Packet Clustering Introduced by Routers: Modeling, Analysis and Experiments

Chiun Lin Lim[1], Ki Suh Lee[2], Han Wang[2], Hakim Weatherspoon[2], Ao Tang[1]
[1] School of Electrical and Computer Engineering, Cornell University
[2] Department of Computer Science, Cornell University
cl377@cornell.edu, kslee, hwang, hweather@cs.cornell.edu, atang@ece.cornell.edu

***Abstract*—Utilizing a highly precise network measurement device, we investigate router's inherent variation on packet processing time and its effect on interpacket delay and packet clustering. We propose a simple pipeline model incorporating the inherent variation and a metric to measure packet clustering. To isolate the effect of the inherent variation, we begin our analysis with no cross traffic and step through setups where the input streams have different data rate, packet size and go through different number of hops. We show that a homogeneous input stream with a sufficiently large interpacket gap will emerge at the router's output with interpacket delays that are negative correlated with adjacent values and have symmetrical distributions. We show that for smaller interpacket gaps, the change in packet clustering is smaller. It is also shown that the degree of packet clustering could in fact decrease for a clustered input. We generalize our results by adding cross traffic. We apply these results to demonstrate how we could reduce jitter by minimizing interpacket gap. All the results predicted by the model are validated with experiments with real routers.**

***Index Terms*—Router modeling, traffic burstiness, queuing theory**

I. Introduction

For real-world network traffic, the observation that packets tend to cluster together or become bursty after passing through one or multiple routers is well-documented for several timescales [2], [1], [11], [9]. On longer timescales, proposed explanations for burstiness are centered on input traffic characteristics such as the distribution of user's idle and active time [17], the distribution of file sizes [5], and TCP congestion control [9], [16]. At the packet level, the clustering could be attributed to contention and scheduling with cross-traffic at the switching fabric and timing variation in routing table lookup. Is there any other factor that can cause bursty traffic, maybe on an even finer timescale? Consider an experimental setup where all the above mentioned factors are not present. Let a single packet stream with fixed packet size, constant interpacket gap, as well as identical destination goes through a single isolated and idle router with no cross traffic. One would expect the router's output packet stream to have the same constant interpacket gap as the input stream. However, it has been observed that the interpacket gap is not constant but instead exhibits some variation on the order of 100 ns at the output, and when the experiment is repeated for various interpacket delay constants, the variation is observed to be sufficient to induce packet clustering in some experiments [8]. Moreover, if the input stream goes through multiple routers, the clustering effect can be more prominent.

In the absence of external factors, such variation could only be explained by factors inherent in the router design itself. Possible explanations include clock drift, buffering scheme and quantization of packets into cells [3]. Our goal in this paper, though, is *not* on the causes of the variation, but on *the effect of the router's inherent variation on interpacket delay and packet clustering*. Such a fine-scale investigation was not feasible experimentally prior to [8], as network measurement devices have significant measurement error, see e.g. [14]. BiFocals as introduced in [8] allows exact timing measurement of network packet streams by directly capturing the physical layer symbol stream in real-time and time-stamping in off-line post-processing. The time-stamps are exact since the precision of the device is smaller than the width of a single symbol.

In this paper, we propose a simple device-independent model incorporating a router's inherent variation (Section II-A). We next propose a metric to quantify packet clustering (Section II-B) and show how packet clustering in terms of this metric changes as it passes through one or multiple router (Section III and IV). We then generalize our results by incorporating cross traffic and interpret the results obtained under the context of jitter and show how to control jitter (Section V). Finally, we verify all the model results with repeatable experiments (Section VI).

II. Problem Setup

A. Model

We first establish some terminologies. The interpacket delay (IPD) is the space, in bits, between the first bit of a packet and the first bit of the subsequent packet. The interpacket gap (IPG) is the space between the last bit of a packet and the first bit of the subsequent packet, i.e. $\text{IPG} = \text{IPD} - \text{packet size}$. We say that a stream of packets is *homogeneous* if all the packets have the same size, the same interpacket delay, and are heading for the same destination. We use shorthand such as 1526B 3G to refer to a homogenous packet stream with 1526-byte Ethernet frames and 3 Gbps data rate. We assume the input packet stream is homogeneous; though all our results in this section hold as long as the IPD is independent and identically distributed (i.i.d.).

