Packet Clustering Introduced by Routers: Modeling, Analysis, and Experiments

CHIUN LIN LIM, KI SUH LEE, HAN WANG, HAKIM WEATHERSPOON, and AO TANG,
Cornell University

In this article, we investigate a 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 two metrics—one to measure packet clustering and one to quantify inherent variation. 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 rates, packet size, and go through a 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. All the results predicted by the model are validated with experiments with real routers. We also investigated several factors that can affect the inherent variation as well as some potential applications of this study.

CCS Concepts: • **Networks → Network performance modeling**;

Additional Key Words and Phrases: Packet clustering, queueing, processing delay

ACM Reference format:
Chiun Lin Lim, Ki Suh Lee, Han Wang, Hakim Weatherspoon, and Ao Tang. 2019. Packet Clustering Introduced by Routers: Modeling, Analysis, and Experiments. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 4, 3, Article 15 (August 2019), 28 pages.
https://doi.org/10.1145/3345032

1 INTRODUCTION

1.1 Motivation

It is well known that the transition time or delay through a router for a packet can be divided into three components—transmission delay, queuing delay, and processing delay. Transmission delay is the time to transmit all bits of a packet onto a data link, queuing delay is the time a packet spent in buffer while waiting for processing and transmission, and processing delay is the time to process the packet to determine the output port to transmit it. Transmission delay is deterministically determined by packet size and link capacity; queuing delay is variable as it depends on load,

Authors' addresses: C. L. Lim and A. Tang, Cornell University, School of Electrical and Computer Engineering, Ithaca, NY, 14853; emails: cl377@cornell.edu, atang@ece.cornell.edu; K. S. Lee, H. Wang, and H. Weatherspoon, Cornell University, Department of Computer Science, Ithaca, NY, 14853; emails: {kslee, hwang, hweather}@cs.cornell.edu.

2376-3639/2019/08-ART15 $15.00
https://doi.org/10.1145/3345032

congestion, and contention at the switching fabric; and processing delay is variable as tasks such as forwarding table lookup and quantization of packets into cells [7] could be variable.

Traditionally, the variation in processing delay is neglected and thus processing delay is usually modeled as deterministic. This is fine when the magnitude of such variation is much smaller than the magnitude of variation due to queueing delay introduced by cross traffic. When the higher-order variation is absent, then variation due to processing delay becomes the only source of variation. Such a situation could happen in various ways. First, it is common in networks where there are only one or several persistent connections, e.g., last mile wired transmission and home network. As such, the variation introduces additional jitter for multimedia and Real-time Transport Protocol (RTP) [32]. Second, it could happen in protocols involving pre-negotiated resources such as Diffserv [3], where prioritized traffic could travel through the network without contention from other traffic classes. The prioritization is meant to produce traffic with characteristics of low delay, low loss, and low jitter. The variation introduces a lower bound for loss and jitter.

More importantly, if processing delay variation stays the same or decreases slower than network speed improvement, then it will become an increasingly notable factor to network performance. For instance, facing the increasingly high performance requirements such as low latency and high throughput, especially for datacenter networks, people have started to reconsider basic network architectures that are inherited from the Internet and often cannot meet performance requirements. In the ideal case, for each packet, one would like to control when it enters the network and which route it takes [30]. In this case, traffic at any point of the network can be controlled with a very fine granularity. However, the key underlying assumption is that one can predict traffic at any point of the network at anytime once input is completely controlled. It is clear that this argument will have problems if one gets to the timescale where the randomness of interpacket delay cannot be ignored. Based on the results in this article, for the current technologies, the variation of interpacket delay can easily be on the order of a few hundred nanoseconds, which makes per packet control feasible for 10Gbits/s network since a standard 1500-byte MTU-sized packet takes 1,230 nanoseconds transmission time there as rightly pointed out in Ref. [30]. It, however, will not be true once the transmission rate increases to 40Gbits/s, 100Gbits/s, or even higher unless the interpacket delay variation can be better controlled though new router technology.

Another example is about realizing ultimate low jitter, which can be critical for real-time applications. The pursuit for low jitter has been around for quite a while, starting with constant bit-rate applications in ATM networks [25, 31] in the 90s, to video applications of the past decade, and more recently, to VR/AR applications. As we will show in this article, as packets go through multiple hops, jitter could grow slower or even decrease if the interpacket gap between packets are small. The phenomenon of decreasing jitter with small interpacket gap has been observed in Ref. [10]. 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 this article's results can provide a solid foundation for evaluating such tradeoffs.

The interpacket gap can have other uses as well. Covert channels are communications channels that are able to hide their existence within existing network protocols [37]. A physical layer covert timing channel could be created by modulating the interpacket gap in between certain packets in a homogeneous packet stream. The key here is that for a less precise measurement instrument, in the absence of inherent variation, the modulation could be detected; but with inherent variation, the modulated packet stream is indistinguishable to an unmodulated packet stream (see Figure 9 of

Ref. [19]). Recent developments show that the modulation could be decoded with a low-bit error rate even with cross traffic and the higher order perturbation effects. Readers interested in the detail should refer to Ref. [19].

In this work, we examine the variation of packet processing delay and its effect on traffic behavior. In particular, we focus on how such variations change as packets pass through routers. We build clear mathematical models to describe such phenomenon and derive properties from the models that will be validated through experiments with routers. Besides the applications mentioned earlier, our investigation can also have potential impact on work related to router modeling, packet clustering, and interpacket delay. Our work allows improvements of router models [8, 13] that treat processing delay as deterministic. Clustering packets affect queuing delay [27], packet loss rate [24], and resource allocation [23]. Interpacket delay, on the other hand, has been used to detect congestion [5], estimate bandwidth with packet train [22], trace encrypted connections [35], and load balance traffic without packet reordering [15]. An accurate understanding of packet clustering introduced by routers can potentially help improve all these technologies.

1.2 Background and Summary

To isolate the effect of processing delay from queuing delay, we consider an experimental setup where we send a stream of packets with fixed packet size and constant interpacket delay through an isolated and idle router with no cross traffic. Such an experiment was performed in Ref. [12], and the variation in processing time was reflected by the observation that the interpacket delay at the router's output exhibited variation on the order of 100ns. More importantly, when the experiment was repeated for smaller interpacket delay constants, the variation was observed to be sufficient to induce packet clustering and even packet loss. Moreover, if the input stream went through multiple routers, the clustering effect was more prominent [12].

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 [4, 6, 14, 20]. On longer timescales, proposed explanations for burstiness are centered on input traffic characteristics such as the distribution of user's idle and active time [36], the distribution of file sizes [9], and Transmission Control Protocol (TCP) congestion control [14, 34]. At the packet level, the clustering could be attributed to contention and scheduling with cross-traffic at the switching fabric. In Ref. [12], though, the experiments are of a finer timescale and all the factors mentioned here are not applicable.

Our main focus in this work is to model and analyze the packet clustering induced by the variation in processing time. We propose a model for the transition time through a router and an accompanying metric to quantify packet clustering. Using the model, we provide analytical explanations for the packet clustering and packet loss that were observed but not analyzed in Ref. [12]. Note that much more detailed delay models for router exist, see, e.g., Refs. [28] and [29]. They, however, tend to be too complex for analysis. We then perform some preliminary investigation on possible causes for variation in processing time. We examine the impact of packet sizes, clock drift, and forwarding table lookup on the variability of processing time. Such a fine-scale investigation was not feasible experimentally prior to Ref. [12], as network measurement devices have significant measurement error; see, e.g., Ref. [24].

The article makes the following contributions:

—We propose a simple device-independent model incorporating variation in processing time. The model admits a pipeline structure. Using the model, we make several predictions on the properties of the interpacket delay at the router's output. We predict that adjacent interpacket delays are negatively correlated (Lemma 2) and the histogram of the interpacket delay is symmetrical in shape (Theorem 1).

Table 1. Interpacket Delay and Interpacket Gap for Various Packet Sizes and Data Rates

Ethernet Frame Size [bytes]	After 64/66b Encoding [bits]	Nominal Data Rate [Gbps]	IPD [ns]	IPG [ns]
1,526	12,588	1	12,211	10,991
520	4,288	6	698	282
72	594	3	192	134
72	594	6	96	38.4
72	594	9	64	6.4

All values except for Ethernet frame size are measured from the physical layer of 10 Gigabit Ethernet.

—A metric is proposed to quantify packet clustering. We show how packet clustering in terms of this metric changes as it passes through one or multiple routers (Theorem 3 and Section 5.2). We also develop conditions on when the output is less clustered than the input (Corollary 1).
—We generalize our results by incorporating cross traffic (Lemma 3 and Theorem 6).
—We are able to provide qualitative and quantitative explanations on the observed phenomenon of Ref. [12]. We also verify all the model results with repeatable experiments using Software-defined Network Interface Card (SoNIC) [18], which is a software-defined network interface card that achieves the same level of precision as BiFocals.

2 OBSERVED PHENOMENON: AN EXAMPLE

In this section, we motivate and illustrate our study by presenting the key phenomenon observed in Figure 6 of Ref. [12]. The data presented here are obtained by replicating the experiment using SoNIC [18]. SoNIC is capable of generating and measuring interpacket delays for a packet stream with negligible error. Hence, any changes to the interpacket delay, even at the level of time to transmit a single bit, can be accurately detected. Readers interested in further details should refer to either paper for more information. We first establish some terminologies and conventions before moving on to describe the various experiment setups and their associated observations.

We start with the basic measurement. Same as Ref. [18], we define *interpacket delay* (IPD) as the time difference between the first bit of successive packets, and *interpacket gap* (IPG) as the time difference between the last bit of the first packet and the first bit of the next packet. Both IPD and IPG are measured on the physical layer of the 10 Gigabit Ethernet (10GbE). Preamble bits and control characters are inserted by 64/66b encoding (see Clause 49 of IEEE802.3 [1]) on each Ethernet frame. So while Ethernet frame sizes range from 72 to 1526 bytes, the actual packet size on the physical layer is longer after 64/66b encoding. Table 1 shows various Ethernet frame sizes and their corresponding packet sizes. Due to 64/66b encoding, 1 bit takes $b = 97$ ps ($64/66 \times 1/10$ Gbps) to transmit. We say that a stream of packets is *homogeneous* if all the packets have the same size, the same interpacket delay, the same payload, and are heading for the same destination. We specify a homogeneous packet stream via two parameters: packet size, l and data rate, r. The interpacket delay is given by the relation

$$IPD = \frac{z}{r} \times l \times b, \tag{1}$$

where z is the capacity rate. We use shorthand such as 1526B 3G to refer to a homogenous packet stream with 1526-byte Ethernet frames and 3Gbps data rate. As variation in processing time is inherent to a router, we refer to variation in processing time interchangeably with *inherent variation* of a router.

