

# On the TCP-Friendliness of VoIP Traffic

Tian Bu<sup>a</sup>, Yong Liu<sup>b</sup> and Don Towsley<sup>c</sup>

<sup>a</sup> Bell Labs  
Lucent Technologies  
Holmdel, NJ 07733  
tbu@bell-labs.com

<sup>b</sup>Electrical & Computer Engineering  
Polytechnic University  
Brooklyn, NY 11201  
yongliu@poly.edu

<sup>c</sup>Computer Science  
University of Massachusetts  
Amherst, MA 01003  
towsley@cs.umass.edu

**Abstract**—Recently, some concerns have been raised about whether the increase of UDP traffic due to Skype type of applications will throttle down regular TCP users. This concern comes from the fact that UDP connections are unresponsive to congestion, i.e., UDP senders do not reduce their sending rates when congestion occurs. On the other hand, most UDP based real time applications, such as Voice-over-IP (VoIP) applications, will drop out completely if the user perceived quality becomes unacceptable as the network congestion increases. This user back-off mechanism effectively reduces UDP traffic rate when the network is congested. In this paper, we investigate the fairness issue between VoIP flows and TCP flows under different network environment. Our study explicitly takes into account both transport layer congestion control mechanism built-in TCP protocol stack and the self-adaptiveness of VoIP users. Various models are developed to characterize VoIP user back-off behaviors in response to call quality degradation resulted from packet loss inside the network. The comparison between VoIP user back-off and TCP rate adaption shows that not only VoIP flows consume much less bandwidth than TCP flows, but also VoIP traffic is indeed very responsive to congestion when the network is overloaded.

## I. INTRODUCTION

Voice-over-IP (VoIP) has recently become one of the hottest applications in the Internet. One single VoIP software, Skype [1], has been downloaded more than 120,000,000 times by users all around the world to call anywhere for free. New VoIP applications pose challenges to traditional telephone companies. The large population of VoIP users also brings up various concerns to the network research community. One major concern is whether VoIP traffic will bring down the Internet. The reasoning behind this is that most VoIP applications are based on UDP connections, which don't react to congestion inside the network. The stability of the current Internet is largely maintained by TCP, specifically, its built-in transport layer congestion control mechanism. It has been urged by network researchers that UDP based applications should implement *TCP-friendly* congestion control to avoid causing *congestion*

*collapse* [2]. Based on a recent measurement study [3], major VoIP applications, such as Skype, still don't adapt their sending rates to congestion. With the increase in VoIP users, it appears that we are facing an increasing danger of congestion collapse. However, the previous argument misses one major factor in VoIP applications – user back-off. A regular VoIP call involves at least two people talking to each other. If any one of them becomes dissatisfied with the call quality, the call will be dropped. In other words, the number of VoIP calls will decrease as the network becomes congested. Call drops can be treated as a user level congestion control mechanism. With this in the loop, we want to revisit the question: “*Is VoIP traffic TCP-Friendly?*”

To answer this question, we need to compare how TCP and VoIP users react to congestion. There is a huge literature on TCP congestion control behavior. Various models have been established to characterize TCP throughput as a function of packet delay and packet loss [4], [5]. Little attention has been paid to the role of user back-off in network congestion control. In this paper, we study the interaction between VoIP user back-off and TCP rate adaption. We present models to evaluate how network resource, specifically bandwidth, is shared among VoIP flows, long-lived and short-lived TCP flows. We quantify VoIP user back-off effect by approximating call drop probability as a function of network loss and delay using subject test results published by ITU [6]. We then demonstrate that, under random loss in the absence of FEC, VoIP flows are fairly friendly to TCP flows in the sense that the total bandwidth utilized by VoIP flows drops faster than that of TCP flows. When either general FEC or media dependent FEC are used, the aggregate VoIP throughput drops slower than the aggregate TCP throughput. However, even when FEC is used, VoIP flows are more vulnerable than TCP flows under bursty loss and transient loss triggered by network failures, where the call quality drops much quicker than under random loss. In addition, VoIP flows, especially low bit rate ones, use much less bandwidth than TCP flows. Therefore, we believe the prevalent of VoIP flows

will not cause a congestion collapse problem even though they do not have a built-in congestion control mechanism.

In a very related work [7], Morris and Tay studied how TCP user behavior helps to maintain the network stability. It was argued that TCP alone cannot account for the Internet's stability. TCP user aborts enable a network sustain a higher demand without causing congestion collapse. There have been works on evaluating the VoIP quality based on network conditions [8] but without considering call drops due to dissatisfying call quality. Tao et al. propose in [9] to improve the VoIP quality through path switching. Such a scheme would help to reduce congestion when it occurs since VoIP flows may move away from a congested link when a call quality degradation is detected. Markopoulou et al [10] assess the VoIP quality over the Internet based on the traces collected from backbone. They conclude that many paths in the Internet provide poor call quality. Their results may be different if the use of FEC is considered.

We start with network loss models which will be used to evaluate the interaction between TCP and VoIP users. In Section III, we characterize how TCP and UDP users adapt to congestion differently. For TCP users, we present their throughput as a function of packet delay and packet loss probability. For UDP users, we introduce the concept of *expected* sending rate to factor in user back-off as a congestion control scheme. VoIP user back-off is determined by the voice call's quality. In Section IV, we develop models to quantify the quality of different VoIP codecs under different network loss environment. Section V presents numerical results about TCP and VoIP interaction. It will become clear that VoIP users are indeed TCP-Friendly. Section VI provides more discussions on the importance of user behavior in studying network resource sharing and congestion control. We conclude the paper in Section VII.

## II. NETWORK LOSS MODELS

The quality perceived by network users is determined by two factors: packet loss and packet delay. To study user behavior in a network environment, we develop models that abstract out packet loss and delay characteristics. In this paper, we will focus on the interaction between TCP flows and VoIP type of UDP flows across Wide Area Networks (WAN). It has been studied in [4] that the TCP sending rate is determined by the end-to-end packet delay and end-to-end packet loss probability. A recent study, [9], has shown that packet losses have more significant impact on voice quality than end-to-end packet delay. In a well-engineered WAN, it is reasonable to assume that the end-to-end packet delay will be

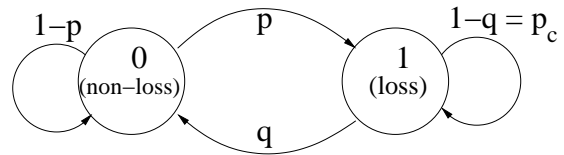


Fig. 1. The Gilbert Model

dominated by the propagation delay from the source node to the destination node. Therefore, to simplify our analysis, we ignore the packet delay variability and focus on network user's adaption to different types of packet losses. We first present three network loss models. To illustrate how bandwidth is shared between types of users, we also introduce a simplified single bottleneck link model, where the packet loss probability is a function of link utilization.