Each packet transitioning through the router experiences a service time, which is the sum of various delays related to processing time and and transmission time. Typically the constant transmission time is the dominant delay while other delays are shorter and could have some inherent variation.

978-1-4799-3001-2/14/$31.00 ©2014 IEEE

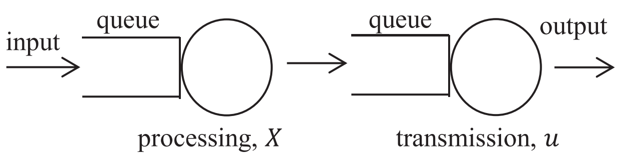

Figure 1. A two-server model of a router with the total service time S being a sum of X, a random processing time and u, a constant transmission time

Without loss of generality, we group all the possible sources of variation together and refer to it as part of processing time. The model then is simple — the total service time, S, of a packet going through a router is the sum of delays through two servers:

$$S = X + u$$

In the first server, the packet takes a random processing time, $X \geq 0$ to be processed. In the second stage, the packet takes a constant transmission time, u to be transmitted (see figure 1). The transmission time is simply the packet size divided by capacity rate we will sometimes refer to u as the packet size. This two-server model means that it is possible for parallel processing and transmission of different packet, i.e. pipelining, to occur.

Each packet of the input stream is indexed with i and its associated processing time, X_i is assumed to be i.i.d. We denote the interpacket delay between packet i and $i+1$ with the random variable D_i. We assume packet arrival rate is smaller than the capacity rate, i.e. $E[D] > u$ and packets are processed faster than it can be transmitted, i.e. $u > E[X]$. We also assume there is sufficient buffer such that neither the processing server nor the transmission server ever overflows. In short, we have modeled the router with two serial servers with deterministic arrivals, independent service time and a first-in, first-out queue discipline.

With this simple model, we are going to analyze how interpacket delay varies as the input stream passes through a router. But first we clarify the notational convention of this paper. We use superscript on the variables to denote router number, e.g. D_i^1 is the ith interpacket delay after going through the first router. The 0 superscript refers to the input packet stream. Since we will have expressions involving exponents, except for 0 or 1, all other integer superscripts should be interpreted as exponents, e.g. D_i^2 is the square of the ith interpacket delay and not the ith interpacket delay after passing through 2 routers. The superscript is sometimes omitted when we are referring to interpacket delay in general while the subscript is sometimes omitted when the statement is applicable to all packets. We denote $E[\cdot]$ as the expectation function.

B. Packet Clustering Metric

We propose a metric to represent the degree of packet clustering. To start, given that the packet arrival rate is smaller than the router capacity rate, we know the average input data rate is the same as the average output data rate. This simple observation implies that the average IPD is the same before and after going through a router. Armed with this observation, we consider the simplest setup with a sequence of 3 packets as shown in Figure 2 to gain intuition. Since the average IPD is the same, the only variable here is the relative position of the second packet. Intuitively, the least clustering setup is when the packets are uniformly spaced, i.e. the IPG between packet 1 and 2 is the same as between packet 2 and 3. Packets are seen to be more clustered as packet 2 is closer to either packet 1 or 3 and the packets are most clustered when two packets have minimal IPG in between.

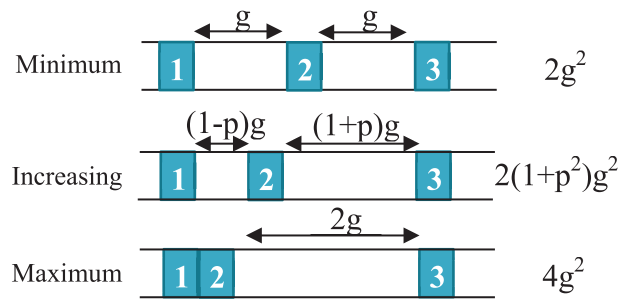

Figure 2. An intuitive figure depicting how packet clustering evolves as interpacket delay changes. The square of the interpacket gap fits the intuitive notion.

