PRR Is Not Enough

Kannan Srinivasan, Maria A. Kazandjieva, Mayank Jain, Philip Levis
Computer Systems Lab
Stanford University
Stanford, CA 94131
{srikank,mariakaz,mayjain}@stanford.edu, pal@cs.stanford.edu.

Abstract

We study the effects of wireless channel burstiness on TCP. We measure TCP throughput over single-hop link traces from MIT's 802.11b Roofnet and Intel Berkeley's 802.15.4 Mirage testbeds. We observe that links with the same packet reception ratio (PRR) have throughput variations of up to 320%. We find that differences in throughput are accompanied by differences in link burstiness. Using the Gilbert-Elliott model, we compute the parameter μ as a measure of burstiness. We show that for sufficiently large data traces, a simple second-order fit over both PRR and μ lowers the estimation error of TCP throughput by 50%. The estimation error reduces by 60%-99% for a more general fit over PRR and μ. We find that while μ has good deductive quality, the corresponding Gilbert-Elliott model is not accurate for simulation: TCP throughput values from empirical links and their corresponding simulated links based on the Gilbert-Elliott model can differ by up to 80%. These results help develop a better understanding of the underlying causes for the wide variations seen wireless network performance.

1. Introduction

This paper is a measurement study of the dynamics and behavior of wireless links. It strives to broaden the insight into real-world systems as well as take a step toward bridging the gap between empirical observations and simulation.

Wireless protocol design is notoriously difficult. This is especially true with open-spectrum technologies, such as 802.11b, 802.11g, 802.11n, Bluetooth and 802.15.4, all of which share a 2.4GHz band with cordless phones, microwaves, and other consumer devices. Being in an open band makes these technologies inexpensive and therefore ubiquitious. However, in practice, this means that their RF environments tend to be complex, interference-heavy, and hard to tackle with clean analytical approaches or simulations.

Researchers have addressed this challenge by deploying, testing, and evaluating protocols on indoor and outdoor mesh testbeds [27, 4, 8, 9, 10, 17, 19, 20, 29]. Deployments demonstrate that a protocol can work in practice, but their results are difficult to generalize. Two separate testbeds can produce completely different results. To explain this variation, deployment studies typically report high-level network properties, such as a connectivity graph, reception ratios, average hop count, and average degree.

These properties seek to succinctly describe the dynamics of a very complex system. While useful for highly general observations – e.g., lossier networks will have lower throughput – they are not sufficient for explaining fine-grained performance results. Deployments can demonstrate whether a protocol works, but generalizing to predicting performance on another wireless mesh remains out of reach.

If these metrics are insufficient for explaining network performance, then what else should we measure? A complete answer to this question would enable a rigorous analytical understanding of wireless protocols that matches experimental results. This paper is a first step towards developing a systematic methodology for characterizing wireless deployments. It provides an insight on a new measurement, channel burstiness, and its effect on TCP throughput.

This paper examines the most simple and basic communication case: a single pair of communicating nodes. Using packet traces from 802.11b and 802.15.4 testbeds, we find that single-hop TCP throughput for two links with the same packet reception ratio can vary by as much as 320%. Therefore, packet reception ratio measurements are not enough to predict TCP throughput. Looking at what differs between such links, we see that links with the same reception ratio lost packets in very different ways. Links with higher throughput have bursts of losses, while links with lower throughput have random, independent losses.

If loss burstiness can affect protocol performance, we need a way to measure it. We draw on several proposals from literature and choose μ as a metric of burstiness. μ is computed from the Gilbert-Elliott channel model and measures average link burstiness. We find that μ, combined with the PRR, can accurately predict TCP throughput for links with PRR between 20% and 60%. Section 7 shows that adding μ into consideration reduces the error of estimated TCP throughput by 50%-99% as opposed to using only the reception ratio of the link. Therefore, reporting a link's μ can give an enhanced understanding of a wireless network.

The strong correlation that μ has with the behavior of TCP raises the question of whether μ describes the actual underlying dynamics in wireless links. If the Gilbert-Elliott model accurately captures what causes these performance variations, then it can be used in simulators to improve their results. To test this hypothesis, we derive Gilbert-Elliott models from measured μ values and link reception ratios. Comparing the TCP throughput of the real links with simulated links shows that they can differ by as much as 80%. While μ describes some link property that is correlated with TCP performance, links do not follow a Gilbert-Elliott model.

This paper's three research contributions are scientific and descriptive. First, we observe that there are TCP throughput variations that PRR cannot fully account for. Second, we find that burstiness can explain much of these variations and reduce throughput estimation error by 50%-99%. Third, we find that while μ has significant deductive value in explaining a network, the Gilbert-Elliott model from which it comes does not accurately describe a link, and that more information, such as burst lengths, is necessary for simulation.

The rest of the paper is structured as follows. Section 2 describes related work on burstiness, TCP, and modeling wireless links. Section 3 describes the testbeds and data traces, as well as our TCP NewReno implementation. Section 4 shows that PRR is not enough to understand TCP's performance. Section 5 presents the Gilbert-Elliott model and a method for quantifying burstiness. Section 6 examines how μ correlates with TCP performance in 802.15.4 and 802.11b networks. Section 8 shows that the time over which we observe a link needs to be long enough for μ to converge to a stable value. Section 9 explores whether the TCP results imply simulators can use a Gilbert-Elliott model to good effect. Finally, Section 10 concludes and discusses future work.

2. Related Work

Although TCP was designed for wired networks, it is widely used in the wireless domain for easy interoperability between the wired backbone and the last mile 802.11 and cellular networks. In addition, TCP/IP over 802.15.4 has recently gained momentum within the embedded network industry because to its interoperability and openness, with the IETF forming working groups [25, 3] to support the effort.

One of the main reasons for TCP's poor performance over wireless is its rate adaptation for packet losses: TCP assumes that all losses are due to congestion while in wireless networks losses can be because of channel errors [7]. This leads to degraded performance of TCP over wireless.

In the past, TCP performance analysis took in to account only the average loss rate [22, 21]. Lakshman et. al. [22] analyze TCP/IP performance in networks where the bandwidth-delay product is higher than the buffering, assuming random packet losses. Kumar et. al. [21] also assume random packet losses but analyze the effects of fast retransmit, fast recovery and coarse timeouts of different types of TCP.

Ignoring correlation of reception in links can lead to inaccurate results [32, 31, 6]. Correlation of reception in wireless links is typically modeled as a simple two state markov chain [13, 28, 32], the Gilbert-Elliott model being the most popular. Altman et. al. [6] model packet loss as a two state Markov process and analyze moments of TCP throughput. Yet, for simplicity, protocol designs typically assume links to have time independent reception [1, 2] with the packet reception ratio being the only modeling parameter. For instance, the immediate retransmissions in many MAC protocols assume that a packet failure and the immediate retry attempt have independent fates.

In practice, many wireless links are not as simple as a 2-state markov model [18, 20]. While it is an easy model to analyze, it is not accurate for all real links.

Jiao et. al. have shown that the two state markov model is inadequate to model link burstiness and that wireless links may be better modeled using inhomogenous markov chains in which the transition probabilities are not constants[18]. Köpke et. al. use real data traces and show that simple models like the Gilbert-Elliott model do not simulate the real links well by comparing the run-length distributions of real links and the links generated by simple models[20]. They show that chaotic maps can yield far better simulation models. There is also work that shows simple models such as the 2-state model can yield inaccurate predictions of higher layer protocol performance [18, 12].