### A. General Network Loss Models

In a general network setting, we classify end-to-end packet losses incurred by users in three different categories: random packet loss; bursty loss due to bursty traffic; and transient loss due to the link/router failures.

1) *Random Loss*: The simplest way to model packet loss is to assume each packet gets dropped inside the network randomly with some probability  $p$ . Most proposed Active Queue Management (AQM) schemes [11], [12] drop packets randomly. When link utilization is low and the traffic is not very bursty, the packet losses from a drop-tail link tend to be independent.

2) *Bursty Loss*: Packet losses on a drop-tail link can be bursty if the traffic it carries is bursty [13]. Some measurement studies have shown that the Internet traffic can present burstiness over a wide range of time-scales [14], [15], [16]. The Gilbert model is most commonly used to describe bursty losses often experienced by end users in the Internet [17]. As shown in Figure 1, a Gilbert loss model has two parameters, unconditional and conditional loss probability, denoted as  $p_u$  and  $p_c$ . The conditional loss probability  $p_c$  captures loss burstiness: the larger  $p_c$ , the more bursty loss is.  $p$  and  $q$  are transition probabilities between non-loss and loss state. They can be computed as:

$$\begin{aligned} q &= 1 - p_c \\ p &= \frac{p_u(1 - p_c)}{1 - p_u} \end{aligned}$$

3) *transient Loss*: Another type of packet losses cause quality degradation is packet losses resulted from transient network failures, such as link failures, router linecard failures, etc [18]. We further assume when

a failure happens, all of the packets traversing the failed equipment are lost until the equipment gets fixed/replaced or packets get re-routed. Since link failures and routing changes may have many different independent causes, we assume network failures happen according to a Poisson process with rate  $\lambda_F$  and that their durations follow some general distribution  $G_F$  with mean  $\mu_F$ .

### B. Single bottleneck: $M/M/1/K$ Queue

An Internet core router takes traffic from a large number of independent TCP and UDP sources. As suggested by [19], the packet arrival can be modeled as a Poisson process the link multiplexing degree is high. There are two major types of queue management schemes, the drop-tail and the AQMs. Since most operational routers employ the simple drop-tail scheme, we model the bottleneck link with finite buffer as a  $M/M/1/K$  queue. More specifically, we model the router  $v$  of capacity  $C_v$  as a drop tail queue with a total buffer size of  $B_v$ . Given the traffic arrival rate  $X_v$  at router  $v$ , the average packet loss probability  $p_v$  and queuing delay  $d_v$  can be approximately calculated from either the  $M/M/1/K$  model

$$p_v = \frac{1 - X_v/C_v}{1 - (X_v/C_v)^{B_v+1}} \left(\frac{X_v}{C_v}\right)^{B_v} \quad (1)$$

$$d_v = \frac{1 - X_v/C_v}{1 - (X_v/C_v)^{B_v+1}} \sum_{i=0}^{B_v} i \left(\frac{X_v}{C_v}\right)^i \quad (2)$$

or the  $M/D/1/K$  model [20]. In either case, the packet drop probability and the queuing delay at a router  $v$  can be obtained as functions of the traffic arrival rate and the link capacity, e.g.,

$$(p_v, d_v) = DT(X_v, C_v) \quad (3)$$

Note that, for a  $M/M/1/K$  queue, we *cannot* assume that two packets arriving to the queue back-to-back will get dropped independently. However, we are only interested in characterizing the packet loss process experienced by packets from the same flow. Given that we are looking at a bottleneck link with high degree multiplexing and negligible queuing delay, packets from one flow will be interleaved by packets from many other flows sharing the same bottleneck link. The packet losses experienced by two adjacent packets from the same flow tends to be independent. From this point on, we assume each flow experiences *random loss* on a  $M/M/1/K$  type of bottleneck link.

## III. NETWORK USER CONGESTION ADAPTION

In this section, we compare how different type of network users adapt to congestion. TCP users react to congestion by adjusting their sending rates according to packet losses incurred inside the network. Unlike TCP flows, almost no UDP flow has built-in transport layer mechanism to adapt their sending rates to congestion. However, depending on upper layer design and user behavior, some UDP flows may reduce their rates for persistent loss or even drop out completely under severe packet loss. We present these models to characterize congestion adaption of long-lived TCP flows, short-lived TCP flows and general UDP flows.

### A. Long-lived TCP Flows

We define a long-lived TCP flow to be a TCP flow whose congestion control phase is significant longer than its slow start phase such that the impact of its slow start phase is negligible. According to [4], for a long-lived TCP flow  $l$  with end-to-end loss probability  $p_l$  and queuing delay time  $d_l$ , the TCP sending rate can be characterized by

$$S_l^T(p_l, d_l) = \frac{M\left(\frac{1-p_l}{p_l} + \frac{W(p_l)}{2} + Q(p_l)\right)}{(R_0 + d_l)\left(\frac{b}{2}W(p_l) + 1\right) + \frac{Q(p_l)F(p_l)T_0}{1-p_l}}, \quad (4)$$

where  $M$  is packet size measured in bits and  $R_0$  is TCP user's two-way propagation delay, and

$$\begin{aligned} W(p) &= 2/3 + 2\sqrt{(1-p)/(3p) + 1/9}, \\ Q(p) &= \min\{1, (1 - (1-p)^3)(1 + (1-p)^3 \\ &\quad (1 - (1-p)^{p-3})) / (1 - (1-p)^{W(p)})\}, \\ F(p) &= 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6. \end{aligned}$$

### B. Short-lived TCP Flows

We define a short-lived TCP flow to be a TCP flow that either does not go into a congestion avoidance phase or its congestion avoidance phase is relatively short. Besides the end-to-end loss  $p^S$  and delay  $d^S$ , the throughput of a short-lived TCP flow  $s$  also depends on its session size  $L_s^S$ , i.e., number of packets in the session. According to [5], the expected duration  $T_s^S$  of a short-lived TCP flow can be characterized by

$$\begin{aligned} E[T_s^S | L_s^S] &= E[D_{ss} | L_s^S] + E[D_{loss} | L_s^S] \\ &\quad + E[D_{ca} | L_s^S] + E[D_{delack} | L_s^S], \end{aligned}$$

where  $E[D_{ss} | L_s^S]$  is the expected duration of the slow start phase obtained from expression (15) in [5];

$E[D_{loss}|L_s^S]$  is the expected delay due to retransmission timeout or fast recovery that happens at the end of the initial slow start phase and is obtained from expression (20) in [5];  $E[D_{ca}|L_s^S]$  is the expected time required to send the remaining data after slow start and loss recovery obtained from expression (24) in [5];  $E[D_{delay}|L_s^S]$  is the expected delay due to delayed acknowledgments and is determined by implementation details. The throughput of  $s$  can be obtained from

$$S_s(p_s, d_s, L_s^S) = L_s^S / E[T_s^S | L_s^S] \quad (5)$$

### C. UDP Flows

Unlike TCP flows, most UDP flows do not adjust their sending rates in response to packet losses. However, depending on upper layer design and user behavior, some UDP flows may reduce their rates in the presence of persistent loss or even drop out completely under severe packet loss. For instance, a VoIP application may change from high bit-rate codec to low bit-rate codec when the lack of available bandwidth is detected through extensive losses such that the call can carry on but at a lower quality. As loss increases, the probability that a user will drop a VoIP call increases while the codec and the playout scheme remain unchanged. However, the call drop probability varies among different codecs, users, and playout schemes. In general, the aggregate sending rate of many UDP flows is a function of the quality metrics, the application behavior, and the user behaviors.