A metric that fits the intuition is the sum of the square of the IPG. The packet clustering metric, c, given the IPG of all packets, g is given by

$$c(g) = \frac{1}{n}\sum_{i=1}^{n} g_i^2 \quad (1)$$

where we have normalized by n, the number of IPGs.

Next, we want to obtain an equivalent metric that is expressed in terms of IPD instead of IPG. To do so, we first clarify what we mean by an equivalent metric. A metric is equivalent to another if it fits the intuitive notion of packet clustering in Figure 2 or more formally, if it is order-preserving, i.e. if packet stream a is rated as more clustered than packet stream setup b under the first metric, then the second metric should also rate packet stream a as more clustered than b. We leave the verification of equivalent metrics for the rest of this subsection to the reader.

We claim that since the packet size is constant, an equivalent metric is the sum of the square of the IPD $c(d) = \sum_{i=1}^{n} d_i^2/n$. The metric $c(d)$ computes a value when each D_i takes on a particular value d_i. Often times, we are interested with D_i in general, in which case, the expected packet clustering metric, $E[c(D)]$ is more relevant. Using the identity $\text{var}(D) = E[D^2] - E[D]^2$ and assuming the interpacket delays are i.i.d., we obtain $\text{var}(D)$ as an equivalent metric, and this is the metric that we will use to analyze packet clustering for the rest of the paper. In short, the metric states that a packet stream is more clustered if its IPD is more variable. Thus, for a homogeneous packet stream, since the IPD is constant, $\text{var}(D) = 0$, which conforms with our intuition that the homogenous packet stream is the least clustered.

For the rest of the paper, for conciseness, we refer to (expected) packet clustering metric as simply packet clustering. We note here that our proposed metric is not the only one that fits the intuitive notion of packet clustering in Figure 2. Other convex metrics are possible and we choose the current metric due to its relative ease of analysis.

III. The Single-Hop Case

We begin our analysis with the single hop case. There are two regions to consider here: when all packets have no waiting time (*large interpacket gap)* and when some of them do (*small interpacket gap*).

A. Large interpacket gap

With no waiting time, the interpacket delay after one router is relatively straightforward to figure out. Consider the sequence of discrete events that occur on packet 1 and 2. Packet 1 arrives at the router, receives service for a time period of $S_1^1 = X_1^1 + u$ and leaves the router. The second packet then arrives and receives service for a time period of $S_2^1 = X_2^1 + u$. The interpacket delay after 1 router for packet 1 and 2 is thus $D_1^1 = D_1^0 - S_1^1 + S_2^1$. One could continue by considering packets $3, 4, 5$ and so on to find that the same results hold:

$$D_i^1 = D_i^0 - S_i^1 + S_{i+1}^1 = D_i^0 - X_i^1 + X_{i+1}^1,\ i = 1, 2, \ldots \quad (2)$$

While equation (2) looks simple, we can extract plenty of information: symmetry of interpacket histogram (Lemma 1 and Theorem 2), and negative correlation of adjacent interpacket delay (Theorem 1).

To prove symmetry, we say that a random variable D has a distribution that is symmetrical about d if $D - d$ has the same distribution as $-(D - d)$, which we write as

$$D - d \sim -(D - d)$$

For such a random variable, its probability density function is symmetrical about d.

Lemma 1. *If the distribution of D^0 is symmetric about $E\left[D^0\right]$, then the distribution of $D_i^0 - X_i + X_{i+1}$ is symmetric about $E\left[D^0\right]$ for all i.*

Proof: Due to space restriction, all the proofs in this paper are omitted and could instead be found in [7]. ∎

However, Lemma 1 is not sufficient to prove symmetry, as the interpacket delay sequence $D_1^1, D_2^1, \ldots$ is not i.i.d. Neighboring terms of the sequence are correlated: $D_1^0 - X_1^1 + X_2^1$ is not independent of $D_2^0 - X_2^1 + X_3^1$ due to the X_2^1 term. Since the interpacket delays are correlated via the X_2^1 term, which has an opposite sign in each of the interpacket delay, we expect the correlation to be negative.

