Network Working Group
Request for Comments: 3714
Category: Informational

S. Floyd, Ed.
J. Kempf, Ed.

March 2004

IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document discusses IAB concerns about effective end-to-end congestion control for best-effort voice traffic in the Internet. These concerns have to do with fairness, user quality, and with the dangers of congestion collapse. The concerns are particularly relevant in light of the absence of a widespread Quality of Service (QoS) deployment in the Internet, and the likelihood that this situation will not change much in the near term. This document is not making any recommendations about deployment paths for Voice over Internet Protocol (VoIP) in terms of QoS support, and is not claiming that best-effort service can be relied upon to give acceptable performance for VoIP. We are merely observing that voice traffic is occasionally deployed as best-effort traffic over some links in the Internet, that we expect this occasional deployment to continue, and that we have concerns about the lack of effective end-to-end congestion control for this best-effort voice traffic.

Table of Contents

1. Introduction

While many in the telephony community assume that commercial VoIP service in the Internet awaits effective end-to-end QoS, in reality voice service over best-effort broadband Internet connections is an available service now with growing demand. While some ISPs deploy QoS on their backbones, and some corporate intranets offer end-to-end QoS internally, end-to-end QoS is not generally available to customers in the current Internet. Given the current commercial interest in VoIP on best-effort media connections, it seems prudent to examine the potential effect of real time flows on congestion. In this document, we perform such an analysis. Note, however, that this document is not making any recommendations about deployment paths for VoIP in terms of QoS support, and is not claiming that best-effort service can be relied upon to give acceptable performance for VoIP. This document is also not discussing signalling connections for VoIP. However, voice traffic is in fact occasionally deployed as best effort traffic over some links in the Internet today, and we expect this occasional deployment to continue. This document expresses our concern over the lack of effective end-to-end congestion control for this best-effort voice traffic.

Assuming that VoIP over best-effort Internet connections continues to gain popularity among consumers with broadband connections, the deployment of end-to-end QoS mechanisms in public ISPs may be slow. The IETF has developed standards for QoS mechanisms in the Internet [DIFFSERV, RSVP] and continues to be active in this area [NSIS,COPS]. However, the deployment of technologies requiring change to the Internet infrastructure is subject to a wide range of commercial as well as technical considerations, and technologies that can be deployed without changes to the infrastructure enjoy considerable advantages in the speed of deployment. RFC 2990 outlines some of the technical challenges to the deployment of QoS architectures in the Internet [RFC2990]. Often, interim measures that provide support for fast-growing applications are adopted, and are successful enough at meeting the need that the pressure for a ubiquitous deployment of the more disruptive technologies is reduced. There are many examples of the slow deployment of infrastructure that are similar to the slow deployment of QoS mechanisms, including IPv6, IP multicast, or of a global PKI for IKE and IPsec support.

Interim QoS measures that can be deployed most easily include single-hop or edge-only QoS mechanisms for VoIP traffic on individual congested links, such as edge-only QoS mechanisms for cable access networks. Such local forms of QoS could be quite successful in protecting some fraction of best-effort VoIP traffic from congestion. However, these local forms of QoS are not directly visible to the end-to-end VoIP connection. A best-effort VoIP connection could experience high end-to-end packet drop rates, and be competing with other best-effort traffic, even if some of the links along the path might have single-hop QoS mechanisms.

The deployment of IP telephony is likely to include best-effort broadband connections to public-access networks, in addition to other deployment scenarios of dedicated IP networks, or as an alternative to band splitting on the last mile of ADSL deployments or QoS mechanisms on cable access networks. There already exists a rapidly-expanding deployment of VoIP services intended to operate over residential broadband access links (e.g., [FWD, Vonage]). At the moment, many public-access IP networks are uncongested in the core, with low or moderate levels of link utilization, but this is not necessarily the case on last hop links. If an IP telephony call runs completely over the Internet, the connection could easily traverse congested links on both ends. Because of economic factors, the growth rate of Internet telephony is likely to be greatest in developing countries, where core links are more likely to be congested, making congestion control an especially important topic for developing countries.

Given the possible deployment of IP telephony over congested best-effort networks, some concerns arise about the possibilities of congestion collapse due to a rapid growth in real-time voice traffic that does not practice end-to-end congestion control. This document raises some concerns about fairness, user quality, and the danger of congestion collapse that would arise from a rapid growth in best-effort telephony traffic on best-effort networks. We consider best-effort telephony connections that have a minimum sending rate and that compete directly with other best-effort traffic on a path with at least one congested link, and address the specific question of whether such traffic should be required to terminate, or to suspend sending temporarily, in the face of a persistent, high packet drop rate, when reducing the sending rate is not a viable alternative.

