

How well can TCP infer network state?*

Dhiman Barman Ibrahim Matta
Computer Science Department
Boston University
Boston, MA 02215, USA
{dhiman, matta}@cs.bu.edu

May 16, 2003

BUCS-TR-2003-011

Abstract

The Transmission Control Protocol (TCP) has been the protocol of choice for many Internet applications requiring reliable connections. The design of TCP has been challenged by the extension of connections over wireless links. We ask a fundamental question: What is the basic predictive power of TCP of network state, including wireless error conditions? The goal is to improve or readily exploit this predictive power to enable TCP (or variants) to perform well in generalized network settings.

To that end, we use Maximum Likelihood Ratio tests to evaluate TCP as a detector/estimator. We quantify how well network state can be estimated, given network response such as distributions of packet delays or TCP throughput that are conditioned on the type of packet loss. Using our model-based approach and extensive simulations, we demonstrate that congestion-induced losses and losses due to wireless transmission errors produce sufficiently different statistics upon which an efficient detector can be built; distributions of network loads can provide effective means for estimating packet loss type; and packet delay is a better signal of network state than short-term throughput. We demonstrate how estimation accuracy is influenced by different proportions of congestion versus wireless losses and penalties on incorrect estimation.

Keywords: TCP; Congestion Control; Error Control; Binary Hypothesis Testing; Maximum Likelihood Ratio Test; Gaussian Distribution; Wireless Links; Simulation.

1 Introduction

Many studies have analyzed the performance of transport protocols, notably TCP [12]. TCP carries most

*This work was supported in part by NSF grants ANI-0095988, EIA-0202067 and ITR ANI-0205294, and by grants from Sprint Labs and Motorola Labs.

of the traffic—around 90% of the bytes—in the Internet [1]. TCP has been designed to do congestion control to achieve stable and fair allocation of resources within the network.

In a wired network, congested links cause packets to get lost when the bottleneck buffer overflows. If a TCP connection traverses a wireless link, for example a GSM cellular network, packets may be corrupted and get lost due to fading or shadowing. Such wireless losses are not an indication of resource scarcity in the routers and it is intuitive that an informed transport protocol would treat such packet losses differently. But for an end-to-end protocol, inferring the nature of loss without any aid from the network is challenging. Nevertheless, many proposals [15, 10] attempted to infer (implicitly or explicitly) the reason of a packet loss, in an end-to-end way, by analyzing measured delays, throughput or other metric.

Our approach is also end-to-end. We elucidate the difference in the output (measured) statistics under different type of losses, and exploit those using signal estimation techniques. It is to be disclaimed that we are *not* overlooking the better performance that may result from infrastructure support other than pure end-to-end solutions, e.g. XCP [14] and Snoop [3]. Such infrastructures have their own cost of deployment and may not be effective, for example with IPsec [17].

In distinguishing network state or the cause of packet loss, we exploit the temporal correlation between losses and the measured end-to-end metrics. Congestion-induced losses are associated with (close to) full buffer size at the bottleneck, whereas wireless losses often sample any queue size and associated delays. This leads to *distinguishable* distributions of the measured samples of network response at the times of different type of loss or network state.

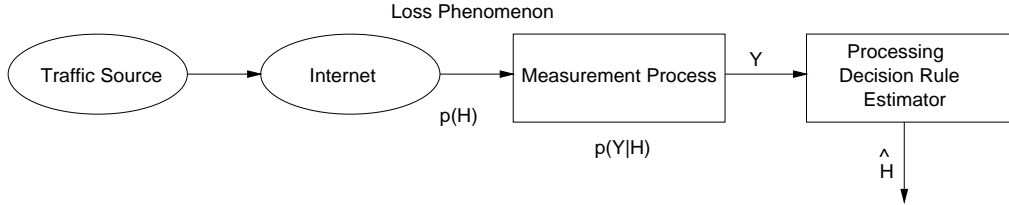


Figure 1. Detection problem components

Figure 1 shows our model where measured samples are noisy observations in the vicinity of the losses. Network conditions result in an output (e.g. packet loss due to congestion or wireless), which we denote by H and we call a *hypothesis*. This outcome generated by the network state is carried by packet samples after a certain time lag, thus the samples are probabilistically affected and serve as observation samples Y . Based on the observation samples, we intend to design a rule to decide what the cause of the loss is or what is the network state/condition. We assume knowledge of the apriori probability of a hypothesis, denoted by $p(H)$, and the probability distribution of the observed metric Y conditioned on the hypothesis, denoted by $p(Y|H)$.¹ Our goal is to obtain the best possible estimate, \hat{H} , that minimizes the average penalty of misclassifying the type of loss or network state. This would give us a handle on the theoretical limits and gains of end-to-end error classification.

To that end, we use Bayes Decision Rules and Maximum Likelihood Ratio Tests. The penalty function should measure the dissatisfaction of the application of its performance, and at the same time the function should be tractable. For example, if the network is congested and the protocol misclassifies a packet loss, i.e. the loss is not attributed to congestion rather to a wireless loss, this congestion loss misclassification may incur more cost than an otherwise wireless loss misclassification. This could be due to increased congestion as the source did not react appropriately (backed off) in response to original congestion. Therefore, it makes sense to map any observation to a hypothesis which will reduce the cost of classification error.

Using Bayes Rule, we have

$$P(H|Y) = \frac{P(Y|H)P(H)}{P(Y)} \quad (1)$$

From equation (1), it follows that if we know the probability of Y under some hypothesis H , the prior probabilities H and the unconditional probability of Y , we can derive the probability of a hypothesis from Y . In

¹Section 11 addresses how to obtain this assumed knowledge of network conditions. We leave the details to a future paper.

this model based approach, we thus attempt to obtain theoretical bounds on the predictive accuracy of TCP under additional knowledge of network conditions.

Our results point to several phenomena of practical interest:

- When the end-to-end loss rate is low to moderate (up to around 10%), noise in the measured delay signals used for detecting loss type (congestion versus random loss/perturbation) can be very accurately modeled by additive white Gaussian noise.
- In most scenarios, end-to-end delay is a vital observed metric and delay samples carry more information about loss type than short-term throughput.
- The short-term throughput distribution is log-normal. Changing load conditions do not change such distribution. Furthermore, we can infer the load or number of competing connections if the conditional distributions at different load levels are sufficiently distinguishable.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces Bayesian hypothesis testing for making a binary decision. Section 4 instantiates Bayesian binary testing for distinguishing between congestion-induced losses and wireless losses. The distinguishability of the delay distributions conditioned on the loss type is validated in Section 5 using ns-2 simulation [2]. Section 6 defines measures to evaluate the performance of the Bayesian binary detector. Section 7 evaluates the detector using delay samples as well as short-term throughput samples. Section 8 presents simulation results for the case of multiple bottlenecks and bursty wireless loss model. Section 9 extends Bayesian detection from 2-ary (binary) to the M-ary decision case. Section 10 validates the 3-ary case using ns-2 simulation. Finally, Section 11 concludes the paper.