The quality of an application carried by a UDP flow is usually determined by the end-to-end loss rate and/or packet delay. For instance, the E-Model [21] relates the quality of a VoIP call to packet loss and the round trip time of the underlying UDP flow. In this paper, the “quality” of an UDP flow class  $i$  refers to the quality perceived by the application above the flow that can be quantified as

$$Q_i = Q_i(p_i^U, d_i^U), \quad (6)$$

where  $p_i^U$  and  $d_i^U$  are respectively the end-to-end packet loss probability and packet delay experienced by a class  $i$  flow. In Section IV, we’ll describe the  $Q(\cdot)$  functions for VoIP flows in detail.

Different users will behave differently for the same application quality. When the quality becomes unacceptable, some UDP flows will react by reducing their sending rates. This could correspond to switching to a lower bit-rate codec in the case of VoIP. Although the quality delivered by low bit rate codec is lower, it can still be better than the quality before reducing the rate. Therefore, the sending rate of a rate adaptive UDP flow

$a$  is determined by the qualities at different sending rate, i.e.,

$$S_a(p_a, d_a) = S_a(Q_a^1(p_a, d_a), \dots, Q_a^n(p_a, d_a)), \quad (7)$$

where  $Q_a^i$  is the quality function when UDP flow  $a$  uses its  $i$ -th codec.

More commonly, when the perceived quality lies below a threshold, some UDP users may drop the session completely. The fraction of class  $i$  users that will drop the session can be expressed as

$$A_i(p_i^U, d_i^U) = A_i(Q_i(p_i^U, d_i^U)) \quad (8)$$

In general,  $A_i$  is a non-increasing function of the flow quality. We will focus on this call drop probability for VoIP application in greater detail in Section IV. Call drops effectively reduce the aggregate traffic rate generated by the UDP flows. To account for this traffic adaption in the form of call drop, we define the *expected* sending rate of a VoIP flow as,

$$\bar{S}_i^U = S_i^U (1 - A_i(p_i^U, d_i^U)), \quad (9)$$

the product of its original sending rate and the probability that the call is not dropped under a given network condition.

## IV. VOIP USER ADAPTION

We focus now on VoIP flows and model how they react to network congestion. In this section, we first describe a model that relates the network delay and loss to the quality of a VoIP call. We formulate the models for regular VoIP flows and VoIP flows using either media-independent FEC or media-dependant FEC. Based on subjective test [6] published by ITU for users’ opinions of the quality of G.729 under various loss conditions, we then show how the users may drop VoIP calls at different call quality. Last, we show how the transient losses may increase VoIP call drop probability.

### A. VoIP Call Quality

Packet loss and delay affect the perceived quality of a VoIP application. Although VoIP packets can be dropped or delayed either in the network or by the application, we focus on the loss and delay taking place inside the network since our goal is to understand how VoIP traffic and TCP traffic share the network bandwidth when there is congestion related delay and loss.

Perceived voice quality is typically measured by the Mean Opinion Score (MOS), a subjective quality score that ranges from 1 (unacceptable) to 5 (excellent) as defined in [22]. MOS values are obtained by carrying

out subjective tests where a group of testers are asked to rate the same speech sample and the MOS is obtained average over all testers. Table I shows the mapping of MOS to user satisfaction as reported in [21], [23]. A MOS score above 4.0 matches the level of quality in the current PSTN. A MOS score in the range between 3.6 and 4.0 corresponds to a medium quality level, where some users are not satisfied. A MOS score in the range between 3.1 and 3.6 corresponds to a low level of quality, where many users are dissatisfied. At MOS score under 3.1, nearly all users are dissatisfied. Some objective models such as PESQ [24], can also generate quality scores close to the subjective results by comparing the impaired voice signal with its original version. However, none of these methods can be used for evaluating VoIP quality to account for network delay and network loss.

The E-Model defined in ITU-T G.107 [21] is a better choice for assessing VoIP quality from the network behavior. It combines all of the impairments into one value  $R$ .

$$R = R_0 - I_s - I_e - I_d + A \quad (10)$$

where  $R_0$  groups the effects of various noise;  $I_s$  includes the effect of other impairments that occur simultaneously with the voice signal;  $I_e$  summarizes the impairments caused by different types of losses;  $I_d$  represents the impairments caused by path delay; Lastly,  $A$  compensates for the above impairments under various user conditions. The R-factor ranges from 0 to 100 and is related to MOS through a non-linear mapping [21];

$$MOS = 1 + 0.035R + 7 \times 10^{-6} R(R-60)(100-R) \quad (11)$$

Among all of the factors in Equation (10), only  $I_d$  and  $I_e$  is affected by network condition. Taking the default values for all other factors defined in [25], the model reduces to

$$R = 93.2 - I_e - I_d, \quad (12)$$

where  $I_e$  accounts for impairments caused by both encoding and transmission losses. For a given codec, encoding loss is predetermined. However, due to the use of PLC and the different levels of loss tolerance by different codecs, the mapping from loss to  $I_e$  is not straightforward. No analytical formulas are available for  $I_e$ . Instead, the  $I_e$  must be obtained from subjective test of voice quality for each particular codec and various loss conditions. Figure 2 plots  $I_e$  values as a function of loss rate for G.711 with PLC and G.729-A + VAD that are provided in Appendix I. Following [8], we perform a curve fitting on these values to approximate  $I_e$  as a function of losses for these codec, i.e.,

$$I_e = \gamma_1 + \gamma_2 \ln(1 + \gamma_3 p) \quad (13)$$

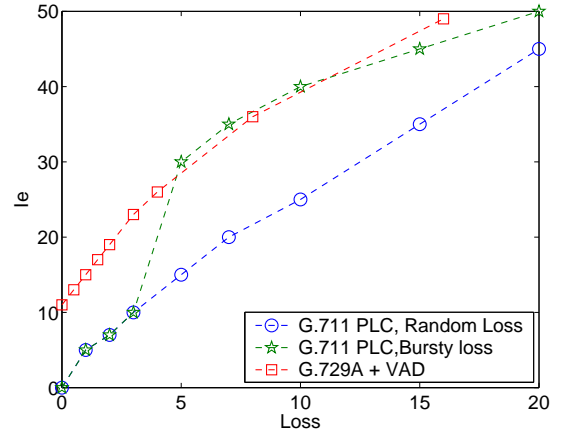


Fig. 2.  $I_e$  for G.711 with PLC and G.729a + VAD

where the values of  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  for different codecs are obtained from the curve fitting of Figure 2. In Figure 2, we plot  $I_e$  for both random loss and bursty loss for G.711. Unfortunately, so far there is no subjective test results for other codecs under bursty loss. We see that for low loss rates, i.e.,  $p < 0.04$  the  $I_e$  values under the conditions of random loss and bursty loss are same. However, for larger loss rates, they differ dramatically. Bursty loss causes a significant degradation in voice quality over random loss.

