# On the relationship between fundamental measurements in TCP flows

Richard G. Clegg, João Taveira Araújo, Raul Landa, Eleni Mykoniati, David Griffin, Miguel Rio
Department of Electrical and Electronic Engineering
University College London, UK
Email: {rclegg, j.araujo, rlanda, e.mykoniati, dgriffin, m.rio}@ee.ucl.ac.uk

***Abstract*—This paper considers fundamental measurements which drive TCP flows: throughput, RTT and loss. It is clear that throughput is, in some sense, a function of both RTT and loss. In their seminal paper Padyhe et al [1] begin with a mathematical model of the TCP sliding window evolution process and come up with an equation showing that TCP throughput is (roughly) proportional to $1/RTT\sqrt{p}$ where $p$ is the probability of packet loss. Their equation is shown to be consistent with data gathered on several links. This paper takes the opposite approach and analyses a large number of packet traces from well-known sources in order to create a data-driven estimate of the functions which relate TCP, loss and RTT. Regression analysis is used to fit models to connect the quantities. The fitted models show different behaviour from that expected in [1].**

## I. Introduction

The functioning of Transmission Control Protocol (TCP) is critical to the behaviour of the Internet. It controls congestion by "backing off" when congestion is detected. Usually this detection is via packet loss but some variants (beginning with Vegas [2]) use round trip time (RTT). TCP uses a sliding window for congestion which controls the amount of data "in flight" between the sender and the receiver. Throughput is then a function of this window size which depends on loss and RTT. It also seems likely that these quantities are not completely independent. Increased throughput may lead to increased loss and short flows may be dominated by RTT as the windows size is initially small.

In their classic paper, Padhye et al [1] begin with assumptions about how TCP works and derive a formula for TCP throughput for a single stream. They derive a number of formulae for how TCP depends on loss and RTT making various assumptions. The simplest formula they give for bandwidth $B(p)$ as a function of packet loss probability is [1] (formula 20):

$$B(p) = \frac{1}{RTT}\sqrt{\frac{3}{2bp}} + o(1/\sqrt{p}),$$

where $RTT$ is the round trip time, $p$ is the probability of packet loss and $b$ is a TCP parameter. For mathematical tractability a number of simplifying assumptions are made e.g. RTT is constant for the connection and $p$ is constant and independent for every packet. The model assumes TCP Reno although many other flavours are used in modern networks.

This paper takes the "data driven" approach, in some ways the opposite to model building. The research begins with the data and from the data attempts to find those equations which best explain the observations. The aim of this paper is to fit equations in the form $T = \beta_0 x^{\beta_1} y^{\beta_2} \cdots$ where $T$ is the throughput of the flow, $x$ and $y$ are observations which might affect this (e.g. loss or RTT) and $\beta_i$ are parameters to be fitted.

The data sets are described in section II and the reconstruction of TCP streams, filtering and processing applied are described in section II-A. The models and results of the fitting are described in section III. Finally, section IV gives conclusions and future work.

### A. Background

Estimation of quantities fundamental to Internet protocols have aroused much research interest. RTT is important for applications such as anycast services [3] and content delivery overlays [4]. Vivaldi [5] is a well known system that estimates RTT by embedding end hosts in a 2+1D coordinate system in order to produce estimates of delays between arbitrary host pairs. iPlane and iPlane nano [6], [7], create an "atlas" of the Internet based on approximate routing paths known as BGP atoms [8]. It builds up a database of estimations of RTT and traffic levels between each of a number of intermediate points and then attempts to reconstruct which paths two end hosts will take to reach each other and the qualities of those paths.

Various papers have taken the model based approach to explaining flow characteristics begun with [1]. For example, [9] extends [1] to the slow start phase of TCP and [10] relaxes some assumptions made about packet losses. In [11] a fluid flow approximation to TCP flow is used to study active queue management.

