H}k^{\gamma}$ where the normalizing constant H can be determined from the equation

$$H = \sum_{k=1}^{n-N+1} k^{\gamma}$$

Therefore the predictor of a crossing probability takes the following form

$$\hat{p} = \sum_{k=1}^{n-N+1} \frac{1}{H} k^{\gamma} I_k$$

The expression for \hat{p} suggests assigning higher weights to the recent past, which corresponds to higher values of k. A parameter γ describes the rate at which weight decay and can be adjusted based on the same history data that were used for an adjustment of a history window size. For adjustment of γ , we propose to minimize a quantity of a total relative error as follows. If we obtain from a "self-verification" dataset many predictors \hat{p}_i together with the following realizations Y_i , then the total relative error can be evaluated as

$$Error = \sum_{i} |\hat{p_i} - Y_i|$$

The quantity of an error introduced above does not tell us much in an absolute sense about the accuracy of our prediction. For that purpose we use an appropriate statistical procedure such as the one introduced in the section 5.2 together with its p-value. However, minimization of a relative error can help tune the rate γ of weights decay.

Overall, our algorithm for tuning parameters to a specific network path may be described as follows. First, we collect sufficient data for conducting a "selfverification" procedure. The minimization of the amount of data and the time required for that purpose will be considered later in this work. We assign to γ a value 0, which corresponds to the case with uniform weights. Then we run the self-verification algorithm for a number of window sizes, where *n* varies within the range of [50 * N; 1000 * N]. We use obtained p-values to obtain an optimal value of *n*, which corresponds to the range where p-values are high. If the range with sufficiently high p-values does not exist, then we still take *n* that corresponds to the maximal reported p-value. With the newly obtained history window-size *n*, we minimize the total relative error and thus find the best rate γ . Then we use the new value of alpha and again try different window sizes to determine the new range of sizes for history-windows. The expected outcome is that the range of applicable window sizes either arises or expands after applying the weight-based technique.

The usage of the weights within the frames of a stationary time-frame can be motivated by the fact that the Internet traffic time-series demonstrates a positive correlation pattern at small lags. In other words, higher values of traffic imply higher values at close time-moments, whereas small values of traffic imply small values. The available bandwidth time-series is closely related to a cross-traffic and sometimes is directly equal to a constant minus the cross-traffic through a tight and narrow link. Therefore we believe that the most recent values of the available bandwidth related time-series should be more important than the others.

5.6 R-statistic: improvement of a verification procedure

In this section we discuss an optimization of the self-verification technique. In short, the self-verification procedure collects a number of predictions of a crossing probability \hat{p}_i together with realizations/non-realizations of the predicted events of crossing and then suggests a statistical test that checks if predicted values of a crossing probability really match oncoming events of a crossing. The output of the self-verification procedure is used by a parameters adjustment module, which was introduced in the previous section. Therefore, it is important to reduce the number of observations and respectively, the boot time that the self-verification procedure needs to produce its output.

The statistical test shown in the section 5.2 assumes the creation of the *D*-statistic on the basis of pairs of quantities \hat{p}_i, Y_i . The *D*-statistic follows the standard normal distribution under the null hypothesis, which indicates that predictions match on-coming events. The procedure has to collect at least 50 independent values of the *D*-statistic, then conduct a normality test on them and report the corresponding p-value.

5.6.1 Timing

First we briefly analyze the time that is required for reporting one value of a D statistic. The time-period up to 0.25 seconds is required for conducting a robust check of the available bandwidth time-series against a certain threshold. Therefore we assume that one observation of the original time-series $\{X\}$ of zeroes and ones requires 0.25 seconds to be reported. Note that Y_i is an observation that equals one when at least 1 out of N observations of the underlying time-series $\{X\}$ equals to one. Let us assume for the sake of preciseness that N = 4. Then the time for observing one value Y_i of the realization of an event of crossing is 1 second. Computer simulations, confirmed by real-life experiments, show that the time needed

to report a good predictor of a crossing probability \hat{p}_i varies from 50 * N to 1000 * N. Assuming that the history window-size is of the order of 500 * N, we note that the history time-frame is of the order of 500 seconds (8.3 minutes). The form of the D statistic assumes that observations of Y_i are independent of each other. This can be achieved in practice only by separating them in time by at least 100 observations of the underlying time-series $\{X\}$. Finally, taking into account that we need at least 30 summands to make the D statistic robust, we note that the time for outputting one value of the D-statistic can be evaluated as t = 500 + (100 * 0.25 + 1) * 30 = 1280 sec (21 minutes). We conclude that the boot time for a self-verification module required for a collection of at least 50 values of the D-statistic is of the order of 16 hours.

Our optimization of the self-verification module is a better replacement for a D statistic. Introduced below, the R statistic possesses the same characteristics as the D statistic while requiring only around 500 seconds (8.3 minutes) for the mentioned above example to be reported. This means that time of the order of only 6 hours is needed for reporting the p-value of the normality test. Therefore we reduce the boot time for the self-verification procedure by a factor of 3.

5.6.2 R-statistic

Consider observations of a Bernoulli zero-one time-series of $\{X_i\}, i \in 1, ..., n, n+1, ..., n+N$, which fall into a history time-frame plus the following N observations. Then we create the time-series W_k according to the following rule:

$$W_k = \max\{X_{k+1}, X_{k+2}, ..., X_{k+N}\} - \hat{p}_k, \ k \in \{N, ..., n\}$$

where \hat{p}_k is a predictor of a crossing probability based on the first k observations of time-series $\{X_i\}$. Then the following statement holds:

$$R = \frac{\sum_{k=N}^{n} W_k}{\sqrt{(n-N+1)\hat{\sigma}_n^2}} \Rightarrow N(0,1)$$

where

$$\hat{\sigma}_n^2 = \sum_{i=N}^n \frac{W_i^2}{n-N+1} + 2\sum_{k=N}^{n-30} \sum_{i=N}^{n-k} \frac{W_i W_{i+k}}{n-N+1-k}$$

Figure 5.6 shows the way how R-statistic is created. Also figure 5.6 illustrates gain in time that is obtained when moving from D-Statistic to R-statistic. For illustration on figure 5.6 we have chosen N = 4, n = 2000 and sampling rate equal to 4 observation per second. To obtain the R statistic, we enter the history window of size n, start with a small history of size N, and then for each summand in a numerator begin to increase the window size one observation at a time. In each summand we predict the crossing probability \hat{p}_k and then check it against the following (unseen by $\hat{p}_k)$ Nobservations of a history data. Naturally, summands obtained in that way are highly dependent. The main advantage of the R statistic over the D statistic is that the former does not require independence of underlying observations W_k , whereas the D statistic requires Y_k to be separated in time. One can see that a variance expression $\hat{\sigma}_n^2$ has a term that considers correlations between observations of W_k values. We can fit one value of the *R*-statistic directly into the history time-frame. Therefore, with the R-statistic, the available data are used much more efficiently. The following section provides a mathematical proof of the statement above telling that the Rstatistic should follow a standard normal distribution under the hypothesis that predictions match realizations.

5.7 R-statistic: theoretical results

We start with a definition of the concept of strongly mixing. It is one of the criterion used to quantify a situation in which the longer the time between observations, the more statistically independent those observations become.

Definition. Let ..., $X_{-1}, X_0, X_1, ..., X_n, ...$ be a strictly stationary sequence of random variables. It is called strongly mixing if $\alpha(n) \to 0$ where $\alpha(n)$ is defined by

$$\alpha(n) = \sup_{A \in F_{-\infty}^{0}, B \in F_{n}^{\infty}} \{ |P(A \cap B) - P(A)P(B)| \}$$



1 value of D-statistic: history of size 2000obs + 30 windows of size 4obs + 30 gaps of size 100obs = 5120obs (or 1280 seconds when 4obs are taken for 1 second)

1 value of R-statistic: we need just history frame with size 2000obs + 1 window of size 4obs = 2004obs (or 501 seconds when 4obs are taken for 1 second)

Figure 5.6: Reducing the time of self-verification procedure when switching from R-statistic to D-statistic

where $F_{-\infty}^0 = \sigma\{X_k : k \le 0\}$ and $F_n^\infty = \sigma\{X_k : k \ge n\}.$

Ibragimov and Linnik proved in [28] the following Central Limit Theorem.

Theorem 1 (18.5.4). Let $V_1, ..., V_n, ...$ be a strictly stationary sequence of random variables with $E(V_n) = \mu$. Assume that the sequence is uniformly bounded (i.e. there is C > 0 with $|V_n| \le C$, a.s.), and strongly mixing with $\sum_{n=1}^{\infty} a(n) < \infty$. Then $\sigma^2 \equiv Var(V_0) + 2\sum_{k=1}^{\infty} COV(V_0, V_k)$ converges. If $\sigma^2 \ne 0$ then $\sum_{k=1}^{n} \frac{V_k - \mu}{V_{\sigma}} \Rightarrow N(0, 1)$

A comment about the proof of Theorem 1. It is based on organizing the sequence into blocks that alternate between long and short ones. The total contribution of the short blocks to the limiting distribution is negligible. One then takes advantage of the assumption on the mixing coefficients, namely: $\sum_{n=1}^{\infty} a(n) < \infty$, by showing that the long blocks are essentially independent. Since the long blocks are identically distributed due to the stationary assumption the result follows.