Theorem 1. *After one hop, the correlation coefficient between two neighboring interpacket delay is -1/2, that is*

$$\rho_{i,j} \triangleq \frac{cov\left(D_i^1, D_j^1\right)}{\sqrt{var\left(D_i^1\right) var\left(D_j^1\right)}} = -\frac{1}{2},\ for\ \ |i-j| = 1 \quad (3)$$

Even though the interpacket delay is negatively correlated, we can still prove symmetry by observing that every other term of the interpacket delay sequence is i.i.d and by using superposition.

Theorem 2. *The interpacket delay histogram of a homogeneous packet stream at a router's output is symmetrical about $E\left[D^0\right]$.*

Note that Lemma 1, Theorem 1 and Theorem 2 all make heavy use of the assumption that the random processing time X is i.i.d. Thus, if experimental data turns out to match the analysis here, then it is reasonable to adopt the i.i.d. assumption. We will see in Section VI that this is indeed the case.

We now switch attention to packet clustering. Since D_i^0, X_i^1 and X_{i+1}^1 are all independent of each other

$$\text{var}\left(D^1\right) = \text{var}\left(D^0\right) + 2\text{var}\left(X^1\right) \quad (4)$$

For a homogeneous input packet stream, D^0 is a constant and thus $\text{var}\left(D^1\right) = 2\text{var}\left(X^1\right)$.

B. Small interpacket gap

The interpacket gap in this region is small while the service time is sometimes long enough to induce a waiting time in the next packet. We define I_i as the processing server idle time in between packet i and $i+1$ and W_i as the waiting time of packet i at the processing server. By considering the sequence of discrete events that occur on packet i and $i+1$, we find that the idle time is given by

$$I_i^1 = \max\left(0, D_i^0 - W_i^1 - X_i^1\right) = \left(D_i^0 - W_i^1 - X_i^1\right)^+ \quad (5)$$

where $a^+ = \max(0, a)$. Similarly, the waiting time of packet $i+1$ is given by

$$W_{i+1}^1 = \max\left(0, W_i^1 + X_i^1 - D_i^0\right) = \left(D_i^0 - W_i^1 - X_i^1\right)^- \quad (6)$$

where $a^- = \max(0, -a)$. In the queuing theory literature, equation (6) is also known as the Lindley equation [12]. The IPD after one-hop is given by

$$D_i^1 = \max\left\{u, I_i^1 + X_{i+1}^1\right\} \quad (7)$$

The identity $a = a^+ - a^-$ gives us

$$D_i^0 - W_i^1 - X_i^1 = I_i^1 - W_{i+1}^1 \quad (8)$$

Combining equation (7) and (8) gives

$$D_i^1 \geq D_i^0 - \left(W_i^1 + X_i^1\right) + \left(W_{i+1}^1 + X_{i+1}^1\right) \quad (9)$$

Recall that for the large interpacket gap region, equation (4) tells us that packet clustering increases by $2\text{var}\left(X^1\right)$. For the small interpacket gap region, some packets are prevented from getting closer together due to the physical requirement of a minimum service time. This implies that intuitively, packet clustering after one router should increase to a value that is less than $2\text{var}\left(X^1\right)$.

Theorem 3. *For a packet stream with i.i.d. interpacket delay and $n \to \infty$ number of packets, after one hop,*

$$var\left(D^1\right) \leq var\left(D^0\right) + 2\left(var\left(X^1\right) - E\left[W^1\right]E\left[I^1\right]\right) \quad (10)$$

Since $E\left[W^1\right]E\left[I^1\right] > 0$, Theorem 3 confirms our intuition that packet clustering increases by a value less than $2\text{var}\left(X\right)$. The theorem also tells us that it is possible for the packet stream to be less clustered if the input packet stream is not homogeneous and $E\left[W^1\right]E\left[I^1\right] > \text{var}\left(X\right)$. However, we may not be able to check for it in practice, since we may not have sufficient knowledge or access to determine $E\left[W^1\right]$ and $E\left[I^1\right]$. We thus have the question: for an input packet stream with i.i.d. IPD, is there an easily verifiable condition to know when the packet stream will be less clustered?