G.107 [21] provides an analytical model for computing  $I_d$ . A simplified version of the function is given in [8]

$$I_d = 0.024d + 0.11(d - 177.3)\mathcal{I}(d - 177.3) \quad (14)$$

where  $\mathcal{I}(x)$  takes value 0 if  $x < 0$  otherwise 1. The delay  $d$  includes network delay, codec related delay, and playout buffering delay. Again, we are only interested in network delay and take default values for other delays.

To summarize, the E-model relates the quality of a VoIP call to the network delay and loss. Thus we can use the function to compute  $R$  values as the  $Q^{udp}$  as defined in Section III. Now we investigate how FEC may impact the VoIP quality.

### B. VoIP Quality with FEC

To recover lost audio frames in VoIP, the most common scheme is FEC. There are two major types of FEC, media independent FEC [26] and media dependent FEC [27], [28]. Media dependent FEC is also referred to as LBR (Low Bit Redundancy) FEC. These FEC schemes differ as to whether or not they make use of audio specific information. Loss can also be handled by repeating the waveform in the last received audio frame. This is called packet loss concealment (PLC) [28], which

can improve perceived quality with no extra bandwidth overhead.

Media independent FEC recovers lost packets exactly. An example is the Reed-Solomon (RS) code. Its notation is  $(n, k)$ , where  $n$  and  $k$  are the number of total units and data only units respectively. An  $(n, k)$  code can recover all losses iff at least  $k$  out of  $n$  packets are received. Under a random loss rate  $p$ , the unrecovered loss probability after FEC for an  $(n, k)$  code is [29]

$$p_{fec} = p \left( 1 - \sum_{i=k}^{n-1} \binom{n-1}{i} (1-p)^i p^{n-i-1} \right) \quad (15)$$

The unrecovered loss probability after FEC under a bursty loss is also derived for different bursty loss model. For the Gilbert bursty loss model shown in Figure 1, the unrecovered loss probability for a  $(3, 2)$  code that account for piggybacking is derived in [30] as

$$p_{fec} = p_u(p_c + pq/2) + p(1 - p_u)p_c/2 \quad (16)$$

In this paper, we will assume a  $(3, 2)$  code is used. Thus the final loss probability after FEC can be obtained from (16) since the piggybacking is widely used. The loss impairment  $I_e$  can be calculated by replacing  $p$  in (13) with  $p_{fec}$ .

Unlike media independent FEC that does bit-exact recovery, the media dependent FEC includes a redundant but lower quality encoding of the same audio in one or more subsequent packets. If the original audio is lost but the redundant version is not, the audio is recovered at a lower quality. The redundant version usually is a recoding of the original one at a lower rate.

In this paper, we will use an example where the original audio is coded with  $G.711$  and the redundant version is coded with  $G.729a$ . We assume the redundant packet is sent right after the original packet. To model bursty loss, we use the Gilbert loss model. The probability that the original packet is lost and the redundant packet is received is  $p_u(1 - p_c)$ ; and the probability that both packets are lost is  $p_u p_c$ . Denote by  $R_{711}(p)$  the quality of the  $G.711$  coding as a function of random loss probability; and  $R_{729a}$  the quality of the  $G.729a$  coding without packet loss. The overall quality perceived by the receiver with this type of media dependent FEC can be calculated as

$$R_{md}(p_u, p_c) = (1 - p_u + p_u p_c) R_{711} \left( \frac{p_u p_c}{1 - p_u + p_u p_c} \right) + p_u (1 - p_c) R_{729a},$$

where  $p_u p_c / (1 - p_u + p_u p_c)$  is the effective packet loss rate seen by the original  $G.711$  coding.

MOS Lower bound	R (emodel) Lower bound	User satisfaction
90	4.34	Very satisfied
80	4.03	Satisfied
70	3.60	Some users dissatisfied
60	3.10	Many user dissatisfied
50	2.58	Nearly all users dissatisfied
0	1	Not recommended

TABLE I  
RELATION BETWEEN MOS, R-VALUE, AND USER SATISFACTION

### C. Call Drops due to Quality Degradation

The MOS score measures the user perceived call quality. As shown in Table I, the user satisfaction drops at low MOS scores. Consequently, more and more users will drop calls as the MOS score decreases. Eventually all users will drop out at very low MOS scores. However, a MOS score only summarizes the opinions of users participating in the test but does not identify the perceived quality of each individual user, which determines whether the user will drop a call or not. In order to find the distribution of user perceived quality for different MOS scores, we make use of some publicly available subjective test results where both MOS and each user's rating are recorded. Since so far we are not aware of any data that relates the user call drop probability to call quality, we have to approximate the distribution of call drops by users' opinions. Since nearly all users are dissatisfied when the MOS score drops under 3.1 according to Table I, we assume that a user would drop a call if he/she rates a call equal to or less than 3<sup>1</sup>.

A set of subjective tests [6] were conducted to characterize  $G.729a$  by ITU-T study group 12, the speech quality experts group. In a test, each of the 24 testers listened to the same audio and give an integer rating between 1 and 5. The MOS score of the audio clip is obtained by averaging over all 24 user ratings. The purpose of experiment 3 in [6] is to evaluate the quality of  $G.729a$  under different network conditions. There are a total of 250 tests for both random and bursty losses at different rates. We plot the results of all 250 test in Figure 3.

By performing a curve fitting of Figure 3, we obtain a function that relates the call drop rates to MOS scores. Since the  $R$  value of a call can be computed from the network loss and queueing delay and there is one to one between  $R$  value and MOS score, we can compute the call drop rate for a given network loss and delay.