"Data-driven" approaches have been taken before. In [12] the authors look at flow duration and flow rate distributions and note a strong correlation between the size of a flow and its bandwidth. The same correlation is noted in [13] and attributed to timeout mechanisms for small/medium sized flows – the authors classify flows according to size (elephant/mouse), duration (tortoise/dragonfly), and then analyse correlations between these classes.

Related work has also tried to predict performance. The sender and receiver have been used to predict RTT and jitter [14]. Forecasting traffic on an Internet link is studied in [15], [16]. End-to-end performance forecasting is attempted in [17] but the data for validation is limited. Lakshman and Madhow [18] look at how several TCP flows compete using a mathematical model of packets entering a single bottleneck

with different TCP flavours and a single bottleneck. The results are a good match to real data.

## II. Data preparation and filtering

This paper relies on analysis of large numbers of passive traces. Two sources are used, MAWI and CAIDA – all of the data analysed in this paper is publicly accessible.

The CAIDA OC48 Traces Dataset from 2002 were used[1]. These data are from 14th of August 2002. 24 traces were used from 16:00 UTC to 19:00 UTC with each trace being 5 minutes long. Overall this data set consists of 1.42 billion packets originally containing 876GB of data.

The CAIDA Anonymised 2011 and 2012, OC192 Internet Traces were used[2]. The 2012 data are 29 traces from Equinix to San Jose on 19th January 2012 from 13:00 to 13:29 UTC with each trace being 1 minute long. This data set consists of 1.58 billion packets and 1.12TB of data. The 2011 data are collected on two separate days at the same site. 26 traces are used from 20th January 2011 from 12:59 to 13:25 UTC – this will be known as OC192 2011 A and consists of 1.3 billion packets originally containing 662GB of data. 14 traces are used from 17th February 2011 from 13:00 to 13:14 UTC – this will be known as OC192 2011 B and consists of 927 million packets originally containing 582 GB of data.

MAWI provides data from 1999 onwards on a Japanese network connecting universities and research institutes http://mawi.wide.ad.jp/. The data consist of the traffic captured for 15 minutes every day and has been used for long-term data analysis [19]. To get a sample which spans a long time period, in this paper data are sampled from the 15th day of every month from October 2006 to December 2012. These are 63 snapshots each 15 minutes long containing 1.36 billion packets originally containing 982 GB of data. This choice was made (rather than selecting adjacent days) to examine the dynamics in data over a long period. Over this period the network had several changes and upgrades.

### A. Data processing

The techniques used to get estimates for per-flow loss, throughput and RTT are relatively commonly used. Multiple pcap files are used from each data source and the processed results aggregated. RTTs can be inferred in two ways. Firstly, if both directions of data are seen then the time between a SYN/SYNACK/ACK triple handshake gives an estimate. This also works even if only one direction of data can be seen provided the SYN and ACK are present. Secondly, if data flows in both directions on the connection then the time between points A and B can be inferred from the measurement point M. If both directions can be seen and some data is sent in both directions then the time M to A to M can be measured by considering the time from a packet seen at M going to A to receive and ACK. The time from M to B to M can be estimated in the same way and the two added together. The average of the SYNACK RTT (first method) and the median data RTT (second method) was used here.

For data flowing from A to B two separate types of loss must be considered. The first case is when a data packet (or its ACK) is lost after the measurement point M. In this case the packet will be seen twice at the measurement point, the loss is inferred from this retransmission. The second case is when the packet is lost before the measurement point. In this case, the packet will be retransmitted and hence seen out of order at the measurement point. Retransmit loss plus out of order loss divided by the number of data packets is the estimate used for the proportion of loss. The loss proportion is defined as the number of losses detected divided by the total number of packets.

In some cases flows are not symmetrical and the data path is not the same as the return path. If some packets are not seen in both directions then the RTT cannot be extracted except with the SYNACK method. It may also be the case that loss cannot be properly estimated if the "wrong side" of the connection is seen. To avoid such issues, flows with packets captured only for one direction have been filtered out – in most cases this is a noticeable but small percentage of the total data. However, in the case of the CAIDA OC192 traces the largest proportion of the data is rejected in this way.