In this paper, we do not propose a new way to quantify burstiness nor do we propose a new model to make simulations better. We look deeply at a simple parameter, μ from the Gilbert-Elliott model and see how correlated it is with TCP's performance. Unlike most work in the literature, we start from testbed data traces to study the TCP performance. We do not use simulations that use abstracted statistical models for the wireless channel but rather feed the real world traces as input to the simulator.

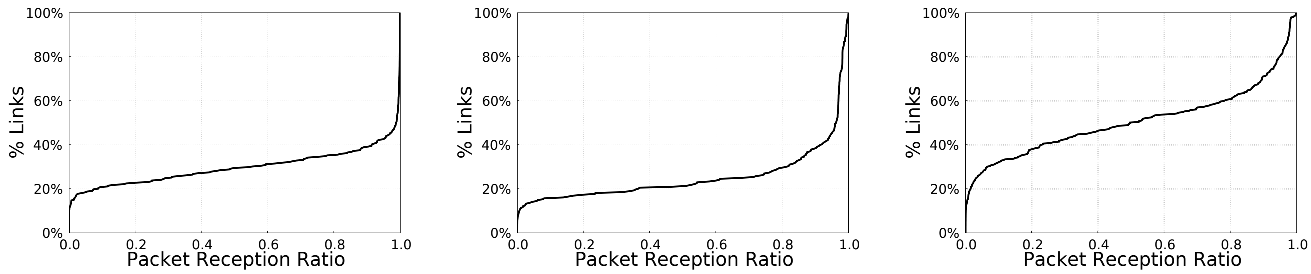
3. Methodology

This section describes the experimental methodologies of this paper, including the data sets, their high-level properties, and the simulation setup used to measure TCP throughput.

3.1 TCP Implementation

To measure TCP throughput, we use a nesC implementation of the NewReno TCP stack [5, 11, 30, 16] on top of TOSSIM [23]. TOSSIM simulates an environment of sensor nodes running TinyOS [15]. TinyOS is an event-driven operating system, specifically designed for network protocol implementations on embedded systems. nesC is an extension of the C language for programming in TinyOS.

To evaluate the performance of TCP, we feed datatraces of packet receptions and losses into TOSSIM. TCP signals the



(a) Mirage - Channel 26 (b) Mirage - Channel 16 (c) Roofnet - 11Mbps

Figure 1: Cumulative distribution of packet reception ratios in the three data sets. Most links are either poor or excellent but there are about 20% to 30% which fall in the intermediate PRR range.

simulator for a packet transmission; the success or failure of that packet is determined by the datatrace entry at the corresponding time. We are only interested in single-hop transmissions in order to understand a link's most basic behavior; we calculate the TCP throughput in packets per second.

While the RFC documents detail most TCP specifications, TCP sender and receiver timeout values are not explicitly stated. The timeout values in typical implementations are tuned for a multi-hop wired network with long Round-trip times. These values are not applicable to the single-hop wireless links studied in this paper. Here we briefly explain our choice of values for these variables.

On the sender's side, the retransmission timeout is subject to an upper and a lower bound. Ideally, the timeout must be greater than the time it takes to transmit one window of data, so not all packets in flight are retransmitted. In our TCP implementation, the maximum congestion window is 130 packets, but this number is rarely reached. For datatraces with 10ms interpacket interval it would take 1.3 seconds to send a window of data, but the size of the window is usually much smaller. Based on this, we can choose the lower bound of the retransmission timeout as 1 second, together with 60 seconds for the upper bound.

The ACK timeout on the receiver's side governs how long to wait for a packet before sending an acknowledgement. The timeout should be significantly lower than the sender's retransmission timeout, so the sender gets up-to-date information when there are a lot of losses. We choose an ACK timeout of 300ms for the data traces with 10ms interpacket interval. Since both timeouts depend on the interval between packet transmissions, in the following subsection we mention how TCP timeout values vary between data sets.

3.2 Datasets

We use datasets from two testbeds, namely Mirage and Roofnet.

3.2.1 802.15.4 Mirage Data

802.15.4 is an IEEE PHY-MAC standard for low power, low data rate networks. We measure 802.15.4 using the Intel Mirage testbed [17] - a set of indoor ceiling nodes.

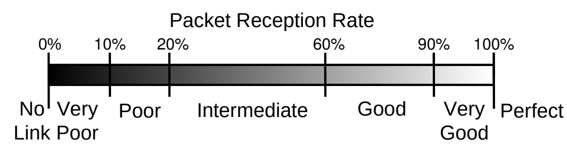


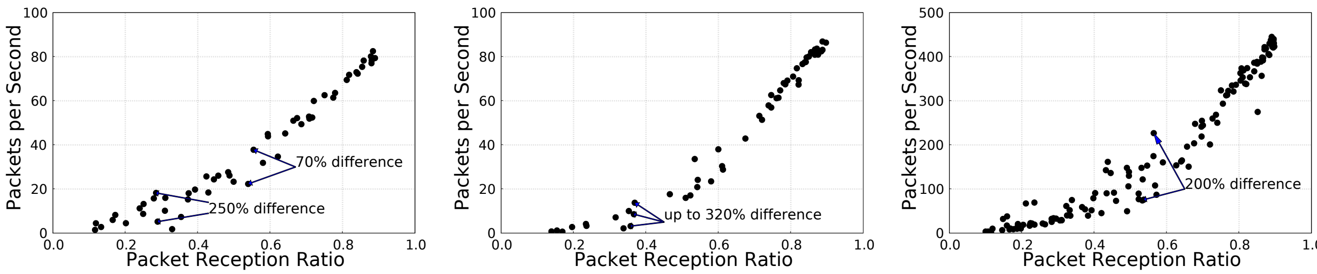
Figure 2: Terminology used to describe links based on PRR. Very poor links have a PRR $<$ 10%, poor are between 10% and 20%, intermediate links are between 20% and 60%, and good and very good links are $>$ 60%. A PRR of 100% is a perfect link.

To collect the Mirage datatraces, each node transmits broadcast packets every 10ms for a 1000-second duration, giving a total of 100,000 packets. All other nodes listen and a central server logs packet receptions. We collected data for two different 802.15.4 channels, channel 16 and channel 26. Channel 16 overlaps with the 802.11b transmission band and thus is subject to external interference from Intel's WiFi network, while channel 26 has very little interference.

3.2.2 802.11b Roofnet Data

In addition to Mirage, we used 802.11b datatraces from MIT's Roofnet study [4]. The traces were collected from an outdoor, wireless mesh network, operating at 11Mbps and 1Mbps. Similarly to Mirage, every node take its turn to broadcast packets.

In the 11Mbps case, packets were sent as fast as possible for 60 seconds; we subsample this data to produce 20000-packet traces with an inter-packet interval of 2ms and total running time of 40 seconds. Since packet transmissions take much longer in the 1Mbps experiments, the inter-packet interval is 20ms, and the the total number of packets - 3000. Section 8 shows that a trace length of 3000 is not enough for values of μ to converge. Therefore, in this paper we only use the 11Mbps Roofnet data traces. Since the Roofnet and Mirage data differ in the number of packets sent per second, 100 versus 500, we scale the TCP timeout parameters appropriately. This reduces the retransmission timeout upper and lower bounds to 200ms and 12 seconds respectively, and the ACK timeout to 60ms.