The first experiment of Ref. [12] sends a homogeneous packet stream through an isolated Cisco 6500 and the resulting IPD of the output stream is measured. We repeat the experiment and plot the IPD histogram in Figure 1(a). The x-axis is the IPD while the y-axis is the count of each IPD on a log-scale. The router's inherent variation causes the actual histogram to be spread out. To verify, the inherent variation is not isolated to just the Cisco 6500 routers; we repeat the experiments for a 1526B 1G input for Cisco 4948 (Figure 1(d)), HP Procurve 2900 (Figure 1(e)), and IBM G8264 (Figure 1(f)). While the histograms have different shapes, the IPD values are all spread out somewhat symmetrically around the input IPD.

Further complications arise when the experiment is repeated with different data rates, packet size, and number of routers. While the variation may seem small when the IPD is large, when the setup is varied by having a higher data rate and smaller packet size, the variation coupled with the small IPD is sufficient to induce packet clusters (see Figure 1(b); note that the highest count is at the leftmost value where the IPG is minimal). The probability of observing such packet clusters further increases if the packets have to pass through multiple routers. In short, inherent variation is observed to induce packet clusters. However, when the input data rate is close to capacity, the effect of inherent variation seems to disappear and packet loss starts to occur (Figure 1(c)).

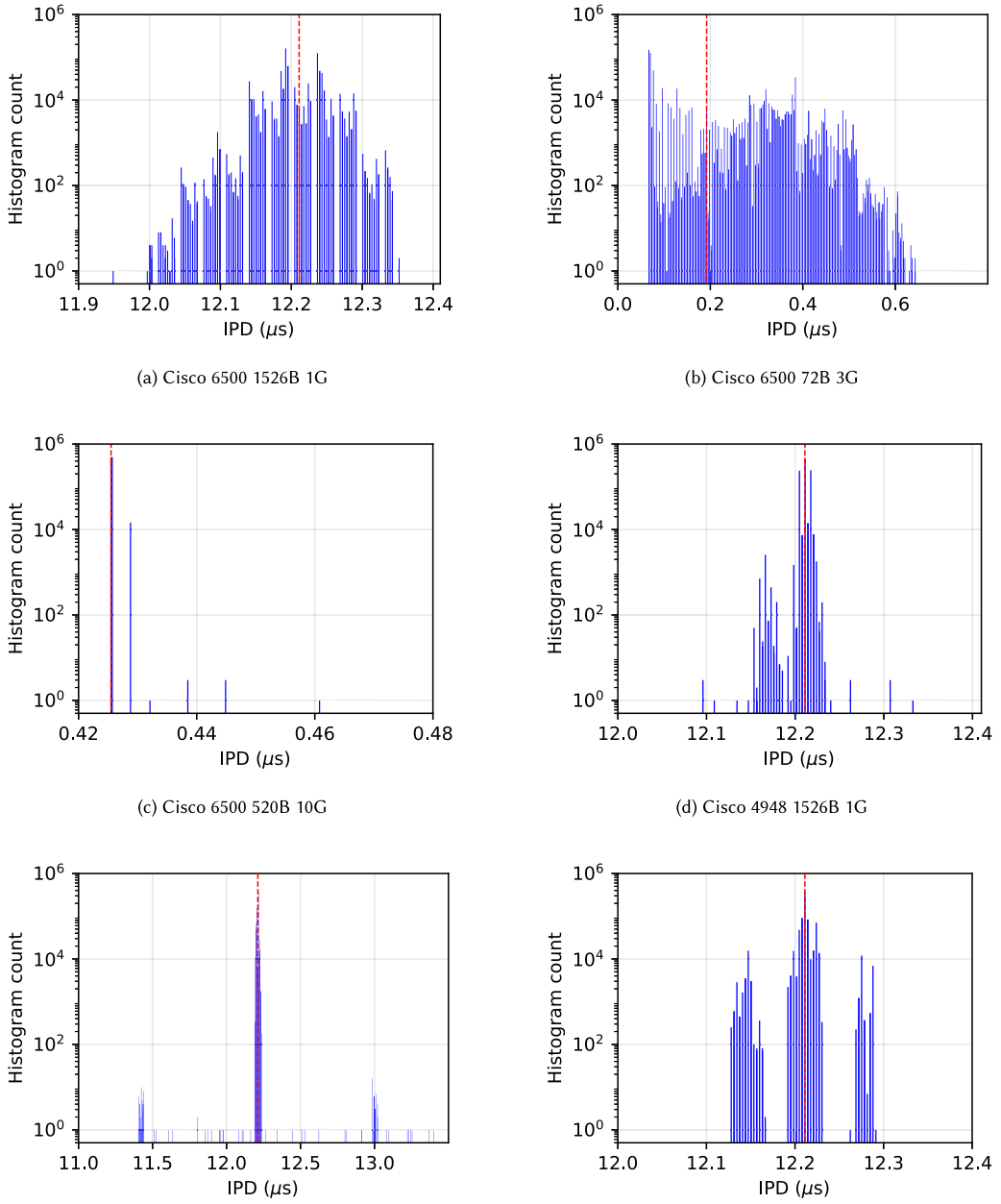
Our experiments and analysis in the subsequent sections are motivated by Figure 1. We first establish the router model with inherent variation (Section 3.1) and quantify packet clustering (Section 3.2). Using the model, we show how IPD and degree of clustering changes as a packet stream passes through one router (Section 4), multiple routers (Section 5), and with cross traffic (Section 6). Finally, we experimentally validate the model and analysis with various routers (Section 7) and explore factors that could affect inherent variation (Section 8).

3 MODELING

Before introducing the router model, we first establish some notational convention. We use an uppercase letter for random variables and the corresponding lowercase letter when the random variable takes on a deterministic value, e.g., $\Pr(D = d)$ is the probability of the random variable D taking on a value of d. We denote $E[\cdot]$ as the expectation function and any random variable with a tilde accent represents its zero-mean equivalent, e.g., $\tilde{X} = X - E[X]$. We use superscript on the variables to denote the router number and subscript to denote the packet number. The superscript is sometimes omitted for the single-hop case while the subscript is sometimes omitted when the statement is applicable to all packets. Table 2 is a notation table explaining the meaning of variables encountered throughout the article.

3.1 Router Model

Though the experiments are carried out using input packet streams that are homogeneous, we generalize and instead assume that the input packet stream has a fixed packet size and an IPD that is independent and identically distributed (i.i.d.). The homogeneous packet stream is then a special case of the model where the IPD is a constant value with probability 1. Each packet of the input stream is indexed with i and transitions through a router over the same input and output port. Each packet experiences processing delay followed by transmission delay. Each delay is modeled as a single server and, thus, the router is modeled as two servers in series (see Figure 2). In general, the processing time may be correlated over time depending on the inner workings of a router. As we mentioned before, the actual workings of how a packet is processed is proprietary. As such, for tractability, we assume the processing time of each packet X_i is i.i.d. As for the transmission time, it is simply the packet size multiplied by the time to transmit one bit, e.g., $u \triangleq l \times b$. This



(a) Cisco 6500 1526B 1G

(b) Cisco 6500 72B 3G

(c) Cisco 6500 520B 10G

(d) Cisco 4948 1526B 1G

(e) HP Procurve 2900 1526B 1G

(f) IBM G8264 1526B 1G

Fig. 1. Histogram of 1 million observed interpacket delay at the output of various routers for different combinations of data rates and packet size. The dotted red line marks the interpacket delay of the homogeneous packet stream.

Table 2. Notation Table

Notation	Meaning
z	Capacity rate
r	Data rate of a packet stream
l	Packet size
b	Time to transmit 1 bit
X_i^j	Processing time of packet i for router j
x_{max}	Maximum packet processing time
u	Packet transmission time
D_i^j	Interpacket delay between packet i and $i+1$ for router j
G_i^j	Interpacket gap between packet i and $i+1$ for router j
C	Expected packet clustering
c	Sample variance of interpacket delaIPD
I_i^j	Processing server idle time in between packet i and $i+1$ for router j
W_i^j	Waiting time of packet i at the processing server of router j

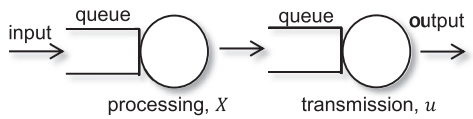


Fig. 2. A two-server model of a router with a random processing time, X and a constant transmission time, u.

two-server model means that it is possible for parallel processing and transmission of different packets, i.e., pipelining, to occur.

We denote the interpacket delay between packet i and $i+1$ with the random variable D_i. The interpacket gap G_i is defined similarly. Since we are modeling actual packets going through the same input port, the interpacket delay in between packets must be larger than the time it takes a router to receive a packet, i.e., $D_i > u$. The router is capable of operating at or near capacity, and, as such, the router should take a shorter amount of time to process than to transmit, i.e., $E[X] < u$. 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 i.i.d. arrivals, independent service time, and a first-in, first-out queue discipline. In Kendall's notation for queues [17], the processing server is a $GI/G/1$ queue, where GI stands for general distribution with independent random values while G stands for general distribution. The transmission server is a $G/D/1$ queue, where D here stands for a deterministic distribution.

The router model is intentionally kept simple. It is meant to explain the observations in Section 2 and is thus mostly limited to scenarios where we have a single input stream going through the same input and output port of a router. When there is cross traffic coming in from other input ports, then the model is applicable only when the cross traffic does not affect processing time, introduce queueing delay, or cause backbone contention. At the same time, due to its simplicity, as long as the limitation holds, the model is applicable for different router architectures, be it cut-through or store-and-forward routers, or routers with one processor per input port or one processor to multiple input ports like in a linecard. We remove some of the limitations by including cross traffic in Section 6.

