8-2016

Modeling Internet Traffic Generations Based on Users and Activities for Telecommunication Applications

Sara Stoudt
Smith College

Pamela Badian-Pessot
Smith College

Blanche Ngo Mahop
Smith College

Erika Earley
Smith College

Jordan Menter
Smith College

See next page for additional authors

Recommended Citation

Stoudt, Sara; Badian-Pessot, Pamela; Mahop, Blanche Ngo; Earley, Erika; Menter, Jordan; Flores, Yadira; Williams, Danielle; Zhang, Weijia; Maharjan, Liza; Bao, Yixin; Rosenbauer, Laura; Nguyen, Van; Mendiratta, Veena; and Tania, Nessy, "Modeling Internet Traffic Generations Based on Users and Activities for Telecommunication Applications" (2016). Mathematics and Statistics: Faculty Publications, Smith College, Northampton, MA.
https://scholarworks.smith.edu/mth_facpubs/34

Authors
Sara Stoudt, Pamela Badian-Pessot, Blanche Ngo Mahop, Erika Earley, Jordan Menter, Yadira Flores, Danielle Williams, Weijia Zhang, Liza Maharjan, Yixin Bao, Laura Rosenbauer, Van Nguyen, Veena Mendiratta, and Nessy Tania

Modeling Internet Traffic Generations Based on Individual Users and Activities for Telecommunication Applications

Sara Stoudt[*a], *Pamela Badian-Pessot*[a], *Blanche Ngo Mahop*[a], *Erika Earley*[a], *Jordan Menter*[a], *Yadira Flores*[a], *Danielle Williams*[a], *Weijia Zhang*[a], *Liza Maharjan*[*a], *Yixin Bao*[a], *Laura Rosenbauer*[a], *Van Nguyen*[a], *Dr. Veena Mendiratta*[b], *and Dr. Nessy Tania*[*a]

[a]*Department of Mathematics and Statistics, Smith College, Northampton, MA*
[b]*Bell Labs, NOKIA, Naperville, IL*

*Students: *sstoudt@berkeley.edu, plb93@cornell.edu, bmahop@wpi.edu, erikatearley@gmail.com, jordanlmenter@gmail.com, yadiflores616@gmail.com, dpwilliams1228@gmail.com, wzhang23@smith.edu, *lmaharjan@smith.edu, ybao@smith.edu, lrosenbauer@smith.edu, vnguyen@smith.edu*
*Mentors: *ntania@smith.edu, veena.mendiratta@nokia.com*

ABSTRACT

A traffic generation model is a stochastic model of the data flow in a communication network. These models are useful during the development of telecommunication technologies and for analyzing the performance and capacity of various protocols, algorithms, and network topologies. We present here two modeling approaches for simulating internet traffic. In our models, we simulate the length and interarrival times of individual packets, the discrete unit of data transfer over the internet. Our first modeling approach is based on fitting data to known theoretical distributions. The second method utilizes empirical copulae and is completely data driven. Our models were based on internet traffic data generated by different individuals performing specific tasks (e.g. web-browsing, video streaming, and online gaming). When combined, these models can be used to simulate internet traffic from multiple individuals performing typical tasks.

KEYWORDS

Internet Traffic Simulation; Stochastic Models; Empirical Copula; Cumulative distribution function; Wireshark

INTRODUCTION

Internet traffic modeling and simulation are useful tools for testing new telecommunication technologies and analyzing the performance and capacity of various network protocols, topologies, and algorithms. The goal of our work was to build mathematical models of internet traffic for multiple users using different applications within the same computer network. Specifically, we collected and analyzed internet traffic data generated by individual users performing specific tasks (e.g. web browsing, e-mailing, video streaming, and online gaming) and then used this data to build mathematical models that simulate internet traffic. Our model takes as inputs the number of users and the proportion of internet applications being used at a given time and outputs artificial traffic data.

Data is transferred over the internet in discrete units known as packets rather than in continuous streams. Each packet can vary in size/length (measured in units of bytes) up to a maximum size that may be imposed by the local network that a user is connected to. Thus, internet traffic consists of the transfers of individual packets, where each packet can be characterized by two random variables: interarrival time and size. We began data collection by capturing the internet traffic generated by a single user utilizing one application. From the data, we built models each describing packet transfers for one user on a single application. We then combined these models to imitate the traffic generated by multiple users using different applications.

As the usage and structure of the internet rapidly changes, mathematical and statistical models for internet traffic are in constant demand. Cleveland et. al.[1] provide the fundamentals of the information found in internet traffic data, and Sanchez et. al.[7] discuss first steps for understanding patterns and properties of interarrival times of packets and bytes per packet. We consider two main approaches to modeling internet traffic data: fitting theoretical distributions to interarrival times and packet size, and building empirical models from representative data. Luo et. al.[3] provide an internet traffic simulation case study whose main features include the fitting of theoretical distributions and visual validation of simulated data. Mah[4] provides an example of an empirical model to describe internet usage. Dainotti et. al.[2] mention the lack of literature about modeling packets and bytes jointly. Possolo[5] describes the background for a copula which can empirically model two distributions jointly. In this paper, we explore both modeling frameworks: theoretical model fitting (by working with known distributions such as the exponential and normal distributions) and empirical model building. We compare and contrast the two methods and discuss the strength and weaknesses of the two approaches.