(a) Mirage Channel 26 (b) Mirage Channel 16 (c) Roofnet

Figure 3: Single-hop TCP throughput measurements for 802.11b and 802.15.4 datatraces. Within a given reception ratio, throughput varies by 20 to 320%.

3.3 Link Packet Reception Ratios

Figure 1 shows the cumulative distribution of reception ratios for the three datasets. A majority of the links have either a very high or very low PRR. The first, rarely leave space for improvement, while the latter might be impossible to improve.

Between 25% and 40% of the links have a variety of PRRs between 0.1 and 0.9. In addition, Sections 4 and 6 show that links with PRR between 0.2 and 0.6 are of significant interest when analyzing TCP throughput.

Figure 2 defines the terminology used in the rest of this paper to describe the reception ratios of wireless links. We use the terms very poor, poor, intermediate, good, very good, or perfect. We begin by examining all links that are poor, intermediate, and good, and then concentrate only on the intermediate links with PRR between 0.2 and 0.6. In the two Mirage datasets there are about 50 single-hop intermediate links, while in the Roofnet dataset there are over 120 intermediate data traces.

4. PRR Is Not Enough

The performance of multihop flows in wireless meshes typically falls well below what link-level bitrates would suggest. For example, simple unicast routing protocols, such as Srcr [8], exhibit approximately ten-fold reduction in TCP throughput between single-hop paths and paths longer than 4 hops. Approaches that take advantage of the broadcast nature of a wireless channel, such as ExOR's opportunistic receptions [9], or COPE and MORE's network coding [19, 10] see significant (35-1000%) throughput gains on some paths, but they still exhibit a ten-fold spectrum in performance.

The poor quality of wireless links is one possible explanation for this disparity. Another explanation is the choices made by routing protocols. It is important to first understand how TCP performance varies for single-hop paths, before investigating performance over multi-hop paths.

Figure 3 shows results for the three datasets described in Section 3. The TCP throughput is plotted against the link's PRR. As expected, PRR captures the general trend of increased throughput for higher reception ratio links. However, there are a number of cases in which two or more links have very similar PRR, yet different TCP throughput. The majority of throughput discrepancies happen in the intermediate reception ratio range, 20% to about 60%. For example, in the Mirage channel 26 data, a link with a PRR of 0.33 achieves TCP throughput of barely 2 packets per second, while another link with a PRR of 0.35 is at over 7 packets per second.

These two links will appear the same based on PRR alone, yet have a difference of over 250% in TCP performance. Similar observations are true for Channel 16 – for example three links with a PRR of 0.37 have TCP throughputs of 3, 8, and 14 packets per second respectively, resulting in a difference of up to 320%. In Roofnet, links with equal reception ratio can experience throughput variations of over 200%.

Therefore, even though PRR gives a coarse-grained understanding of what the expected throughput of a link is, it is not enough. However, in these experiments, PRR is the only variable in the classical formula for TCP throughput, as both RTT and MTU are fixed. The formula for predicting TCP throughput is:

$$Throughput = 1.3 \cdot \frac{MTU}{RTT \cdot \sqrt{Loss}}$$

There have been many modifications proposed to this equation to account for other TCP parameters such as timeouts and the number of packets acknowleged by a single ACK [24]. One drawback that Padhye et al. [26] have addressed is that the equation does not work well for channels with more than 2% losses.

All suggested modifications to the TCP throughput formula consider a wired multi-hop packet switched network. On the other hand, our experiments look at single-hop wireless TCP connections where packets are sent at regular intervals, and flight size and TCP performance are not RTT-constrained. In addition, links in the Mirage and Roofnet datasets exhibit losses much greater than 2%, and it is not clear how well the classic throughput equation will handle links with such low PRRs.

Furthermore, current TCP equations assume uniformly spaced losses. This is a worst case scenario since each loss

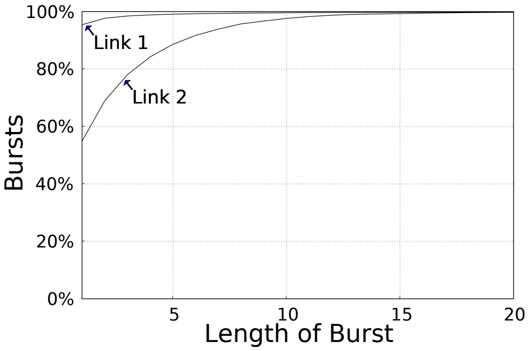


Figure 4: CDF of loss burst length for two links with the same PRR, in the Mirage testbed. Since one link has isolated losses more than 95% of the time, its TCP throughput is much lower than the other link.

maximally decreases the congestion window size. Losses on wireless links are often correlated, making links bursty. It is the temporal distribution of losses, in addition to their number, that affects TCP performance.

Figure 4 shows the cumulative distribution of loss burst lengths for the two links in the Mirage channel 26 dataset, shown in Figure 3, that have a TCP throughput difference of 250%. In order to focus on shorter runs of losses, we only show the x-axis up to 40 packets, capturing 99% of the bursts. For one link, 55% of all losses happen in isolation – the lost packet was preceded and followed by one or more successes. For the second link, that value is over 95%. Intuitively, the latter link has more independent losses, and as expected [14], is the link with lower throughput. These observations suggest that channel burstiness is a useful parameter that can give an insight in to protocol performance.

If such large differences in TCP throughput are possible over single-hop paths, then it is no surprise that multi-path mesh networks can exhibit significant variations in their performance. Aguayo et al.'s study of Roofnet [4] found that many of its links had bursty behavior, and studies of 802.15.4 have found similar behavior [29].

Therefore, in such networks, PRR is not enough to accurately characterize links. Instead, we need to expand the set of properties we measure and report on wireless links – our experimental lexicon. We need to find a suitable parameter for reporting burstiness in addition to other measurements. First, we must find the best way to quantify burstiness and then understand how this new parameter alters TCP throughput predictions.

5. Quantifying Burstiness

This section introduces the Gilbert-Elliott channel model that has been widely used in the RF community. It is a method for modeling channel memory, which has a direct implication on its burstiness. Although most literature treats the Gilbert-Elliott model as a bit model, we use it as a packet model, i.e. we model bursts of packet errors and successes rather bursts of bit errors and successes.

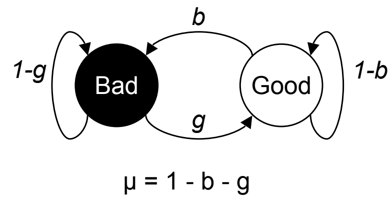


Figure 5: The Gilbert-Elliott model of a wireless link.

5.1 Defining Burstiness

In simple terms, link burstiness is the property of having temporally correlated losses and successes on a wireless link. A good first step towards defining what a bursty link looks like is to define what a bursty link does not look like. A link is not bursty when its packet successes and failures are not temporally correlated, i.e. the success and failure of each packet delivery is independent of all previous and future packet transmissions. While such a link will have runs of successes and failures, the lengths of these runs will follow a geometric distribution.