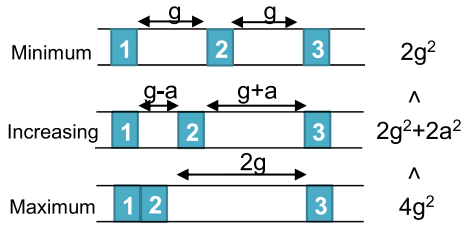


Fig. 3. An intuitive figure depicting how packet clustering evolves as interpacket delay changes. The square of the interpacket gap fits the intuitive notion.

3.2 Packet Clustering Metric

We propose using the packet-averaged interpacket delay to represent the degree of packet clustering, i.e.,

$$C = \frac{1}{n}\sum_{i=1}^{n} \text{var}\,(D_i) \tag{2}$$

When D_i is i.i.d., the expression simplifies to the variance of the interpacket delay. The metric basically states that a packet stream is more clustered if its IPD is more variable.

We choose this metric as it reflects our intuitive notion of packet clustering and is amenable to analysis. What is packet clustering? Let's build some intuition using a simple example. To start, given that the packet arrival rate is smaller than the 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 three packets as shown in Figure 3. Since the average IPD is the same, the only variable here is the relative position of the second packet. Intuitively, the setup with the least clustering 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. As shown in Figure 3, a metric that fits the intuitive ordering is the sum of the squares of the IPG. Similarly, the sample variance of the IPD would fit the intuitive ordering as it is the difference between the average sum of squares and the average squared, i.e.,

$$c = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2 \tag{3}$$

This suggests that we could use var (D) as the metric. But as we will see in the next section, the interpacket delay is in general not i.i.d.; hence, we use Equation (2) instead.

Our metric is closely-related to packet burstiness, which is intuitively understood to be the statistical variability of packet stream. Different papers define what variability is differently. For instance, Ref. [14] uses variance of the wavelet coefficients while Ref. [4] uses correlated loss. The metric used reflects the objective of the paper—Ref. [14] is studying burstiness over different timescales while Ref. [4] is studying packet loss. We are studying the phenomena of packets clustering together and, hence, the choice of variability of interpacket delay.

4 ANALYSIS: 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 possibly do (*small interpacket* gap). Packets have no waiting time if they always arrive after the prior packets have

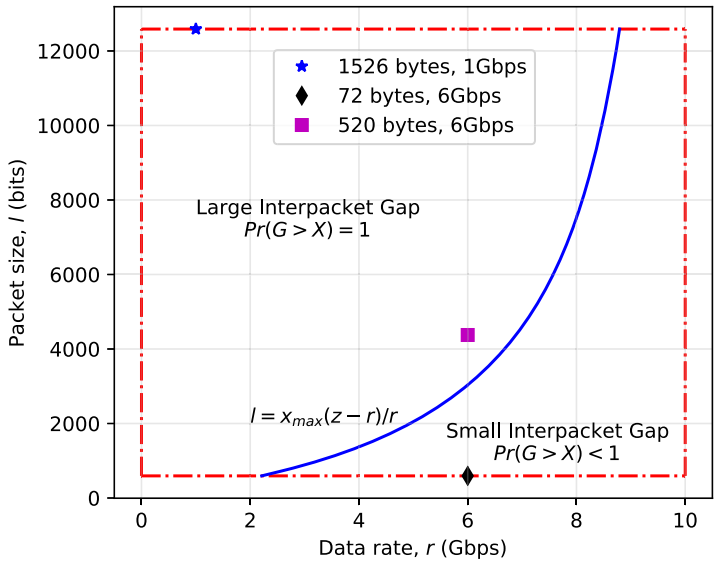


Fig. 4. Large and small interpacket gap region for a router with $x_{max} = 2200b$ for all packet sizes.

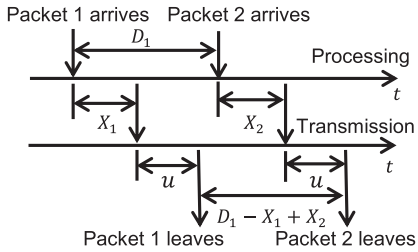


Fig. 5. Large interpacket gap.

been transmitted from the router. A sufficient condition for this to occur is the interpacket delay being greater than the sum of processing and transmission time, i.e.,

$$D > X + u \iff G > X, \quad \text{holds with probability 1,} \tag{4}$$

where D and G are, respectively, the IPD and the IPG of the input packet stream.

To illustrate, consider the parameter space (l, r) for a homogeneous packet stream. Denote g as the interpacket gap constant and x_{max} as the maximum processing time. The condition is then $g > x_{max}$; from Equation (1), we find

$$g + (l + b) = \frac{r}{z} \times l \times b \Longrightarrow g = \frac{(z - r)}{r} \times l \times b \Longrightarrow l > \left(\frac{r}{(z - r) \times b}\right) x_{max} \tag{5}$$

The region surrounded by borders in Figure 4 represents all the feasible parameter combinations for a homogeneous packet stream. We divide it into the two regions according to Equation (5) using $x_{max} = 2200b$ and assuming x_{max} is the same for all packet sizes.

4.1 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 packets 1 and 2 as shown in Figure 5.

Packet 1 arrives at the router and takes a time period $X_1 + u$ to be processed and transmitted. The second packet then arrives and takes $X_2 + u$ to be processed and transmitted. The interpacket delay after 1 router for packets 1 and 2 is thus $D_1^1 = D_1 - X_1 + X_2$. One could continue by considering packets 3, 4, 5, and so on to find that the same results hold:

$$D_i^1 = D_i - X_i + X_{i+1},\ i = 1, 2, \ldots \tag{6}$$

While Equation (6) looks simple, we can extract plenty of information: symmetry of interpacket histogram (Lemma 1 and Theorem 1), negative correlation of adjacent interpacket delay (Theorem 2), and an accurate unbiased estimator to represent a router's inherent variation (Lemma 2).

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

$$Y - y \sim -(Y - y)$$

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

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

Proof. D is symmetric about $E[D]$ implies that

$$D - E[D] \sim -(D - E[D]) \tag{7}$$

Since the sequence $X_1, X_2, \ldots$ is i.i.d.,

$$-X_i + X_{i+1} \sim X_i - X_{i+1} \tag{8}$$

In general, if random variables K and L are independent, and, similarly, M and N are independent, and $K \sim M$ while $L \sim N$, then $K + L \sim M + N$. Putting Equations (7) and (8) together gives

$$D_i - E[D] - X_i + X_{i+1} \sim -(D_i - E[D] - X_i + X_{i+1}),$$

which proves that the distribution of $D_i - X_i + X_{i+1}$ is symmetric about $E[D]$. □

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: $D_1 - X_1 + X_2$ is not independent of $D_2 - X_2 + X_3$ due to the X_2 term. Since the interpacket delays are correlated via the X_2 term, which has an opposite sign in each of the interpacket delay, we expect the correlation to be negative. In simple words, if the processing time of packet 2, X_2, is large, D_1^1 would be large while packet 3 will catch up to packet 2 making D_2^1 small, and vice versa. We state the negative correlation formally.

Lemma 2. *After one hop, the correlation coefficient between two neighboring interpacket delays is −1/2, that is*

$$\rho_{i,j} \triangleq \frac{cov\left(D_i^1, D_j^1\right)}{\sqrt{var\left(D_i^1\right) var\left(D_j^1\right)}} = -\frac{1}{2},\ for\ |i - j| = 1 \tag{9}$$

Proof. The covariance of the interpacket delay is, by definition,

$$\begin{aligned} \text{cov}\left(D_i^1, D_j^1\right) &\triangleq E\left[\tilde{D}_i^1 \tilde{D}_j^1\right] \\ &= E\left[\left(\tilde{D}_i - \tilde{X}_i + \tilde{X}_{i+1}\right)\left(\tilde{D}_j - \tilde{X}_j + \tilde{X}_{j+1}\right)\right] \end{aligned}$$

Expanding the equation gives us nine terms. However, five terms involving either $\tilde{D}_i$ or $\tilde{D}_j$ are equal to zero since they are independent of all other terms and thus imply, for instance, $E[\tilde{D}_i\tilde{X}_{j+1}] = E[\tilde{D}_i]E[\tilde{X}_{j+1}] = 0$. For the other four terms, since the sequence $X_1, X_2, \ldots$ is i.i.d.,

$$E[\tilde{X}_i\tilde{X}_j] = \begin{cases} E[\tilde{X}_i]E[\tilde{X}_j] = 0, & \text{if } i \neq j \\ E[(\tilde{X}_i)^2] = \text{var}(X), & \text{if } i = j \end{cases}$$

Taking the sum of all four terms, give

$$\text{cov}\left(D_i^1, D_j^1\right) = \begin{cases} 2\text{var}(X), & \text{if } i = j \\ -\text{var}(X), & \text{if } |i-j| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Applying Equation (10) on the definition in Equation (9), and using the fact that $\text{var}(D_i^1) = \text{cov}(D_i^1, D_i^1)$ give the desired answer. □

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 1. *The interpacket delay histogram of a homogeneous packet stream at a router's output is symmetrical about $E[D]$.*

Proof. Consider the first and third IPD: $D_1^1 = D_1 - X_1 + X_2$ and $D_3 = D_3 - X_3 + X_4$. Since $X_1, X_2, \ldots$ and $D_1, D_2, \ldots$ are both i.i.d., D_1^1 and D_3^1 are i.i.d. of each other. We can easily generalize this and find that the odd-indexed sequence $D_1^1, D_3^1, \ldots$ is i.i.d., and so is the even-indexed sequence $D_2^1, D_4^1, \ldots$. From Lemma 1, we know that the IPD histogram of either the odd or the even-indexed sequence is symmetrical about $E[D]$. The IPD histogram of the whole sequence is just the superposition of the two, and is, hence, symmetrical about $E[D]$. □

Theorem 1 is significant as it validates the simple pipeline model by matching and providing an explanation for the observed output (Figure 1(a)).