For our purpose in this article we need to convert Theorem 1 into a Central Limit

Theorem, where the mean μ is replaced by an estimate that is updated regularly with the additional data available to us as time moves on. We denote $\bar{\nu}_n = \frac{\sum_{i=1}^n V_i}{n}$, n = 1, 2, ... and we have

Theorem 2. Under the assumptions of Theorem 1 we have

$$\frac{\sum_{k=1}^{n} W_k}{\sqrt{n\sigma}} \Rightarrow N(0,1)$$

where $W_n = V_n - \bar{\nu}_n$.

Comments about the proof of Theorem 2. We will not provide a full detailed proof due to the length consideration. Instead we make a few comments. Denote: $a_{k,n} = 1 - b_{k,n}$, where $b_{k,n} = \sum_{i=k}^{n} \frac{1}{i}$, k = 1, ..., n, and then with $\mu = E(V_n)$ we get

$$\frac{\sum_{k=1}^{n} W_k}{\sqrt{n}\sigma} = \sum_{k=1}^{n} \frac{a_{k,n}V_k}{\sqrt{n}\sigma} = \sum_{k=1}^{n} \frac{a_{k,n}(V_k - \mu)}{\sqrt{n}\sigma}$$

were the first equality can be verified by simple algebra while the second follows from $\sum_{k=1}^{n} a_{k,n} = 0$, which in turn follows from $\sum_{k=1}^{n} b_{k,n} = \sum_{k=1}^{n} \sum_{i=k}^{n} \frac{1}{i} = \sum_{i=1}^{n} (\frac{1}{i}) \sum_{k=1}^{i} 1 = n$. We would like to show now that

(1)
$$E(\sum_{k=1}^{n} \frac{a_{k,n}(V_k - \mu)}{\sqrt{n}})^2 \to \sigma^2 \text{ as } n \to \infty$$

We will show that

(2)
$$\sum_{k=1}^{n} \frac{a_{k,n}^2}{n} \to 1 \quad as \ n \to \infty$$

0

Indeed

$$\frac{\sum_{k=1}^{n} a_{k,n}^2}{n} = \frac{\sum_{k=1}^{n} (1-b_{k,n})^2}{n} = -1 + \frac{\sum_{k=1}^{n} b_{k,n}^2}{n}$$
(since $\sum_{k=1}^{n} b_{k,n} = n$)

So to establish (2) we need to show that

(3)
$$\frac{\sum_{k=1}^{n} b_{k,n}^2}{n} \to 2 \text{ as } n \to \infty$$

Since $b_{k,n} \sim -log(\frac{k}{n})$ we calculate

$$\frac{\sum_{k=1}^{n} b_{k,n}^2}{n} \sim \sum_{k=1}^{n} \log^2(\frac{k}{n})(\frac{1}{n}) \to \int_0^1 \log^2(x) dx = [x[\log(x)(\log(x) - 2) + 2]]|_0^1 = 2$$

and (2) is established. In a similar way we can prove that for each fixed k > 0 we get

(4)
$$\sum_{i=1}^{n-k} \frac{a_{i,n}a_{i+k,n}}{n} \to 1 \text{ as } n \to \infty$$

A standard calculation gives

$$E\left(\sum_{k=1}^{n} \frac{a_{k,n}(V_k - \mu)}{\sqrt{n}}\right)^2 = \left(\sum_{k=1}^{n} \frac{a_{k,n}^2}{n}\right) Var(V_0) + 2\sum_{k=1}^{n-1} \left(\sum_{i=1}^{n-k} \frac{a_{i,n}a_{i+k,n}}{n}\right) COV(V_0, V_k)$$

which leads to (1) by using (2) and (4). To prove Theorem 2, we now need to repeat the idea of alternating short and long blocks as was explained in the comment after Theorem 1. Finally we remark that an attempt to prove Theorem 2 by using Theorem 1 and $\bar{\nu}_n \sim \mu$ (justified by the ergodic theorem) will fail since the difference between the relevant 2 quantities is not negligible even in the independent case.

Theorem 2' (Extension of Theorem 2 to weighted sample means). Fix $\gamma > 0$ and let $W_n = V_n - \bar{\nu}_n$ where $\bar{\nu}_n = \frac{\sum_{k=1}^n k^{\gamma} V_k}{T_{n,\gamma}}$ where $T_{n,\gamma} = \sum_{k=1}^n k^{\gamma}$.

Under the assumptions of Theorem 1, we have:

$$\frac{\sum_{k=1}^{n} W_k}{\sqrt{n}\sigma} \Rightarrow N(0, \frac{1}{2\gamma + 1})$$

Comments on the proof. Here we will define $b_{k,n} = k^{\gamma} \sum_{i=k}^{n} \frac{1}{T_{i,\gamma}}$, k = 1, ..., n, and we still get $\sum_{k=1}^{n} b_{k,n} = n$, and

$$\frac{\sum_{k=1}^{n} W_k}{\sqrt{n}\sigma} = \sum_{k=1}^{n} \frac{a_{k,n}(V_k - \mu)}{\sqrt{n}\sigma}$$

As before we will concentrate only on the asymptotic behavior of the second moment:

$$(1)' \quad E(\sum_{k=1}^{n} \frac{a_{k,n}(V_k - \mu)}{\sqrt{n}})^2 \to \frac{\sigma^2}{2\gamma + 1} \quad as \ n \to \infty$$

To prove (1)' we need to show

$$(3)' \quad \frac{\sum_{k=1}^{n} b_{k,n}^2}{n} \to 1 + \frac{1}{2\gamma + 1} \quad as \ n \to \infty$$

which is proved as follows: $T_{n,\gamma} = \sum_{k=1}^{n} k^{\gamma} \sim \frac{n^{\gamma+1}}{\gamma+1}$ and then

$$b_{k,n} = k^{\gamma} \sum_{i=k}^{n} \frac{1}{T_{i,\gamma}} \sim (\gamma+1)k^{\gamma} \sum_{i=k}^{n} \frac{1}{i^{\gamma+1}} \sim -\frac{\gamma+1}{\gamma} k^{\gamma} [\frac{1}{i^{\gamma}}]_{i=k}^{n} = \frac{\gamma+1}{\gamma} (1-\frac{k^{\gamma}}{n^{\gamma}})$$

So we get

$$\frac{\sum_{k=1}^{n} b_{k,n}^2}{n} \sim \left[\frac{\gamma+1}{\gamma}\right]^2 \sum_{k=1}^{n} (1 - \frac{k^{\gamma}}{n^{\gamma}})^2 (\frac{1}{n}) \rightarrow$$
$$\left[\frac{\gamma+1}{\gamma}\right]^2 \int_0^1 (1 - x^{\gamma})^2 dx = 1 + \frac{1}{2\gamma+1} \quad as \ n \to \infty$$

Equipped with (3)' we get:

$$\sum_{i=1}^{n-k} \frac{a_{i,n}a_{i+k,n}}{n} \to \frac{1}{2\gamma + 1}, \ k \ge 0$$

and (1)' follows as before.

It will useful for us to replace Theorems 2 and 2' by the following version which follows immediately from Theorem 2 by a standard probability argument.

Theorem 3. Fix $\gamma \geq 0$. Under the assumptions of Theorem 1 we have

$$\frac{\sum_{k=1}^{n} W_k}{\sqrt{n\sigma_n^2}} \Rightarrow N(0, \frac{1}{2\gamma + 1})$$

if $\frac{\sigma_n^2}{\sigma^2} \to 1$ in probability.

Three methods to estimate σ^2 . Here we use $\gamma = 0$. From (1) we see that if n is large enough we get

$$E(\sum_{k=1}^{n} \frac{W_k}{\sqrt{n}})^2 = E(\sum_{k=1}^{n} \frac{a_{k,n}(V_k - \mu)}{\sqrt{n}})^2 \sim \sigma^2$$

Using the fact that $\{W_n\}$ is approximately stationary we see that we can utilize Method 1:

$$\hat{\sigma}_n^2 = \sum_{i=1}^n \frac{W_i^2}{n} + 2\sum_{k=1}^{n-M} \sum_{i=1}^{n-k} \frac{W_i W_{i+k}}{n-k}$$

where M is a fixed number (say 30). This works since by the ergodic theorem $\sum_{i=1}^{n} \frac{W_i^2}{n} \sim E(W_0^2)$ and $\sum_{i=1}^{n-k} \frac{W_i W_{i+k}}{n-k} \sim E(W_0 W_k)$.

Alternatively we may use Method 2:

$$\hat{\sigma}_n^2 = \sum_{k=1}^{n-M} \frac{(\sum_{i=k}^{k+M_n} W_i)^2}{M_n(n-M_n)}$$

where we can take for example $M_n \sim \sqrt{n}$. Again by ergodic theorem we get $\sigma_n^2 \sim$

 $E(\sum_{k=1}^{M_n} \frac{W_k}{\sqrt{M_n}})^2$, so the requirement $\frac{\sigma_n^2}{\sigma^2} \to 1$ in probability, does hold.

