

MODELING THE DELAYS OF SUCCESSIVELY-TRANSMITTED INTERNET PACKETS

Mark Kalman and Bernd Girod

Information Systems Laboratory
Stanford University
{mkalman, bgirod}@stanford.edu

ABSTRACT

We introduce an Internet packet delay and loss model for use in rate-distortion optimized media streaming. Our scheme uses feedback to learn the delay over the channel in the recent past and then finds probability mass functions for delays at later times by using chains of approximated conditional delay distributions. With an Internet delay trace we have collected and a performance metric that we introduce, we show that our scheme is far more accurate than the commonly-used model which assumes packet delays to be i.i.d. Γ -distributed, at predicting arrival outcomes for groups of successively-transmitted Internet packets. In addition, we outline our Linux-based one-way delay trace collection techniques and we show the suitability of the shifted Γ distribution for modeling conditional packet delays over Internet links that include a cable modem last hop.

1. INTRODUCTION

In this paper we present a model for packet delay and loss over the Internet that improves upon those used in schemes for rate-distortion optimized packet transmission scheduling [1]. These schemes aim to select a transmission sequence for media packets in a way that optimizes the playout quality of the media stream while meeting transmission rate constraints. The schemes rely on accurate statistical models for the delay and loss experienced by groups of packets transmitted over the Internet in order to find an optimized transmission sequence.

To create and evaluate a model of Internet packet delay and loss, we first needed to collect and analyze Internet packet delay traces. Collecting accurate traces is a subject in itself [2, 3, 4]. In this paper we outline a method of collecting traces that makes use of two network hosts running Linux.

In previous works on optimized streaming, the delays of successive packets have been modeled as i.i.d. Γ -distributed random variables, with loss also occurring independently [1]. In some works, the parameters of this distribution are modulated from one transmission time to the next by an underlying K -state hidden Markov model [5]. The i.i.d. model has the obvious shortcoming that it does not capture any dependence among successive delay times. It performs better when parameters are estimated on the fly, as described in [1] but even then, for instance, it cannot capture the property that packets rarely arrive out-of-order on most Internet links. The K -state models with a small number of states will have similar behavior to the i.i.d. model. With a large number of

states, the question arises of how to estimate state transition probabilities and state-dependent delay and loss distributions.

The scheme that we propose, which models successive delays over the channel as a 1st order discrete Markov process, can be viewed as a K -state model with a state for each discretized delay outcome. In our scheme, the state of the channel at a given time is the discretized delay (with loss modeled as infinite delay) that a packet sent over the channel at that time would experience. In this paper we use a collected Internet delay trace to show that the state transition probabilities needed for such a scheme can be approximated with shifted Γ distributions, and we show that the scheme outperforms the i.i.d. Γ model.

We begin in Sec. 2 with a discussion of our Linux-based trace collection methodology in light of the trace collection guidelines suggested in [2]. In Sec. 3 we examine the conditional distributions of successive delays in a collected Internet delay trace. In Sec. 4 we present our model. In Sec. 5 we present results comparing our model with the i.i.d. delay and loss model, using a performance metric that we introduce.

2. TRACE COLLECTION METHODOLOGY

We developed and tested our channel models through the use of a packet delay and loss trace measured over an Internet link with a cable model last hop. As detailed in [2, 3, 4], collecting accurate traces can be challenging. In this section we discuss our trace collection methods and present some trace attributes as suggested by the guidelines in [2].

Our traces are lists of one-way delay measurements taken bidirectionally between two Internet hosts. During the trace collection experiments, each host would transmit a small probe packet every 50 ms. The delays for the probe packets were computed, and the final traces are a listing, for each direction, of send times and delays.