Additionally, we also consider two different frameworks for modeling internet traffic. We can model at the individual packet level, simulating interarrival time and bytes *per packet*; alternatively, we can also aggregate and predict the number of packets and total bytes that are transferred *per second*. The application for the model determines which framework is more appropriate. If high granularity is a focus, flow of individual packets should be used. However, the interarrival times between packets can be extremely short (order of nanoseconds). Therefore, if a simulation over a longer time frame is needed, modeling packets per second is often more computationally feasible. We constructed our theoretical model using the first framework; for our empirical model, we use the second approach.

METHODS AND PROCEDURES

Data Collection and Preliminary Analysis

Prior to building our computational models, we first collected internet traffic data using Wireshark[8], an open-source packet analyzer. Here, individuals connected to an internet network from their personal computers and captured their packet data using Wireshark for a specific time period (e.g. 10, 15, or 60 minutes). Their internet activity was limited to one of the following: web browsing, video streaming, or online gaming. Most of the data sets were obtained from users connecting to the local wireless network at Smith College. For our mathematical models, we used data collected for 60 minute intervals and focused on two quantities: the packet size and interarrival time of packets. The interarrival times, which are the time intervals between consecutive packets, are obtained by first sorting the timestamps of packet arrivals and then subtracting the consecutive times.

Theoretical Models

Our first modeling approach was to fit the histograms of interarrival time and packet size to a number of commonly used theoretical distributions. Here, we treated the packet size and the interarrival time as two independent random variables. Further on, we describe an empirical modeling approach where the two quantities are described by a joint bivariate probability distribution with no independence assumptions.

One common model for describing waiting times is the exponential distribution based upon the assumption that there is a specific constant probability of an event occurring within a short period of time (constant probability rate)[6]. We found that packet interarrival times cannot be described by a single exponential distribution. Rather, they are better described by a hyperexponential distribution which describes a mixture of several exponential distributions (different rates, each chosen with specific probability). More concretely, we assume that packets can be grouped into multiple categories (e.g. by protocol used, or by whether it is a packet header, trailer, or a payload), labeled as $i = 1, 2, 3, ..., N$. A packet of category i is generated with probability p_i ; within category i, the interarrival time of packets follows an exponential distribution with rate λ_i. Then, the probability density function (PDF) and the cumulative distribution function (CDF) for the interarrival times of packets (random variable t) can be written respectively as:

$$\text{PDF: } f(t) = \sum_{i=1}^{N} p_i \cdot \lambda_i \exp(-\lambda_i\, t), \qquad \text{CDF: } F(t) = 1 - \sum_{i=1}^{N} p_i \cdot \exp(-\lambda_i\, t). \qquad \textbf{Equation 1.}$$

To minimize the number of unknown parameters, we chose $N = 3$ (this was based on trial and error, we found that N=3 is the minimal number needed to obtain a good fit to the data). We obtained parameter values for λ_i and p_i by grouping the data into three intervals: $[0, T_1)$, $[T_1, T_2]$, and strictly greater than T_2. Within each group/interval, we set p_i as the fraction of the data that lies within the interval and λ_i as the inverse of the mean of the data within the corresponding interval (maximum likelihood estimate for the rate parameter of an exponential distribution). We obtained the values for T_1 and T_2 by performing nonlinear least-squares optimization minimizing the difference between the CDF given in **Equation 1.** and the empirical CDF (ECDF) from the data. Denote $\{\tau_k\}_{k=1}^{M}$ as the interarrival time data, sorted from shortest to longest, then the ECDF is a step function with values given by $F_E(\tau_k) = k/M$ where $k = 1, 2, \cdots, M$. The square-difference between the two CDFs is then computed according to

$$D(T_1, T_2) = \sum_{k=1}^{M} \left(F_E(\tau_k) - F(\tau_k; T_1, T_2)\right)^2. \qquad \textbf{Equation 2.}$$

Note that the value of the CDF F in **Equation 1.** is dependent on the parameter values $\{p_i, \lambda_i\}$ which in turn are dependent on the values of the interval end-points T_1 and T_2. Minimization of the square-difference function D is done by using the Matlab function fmincon (T_1 and T_2 are constrained so that $0 < T_1, T_2 < \tau_M$).

To simulate an interarrival time for the next packet (random variable t), we draw from the probability distribution in **Equation 1.** as follows:

1. Generate a pair of random variables r_1 and r_2 from the uniform distribution in $[0, 1]$.
2. If $r_1 \leq p_1$ then choose the rate λ_1 and convert r_2 into an exponentially distributed random variable via $t = -\log(r_2)/\lambda_1$.
3. Else if $p_1 < r_1 \leq p_2$, then choose rate λ_2 and $t = -\log(r_2)/\lambda_2$.
4. Otherwise, $r_1 > p_2$, so choose rate λ_3 and $t = -\log(r_2)/\lambda_3$.

As will be discussed later on (see histogram on **Figure 4**), packet length distributions generated by different applications tend to be bimodal, so we fitted the data to two different normal distributions separated by a uniform distribution. Thus, we again consider three different possibilities: a packet length is generated with probability f_1 from a normal distribution with mean μ_1 and standard deviation σ_1, with probability f_2 from a normal distribution with mean μ_2 and standard deviation σ_2, and with probability $f_3 = 1 - (f_1 + f_2)$ from a uniform distribution in $[L_1, L_2]$. Then the cumulative distribution function for the packet size (random variable l) can be written by combining the CDFs for the normal and uniform distributions:

$$G(l) = \frac{f_1}{2}\left[1 + \operatorname{erf}\left(\frac{l - \mu_1}{\sqrt{2\sigma_1^2}}\right)\right] + \frac{f_2}{2}\left[1 + \operatorname{erf}\left(\frac{l - \mu_2}{\sqrt{2\sigma_2^2}}\right)\right] + f_3\left[\frac{l - L_1}{L_2 - L1}\right]. \qquad \textbf{Equation 3.}$$

For data-fitting, we again binned the data by choosing two cut-off values, L_1 and L_2. Further inspection of the data revealed that a proportion of the packet length data has a constant value of 1440 bytes (likely the standard size for payload packets in our network), so we use $L_2 = 1440$ bytes and set $\mu_2 = 1440$ and $\sigma_2 = 0$ (mean and standard deviation of the normal distribution corresponding to the second peak in the bimodal histogram). L_1 is chosen through a least-squares procedure.