<sup>1</sup>In a subjective test, the tester only give integer ratings

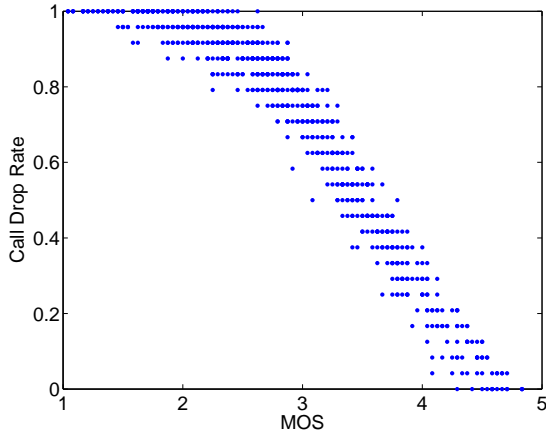


Fig. 3. Call drop rate for different MOS

Although the test is for *G.729a* codec, we believe the function of call quality and network condition should also apply to other codec because this is obtained from subjective test where tester only make decision on call quality without knowing the types of codecs.

*D. Call Drops due to Transient Network Failures*

A transient failure may occur in a network due to many reasons such as a link outage. It will take anywhere between several seconds to tens of minutes to reroute the traffic [18]. Although transient network failures are bad for all network flows, a VoIP flow is especially vulnerable. No conversation can be maintained during the failure period; thus the users would have to drop calls. On the other hand, TCP flows may resume their transmission as soon as the failure recovers.

Since network failures tend to occur as independent events, we may assume the network failure arrives according to a Poisson process at rate  $\lambda_F$  and with a mean duration  $\mu_F$ . When a VoIP flow starts, if it sees a network failure, the call cannot even be set up. Therefore, the VoIP call is terminated immediately. This happens with probability  $1 - P_I$ , where  $P_I = e^{-\lambda_F/\mu_F}$  is the probability that no server is busy in a  $M/G/\infty$  system with job arrival rate  $\lambda_F$  and mean job duration  $\mu_F$ . After a VoIP flow starts successfully, it will drop if it sees a network failure before the end of the call. The call drop probability due to transient loss is  $P(X < T)$ , where  $X$  is an exponential random variable representing the failure inter-arrival time;  $T$  is the original call duration with a general probability density function  $g_T(t)$ . If a call is dropped due to transient loss, i.e.  $X < T$ , its actual duration will be  $X$ . Therefore, after factoring in transient packet loss, we can calculate the actual call duration as  $\min(T, X)$ . Assume independence between

Codecs	Media Bit Rate	UDP/RTP Overhead	Overall Bit Rate
<i>G.711</i>	64	16	80
<i>G.711</i> RS(3,2) FEC	96	16	112
<i>G.711</i> Media FEC	72	16	88
<i>G.729a</i>	8	16	24
<i>G.729a</i> RS (3,2) FEC	12	16	28

TABLE II  
VOIP BIT RATE (KBPS)

$T$  and  $X$ , the call drop probability is

$$P(X < T) = 1 - \int_{t=0}^{\infty} e^{-\lambda_F t} g_T(t) dt \quad (17)$$

The average call duration is

$$E[\min(X, T)] = \frac{1}{\lambda_F} - \frac{1}{\lambda_F} \int_{t=0}^{\infty} e^{-\lambda_F t} g_T(t) dt \quad (18)$$

V. INTERACTION BETWEEN TCP USERS AND VOIP USERS

In this section, based on network loss models and user adaption models established in Sections II, Section III and IV, we present numerical results about the interaction between TCP users and VoIP users under different network conditions and when different FECs are used.

Since each VoIP flow does not perform congestion control thus its sending rate is constant<sup>2</sup> during the entire call as long as the call is not dropped. The sending rate of different VoIP codecs are listed in Table II. As discussed in Section III-C, VoIP traffic adapts to network congestion through call drops. In order to evaluate the fairness between a TCP flow and a VoIP flow, we look at VoIP's *expected* sending rate as defined in (9). The expected sending rate represents how much traffic a VoIP call generates on average after factoring in call drops due to call quality degradation. In the rest of the paper, we will refer the *expected* sending rate of a VoIP flow as its sending rate when the context is clear.

We first study the fairness between a TCP flow and a VoIP flow under random loss, where we compare the (expected) sending rate of the TCP flow and the VoIP flow. We then make the loss bursty and compare their sending rates again. We also evaluate how the transient losses impact the fairness. At last, on a single bottleneck link, we show how the bandwidth shares change among TCP and VoIP users when we change their populations and how the VoIP call quality drops as the link utilization increases.

<sup>2</sup>assuming no codec changes

A. Impact of Codec, Loss pattern, and FEC on Fairness

In Figure 4, we plot the throughput of a short-lived TCP flow and various types of VoIP flows relative to that of a long-lived TCP flow as random loss rate increases from 0.5% to 20%. Each point on a relative throughput curve measures TCP-Friendliness of the corresponding VoIP or short-lived TCP flow under certain packet loss probability. The slope of the curve demonstrates how the TCP-Friendliness changes as the network gets more congested. A flat line suggests the corresponding flow adapts its rate relatively the same way as a long-lived TCP flow; whereas up-slope/down-slope suggests a slower/faster flow rate reduction than that of a long-lived TCP flow.

From Figure 4, we observe that all VoIP flows use less bandwidth than either long-lived or short-lived TCP flows. Moreover, a VoIP flow using the low bit rate codec *G.729a* consumes only about 1% of the bandwidth of a long-lived TCP flow when the loss rate is very low, and never more than 3% under all loss rates. A VoIP flow using *G.711* codec reduces its rate more slowly than a long-lived TCP flow when loss increase from 0.5% to 7%. As loss rate increases further, the *G.711* flow reduces its rate faster than a long-lived TCP flow. We also observe the similar transition for *G.729a* flows where the turning point occurs at a loss rate of 4%. Note that the *G.729a* drops more dramatically than *G.711*. To summarize, the VoIP flow without FEC is more aggressive than TCP flows at low loss rate but less aggressive when loss become larger than 7%.

Adding FEC to a VoIP codec changes the TCP friendliness. The *G.711* VoIP with FEC is more aggressive than a VoIP flow when the loss rate is lower than 15% but the gradient drops as loss increases. A flat line for loss rate higher than 15% suggests *G.711* with FEC reduces its rate at the same rate as TCP when network congestion level is high. The *G.729a* VoIP flow with FEC is also more aggressive than the one without FEC and show very similar behavior as *G.711* flow with FEC. We only show the results for media independent FEC and the results for media dependent FEC are similar.