We now switch attention to packet clustering. Packet clustering could be determined using Equation (6). Since D_i, X_i, and X_{i+1} are all independent of each other

$$C(D^1) = \text{var}(D) + 2\text{var}(X) \tag{11}$$

For a homogeneous input packet stream, D is a constant; thus, $C(D^1) = 2\text{var}(X)$. The variance of the processing time $\text{var}(X)$ captures the magnitude of the inherent variation and is an important expression that continues to reappear for the rest of the article. We thus denote $\text{var}(X)$ as the metric characterizing the inherent variation of the router. For practical purposes, it is important to be able to estimate it from available data as accurately as possible. The analysis in this section suggests a simple estimation method: Send a homogeneous packet stream with a sufficiently large IPD through the router and then find the sample variance of the IPD at the output. However, due to the negative correlation, we need a correction factor to make it an unbiased estimation.

Theorem 2. *Given $D_1^1, D_2^1, \ldots, D_n^1$, the sequence of interpacket delays from the output of a router fed with a homogeneous packet stream with large interpacket gap, the estimator*

$$\beta = \frac{n}{(n+1)(n-1)} \sum_{i=1}^{n} \left(D_i^1 - \bar{D}\right)^2,$$

where $\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D_i^1$ is the sample mean, as well as an unbiased estimator of $2var(X)$, i.e., $E[\beta] = 2var(X)$.

Proof.

$$
\begin{aligned}
& n\,(n+1)\,(n-1)\,E\,[\beta] \\
&= E\left[\sum_{i=1}^{n}\left(n\tilde{D}_i^1 - \sum_{j=1}^{n}\tilde{D}_j^1\right)^2\right] \\
&= \mathrm{var}\,(X)\cdot\sum_{i=1}^{n}\left[2n^2 - 4n + 2n\sum_{|i-j|=1} 1 + 2n - 2\,(n-1)\right] \\
&= 2n\mathrm{var}\,(X)\,[n^2 - 2n + 2\,(n-1) + n - (n-1)] \\
&= 2n\,(n+1)\,(n-1)\,\mathrm{var}(X),
\end{aligned}
$$

where the first equality follows from applying the definition of β, the second from applying correlation values from Lemma 2, and the third from noting that if we expand out $(\sum_{j=1}^{n}\tilde{D}_j^1)(\sum_{k=1}^{n}\tilde{D}_k^1)$, there are $2\,(n-1)$ terms where $|j-k|=1$. □

4.2 Small Interpacket Gap

The interpacket gap in this region is small enough that the service time is sometimes long enough to induce a waiting time in the next packet. The IPD histogram is no longer symmetrical, as its left portion is distorted by the physical requirement of a non-negative interpacket gap (Figure 1(b)). To get a clearer picture and to figure out how packet clustering happens, we follow the development of the previous section by considering the sequence of discrete events that occur on packet i and $i+1$. It turns out that we need to consider three general cases: positive idle time (Figure 6(a)), zero idle time (Figure 6(b)), and positive waiting time at the transmission server (Figure 6(c)). 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.

Before diving into the analysis, observe that while we have assumed $E\,[X] < u$, occasionally it is possible for the processing time to exceed the packet transmission time as depicted in Figure 6(b). This implies that if the input packet stream is sent at the maximum data rate, a queue is going to build up whenever the processing time exceeds the transmission time. Eventually, the buffer overflows and packet loss occurs as shown in Figure 1(c).

From Figure 6(a) and (c), we see that the idle time is given by

$$I_i = \max\,(0, D_i - W_i - X_i) = (D_i - W_i - X_i)^+, \tag{12}$$

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

$$W_{i+1} = \max\,(0, W_i + X_i - D_i) = (D_i - W_i - X_i)^-\,, \tag{13}$$

where $a^- = \max\,(0, -a)$. In the queuing theory literature, Equation (13) is also known as the Lindley equation [21]. Combining all cases, the IPD after one-hop is given by

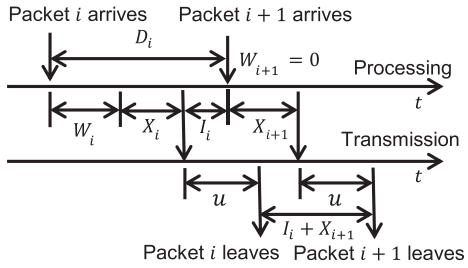
$$D_i^1 = \max\,\{u, I_i + X_{i+1}\} \tag{14}$$

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

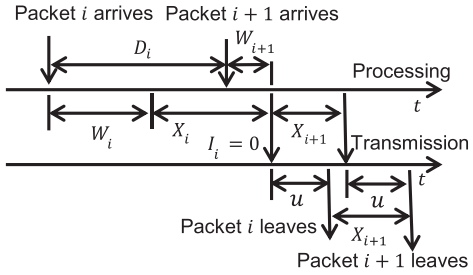
$$D_i - W_i - X_i = I_i - W_{i+1} \tag{15}$$

Combining Equations (14) and (15) gives

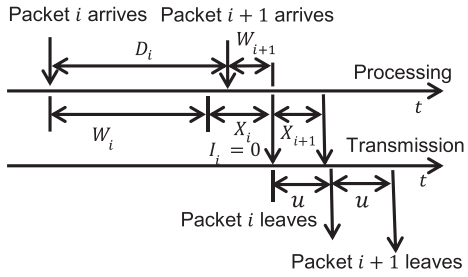
$$D_i^1 \geq D_i - (W_i + X_i) + (W_{i+1} + X_{i+1}) \tag{16}$$



(a) IPD when there is positive idle time



(b) IPD when there is zero idle time



(c) IPD when there is positive waiting time at the transmission server

Fig. 6. Three possible sequence of events in between the arrivals and departures of packets i and $i+1$.

Recall that for the large interpacket gap region, Equation (11) tells us that packet clustering increases by 2var (X). For the small interpacket gap region, some packets are prevented from getting closer together due to the physical requirement of a fixed transmission time. This implies that, intuitively, packet clustering after one router should increase to a value that is less than 2var (X).

Theorem 3. *For a packet stream with i.i.d. interpacket delay and $n \to \infty$ number of packets, after one hop, $C_v(D^1)$ is given by*

$$var(D^1) \leq var(D) + 2\left(var(X) - E[W]\,E[I]\right) \tag{17}$$

Proof. We start with Equation (14) and, noting that forcing a minimum value reduces variance

$$\text{var}(D_i) \leq \text{var}(I_i + X_{i+1}) = \text{var}(I_i) + \text{var}(X_{i+1}), \tag{18}$$

where the second equality follows from independence of I_i^1 and X_{i+1}^1. We apply the identity: for any random variable $Y = Y^+ - Y^-$, $\text{var}(Y) = \text{var}(Y^+) + \text{var}(Y^-) + 2E[Y^+]\,E[Y^-]$ on $Y = D_i^0 - W_i^1 - X_i^1$ to find an expression for var (I_i)

$$\begin{aligned}&\text{var}\left(D_i^0 - W_i^1 - X_i^1\right)\\&\quad= \text{var}\left(I_i^1\right) + \text{var}\left(W_{i+1}^1\right) + 2E\left[I_i^1\right]E\left[W_{i+1}^1\right]\end{aligned} \tag{19}$$

Since X_i is independent of D_i and W_i, and D_i is independent of W_i, left-hand side (LHS) of Equation (19) is equal to var (D_i) + var (W_i) + var (X_i). Rearranging the terms, we obtain an expression for var (I_i) and substitute it into Equation (18) to find

$$\begin{aligned}\text{var}\left(D_i^1\right) \leq\ &\text{var}\,(D_i) + \text{var}\,(W_i) + 2\text{var}\,(X)\\&- \text{var}\,(W_{i+1}) - 2E\,[I_i]\,E\,[W_{i+1}]\end{aligned} \tag{20}$$

As $n \to \infty$, each term converges to equilibrium values, e.g., $D_i^1 \to D^1$, $I_i \to I$ and $W_{i+1} \to W$. We obtain

$$\text{var}(D^1) \leq \text{var}\,(D) + 2\,(\text{var}\,(X) - E\,[W]\,E\,[I]) \tag{21}$$

as desired. □

Since $E\,[W]\,E\,[I] > 0$, Theorem 3 confirms our intuition that packet clustering increases by a value less than 2var (X). 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 $E\,[W]\,E\,[I] >$ var (X). However, we may not be able to check for it in practice, since we may not have sufficient knowledge or access to determine $E\,[W]$ and $E\,[I]$. 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* $2var(X) \leq E[\{(D - X)^-\}^2]$.

Proof. We apply a lower bound by Kingman [16]

$$2E\,[I]\,E\,[W] \geq E[\{(D - X)^-\}^2]$$

If $2\text{var}(X) \leq E[\{(D - X)^-\}^2]$, then the lower bound and Equation (17) gives

$$\begin{aligned}\text{var}(D^1) &\leq \text{var}\,(D) + 2\text{var}\,(X) - E[\{(D - X)^-\}^2]\\&\leq \text{var}(D),\end{aligned}$$

i.e., packet clustering has decreased after passing through a router. □

Note that $E[\{(D - X)^-\}^2]$ is large if D 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 $\text{var}(D) + 2\text{var}(X) - E[\{(D - X)^-\}^2]$ is an upper bound for $\text{var}(D^1)$, the upper bound is getting smaller as either scenario occurs. As such, it is reasonable to expect *the change in packet clustering,* $var(D^1) - var(D)$ *to decrease as the input packet stream is getting more clustered or as the input data rate increases* (see Figure 12). 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 10(a))

5 ANALYSIS: THE MULTI-HOP CASE

In this section, we extend our results to the multiple-router scenario. Do note that the 0 superscript refers to the input packet stream and brackets are used whenever exponents are involved, e.g., X^2 is the processing time at the second router while $(X)^2$ is the processing time squared. We denote m as the number of routers and the routers are not assumed to be identical unless otherwise stated. There are now three regions to consider. Similar to before, the regions depend on the interpacket