Finally, following Peligrad and Shao [47] we can consider as a method 3 the following expression for σ_n^2 :

$$\hat{\sigma}_n^2 = \frac{1}{\log(n)} \sum_{k=1}^n (\bar{\nu}_i - \bar{\nu}_n)^2$$

since under our conditions

$$E\left|\frac{\frac{1}{\log(n)}\sum_{k=1}^{n}(\bar{\nu}_{i}-\bar{\nu}_{n})^{2}}{\sigma^{2}}-1\right|\to 0 \quad as \ n\to\infty$$

5.8 R-statistic: experimental evaluation

We first evaluate our new self-verification procedure and then use it for evaluation of the efficiency of the proposed improvement on the weighted prediction of a crossing probability.

To evaluate the self-verification procedure, we first generated stationary sequences of zeroes and ones and run the self-verification procedure on them. We first generated ARMA(p,q), $p,q \in \{3,4,5,6,7,8,9\}$ stationary sequences, chose a reasonable threshold and then produced "zero" if the time-series remained below the threshold and "one" otherwise. Since generated time-series were known to be stationary by the experimental setup, we always expected to obtain high p-values as an output of the self-verification procedure. Note that zeroes and ones were highly dependent on each other in our setup. We experimented with different windowsizes to learn what is the minimal history-window size that works well for the *selfverification* procedure. This experimentation was conducted because we wanted to separate clearly the potential occurrences of low p-values when running over real data due to a failure of the self-verification procedure to do its job. It turned out that beginning from history-sizes of 200, the self-verification procedure works well

	History size a (# of observations)	200	400	600	800	1000	1200	Valid Range of History-Size	
Ľ	history size in (# Or Observations)							Observations	Seconds
1	p-values before weights adjustment	0.0263	0.3196	0.3702	0.0023	0.0217	0.03022	400-600	100-150
	p-values after weights adjustment	0.11	0.8845	0.6566	0.1258	0.2026	0.2793	400-1200	100-300
2	p-values before weights adjustment	0.0162	0.277	0.1436	0.0198	0.0367	0.0338	400-600	100-150
	p-values after weights adjustment	0.0759	0.6941	0.6769	0.5685	0.2995	0.0158	400-1000	100-250
3	p-values before weights adjustment	0	0	0	0.0072	0.073	0.0061	0	0
	p-values after weights adjustment	0	0	0.0324	0.123	0.251	0.0042	800-1000	200-250
4	p-values before weights adjustment	0.011	0.0974	0.3438	0.272	0.01225	0.0241	600-800	150-200
	p-values after weights adjustment	0.279	0.2048	0.4899	0.1303	0.87	0.6	200-1200	50-300
5	p-values before weights adjustment	0.021	0.054	0.0063	0.003	0.0086	0.0023	0	0
	p-values after weights adjustment	0.026	0.32	0.56	0.054	0.0043	0.0021	400-600	100-150
6	p-values before weights adjustment	0.0522	0.0776	0.3744	0.1035	0.0007	0	600-800	150-200
	p-values after weights adjustment	0.1333	0.4703	0.171	0.6262	0.0198	0.005	200-800	50-200
7	p-values before weights adjustment	0	0	0	0.005	0.5852	0.0085	1000	250
	p-values after weights adjustment	0	0	0.0043	0.103	0.532	0.342	800-1200	200-300

Table 5.2: P-values for different history window sizes before and after weight adjustment.

(reported p-values follow the pattern of uniform distribution on interval [0, 1]). We conclude with figure very similar to 5.2, which shows two densities that describe behavior p-values for our self-verification procedure.

For evaluation of weights with decaying rate adjusted to a particular network path, we used the Auckland II NLANR traces [1] as well as our own measurements of end-to-end available bandwidth.

The Auckland-II dataset [1] is a collection of IP header traces captured with a DAG2 system at the University of Auckland Internet uplink. It consists out of multiple chunks of continuously taken measurements. We used this dataset considering that the available bandwidth over a link is merely a constant, minus the traffic through the link. We processed traces and our output was a link traffic sampled at the rate 4 observations per second. Then we applied our prediction technique with different windows-sizes N and different thresholds M against which we wanted to estimate the crossing probability.

For each trace in Auckland-II dataset we followed the following setup. We first ran a prediction and self-verification module, thus obtaining a range of historywindows that were suitable for a prediction. Then we ran the parameters adjustment procedure and after that again obtained the range of windows that were good for a prediction.

The second part of our experimental verification deals with measurements of end-

to-end available bandwidth that were collected by us over different Internet paths. Measurement nodes were located at Michigan State University, Chinese University of Hong Kong and Shenyang Institute of Automation, Chinese Academy of Sciences. For measuring time-series of indicators $\{X_i\}$ we used a Pathload [29] measurement tool modified based on our algorithm introduced in chapter 4 for a robust trend detection. The difference between working with traffic trace and actual sampling of an available bandwidth is that in the latter case we are unable to alter a threshold because measurements were conducted against one threshold level at a time.

Table 5.2 shows p-values for different windows sizes before and after adjustment of parameters. We present a number of representative samples from the Auckland traces and from our measurements. We predicted the crossing probability for N varying around 5, which corresponds to the time-period of the order of a second. We can easily see that the range of history windows that are valid for a prediction increases after application of an adjustment procedure. In some cases we even see that the suitable history window sizes were not available before running the weights adjustment procedure. This means that in cases mentioned above, the weights adjustment procedure made predictions possible.

We emphasize that the p-value of a test is roughly a likelihood to observe what has been actually observed under the assumption that predictions of a crossing probability match on-coming events. If this likelihood becomes small (say less than 0.1) we conclude that it is very unlikely to see what we see under the assumption of the Null hypothesis and therefore the Null hypothesis has to be rejected with the conclusion that the predictions are not accurate anymore. We therefore assume for our algorithm that if the reported p-value is more than 0.1, then the window that corresponds to that p-value is valid for a prediction. According to this scheme we report in the two right columns the ranges of windows that turned out to be valid for a prediction. In the first column the range is presented in the number of observations while in the second in seconds.

In order to show the performance of the algorithm on many samples, we provide figure 5.7. Figure 5.7 shows two histograms of ranges of acceptable window sizes



Figure 5.7: Histograms of acceptable range of lengths of history windows before and after weights adjustment. The black histogram shows the range distribution after adjustment whereas the gray histogram shows the range distribution before adjustment.

(upper bound minus lower bound) reflecting 100 samples of behavior of an available bandwidth at different days and places. The grey histogram is related to the case when uniform weights were used while the black histogram shows a range of acceptable windows when weight adjustment procedure has been applied. We can see that a gray histogram captures zero, which means that for significant number of traces the acceptable range of history-windows does not exist. The black histogram shows that the application of weight-based adjustment increases on average more than 3 times the range of history windows, which is good for predictions.

5.9 Limits of a verification procedure

In this section we discuss limits of applicability of the above-introduced self-evaluation procedure. An application of our self-evaluation technique naturally fails when reported values of a crossing probability are extremely small. Consider, for example, a limit case when the whole history window does not capture even one event of a crossing. Then the predicted value of a crossing probability will be evaluated as zero and the estimated value of a variance $\hat{\sigma}_n^2$ in the expression for R statistic also becomes zero. The theoretical proof in section 5.7 also assumes a non-degenerate case for predicted values of a crossing probability and thus a non-degenerate case for the variance σ^2 . In practice one can observe that if the predicted values of a crossing probability are too close to zero then reported p-values become small just because self-evaluation procedure does not do its job. We note that if the crossing probability is nearly zero then the operator concludes that network environment is safe and tele-operations can be allowed anyway. An operator needs to track cases when the crossing probability significantly differs from zero and for these cases self-evaluation technique works well.

We believe, therefore, that it is important to understand for how small values of a predicted crossing probability the self-evaluation technique still works correctly.

With the Auckland-II dataset we were in possession of exact values of available bandwidth time-series. Therefore we were able to freely alter threshold against which a crossing probability had to be estimated.

Altering the crossing probability threshold made it possible to determine the range of applicability of our verification technique with respect to the value of a crossing probability which has to be estimated. We note that with a decrease of the value of a crossing probability (by moving down an available bandwidth threshold), the history window size needs to be increased for avoiding a degenerate case in the denominator for an R-statistic. Then the history window can still capture a couple events of a crossing. In other words by increasing the history window we increase the granularity at which the crossing probability can be predicted and therefore allow crossing probability values to be closer to zero. At the same time the upper bound for the length of a history window does not depend on the value of a reported probability. This bound is solely defined by the time-period during which one can still consider the available bandwidth time-series as stationary. Therefore for each fixed future window size N there exists a minimal value of a crossing probability P at which the acceptable range for a history window size decays completely.

We note that the size N of a future time-frame over which the prediction of a crossing probability is made also affects the range of history window sizes. Suppose, for example, that a crossing probability P is fixed at a certain value. Then it is natural to assume that with an increase of the window size N, the history size n also has to be increased (such that history window of size n still contains a significant number of non-overlapping windows of size N).

Based on aforesaid we conclude that the minimal value of a crossing probability for which the self-evaluation test still works well should be increased with an increase of a window size N over which the prediction is made.