In order to avoid effects caused by the truncation of flows due to flow end, flows are also removed if they have any packets within 2 seconds of the end of a trace – this is typically a small but significant proportion of flows (for example, 1.24% of flows in the OC48 data but as high as 5.26% in the OC192 2012 data).

## III. Delay and throughput statistics

Before fitting the models it is useful to look at the quantities we are planning to fit and how they relate. The quantities can be plotted in pairs as "heatmaps" (or 2D histograms) with, for example, RTT on one axis and throughput on another. The axes are split into logarithmic bins over the range of the data (except when loss is on an axis in which case it is linear). Each bin is coloured according to the number of flows which fall into this category.

Figure 1 shows results from the three different data sets for different parameter pairs. All data sets show the same pattern when comparing the same two parameters. Figure 1(top) shows the relationship between throughput and the number of packets in a flow. Since the early part of the flow is "slow" (because of the cautious initial window size of TCP) it should be no surprise that a correlation between number of packets and throughput is seen for small flows. However, it seems that this correlation continues even for flows with many packets. Noting the logscale on this diagram the relationship seems to continue for much longer than might be expected, seemingly beyond the first 1,000 packets of the flow. Figure 1(middle) shows the correlation between throughput and RTT. This shows the expected connection that increased RTT is correlated with decreased throughput. Figure 1(bottom) shows

[1]Colleen Shannon, Emile Aben, kc claffy, Dan Andersen, Nevil Brownlee http://www.caida.org/data/passive/passive_oc48_dataset.xml

[2]kc claffy, Dan Andersen, Paul Hick http://www.caida.org/data/passive/passive_2011_dataset.xml

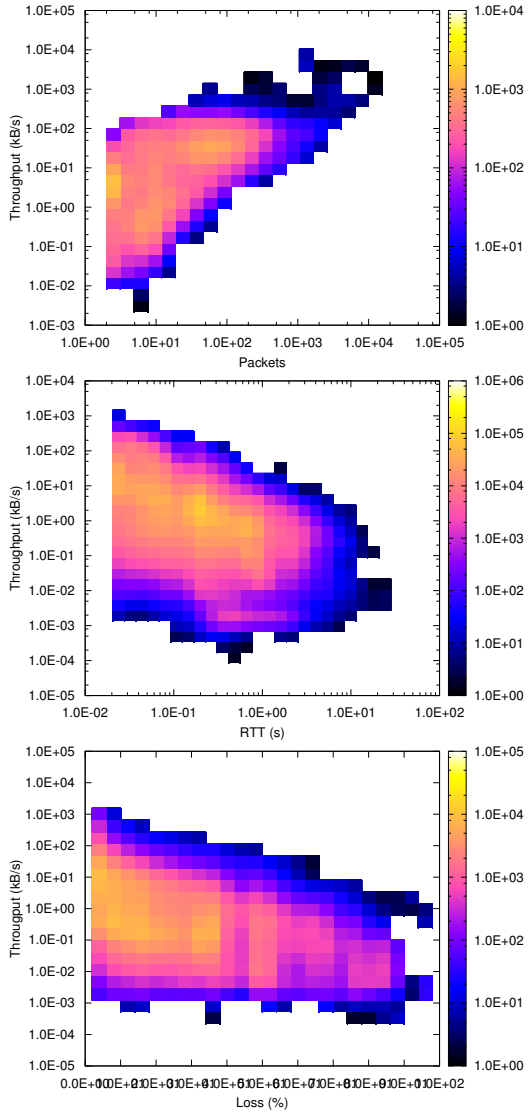


Fig. 1: Throughput against number of packets in a flow for OC192 2012 (top), throughput against RTT for OC48 (middle) and throughput against loss for MAWI (bottom).

the relationship between throughput and loss in the data. The apparent large proportion of files with extremely high loss is misleading. The OC48 data has an average of 5.65% loss which is high but the OC 192 2012 data has an average loss rate across flows of only 0.684%. Again a clear relationship can be seen in the diagram with, as expected, high loss correlated with low throughput. A plot of RTT versus number of packets in a flow (not shown due to space restrictions) shows the connection between RTT and the number of packets in the flow. No particular connection was expected between these quantities but a clear connection emerges. Flows with low RTT have more packets. This is not because flows are being terminated by the measurement cutting off (flows which cut off at the end of the measurement period are filtered). It may be that users terminate "slow downloads" of long files or that large files are more likely to be obtained from nearer caching servers (because the throughput is more important).