Figure 5 plots the relative throughput when packet loss is bursty. To generate bursty loss, we use the Gilbert model with conditional loss probability  $p_c = 0.3$ . From the figure, we observe that all VoIP flows become less aggressive when loss is bursty. Specifically, *G.711* starts to show the down-slope at lower loss rate than in Figure 4. The call quality improvement due to added FEC is less under bursty loss than under random loss. When the loss rate is higher than 15%, instead of a flat line as in Figure 4, the *G.711* with FEC actually shows a

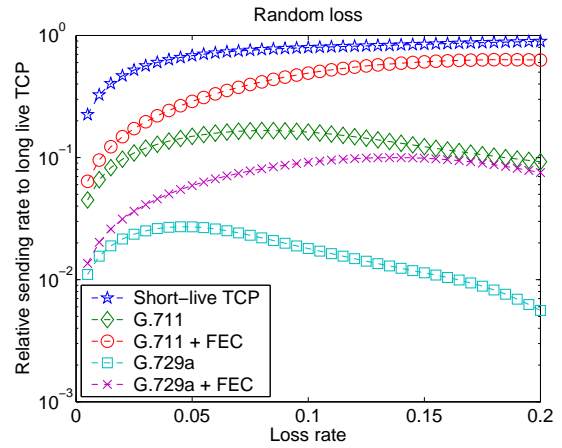


Fig. 4. Random loss: VoIP always have lower rate, back off faster when loss rate is high

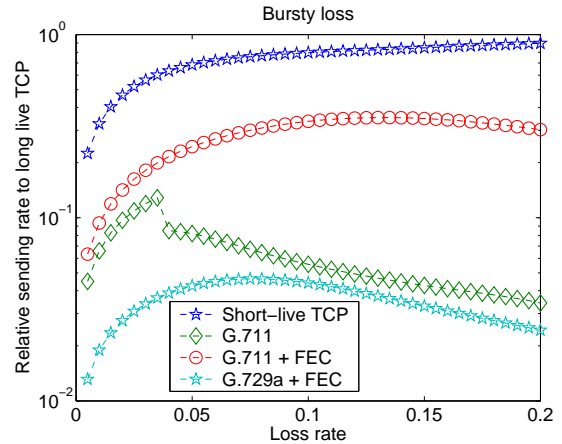


Fig. 5. VoIP is more vulnerable to bursty loss and FEC provides less the quality improvement than in random loss case.

slow down-slope. The down-slope is more dramatic for *G.729a* with FEC when loss is higher than 10%.

B. Single bottleneck Link

In this section, we evaluate how VoIP flows, long-lived and short-lived TCP flows share a single bottleneck. The bottleneck link  $b$  has a capacity of  $C_b$  and finite buffer of size  $K_b$ . Denote by  $p_b$  and  $d_b$  the packet loss probability and packet delay on this link. The traffic traversing this link consists of three components:

- **Long-lived TCP Flows:** There are  $N^T$  classes of long-lived TCP flows and  $M_i^T$  flows in class  $i$ . All flows in the same class share the same route, thus the same propagation delay  $g_i^T$ . Since link  $v$  is the only bottleneck link, the end-to-end packet loss probability for all TCP classes is  $p_b$ , the end-to-end packet delay for TCP class  $i$  is  $d_i^T = g_i^T + d_b$ . The aggregate TCP sending rate from all long-lived



flows is

$$X_T = \sum_{i=1}^{N^T} M_i^T S_i^T (p_b, g_i^T + d_b) \quad (19)$$

- **Short-lived TCP Flows:** There are a total of  $N^S$  classes of short-lived TCP flows. All TCP flows in the Class  $i$  share the same route thus have the same end to end loss  $p_i^S = p_b$  and delay  $d_i^S = g_i^S + d_b$ , where  $g_i^S$  is the one ways propagation delay for class  $i$ . Short-lived TCP flows in Class  $i$  arrives according to a Poisson process at rate  $\lambda_i^S$ ; in average each flow will generate traffic with volume  $L_i^S$ . Then the aggregate short-lived TCP traffic rate is

$$X_S = \sum_{i=1}^{N^S} \lambda_i^S L_i^S. \quad (20)$$

The average throughput of short-lived TCP flows can be calculated as in (5).

- **VoIP Flows:** Assume there are a total of  $N^U$  classes of VoIP flows. The VoIP flows in class  $i$  arrives according to a Poisson process at rate  $\lambda_i^V$ . The average duration is  $T_i^V$ . All flows within Class  $i$  send data at rate  $S_i^V$ . The end-to-end loss rate and the queuing delay of flows in Class  $i$  are  $p_b$  and  $d_i^V = g_i^V + d_b$ , where  $g_i^V$  is the propagation delay for VoIP flows in class  $i$ . According to Little's Law, without user dropping out, the average number of VoIP sessions is  $\sum_{i=1}^{N^U} \lambda_i^V T_i^V$ . With dropping out, the actual average traffic rate generated by all UDP sessions (without considering the traffic generated by one user before his dropping out) is

$$X_V(p_b, d_b) = \sum_{i=1}^{N^U} \lambda_i^V T_i^V S_i^V (1 - A_i^V(Q_i(p_b, d_b))), \quad (21)$$

$S_i^V$  is given except for adaptive rate UDP where the  $S_i^V$  is a function of  $p_b$  and  $d_b$  as shown in (7); voice quality  $Q_i(\cdot)$  and call drop probability  $A_i^V(\cdot)$  can be calculated for different VoIP codecs as described in Section IV.

Therefore, the total traffic rate on link  $b$  is

$$X(p_b, d_b) = X_T(p_b, d_b) + X_S(p_b, d_b) + X_V(p_b, d_b), \quad (22)$$

together with  $M/M/1/K$  equations (1) and (2), we can calculate packet loss probability  $p_b$  and packet delay  $d_b$  on the bottleneck link. Consequently, the bandwidth share for each type of flows can be calculated according to (19), (20) and (21). In our numerical examples, for

the clarity of presentation, we only generate one class for each type of flows.

We first investigate the bandwidth allocation among different types of flows as the number of flows in each type grows proportionally. We have six types of flow share a single  $1Gbps$  link of buffer size 10 packets. The six flow types are long-lived TCP, short-lived TCP,  $G.711$ ,  $G.729a$ ,  $G.711$  with FEC, and  $G.729a$  with FEC. We assume the round trip time is 60ms. The short-lived TCP has a mean session size of 20 packets. The VoIP session has a mean duration of 120 seconds. Initially, we set the number of long-lived TCP flows and the arrival rates of short-lived TCP and VoIP flows such that the bottleneck bandwidth is shared equally among all six types of flows. Specifically, the number of long-lived TCP flows is 200, and the arrival rates of short-lived TCP,  $G.711$ ,  $G.729a$ ,  $G.711$  with FEC, and  $G.729a$  with FEC, 800/sec., 18/sec., 80/sec., 11/sec., 53/sec. respectively. We then scale up the number of long-lived TCP flows and the arrival rates of all other five types of flows proportional to vy factor  $k$ . Figure 6 plots the bandwidth shares of all types of flows as we scale up the traffic. We observe from the figure that the share of short-lived TCP flow increases proportionally with its arrival rate. This is because the aggregated throughput of all short-lived TCP flows does not depends on the network condition but only depends on the arrival rate and average session size. The share of  $G.711$  with FEC increases first then stabilizes whereas the share of  $G.729a$  with FEC first increases slowly then decreases slowly. The bandwidth share of  $G.711$  drops all the time. Long-lived TCP flows first loss their share faster than  $G.711$  but then stabilize and share roughly the same amount of bandwidth as  $G.711$ . The share of  $G.729a$  drops all the time, which only has more shares than TCP for a little while then loss its share much faster than long-lived TCP.