The algorithm for drawing from this packet length distribution is also similar to that for interarrival time. However, rather than generating a random variable that is exponentially distributed, here we use either a normal or uniform distribution.

The gaming application is a special case as the packet length distribution is not bimodal and has an exponential shape instead. We observed that online gaming can be more data intensive (e.g. some packets are larger than 4000 bytes) and the packet size has a higher standard deviation than other applications. To describe the packet size distribution for gaming, we fit the data to an exponential distribution with rate α. The parameter values obtained from fitting the data for different applications are listed in **Table 1**.

Interarrival Times **Packet Length**

Video Streaming

Rates (/s)	**Frequency**
λ_1 = 2.68e+05	p_1=0.2706
λ_2 = 1.42e+03	p_2=0.6997
λ_3 = 4.58	p_3=0.0297

Parameters	**Frequency**
μ_1 = 60.24, σ_1 = 10.80	f_1=0.3694
μ_2 = 1440, σ_2 = 0	f_2= 0.5930
Uniform [100, 1440]	f_3=0.0376

Web Browsing

Rates (/s)	**Frequency**
λ_1 = 3.97e+05	p_1=0.3851
λ_2 = 1.26e+03	p_2=0.3813
λ_3 = 7.63	p_3=0.2285

Parameters	**Frequency**
μ_1 = 69.09 σ_1 = 18.15	f_1=0.5640
μ_2 = 1440 σ_2 = 0	f_2=0.2824
Uniform [120, 1440]	f_3=0.1536

Online Gaming

Rates (/s)	**Frequency**
λ_1 = 1.14e+04	p_1=0.3604
λ_2 = 59.03	p_2=0.6371
λ_3 = 5.88	p_3=0.0025

Parameters	**Frequency**
α = 0.0068	1

Table 1. Table of parameter values for different applications.

Empirical Copula Method

As an alternative to theoretical distributions, we use the copula method to simulate internet traffic. The copula is a flexible, non-parametric approach that does not rely on any distributional assumptions. On the other hand, we may not obtain certain information that convey average behaviors that can be inferred from parameter values obtained from data-fitting to the previous theoretical models.

The copula method can be used for internet traffic simulation at a fine individual packet level (interarrival times in the order of nano seconds). However, this approach is not computationally feasible for simulations with longer time scales, so we instead simulated the number of packets and total bytes that are generated per second. Either way, the copula method allows us to model the two random variables of interest jointly, rather than assuming that they are independent. This method is completely data driven which allows it to be easily applied to any data set. Different copulae can be made to reflect different types of users based on usage, time of day, spatial location, etc. We model user variability by modeling each type of application separately. However, we did not distinguish between different users performing the same task.

Based on a data set from one user with a specific task, we can build a copula as follows:

1. Find the empirical cumulative distributions for packets-per-second and bytes-per-second.
2. Map the corresponding ECDF values to a unit square by using the ECDF of packets-per-second as the x-coordinate and the total bytes-per-second as the y-coordinate.

These steps are schematically shown in **Figure 1**.

Once we generate a copula surface, we can draw from this joint distribution to obtain the two random variables of interests as follows (see schematics illustration in **Figure 2**):

1. Simulate packets-per-second by pulling a random number x from the uniform distribution in [0,1]. This gives an ECDF value for the packets-per-second variable (shown in **Figure 1**) which can then be inverted to obtain the packets-per-second value.

2. Next, simulate bytes-per-second conditional on x; this is akin to taking a slice of the copula surface. Fixing x, draw from the probability distribution function for the ECDF values for the second variable as illustrated in **Figure 2**. This random number (in [0,1]) corresponds to an ECDF value and yields the simulated value for bytes-per-second after inversion.

3. Repeat this process many times until the elapsed time is achieved. Each pair of random variables from steps 1-2 above is a simulated second of internet traffic for one user. We can combine simulations from different copulae in a simulator.