We used the Auckland-II dataset for experimental determination of how the mentioned above minimal predicted value of a crossing probability depends on the size N of the future window over which the prediction has to be made. Also we compared applicability limits of prediction/verification techniques with R-statistic and weights against the one with D statistic .

For each technique (the new one related to *R*-statistic and weights-based pre-



Figure 5.8: Dependency of minimal value of a crossing probability for which verification procedure works well on the future time-window over which prediction is made and on prediction/verification technique

diction, and the one related to *D*-statistic) we conducted the following procedure. First, we fixed the future window size N at a certain value. Then we made a binary search for a threshold which on one hand still reports the maximal p-value higher than 0.1 and on another hand has the value of a crossing probability as small as possible. Note, that for each value of N and each threshold level we tried a set of different history window sizes and finally selected the maximal p-value among those that were reported. In other words, if at least one history window size existed for which the reported p-value was higher than 0.1 then we were further decreasing the available bandwidth threshold for achieving further decrease of the value of a crossing probability. Finally, after the binary search converged, we outputted the value N together with values of a crossing probability. These minimal and acceptable crossing probability values were obtained while checking available bandwidth against the "boundary" threshold obtained with the help of mentioned above binary search. In figure 5.8 we marked these values by red "+" signs for the experiment with old prediction/verification technique without weights. Minimal acceptable crossing probability values related to the R-statistic based prediction/verification technique are marked in figure 5.8 by blue "*" signs.

The same procedure has been repeated for $N \in \{1, 2, 3, 4, 5, 6\}$, where one observation corresponds to a time-period of 0.25 seconds. Therefore the future time-frame over which the prediction was made varied within the range of [0.25; 1.5] seconds. Note, that for representation purposes we slightly shifted apart "+" and "*" signs. For example, "+" and "*" related to a time-period of 0.25 seconds are shown as observations related to 0.24 and 0.26 seconds respectively.

Finally, on figure 5.8 we connected the mean values of a minimal crossing probabilities reported for each N with two solid lines (for old and new techniques respectively). From figure 5.8 we conclude that the verification approach still works for real data when the value of a crossing probability P drops as low value as 0.01. Figure 5.8 shows that, as it was expected, the minimal value of P increases with an increase of a window size N over which the prediction is made. Also from figure 5.8 clearly follows that new prediction/verification technique significantly decreases lower bound for an acceptable values of a crossing probability.

5.10 Conclusion

The basic result in this Chapter represents a robust non-parametric method of computation of a "crossing probability". This probability defines a quality of an end-toend network path with respect to a specific real-time task. The probability represents the chances of a real-time task execution to fail due to the inability of the path to give sufficient bandwidth to communicating parties.

Our approach has a capability of dynamically choosing the length of the window n that is used for prediction. Since we provide a robust verification technique, our algorithm can continuously do a self-check of the accurateness of the prediction at run-time and automatically adjust the length of a prediction window. Low p-values of a test can be considered as a clear indicator that for a given network path

the window size n does not do a good job in prediction and therefore needs to be adjusted.

After presenting a basic non-parametric prediction/verification technique we introduce two significant improvements of a non-parametric approach to estimation of a crossing probability. First, the improvement deals with tuning basic parameters such as the size of the history n used for a prediction and also the coefficient α that can be treated as a rate at which the network path "forgets" its history. This is important because there exists a large variety of patterns of traffic along different Internet links. Some links may be highly multiplexed by a large number of users and some may have only sporadical traffic bursts associated with instantaneous user activity. At some links, the majority of a traffic may be constituted by a video-transfer, which has highly-revealed long-memory nature and at some links traffic may even represent a Gaussian time-series with exponentially decaying autocorrelation, and therefore short memory properties. Since the available bandwidth time-series is tightly related to a cross-traffic, we see automatic tuning to a specific network path as a crucial factor, which makes our approach applicable to a wide range of behaviors of available bandwidth time series across the Internet. Second, the improvement deals with an efficient procedure that helps to verify the correctness of the prediction and therefore is a kernel for efficient parameters adjustment to a specific network path. We accompany both, D-statistic and R-statistic with a solid theoretical argumentation that proves the validity of a chosen verification procedures. Therefore the work represents a combination of theoretical results and experimental verification of these results applicability to various real-life data-sets. We test the efficiency of a weight-based approach with decaying rate α on publicly available traffic data-sets as well as on data-sets of available bandwidth collected by us on a number of Internet paths.

Chapter 6

Re-route Sequence Planning Problem

As it was stated in Chapter 3, a tele-operation can be viewed as a number of flows with predefined QoS requirements. These flows need to be allocated across the Internet through the best possible network paths.

We believe that the promising suggestion is to consider allocation of tele-operation "facilitators" at geographically dispersed parts of the Internet. The idea is in setting up an application-level virtual network that facilitates allocation and reallocation of channels through different network paths. There are a number of efforts that consider dynamic network high-level route allocation, deallocation with the help of overlay networks [2, 37] for optimization of network delay or guaranteeing compliance to a given QoS requirements.

In this chapter we take one step towards setting up such an infrastructure while considering a mathematical aspect of a Reroute Sequence Planning Problem [32]. The problem can be briefly stated as follows. Assume that a set of channels with a given bandwidth requirements or demands is allocated through the currently optimal set of network paths. Then sometime later another optimal set of network paths is computed. The second set of paths reflects the changed QoS conditions of the network environment and therefore may differ from the previous set. The Reroute Sequence Planning problem chooses the optimal sequence of channel reallocations from an old set of paths to a new set of paths that minimizes the amount of extra network-link bandwidth required for conducting the reallocation procedure.

This chapter deals with the well-known mathematical "Dynamic Vector Summation" problem to which the Reroute Sequence Planning problem can be easily reduced. In particular, we give a tight bound for maximum extra link bandwidth capacity required for carrying out reroute procedure. We also provide an algorithm that finds in polynomial time the sequence of route reallocations that complies proposed bandwidth bound.

6.1 Introduction

The compact vector summation as a pure mathematical tool has been introduced and developed at the beginning of 20-th century by Levy [35] and Steinitz [55]. The problem can be stated as follows: given vectors in a normed linear space find their rearrangement π for which the maximum deviation R_{π} of partial sums from zero is as small as possible; or, estimate R_{π} under the best rearrangement π .

There are at least two sound functional-analytic problems where these methods are being used. One of them is the description of the set of sums of a conditionally convergent series in a topological vector space. Another problem is the famous Kolmogorov problem (open so far) on the system of convergence of abstract Fourier series under a rearrangement of the orthonormal basis. For an extensive bibliography devoted to this subject, see [25, 33, 22, 10].

There are also a series of problems in scheduling theory that reduce to this problem stated in the next section (see [52]). The Reroute Sequence Planning (RSP) problem described by Makai [41] is an example of such a scheduling problem in telecommunication networks. Below we give a description of RSP problem.

A network consists of the set V of vertices and the set E of edges connecting certain pairs from V (not necessarily all pairs), |E| = d (the number of edges is d). Consider a system of paths $\mathcal{P} = (P_1, \ldots, P_n)$, the path P_i connecting the



Figure 6.1: Network graph example

vertices s_i and t_i , i = 1, ..., n. By a path connecting s_i and t_i we mean a broken line consisting of edges with the starting point at s_i and endpoint at t_i (of course one can state the problem formally, entirely in terms of the abstract graph theory).

Fig. 6.1 shows an example network consisting of 8 vertices and 8 edges. Consider a system $\mathcal{P} = (P_1, P_2)$ of 2 paths, where P_1 connects s_1 and t_1 and consists of the edges e_1, e_4, e_6 and e_7 , while P_2 connects s_2 and t_2 and consists of e_2, e_4 and e_8 .

Each of the paths $P_k, 1 \le k \le n$ imposes a demand v_k on each of the edges of the path. If an edge belongs to several paths, then the total demand u(e) imposed to this edge is the sum of corresponding demands. For example, in Fig 6.1 the edge e_4 which is common for P_1 and P_2 has the total demand $v_1 + v_2$. The demand on the edges e_1, e_6 and e_7 is v_1 , while the demand on e_2 and e_8 is v_2 .

Each of the edges e_k has a value of *capacity* c_k assigned to it. Capacity for a given edge represents the maximal demand that can be imposed to the edge. If for every edge in E the total imposed demand does not exceed the edge capacity, then

the network is believed to operate without violation.