Based on the same bottleneck model, Figure 7 shows how the call quality changes as the link utilization increases. The quality of  $G.729$  call starts to decrease as link utilization approaches 75%. It then drops a lot as link utilization grows further. The quality of  $G.711$  calls starts to decrease at 80% link utilization and drop slower than  $G.729a$ . The  $G.729a$  with FEC maintain its quality very well until link utilization reach 90%. The  $G.711$  with FEC can maintain its quality even when link utilization goes above 95%

We now investigate how the throughput of individual long-lived TCP flow changes as the number of VoIP flows increase. We use the same bottleneck link setting and assume there are only two types of flows in the network: fixed 10 long-lived TCP flows and  $G.711$  VoIP

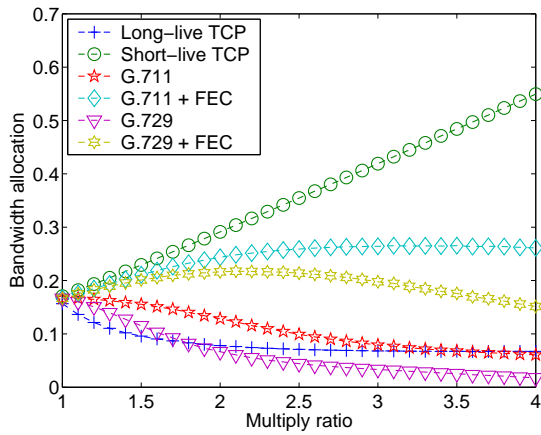


Fig. 6. How different types of flows share bandwidth in a single bottleneck link

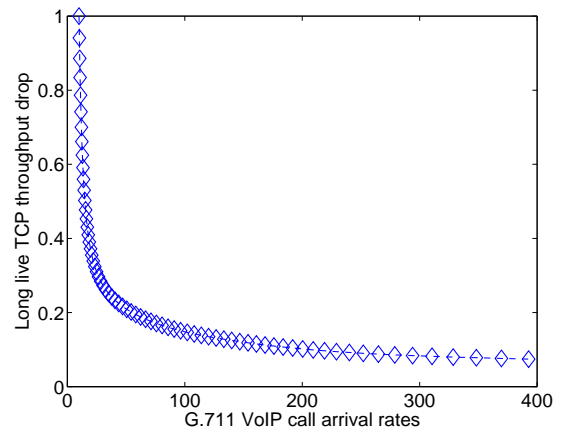


Fig. 8. TCP throughput drops as *G.711* call arrival rate increases

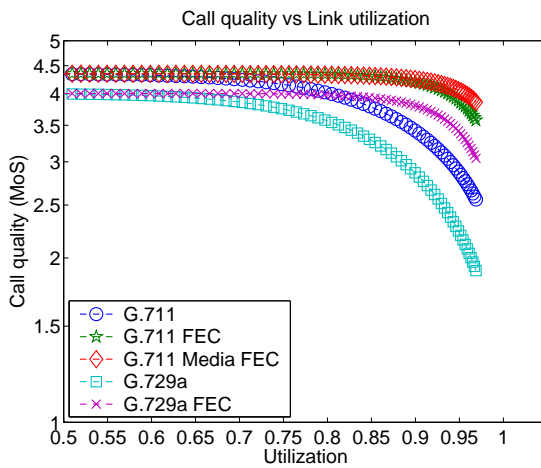


Fig. 7. Call quality for different codecs as link utilization increases

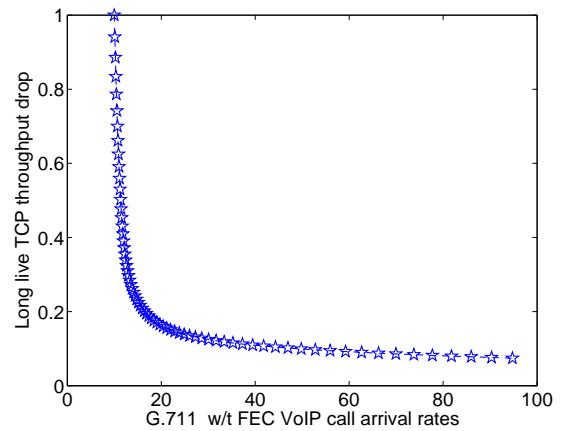


Fig. 9. TCP throughput drops quicker when FEC is used by *G.711* flows.

flows with mean duration 120 seconds and a variable arrival rate. We start with an arrival rate of 10 calls/second and then gradually increase the VoIP arrival rate. In Figure 8, we plot the TCP throughput as we increase *G.711* flows arrival rate. TCP throughput drops fast when we first triple the VoIP call arrival rate. TCP throughput drop slows down as we further increase the VoIP arrival rate. The curve become almost flat as we increase the arrival rate beyond 20 times. Figure 9 plots changes in TCP throughput when FEC is used by *G.711* flows. The trend is similar except that TCP throughput drops faster than in Figure 8. This is because, with the same arrival rate, *G.711* flows with FEC will have better quality and less likely to drop out than *G.711* flows without FEC. Therefore they can grab more bandwidth from TCP flows.

### VI. DISCUSSIONS

In the previous sections, we have shown that VoIP applications are TCP-Friendly due to their user back-

off mechanism. In this section, we provide more general discussions on how user behavior and application design affect network resource sharing and congestion control.

Most previous studies of fair sharing of network resource focused on the transport layer and below [31]–[33]. Due to the diversity in network application design and user behavior, the fairness TCP tries to maintain at the transport layer is no longer sufficient to ensure the fair sharing of network resource and the stability of the whole network. Recently, there is an increasing trend to employ application level control to improve users’ performance over shared wide area networks. For example, *Parallel TCP* is employed by some web and peer-peer applications to accelerate data access [34], [35]. Most popular peer-peer file sharing applications, such as Kazaa [36], BitTorrent [37], open multiple concurrent TCP connections to download large files. It has been shown that the aggregate behavior of concurrent TCP connections is much more aggressive than a single TCP connection [34], [35]. Unlike interactive VoIP sessions, those data transfer sessions are very resilient to

packet loss and packet delay. Peer-Peer users normally don't drop their sessions. As pointed out in [7], non-reactive large TCP downloads can significantly degrade network performance and potentially result in congestion collapse. Compared with UDP based VoIP applications, those TCP based Peer-Peer file sharing applications are more responsible for network congestion. As a simple practice, how many users will still want to use Skype to make a phone call if there are several BitTorrent or Kazaa sessions running on the same computer?