2 Related Work

Many studies of TCP over wireless links have shown that goodput can be improved by identifying the cause of packet loss and taking appropriate actions accordingly [5]. While the research community is unsure about how TCP should modify its error control to react to packet loss of different types, many proposals suggest that TCP should refrain from congestion control actions in response to wireless losses. Other proposals suggest corrective measures at lower layers [6], but such measures need infrastructure support and may not be effective, for example with IPsec [17].

In this paper, we are primarily interested in evaluating the performance of end-to-end solutions under a given model through Maximum Likelihood Ratio estimation techniques. End-to-end estimation schemes differ in the performance measure(s) they use to infer network conditions and cause of loss. In [16], the authors claim that the congestion information carried by the round-trip time (RTT) samples is not sufficient to predict packet loss reliably. Although our approach uses two conditional RTT distributions, one around congestion losses and the other around wireless losses, we focus on the *distinguishability* of these two distributions. We then obtain *theoretical bounds* on the inherent power of TCP in predicting network conditions.

In another end-to-end technique [7], the authors summarize that congestion-avoidance based loss predictors (based on Vegas, delay gradient and throughput gradient) cannot perform better than a random coin predictor. According to them, a predictor would accurately estimate congestion losses if (a) congestion losses are preceded by long queue build up, (b) the queue build up results in losses, and (c) the loss predictor correctly senses the queue build up. In our evaluation, we show that the sample distributions conditioned on the type of loss is more informative in most cases than simple random coin flips.

Another approach attempts to understand the path characteristics by modeling it as a multi-state space Markov model. In [15], the authors have developed a technique based on the loss pair measurement technique and Hidden Markov Models (HMMs). They have used the intuition that the delay distribution around wireless losses is different from that around congestion losses. The classification of loss is done based on the state of the HMM which captures certain delay features. In our work, we present a simpler but effective detector based on the delay distributions. The detector does not need samples for training the HMM and can be used as a predictor/estimator by TCP for error control in real time. We also obtain bounds on the ef-

fectiveness of our Bayesian detector for different lossy paths and different values of classification / misclassification.

Lastly, some solutions have strived to abstract away loss nature by capturing the distinguishing features indirectly. For example, TCP Westwood [8] measures the available throughput of a connection and uses the estimated rate to control the sending rate of the TCP source. On the other hand, our approach in this paper separates error classification from error control, so we can isolate and evaluate the general error detection problem in which we may consider two or more hypotheses about the network state.

3 Bayesian Binary Hypothesis Testing

In this section, we use Bayesian binary hypothesis testing to infer the reason of packet loss. We consider the simplest classification—a packet loss is either due to congestion (i.e. buffer overflow) or due to wireless (i.e. transmission error). So we have two possible network states, which we label through hypotheses H_c , corresponding to “congestion loss hypothesis”, and H_w , corresponding to “wireless loss hypothesis.”

There is a probabilistic relationship between an observed metric y and a hypothesis H_c or H_w . The decision rule divides the space of possible observations into two disjoint regions, Z_c and Z_w , such that whenever an observation falls into Z_c (Z_w), the decision that H_c (H_w) is the correct hypothesis is made. These decision regions are established to maximize appropriate criterion of performance, corresponding to the probability of a correct decision.

More formally, in our approach we have three models: (i) a model of the network state, (ii) a model of the observations, and (iii) decision rules.

The model of the network state is captured by the prior probabilities, $P(H_c)$ and $P(H_w)$. The observation model captures the relationship between the observed quantity y and the unknown H_c or H_w by the conditional densities $p_{Y|H}(y|H_c)$ or $p_{Y|H}(y|H_w)$. Our decision rule $D(y)$ is obtained by minimizing the average cost (“Bayes risk”).

Let C_{wc} denote the cost of deciding that $D(y) = H_w$ when the hypothesis H_c is true (i.e., misclassifying congestion loss). Similarly, we denote by C_{cw} the cost of deciding that $D(y) = H_c$ when the hypothesis H_w is true (i.e., misclassifying wireless loss). Then the Bayes

risk of the decision rule is given by:

$$\begin{aligned}
E[C_{D(y),H}] &= C_{cw}P\{D(y) = H_c, H_w \text{ true}\} + \\
&\quad C_{wc}P\{D(y) = H_w, H_c \text{ true}\} \\
&= E[E[C_{D(y)}|y]] \\
&= \int E[C_{D(y)}|y]p_Y(y)dy \quad (2)
\end{aligned}$$

In our formulation, we make the following assumptions. First, we assume that we know the prior probabilities $p_{Y|H}(y|H_c)$ and $p_{Y|H}(y|H_w)$, that is, we know the distributions of the observed measure conditioned on loss type. Second, we assume knowledge of the a-priori probabilities, $P(H_c)$ and $P(H_w)$. Third, we assume that the penalty of misclassification of losses is constant and there is no penalty for correct classification.

From equation (2), we can minimize the penalty of misclassification by minimizing $E[C_{D(y)}|y]$ for each value of the observed sample, y . Thus, the optimal decision is to choose the hypothesis that yields the smallest value of the conditional penalty cost $E[C_{D(y)}|y]$ for a given value of y . The conditional expected penalty is given by:

$$\begin{aligned}
E[C_{D(y),H}|y] &= C_{cw}P\{D(y) = H_c, H_w \text{ true}|y\} \\
&\quad + C_{wc}P\{D(y) = H_w, H_c \text{ true}|y\} \quad (3)
\end{aligned}$$

For a given observation value y , the expected value of the conditional penalty if we choose to assign the observation to H_w or H_c is given by:

$$\text{If } D(y) = H_w : E[C_{D(y)}|y] = C_{wc}p_{H|y}(H_c|y) \quad (4)$$

$$\text{If } D(y) = H_c : E[C_{D(y)}|y] = C_{cw}p_{H|y}(H_w|y) \quad (5)$$

Given the above conditions, the optimal decision is one that results in the smaller of the two conditional costs. Using Bayes rule and reorganizing equations (4) and (5), we have:

$$\mathcal{L}(y) = \begin{cases} P(H_c)C_{wc}p_{Y|H}(y|H_c) & \begin{matrix} \geq \\ < \end{matrix} \\ P(H_w)C_{cw}p_{Y|H}(y|H_w) & \begin{matrix} \leq \\ > \end{matrix} \end{cases} \quad (6)$$

where $\begin{matrix} \geq \\ < \end{matrix}$ denotes choosing H_c if the inequality is $>$ and choosing H_w if the inequality is $<$. Henceforth, to simplify our notation, we simply use P_c to denote $P(H_c)$, and P_w to denote $P(H_w)$.

In the network, the packet samples carry the network state information to the receivers or senders delayed by propagation time. Moreover, those samples which make their way through the network are only

discrete samples of the network state. But due to temporal and spatial locality, we assume that samples received immediately around the time of loss have most energy and information of the state. This is also an objective of our expedition to know how effective are these samples.

4 Application of Detection in Transport Protocols

We are interested in applications which employ TCP or TCP-friendly transport protocols. Those transport protocols attempt to adapt to current network conditions, which can be estimated by measuring end-to-end packet delays, short-term throughput and packet losses.