Clock synchronization is a major issue in one-way delay trace collection, because send times are recorded with one host's clock and arrival times with the other's. To synchronize clocks we collected estimated offsets with two-way packets, and performed a supervised, piecewise, linear regression to interpolate offset values. During trace collection, one host would send two-way packets every 200 ms in addition to the probe packets. The host receiving a two-way packet would note the time and send it back to its origin. Afterwards, the two-way packets with round-trip times (easily measurable since send and round-trip receive times are taken with the same clock) near the minimum recorded round-trip time were used to calculate offset estimates. Assuming symmetric delays in both directions and no congestion (hence minimum round-trip time) in either direction, the one-way trip time can be estimated to

This work was funded by a generous grant from ST Microelectronics Inc.

be half of the round-trip time. The offset estimate is then the difference between half the round-trip time, and the one-way delay as measured by the two clocks. For each round-trip packet two offset estimates can be taken because there is a measured delay for the forward trip, and another for the return trip. If the link is asymmetric, the error in the offset estimate due to asymmetry will have opposite sign for the estimate taken with forward trip and with the return trip. When the two are averaged, the asymmetry error drops out [3]. Over 50 second periods, or periods bounded by jumps in the clock values seen in recorded sending times as described in [4], round-trip packets within a millisecond margin of the minimum round-trip time were used as offset estimate samples to find the constant and linear drift terms for the offset that were the best fit in the least-squares sense.

Running a general purpose operating system like Linux on the trace collection hosts introduces other potential sources of timing inaccuracy. The OS shares execution time among multiple processes. Clock readings that record transmission and arrival times will be inaccurate when the trace collection process is swapped out at inopportune times. To avoid this problem we use Linux's `sched_setscheduler` system call to set the OS's process scheduler to FIFO, and then we set our process to the highest priority. This way our trace collection process takes over the whole machine and does not compete with other processes for execution. Other measures that we take are to use the `ioctl` system call to get packet arrival times according to the kernel's packet arrival handling interrupt instead of a later call to `gettimeofday` in our user-level program. In addition, all of the timing is done with busy loops instead of with potentially inaccurate timers, and all read calls are set to non-blocking so that timing loops can continue uninterrupted.

In traces taken over an ethernet link to assess timing accuracy, accuracy was typically in the 10s of μ s, however, delay spikes of up to 2 ms due maybe to collisions on the ethernet but more likely due to timing errors in the host were noted. We thus can only claim that the best accuracy we could achieve is 2 ms. Examining traces over Internet links, we noted that the offset estimation led to minimum one-day delays that vary up to 2 ms, leading us to conclude that the offset estimation yielded an additional two or three ms of inaccuracy. We thus quantized traces to bins 5 ms wide, and estimate roughly that the accuracy is plus or minus one bin, or approximately 10 ms.

For the results shown in the rest of the paper, we collected a 5-hour trace between a machine at Stanford University and a residence in Mountain View, California which connected to the Internet via cable modem. There were 14 Internet hops between the two machines. The downlink bandwidth to the cable modem from the Stanford host, as estimated by an ftp file transfer, was in excess of 1.5 Mbits per second. The uplink bandwidth was approximately 230 Kbits per second. The trace collection process sent 25 (20 one-way and 5 round-trip) 32 byte packets (IP header, UDP header, and sequence number) per second for a total of 6.4 Kbits per sec. The probe packets thus utilized less than 2.8 % of the available upstream bandwidth, and 0.42 % of the available downstream bandwidth.

3. CONDITIONAL DELAY DISTRIBUTION IN THE CABLE MODEM LINK TRACE

The channel model that we will present in Sec. 4 makes use of conditional probability mass functions (pmfs) $p(x_n|x_{n-1})$ where the

random variables X_n are the succession of delay outcomes over the channel at 50 ms intervals. The purpose of this section is to show that these pmfs are approximated well by shifted Γ distributions.