We have shown that VoIP traffic won't endanger the current Internet because of its low data rate and its user back-off mechanism. As network bandwidth increases, video over network becomes more practical. Video applications require much higher data rates than VoIP. It has been conjectured that humans are more tolerant to video quality degradation than to voice. Video users may be less conservative in user back-off, or equivalently more aggressive in grabbing bandwidth. At the same time, current TCP has proven to be inefficient at obtaining available bandwidth in future high bandwidth networks. New congestion control protocols, such as HighSpeed TCP [38], Scalable TCP [39], FAST [40], have been proposed to achieve fast end-end data transmission. How video applications interact with new congestion control schemes deserves further study.

## VII. CONCLUSION

In this paper, we investigate the TCP-Friendliness of VoIP traffic under different network environment. Various models have been established to compute VoIP user back-off probabilities as functions of packet loss rate in different network environment. By comparing VoIP user back-off with TCP rate adaption, we show that VoIP flows not only consume less bandwidth than TCP flows, they are also very responsive to congestion when the network is highly loaded. Based on this, we disprove the conjecture that VoIP is not TCP-Friendly and the Internet will melt down eventually as VoIP type of UDP traffic increases. More broadly, our work demonstrates the importance of user behavior in the study of network resource sharing and congestion control. We predict that user behavior and application level design will play an increasingly important role in network analysis along with the evolution of network infrastructure, networked applications, as well as users of networks.

## REFERENCES

- [1] Unknown, *Skype Web Page*, <http://www.skype.com>.
- [2] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.
- [3] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *Columbia University Technical Report CUCS-039-04*, 2004.
- [4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling tcp throughput: A simple model and its empirical," in *Proceedings of ACM/SIGCOMM*, 1998.
- [5] N. Cardwell, S. Savage, and T. Anderson, "Modeling the performance of short tcp connections," Computer Science Department, University of Washington, Tech. Rep., November 1998.
- [6] "Subjective test plan for characterization of an 8 kbit/s speech code, ITU-T study group 12, speech quality experts group," Sep 1995.
- [7] R. Morris and Y. Tay, "A model for analyzing the roles of network and user behavior in congestion control," Computer Science Department, Massachusetts Institute of Technology, Tech. Rep. MIT-LCS-TR898, May 2004.
- [8] R. G. Cole and J. Rosenbluth, "Voice over ip performance monitoring," *Computer Communication Review*, vol. 31, no. 2, pp. 9–24, April 2001.
- [9] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Improving voip quality through path switching," in *Proceedings of IEEE Conference on Computer and Communications (INFOCOM)*, March 2005.
- [10] A. Markopoulou, F. Tobagi, and M. J. Karam, "Assessing the quality of voice communications over internet backbones," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 747–760, Oct 2003.
- [11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
- [12] C. Holot, V. Misra, D. Towsley, and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE Conference on Computer and Communications (INFOCOM)*, 2001.
- [13] O. Boxma and Dumas, "The busy period in the fluid queue," in *Proceedings of Sigmetrics/Performance'98*, 1998, pp. 100–110.
- [14] V. Paxson and S. Floyd, "Wide Area traffic: The failure of poisson modelling," in *Proceedings of SIGCOMM*, 1994.
- [15] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking* 2, vol. 6, pp. 1–15, 1994.
- [16] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, December 1997.
- [17] M. Jainik, S. B. Moon, J. F. Kurose, and D. F. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *IEEE INFOCOM*, 1999, pp. 345–352.
- [18] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *Proc. of ACM SIGCOMM Internet Measurement Workshop 2002, Marseille, France*, Nov 2002.
- [19] J. Cao and K. Ramanan, "A poisson limit for buffer overflow probabilities," in *Proceedings of IEEE Conference on Computer and Communications (INFOCOM)*, March 2002.
- [20] S. Alouf, P. Nain, and D. Towsley, "Inferring network characteristics via moment-based estimators," in *IEEE Infocom 2001 Conference*, 2001.
- [21] "The emodel: A computational model for use in transmission planning, ITU-T recommendation g.107," Dec 1998.
- [22] "Method for subjective determination of transmission quality, ITU-T recommendation p.800," Aug 1996.
- [23] "Definition of categories of speech transmission quality, ITU-T recommendation g.109," Sept 1999.

- [24] "An objective method of end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, ITU-T recommendation p.862," Jan 2001.
- [25] "Appendix i: Provisional planning values for the equipment impairment factor  $i_e$ , ITU-T recommendation g.113," Sept 1999.
- [26] J. Rosenberg and H. Schulzrinne, "An rtp payload format for generic forward error correction," Dec 1999.
- [27] V. Hardman, A. Sasse, M. Handley, and A. Watson, "A reliable audio for use over the internet," in *Inet 95*, Honolulu, Hawaii, Jun 1995.
- [28] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, Sept 1998.
- [29] J. Rosenberg, L. Qiu, and H. Schulzrinne, "Integrating packet fec into adaptive voice playout buffer algorithms on the internet," in *IEEE Infocom*, Tel Aviv, Israel, Mar 2000.
- [30] W. Jiang and H. Schulzrinne, "Comparison and optimization of packet loss repair methods on voip perceived quality under bursty loss," in *NASDAV*, 2002.
- [31] J. M. Jaffe, "Bottleneck flow control," *IEEE Transactions on Communications*, vol. COM-29, no. 7, pp. 954–962, July 1981.
- [32] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, June 1989.
- [33] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices proportional fairness and stability," *Journal of the Operational Research Society*, 1998.
- [34] Y. Liu, W. Gong, and P. Shenoy, "On the impact of concurrent downloads," in *Proceedings of Winter Simulation Conference*, December 2001, pp. 1300–1305.
- [35] T. J. Hacker, B. D. Noble, and B. D. Athey, "Improving throughput and maintaining fairness using parallel tcp," in *Proceedings of the IEEE INFOCOM*, 2004.
- [36] unknown, *Kazaa Web Site*, <http://www.kazaa.com/>.
- [37] B. Cohen, *Bittorrent Web Site*, 2004, <http://bitconjurer.org/BitTorrent/>.
- [38] S. Floyd, "RFC 3649: Highspeed TCP for large congestion windows," IETF, Tech. Rep., 2003.
- [39] T. Kelly, "Scalable TCP: improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, pp. 83–91, April 2003.
- [40] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, performance," in *Proceedings of IEEE Conference on Computer and Communications (INFOCOM)*, March 2004.