During congestion-induced losses, delay samples carried by packets, which make it through the bottleneck queue, are expected to sample higher values of the queue size. Since it is difficult to measure the actual queue size at the time of a packet drop, the sample value is carried by a previous or following packet and is perturbed by cross-traffic behavior. Furthermore, if a path has a wireless link, the delay samples carried by packets preceding or following a packet lost due to wireless transmission errors will have different characteristics. In this case, the received samples are expected to experience a wider range of queue lengths. Figure 2 shows the average packet delays measured by TCP prior to a congestion-induced loss or a wireless loss. At every lag, the average packet delay conditioned on a wireless loss is almost the same as the unconditional average packet delay.

On average, all the samples before congestion losses should see an almost full bottleneck queue size.² Denote the corresponding average delay by $y \approx m_c$. Since cross traffic and the measurement process introduce noise, we see a perturbed sample value. We assume that noise is white Gaussian with zero mean and variance σ_c^2 .

Prior to wireless losses, the delay samples see a lower average delay. Denote the corresponding average delay by $y \approx m_w$. Again, we assume that noise is white Gaussian with zero mean and variance σ_w^2 .³ We can now pose the detection or classification of congestion versus wireless loss as a scalar Gaussian detection prob-

²How close to a full buffer depends on the behavior of the cross-traffic.

³The case of purely colored noise is both physically less relevant and mathematically more difficult [13].

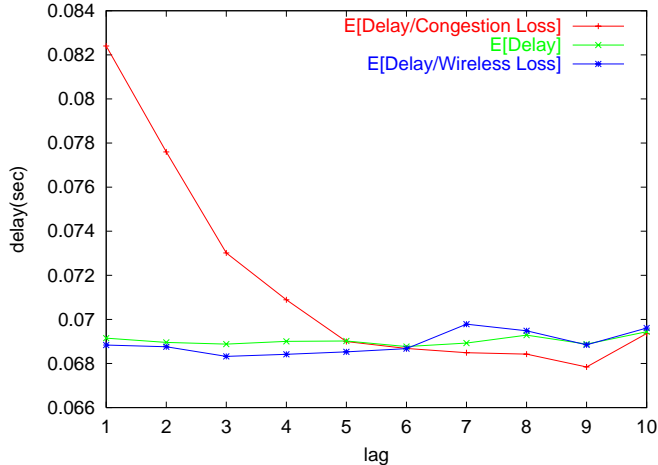


Figure 2. Expected delay (conditional and unconditional) at different lags for a TCP connection

lem:

$$\begin{aligned} H_c : y &= m_c + N(0, \sigma_c^2) \\ H_w : y &= m_w + N(0, \sigma_w^2) \end{aligned}$$

Substituting in equation (6), we get:

$$\mathcal{L}(y) = \left[\frac{\left(\frac{1}{\sqrt{2\pi\sigma_c^2}} \right) e^{-\frac{(y-m_c)^2}{2\sigma_c^2}}}{\left(\frac{1}{\sqrt{2\pi\sigma_w^2}} \right) e^{-\frac{(y-m_w)^2}{2\sigma_w^2}}} \right] \begin{matrix} H_c \\ > \\ < \\ H_w \end{matrix} \frac{C_{cw}P_w}{C_{wc}P_c} \quad (7)$$

Taking the ln of both sides and rearranging the terms, we have:

$$-\frac{(y-m_c)^2}{2\sigma_c^2} + \frac{(y-m_w)^2}{2\sigma_w^2} \begin{matrix} H_c \\ > \\ < \\ H_w \end{matrix} \ln\left(\frac{\sigma_c\Gamma}{\sigma_w}\right) \quad (8)$$

where $\Gamma = \frac{C_{cw}P_w}{C_{wc}P_c}$. Figure 3 illustrates the two possible loss scenarios. Γ determines the degree of correct classification (or misclassification). The area denoted by P_D represents the correct classification of congestion, whereas the area denoted by P_F represents the misclassification of wireless loss as congestion-induced. The value of Γ depends on the penalties of misclassification as well as the ratio of wireless to congestion loss probabilities. Note that in practice, the two penalties of misclassifying loss type, C_{cw} and C_{wc} , are not necessarily equal and greatly depend on the source behavior, in our case, TCP. Furthermore, the degree of wireless losses, P_w , affects the sending rate of TCP, which in turn determines the degree of congestion losses, P_c . In this paper, we vary the value of Γ so we quantify the potential gains and limits of end-to-end error classification.

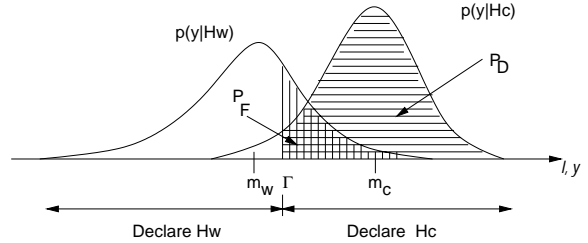


Figure 3. Scalar Gaussian case for binary hypothesis testing for distinguishing between congestion loss and wireless loss

5 Validation

In this section, we describe the tests we conducted to evaluate the characteristics of delays experienced by TCP flows in the presence of congestion and wireless losses. We conducted our experiments using the ns-2 network simulator [2]. The network topology used in the simulation is shown in Figure 4.

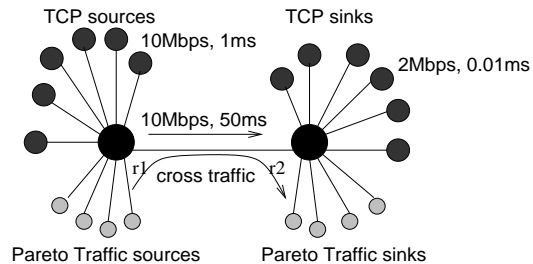


Figure 4. Wireless last-hop network topology setup

We have a number of TCP traffic source-destination pairs. The link from $r2$ to each TCP traffic sink has been assigned 2Mbps bandwidth and 0.01ms propagation delay. These links represent access wireless links with transmission errors. All other links are error free with 10Mbps bandwidth and 1ms propagation delay except the shared (bottleneck) wired link $r1 \rightarrow r2$ whose bandwidth is 10Mbps and delay is 50ms. The buffer size at $r1 \rightarrow r2$ is equal to the bandwidth-delay product and all other buffer sizes are set to default value of 50 packets. All the TCP sources and On/Off cross traffic UDP sources are started randomly between 0 sec and 3 sec and the simulations are run til 1000 sec. For each cross connection, the On and Off periods are Pareto distributed with average duration of 100ms each and shape parameter of 1.5. Unless otherwise specified, in all experiments we use 10 TCP connections and 20