Corollary 1. *An input packet stream with i.i.d. interpacket delay will be less clustered after going through a router if* $2var\left(X^1\right) \le E\left[\left\{\left(D^0 - X^1\right)^-\right\}^2\right]$.

Note that $E\left[\left\{\left(D^0 - X^1\right)^-\right\}^2\right]$ is large if D^0 has a large probability of taking small values, i.e. a large portion of packets are close together. There are two scenarios for the packet to be closer together: one, the input packet stream is getting more clustered and two, the input data rate is getting higher, as higher data rate implies smaller interpacket delay. Since $\text{var}\left(D^0\right) + 2\text{var}\left(X^1\right) - E\left[\left\{\left(D^0 - X^1\right)^-\right\}^2\right]$ is an upper bound for $\text{var}\left(D^1\right)$, the upper bound is getting smaller as either scenario occurs. As such, it is reasonable to expect that *the change in packet clustering,* $var\left(D^1\right) - var\left(D^0\right)$ *to decrease as the input packet stream is getting more clustered or as the input data rate increases* (see Figure 5). In cases where the input packet stream is very clustered, we have shown in Corollary 1 that the change in packet clustering could in fact be negative (see Figure 4)

IV. The Multi-Hop Case

In this section, we extend our results to the multiple-router scenario. We denote m as the number of routers and the routers are not assumed to be identical unless otherwise stated. There are now 3 regions to consider: when packets have no waiting time for all m routers (*large interpacket gap)*, when packets have no waiting time for first $j < m$ routers (*medium interpacket gap)* and when some packets have positive waiting time while passing through each router (*small interpacket gap*).

A. Large interpacket gap

The analysis is this subsection is mostly a straightforward generalization of the results of the single-hop case. Apply equation (2) repeatedly to obtain, for all $l \le m$

$$D_i^l = D_i^0 - \sum_{k=1}^{l} X_i^k + \sum_{k=1}^{l} X_{i+1}^k \tag{11}$$

The other results concerning negative correlation and symmetry of the IPD histogram generalize in the same manner. In addition, we have a new result concerning how packet clustering changes as it passes through the routers. We state them all formally in the following theorem.

Theorem 4. *In the large interpacket gap region, for any* $1 \le l \le m$, *the interpacket delay sequence* $D_1^l, D_2^l, \ldots$ *has the following 3 properties: adjacent interpacket delay has a correlation coefficient of* $-1/2$, the IPD histogram at the final router's output is *symmetrical and if the* m *routers are identical, then packet clustering increases linearly with the number of routers.*

B. Medium and small interpacket gap

For medium interpacket gap, for the first j routers where there is no waiting time, packet clustering increases linearly and for the remaining routers, the change in packet clustering would be similar to the small interpacket gap region, which we will analyze now.

For small interpacket gap, the results are not as easy to generalize from the single hop case as large interpacket gap. The main difficulty arises from the correlated interpacket delay after the first hop. Such correlation implies that the setup may not converge to a steady state distribution even if the input packet stream is sufficiently long. As such, we approximate by ignoring the correlation and assuming that the input to all the routers have i.i.d. interpacket delay and our aim in this subsection is not to derive analytical expressions but to use the results from Section III-B to argue qualitatively how packet clustering would evolve with increasing number of identical hops.

Recall that after Corollary 1 we argue that the change in packet clustering, $\text{var}\left(D^1\right) - \text{var}\left(D^0\right)$ would decrease as the input packet stream is getting more clustered or as the input data rate increases. Given the i.i.d. input assumption to all routers, we could now apply this argument at each hop. This means that if we start with a homogeneous input packet stream, then packet clustering would be an increasing and concave function in the number of identical hops. Conversely, if we start out with a very clustered input packet stream, then packet clustering would be a decreasing and convex function in the number of identical hops (see Figure 4).