Fig. 1 shows the agreement between estimates of $p(x_n|x_{n-1})$ (shown as solid lines) compiled empirically from our 14-hop cable modem trace, and shifted Γ approximations (shown as dashed lines), for $x_{n-1} = 35, 70, 140,$ and 280 ms (the minimum one-way delay for the trace was 35 ms). These results are similar to those discussed in [6], however in that work the inter-arrival times $X_n - X_{n-1}$ of round-trip packets were observed to be Γ -distributed, and here we are examining conditional delay distributions.

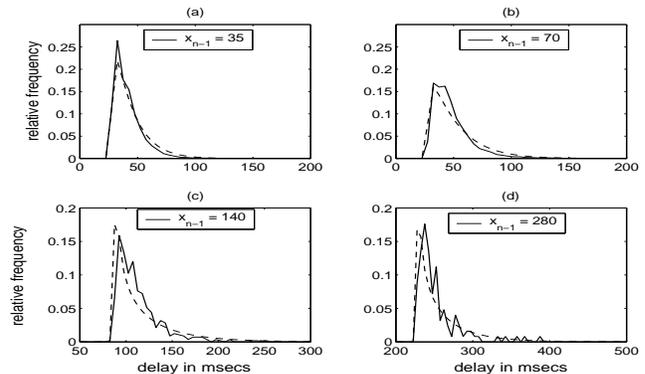


Figure 1: Agreement between empirically compiled estimates of $p(x_n|x_{n-1})$ (solid lines) and shifted Γ approximations (dashed lines) for the measured delay and loss trace for the 14 hop path between Stanford University and a cable-modem connected host in Mountain View, California.

In Sec. 5 our model will make use of shifted Γ approximations to the pmfs $p(x_n|x_{n-1})$. The shifted Γ approximations can be specified with three parameters: shift κ , mean μ , and variance σ^2 . Fig. 2 (a), (b), and (c) plot these parameters as measured from the 14-hop cable modem trace. The solid lines in the plots indicate the parameter values as measured. The shift κ is measured to be the point, starting from zero, at which the empirical distribution becomes non-zero. The mean μ is shown with the shift κ removed. The dotted lines in Fig. 2 (a), (b), and (c) indicate the approximated parameter values that will be used in Sec. 5.

4. PROPOSED CHANNEL MODEL

Suppose we would like to model the random delay outcomes X_0, X_1, \dots, X_N for a group of successively transmitted packets. The best model would be the joint distribution, $p_{X_0, X_1, \dots, X_N}(x_0, x_1, \dots, x_N)$ conditioned on any delay information that is available. Since the joint distribution is generally not available, we must approximate. The simplest approximation is to assume the delays are independent. This gives

$$p_{X_0, X_1, \dots, X_N}(x_0, x_1, \dots, x_N) = p_X(x_0)p_X(x_1) \dots p_X(x_N).$$

In our model we instead assume the Markovity of successive delays. We thus approximate the joint distribution with chains of conditional distributions

$$p(x_0, x_1, \dots, x_n) = p(x_0)p(x_1|x_0) \dots p(x_n|x_{n-1})$$

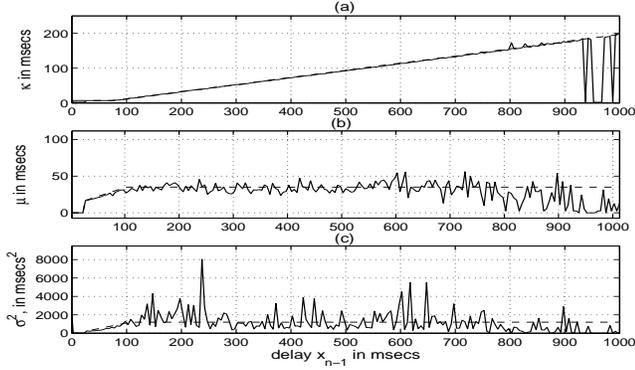


Figure 2: Parameters κ , μ , and σ^2 , for the Γ approximations of the conditional distributions $p(x_n|x_{n-1})$, as a function of delay x_{n-1} . The dotted traces are approximated parameter values used in Sec. 5.