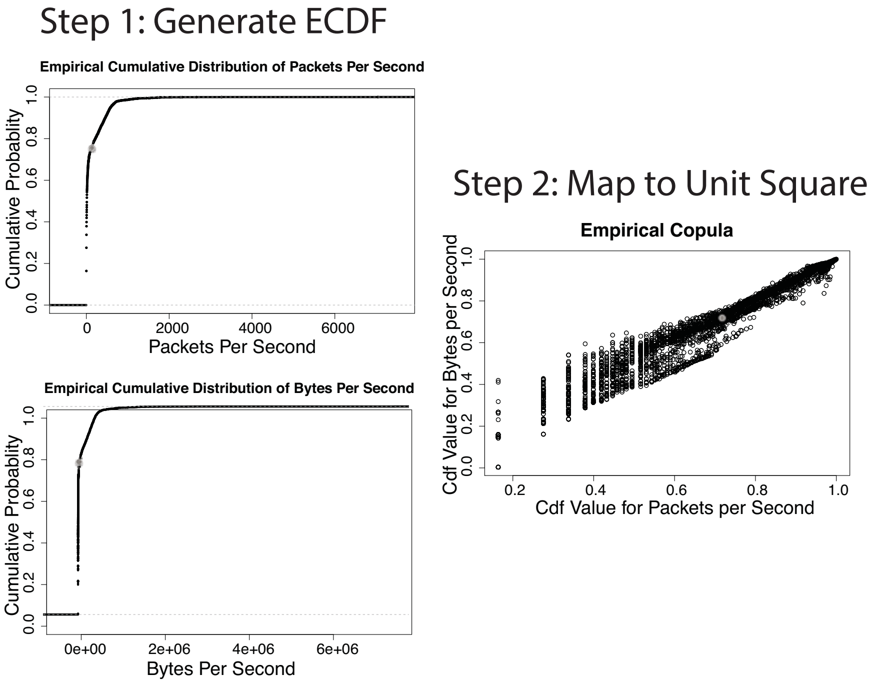

Figure 1. Schematics of steps for building copula.

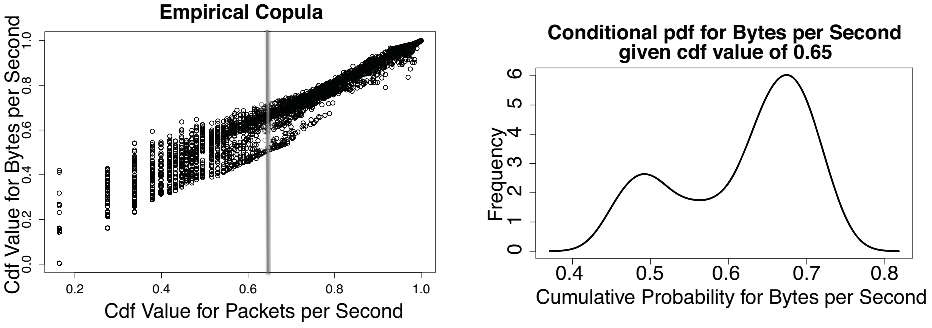

Figure 2. Schematics for simulating from a copula.

RESULTS AND DISCUSSION

Preliminary Data Analysis

Prior to building and simulating the computational models, we analyzed the interarrival time and packet length data collected from a single user performing specific applications for 60 minutes. In **Figure 3**, we show the density plots (normalized histogram) for the distribution of interarrival times for different applications. Here, the interarrival time is shown in log scale as their values are typically short, but can vary by orders of magnitudes. The density plots showing the distribution of packet sizes are shown in **Figure 4**. With the exception of online gaming, the distribution of packet size appears to be bimodal (distributions with two peaks). We expect that the smaller packets carry the "control information" (packet headers/trailers) that contain short information needed to deliver the packet, while the larger packets contain the actual data ("payload").

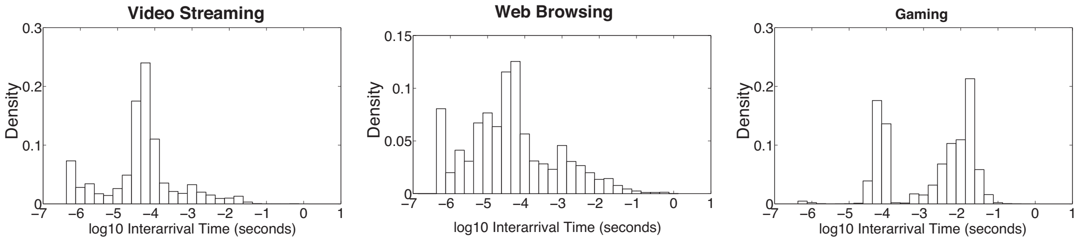

Figure 3. Density plots of interarrival time for applications. Each histogram corresponds to one data set collected from one user performing one task for 60 minutes.

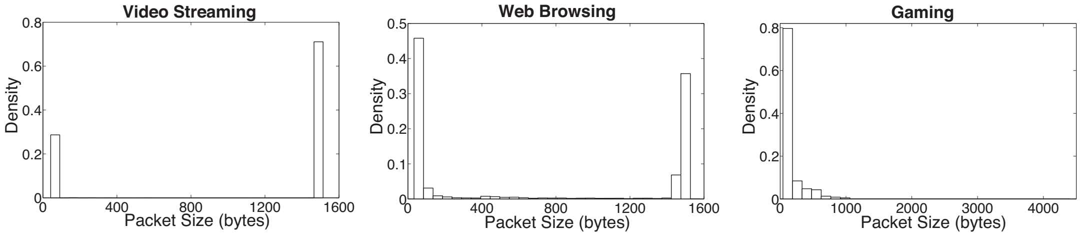

Figure 4. Density plots of packet sizes for different applications. Each histogram corresponds to one data set collected from one user performing one task for 60 minutes.

Simulation Results from Theoretical Models

Using parameter values obtained from data fitting in **Table 1**, we first assessed the fit of the model (using parameter values obtained from least-square optimization) to the data. In **Figure 5**, we compared the cumulative distribution functions from the model to the empirical cumulative distribution functions of the data. The models fit the data relatively closely though the models do not fully capture the variations in the data. We then simulated the timing and size of individual packets by drawing from respective theoretical distributions given in **Equation 1.**, **Equation 3.** Combining the theoretical distributions for different applications (parameter values listed in Table 1), traffic for multiple users can be simulated. For simplicity, we assume that each user only performs a single task but we can specify the proportions of users performing each activities. On **Figure 6**, we show the traffic generated by 15 users (10% gaming, 40% video streaming, and 50% web browsing).