### A. *Fitted models*

Linear models assess relationships between observations of a variable to be modelled and several explanatory variables. The models fit parameters to given equations relating the variables (the equations need not be linear but the terms must be combined in a linear or log-linear way). These models are standard in a wide variety of fields for understanding the relationship between variables. So, for example $T = \beta_0 D^{\beta_1} p^{\beta_2} \varepsilon$ where $T$ is the throughput of the flow, $D$ is the delay ($RTT$) of the flow, $p$ is the proportion of packet loss in the flow and $\varepsilon$ is an lognormal error term with mean 1. The $\beta_i$ are the parameters the model fits to reduce the variance of the error term. The "goodness of fit" of such a model is judged by the $R^2$ (coefficient of variation) parameter which is in the range $(0, 1)$ with 1 indicating the data fits the equation perfectly.

The data sets are split into equal-sized, non-intersecting calibration, cross-validation and test data sets with each flow being randomly assigned to one and only one such set. This is a standard procedure in statistical modelling to allow a large number of models to be fitted but avoiding the possibility of over-fitting by creating models which apply only to that data. Initial model fitting to discover $\beta$ parameters is done with the calibration data. Fitting of exogenous parameters and comparison between these models is done with the cross-validation data. The final reported goodness of fit is given on the test data.

As is standard, the model fitting is done by performing a log fitting on a linear model $\log(T) = \log(\beta_0) + \beta_1 \log(D) + \beta_2 \log(p) + \log(\varepsilon)$ which is simply a transform of the original model. A problem arises with $p$ since many flows (the vast majority) have no loss at all. The fitting, however, is done on the logarithm of the quantities (to get the multiplicative model). To avoid the problem with $\log 0$ a constant term was added to $p$ – this value $p_m$ was fitted as a separate exogenous parameter by scanning a range of values and adding the one with the best $R^2$ value in the cross validation data. Note that the value of $p_m$ chosen was often at the extreme of the range used (hence a number of models have the same fitted $p_m$). Models were also fitted to large flows ($P > 1000$ only) to see if fit was improved by looking at only flows which had gone past the initial exponential growth phase. A comparison was also made by fitting $T = \beta_0/(D\sqrt{p + p_m})$ and this will be referred to as the base model as it is the closest model that can be fitted to the Padhye et al [1] model (with the $p_m$ term necessarily added to avoid the $p = 0$ problem). All results are given to three significant figures. In these results the $R^2$ values are always those on the test data whereas the $\beta$ parameters are fitted on the calibration data and the $p_m$ is optimised over the cross-validation data.

The OC192 2012 passive data set after processing has 4.47 million packets containing 3.68GB of data in 57.4 thousand flows. The loss rate is low at 0.684%. The following table shows the best models and base model with fitted parameters.

| Model for $T$ | $R^2$ | Note |
|---|---|---|
| $15.7D^{-0.94}(p+p_m)^{-0.563}P^{0.456}$ | 0.641 | $p_m = 0.105$ |
| $77.2D^{-0.975}P^{0.455}$ | 0.635 | |
| $316/(D\sqrt{p+p_m})$ | 0.0227 | $p_m = 0.105$ |

The best model showed $T$ had a relationship with $D$ and $p$ with nearly the expected coefficients $-0.94$ not $-1.0$ and $-0.563$ not $-0.5$ but also a relationship with length in packets $P$. The high $R^2$ shows this model is an excellent fit to the data but the model is almost as good if the dependence on loss $p$ is dropped. The base model without $P$ is a poor explanation of the data.