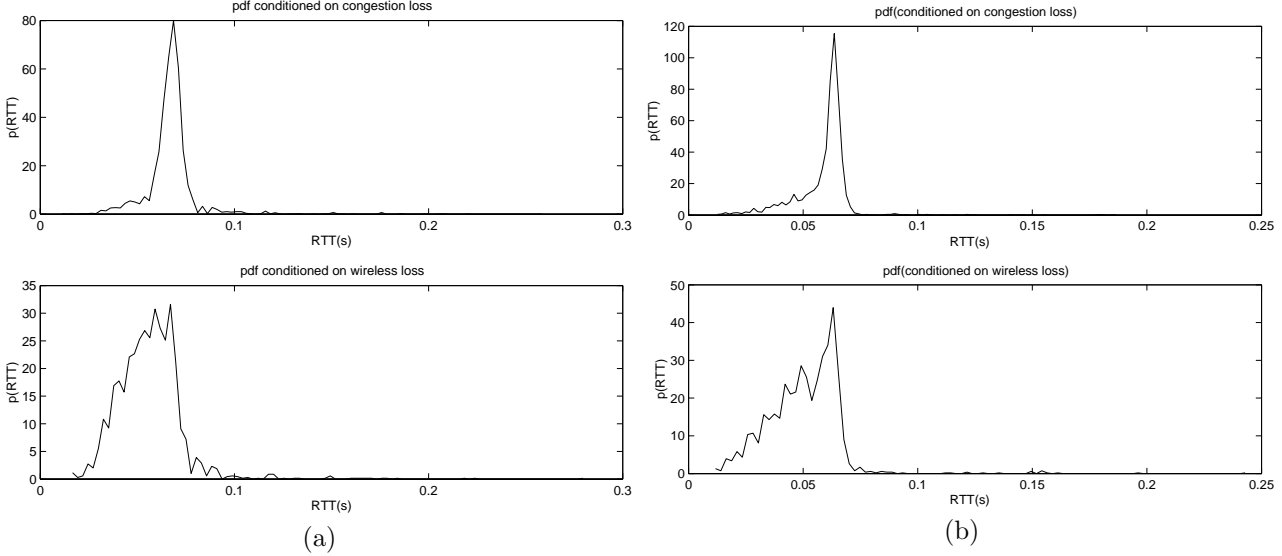


Figure 5. Delay characteristics of a TCP connection. (a) $p_c = 1.5\%$, $p_w = 0.8\%$, $m_c = 0.0672$, $\sigma_c = 0.0138$, $m_w = 0.0563$, $\sigma_w = 0.0173$. **(b)** $p_c = 3\%$, $p_w = 2\%$, $m_c = 0.0594$, $\sigma_c = 0.0123$, $m_w = 0.0531$, $\sigma_w = 0.014$.

Pareto cross-traffic connections.

Figures 5(a) and (b) plot the delay distribution for a representative TCP connection for different loss rates. In Figure 5(b), 30 Pareto cross-traffic connections are used. The experimental probability density functions (pdf's) indeed fit a Gaussian distribution with 95% confidence. We observe that keeping everything else the same, increasing the wireless loss rate reduces the congestion loss rate as TCP (without any enhancement) responds to wireless losses as if they were congestion-induced, i.e. TCP backoffs and reduces its sending rate (congestion window). The pdf's in Figure 5 show that the delay samples collected before wireless losses are wider spread than those collected before congestion losses. The latter distribution has a relatively higher mean but lower variance.

6 Performance Metrics

Having discussed the form of an optimal error detection test (cf. Section 3) and the nature and collection of data samples in the loss type detection problem (cf. Section 4), we now address how to characterize the performance of the decision rules.

To evaluate the performance of a decision rule, we have two metrics, namely (i) the expected value of the cost $E[C_{D(y)}]$ (cf. equation 2), and (ii) the probability of misclassification error, $\Pr[\text{Error}]$ (given by equation (10) below). Referring to equation (2), we can

write the expected cost as follows:

$$\begin{aligned}
 E[C_{D(y)}] &= C_{wc}P\{\text{Decide } H_w|H_c\}P_c + \\
 &C_{cw}P\{\text{Decide } H_c|H_w\}P_w \\
 &= (C_{wc}P_c + C_{cw}P_w) - \\
 &(C_{wc}P_cP\{H_c|H_c\} + \\
 &C_{cw}P_wP\{H_w|P_w\})
 \end{aligned} \tag{9}$$

Note that in equation (9) we used the fact that $P\{H_c|H_c\} + P\{H_w|H_c\} = 1$, and that $P\{H_c|H_w\} + P\{H_w|H_w\} = 1$. The first component (comprised of the first two terms) is independent of the decision rule used by a protocol; it is based only on the ‘‘prior’’ elements of the problem; and it represents a fixed penalty. The second component varies as a function of the decision rule. To minimize cost, the probabilities of correctly classifying congestion-induced (wireless) losses as of congestion (wireless) type should be maximized.

The probability of error can be expressed using Bayes rule as follows:

$$\Pr[\text{Error}] = P\{H_w|H_c\}P_c + P\{H_c|H_w\}P_w \tag{10}$$

Thus, we can determine the performance of a decision by calculating $P\{H_c|H_c\}$ and $P\{H_w|H_w\}$, which should be maximized. We know that Maximum Likelihood Tests are optimal and thus we focus on knowing the values of $P\{H_c|H_c\}$ and $P\{H_w|H_w\}$ for every possible value of the threshold, $\Gamma = \frac{C_{cw}P_w}{C_{wc}P_c}$ (cf. equation (7)).

To compactly represent every possible decision rule, we plot the values of $(1 - P\{H_w|H_w\})$, $P\{H_c|H_c\}$ as Γ is varied. Thus, we would like to find Γ that maximizes $P\{H_c|H_c\}$ while at the same time, minimizes the misclassification probability $(1 - P\{H_w|H_w\})$ —put another way, we want to maximize the difference between these two probabilities. In the standard estimation literature, this kind of plot is termed *Receiver Operating Characteristics (ROC)* for the detection problem; $P\{H_c|H_c\}$ is denoted by P_D ; and $(1 - P\{H_w|H_w\})$ is denoted by P_F . Any point on an ROC curve corresponds to a particular choice of the threshold Γ .

To that end, from equation (6), expressing a general decision rule test as $\mathcal{L}(y) \underset{H_w}{\overset{H_c}{\geq}} \Gamma$ where $\mathcal{L}(y)$ is a random variable, we have:

$$\begin{aligned}
 P\{H_c|H_c\} &= \int_{\{y|\text{Choose } H_c\}} p_{Y|H}(y|H_c) dy \\
 &= \int_{\mathcal{L} > \Gamma} p_{L|H}(\mathcal{L}|H_c) d\mathcal{L} \\
 P\{H_w|H_w\} &= 1 - P\{H_c|H_w\} \\
 &= 1 - \int_{\{y|\text{Choose } H_c\}} p_{Y|H}(y|H_w) dy \\
 &= 1 - \int_{\mathcal{L} > \Gamma} p_{L|H}(\mathcal{L}|H_w) d\mathcal{L}
 \end{aligned}$$

In our scalar Gaussian detection problem (cf. equation (7)), we have $P\{H_c|H_c\} = Q\left(\frac{\Gamma - m_c}{\sigma_c}\right)$ and $P\{H_w|H_w\} = 1 - Q\left(\frac{\Gamma - m_w}{\sigma_w}\right)$ where $Q(x)$ is the error function⁴. In Figure 6, we plot the curves of $P_D = P\{H_c|H_c\}$ and $P_F = (1 - P\{H_w|H_w\})$ as function of the threshold, Γ . Ideally, we would like to identify the threshold value which maximizes the difference between P_D and P_F . These results correspond to the experimental setup of Figure 5(a). In this case, we have $p_c = 1.5\%$, $p_w = 0.8\%$, $m_c = 0.0672$, $\sigma_c = 0.0138$, $m_w = 0.0563$, $\sigma_w = 0.0173$.⁵ The optimal value of Γ is found to be 0.0582 which corresponds to a misclassification penalty ratio of $\frac{C_{cw}}{C_{wc}} = 0.1091$.