In reroute sequence planning (RSP) the system $\mathcal{P} = (P_1, \ldots, P_n)$ of paths should be replaced by another system $\mathcal{Q} = (Q_1, \ldots, Q_n)$ that connects the same vertices s_i and t_i , $i = 1, \ldots, n$. Each path Q_i in \mathcal{Q} has the same demand v_i as the corresponding path P_i in \mathcal{P} . In a reroute sequence procedure at each step only one P_i is being replaced by the corresponding Q_i . The RSP problem consists of finding a sequence g_1, \ldots, g_n of rearrangements of $\{1, \ldots, n\}$ such that the successive replacement P_{g_1} by Q_{g_1}, \ldots, P_{g_n} by Q_{g_n} does not cause the demand v(e) after each replacement to exceed the capacity c(e) for each e. Denote by $x_i(e)$ the change in demand for the edge e when P_i is replaced by Q_i . We have

$$x_i(e) = \begin{cases} v_i, & \text{if } e \notin P_i, \ e \in Q_i \\ -v_i, & \text{if } e \in P_i, \ e \notin Q_i \\ 0 & \text{otherwise.} \end{cases}$$

The vectors x_1, \ldots, x_n belong to the space \mathbf{R}^d , where d, the dimension of the space, is the number of edges. Denote by $L_0(e)$ the starting demand on the edge e, and assume we have chosen the permutation defined by a sequence g_1, \ldots, g_n of replacements P_{g_1} by Q_{g_1} , P_{g_2} by Q_{g_2}, \ldots, P_{g_n} by Q_{g_n} . Then the demand on the edge e, after the *l*-th step is

$$L_0(e) + x_{g_1}(e) + \ldots + x_{g_l}(e)$$
.

This implies that the demand on the edge e at each step does not exceed

$$L_{0}(e) + \max_{1 \le l \le n} \max_{e \in E} |xg_{1}(e) + \dots + xg_{l}(e)|$$

= $L_{0}(e) + \max_{1 \le l \le n} ||xg_{1} + \dots + xg_{l}||_{\infty} = L_{0}(e) + |||x_{\pi}|||$

where $\|\cdot\|_{\infty}$ denotes the maximum-norm, while $\||x_{\pi}\|| = \max_{1 \leq l \leq n} \|x_{g_1} + \ldots + x_{g_l}\|_{\infty}$. Thus, we have come to the problem of finding a permutation σ minimizing

 $|||x_{\pi}|||.$

In Section 6.2 we state the problem on compact vector summation and prove the basic Transference Lemma that provides easy proofs for existing results and reduces the problem to that of choosing signs. Some of consequences of the lemma are given in Section 6.3. In section 6.4 we consider the case of maximum-norm which has many applications. In the concluding section 6.5 we give an algorithmic result of finding an appropriate permutation σ for a general norm, given a sign-algorithm. The algorithm runs in a polynomial time, provided that the sign algorithm runs in a polynomial time.

6.2 Setting Of The Problem And Transference Lemma

Let X be a normed space, $x = (x_1, ..., x_n), x_1, ..., x_n \in X$. Denote

$$|||x_{\pi}||| = \max_{k \le n} ||x_{\pi(1)} + \ldots + x_{\pi(k)}||,$$

where $\pi: \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a permutation of indices $\{1, \ldots, n\}$.

The main problem (on compact vector summation) we study in this paper is as follows.

PROBLEM: Given elements $x_1, ..., x_n$ from a normed space X

(i) Estimate $\min_{\pi} |||x_{\pi}|||$.

(ii) Find a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ ensuring the estimate in (i).

We will study both theoretical and algorithmical aspects of Problem. The following lemma plays a crucial role in solving the problem and in algorithmic construction of an appropriate permutation π .

For a permutation $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ we write $\pi = (k_1, \ldots, k_n)$, where $k_i = \pi(i), i = 1, \ldots, n$. The permutation $\pi_b = (k_n, k_{n-1}, \ldots, k_1)$) is called the *backward* permutation. Given an ordered collection of vectors $x = (x_1, \ldots, x_n) \subset$ X we put

$$|||x_{\pi}\theta||| \equiv \max_{1 \le k \le n} ||x_1\theta_1 + \ldots + x_k\theta_k||;$$

 $s = -\sum_{i=1}^{n} x_i$; s, x and x, s will stand for the collections (s, x_1, \ldots, x_n) and (x_1, \ldots, x_n, s) respectively.

Lemma 6.2.1. Let X be a normed space, $x_1, ..., x_n \in X$. Then for any permutation $\pi : \{1, ..., n\} \rightarrow \{1, ..., n\}$ and any collection of signs $\theta = (\theta_1, ..., \theta_n), \ \theta_i = \pm 1, \ i = 1, ..., n$

$$|||x_{\pi}||| + |||x_{\pi_{b}}\theta||| \ge 2|||x_{\pi^{*}}||| - ||s||, \qquad (6.2.1)$$

where $\pi^* = \pi(k_l^+), \pi(k_{l-1}^+), \dots, \pi(k_1^+), \pi(k_l^-), \pi(k_2^-), \dots, \pi(k_m^-), \text{ and } k_1^+ < k_2^+ < \dots < k_l^+$ are all the indices with $\theta_{k_1^+} = \theta_{k_2^+} = \dots = \theta_{k_l^+} = +1$, while $k_1^- < k_2^- < \dots < k_m^-$ are all the indices with $\theta_{k_1^-} = \theta_{k_2^-} = \dots = \theta_{k_m^-} = -1$.

Proof. Using the fact that |||x||| enjoys the properties of a norm in the space of sequences, we get by the triangle inequality for $||| \cdot |||$

$$\begin{aligned} |||x_{\pi}||| + |||x_{\pi_{b}}\theta||| &= |||x_{\pi}, s||| + |||x_{\pi_{b}}\theta||| &= |||s, x_{\pi_{b}}||| + |||x_{\pi_{b}}\theta||| \\ &\geq \max(|||s, 2x_{\pi_{b}}^{+}|||, |||s, 2x_{\pi_{b}}^{-}|||) &= |||s, 2x_{\pi_{b}}^{+}, 2x_{\pi}^{-}, s||| &= |||s, 2x_{\pi_{b}}^{+}, 2x_{\pi}^{-}||| &\geq \\ &2|||x_{\pi_{b}}^{+}, x_{\pi}^{-}||| - ||s|| &= 2|||x_{\pi^{*}}||| - ||s|| . \end{aligned}$$

Lemma is proved.

The next lemma converts our rearrangement problem into a well-known combinatorial problem on choosing signs.

Lemma 6.2.2 (Transference Lemma: An extension of a Transference Lemma proved in [16]). Let X be a normed space, $x_1, ..., x_n \in X$, $s = -\sum_{1}^{n} x_k$.

(i) Then there exists a permutation $\sigma : \{1, ..., n\} \rightarrow \{1, ..., n\}$ such that for any $\theta = (\theta_1, ..., \theta_n)$, $\theta_i = \pm 1, i = 1, ..., n$ the following inequality holds:

$$|||x_{\sigma}||| \leq |||x_{\sigma_b}\theta||| + ||s||,$$
 (6.2.2)

where σ_b stands for the backward permutation, where indices go in the opposite
direction. As the permutation σ one can take the optimal permutation, the one at which $|||x_{\pi}|||$ attains its minimum.

(ii) If $\sum_{k=1}^{n} x_k = 0$, then for the optimal permutation σ and any $\theta = (\theta_1, \dots, \theta_n)$, $\theta_i = \pm 1, i = 1, \dots, n$

$$|||x_{\sigma}||| \le |||x_{\sigma}\theta|||. \tag{6.2.3}$$

Proof. (i) follows from Lemma 6.2.1, if we take as π in (6.2.1) the optimal permutation σ . (ii) follows from (i).

Remark. Part (ii) of Lemma 6.2.2 has been proved in [16]. A straightforward use of (6.2.3) along with the triangle inequality would lead to a worse version of (6.2.2):

$$|||x_{\sigma}||| \leq |||x_{\sigma}\theta||| + 2||s||$$

that worsens related algorithmic estimates.

6.3 Application of Transference Lemma to Problem. An estimation of $\min_{\pi} |||x_{\pi}|||$

Theorem 6.3.1. For any collection (x_1, \ldots, x_n) from a normed space X there exists a permutation $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ such that

$$|||x_{\sigma}||| \leq \mathbf{E} \max_{k \leq n} \|\sum_{1}^{k} x_{\sigma(i)} r_{i}\| + \|\sum_{1}^{n} x_{i}\| \leq 2\mathbf{E} \|\sum_{1}^{n} x_{i} r_{i}\| + \|\sum_{1}^{n} x_{i}\|, (6.3.1)$$

where r_1, \ldots, r_n are Rademacher random variables (independent and taking two values -1 and +1 with probability 1/2), and **E** is the operation of expectation.

Proof. It is enough to average both sides the inequality (6.2.2) of Lemma 6.2.2 with respect to all collections of signs $\theta = (\theta_1, \ldots, \theta_n)$ and then use the Levy inequality.

Remark. In fact Lemma 6.2.2 gives a stronger result (see [16]), namely

$$\frac{1}{n!}\sum_{\pi}|||x_{\pi}||| \leq 2\mathbf{E}||\sum_{1}^{n}x_{i}r_{i}|| + ||\sum_{1}^{n}x_{i}||.$$

In other words, (6.3.1) holds for a typical permutation σ .

In the case of l_2 -norm (or, more general, for type-2 norms) the last inequality implies the following:

$$\frac{1}{n!} \sum_{\pi} |||x_{\pi}|||^{2} \le C \sum_{i=1}^{n} ||x_{i}||^{2} + ||\sum_{i=1}^{n} x_{i}||^{2},$$

where C depends only on the norm. The inequality goes back to [22] where it has been stated for the scalar case.

To our knowledge there is no any estimation for $\min_{\pi} |||x_{\pi}|||$ that is better than (6.3.1) for an infinite-dimensional X.