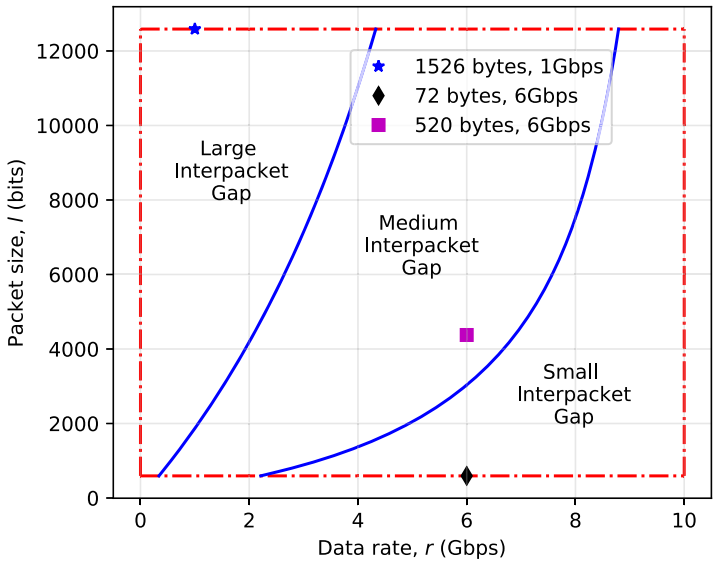


Fig. 7. Large, medium, and small interpacket gap region for $m = 8$ identical routers and $x_{max} = 2{,}200$ bits.

gap and maximum processing time. If all packets have zero waiting time while passing through all routers, i.e., $G^0 > \sum_{k=1}^{m} x_{max}^k$, then we are in the large interpacket gap region. If all packets have zero waiting time for the first j routers and some packets have positive waiting time for the remaining routers, i.e., there exists $j < m$ with $G^0 > \sum_{k=1}^{j} x_{max}^k$ but $G^0 < \sum_{k=1}^{j+1} x_{max}^k$ with positive probability, then we are in the medium interpacket gap region. In the small interpacket gap region, some packets have positive waiting time while passing through each router, i.e., $G^0 < \sum_{k=1}^{l} x_{max}^k$ with positive probability for $l = 1, \ldots, m$.

For $m = 8$ identical routers, the large, medium, and small interpacket gap region are shown in Figure 7. With reference to the single-hop Figure 4, the small interpacket gap region remains the same while the large interpacket gap region is now divided into two. As the number of routers grows, the medium interpacket gap region grows while the large interpacket gap region shrinks.

5.1 Large Interpacket Gap

The analysis is this section is mostly a straightforward generalization of the results of the single-hop case. Note that for this section only, the integer exponent refers to the router number. First, consider $m = 2$. Since we are in the large interpacket gap region, Equation (6) holds and we have

$$D_i^2 = D_i^1 - X_i^2 + X_{i+1}^2$$

Applying Equation (6) again to D_i^1 gives

$$D_i^2 = D_i^0 - \left(X_i^1 + X_i^2\right) + \left(X_{i+1}^1 + X_{i+1}^2\right)$$

Generalizing, we obtain for all $l \leq m$

$$D_i^l = D_i^0 - \sum_{k=1}^{l} X_i^k + \sum_{k=1}^{l} X_{i+1}^k \tag{22}$$

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* $1 \leq l \leq m$*, the interpacket delay sequence* $D_1^l, D_2^l, \ldots$ *has the following three 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.*

Proof. To prove the first two properties, denote $Y_i = \sum_{k=1}^{l} X_i^k$ to get

$$D_i^l = D_i^0 - Y_i + Y_{i+1}$$

This is exactly Equation (6), and since the sequence $Y_1, Y_2, \ldots$ is i.i.d., negative correlation and symmetry follow from Theorems 2 and 1. Finally,

$$\text{var}\left(D_i^l\right) = \text{var}(D^0) + 2\sum_{k=1}^{l} \text{var}(X^k)$$

For a homogeneous packet stream and identical routers, $\text{var}(D_i^l) = 2l\text{var}(X)$, and, thus, $C_v(D^l) = 2l\text{var}(X)$. □

5.2 Medium and Small Interpacket Gap

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

For small interpacket gap, the results are not as easy to generalize from the single hop case as the 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. Additionally, even if the setup does converge, we could go through the same steps as the proof of Theorem 3 to find that, due to correlation, we now have an additional covariance term, $\text{cov}(D^0, W)$, which does not have a clear interpretation. As such, we approximate by ignoring the correlation and assuming that the input to all the routers have i.i.d. interpacket delay. Our aim in this section is not to derive analytical expressions but to use the results from Section 4.2 to argue qualitatively how packet clustering would evolve with an increasing number of identical hops.

Recall that after Corollary 1, we argue that the change in packet clustering, $\text{var}(D^1) - \text{var}(D^0)$, 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. Since $\text{var}(D^1) - \text{var}(D^0)$ could be negative when packet clustering exceeds the threshold specified in Corollary 1, we expect packet clustering to converge to a value for a sufficiently large 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 and would also converge to a fixed value for a sufficiently large number of identical hops (see Figure 10(a)).

The second implication is that if we have two input packet streams with equal packet clustering but a 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 streams with a 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 function for 4G would be strictly above that of 6G's (see Figure 12).

Table 3. Notation Table for Cross-traffic Analysis

Notation	Meaning
Δ	Interpacket delay after passing through the processing server
W_i^p, W_i^r	Total waiting time of packet i at the processing and transmission server, respectively
T_i^p, T_i^r	Waiting time of packet i due to target traffic at the processing and transmission server, respectively
V_i^p, V_i^r	Waiting time of packet i due to cross traffic at the processing and transmission server, respectively
I_i^p, I_i^r	Time not spent processing target traffic in between packet i and $i+1$ at the processing and transmission server, respectively

6 ANALYSIS: ADDING CROSS TRAFFIC

In this section, we add in cross traffic. 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 shares the same processing server, the *cross traffic*. The cross traffic may or may not be destined for the same output port as the target traffic. It is well known that Internet traffic is usually not Poisson in nature and is positively correlated over time [20]. 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.

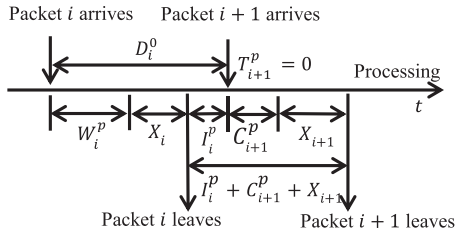
The notational convention here differs from the previous section. Our analysis here focuses on the single-hop case; thus, we drop the superscript for denoting router number. Instead, the superscript is used for denoting whether the variable belongs to the processing server (p) or the transmission server (r). We include a notation table (Table 3) specific to just this section.

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, 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

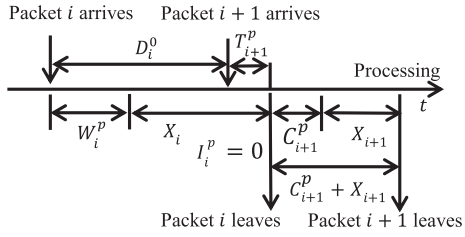
$$W_i^p = T_i^p + V_i^p, \tag{23}$$

where T_i^p is the waiting time incurred on the ith target packet till target packet $i-1$ is processed and V_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, V_i^r, and I_i^r are defined analogously for the transmission server.

The analysis to derive the interpacket delay and packet clustering follows the same path as the model with no cross traffic. There are two general cases as shown in Figure 8. Going through the



(a) Positive idle time $I_i^p > 0$



(b) Zero idle time $I_i^p = 0$

Fig. 8. Two possible sequence of events at the processing server in between the arrivals and departures of target packets i and $i+1$.

same steps as before, we arrive at

$$I_i^p = \left(D_i^0 - X_i - W_i^p\right)^+ \tag{24}$$

$$T_{i+1}^p = \left(D_i^0 - X_i - W_i^p\right)^- \tag{25}$$

$$\Delta_i = I_i^p + V_{i+1}^p + X_{i+1} \tag{26}$$

$$D_i^0 - X_i - W_i^p = I_i^p - T_{i+1}^p \tag{27}$$

$$\Delta_i = D_i^0 - W_i^p - X_i + W_{i+1}^p + X_{i+1} \tag{28}$$

We next find out the change in packet clustering.

Lemma 3. *For a target packet stream with i.i.d. interpacket delay and $n \to \infty$ number of packets, if the current setup with cross traffic converges, then*

$$var\,(\Delta) = var(D^0) + 2\left[var\,(X) + cov\,(I^p + C^p, W^p)\right] \tag{29}$$

Proof. The proof is almost identical to the proof for Theorem 3 and thus some details will be skipped. We start with Equation (26)

$$\text{var}\,(\Delta_i) = \text{var}\left(I_i^p\right) + \text{var}\left(V_{i+1}^p\right) + \text{var}\,(X) + 2\text{cov}\left(I_i^p, V_{i+1}^p\right) \tag{30}$$

We will find an expression for $\text{var}(I_i^p)$ using the same identity and with Equations (24) and (25)

$$\begin{aligned}&\text{var}\left(D_i^0\right) + \text{var}\,(X) + \text{var}\left(W_i^p\right)\\ &\quad = \text{var}\left(I_i^p\right) + \text{var}\left(T_{i+1}^p\right) + 2E\left[I_i^p\right]E\left[T_{i+1}^p\right]\end{aligned} \tag{31}$$

Rearranging, substituting the expression for $\text{var}(I_i^p)$ into Equation (30), and taking the limit as $i \to \infty$ where each term converges to equilibrium distribution, we find

$$\begin{aligned}\text{var}(\Delta) = \text{var}(D^0) + 2\text{var}(X) - 2E[I^p]E[T^p] + \text{var}(W^p) \\ - \text{var}(T^p) + \text{var}(C^p) + 2\text{cov}(I^p, C^p)\end{aligned} \tag{32}$$

Since $\text{var}(W^p) - \text{var}(T^p) = \text{var}(C^p) + 2\text{cov}(T^p, C^p)$, $-E[I^p]E[T^p] = \text{cov}(I^p, T^p)$ and $\text{var}(C^p) = \text{cov}(C^p, C^p)$, the last five terms on the right-hand side (RHS) of Equation (32) simplifies to $2\text{cov}(I^p + C^p, W^p)$; thus, we are done. □

We then move on to the transmission server. Now note that the analysis is identical to the two cases shown in Figure 8, with Δ replacing D^0, u replacing X_i and X_{i+1}, and waiting and idle time terms for the transmission server replacing analogous terms for the processing server.