We can find the probability of a *packet arrival outcome*, a particular combination of on-time and late-arriving packets, by summing over the approximated joint distribution. So, for instance, if we'd like to find the probability that packets sent at t_0 , t_1 and t_2 arrive by their deadlines t_{d_0} , t_{d_1} , and t_{d_2} , we evaluate

$$\Pr\{X_0 < t_{d_0} - t_0, X_1 < t_{d_1} - t_1, X_2 < t_{d_2} - t_2\} = \sum_{x_2=0}^{t_{d_2}-t_2} \left(\sum_{x_1=0}^{t_{d_1}-t_1} \left(\sum_{x_0=0}^{t_{d_0}-t_0} p(x_0)p(x_1|x_0) \right) p(x_2|x_1) \right) \quad (1)$$

The parentheses around the inner sums in (1) are to emphasize the interpretation of the channel as making state transitions, where the state of the channel at a given time is the delay for packets sent at that time. In this interpretation the conditional pmfs $p(x_n|x_{n-1})$, serve as the transition probabilities between pairs of states.

Without feedback, we only know that the channel at time t_0 is in some delay-state with distribution $p_X(x)$, the marginal. We find the delay-state distribution for future time-steps with the help of the conditional distributions that serve as the state transition probabilities. Because some delay-states fall outside the region specified by the left side of (1), we don't propagate the probability mass corresponding to entering those states forward in the chain. Thus after each sum we end up with a distribution over the delay-states of the channel at a particular time such that the delay-states of previous time-steps fell in the into the desired range.

Feedback in the form of an acknowledgement can specify the actual forward delay-state in the recent past. A probability distribution can be propagated forward in time from that known state. Suppose, for instance, that through feedback we know that the delay at transmission time-step 0 is d_{A_0} , and we would like to know the probability that packets transmitted at times 1 and 2 arrive by their deadlines:

$$\Pr\{X_1 < t_{d_1} - t_1, X_2 < t_{d_2} - t_2 | X_0 = t_{A_0}\} = \sum_{x_2=0}^{t_{d_2}-t_2} \sum_{x_1=0}^{t_{d_1}-t_1} p(x_1|d_{A_0})p(x_2|x_1)$$

Similarly we can incorporate negative acknowledgements. For instance if feedback shows that the delay for a packet sent at

$t_0 > d_{N_0}$, we would use the conditional distribution for that time step:

$$p_{X_0}(x_0) = \begin{cases} 0 & \text{if } x_0 \leq d_{N_0} \\ \frac{p(x_0)}{1 - \sum_{x=0}^{d_{N_0}} p(x)} & \text{otherwise} \end{cases}$$

Finally, feedback may specify that a packet has been acknowledged and a later packet is negatively acknowledged, for instance $x_0 = d_{A_0}$ and $x_1 > d_{N_1}$. In this case the distribution for X_1 would be:

$$p_{X_1}(x_1) = \begin{cases} 0 & \text{if } x_1 \leq d_{N_1} \\ \frac{p(x_1|d_{A_0})}{1 - \sum_{x'=0}^{d_{N_0}} p(x'|d_{A_0})} & \text{otherwise} \end{cases}$$

5. PERFORMANCE RESULTS FOR THE PROPOSED MODEL

This section contains a comparison of the performance of our channel model with the performance of the model in [1] that assumes that packet delays are drawn independently from a shifted Γ distribution with parameters that are estimated from feedback. To test the models, we examined how accurately they would predict packet arrival outcomes if loss and delay were determined according to our measured channel trace.

Our collected channel trace contains approximately 5 hours of one-way delay measurements taken in the forward and reverse directions from a server to a client at intervals of $\Delta = 50$ ms. In our testing, we examined, for each 50 ms time step in the trace, the packet arrival outcome probabilities that the models would have generated given the feedback information that would have been available at the server at that time step. For feedback, we assume that the client sends a feedback packet every 50 ms reporting on the delays of the most recent arrivals from the server. We further assume that the server sends at least one packet every 50 ms that can be reported on.