The MAWI data set is the largest analysed here and the data set is also the longest in time, spanning several years whereas the other data sets were continuous over minutes or hours. After processing the MAWI data set has 243 million packets containing 174GB of data in 5.04 million flows and a mean loss rate of 2.38% (although this changes greatly during the lifetime of the flow). The table below shows three fitted models.

| Model for $T$ | $R^2$ | Note |
|---|---|---|
| $0.15D^{-0.664}(p+p_m)^{-0.416}P^{0.635}$ | 0.282 | $p_m = 0.0132$ |
| $0.648D^{-0.583}P^{0.576}$ | 0.332 | $P > 1000$ |
| $111/(D\sqrt{p+p_m})$ | 0.0904 | $p_m = 0.105$ |

Perhaps because of its length and hence variation in time no models here were a particularly good fit. The best model relates RTT, loss and flow length in packets but has $R^2 = 0.282$. A better model exists of only long flows fitting against only RTT and length of flow. The base model has a particularly poor fit and low $R^2$.

The OC48 from 2002 is the oldest data set analysed. After processing it has 93 million packets and 48.9 GB of data, 2.85 million flows and a mean packet loss rate of 5.65%. This relatively high mean packet loss was a product of quite a congested network and some flows with very high loss.

| Model for $T$ | $R^2$ | Note |
|---|---|---|
| $102D^{-0.929}(p+p_m)^{0.391}P^{0.339}$ | 0.362 | $p_m = 0.105$ |
| $29.7D^{-0.89}P^{0.354}$ | 0.35 | |
| $193/(D\sqrt{p+p_m})$ | 0.207 | $p_m = 0.105$ |

The best fit model here is quite strange as the coefficient for $p$ is in the wrong direction. However, as has been previously noted $P$ and $p$ are correlated. Removing $p$ did not greatly reduce the fit of the model as assessed by $R^2$. The base case model had the highest $R^2$ of all the data sets.

The two OC192 data sets from 2011 are shown next. These are the same link on different days (approximately a month apart) and might be hoped to have similar fits to models. After processing, OC192 2011 A has 6.97 million packets in 26 minutes and OC192 2011 B has 6.31 million in only 14 minutes. Set A had 5.26 GB of data and B had 5.51 GB. Set A had 125 thousand flows and set B 66.1 thousand. The loss rate for A is 0.766% and for B is 0.43%. As can be seen, the two data sets are quite different in terms of traffic level with B having more traffic per unit time but less loss. The models for A are:

| Model for $T$ | $R^2$ | Note |
|---|---|---|
| $0.712D^{-0.665}(p+p_m)^{-0.661}P^{0.429}$ | 0.454 | $p_m = 0.105$ |
| $4.62D^{-0.698}P^{0.41}$ | 0.448 | |
| $251/(D\sqrt{p+p_m})$ | 0.109 | $p_m = 0.105$ |

The best fit model had the usual form, expected parameters and a middling $R^2$ value. Again removing the $p$ did not greatly harm the fit to the data and the base case model was a much worse fit. The models for B are:

| Model for $T$ | $R^2$ | Note |
|---|---|---|
| $21.5D^{-0.924}(p+p_m)^{-0.581}P^{0.419}$ | 0.616 | $p_m = 0.105$ |
| $156D^{-0.981}P^{0.386}$ | 0.611 | |
| $562/(D\sqrt{(p+p_m)})$ | 0.19 | $p_m = 0.105$ |

For OC192 2011 B, similar patterns are seen to the A data but the fit to data is much better. Again, the dependence on $p$ is not strong but the fit with flow length is.

In all cases the model form $T = \beta_0 D^{\beta_1}(p+p_m)^{\beta_2}P^{\beta_3}$ was the best fit even after accounting for the extra parameter (using adjusted $R^2$). In one case a subset of the data was a better fit. In every case removing the $p$ term did not greatly worsen the fit. It seems that the dependence on the packet length is a robust finding although the exact parameter size varies. Experimenting on only long flows to remove the effects of slow start did not remove this finding.