In the rest of this section and in section 6.4 we will be finding estimations for $|||x_{\sigma}|||$ for an appropriate σ through Transference Lemma and a known estimation of $|||x\theta|||$ for an appropriate collection of signs θ . Such a well-known estimation of $|||x\theta|||$ is the following.

Theorem 6.3.2. ([18], generalization by [24]). Let X be a normed space of dimension $d, x_1, \ldots, x_n \in X$, $||x_i|| \leq 1$, $i = 1, \ldots, n$. Then there exists a collection of signs $\theta = (\theta_1, \ldots, \theta_n)$ such that

$$|||x\theta||| \leq 2d.$$

Corollary 6.3.3. If $x_1, \ldots, x_n \in X$, dim X = d, $||x_i|| \le 1$, $i = 1, \ldots, n$, then there is a permutation $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ such that

$$|||x_{\sigma}||| \le 2d + ||x_1 + \ldots + x_n||.$$
(6.3.3)

Remark. The inequality (6.3.3) is the famous Steinitz (1916) inequality. It can be proved in a straightforward manner, without appealing to Lemma 6.2.2 (see [33]). We believe, however, that the proof using Lemma 6.2.2 is simpler.

6.4 The case of l_{∞}^d

This is the d-dimensional space \mathbb{R}^d with the maximum-norm. This case is most important for applications in scheduling theory. For low dimensions the general estimate (6.3.3) is the best among known ones. However, in most applications to scheduling theory the dimension d is much larger than n. Instead of using the Dvoretzki-Hanani theorem in this case we exploit the following assertion.

Theorem 6.4.1. If $x_1, \ldots, x_n \in l_{\infty}^d$, $||x_i|| \leq 1, i = 1, \ldots, n$, then for the Rademacher random variables r_1, \ldots, r_n

(i)

$$\mathbf{E}||\sum_{k=1}^{n} x_k r_k||^2 \le Cn \ln d,$$

where C is an absolute constant;

(ii) $\frac{1}{2^n} \sum_{\theta} |||x\theta||| \le C_1 \sqrt{n \ln d}$, where C_1 is an absolute constant; (iii) There exists a θ such that

$$|||x\theta||| \le C_1 \sqrt{n \ln d} \tag{6.4.1}$$

Proof. (i) Denote $X^{(k)} = \sum_{i=1}^{n} x_i^{(k)} r_i, k = 1, \dots, d$. Using the integration by parts formula we get for any $\delta > 0$

$$\mathbf{E}||\sum_{1}^{n} x_{i}r_{i}||^{2} = \mathbf{E}\max_{1 \le k \le d} |X^{(k)}|^{2} = \int_{0}^{\infty} P(\max_{1 \le k \le d} |X^{(k)}| > t)dt^{2} = \int_{0}^{\delta} |dt^{2} + \int_{\delta}^{\infty} |dt^{2} \le \delta^{2} + \int_{\delta}^{\infty} \sum_{k=1}^{d} P(|X^{(k)}| > t)dt^{2} \le \delta^{2} + d\int_{\delta}^{\infty} P(|Y| > t)dt^{2},$$

where Y is the $X^{(l)}$ for l where $\int_{\delta}^{\infty} P[|X^{(k)}| > t] dt^2$ attains its maximum. Now using the following well-known estimation for the tail probability

$$P(|Y| > t) \le 2 \exp\{-\frac{t^2}{2\sum_{1}^{n} y_k^2}\},\$$

where $Y = \sum_{1}^{n} y_{i} r_{i}$, we get

$$\mathbf{E} ||\sum_{1}^{n} x_{i} r_{i}||^{2} \leq \delta^{2} + d \int_{\delta}^{\infty} 2e^{-\frac{t^{2}}{2n}t} dt = \delta^{2} + 2dn e^{-\frac{\delta^{2}}{2n}}.$$

Putting $\delta^2 = 2n \ln 2d$ we find

$$\mathbf{E} || \sum_{i=1}^{n} x_i r_i ||^2 \le n(1+2\ln 2d).$$

This proves (i).

(ii) is a consequence of (i). Indeed,

$$\frac{1}{2^n} \sum_{\theta} |||x\theta||| = \mathbf{E} |||xr||| \le (\mathbf{E} |||xr|||^2)^{1/2} \le (2\mathbf{E} ||\sum_{1}^n x_k r_k||^2)^{1/2} = \sqrt{2}C^{1/2}\sqrt{n\ln d}$$

(iii) is an immediate consequence of (ii).

Remark. 1. Theorem 6.4.1 (iii) has been proved first by use of combinatorial methods by Spencer [53]. Statement 6.4.1 (ii) shows that not only does the desired θ exist, but (6.4.1) holds for a typical θ .

2. In another paper Spencer [54] has proved a surprising result saying that in the case of d = n there is a collection of signs $\theta = (\theta_1, \ldots, \theta_n)$ such that $|||x\theta||| \le C\sqrt{d}$, where C is an absolute constant. However, as the following example shows, in the case of arbitrary d and n Theorem 6.4.1 (iii) cannot be improved.

Example. Consider a matrix whose rows are all possible sequences of length n consisting of +1's and -1's. Our vectors x_1, \ldots, x_n in l_{∞}^n are all the columns of the

matrix. If one of the columns is multiplied by -1, we still get a matrix consisting of all different ± 1 -sequences of length n, and in particular there is a sequence consisting of +1's only. This implies that $|||x\theta||| = n$ for any $\theta = (\theta_1, \ldots, \theta_n)$. In this example $d = 2^n$, and $|||x\theta||| = n = \sqrt{n \ln d}$ which up to a constant coincides with (6.4.1).

The application of the transference lemma leads to the following assertion.

Theorem 6.4.2. If $x_1, \ldots, x_n \in l_{\infty}^d$, $||x_i|| \leq 1, i = 1, \ldots, n$, then there is a permutation $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ such that

$$|||x_{\sigma}||| \leq C\sqrt{n \ln d} + ||x_1 + \ldots + x_n||,$$

where C is an absolute constant.

6.5 An algorithm for finding σ

In all previous results concerning estimation of $\min_{\pi} |||x_{\pi}|||$ (Theorem 6.3.1, Corollary 6.3.3, Theorem 6.4.2) we stated only the existence of a permutation σ ensuring the corresponding estimation. In this section we give a constructive way of finding σ provided that the respective (much simpler) sign-problem possesses an algorithmic solution.

Let us show first that a greedy algorithm (that seems to be natural) is not, in general, the best.

Given vectors $x_1, \ldots, x_n \in X$ a greedy algorithm chooses at each step a vector that minimizes the norm of the next partial sum. In other words, on step 1 it chooses an element x_{n_1} that has a minimal norm. On step 2 it selects an element $x_{n_2}, n_2 \neq n_1$ such that $||x_{n_1} + x_{n_2}|| \leq ||x_{n_1} + x_{n_k}||$ for any $k \neq n_1$, etc.

The following simple example constructed by Jakub Wojtaszchik (oral communication) shows that a greedy algorithm is not in general the best one even in the 2-dimensional case.

Example. Consider *n* groups of vectors each consisting of the three following vectors: (1, 1), (2, -3), and (3, -2). Obviously, the greedy algorithm chooses at the first *n* steps the vectors $(1, 1), (1, 1), \ldots, (1, 1)$. Therefore, if π_g is the greedy

permutation, then $|||X_{\pi g}||| \ge n$. In fact, $|||x_{\pi g}||| = n + 2$. While the optimal algorithm arranges the vectors as follows: (1,1), (2,-3), (-3,2), (1,1), (2,-3), (-3,2),

Consider now a collection $x_1, \ldots, x_n \in X$, $||x_i|| \le 1, i = 1, \ldots, n$, where X is a general normed space. Assume that for any permutation π there is a collection of signs $\theta = (\theta_1, \ldots, \theta_n)$ such that $|||x_{\pi}\theta||| \le D$ for some D > 0, and that the desired θ can be found algorithmically. This is the case for vectors involved in Theorem 6.3.2 and Theorem 6.4.1.

Remark. For a general d-dimensional X such a polynomial sign-algorithm with D = 2d was constructed by [6], for $X = l_{\infty}^d$ such an algorithm with $D = \sqrt{2n \ln d}$ was described by Spencer [53]. Another algorithmic type result that uses a sign procedure was found earlier in [15]. An assertion similar to Theorem 6.5.1 was stated without proof by Makai [41] for the case s = 0. His arguments are based on Transference Lemma 6.2.2.

Below is the description of the permutation-algorithm.