One advantage of using the theoretical method is fast computation at a fine packet size level. Here, the random variables of interest (namely packet lengths and interarrival times) can be generated quickly as they involve standard distributions (normal, uniform, or exponential) that can be inverted/transformed easily.

On the other hand, data-fitting to theoretical distributions also presents some limitations. First, we assumed that the interarrival times and packet lengths are two independent random variables. In reality, these may be dependent; for example,

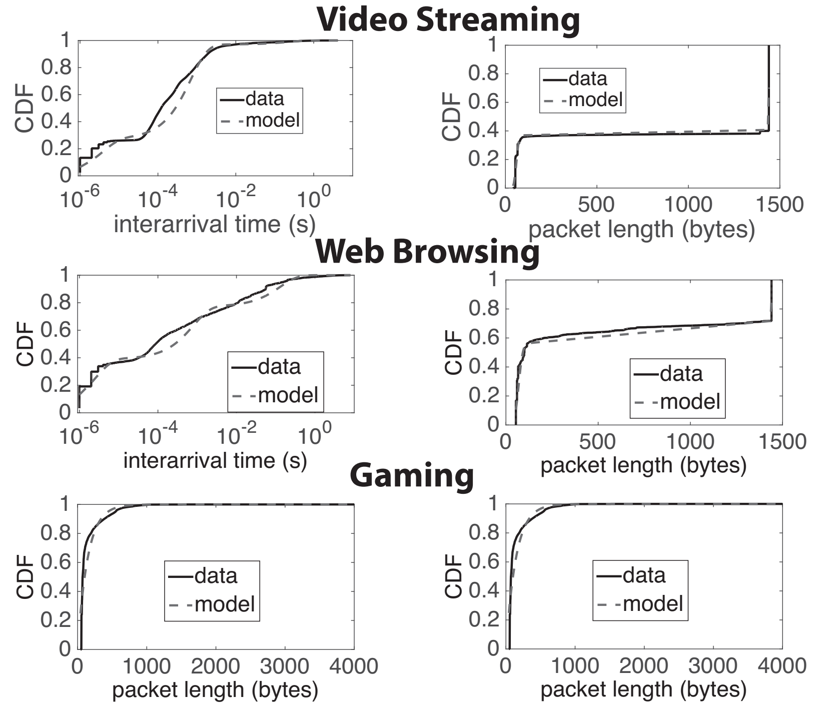

Figure 5. Data fitting results: comparison of ECDFs from data and simulation from theoretical models for different activities.

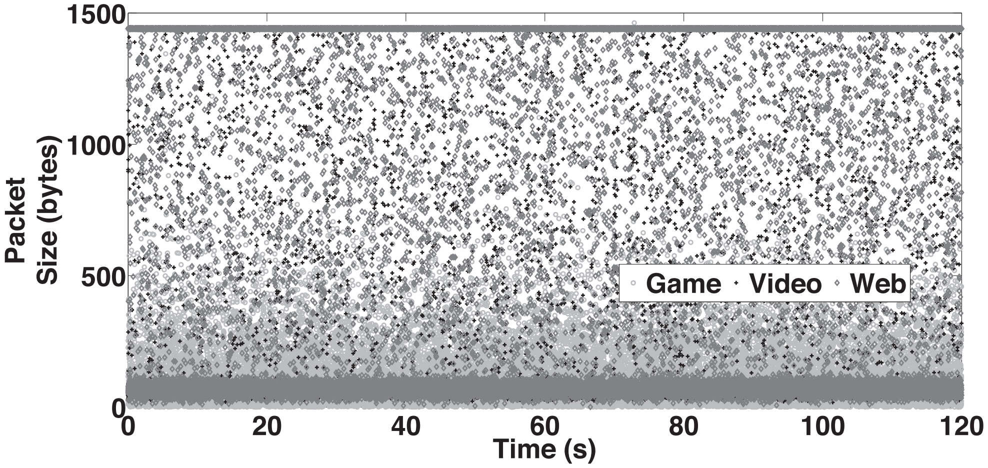

Figure 6. Internet traffic simulation obtained from the theoretical model for 15 users performing the following fractions of activities: (10% gaming, 40% video streaming, and 50% web browsing).

smaller packets carrying control information may be generated more quickly to establish communications between computers, while larger packets carrying payloads are dependent on user activities (longer time scale than nanoseconds) and may be limited by the speed of data transfer (bandwidth cap) imposed by the network. Next, the simulation results are also dependent on the parameter values, which can vary significantly from one data set to the next. The least squares data-fitting procedures can be sensitive to the choice of initial guesses. We now turn to an empirical method that allows us to overcome some of these limitations.

Simulation Results from Empirical Copula Method

We first built the copula surface for each activity and plotted the copulae in **Figure 7**. Each activity generates a copula with distinct features, confirming our approach that we should model them separately. Combining the different copulae, we can simulate multiple users performing different tasks. A simulation of internet traffic for 10 users (25% Video, 50% Web, 25% Gaming) over the course of an hour is shown in **Figure 8**. Computational costs associated with the copula method are higher as the method requires repeated resampling of a large data set. Here we reduced the computational time by considering the number of packets and bytes transferred in a given second, rather than simulating individual packets.