A *bursty* link is one where there is a positive correlation between similar packet events. Given a reception ratio R, the probability of receiving a second packet after one arrives successfully is greater than R, while the probability of receiving a second packet after one does not arrive successfully is less than R.

It is also possible for a link to be *oscillatory*. In an oscillatory link, there is a negative correlation between similar packet events. Links that quickly swing back and forth between periods of good and bad reception can introduce this behavior. Receiving a packet makes it more likely that a period of bad reception is about to follow and not receiving a packet makes it more likely that a period of good reception is about to occur.

5.2 Gilbert-Elliott Model

The Gilbert-Elliott model of a wireless link, shown in Figure 5 is a two-state Markov process. The states of the process correspond to the link being in a “good” state (zero probability of packet errors) or the link being in a “bad” state (nonzero probabiity of packet errors). The Gilbert-Elliott model has three parameters, g, the probability of transitioning to the good state from the bad state; b, the probability of transitioning to the bad state from the good state; and h, the loss rate in the bad state. The model assumes the good state has no errors and many analytical formulations assume, for simplicity, that the bad state loses all packets ($h = 100\%$). The rest of this paper assumes h equal to 100%.

The memory of the Gilbert-Elliott model is given by a parameter μ. A model's μ, defined as $1 - g - b$, has a direct implication on the burstiness of the channel. If μ is zero, then

$b = (1-g)$ and $g = (1-b)$. Put another way, the probability of transitioning to the good or bad state is independent of what state the model is in. As μ approaches 1, the probability of transitioning away from states approaches zero, causing the links to be very bursty. As μ approaches -1, the states are more likely to transition than to stay the same, causing oscillatory behavior.

In the case where $h = 100\%$, the parameters g and b are the probability of a successful packet following a packet failure and the probability of a packet failure following a successful packet. Encoding packet delivery as a binary string, $g = \frac{|01|}{|01|+|00|}$ and $b = \frac{|10|}{|10|+|11|}$.

5.3 Calculating μ For Empirical Channels

μ can be calculated for empirical channels from the data traces outlined in Section 3. As mentioned earlier, we use a simplified Gilbert-Elliott model where each failure corresponds to the bad state and each success corresponds to the good state. Looking at empirical data traces can give all state transitions for this model by labeling each failure as bad state and success as good state. States can then be encoded by binary numbers with 0 denoting bad state and 1 denoting good state. For example, a good to bad state transition will correspond to the occurance of a "10" in the bit stream and a bad to good transition to the occurance of a "01". Thus, using the packet delivery data encoded as a binary string and the equations for g and b from above, we can find g and b and also $\mu = 1 - g - b$.

6. TCP Performance

This section investigates the effect of burstiness on TCP performance for different testbeds. The TCP simulator (explained in Section 3) is run on packet traces obtained for the various testbeds, Mirage on Channels 26 and 16 and Roofnet, to evaluate TCP performance in packets per second.

In order to measure the effect of burstiness for different links, we synthesize an independent link with the same PRR as the empirical link under consideration. For an independent link, each packet has a probability of success equal to the PRR of the link,regardless of the fate of previous or future packets. Thus, to generate data for the independent link, we generate a bit pattern where each bit is 1 (meaning success) with a probability equal to PRR and 0 otherwise. The length of the synthesized trace is made equal to the length of the empirical trace for each link. For comparison of TCP throughput, the TCP simulator is then run on the data traces for the empirical link and the corresponding synthesized independent link and difference in throughput for the bursty link is noted.

6.1 Initial Observations

Figure 6 shows the plot of TCP throughput difference for the Mirage channel 26 trace for all links with PRRs between 10% and 90%. TCP throughput difference for various links

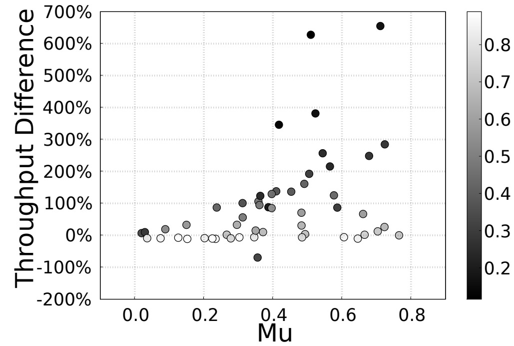


Figure 6: TCP throughput difference compared to a synthesized independent link for links with different μ and PRRs for Channel 26 Mirage testbed.

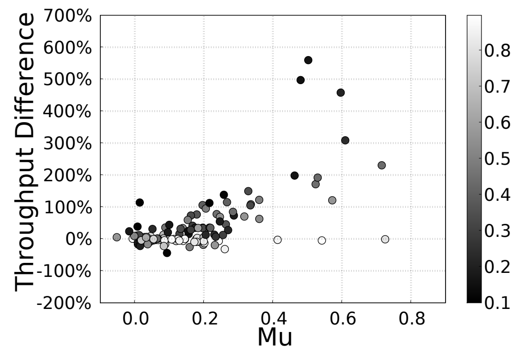


Figure 7: TCP throughput difference compared to a synthesized independent link for links with different μ and PRRs for the Roofnet testbed running at 11mbps.

is measured in comparison to the synthesized independent links for the same PRR and plotted versus the μ values for these links. The PRR levels are coded in shades of gray. Figure 7 shows the same plot for Roofnet data. The general trend of increasing throughput with increasing μ is visible in both plots, but there are a lot of outliers.

The outliers tend to occur at high PRRs ($> 60\%$), where the difference is less than expected and very low PRRs ($< 20\%$), where the difference is greater than expected. A better understanding of behavior of the Gilbert-Elliott model for high and low PRRs can explain the existance of outliers for these PRR values.

6.2 Gilbert-Elliott Model Behavior On Extreme PRRs

The increase in TCP throughput as an effect of burstiness is the result of an increase in the number and the length of runs of successes in a link trace compared to the independent link with the same PRR. As an approximation, we can compare the difference by looking at the probability of 2 consecutive successes in a trace for the two cases. To calculate this probability for the Gilbert-Elliott model explained in Section 5, we need to calculate the values of parameters g and b and

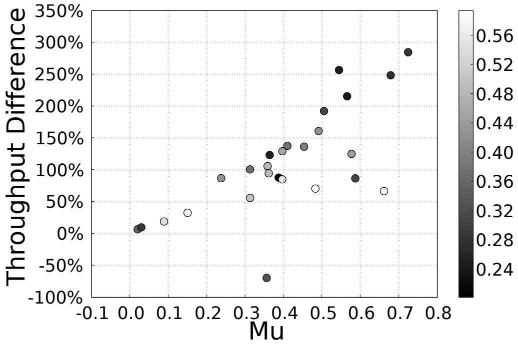


Figure 8: TCP throughput difference compared to a synthesized independent link for links with different μ and intermediate links for channel 26 Mirage testbed.

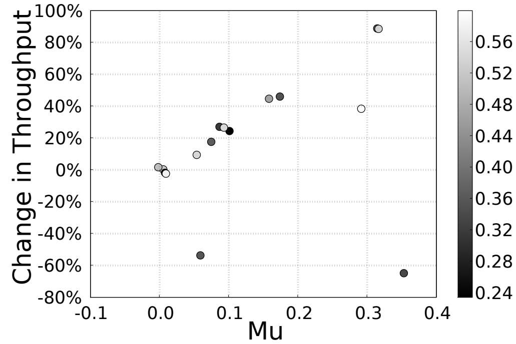


Figure 9: TCP throughput difference compared to a synthesized independent link for links with different μ and intermediate links for channel 16 Mirage testbed.

the steady state probabilities of being in the good and the bad states in terms of PRR and μ.