The second implication is that if we have two input packet streams with equal packet clustering but different data rate, then for the input stream with a higher data rate, its rate of change for packet clustering in the number of identical hops would be lower. For instance, suppose we have two homogeneous packet stream with 4G and 6G data rate. Then while packet clustering would evolve in an increasing and concave manner in the number of identical hops for both, the packet clustering function for 4G should be strictly above that of 6G's (see Figure 5).

V. Adding Cross Traffic

In this section, we add in cross traffic and show that the results and discussions from Section III carry over. We need new terms and assumptions with the addition of cross traffic. We call the packet stream that we are tracking as it goes through the router the *target traffic* while any other traffic that interferes with it, the *cross traffic*. It is well-known that Internet traffic is usually not Poisson in nature and is positively correlated over time [11]. For such cross traffic, the setup that we are currently analyzing does not necessarily converge to the steady state distribution. To make the analysis tractable, we make the assumption that the cross traffic is not time varying, i.e. stationary, and the current setup does converge to the steady state distribution. We assume packets are served in a first-in, first-out manner. This is again a simplifying assumption as packets that arrive at different router input ports typically have to undergo contention and scheduling and the final packet service order is not necessarily first-in, first-out.

There are two key steps to analyzing the interaction of the two types of traffic. The first step is to divide our analysis into two stages by first finding the interpacket delay after passing through the processing server, Δ^1, before moving on to deal with the transmission server. The second step is to further subdivide the waiting time and idle time by source. The waiting time of the ith target packet at the processing server is now $W_i^p = T_i^p + C_i^p$ where T_i^p is the waiting time incurred on the ith target packet till target packet $i-1$ is processed and C_i^p is the waiting time incurred on the ith target packet due to cross traffic that is processed after target packet $i-1$. We define I_i^p as the idle time the processing server spends not processing any target traffic after the departure of the ith packet and before the arrival of the $(i+1)$th packet. Note that it is possible for the processing server to be processing cross traffic during such idle time. The terms W_i^r, T_i^r, C_i^r and I_i^r are defined analogously for the transmission server. To prevent notational clustering, we will drop the superscript 1 from all non interpacket delay terms in this section.

The analysis to derive the interpacket delay and packet clustering follow the same path as the model with no cross traffic. Due to space restriction, we skip the intermediate analysis and instead jump straight to the final result.

Theorem 5. *With cross traffic, the change in packet clustering, var $\left(D^1\right) -$ var $\left(D^0\right)$ is upper bounded by*

$$2\left[var\left(X\right)+var\left(C^p\right)+cov\left(I^p,C^p\right)+var\left(C^r\right)+cov\left(I^r,C^r\right)\right] - E\left[\left\{\left(D^0-X-C^p\right)^-\right\}^2\right] - E\left[\left\{\left(\Delta^1-u-C^r\right)^-\right\}^2\right] \quad (12)$$

Similar to the case with no cross traffic, the last two terms on the RHS of (12) is large if D^0 has a large probability of taking small values, and as such, the observation from the end of Section III-B carry over and it is reasonable to expect that even with cross traffic, the change in packet clustering, var $\left(D^1\right) -$ var $\left(D^0\right)$ to decrease as the input packet stream is getting more clustered or as the input data rate increases. We now apply the observation that we just made to understand jitter.

There exists several definitions for jitter and the one we are going to look at is interpacket delay variation (IPDV) as defined in RFC5481 [15]. More specifically, we are investigating the variance of IPDV, which we will refer to interchangeably with jitter. In the notation of this paper, IPDV is defined as $D_i^n - D_i^0$ and thus for homogeneous input traffic, its variance is simply var $\left(D_i^n\right)$ and we could apply the results that we have so far in the context of jitter.