7 Performance Evaluation

In this section, we plot the ROC curves of a TCP connection for varying path characteristics using either

⁴ $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$.

⁵Note that if a TCP source, augmented by such Bayesian error classification, is modified to take different transmission control actions in response to different types of losses or network state, these values are likely to change.

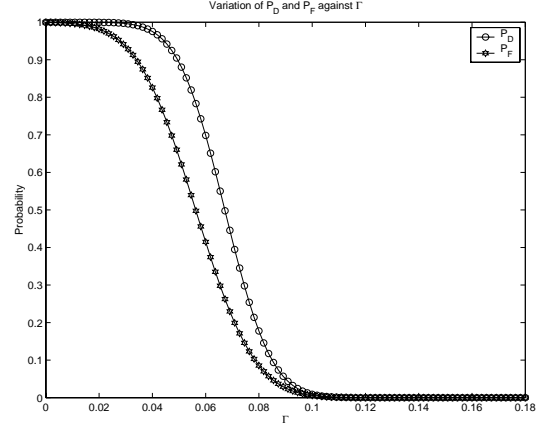


Figure 6. P_D and P_F for the case in Figure 5(a)

delay or short-term throughput samples. Each point on the ROC curve was obtained by averaging five independent runs.

7.1 Using Delay Samples

We use ns-2 [2] simulation to evaluate the performance of detecting the type of loss (congestion vs. wireless) based on RTT samples.

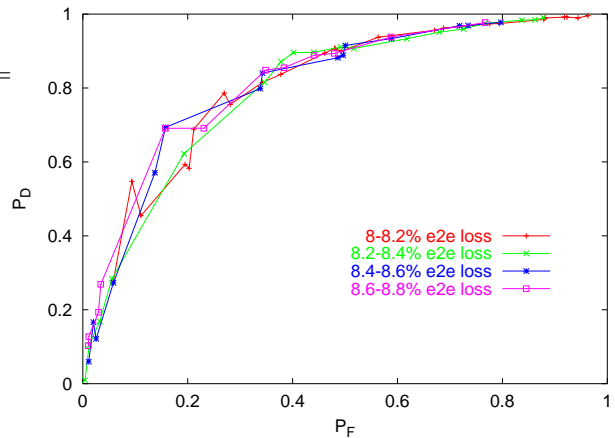


Figure 7. P_D and P_F as function of Γ

Figure 7 shows ROC curves for a TCP connection. Each curve represents a particular value of end-to-end packet loss rate. Each point on an ROC curve corresponds to a certain value of $\Gamma = \frac{C_{cw}P_w}{C_{wc}P_c}$ (cf. equation 7). Here we take $C_{cw} = C_{wc} = 1$ and vary Γ from $-\infty$ to ∞ by varying the ratio of the prior probabilities, P_w and P_c . Although wireless losses (occurring at P_w) are considered “exogeneous”, i.e. they are independent of the

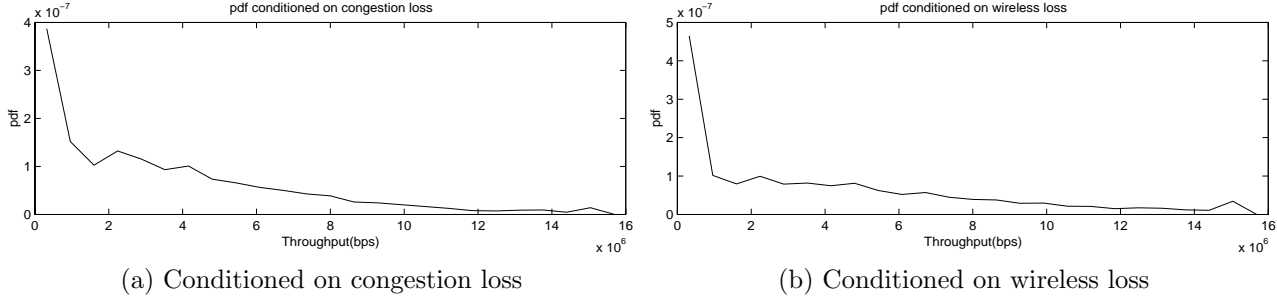


Figure 8. Throughput distribution of TCP, $p_c \approx 0.03$, $p_w \approx 0.02$, timescale = 0.1 second

behavior of TCP sources, congestion losses (occurring at P_c) are “endogenous.” In particular, the value of P_w actually affects P_c since TCP (without modification) responds to wireless losses as if they were congestion losses. To find an ROC for a TCP flow, we increased or decreased both cross traffic rate and wireless loss rate and considered those pairs of values, (cross traffic rate, wireless loss rate), which give approximately the same end-to-end loss rate.

As explained earlier in Section 4, we solve the error classification problem assuming conditional Gaussian distributions of the delay samples preceding wireless losses and congestion losses. The mean and variance of each of these two conditional distributions are then calculated. Figure 7 shows that the *measured* accuracy of the classification is quite good (even at reasonably high end-to-end loss rate), for example, $P\{H_c|H_c\}$ could be as high as 90% while maintaining a high $P\{H_w|H_w\}$ at 60%. This means that an error control scheme, equipped with such a Bayesian error classification, can largely maintain the conservative nature of TCP by backing off in response to correctly classified congestion losses, while correctly detecting the majority of wireless losses and reacting to them more intelligently. This good accuracy in classification implies that if the end-to-end loss rate is low to moderate, the Gaussian noise assumption is quite accurate. Furthermore, delay samples indeed carry good enough information about loss type. Another observation is that if we are ready to sacrifice a little in doing correct congestion classification, we can obtain an even better wireless loss classification.

Remark: The ROC curves we show provide an upper bound on $P_D = P\{H_c|H_c\}$ for each value of $P_F = 1 - P\{H_w|H_w\}$. The lower bound is just a mirror image of the upper bound around the 45° line $P_D = P_F$.

7.2 Using Throughput Samples

In [9], the authors used short-term throughput to identify congestion in mobile ad-hoc networks. Figure 8 shows the distribution of short-term throughput, measured over 0.1-second intervals, of a TCP connection (sharing a bottleneck with 9 TCP and 30 Pareto connections). The conditional distributions seem to fit one-sided Normal. We have observed that as wireless loss increases (or congestion loss decreases) the conditional distributions tend to be more close to two-sided Normal distributions.