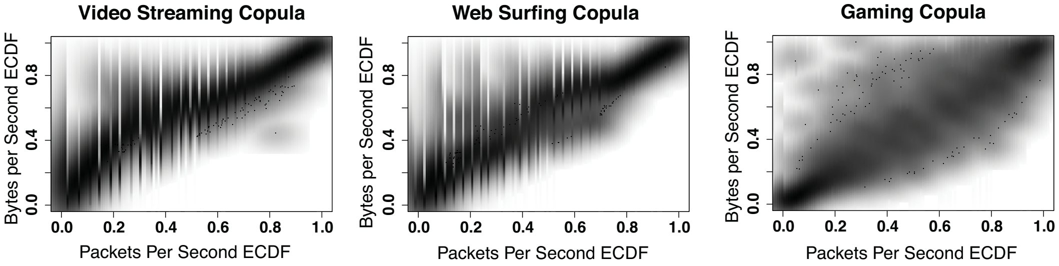

Figure 7. Copulae for different activities.

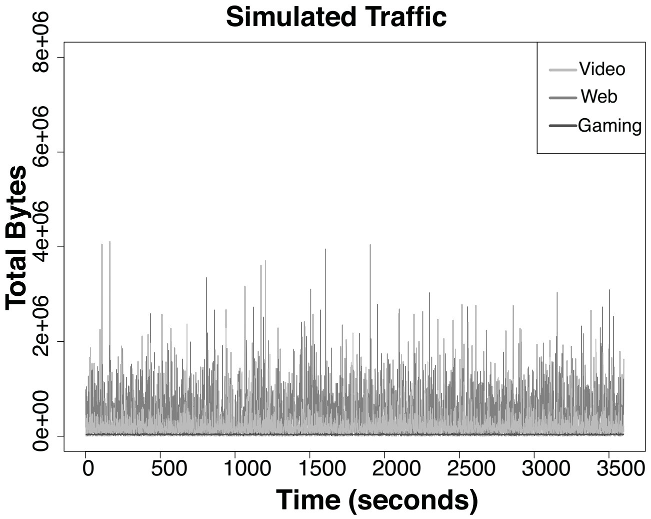

Figure 8. A simulation from the copula method for internet traffic for 1 hour from 10 users (25% video, 50%, and 25% gaming).

Model Verifications and Comparisons

We first compared simulations from both the theoretical model and the empirical model. To do this, we built each model on the same data sets and generated simulations based on the same inputs (number of users, time frame, and proportion of users performing each activity). Since the copula method is completely data driven and a finer granularity gives us more data to build the copula, we wanted to make sure that the empirical method was not adversely affected by our choice of fairly small data sets. Therefore, we compared results at the second level, half-second level, and quarter second level to ensure that our time scale does not affect the comparisons. For brevity, we show the results at the second level here, but the results from the half-second and quarter second levels were similar.

First, we assess the simulations from these two methods visually. We can see in **Figure 9** that the simulations from the theoretical model have less variability than those from the empirical method. It is reassuring to see a sense of consistency;

both models have a similar center. Depending on the application, there could be advantages to having different properties in the simulations. For capacity testing, it may be beneficial to have simulations that have higher variability and capture the information in the tails of the internet traffic. For an application where we are more interested in ascertaining information about the "average" user, the theoretical model may be preferred.

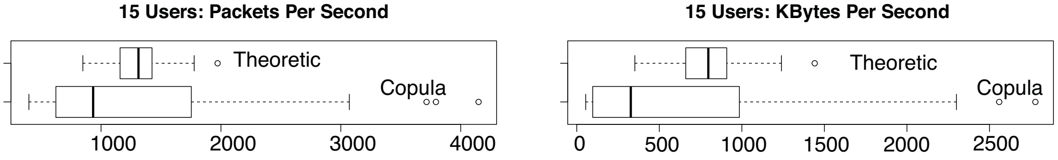

Figure 9. Comparison of simulation results from theoretical and empirical (copula) models.

We then measure the goodness of fit of both models. However, the empirical copula method is non-parametric (no parameter fitting was performed). Therefore, standard goodness of fit techniques for assessing how well the model conforms to the data, do not apply. Therefore we need another way to assess fit. One way we can do this is to mimic the Kolmogorov-Smirnov test; this test measures the maximum difference between the values of the empirical cumulative distributions from model simulations and the data.

To get a sense of how this would change given different data, we can "shuffle" our data set (by drawing with replacement a sample of 1000 points from the full data set) and then compute the maximum difference in empirical cumulative distribution functions. When we do this many times, we get a distribution for our maximum difference. We can do this for many different types of methods, including the theoretical model, to get a sense for how well each model works. To put our results in context we can test a naive model as our "worst case scenario" and a true data shuffle as our "best case scenario." We choose our naive model to be a uniform copula, a flat unit square.

In **Figure 10**, we can see that, as expected, the simple resampling of the true data yields the smallest maximum difference centered, for example around 0.05 for web packets, and the uniform copula yields the largest maximum difference centered around 0.912 for web packets. The copula method yields maximum difference values closer to those of the resampling. In fact, the distributions for these two overlap. The theoretical method has larger maximum differences but note that here we were comparing coarser grained result (per-second level) while data-fitting for the theoretical model was done at the fine individual-packet level. The same relative ordering of the goodness of fit measure holds for the packets and bytes for web surfing, video streaming, and online gaming with the distributions for resampling and the copula overlapping, and the distributions for the theoretical method being closest to those of the uniform copula method. Nonetheless, the theoretical model still yields maximum differences that are smaller than the uniform copula method.