### B. *Evolution of parameters*

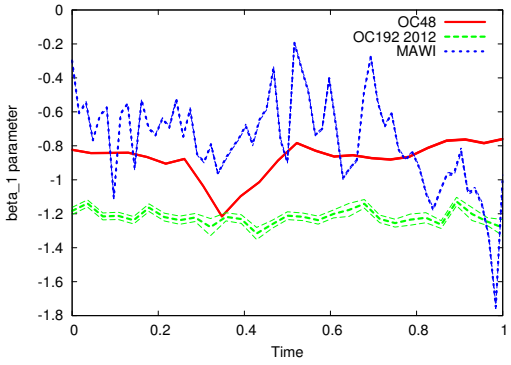


Fig. 2: The evolution of the $\beta_1$ parameter through time.

Finally it is interesting to investigate they dynamics of these data in the individual data files. Here the simple model $T = \beta_0 D^{\beta_1}$ is fitted for the data from individual pcap files. For the OC48 data each file represents 5 minutes of data, for the OC192 2012 data each file represents 1 minute and for the MAWI data each file represents 15 minutes but separated by one month. Figure 2 shows how the OC48 and OC192 2012 data vary through time. The time axis is normalised so that 0 is the first and 1 is the last data file plotted. The lines are printed with confidence intervals based on the standard error around them (for the OC48 and MAWI data these are too close to see). Model fits on the entire data $\beta_1 = -1.22 \pm 0.00439$ for the OC192, $-0.91 \pm 0.000804$ for the OC48 data and $-0.76 \pm 0.000812$ for MAWI. The parameter does not vary

much through the OC192 data but it is seen that the model estimate is based on the majority of the data having $\beta_1 > -0.9$ and a large excursion for several data files which move $\beta_1$ down to $-1.2$. This dynamic behaviour suggests that the model is not "universal" for this link, that is changes in traffic behaviour change how the model fits. The periods of decreased $\beta_1$ for OC48 corresponds to a sharp traffic increase (around 20%) but a decrease in mean packet loss (suggesting high throughput sources on low loss paths). The MAWI data is the most dynamic in its behaviour and this concurs with the idea that the parameter values may be "stable" for nearby time periods but vary over longer periods as the levels of traffic change (the OC48 and OC192 data are from subsequent minutes, the MAWI data is separated by a full month).

## IV. Conclusions

What can be learned from analysis of the relationships between RTT, loss, flow length and throughput? The data here only weakly supported the Padyhe et al relationship $T = 1/(D\sqrt{p})$, however, this should not be taken as a criticism of that paper as the contexts are so different. The value of $p$ from the Padhye model is the mean probability of a packet in a flow being lost which cannot be directly observed. Here this is represented by the observed proportion of loss (also callsed $p$ here) and an additive constant $p_m$ to avoid a zero problem. The Padhye model is only meant to fit for long flows – model fits were also tried on long flows only in this paper but the results were little different in most cases. This said, four of the five data sets fitted parameters for $p$ and $D$ close to $-0.5$ and $-1.0$ of the Padhye model. In the end though, the correlation between throughput and observed loss in flows was poor.

More important than the correlation with packets lost was the relationship with the flow length in packets $P$. In all data sets there was a strong relationship with values around $P^{0.5}$ and this was a much more important relationship than that with loss. This backs up the work of Zhang et al [12] which also notes that $T$ and $P$ are correlated but doesn't attempt to find a functional form. In fact, with the exception of the OC48 data the full fitted model form is markedly similar for the traces despite the fact that they are from different times on different equipment with different levels of congestion. For the relationship between only throughput, RTT and flow length in packets, all models had similar relationships. Looking at the dynamic evolution of the model parameters, it seems that the parameters are stable over the shorter term (minutes) and vary in the longer term (years) by a greater amount.