We call the steady state probabilities of being in the good and the bad states G and B respectively. Then, G = PRR and B = 1 - PRR. Further, in steady state, $G * b = B * g$. Using these and $\mu = 1 - b - g$, we get $b = (1 - PRR) * (1 - \mu)$ and $g = PRR * (1 - \mu)$. The probability of 2 consecutive successes is given by $G * (1 - b)$.

For a PRR of 0.1 and μ of 0.6, the probability of 2 consecutive successes is 0.06 for the bursty link and 0.01 for the independent link. This represents a large percentage increase in the probability of getting a burst and correspondingly, a large increase in the observed TCP throughput difference.

For a PRR of 0.9 and μ of 0.6, the probability of 2 consecutive successes is 0.864 for the bursty link and 0.81 for the independent link. This represents a very small percentage increase in the probability of getting a burst and correspondingly, a small increase in the observed TCP throughput difference. Therefore very high and very low PRR values do not adhere to the general trend observed for TCP throughput difference.

6.3 Observations For Intermediate Links

Figure 8 shows the plot of TCP throughput difference for intermediate links for channel 26 on the Mirage testbed. These links show a clear correlation between increasing throughput difference and increasing μ. Figure 9 shows the same plot for Mirage channel 16. Figure 10 shows the TCP throughput difference with μ for the Roofnet traces. In all three testbeds, an increase in μ has an accompanying increase in TCP throughput: this effect appears across different wireless channels in the 2.4Ghz band and across different data rates.

7. Data Fitting Analysis

Section 4 observed large variations of TCP throughput at intermediate PRR values. Section 6 showed that the variations in TCP throughput can be attributed to link burstiness

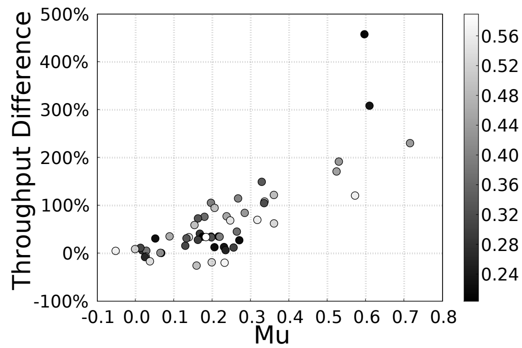


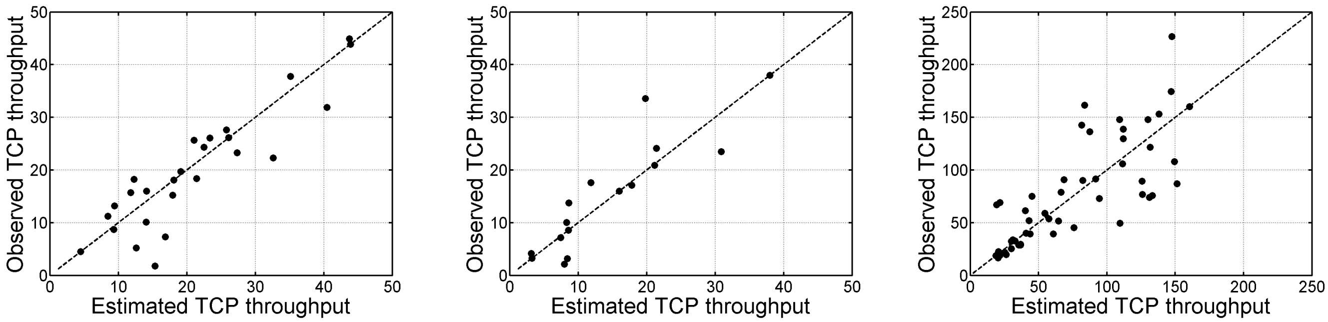
Figure 10: TCP throughput difference compared to a synthesized independent link for links with different μ and intermediate links for Roofnet testbed running at 11mbps.

and the Gilbert-Elliott channel memory parameter μ can be used to quantify this burstiness.

To evaluate the effect of the calculated μ on TCP throughput, this section looks at the accuracy of estimation of TCP throughput using only PRR and compares this with the accuracy obtained by using both PRR and μ. TCP estimates are calculated by finding the best fits based on the least L1 norm of error[1]. L1 norm is used rather than Least Squares because it is more robust to outliers.

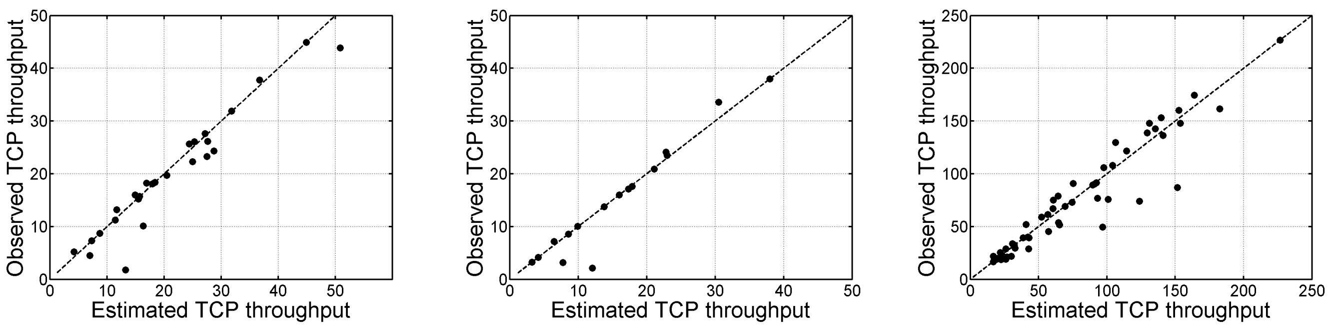
The TCP throughput expression in Section 4 shows the throughput to be proportional to the inverse square root of the loss rate, i.e. $\propto \frac{1}{\sqrt{1-PRR}}$. As discussed before, this expression may not be applicable to single hop links with high loss rates. Our experiments have also shown that a second order estimate provides a good fit for observed TCP throughput based on PRR values. Comparing best fits for TCP throughput for linear fit, quadratic fit and fit with $\frac{1}{\sqrt{1-PRR}}$ shows the

[1]L1 norm of the error is the sum of the absolute values of error at each data point



(a) Mirage - Channel 26 (b) Mirage - Channel 16 (c) Roofnet

Figure 11: Best fit of observed TCP throughput based on corresponding link PRR values for intermediate links in different testbeds. The x-axis is the estimate of TCP throughput based on observed PRR and y-axis is the actual TCP throughput



(a) Mirage - Channel 26 (b) Mirage - Channel 16 (c) Roofnet

Figure 12: Best second-order fit of observed TCP throughput based on corresponding link PRR values and μ values for PRRs (between 20% and 60%) for different testbeds. The x-axis is the second-order estimate of TCP throughput based on the link PRR and μ and y-axis is the actual TCP throughput

best fit average normalized errors[2] are within 10% of each other. For example, the linear, quadratic and $\propto \frac{1}{\sqrt{1-PRR}}$ fits for Mirage Channel 26 data trace have average normalized errors equal to 0.0389, 0.0419, 0.0396 respectively. Including all three terms in a fit provides the best PRR based fit. In this case, the TCP throughput estimate is given by:

$$a * \frac{1}{\sqrt{1-PRR}} + b * PRR + c * PRR^2 + d$$

Figure 11 shows the best PRR based fit estimates of TCP throughputs for the different testbeds for intermediate links. The estimates are plotted against the observed TCP throughput values.