From the discussion that we just had, we know that jitter would grow slower or even decrease if the interpacket gap in between packets are smaller. The phenomena of decreasing jitter with smaller interpacket gap has been observed in [6], albeit the definition of jitter used is different. Thus, to control jitter, we need to decrease interpacket gap, which could be achieved in three ways. The first is by sending at a higher data rate, which will increase the end-to-end delay as a tradeoff. Alternatively, we could send with smaller packet sizes, though this means we have to send more packets and thus more data overhead in terms of packet headers. Finally, we could send the traffic via a dedicated, rate-limited tunnel through the network though setting up such a tunnel is expensive. Each method of controlling jitter comes with its own tradeoff, and we leave a more thorough investigation for future work.

VI. Experimental Validation

A. Experiment Setup

To validate our model, we use SoNIC [10], which is a software-defined network interface card that achieves the same level of precision as BiFocals. We deploy a SoNIC board on a Dell T7500 workstation, and use a Cisco Catalyst 6500 router. We mention here that the results are consistent in all other routers where the experiments are repeated. This includes Cisco 4948, IBM G8264 and HP Procurve 2900.

B. Validation

Experiment 1: Negative correlation of adjacent interpacket delays (Theorem 1). The correlation coefficient is computed using 1 million IPDs obtained from experiment for various setups. The results are summarized in Table I. We can see that the computed values are close to the value of -0.5. The The 3 starred values are setups that belong to the small interpacket delay region.

Experiment 2: IPD properties for the multi-hop, large IPG setup (Theorem 4). For the large interpacket gap region, a 1526B 1G homogeneous packet stream goes through 8 identical routers and the IPD is recorded. The IPD histogram is observed to be symmetrical and packet clustering grows linearly. In addition, the correlation coefficient between adjacent IPD is calculated as -0.4858, which is close to the theoretical value of -0.5.

Experiment 3: Packet clustering could decrease after passing through one router (Corollary 1 and Theorem 5). We verify that for a very clustered input packet stream, packet clustering decreases after the packet stream passes through the

	1526-byte packet	72-byte packet
1 Gbps	-0.4942	-0.5022
3 Gbps	-0.5494	-0.8673*
6 Gbps	-0.4740	-0.1788*
9 Gbps	-0.4931	-0.6924*

Table I
Correlation coefficient values

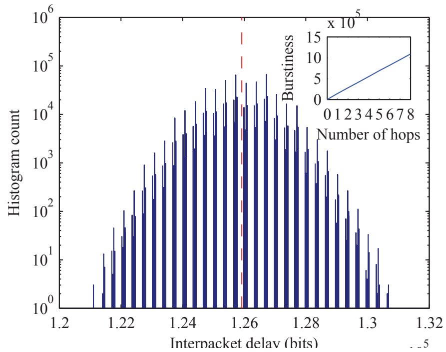

Figure 3. Interpacket delay after 8 hops for homogeneous input traffic of 1526-byte packet and 1 Gbps data rate. Inset shows the linear growth of traffic burstiness.

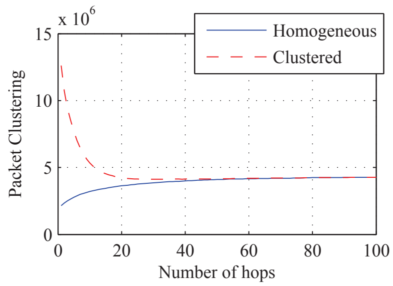

Figure 4. How packet clustering evolves over multiple routers for clustered vs. homogenous packet stream. Both packet streams are 72B 3G.

router. Without cross traffic, the result is shown in Figure 4. The clustered packet stream has 10 packets clustered together with minimal IPG and one huge gap before the next cluster. As a comparison, we also plotted the change in packet clustering for a homogenous packet stream with the same packet size and data rate. Notice that as the number of hops increases, packet clustering varies in an increasing and concave manner for the homogeneous packet stream, and decreasing and convex for the clustered packet stream. This agrees with the discussion in Section IV-B.

Experiment 4: How packet clustering evolves for different data rates with increasing number of hops (Section IV-B). Figure 5 shows experimental data for fixed packet size but varying data rates. For the first few hops, packet clustering evolves almost linearly for all data rates. As the number of hops increases, the increment decreases at a faster rate for the higher rate. The figure tells us that for increasing data rates, packet clustering increases at a decreasing rate, in agreement with the discussion in Section IV-B.