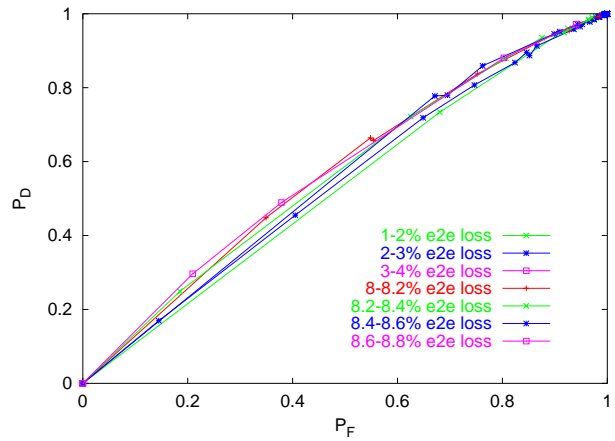


Figure 9. ROC plots of TCP using throughput samples

We show the performance of Bayesian error classification through ROC plots in Figure 9. We observe that the accuracy of using Gaussian hypothesis testing is not good. The performance is only a little better than the linear curve $P\{H_c|H_c\} = P\{H_c|H_w\} = 1 - P\{H_w|H_w\}$. That is, the performance of error classification is a zero-sum game—if we want a high correct classification of congestion losses ($P\{H_c|H_c\}$ close to 1), then we end up with a low correct classifica-

tion of wireless losses ($P\{H_w|H_w\}$ close to zero)! This shows that prediction of loss type based on short-term throughput is not effective.

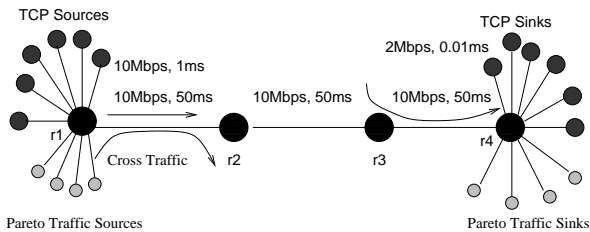


Figure 10. Topology with multiple bottlenecks

8 Other Scenarios

Multiple Bottlenecks: In this scenario, we simulated a network with two bottlenecks. The introduction of additional bottlenecks stresses the model as the samples become more dispersed. The simulated topology is shown in Figure 10. The access links of the TCP receivers are the wireless links. The wired links $r1 \rightarrow r2$ and $r3 \rightarrow r4$ constitute the bottleneck links. Other parameters are the same as those of the single-bottleneck topology used earlier.

Figure 11 confirms that the Bayesian based error classification based on delay samples performs well, and that delay is still a better performance metric than short-term throughput.

Markov Wireless Loss Model: Figure 12 plots the ROC curves based on delay and throughput metrics for a TCP flow that is subjected to correlated wireless errors modeled using a two-state Markov model. Again we observe that delay is a better performance metric than short-term throughput.

9 Congestion Detection using M-ary Hypothesis Testing

In this section, we describe and motivate the criterion based on which we classify losses or network load using M-ary hypothesis. One motivation is to estimate the network load in terms of number of competing TCP connections. A TCP source can then directly adapt to its bandwidth fair share [11]. Another motivation is to classify losses based on the estimated level of load (congestion). For example, one may classify a loss as wireless if the estimated load is low [7].

The degree of congestion is implied through different bottleneck utilization. In the M-ary case, we assume M different values of load condition L_i , $i = 0 \dots M-1$. These load conditions occur at prior probabilities $P_i = \Pr(L_i)$ such that $\sum_i P_i = 1$, and conditional densities denoted by $p_{Y|H}(y|L_i)$. For our Bayesian detection, a natural measure of network load is short-term throughput, which we found it to follow a log-normal distribution⁶.

Extending our arguments made in binary hypothesis testing, the optimal decision rule to choose hypothesis L_k given observation y minimizes the average cost of misclassification, and is given by:

$$\sum_{j=0}^{M-1} C_{kj} p_{H|Y}(L_j|y) \underset{\text{Not } L_i}{\overset{\text{Not } L_k}{\geq}} \sum_{j=0}^{M-1} C_{ij} p_{H|Y}(L_j|y) \quad \forall \text{ unique } i, k \text{ pairs} \quad (11)$$

Each of the $\frac{M(M-1)}{2}$ comparisons eliminates one of the hypotheses. We define the following set of likelihood ratios:

$$\mathcal{L}_j(y) = \frac{p_{Y|L}(y|L_j)}{p_{Y|L}(y|L_0)} \quad j = 0, \dots, M-1$$

where we take $\mathcal{L}_0 = 1$. Combining these likelihood ratios with Bayes rule, we have the following form of the optimal Bayes M-ary decision rule:

$$\sum_{j=0}^{M-1} C_{kj} p_{L|Y}(L_j|y) \mathcal{L}_j(y) \underset{\text{Not } L_i}{\overset{\text{Not } L_k}{\geq}} \sum_{j=0}^{M-1} C_{ij} p_{L|Y}(L_j|y) \mathcal{L}_j(y) \quad \forall \text{ unique } i, k \text{ pairs} \quad (12)$$

To keep the problem simple and illustrative, we pose the network state detection problem as 3-ary hypothesis testing with prior probabilities $p(y|L_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(\ln y - \mu_j)^2}{2\sigma_j^2}}$,

where μ_j and σ_j^2 are the mean and variance of the observed throughput (in log scale) under the j^{th} load condition. Assuming the misclassification penalty costs

⁶The throughputs from clients to a web server are also found to follow a log-normal distribution in [4]. Note that this load estimation problem is different from the earlier loss type estimation problem, where we found delay to be a better metric.