Theorem 5. *Assuming the setup converges to equilibrium distributions, then*

$$\begin{aligned}var(D^1) \leq\ & var(D^0) + 2var(X) + 2cov(I^p + C^p, W^p) \\ & + 2cov(I^r + C^r, W^r)\end{aligned} \tag{33}$$

Proof. All the steps are the same as the proof of Lemma 3, except for Equation (31) where we have an additional covariance term, $2\text{cov}(\Delta_i, W_i^r)$ due to the correlation of Δ_i with adjacent interpacket delay values. Substituting Δ_i using Equation (28) and noting that all the terms are independent of W_i^r except W_i^p, we arrive at

$$2\text{cov}\left(\Delta_i, W_i^r\right) = -2\text{cov}\left(W_i^p, W_i^r\right) < 0$$

As we assume packets are processed at a faster rate than they could be transmitted, a larger waiting time at the processing server implies that we should expect a larger waiting time at the transmission server; thus, $\text{cov}(W_i^p, W_i^r) > 0$. For the proof here, the equality of Equation (31) is thus replaced with $\geq$, which carries through to give

$$\text{var}(D^1) \leq \text{var}(\Delta) + 2\text{cov}(I^r + C^r, W^r) \tag{34}$$

Now apply Equation (3) and we are done. As a simple check, when there is no cross traffic, $C^p = 0, C^r = 0, I^p = I, T^p = W$, $\text{cov}(I, W) = -E[I]E[W]$, $\text{cov}(I^r, T^r) = -E[I^r]E[T^r] < 0$, and the expression degenerates to Equation (17). □

Before showing that packet clustering can decrease, we backtrack to the large interpacket gap scenario. From Equation (28) and its analogous version for the transmission server, we have

$$D^1 = D_i^0 - W_i^p - W_i^r - X_i + W_{i+1}^p + W_{i+1}^r + X_{i+1} \tag{35}$$

Consider what happens when the target traffic is homogeneous with a sufficiently large interpacket gap. The waiting time is mostly due to cross traffic, i.e., $W_i^p \approx V_i^p$ and $W_i^r \approx V_i^r$. Since the cross traffic is stationary, V_i^p is approximately independent of and has similar distribution as V_{i+1}^p and similarly for V_i^r and V_{i+1}^r. In such a scenario, we could apply the same reasoning as Section 4.1 to find that at the output, the target traffic has interpacket delays with adjacent values having a correlation of $-1/2$ and with a histogram that is symmetrical in shape (see Section 7.2).

We now show that even with cross traffic, it is still possible for packet clustering to decrease. Theorem 1 tells us that packet clustering could decrease when the input stream is already clustered. With the addition of cross traffic, we should expect the decrease could only happen when the input stream is more clustered than is required for the no cross traffic scenario.

Theorem 6. *With cross traffic, the change in packet clustering is upper bounded by*

$$\begin{aligned} & var(D^1) - var(D^0) \\ & \leq 2\left[var\left(X\right) + var\left(C^p\right) + cov\left(I^p, C^p\right) + var\left(C^r\right) + cov\left(I^r, C^r\right)\right] \\ & \quad - E[\{(D^0 - X - C^p)^-\}^2] - E[\{(\Delta - u - C^r)^-\}^2] \end{aligned} \tag{36}$$

We need the help of the following lemma before proving.

Lemma 4. *At steady state,*

(i) $E[I^p] = E[D^0 - X - W^p]$
(ii) $E[\{I^p\}^2] \leq E[\{(D^0 - X - C^p)^+\}^2]$
(iii) $\text{var}(I^p)$ is upper bounded by $\text{var}(D^0) + \text{var}(X) + \text{var}(C^p) - E[\{(D^0 - X - C^p)^-\}^2]$
(iv) $\text{var}(I^r)$ is upper bounded by $\text{var}(\Delta) + \text{var}(C^r) - E[\{(\Delta - u - C^r)^-\}^2]$

Proof. The first expression could be obtained by taking the expectation on Equation (26), applying $E[\Delta] = E[D^0]$, and rearranging the terms. As for the second equation, start from the inequality $(D^0 - X - W^p)^+ \leq (D^0 - X - C^p)^+$, square both sides, take expectation, and apply Equation (24). The upper bound for $\text{var}(I^p)$ is derived by applying the two expressions we just derived.

$$\begin{aligned} & \text{var}(I^p) \\ & = E[\{I^p\}^2] - E[I^p]^2 \\ & \leq E[\{(D^0 - X - C^p)^+\}^2] - E[D^0 - X - C^p]^2 \\ & = E[(D^0 - X - C^p)^2 - \{(D^0 - X - C^p)^-\}^2] \\ & \quad - E[D^0 - X - C^p]^2 \\ & = \text{var}(D^0 - X - C^p) - E[\{(D^0 - X - C^p)^-\}^2] \end{aligned}$$

Expand out the first term and we are done. The upper bound for $\text{var}\left(I^r\right)$ is derived in a similar manner. □

We now prove Theorem 6.

Proof. Start from $\Delta = I^p + X + C^p$, take the variance, and apply the upper bound for $\text{var}\left(I^p\right)$ from Lemma 4 to obtain

$$\begin{aligned} & \text{var}(\Delta) - \text{var}(D^0) \\ & \leq 2\left[\text{var}\left(X\right) + \text{var}\left(C^p\right) + \text{cov}\left(I^p, C^p\right)\right] \\ & \quad - E[\{(D^0 - X - C^p)^-\}^2] \end{aligned} \tag{37}$$

Similarly, with $D^1 = I^r + u + C^r$, take the variance, and apply the upper bound for $\text{var}\left(I^r\right)$ from Lemma 4 to obtain

$$\begin{aligned} & \text{var}(D^1) - \text{var}\left(\Delta\right) \\ & \leq 2\left[\text{var}\left(C^r\right) + \text{cov}\left(I^r, C^r\right)\right] - E[\{(\Delta - u - C^r)^-\}^2] \end{aligned} \tag{38}$$

Sum Inequalities (37) and (38), and we are done. □

Similar to the case with no cross traffic, the last two terms on the RHS of Equation (36) is large if D^0 has a large probability of taking small values; as such, the observation from the end of Section 4.2 carries over and it is reasonable to expect that, even with cross traffic, the change in packet clustering $\text{var}(D^1) - \text{var}(D^0)$ will decrease as the input packet stream gets more clustered or as the input data rate increases.

7 EXPERIMENTS

7.1 Setup

To carry out our experiments, we deploy a SoNIC board on a Dell T7500 workstation and use a Cisco Catalyst 6500 router. The workstation is equipped with two 2.93GHz 6-core Xeon 5670 processors, and with 12GB RAM, 6GB connected to each of the processors. Two Myricom SFP+ LR transceivers are plugged into the SoNIC board for wire connections. The Cisco router uses Cisco Supervisor Engine 720 for the control plane, and four-port 10 Gigabit Ethernet Fiber module (WS-X6704-10GE) for the data plane. The router is configured to run in default L2 forwarding mode. We connect two ports of the SoNIC board to two 10GbE ports of the router via LC/SC optical fibers. Then, we use one port of SoNIC to generate packets and the other port of SoNIC to capture packets routed from the router.

We extend the functionality of SoNIC to mimic a multi-hop routing environment. In particular, after capturing and recording the IPD from the output port, we use it to generate a new stream of packets with IPDs identical to the recorded data. This new packet stream is then sent to the input port, and the process can be repeated any number of times. This is possible because SoNIC can capture what is sent including the size of packets along with headers and the number of idle characters and bits between any two packets. Therefore, SoNIC can easily reconstruct an identical packet stream with captured information. We use this functionality to validate our model and scale the experiment up to 20 hops or more.

All the experiments are performed with the target traffic having at least 1 million packets. For the cross traffic, the packet size is assumed to follow a log-normal distribution [11], while its variance could be estimated from Internet packet size study, e.g., Ref. [33], to be in the order of 10^6 bits^2 or higher. The interpacket delay is chosen to follow the exponential distribution, and if the value is smaller than the packet size, then the two packets simply chain together.

For setups with cross-traffic, the cross traffic is sent in through a different input port but shares the same output port as the target traffic. We capture the packet stream at the output port and drop the cross traffic packets from the packet stream before measuring the interpacket delays of the target traffic. We note here that our experiments here with cross-traffic are, similar to our analysis, limited to only the single-hop case.

In all the experiments, packet clustering is calculated using Equation (3), i.e., by calculating the sample variance of the interpacket delays of the target traffic. The graphs we plot are normalized such that 1 unit of packet clustering is equivalent to $b^2 \approx (0.1 \text{ ns})^2$. Thus, 10^6 units of packet clustering correspond to a variation in the order of $0.1\text{ns} \times 10^3 = 100\text{ns}$. For experiment with multiple hops, 20 hops are usually enough for practical purposes, though we sometimes extend to 100 hops when we are concerned with the asymptotic behavior.

7.2 Results

Experiment 1. Negative Correlation of Adjacent Interpacket Delays (Theorem 2). The correlation coefficient is computed using 1 million IPDs obtained from experiment for various setups. The results are summarized in Table 4. We can see that the computed values are close to the value of -0.5. The three setups where the correlation coefficients are far from -0.5 are from the small interpacket delay region. Even then, they confirm our intuition that the correlation coefficient for adjacent IPD is negative.

Experiment 2. Symmetry in the Observed Distribution of Output IPD for Large Interpacket Gap (Theorem 2 and Theorem 1). We already know that this is confirmed by Figure 1(a) for the no cross traffic case. For the cross traffic case, its observation is mentioned in RFC5481 [26]. We show

Table 4. Correlation Coefficient Values under Various Data Rates and Packet Sizes

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

The 3 starred values are setups that belong to the small interpacket delay region (see Figure 4).