Based on the fit terms for PRR, a similar set of terms for μ, i.e. $\frac{1}{\sqrt{1-\mu}}$, μ and μ^2 can be used for fitting TCP throughput readings. Taking all terms for PRR and μ and combinations of these terms (for example $PRR.\mu$, $PRR^2.\frac{1}{\sqrt{1-\mu}}$ etc.)

Testbed	Estimation Error ($x10^{-2}$)			% Improvement	
	PRR Only	PRR and mu		15 term	2nd order
		15 term	2nd order		
Mirage Ch26	3.74	1.13	1.91	69.88%	48.97%
Mirage Ch16	3.12	0.0159	1.30	99.49%	58.16%
Roofnet	4.30	1.70	1.96	60.41%	54.45%

Figure 13: Average error in estimating TCP throughput using only PRR and using both PRR and μ.

gives an expression with 15 terms for fitting TCP throughput. A best fit over this expression gives a significant reduction in estimation error as shown in Figure 13. Fitting over 15 parameters and subsequently reporting those parameters may not be feasible in practice. The second order expression based on μ and PRR can provide an approximation for fitting TCP throughput data. In its most general form, a second order fit for the TCP throughput based on both PRR and μ is

[2]The errors are normalized with respect to packets per second at the link layer and averaged over the number of links. Packets per second on the link layer are 500 for the Roofnet traces and 100 for the Mirage traces.

given by an equation of the form:

$$a * PRR + b * PRR^2 + c * \mu + \ d * \mu^2 + e * PRR * \mu + f$$

where a, b, c, d, e and f are constant coefficients. Figure 12 shows the best PRR and μ based second order fit estimates of TCP throughputs for the different testbeds and intermediate PRRs. As can be seen from the figures, incorporating μ for estimating TCP throughput improves the fit considerably even with the second order approximate fit being used. Numerically, incorporating μ into the estimate reduces the L1 norm of the error by more than 60% by using the 15 term expression and by approximately 50% by using the second order fit expression. Figure 13 summarizes the results for the average error in estimation for different datasets using only PRR and both PRR and μ. The table shows results from both using the 15 term expression combining μ and PRR and the second order approximation. The percentage improvement shown is improvement of each fit over the PRR only fit. The average errors are normalized based on the link level packets per second and averaged over the number of links.

An observation in the best fitting process is that the coefficients for the fits over different traces vary significantly. This can mean that the nature of dependance of TCP throughput on PRR and μ changes across different rates and protocols. However, it is clear that including μ into an expression for estimating TCP throughput over a wireless link can significantly improve the estimate. The expressions used in this section are all heuristic and only serve to compare the estimation qualities of PRR alone and PRR combined with μ. These results point to the fact that deriving analytical expressions for TCP throughput on wireless channels should take a measure of channel burstiness into account.

8. Convergence of μ

This section investigates the number of packets needed to measure μ. For short packet traces containing 6000 packets, the throughput differences in TCP do not form a very good fit with μ and PRR.

This observation can be explained by looking at some specific links in the Roofnet datasets with a reduced trace length. The traces are sub-sampled at a packet interval of 10ms to reduce the length of each trace to 6000 packets. Figure 14 shows the TCP throughput difference versus μ for the sub-sampled links. Two links, marked as points A and B in the figure, have similar values for PRR and μ but show very different TCP throughput values. Figure 15 shows the distribution for the bursts of successes for A and B. The larger values of the distribution occuring for smaller bursts have been capped off to look at the infrequent occurances of longer bursts. A shows a couple of instances of very long bursts of successes, one being 117 packets long and the other 158 packets long. These long bursts are missing from the trace of B. Very long bursts, though infrequent, can cause a considerable improvement in TCP throughput. Since the

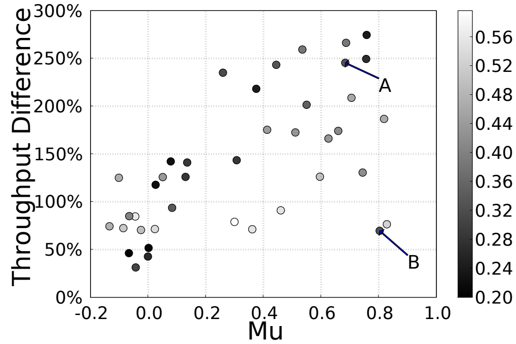
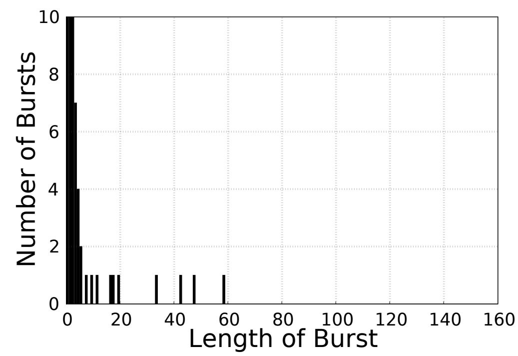
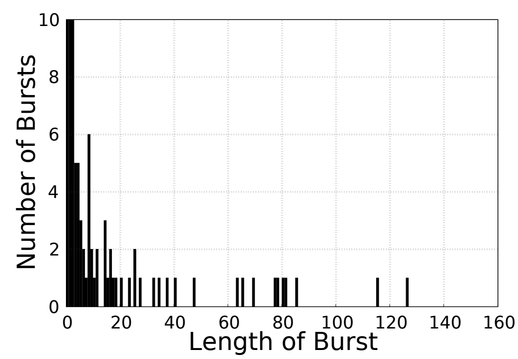


Figure 14: TCP throughput difference for sub-sampled data compared to a synthesized independent link for links with different μ and intermediate PRRs for Roofnet testbed. The sub-sampling interval is 10 ms



(a) Link A



(b) Link B

Figure 15: Distribution of burst lengths for 2 links. Link A has two bursts of successes longer than 100 packets. Link B does not exhibit any burst longer than 100 packets

overall number of packets is not too large (6000 packets), the effect of such a burst does not get completely amortized.

These observations show that the length of the sub-sampled traces (6000 packets) may not be enough for the values of μ and PRR to converge for a given link. To validate this claim, we divide the Mirage testbed packet traces containing

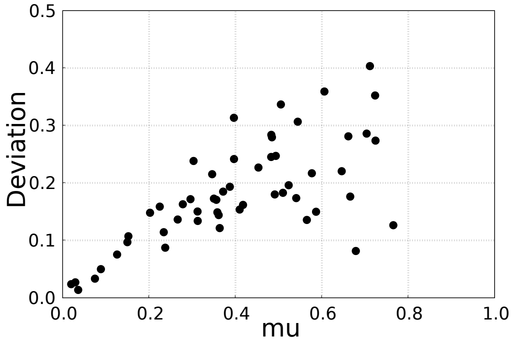


Figure 16: Root mean square of deviations from the overall trace μ for μ measured over all chunks of 6000 packets in each trace plotted against the overall trace μ.