We start with an arbitrary permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$. Then we find a θ satisfying $|||x_{\pi_b}\theta||| \leq D$, where π_b is the backward permutation (see Lemma 6.2.1). According to Lemma 6.2.1,

$$|||x_{\pi}||| + |||x_{\pi_{b}}\theta||| \ge 2|||x_{\pi_{1}}||| - ||s||$$
(6.5.1)

for some permutation π_1 . An easy algorithm determining π_1 is described in Lemma 6.2.1 and runs in O(n) time. Using (6.5.1) we get

$$|||x_{\pi_1}||| \le \frac{1}{2}(|||x_{\pi}||| + |||x_{\pi_b}\theta||| + ||s||) \le \frac{1}{2}(|||x_{\pi}||| + D + ||s||).$$
(6.5.2)

Then we find $\theta^{(1)}$ such that $x_{\pi_{1b}}\theta^{(1)} \leq D$ and a permutation π_2 such that

$$|||x_{\pi_1}||| + |||x_{\pi_1b}\theta^{(1)}||| \ge 2|||x_{\pi_2}||| - ||s||$$

In the same way as in (6.5.2) we get

$$|||x_{\pi_1}||| \le \frac{1}{2^2}|||x_{\pi}||| + \frac{3}{2^2}(D + ||s||).$$

After l iterations we get

$$|||x_{\pi_l}||| \le \frac{|||x_{\pi}|||}{2^l} + (1 - \frac{1}{2^l})(D + ||s||) \le D + ||s|| + \delta$$

where $\delta = \frac{1}{2l} |||x_{\pi}|||$. Noticing that $|||x_{\pi}||| \le n$, by an appropriate choice of l, we can make δ smaller than any fixed $\epsilon > 0$ (to get $\delta < \epsilon$, $l = [\ln \frac{n}{\epsilon}] + 1$ iterations are needed, where [x] denotes the integral part of x.)

Thus we have proved the following.

Theorem 6.5.1. Let X be a normed space, $x = (x_1, ..., x_n), x_1, ..., x_n \in X$, $||x_i|| \le 1$, i = 1, ..., n. Assume that for any permutation $\pi : \{1, ..., n\} \rightarrow \{1, ..., n\}$ there is an algorithm with a polynomial complexity to define $\theta = (\theta_1, ..., \theta_n)$ such that

$$|||x_{\pi}\theta||| \le D , \qquad (1)$$

where D does not depend on π . Then for any $\varepsilon > 0$ there exists an algorithm with a polynomial complexity to define σ such that

$$|||x_{\sigma}||| \le D + ||s|| + \varepsilon,$$

where $s = -\sum_{i=1}^{n} x_i$ The complexity of the algorithm is $O(C \cdot \log(n/\varepsilon))$, where C is the complexity of the sign-algorithm (for finding θ).

Chapter 7

Summary and Future Research

In this chapter we summarize findings described in previous chapters and also briefly specify future research directions that are suggested by these findings.

7.1 Summary

This dissertation proposes to consider a new quantity that describes the behavior of available bandwidth, which is called "crossing probability". Crossing probability is a likelihood that the available bandwidth time series drops below a certain quality-critical threshold M for the quality-critical time-period T. Both, the threshold and the time-period have to be defined by specification of a real-time task that has to be executed.

At the beginning we concentrated our efforts on estimating crossing probability with a parametric model that relies on observations of an available bandwidth obtained with the help of a Pathload measurement tool. We used Pathload for collecting available bandwidth measurements because our research showed that it was the most robust and accurate measurement tool among those available. We successfully applied the $ARCH^2$ model used in econometrics to modeling available bandwidth behavior. In particular, we combined two known estimation techniques used in econometrics thus obtaining our method for quick estimation of model parameters together with their confidence interval. We used known results from extreme values theory for computation of a crossing probability for arbitrary threshold Mand time-period T. Then we verified an efficiency of our approach on a number of Internet paths.

Before going any further with available bandwidth modeling, the dissertation discusses some work in improving the One Way Delay trend detection mechanism of present available bandwidth measurement tools. Instead of using empirically defined cutout thresholds for trend-detection metrics, we propose to detect trend based on the p-value of linear regression t-test conducted against slope of one way delay as a function of a packet sequence number.

We also suggested a simple way of dealing with the interrupt coalescence / context switch effect. Our experiments clearly indicated that both Type I and Type II errors of trend detection were significantly smaller with t-test than errors reported by extensively used before PCT/PDT test.

The next part of the dissertation deals with a non-parametric approach that relies on binary observations of $\{0, 1\}$, where 0 indicates that available bandwidth at a particular time-instance is higher than a specified critical threshold and 1 indicates that available bandwidth is lower than a critical threshold. Our non-parametric model gives an answer to the question: "What is the probability that among the next N observations of indicators will be present at least one "1"" in contrast to our parametric model that gives an answer to the question: "What is the probability that the minimal over next N observations value of an available bandwidth time series drops below the level of M".

With the non-parametric approach relying on $\{0, 1\}$ observations, one can predict what happens within the frames of the next couple of seconds while considering a history time-period of 10 minutes. The parametric approach that relies on slow available bandwidth measurement tools needs observations collected for the period of at least 2 hours for predicting of what happens within next 5 minutes.

In addition to introducing of a non-parametric model for a "crossing probability" we developed a self-verification procedure that combines predicted probabilities of realization of the event of a crossing and actual realizations/non-realizations of the mentioned event into a statistic that follows the standard normal distribution if predicted probabilities "match" on-coming realizations. We also present a theoretical proof of the fact that the proposed self-verification statistic really follows a standard normal distribution given that the prediction module works correctly.

The self-verification procedure opened further opportunities to adjust the prediction module to a particular network path. We output predictions of a crossing probability based on a number of sizes of a history time-frame used for a prediction. For an actual prediction we use the window size that is "confirmed" by a self-verification test.

Then we added weights to the crossing probability prediction. Even within the frames of a history window the most recent past inputs into prediction with a weight higher than the less recent past. We adjust the rate at which weights decay to a particular network path. Experiments with real-life data-sets of available bandwidth time-series confirmed that introducing weights significantly increases the range of history window-sizes valid for a crossing-probability prediction.

The last part of this dissertation is dedicated to the future research of possible actions that communicating parties may undertake if a crossing probability becomes unacceptably large. One of the promising scenarios of actions dedicated to supporting tele-operation under a degraded quality of service is the reallocation of part of the communication channels through different network paths with the help of external nodes-facilitators. In the context of such a reallocation arises an interesting and important "Reroute Sequence Planning (RSP)" problem. The problem chooses the optimal (or nearly optimal) sequence of channel reallocations (once they are planned) in a way that minimizes extra capacities required along the network for carrying out the reallocation procedure.

In the last part of this dissertation we develop a solid theoretical basis for the algorithm that finds a lower bound for an extra needed capacity (for carrying out reallocation) as well as actually finds in polynomial time a sequence of reallocations from old paths to new paths that fit the mentioned above bound.

A part of the dissertation was published in the frames of following references: [13,

12, 14]

7.2 Future research

7.2.1 QoS overlay network infrastructure

As it was described in Chapter 6 we see a promising direction in setting up an overlay infrastructure that helps in dynamic allocation-reallocation of communication channels through various network paths in a way that minimizes likelihood of a failure of remote real-time task execution. Chapter 6 makes a step towards setting up such an infrastructure. However, it is clear that there are still many unsolved directions to follow. Directions include algorithms for finding the best set of paths with respect to particular QoS-related probabilistic requirements, detecting an appropriate time-moment for triggering the reroute sequence procedure, finding the best reroute-sequence procedure with respect to probabilistic (but not hard-coded as in Chapter 6) QoS characteristics, and many others.

7.2.2 QoS model improvement

The Quality of Service model that was introduced in Chapter 3 assumes that realtime task can be specified with some fixed value of a rate (bandwidth) that the task requires for a safe execution. For most tele-operations, however, the vast majority of tele-operation traffic is constituted by a real-time video. Traffic associated with video transfer may be bursty. Various encoding formats, for example, send only change from the current video frame to the previous one when camera moves slowly and there are no significant scene changes (B-frames). If a rapid scene change takes place then the whole newly-captured frame has to be sent at once, which causes bursts in video-traffic. It may not be efficient to reserve bandwidth for a video-traffic at such a "burst-level" because for most of the time the high bandwidth will remain unused. On the other hand, reservation of bandwidth at lower levels may lead to unexpected degradation of service. Therefore a precisely estimated "crossing probability" as it was introduced in Chapter 3 may not do a good job for estimating the likelihood of an actual degradation of service during the execution of a remote real-time task. Under a more complex model, the likelihood of degradation of service for the period of time T can be viewed as a probability that the traffic associated with remote real-time task execution exceeds the available bandwidth time-series at any moment during the critical period of time T. In order to estimate such a likelihood, one has to model traffic associated with the task in addition to modeling of available bandwidth time-series. Liu and Mao [38] analyze various prediction algorithms for real-time video traffic and introduce another prediction algorithm. One of our goals therefore is to unify results presented in [38] with our results in available bandwidth estimation and come out with a more robust model for estimation of a likelihood of a failure of remote real-time task.