CONCLUSIONS

We have described two approaches for generating artificial internet traffic consisting of packet lengths and interarrival times simulations. After collecting individual user data, we produced models that describe traffic generated during web surfing, video streaming, and gaming, which were then combined to form an internet traffic simulator. We considered two methods. The first method is based on fitting known theoretical distributions to the data to simulate individual packets; the second method used an empirical copula to simulate packets per second. The theoretical model allows for fast computation, but packet length and interarrival time are assumed to be independent variables. This limitation is removed when the copula method is used. However, the copula method can be computationally expensive, especially for simulating individual packets with very short interarrival times. To speed up computations, we simulated packets per second instead. When we evaluated how well each methods capture the underlying data, we found that the theoretical method models the center of the distribution well but does not capture the tails as well as the empirical method. Both models show significant improvement over a naive approach with the empirical method slightly outperforming the theoretical method (measured by Kolmogorov-Smirnov statistics i.e. ECDF max difference for comparison).

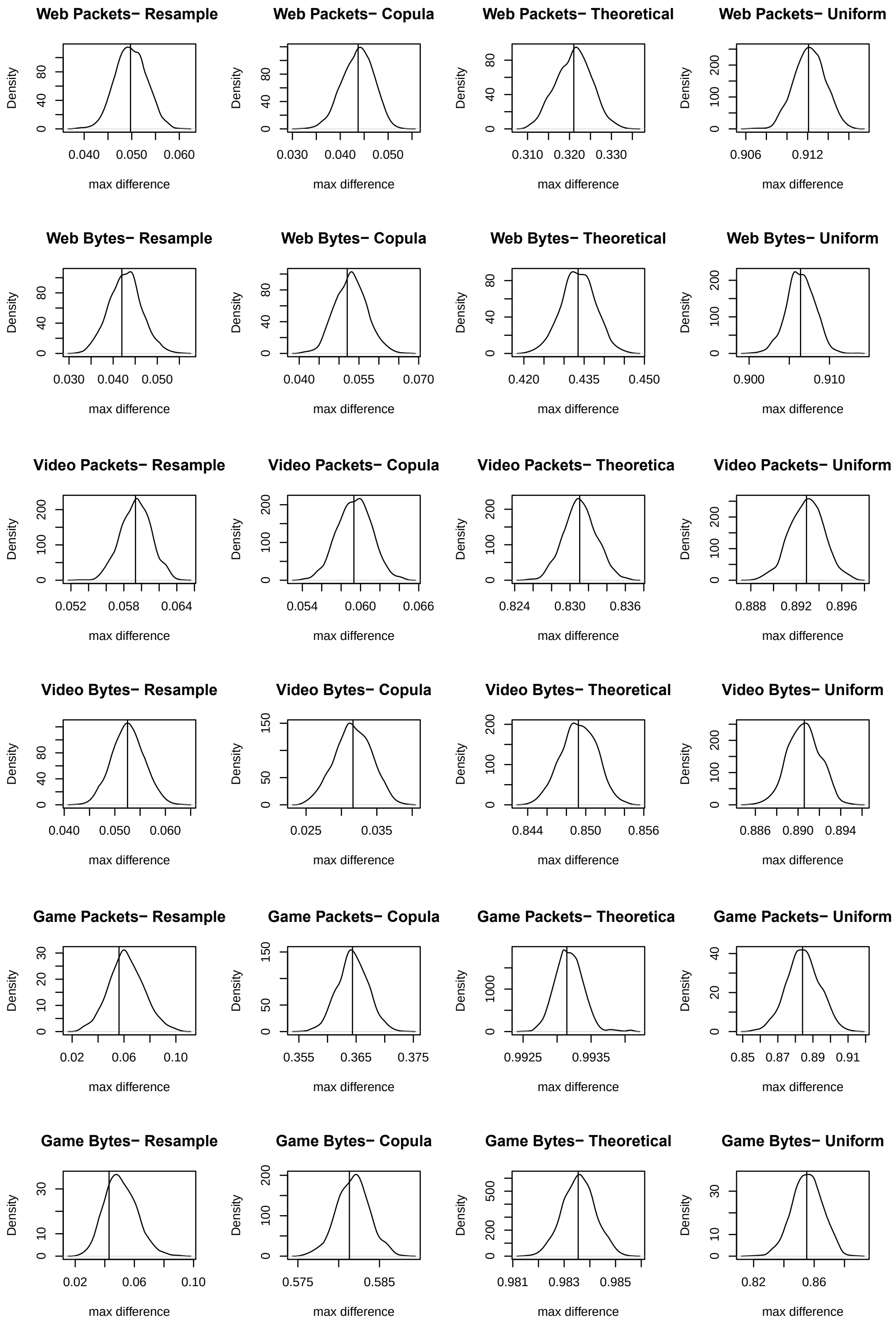

Figure 10. Maximum difference in ECDF comparisons.

The methods presented here form an underlying basis for developing more realistic models in the future. Here, we do not include any network signaling effects, which describe the relationship between user requests and network response. There is a difference between what a user asks for and what the network returns; this may depend on the network structure itself as well as the load or number of users utilizing the network. For example, if many users are inundating the network with requests, it may return less than the requested amount per user based on a capacity constraint. Our traffic simulation results can also be compared to a larger data set from a network utilized by multiple users at different time periods during the day (capturing varying traffic loads and the network limitation). We can then also include suitable dampening effects in our simulator as the number of users increases.

Each distinct protocol used for packet transfer may exhibit different interarrival time and packet length distributions as each protocol serves different functions (e.g. security/encryption, peer-to-peer network establishment, or information transfer). In the future, more realistic models which take account protocol signaling can be developed. To further explore these effects, the Wireshark data can first be reanalyzed and the model can be expanded to include the directions of packet transfers. Protocol signaling may also lead to bursts of packet transfers. The theoretical method presented here is based on Poisson processes with no memory thus does not capture any bursting behavior. Packet simulations using the copula method are also memory-less in that the generation of consecutive packets are independent from one another; however, bursting tendencies may be partially captured if there is a higher density for shorter interarrival times in the copula surface. Further analyses of correlations of interarrival times or size between consecutive packets is an interesting future direction in order to investigate the extent of "memory" between consecutive packet transfers.