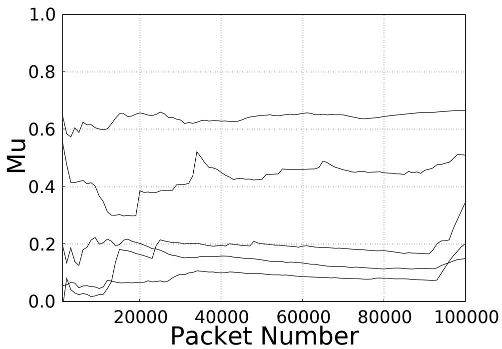


Figure 17: A set of representative links for Mirage channel 26 showing that in most cases the value of μ levels off as the number of packets exceeds 40000. There are a few outliers that have their μ change towards the very end of the data trace, but we see this in under 10% of all Mirage links.

100,000 packets each in chunks of 6000 packets and compute the deviation of the μ calculated for every chunk from the μ calculated over the entire trace. The overall deviation for a trace is calculated as the root mean square of the deviation values for all the chunks in a trace. Figure 16 shows the deviation values for different traces against their respective μ values. The deviation values in the figure show that chunks of 6000 packets in the traces give very different values for μ compared to those obtained for the whole 100,000 packet trace. This confirms the claim that 6000 packets are not enough for the values of μ and PRR to converge for wireless links.

While Figure 16 points to the fact that 6000 packets are not enough, it does not immediately show that longer traces do have converging values for μ. Therefore, we look at how μ changes as more and more packets are added to a trace. Using the empirical data from Mirage channel 26, we compute μ for the first 1000, 2000, 3000 up to 100000 packets. Overall, the results show that after about 40000 packets μ begins to stabilize.

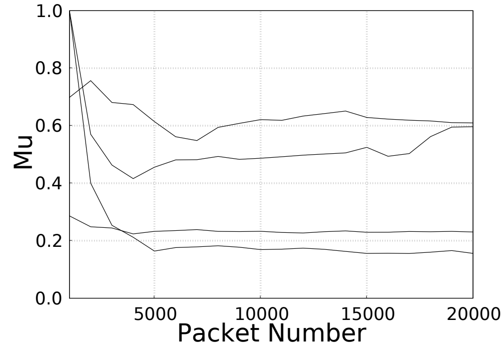


Figure 18: A set of representative links for Roofnet 11Mbps showing that in most cases the value of μ levels off as the number of packets exceeds 10000. There are a few outliers that have their μ change towards the very end of the data trace, but we see this in under 10% of all Roofnet links.

Figure 17 presents data from five links that serve as a representative subset of all Mirage links. Up until about 40000 packets, there is a significant variation in μ values that begins to level off afterwards. For three of the links in the figure, it is not hard to see that μ stabilizes; for the other two, however, kinks appear at the very end of the trace. This happens in the last 8000 packets or so. We observe similar behavior in under 10% of all empirical links, while all others follow the pattern of stabilizing after 40000 to 50000 packets. One possible explanation for the strange behavor of those outlier links is that they enter a more bursty state late in the data collection experiment, but since we stop observing them at 100000 packets we do not see the expected stabilization of μ. Figure 18 shows the representative links for the Roofnet testbed. In this case, we observe that the value of mu converges for trace lengths greater than 10000 packets.

The length of packet traces is critical for convergence of the parameter μ and the estimation of TCP throughput using μ and PRR. The 3000 packet long traces available for the Roofnet testbed running at 1Mbps are not long enough for the value of μ to converge because of which they have been omitted from the TCP analysis in this paper.

9. Modeling with μ

Previous sections showed that for both 802.11b and 802.15.4 datatraces, the values of μ have strong correlation with TCP throughput. A natural question is whether the Gilbert-Elliott model, which is the basis for μ, can be used to model links with behavior similar to that of empirically observed ones. This section explores whether given μ and reception ratio, the Gilbert-Elliott model simulates links accurately.

9.1 Gilbert-Elliott Synthetic Links

Section 5 presented how μ can be calculated from a data trace of packet receptions and failures. With the Gilbert-

Elliott model, the reverse is also possible. Knowing PRR and μ is enough to give the transition probabilities for the two-state Markov chain and these probabilities are enough to produce a simulated link of losses and successes. If the Gilbert-Elliot model is a valid approach to simulating links, we expect to see similar TCP performance for pairs of empirical and simulated links with equal PRR and μ values.

9.2 Results

Figure 19 compares the TCP throughput values for the empirical and simulated links for the different testbeds. In all three figures, the x-axis shows the empirical TCP throughput and the y-axis the throughput for the link generated by the Gilbert-Elliott model. Note that since the Roofnet data is every 2ms instead of 10ms, the y-axis goes up to 500 packets per second. The $x = y$ is shown for reference – datapoints on the line denote matched performance between simulated and real-world links. This figure shows that synthesized links perform very differently compared to the empirical data they try to simulate.

For the Mirage channel 26 dataset, most datapoints fall under the $x = y$ line; simulated links have considerably lower TCP throughput. This leads to the conclusion that links modeled using Gilbert-Elliott are not only quite different from empirical links, but also have a tendency to underperform. Only about 12% of all links are within 10% of their empirical partners. In the extreme cases, synthetic links can report a throughput which is as much as 80% less than that of real testbed measurements.

The results for Mirage channel 16 follow the same trend. About 50% of the links are within 10% of the empirical data traces, while many have lower TCP throughput, as much as 60% less. In the 802.11b experiments, we again see simulated links with throughput that greately differs from the expected – up to 80%.

Therefore while μ, as computed from a testbed trace, has a good predictive value for TCP throughput, the reverse is not true: simulation of a link using μ and the Gilbert-Elliott model is not accurate. More specifically, for our throughput metric, synthetic links consistently underperform compared to the empirical ones.

It must be that μ and TCP are correlated to a third variable, a characteristic that is currently missing from our observations. More specifically, μ does not capture information about infrequently occuring long bursts of successes. This is a key ingredient in understanding the throughput that TCP's windowing achieves due to the Additive Increase/Multiplicative Decrease scheme of congestion control. Very long bursts allow large size of the congestion window, larger flight size, and therefore, increased throughput.

Figure 20 shows the length of the longest burst of successes for each empirical link and its corresponding simulated link. If the Gilbert-Elliott model were to be used for simulating wireless link behavior, we would expect datapoints close to the 45-degree line. In reality, we discover

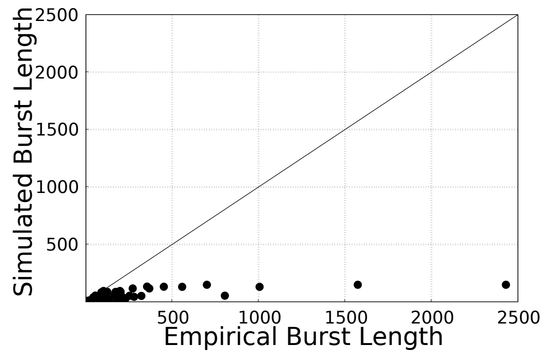


Figure 20: About 50% of empirical links in the Mirage channel 26 data have maximum burst lengths that are larger than the highest burst length the Gilbert-Elliott model ever produces.

that many empirical links have long bursts of successes that the Gilbert-Elliott model is unable to reproduce.