Despite the complex nature of the interactions being examined, extremely simple models can explain almost 2/3 of the variance in the data in the best case found here ($R^2 = 0.641$). With development this type of data driven explanatory model could provide great insights into the real behaviour of TCP in the wild. Future work could include other model components: TCP flavour, maximum window size and separate loss mechanisms. While the work in this paper came from the analysis of a large number of traces, the actual model fitting for the statistical models is lightweight and can be done in on the fly in real time. That is to say, a machine monitoring its own TCP connection could estimate model parameters and hence predict flow completion times very simply before a connection was started and improve this prediction while the connection was ongoing.

**Acknowledgements**: The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] in the ENVISION and FUSION projects under grant agreements 248565 and 318205.

## References

[1] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," *Proc. of ACM SIGCOMM*, pp. 303–314, 1998.

[2] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," in *Proceedings of the conference on Communications architectures, protocols and applications*, ser. Proc. of ACM SIGCOMM, 1994, pp. 24–35.

[3] E. Mykoniati, L. Latif, R. Landa, B. Yang, R. Clegg, D. Griffin, and M. Rio, "Distributed overlay anycast table using space filling curves," in *Proc. of the IEEE INFOCOM Global Internet Symposium*, 2009.

[4] S. Agarwal and J. R. Lorch, "Matchmaking for online games and other latency-sensitive p2p systems," in *Proc. of ACM SIGCOMM*, 2009, pp. 315–326.

[5] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris, "Practical, distributed network coordinates," *SIGCOMM Comp. Comm. Rev.*, vol. 34, no. 1, pp. 113–118, 2004.

[6] H. V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and A. Venkataramani, "A structural approach to latency prediction," in *Proc. of ACM Internet Meas. Conf.* New York, NY, USA: ACM, 2006, pp. 99–104.

[7] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane nano: Path prediction for peer-to-peer applications," in *Proc. of ACM NSDI*, 2009, pp. 137–152.

[8] A. Broido and kc claffy, "Analysis of routeviews bgp data: policy atoms," in *Network Resource Data Management Workshop*, Santa Barbara, CA, May 2001.

[9] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *Proc. of IEEE INFOCOM*, 2000.

[10] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," in *Proc. of ACM SIGCOMM*, 2000.

[11] Y. Liu, F. Lo Presti, V. Misra, D. Towsley, and Y. Gu, "Fluid models and solutions for large-scale IP networks," in *Proc. of ACM SIGMETRICS*, 2003, pp. 91–101.

[12] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of internet flow rates," *SIGCOMM Comp. Comm. Rev.*, vol. 32, no. 4, pp. 309–322, 2002.

[13] K.-c. Lan and J. Heidemann, "A measurement study of correlations of internet flow characteristics," *Comput. Netw.*, vol. 50, no. 1, pp. 46–62, 2006.

[14] S. Kaune, K. Pussep, C. Leng, A. Kovacevic, G. Tyson, and R. Steinmetz, "Modelling the internet delay space based on geographical locations," in *Proc. of the Euromicro PDP*, feb. 2009, pp. 301–310.

[15] P. S. P. Cortez, M. Rio and M. Rocha, "Topology aware internet traffic forecasting using neural networks," *Lecture Notes in Computer Science*, vol. 4669, pp. 445–454, Sep. 2007.

[16] K. Papagiannaki, N. Taft, Z. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic," *IEEE Trans. on Neural Nets*, vol. 16, no. 5, pp. 1110–1124, Sep. 2005.

[17] Q. He, C. Dovrolis, and M. Ammar, "On the predictability of large transfer TCP throughput," *SIGCOMM Comp. Comm. Rev.*, vol. 35, no. 4, pp. 145–156, 2005.

[18] T. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. on Networking*, vol. 5, no. 3, pp. 336–350, 1997.

[19] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho, "Seven years and one day: Sketching the evolution of Internet traffic," in *Proc. of IEEE INFOCOM*, 2009, pp. 711–719.