We evaluated what predictions the two channel models would have generated at each 50 ms time-step in the trace. The predictions that our models produce are estimates \hat{p} for the probability that a particular arrival outcome will occur. Packet arrival outcomes are Bernoulli trials that occur when some set of packets arrive and another set does not. We chose to examine the models' abilities to estimate the probability of an outcome in which the three packets from one set, set \mathcal{A} , arrive by a deadline and another packet, \mathcal{U} , does not. For each 50 ms time step in the trace, we evaluated what \hat{p} the models would have produced for the 66 transmission schedules such that the \mathcal{A} packets are sent in three successive slots with starting slot ranging from from $t_{now} - 10\Delta$ to $t_{now} + \Delta$, and the \mathcal{U} packet is sent in a slot that is after the \mathcal{A} but before $t_{deadline} = t_{now} + 210$ ms. t_{now} is the position in the trace at which the models are making their estimates.

For each evaluation experiment, we find what probability estimates \hat{p} the model would have produced at a time position in the trace. By looking ahead in the trace we can determine whether the Bernoulli arrival outcomes would actually have been 'success' or not. By looking at the approximately 27 million Bernoulli trials that we evaluated, we can judge the accuracy of the models' estimates \hat{p} by comparing them to the relative frequency of successful outcomes given the estimate. For instance, if a model produces estimate $\hat{p} = 0.2$ for 100,000 different trials, we would say that the model is accurate if there were actually 20,000 successful outcomes in the 100,000 trials.

Because the estimates \hat{p} are real-valued, however, there may be many \hat{p} values that only appear once in all of the trials and thus there would be no relative frequency to compare estimates to. We therefore quantized the estimates \hat{p} to the nearest bin $b = \{0.000, 0.001, \dots, 0.999, 1.000\}$. Over all of the approximately 27 million trials, we then counted, for each bin, the number times N_i that \hat{p} quantized to bin b_i , and the number of times m_i that the outcome was actually ‘success’ for those N_i trials. We could then calculate, for each b_i , the distance between quantized probability estimates b_i and the relative frequency of successful outcomes, m_i/N_i .

We use a distance metric to combine the distances between the b_i and their relative frequencies into one weighted distance to serve as a performance measure for channel models. Let our distance metric be

$$d = \sum_i c_i \left| b_i - \frac{m_i}{N_i} \right|$$

which is a weighted sum of the L_1 distances between the bins and the relative frequencies of the outcomes when model-estimated \hat{p} quantizes to a certain bin. If the model-estimated \hat{p} is correct, m_i , the number of successful outcomes out of N_i Bernoulli trials is a Binomial random variable with variance $N_i b_i (1 - b_i)$. The variance of the relative frequency (the random variable $\frac{m_i}{N_i}$) is therefore $\frac{b_i(1-b_i)}{N_i}$. The tighter the variance of the relative frequency for a particular bin, the more certain we are that the distance $\left| b_i - \frac{m_i}{N_i} \right|$ is due to inaccuracy of the quantized estimates b_i and not due to the variance of the relative frequency. We therefore set weights w_i to be inversely proportional to the standard deviation of the relative frequencies.

$$w_i = \sqrt{\frac{N_i}{p_i(1-p_i)}}, \quad c_i = \frac{w_i}{\sum w_i}$$

Fig. 3 plots relative frequencies versus quantized probability estimates for the trials described above. The top plot is for the model described in [1], in which packet delays and losses are assumed to be independent and shifted Γ -distributed. The parameters μ and σ^2 of the Γ distribution are estimated from the previous 20 feedback packets. κ is set to the minimum forward-trip time of 35 ms. Feedback information is used to condition the distribution as described in [1]. The second plot is for our scheme in the case where the model uses histograms for the conditional pmfs $p(x_n | x_{n-1})$ that are compiled from the same trace over which the evaluation is done. The third plot is for our proposed scheme when the conditional pmfs are approximated by Γ distributions with parameters as shown by the dotted lines in Fig. 2.