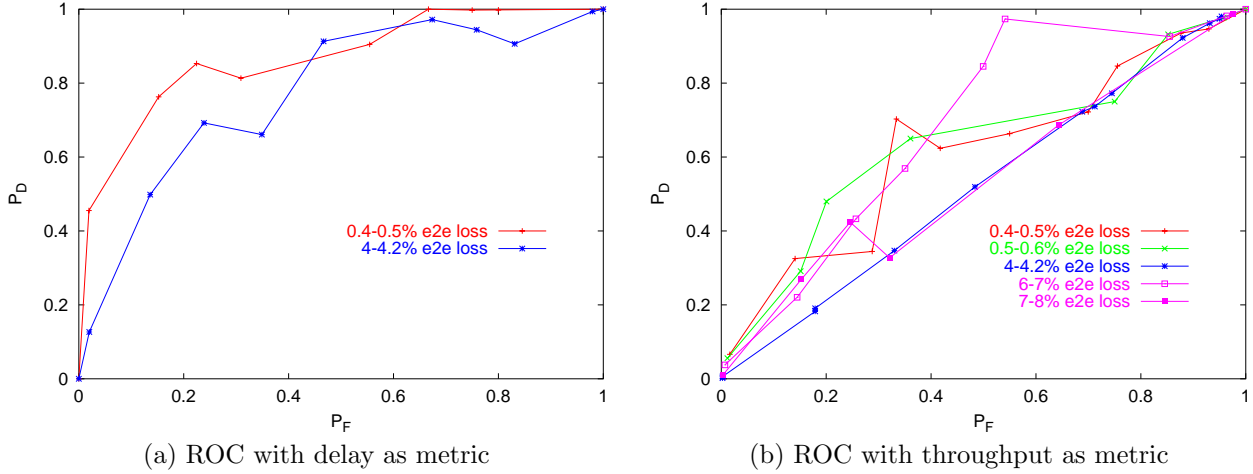


Figure 11. ROC plots for a TCP flow crossing multiple bottlenecks

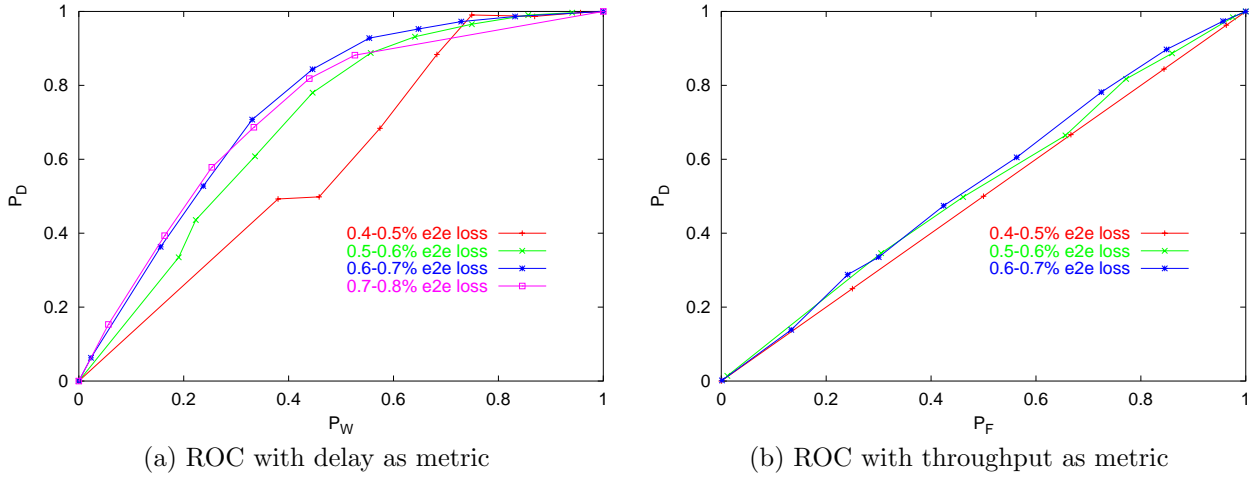


Figure 12. ROC plots for a TCP flow under Markovian wireless loss

are equal, the rules reduce to:

$$\frac{-(y - m_1)^2}{2\sigma_1^2} + \frac{(y - m_0)^2}{2\sigma_0^2} \underset{\text{Not } H_1}{\overset{\text{Not } H_0}{>}} \ln \left(\frac{\sigma_1 P_0}{\sigma_0 P_1} \right) \quad (13)$$

$$\left[\ln \left(\frac{\sigma_0}{\sigma_2} \right) - \frac{(y - m_2)^2}{2\sigma_2^2} + \frac{(y - m_0)^2}{\sigma_0^2} \right] \underset{\text{Not } H_2}{\overset{\text{Not } H_1}{>}} \frac{P_1}{P_2} \left[\ln \left(\frac{\sigma_0}{\sigma_1} \right) - \frac{(y - m_1)^2}{2\sigma_1^2} + \frac{(y - m_0)^2}{2\sigma_0^2} \right] \quad (14)$$

$$\frac{-(y - m_2)^2}{2\sigma_2^2} + \frac{(y - m_0)^2}{2\sigma_0^2} \underset{\text{Not } H_2}{\overset{\text{Not } H_0}{>}} \ln \left(\frac{\sigma_2 P_0}{\sigma_0 P_2} \right) \quad (15)$$

The comparisons are shown for a generic case in Figure 13. The dimension of the “likelihood space” is dependent on number of hypotheses, not the dimension of the observation, which may be greater than or less

than the likelihood dimension.

10 Validation of 3-ary Hypothesis Testing

To evaluate the above 3-ary case, we used Bayesian detection techniques on TCP sources sharing a single bottleneck as done previously. In our ns-2 simulation, we consider three hypotheses corresponding to three different distributions of instantaneous throughput observed by a TCP connection under different load conditions. We change the load in the network by changing the number of cross-traffic connections or changing the number of competing TCP connections. We obtain the prior probabilities $P_i = \Pr(L_i)$ based on the fraction of time a particular load (i.e., a certain number of con-

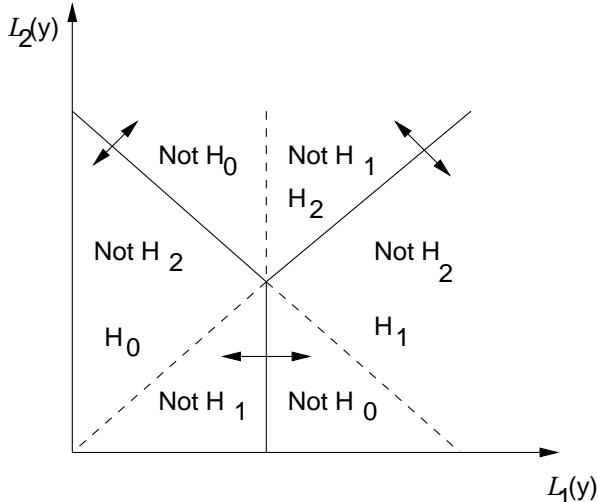


Figure 13. Decision boundaries in the space of likelihoods for an M-ary detection problem

nections) prevails during a simulation. We generate three different load conditions observed by the monitored TCP connection as follows: first, 9 TCP connections and 10 other cross-traffic connections start to compete with the monitored TCP connection for the first 1000 seconds, followed by 6 additional TCP connections for the next 1000 seconds, and lastly, 15 additional TCP connections for the last 1000 seconds. Therefore, $P_0 = P_1 = P_2 = \frac{1}{3}$ in this case.

Figure 14 shows the conditional pdfs of instantaneous throughput of a representative TCP connection for different total number of competing TCP connections. A plot labeled “TCP x ” represents a network load level of x competing TCP connections.

To evaluate the accuracy of the 3-ary Bayesian detection, we calculate the average values of $p(L_0|L_0)$, $p(L_1|L_1)$ and $p(L_2|L_2)$, where $p(L_i|L_i)$ denotes the probability of correctly classifying load level L_i (here L_0 corresponds to TCP9, L_1 to TCP15 and L_2 to TCP30). For the experiments in Figure 14, $p(L_0|L_0) = 0.73$, $p(L_1|L_1) = 0.25$, and $p(L_2|L_2) = 0.30$. Consistent with Figure 3, we observe that the correct classification probability degrades as the conditional distributions overlap, i.e. it becomes harder to distinguish the different P_D regions. Such distinguishability increases for more disparate network load levels.