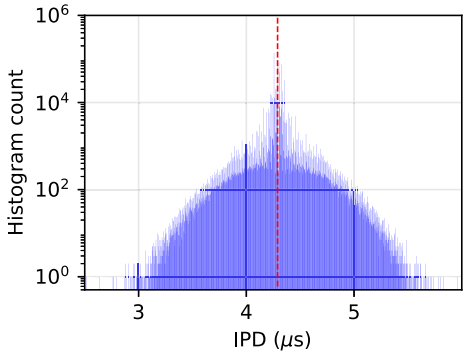


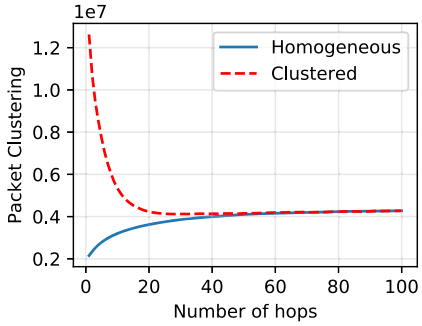
Fig. 9. IPD histogram for a target traffic of 520B 1G and a cross traffic of 1.2G and a packet size following a log-normal distribution with a mean of 520 bytes and variance of 5.12×10^6 bits^2.

here an experiment with cross traffic demonstrating the symmetry in Figure 9. The correlation coefficient of adjacent interpacket delays is calculated to be −0.4944.

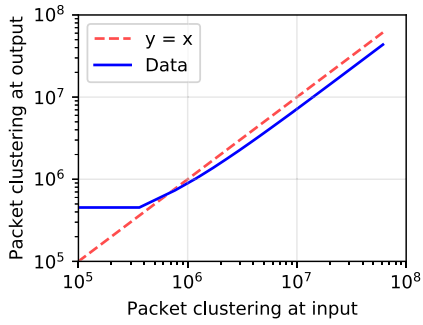
Experiment 3. Packet Clustering Could Decrease after Passing through One Router (Corollary 1 and Theorem 6). We verify that for a very clustered input packet stream, packet clustering decreases after the packet stream passes through the router. Without cross traffic, the result is shown in Figure 10(a). 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 a decreasing and convex way for the clustered packet stream. This agrees with the discussion in Section 5.2.

With cross traffic, the result over a single hop is shown in Figure 10(b). The red dashed line is the $y = x$ line while the blue line represents the collected data on packet clustering at the output. The height difference between the blue line and the red line represents the change in packet clustering. When the blue line is above the red line, it means that change in packet clustering from input to output is positive and vice versa. We see from the figure that as packet clustering at the input increases, the change in packet clustering decreases and eventually goes negative.

Experiment 4. 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 eight identical routers and the IPD is recorded. The IPD histogram is observed to be symmetrical and packet clustering grows linearly (see Figure 11(a) and (b)). In addition, the correlation coefficient between adjacent IPD is calculated as −0.4858, which is close to the theoretical value of −0.5.

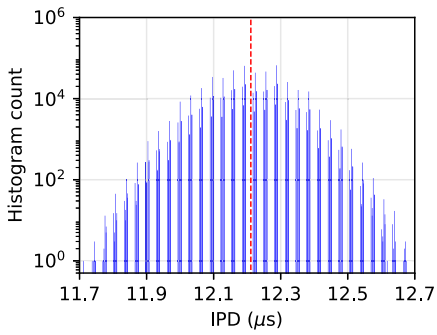


(a) How packet clustering evolves over multiple routers for clustered vs. homogenous packet stream. Both packet streams are 72B 3G.



(b) How packet clustering changes from input to output. The target traffic has a data rate of 3G, packet size of 800B and packet clustering controlled by the parameter of a geometric random variable representing the number of packets clustered together.

Fig. 10. Results for Experiments 3.



(a) Interpacket delay after 8 hops for homogeneous input traffic of 1526-byte packet and 1 Gbps data rate.

(b) Linear growth of packet clustering.

Fig. 11. Results for Experiments 4.

Experiment 5. How Packet Clustering Evolves for Different Data Rates with Increasing Number of Hops (Section 5.2). Figure 12 shows experimental data for fixed packet size but varying data rates. The 4-, 6-, and 7Gbps setup represent medium IPD, and the 8Gbps represents the small IPD. 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 5.2.

Experiment 6. Validation via Simulation. A good way to check the validity of the model is to simulate the model and compare it with experimental data with the same parameters. We code up the model in Matlab and compare it against a Cisco 6500 router. The variance of the processing time is estimated using Lemma 2, and we observe that if we are only concerned about packet clustering, then the underlying distribution does not matter too much.

Figure 13 shows how packet clustering changes as a homogeneous traffic passes through multiple routers for experimental and simulated data. Note that while we have used an exponential

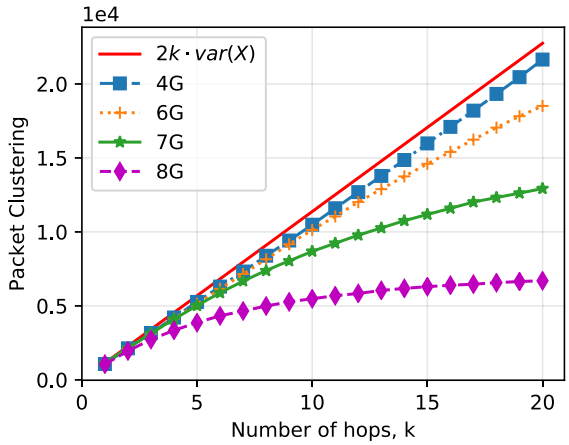


Fig. 12. How packet clustering evolves as a homogenous packet stream of 520-byte packets passes through a different number of routers for various data rates.

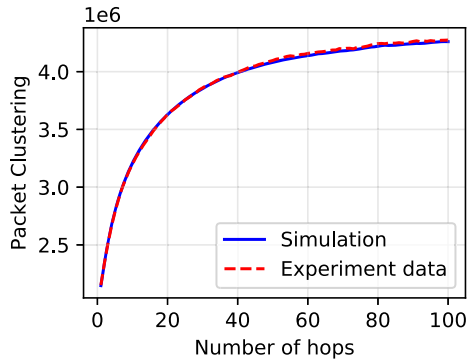


Fig. 13. Simulation vs. experiment. The input homogeneous traffic has 72-byte packet and 3Gbps data rate.

distribution to represent X, the simulated values for the model still agree closely with the experimental data even after 100 hops.

8 DISCUSSION: POSSIBLE FACTORS AFFECTING INHERENT VARIATION

So far, we have provided modeling for packet clustering effects. We then further did analysis based on our model and carried out experiments to verify our model and predictions. These are phenomenological studies in the sense that they try to explain observations quantitatively without connecting to underlying reasons. In this section, we briefly discuss some plausible mechanisms underlying such phenomenon.

In the absence of external factors, inherent variation could only be explained by factors inherent in the router design itself. Router design such as route caching, state transition, quantization of packets into cells [7], clock drift, and the like, could all have an effect on inherent variation. However, while white papers—for instance, Ref. [2]—exist to explain conceptually how a packet transitions through a commercial router, the actual implementation is proprietary. As such, our investigations are limited to packet sizes and destination addresses, which we could experimentally control and validate. These investigations are preliminary in nature. They are meant to shed

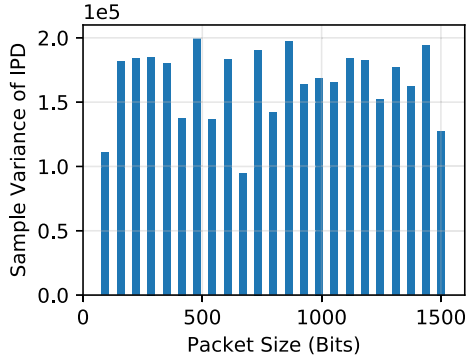


Fig. 14. Sample variance of IPD for various packet sizes. Each bar is spaced apart by 64 bits.

some insight into the behavior of inherent variation and to point at some interesting observations. We leave a comprehensive study of all these factors as possible future work.

8.1 Packet Size

We test the effect of packet size by sending streams of homogeneous traffic with fixed packet sizes of 94, 158, 222, . . . , 1502 bits through a router. We then measure the sample variance of the IPD at the output of the router. To guarantee that this is an accurate description of the inherent variation of the router according to Lemma 2, we need the packet streams to be spaced sufficiently apart and, in this case, we have an interpacket gap of 113,340 bits for all streams. The result is shown in Figure 14. There is no obvious relationship between packet size and inherent variation, and the values vary significantly as the highest value is more than twice the lowest value. The high variation is not an artifact of insufficient sample size, as we repeated the experiments multiple times to get similar values.

8.2 Forwarding Table Lookup

When a router runs as a L2 switch, it records the source MAC address of a packet in its forwarding table upon receiving the packet. After that, any packets destined to the learned MAC address will be forwarded according to the MAC table. For our experiments, we make sure that the MAC address of the receiver is known to the router by generating a few packets from the receiver to the sender. Then, using the idea from the proof of Theorem 1, we generate, using SoNIC, a 1526B 1G stream of 1 million packets that alternate destination MAC addresses between the receiver's MAC address (odd) and random MAC addresses (even). By doing so, we are able to enforce flow table entry to be cached for known MAC address and table lookup to fail for packets with random destination MAC addresses. A failed lookup causes the packet to be broadcasted to all output ports. We record the packet stream at the receiver and split it into two streams of even and odd packets, and measure the sample variance of IPD of each stream.

The calculated sample variance is 1.36×10^5 seconds^2 and 1.53×10^5 seconds^2 for the odd and even packet stream, respectively, which shows that the processing time is different depending on whether the forwarding table lookup is successful or not.