Fig. 3 shows qualitatively that relative frequencies of outcomes more closely match estimated outcome probabilities for our proposed model as compared to the i.i.d. Γ model. For the quantitative distance metric, d , defined in (5), we get $d = 0.111, .015$, and $.033$, respectively for the i.i.d. model, for our model when it uses compiled histograms for the pmfs, and for our model when the conditional pmfs are approximated by Γ distributions with parameters as shown by the dotted lines in Fig. 2. By this metric, our model, when using approximated conditional pmfs, chooses estimates \hat{p} such that distances to actual relative frequencies are 3.4 times smaller than those of the common i.i.d. Γ model.

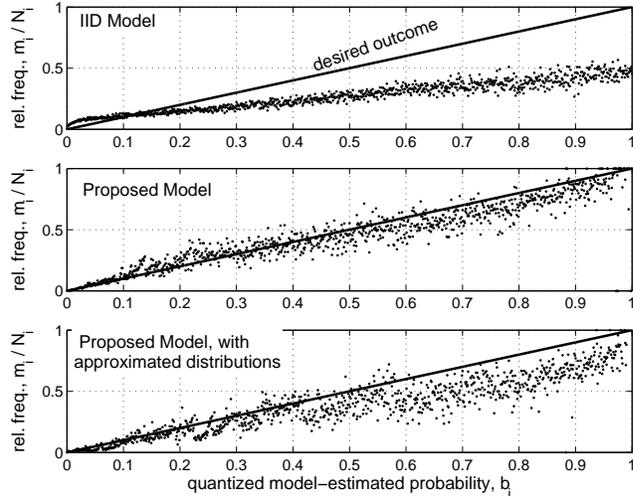


Figure 3: Relative frequency versus model-determined probability estimates for the i.i.d. Γ model (top), for our model using empirically compiled conditional pmfs (middle), and using Γ -approximated conditional pmfs (bottom)

6. CONCLUSIONS

We have presented a new delay and loss model for groups of packets successively transmitted over the Internet. In this paper we have also outlined a delay trace collection method that uses only commonly available Linux hosts, and have found that, on Internet links with a cable-modem hop, shifted Γ -distributions can be good models for the packet delay conditioned on the delay of a previously sent packet. In our experimental results, we have shown, using a quality metric that we have introduced and a cable-modem delay trace that we have collected, that our model can evaluate the probabilities of packet arrival outcomes 3.4 time more accurately than the i.i.d. Γ model commonly used in rate-distortion optimized media streaming.

7. REFERENCES

- [1] Philip A. Chou and Zhouong Miao, “Rate-distortion optimized streaming of packetized media,” Tech. Rep. MSR-TR-2001-35, Microsoft Research, February 2001, (also submitted to IEEE Transactions on Multimedia).
- [2] G. Almes, S. Kalidindi, and M. Zekauskas, “A One-way Delay Metric for IPPM,” RFC 2679, Sept. 1999.
- [3] W. Jiang and H. Schulzrinne, “Qos measurement of internet real-time multimedia services,” in *Proceedings NOSSDAV 2000*, June 2000.
- [4] Sue Moon, Paul Skelley, and Don Towsley, “Estimation and removal of clock skew from network delay measurements,” in *IEEE INFOCOM*, New York, NY, March 1999.
- [5] Jacob Chakareski and Bernd Girod, “Server diversity in rate-distortion optimized media streaming,” in *IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.
- [6] J. C. Bolot, “Characterizing end-to-end packet delay and loss in the internet,” *Journal of High Speed Networks*, vol. 2, no. 3, pp. 289–298, September 1993.