Another possible route to take in the future is the collection of a wider variety of applications that generate internet traffic, e.g. music streaming, file-sharing, and real time audio sessions (e.g. with Skype and WhatsApp). As we build up these approaches with a wider variety of applications, it will be interesting to compare the performance of our models with standardized internet traffic simulators such as the Third Generation Partnership Project 2 (3GPP2)[9] and the ns-3 Network Simulator[10].

ACKNOWLEDGEMENTS

This project was funded by NSF grant DMS-1345499 through the MAA Preparation for Industrial Careers in Mathematical Sciences Program (PIC Math) and NSF grant DMS-1143716 for the Center for Women in Mathematics, Smith College. We thank Smith College ITS for consultations on campus-wide data. We are also grateful for helpful feedback from other research teams at Bell Labs.

REFERENCES

1. Cleveland, W.S. and Sun, D.X. (2000) Internet Traffic Data, *J Am Stat Assoc* 95, 979–985.
2. Dainotti, A., Pescape, A., Rossi, P.S., Palmieri, F., and Ventre, G. (2008) Internet Traffic Modeling by Means of Hidden Markov Models, *Comput Netw* 52, 2645–2662.
3. Luo, S., and Marin, G.A. (2005) Realistic Internet Traffic Simulation Through Mixture Modeling and a Case Study, in *Proceedings of the 37th Conference on Winter Simulation*, 2408–2416.
4. Mah, B.A. (1997) An Empirical Model of HTTP Network Traffic, in *Proc Infocom '97*, 592–600.
5. Possolo, A (2010) Copulas for Uncertainty Analysis, *Metrologia* 47, 262–271.
6. Ross, S (2002) A First Course in Probability, 6th ed., Prentice Hall, New Jersey.
7. Sanchez, J. and He, Y (2005) Internet Data Analysis for the Undergraduate Statistics Curriculum, *J Stat Educ* 13, 1–20.
8. Wireshark, development release 1.12.5, *https://www.wireshark.org* (accessed June 2016).
9. Third Generation Partnership Project 2, *http://www.3gpp2.org/* (accessed August 2016).
10. ns-3 Network Simulator, *https://www.nsnam.org/* (accessed August 2016).

ABOUT THE STUDENT AUTHORS

Sara Stoudt is currently a Ph.D. student in the Department of Statistics at the University of California, Berkeley. She is also a participant in the Environment and Society: Data Science for the 21st Century training program and a National Physical Science Consortium Fellow with the National Institute of Standards and Technology. She completed a B.A. in Mathematics from Smith College in 2015. She is interested in applied and computational statistics with emphasis on environmental data problems.

Pamela Badian-Pessot is currently a Ph.D. student at the School of Operations Research, Cornell University. She completed a B.A. in Mathematics and Economics from Wells College in 2013, and the Post-Baccalaureate Program at the Center for Women in Mathematics, Smith College in 2015.

Blanche Ngo Mahop is currently pursuing a Ph.D. in Computational and Applied Mathematics at the Worcester Polytechnic Institute. She completed a B.S. in Applied Mathematics from Howard University in 2014, and the Post-Baccalaureate Program at the Center for Women in Mathematics, Smith College in 2015.

Erika Earley is currently working as a resident software engineer at Google in New York City. She completed a B.A. in Mathematics from Smith College in 2015.

Jordan Menter is currently working as a Data Analytics Consultant with the MassMutual Data Science Development Program in Amherst, MA. She will begin pursuing an M.S. in Computer Science at the University of Massachusetts, Amherst in the fall of 2016. She completed a B.A. in Mathematics from Smith College in 2016.

Yadira Flores is currently an Application Developer for Wharton Research Data Services in Philadelphia, PA. She completed a B.A. in Mathematics from Smith College in 2015.

Danielle Williams is currently pursuing a career in education in the School District of Philadelphia. She completed a B.S. in Mathematics and Secondary Education from East Stroudsburg University in 2014, and the Post-Baccalaureate Program at the Center for Women in Mathematics, Smith College in 2015.

Weijia Zhang (Vega) is currently interning for the Quantitative Management Associate Program at Bank of America. She is pursuing her B.A. in Mathematics and Statistics at Smith College. She is interested in using quantitative skills solving problems, especially in the field of finance.

Liza Maharjan is a current undergraduate student at Smith College pursuing her undergraduate degree in Mathematics and Economics. She plans to pursue an M.Phil. in Development Studies at Oxford University following graduation in 2017.

Yixin Bao is currently pursuing her B.A. in Mathematics at Smith College. She also studied abroad at the London School of Economics, Department of Statistics during her junior year.

Laura Rosenbauer is currently pursuing her B.S. in Engineering at Smith College.

Van Nguyen is currently pursuing her B.A. in Computer Science and Economics at Smith College.

PRESS SUMMARY

Computational models that can generate artificial internet traffic are useful tools for testing the limits and robustness of new telecommunication technologies. In this paper, we collected data in order to build computational models that take into account the arrival time of packets (units for information transfer over the internet) and their sizes. We then developed and compared two distinct mathematical/statistical approaches for building our internet traffic models; one model is computationally faster but less accurate, while the other is more computationally expensive but can capture extreme points seen in the data. Thus, depending on the desired accuracy and computational power, users can utilize our models to generate artificial internet traffic that takes into account the number of users and the activities performed (e.g. web-browsing, video streaming, online gaming).