For example, over all 54 links from the Mirage channel 26 data, the longest burst created by the Gilbert-Elliott model is 149. At the same time, almost 50% of the empirical links have bursts much longer than that, up to 600 packets.

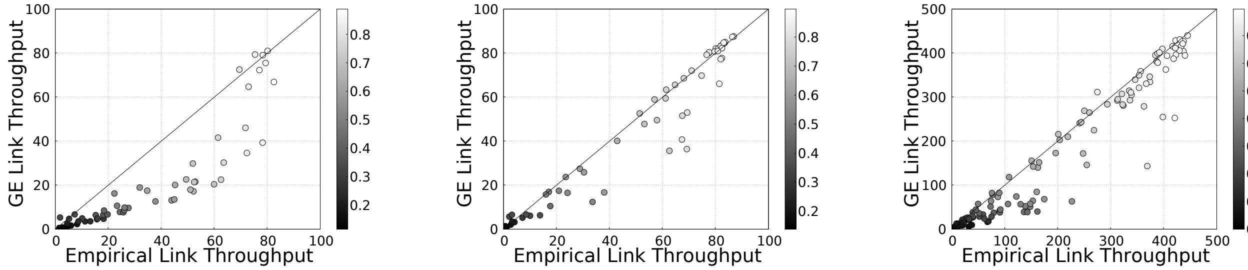
This observation explains why a Gilbert-Elliott synthetic link cannot achieve a TCP throughput as high as that of an empirical link. A single burst on the order of hundreds of packets can cause a big difference in TCP's windowing behavior. Therefore, to be able to model and simulate links' behavior in terms of burstiness, future work must combine the current version of μ with information about burst lengths.

10. Conclusion

This paper takes a first step towards extending the existing lexicon for describing wireless networks. While high-level metrics such as throughput and reception ratio can provide some coarse grained information, they are not enough to accurately describe a network. In Section 4 we saw that multiple TCP throughput values map to the same PRR, and multiple PRRs to the same observed TCP performance. Therefore, simply reporting one or the other can lead to misleading conclusions.

Observing links' burstiness, i.e. the temporal correlations of packet successes and losses, allowed us to refine our understanding of link behavior. We used μ as a metric of burstiness and discovered that, when combined with PRR, it can reduce estimation error of TCP throughput by approximately 50%. These results indicate that reporting a link burstiness metric along with other network parameters can give a better estimate of network performance.

While μ has a strong deductive value when computed on empirical links of intermediate reception ratios, it does not lend itself to simulations. Section 9 showed that using the Gilbert-Elliot model to generate synthetic links with given



(a) Mirage Channel 26 (b) Mirage Channel 16 (c) Roofnet

Figure 19: TCP throughput for the Gilbert-Elliott synthetic and empirical links for the three datasets. Most links generated using the Gilbert-Elliott model fail to achieve the TCP throughput that empirical links with the same PRR and μ have.

μ and PRR does not accurately simulate TCP throughput. There are dynamics in the links that these two parameters do not capture.

One characteristic of links that is not reflected in μ is the occurance of infrequent, very long bursts. Therefore, μ must evolve in a burstiness metric that captures more than just the number of state transitions; it needs to accurately reflect the distribution of burst lengths, including extremely long bursts. In addition, this new burstiness metric should be able to generalize for extremely low or high PRR values, which μ does not.

11. REFERENCES

[1] ANSI/IEEE Std 802.11 2007 Edition.

[2] ANSI/IEEE Std 802.15.4 2003 Edition.

[3] Routing over low power and lossy networks (roll). Charter, `http://www.ietf.org/html.charters/roll-charter.html`.

[4] D. Aguayo, J. C. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, pages 121–132, 2004.

[5] M. Allman, V. Paxon, and W. Stevens. TCP Congestion Control. Internet Network Working Group RFC2581, April 1999.

[6] E. Altman, K. Avrachenkov, and C. Barakat. Tcp in presence of bursty losses. *Perform. Eval.*, 42(2-3):129–147, 2000.

[7] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving tcp performance over wireless links. *IEEE/ACM Trans. Netw.*, 5(6):756–769, 1997.

[8] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, 2005.

[9] S. Biswas and R. Morris. Exor: opportunistic multi-hop routing for wireless networks. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, 2005.

[10] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007.

[11] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. Internet Network Working Group RFC2582, April 1999.

[12] S. Floyd and E. Kohler. Internet research needs better models. *SIGCOMM Comput. Commun. Rev.*, 33(1):29–34, 2003.

[13] E. N. Gilbert. Capacity of a burst-noise channel. *Bell System Tecnical Journal*, 39:1253–1266, 1960.

[14] A. Gurtov and S. Floyd. Modeling wireless links for transport protocols. *SIGCOMM Comput. Commun. Rev.*, 34(2):85–96, 2004.

[15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System Architecture Directions for Networked Sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000. TinyOS is available at http://webs.cs.berkeley.edu.

[16] U. o. S. C. Information Sciences Institute. Transmission Control Protocol. Internet Network Working Group RFC793, September 1981.

[17] Intel Research Berkeley. Mirage testbed. `https://mirage.berkeley.intel-research.net/`.

[18] C. Jiao, L. Schwiebert, and B. Xu. On modeling the packet error statistics in bursty channels. In *LCN '02: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, pages 534–541, Washington, DC, USA, 2002. IEEE Computer Society.

[19] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: practical wireless network coding. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, 2006.

[20] A. Köpke, A. Willig, and H. Karl. Chaotic maps as parsimonious bit error models of wireless channels. In *Proc. of IEEE INFOCOM*, San Francisco, USA, Mar. 2003.

[21] A. Kumar. Comparative performance analysis of versions of tcp in a local network with a lossy link. *IEEE/ACM Trans. Networking*, August 1998.

[22] T. Lakshman and U. Madhow. The performance of tcp/ip for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. Networking*, June 1997.

[23] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Simulating large wireless sensor networks of tinyos motes. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, 2003.

[24] M. Handley and S. Floyd and J. Padhye and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specifications. Request For Comments: 3448, January 2003.

[25] N. Kushalnagar and G. Montenegro and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. Request For Comments: 4919, August 2007.

[26] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe. Modeling TCP throughput: A simple model and its empirical validation. *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 303–314, 1998.

[27] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. In *SIGCOMM*, pages 51–62, 2006.

[28] P. M. Soni and A. Chockalingam. Analysis of link layer backoff schemes on point-to-point markov fading links. *IEEE Trans. on Communications*, 51(1):29–32, 2003.

[29] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Some implications of low power wireless to ip networking. In *The Fifth Workshop on Hot Topics in Networks (HotNets-V)*, Nov. 2006.

[30] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. Internet Network Working Group RFC2001, January 1997.

[31] M. Zorzi, A. Chockalingam, and R. R. Rao. Throughput analysis of tcp on channels with memory. *IEEE Selected Areas In Communications*, July 2000.

[32] M. Zorzi, R. R. Rao, and L. B. Milstein. On the accuracy of a first-order markov model for data transmission on fading channels. In *Proc. IEEE ICUPC'95*, 1995.