VII. Conclusion and Future Works

Our model has formed a framework for understanding how inherent variation in a router affects the input-output characteristic of a router. There are plenty of interesting research directions that we could pursue with this paper as a starting point. On the practical side, we could delve into the workings of a router, try to understand how inherent variation comes about and build a practical router model in the same spirit as

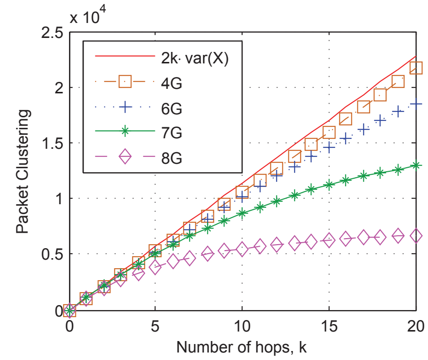

Figure 5. How packet clustering evolves as a homogenous packet stream of 520-byte packets passes through different number of routers for various data rates.

[4] but which incorporates inherent variation as an important parameter. On the application side, as mentioned before, we could investigate deeper into the tradeoffs between the various methods of controlling jitter by minimizing interpacket gap. We could also look at bandwidth estimation using packet-train dispersion [13] and see if our analysis could help in improving the accuracy of existing bandwidth estimation methods.

References

[1] Jean-Chrysotome Bolot. End-to-end packet delay and loss behavior in the internet. *SIGCOMM*, 1993.

[2] A. Broido, R. King, E. Nemeth, and KC Claffy. Radon spectroscopy of inter-packet delay. In *HSN Workshop*, 2003.

[3] H.J. Chao and B. Liu. *High Performance Switches and Routers*. Wiley, 2007.

[4] R. Chertov and S. Fahmy. Forwarding devices: From measurements to simulations. *ACM T. Model Comput. S.*, 21(2):12, 2011.

[5] M.E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, 1997.

[6] H. Dahmouni, A. Girard, and B. Sansò. An analytical model for jitter in IP networks. *Ann. Telecommun.*, 67(1-2):81–90, 2012.

[7] C.L. Lim et. al. Packet clustering introduced by routers. Technical report, 2013. http://networks.ece.cornell.edu/tech/Lim13TR.pdf.

[8] D.A. Freedman, T. Marian, J. H. Lee, K. Birman, H. Weatherspoon, and C. Xu. Exact temporal characterization of 10 Gbps optical wide-area network. In *IMC*, 2010.

[9] H. Jiang and C. Dovrolis. Why is the internet traffic bursty in short time scales. In *Sigmetrics*, 2005.

[10] K.S. Lee, H. Wang, and H. Weatherspoon. SoNIC: Precise Realtime Software Access and Control of Wired Networks. In *NSDI*, 2013.

[11] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of ethernet traffic. In *SIGCOMM*, 1993.

[12] D.V. Lindley. The theory of queues with a single server. In *Math. Proc. Carmbridge*, volume 48, pages 277–289. Cambridge University Press, 1952.

[13] X. Liu, K. Ravindran, and D. Loguinov. A queueing-theoretic foundation of available bandwidth estimation: Single-hop analysis. *IEEE/ACM Trans. Netw..*, 15(4):918 –931, aug. 2007.

[14] T. Marian, D.A. Freedman, K. Birman, and H. Weatherspoon. Empirical characterization of uncongested optical lambda networks and 10gbe commodity endpoints. In *DSN*, 2010.

[15] A. Morton and B. Claise. RFC 5481: Packet delay variation applicability statement. 2009.

[16] A. Tang, Andrew L., Jacobsson K., Johansson K., Hjalmarsson H., and Low S. Queue dynamics with window flow control. *IEEE/ACM Trans. Netw.*, 18(5):1422–1435, 2010.

[17] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Trans. Netw.*, 5(1):71–86, 1997.