We conclude this dissertation with a hope that the introduced parametric and non-parametric approaches to modeling of an available end-to-end bandwidth together with the improvement of present OWD trend detection mechanism and novel self-verification / self adjustment module will facilitate the extension of teleoperations from "traditional" isolated and fully controlled lines to inexpensive and easily accessible Internet. BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Auckland-ii trace archive. "http://pma.nlanr.net/Traces/long/auck2.html".
- [2] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. In 18th ACM Symposium on Operating Systems Principles (SOSP), 2001.
- [3] Javier Aracil, Richard Edell, and Pravin Varaiya. A phenomenological approach to internet traffic self-similarity. In 35th Annual Allerton Conference on Communication, Control and Computing, 1997.
- [4] M. Arlitt, R. Friedrich, and T. Jin. Workload characterization of a web proxy in a cable modem environment. In ACM Performance Evaluation Review, 1999.
- [5] M. Arlitt and T. Jin. A workload characterization study of the 1998 world cup web site. In *IEEE Network*, 2000.
- [6] I. Barány and V.S. Grinberg. On some combinatorial questions in finitedimensional spaces. *Linear Algebra and its Applications*, 41:1–9, 1981.
- [7] J. Beran, M. Sherman, S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. In *IEEE Transactions on Communications*, 1995.
- [8] M. Borella and G. Brewster. Measurement and analyses of long-range packet dependent behavior of internet packet delay. In *IEEE INFOCOM*, 1998.
- [9] Arup Bose and Kanchan Mukherjee. Estimating the ARCH parameters by solving linear equations. *Journal of time series analysis*, 24(2):127–136, 2002.
- [10] J. Bourgain. \bigwedge_p -sets in analysis: results, problems and related aspects. Handbook of the geometry of Banach spaces, I:195-232, 2001.
- [11] Peter J. Brockwell and R.A. Davis. *Time Series: Theory and Methods*. Springer, New York, 1991.
- [12] Alexander Chobanyan, Matt Mutka, Zhiwei Cen, and Ning Xi. One way delay trend detection for available bandwidth measurement. In *IEEE GLOBECOM* 2005, 2005.
- [13] Alexander Chobanyan, Matt Mutka, Shlomo Levental, and Ning Xi. Behavior of available end-to-end bandwidth: Non-parametric approach. In International Conference on Quantitative Evaluation of SysTems (QEST), 2006.

- [14] Alexander Chobanyan, Matt Mutka, V.S. Mandrekar, and Ning Xi. Modeling available bandwidth for an efficient qos characterization of a network path. In 4th IFIP-TC6 Networking Conference, Networking2005, 2005.
- [15] Sergei Chobanyan. The structure of the set of sums of a conditionally convergent series in a normed space. Mat Sb (N.S.), 128(170)(1):50-65, 1985.
- [16] Sergej Chobanyan. Convergence a.s. of rearranged series in Banach spaces and associated inequalities. Appeared in Probability in Banach spaces 9 edited by J. Hoffmann-Jorgensen and J. Kuelbs and M.B. Markus. Birkhauser, 1994.
- [17] M. Crovella and A. Bestarvos. Self-similarity in world-wide web traffic: Evidence and possible causes. In ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, 1996.
- [18] A. Dvoretzky and H. Hanani. Sur les changements des signes des termes d'une serie a termes complexes. C.R. Acad. Sci. Paris, 255:516–518, 1947.
- [19] Paul Embrechts, Claudia Kluppelberg, and Thomas Mikosch. Modelling Extremal Events for Insurance and Finance. Springer, Berlin Heidelberg, 1997.
- [20] R.F. Engle. Autoregressive conditional heteroscedasticity and estimates of variance of UK inflation. *Econometrica*, (50):987–1008, 1982.
- [21] M. Garrett and W. Willinger. Analysis, modeling, and generation of self-similar vbr video traffic. In SIGCOMM, 1994.
- [22] A. Garsia. Topics in almost everywhere convergence. Lectures in Advanced Mathematics. Markham Publishing Co., Chicago, 1970.
- [23] N. Giroux and S. Ganti. Quality of Service in ATM Networks. Prentice Hall, Upper Saddle River, NJ, 1999.
- [24] V.S. Grinberg and S.V. Sevastyanov. Value of the steinitz constant. Funktsional. Anal. i Prilozhen, 14:56–57, 1980.
- [25] I. Halperin and T. Ando. Bibliography: Series of vectors and Riemann sums. Hokkaido University, Sapporo, 1989.
- [26] H. Hotelling. Multivariate Quality Control. Techniques of Statistical Analysis. Eisenhart C, Hastay MW, Wallis WA, McGraw-Hill, New York, 1947.
- [27] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth techniques. In IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling, 2003.
- [28] Ibragimov and Linnik. Independent and stationary sequences of random variables. Wolters-Noordhoff, The Netherlands, 1971.

- [29] Manish Jain and Constantinos Dovrolis. Pathload: a measurement tool for available bandwidth estimation. In *Passive and Active Measurements Work*shop, 2002.
- [30] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. In ACM/IEEE Transactions on Networking, 2003.
- [31] Manish Jain and Constantinos Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Internet Measurement Conference*, 2004.
- [32] Balazs Gabor Jozsa and Marton Makai. On the solution of reroute sequence planning problem in mpls networks. Computer Networks: The International Journal of Computer and Telecommunications Networking, 42(2):199– 210, 2003.
- [33] M.I Kadets and V.M. Kadets. Series in Banach Spaces. Conditional and Unconditional Convergence. Birkhauser Verlag, 1997.
- [34] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.
- [35] P. Levy. Sur les series semi-convergents. Nouv. Ann. Math., 5:506-511, 1905.
- [36] Mingzhe Li, Mark Claypool, Robert Kinicki, and Jim Nichols. Characteristics of streaming media stored on the internet. Technical Report WPI-CS-TR-03-18, Computer Science Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, Massachusetts 01609-2280, May 2003.
- [37] Zhi Li and P. Mohapatra. QRON: QoS-aware routing in overlay networks. *IEEE Journal on Selected Areas in Communications*, 22(1):29–40, 2004.
- [38] Huabing Liu and Guoquiang Mao. Prediction algorithms for real-time variablebit video. In Asia Pacific Conference on Communications, 2005.
- [39] Regina Y. Liu and Kesar Singh. Moving Blocks Jacknife and Bootstrap Capture Weak Dependence. Appeared in Exploring the Limits of Bootstrap edited by LePage and Billiard. John Wiley Sons, New York, 1992.
- [40] X. Liu, K. Ravindran, B. Liu, and D. Loguinov. Single-hop probing asymptotics in available bandwidth estimation: A sample-path analysis. In ACM SIGCOMM conference on Internet measurement, 2004.
- [41] Marton Makai. Reroute sequence planning in telecommunication networks and compact vector summation. *Applied Mathematics and Computation*, 150(3):785–801, 2004.

- [42] Cao Le Thanh Man, Go Hasegawa, and Masayuki Murata. A new available bandwidth measurement technique for service overlay networks. In 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services, 2003.
- [43] B. Melander, M. Bjorkman, and P. Gunningberg. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In *Global Internet* Symposium, 2000.
- [44] B. Melander, M. Bjorkman, and P. Gunningberg. Regression-based available bandwidth measurements. In International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2002.
- [45] Jiri Navratil and R. Les. Cottrell. ABwE:a practical approach to available bandwidth estimation. In *Passive and Active Measurements Workshop*, 2003.
- [46] Hikori Nishikawa, Takuya Asaka, and Tatsuro Takashi. ABdis: Approach to estimate available bandwidth distribution using a multi-rate probe. In International Conference on Communication and Broadband Networking (ICBN04), 2004.
- [47] M. Peligrad and Q. Shao. Central limit theorem for weakly dependent random variables. *Journal of Theoretical Probability*, 1994.
- [48] Ravi Prasad, Manish Jain, and Constantinos Drovolis. Effects of interrupt coalescence on network measurements. In *Passive and Active Measurements Workshop*, 2004.
- [49] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. Pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and Active Measurements PAM workshop*, 2003.
- [50] V.J. Ribeiro, M. Coates, R.H. Riedi, S.Sarvotham, and R.G. Baraniuk. Multifractal cross traffic estimation. In *ITC specialist seminar on IP traffic Mea*surement, 2000.
- [51] John A. Rice. Mathematical Statistics and Data Analysis. Duxbury Press, Belmont, California, 1995.
- [52] S.V. Sevastyanov. On some combinatorial methods in scheduling theory. *Discrete Applied Mathematics*, 55:59–82, 1994.
- [53] J. Spencer. Balancing games. Journal of Combinatorial Theory, B23:68-74, 1977.
- [54] J. Spencer. Balancing vectors in max norm. Combinatorica, 6(1):55-65, 1986.
- [55] E. Steinitz. Bedingt konvrgente reihen und convexe systeme. Journal fur die Reine und Angewandte Mathematik, 146:68–111, 1916.

- [56] Jacob Strauss, Dina Katabi, and Frans Kaashoek. A measurement study of available bandwidth estimation tools. In *IMC*, 2003.
- [57] B. Tsybakov and N. Georganas. On self-similar traffic in atm queues: Definitions, owerflow probability bound, and cell delay distribution. In *IEEE/ACM Transactions on Networking*, 1997.
- [58] Sven Ubik and Antonin Kral. End-to-end bandwidth estimation tools. Technical Report WPI-CS-TR-03-18, CESNET, November 2003.
- [59] A.A. Weiss. Asymptotic theory for ARCH models: Estimation and testing. Econometric Theory, (2):31, 1986.
- [60] Moshe Zukerman, Timothy D. Neame, and Ronald G. Addie. Internet traffic modeling and future technology implications. In *IEEE INFOCOM*, 2003.