11 Conclusion and Future Work

With the fast growth of the Internet in scope and scale, the congestion-oriented design of TCP has been

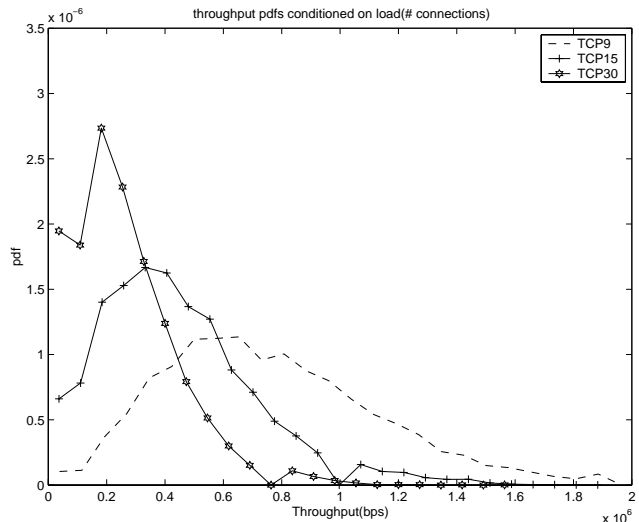


Figure 14. Log normal pdfs under different load conditions, $m_0 = 0.75, \sigma_0 = 0.38, m_1 = 0.43, \sigma_1 = 0.28, m_2 = 0.24, \sigma_2 = 0.19$ (Mbps)

challenged. Many studies have reported on the degradation in TCP performance in heterogeneous settings, and many proposed modifications to TCP or the network itself. In this paper, we step back and examine how well can TCP estimate the state of the network. We formulated the state estimation problem as statistical hypothesis testing and used Maximum Likelihood Ratio Tests. We found that even with Gaussian-based estimation, we can achieve high accuracy in state or loss type classification. Furthermore, we found that for loss classification (congestion versus wireless losses), delay measurements carry more information than short-term throughput.

Our analysis assumed knowledge of network conditions which may not be readily available to a transport protocol. In particular, we assumed knowledge of the delay or short-term throughput distributions conditioned on the network state or loss type. Furthermore, we computed the ROCs for varying prior probabilities P_c and P_w and misclassification penalties—the location of the actual operating point on the ROC curve depends on their values. Our goal in this paper was to develop the optimal estimation decision rules and quantify the best possible performance that a Bayesian estimator can achieve.

We are currently investigating ways to obtain the assumed knowledge of network conditions. For example, we can estimate the conditional distributions analytically under certain reasonable assumptions on the Internet path (for example, as in [18]). Also, a TCP de-

detector can find the operating point on the ROC curve which would satisfy the requirements of the application. In general, the TCP detector should attempt to maximize $P\{H_c|H_c\}$ subject to a high $P\{H_w|H_w\} \geq \alpha$. This way congestion control actions are taken in response to congestion, while avoiding a degradation in TCP throughput during wireless losses. A solution to this constrained optimization problem is referred to as *Neyman-Pearson rule* [13]. In the case of the optimal Bayes Likelihood Ratio Test, this solution leads to an operating point on the ROC curve which corresponds to the desired $P\{H_w|H_w\}$. The effectiveness of the control can then be assessed based on how high $P\{H_c|H_c\}$ is. We will report on these results in a future paper.

Acknowledgment

Thanks to W. Clem Karl for his helpful suggestions.

References

- [1] *Bandwidth Used by Different Traffic Types*. <http://www.icir.org/floyd/ccmeasure.html#bandwidth>.
- [2] *The Network Simulator - ns-2*. <http://www.isi.edu/nsnam/ns>.
- [3] H. Balakrishnan, S. Seshan, and R. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. In *IEEE/ACM Wireless Networks, Vol 1. No. 4*, pages 10–25, December 1995.
- [4] H. Balakrishnan, S.Seshan, M. Stemm, and R.Katz. Analysing Stability in Wide-Area Network Performance. In *Proceedings of ACM SIGMETRICS'97*, June 1997.
- [5] Hari Balakrishnan, V. Padmanabhan, S. Seshan, and R. Kartz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. In *Proceedings of ACM SIGCOMM'96*, 1996.
- [6] Hari Balakrishnan, S. Seshan, and R. Kartz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. In *ACM Wireless Networks, 1(4)*, Dec 1995.
- [7] Saad Biaz and Nitin H. Vaidya. Distinguishing Congestion Losses from Wireless Transmission Losses: A Negative Result. In *Proceedings of IEEE Symposium ASSET'99*, 1999.
- [8] Antonio Capone and Fabio Martignon. Bandwidth Estimates in the TCP Congestion Control Scheme. In *Proceedings of IWDC'01*, Italy, 2001.
- [9] Zhenghua Fu, Benjamin Greenstein, Xiaoqiao Meng, and Songwu Lu. Design and Implementation of a TCP-Friendly Transport Protocol for Adhoc Wireless Networks. In *Proceedings of IEEE ICNP'02*, Paris, 2002.
- [10] L. A. Grieco and S. Mascolo. TCP Westwood and Easy RED to Improve Fairness in High-Speed Networks. In *Proceedings of Seventh International Workshop on Protocols For High-Speed Networks (PfHSN'2002)*, April 2002.
- [11] Guanghui He, Yuan Gao, Jennifer C. Hou, and Kihong Park. A Case for Exploiting Self-Similarity of Network Traffic in TCP Congestion Control. In *Proceedings of the 10th IEEE Conference on Network Protocols*, Paris, November 2002.
- [12] Van Jacobson. Congestion Avoidance and Control. In *Proceedings of ACM SIGCOMM*, August 1998.
- [13] Thomas Kailath and H. Vincent Poor. Detection of Stochastic Processes. *IEEE Transactions of Information Theory*, 44(6), October 1998.
- [14] Dina Katabi, Mark Handley, and Charles Rohrs. Internet Congestion Control for High Bandwidth-Delay Product Networks. In *Proceedings of ACM SIGCOMM 2002*, August 2002.
- [15] Jun Liu, Ibrahim Matta, and Mark Crovella. End-to-End Inference of Loss Nature in a Hybrid Wired/Wireless Environment. In *Proceedings of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.
- [16] Jim Martin, Arne Nilsson, and Injong Rhee. A New Method for Analysing Feedback-based Protocols with Applications to Engineering Web Traffic over the Internet. In *Proceedings of ACM SIGMETRICS, CA*, 2000.
- [17] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, and N. Vaidya. Long Thin Networks. In *RFC 2757*, Jan 1997.
- [18] Olav Osterbo. An Approximate Method to Calculate the Distribution of End-to-End Delays in Packet Networks. Technical report, <http://www.telenor.com/rd/publisering/pub102.shtml>, 2002.