8.3 Clock Drift

While analyzing our interpacket delay data, we come across an interesting observation when we plot the packet timing difference as shown in Figure 15. For an homogenous input packet stream, the second packet is one IPD away from the first packet. At the output, it is d_1. The timing difference

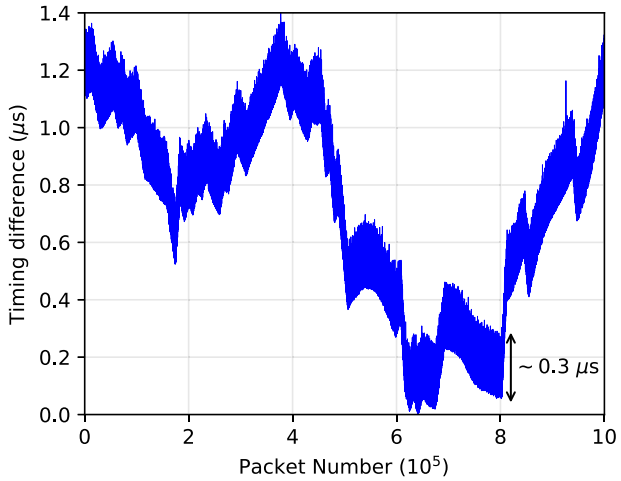


Fig. 15. This figure shows the difference between packet timing at Cisco 6500 output and input for homogeneous traffic with 1500-byte packets and a 1Gbps data rate. All the values have been shifted by a constant to illustrate the magnitude of the changing latency difference. The band height of approximately $0.3\mu s$ is in agreement with the spread of values in Figure 1(a).

of the second packet is defined to be IPD $- d_1$. The timing difference computes the deviation of the ith output packet from its temporal location in a homogeneous packet stream. Generalizing, timing difference of the ith packet is $(i-1) \times \text{IPD} - \sum_{j=1}^{i-1} d_j$. The expectation is that the timing difference figure would show fluctuations around a horizontal line. Instead, other than the fluctuations, there is also a drift that averages out to zero when we run the experiment for a sufficienctly long time. We suspect this is due to clock drift, e.g., the router's clock and SoNIC's clock are not running at the same frequency. As the drift operates on a longer timescale and is smaller in magnitude compared to inherent variation (in the order of μs over a million packets), our model remains valid as long as the drift is stationary; thus, we can introduce a drift term to the model. A more thorough study is needed, though, to confirm the source and nature of the drift and how it affects the model.

9 CONCLUSION

A model has been developed to form a framework for understanding how inherent variation in a router affects the input-output characteristic of a router. Using the proposed packet clustering metric, we have carefully analyzed over and experimentally validated various setups with different packet sizes, data rates, number of hops, and with or without cross traffic. We have shown that when the interpacket gap is sufficiently large, the interpacket delay at the output will always have a correlation of $-1/2$ with adjacent values, and have a histogram that is symmetrical in shape. In a more general setting, we have shown that, rather surprisingly, it is possible for packet clustering to decrease after passing through a router. More importantly, we show that the rate of change in packet clustering going from the input to the output of a router is decreasing in the interpacket gap. This result allows us to understand how we could potentially reduce jitter by minimizing interpacket gap.

REFERENCES

[1] 2008. IEEE Standard 802.3-2008. Retrieved from https://standards.ieee.org/standard/802_3-2008.html.
[2] 2016. Cisco Catalyst 6500 white paper. Retrieved from https://www.cisco.com/c/en/us/products/switches/catalyst-6500-series-switches/white-paper-listing.html.

[3] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. 1998. RFC 2475: An architecture for differentiated services. Internet Engineering Task Force (IETF) RFC. https://tools.ietf.org/html/rfc2475.
[4] Jean-Chrysotome Bolot. 1993. End-to-end packet delay and loss behavior in the Internet. *ACM SIGCOMM Comput. Commun. Rev.* 23, 4 (Oct. 1993), 289–298. DOI: https://doi.org/10.1145/167954.166265
[5] Lawrence S. Brakmo and Larry L. Peterson. 1995. TCP Vegas: End to end congestion avoidance on a global Internet. *IEEE J. Sel. Areas Commun. (J-SAC)* 13, 8 (1995), 1465–1480.
[6] Andre Broido, Ryan King, Evi Nemeth, and K. C. Claffy. 2003. Radon spectroscopy of inter-packet delay. In *High-Speed Networking (HSN) Workshop*.
[7] H. Jonathan Chao and Bin Liu. 2007. *High Performance Switches and Routers*. John Wiley & Sons.
[8] Roman Chertov and Sonia Fahmy. 2011. Forwarding devices: From measurements to simulations. *ACM Trans. Model. Comput. Simul.* 21, 2, Article 12 (Feb. 2011), 23 pages. DOI: https://doi.org/10.1145/1899396.1899400
[9] Mark E. Crovella and Azer Bestavros. 1997. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Trans. Networking* 5, 6 (1997), 835–846.
[10] Hamza Dahmouni, André Girard, and Brunilde Sansò. 2012. An analytical model for jitter in IP networks. *Ann. Telecommun.* 67, 1–2 (2012), 81–90.
[11] Allen B. Downey. 2001. The structural cause of file size distributions. In *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 361–370.
[12] Daniel A. Freedman, Tudor Marian, Jennifer H. Lee, Ken Birman, Hakim Weatherspoon, and Chris Xu. 2010. Exact temporal characterization of 10 Gbps optical wide-area network. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC'10)*. ACM, New York, NY, 342–355. DOI: https://doi.org/10.1145/1879141.1879187
[13] Nicolas Hohn, Darryl Veitch, Konstantina Papagiannaki, and Christophe Diot. 2004. Bridging router performance and queuing theory. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 32. ACM, 355–366.
[14] Hao Jiang and Constantinos Dovrolis. 2005. Why is the internet traffic bursty in short time scales? In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'05)*. ACM, New York, NY, 241–252. DOI: https://doi.org/10.1145/1064212.1064240
[15] Srikanth Kandula, Dina Katabi, Shantanu Sinha, and Arthur Berger. 2007. Dynamic load balancing without packet reordering. *ACM SIGCOMM Comput. Commun. Rev.* 37, 2 (March 2007), 51–62. DOI: https://doi.org/10.1145/1232919.1232925
[16] John F. C. Kingman. 1970. Inequalities in the theory of queues. *J. R. Stat. Soc. Ser. B Stat. Method.* 32, 1 (1970), 102–110.
[17] Hisashi Kobayashi and Brian L. Mark. 2008. System modeling and analysis: Foundations of system performance evaluation. (2008).
[18] Ki Suh Lee, Han Wang, and Hakim Weatherspoon. 2013. SoNIC: Precise realtime software access and control of wired networks. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI) (NSDI'13)*. USENIX Association, Berkeley, CA, 213–266.
[19] Ki Suh Lee, Han Wang, and Hakim Weatherspoon. 2014. PHY covert channels: Can you see the idles? In *USENIX Symposium on Networked Systems Design and Implementation (NSDI) (NSDI'14)*. USENIX Association, Berkeley, CA, 173–185.
[20] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. 1993. On the self-similar nature of ethernet traffic. *ACM SIGCOMM Comput. Commun. Rev.* 23, 4 (Oct. 1993), 183–193. DOI: https://doi.org/10.1145/167954.166255
[21] David V. Lindley. 1952. The theory of queues with a single server. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 48. Cambridge University Press, 277–289.
[22] Xiliang Liu, Kaliappa Ravindran, and Dmitri Loguinov. 2007. A queueing-theoretic foundation of available bandwidth estimation: Single-hop analysis. *IEEE/ACM Trans. Networking* 15, 4 (Aug. 2007), 918–931.
[23] Yuanqiu Luo and Nirwan Ansari. 2005. Bandwidth allocation for multiservice access on EPONs. *IEEE Commun. Mag.* 43, 2 (2005), S16–S21.
[24] Tudor Marian, Daniel A. Freedman, Ken Birman, and Hakim Weatherspoon. 2010. Empirical characterization of uncongested optical lambda networks and 10gbe commodity endpoints. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 575–584.
[25] Wassim Matragi, Khosrow Sohraby, and Chatschik Bisdikian. 1997. Jitter calculus in ATM networks: Multiple nodes. *IEEE/ACM Trans. Networking* 5, 1 (Feb. 1997), 122–133.
[26] A. Morton and B. Claise. 2009. RFC 5481: Packet delay variation applicability statement. Internet Engineering Task Force (IETF) RFC. https://tools.ietf.org/html/rfc5481.
[27] Kihong Park, Gitae Kim, and Mark Crovella. 1996. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *IEEE International Conference on Network Protocols (ICNP)*. IEEE, 171–180.
[28] Li-Shiuan Peh and William J. Dally. 2001. A delay model and speculative architecture for pipelined routers. In *7th International Symposium on High-Performance Computer Architecture*. IEEE, 255–266.
[29] Li-Shiuan Peh and William J. Dally. 2001. A delay model for router microarchitectures. *IEEE Micro* 21, 1 (2001), 26–34.

[30] Jonathan Perry, Amy Ousterhout, Hari Balakrishnan, Devavrat Shah, and Hans Fugal. 2014. Fastpass: A centralized "Zero-Queue" datacenter network. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, New York, NY, 307–318. DOI: https://doi.org/10.1145/2740070.2626309

[31] Aleksandr Privalov and Khosrow Sohraby. 1998. Per-stream jitter analysis in CBR ATM multiplexors. *IEEE/ACM Trans. Networking* 6, 2 (1998), 141–149.

[32] Henning Schulzrinne, Steven Casner, R. Frederick, and Van Jacobson. 2003. RFC 3550 RTP: A Transport Protocol for Real-Time Applications. Internet Engineering Task Force (IETF) RFC. https://tools.ietf.org/html/rfc3550.

[33] Rishi Sinha, Christos Papadopoulos, and John Heidemann. 2007. Internet packet size distributions: Some observations. Retrieved from http://netweb.usc.edu/~rsinha/pkt-sizes.

[34] Ao Tang, Lachlan L. H. Andrew, Krister Jacobsson, Karl H. Johansson, Håkan Hjalmarsson, and Steven H. Low. 2010. Queue dynamics with window flow control. *IEEE/ACM Trans. Networking* 18, 5 (2010), 1422–1435.

[35] Xinyuan Wang, Douglas S. Reeves, and S. Felix Wu. 2002. Inter-packet delay based correlation for tracing encrypted connections through stepping stones. In *European Symposium on Research in Computer Security (ESORICS)*. Springer, 244–263.

[36] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. 1997. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Trans. Networking* 5, 1 (1997), 71–86.

[37] Sebastian Zander, Grenville Armitage, and Philip Branch. 2007. A survey of covert channels and countermeasures in computer network protocols. *IEEE Commun. Surv. Tutorials* 9, 3 (2007), 44–57.

Received December 2017; revised March 2019; accepted April 2019