The concerns in this document about fairness and the danger of congestion collapse apply not only to telephony traffic, but also to video traffic and other best-effort real-time traffic with a minimum sending rate. RFC 2914 already makes the point that best-effort traffic requires end-to-end congestion control [RFC2914]. Because audio traffic sends at such a low rate, relative to video and other real-time traffic, it is sometimes claimed that audio traffic doesn't require end-to-end congestion control. Thus, while the concerns in this document are general, the document focuses on the particular issue of best-effort audio traffic.

Feedback can be sent to the IAB mailing list at iab@ietf.org, or to the editors at floyd@icir.org and kempf@docomolabs-usa.com. Feedback can also be sent to the end2end-interest mailing list [E2E].

2. An Example of the Potential for Trouble

At the November, 2002, IEPREP Working Group meeting in Atlanta, a brief demonstration was made of VoIP over a shared link between a hotel room in Atlanta, Georgia, USA, and Nairobi, Kenya. The link ran over the typical uncongested Internet backbone and access links to peering points between either endpoint and the Internet backbone. The voice quality on the call was very good, especially in comparison to the typical quality obtained by a circuit-switched call with Nairobi. A presentation that accompanied the demonstration described the access links (e.g., DSL, T1, T3, dialup, and cable modem links) as the primary source of network congestion, and described VoIP traffic as being a very small percentage of the packets in commercial ISP traffic [A02]. The presentation further stated that VoIP received good quality in the presence of packet drop rates of 5-40% [AUT]. The VoIP call used an ITU-T G.711 codec, plus proprietary FEC encoding, plus RTP/UDP/IP framing. The resulting traffic load over the Internet was substantially more than the 64 kbps required by the codec. The primary congestion point along the path of the demonstration was a 128 kbps access link between an ISP in Kenya and several of its subscribers in Nairobi. So the single VoIP call consumed more than half of the access link capacity, capacity that is shared across several different users.

Note that this network configuration is not a particularly good one for VoIP. In particular, if there are data services running TCP on the link with a typical packet size of 1500 bytes, then some voice packets could be delayed an additional 90 ms, which might cause an increase in the end to end delay above the ITU-recommended time of 150 ms [G.114] for speech traffic. This would result in a delay noticeable to users, with an increased variation in delay, and therefore in call quality, as the bursty TCP traffic comes and goes. For a call that already had high delay, such as the Nairobi call from the previous paragraph, the increased jitter due to competing TCP traffic also increases the requirements on the jitter buffer at the receiver. Nevertheless, VoIP usage over congested best-effort links is likely to increase in the near future, regardless of VoIP's superior performance with "carrier class" service. A best-effort VoIP connection that persists in sending packets at 64 Kbps, consuming half of a 128 Kbps access link, in the face of a drop rate of 40%, with the resulting user-perceptible degradation in voice quality, is not behaving in a way that serves the interests of either the VoIP users or the other concurrent users of the network.

As the Nairobi connection demonstrates, prescribing universal overprovisioning (or more precisely, provisioning sufficient to avoid persistent congestion) as the solution to the problem is not an acceptable generic solution. For example, in regions of the world where circuit-switched telephone service is poor and expensive, and Internet access is possible and lower cost, provisioning all Internet links to avoid congestion is likely to be impractical or impossible.

In particular, an over-provisioned core is not by itself sufficient to avoid congestion collapse all the way along the path, because an over-provisioned core can not address the common problem of congestion on the access links. Many access links routinely suffer from congestion. It is important to avoid congestion collapse along the entire end-to-end path, including along the access links (where congestion collapse would consist of congested access links wasting scarce bandwidth carrying packets that will only be dropped downstream). So an over-provisioned core does not by itself eliminate or reduce the need for end-to-end congestion avoidance and control.

There are two possible mechanisms for avoiding this congestion collapse: call rejection during busy periods, or the use of end-to-end congestion control. Because there are currently no acceptance/rejection mechanisms for best-effort traffic in the Internet, the only alternative is the use of end-to-end congestion control. This is important even if end-to-end congestion control is invoked only in those very rare scenarios with congestion in generally-uncongested access links or networks. There will always be occasional periods of high demand, e.g., in the two hours after an earthquake or other disaster, and this is exactly when it is important to avoid congestion collapse.

Best-effort traffic in the Internet does not include mechanisms for call acceptance or rejection. Instead, a best-effort network itself is largely neutral in terms of resource management, and the interaction of the applications' transport sessions mutually regulates network resources in a reasonably fair fashion. One way to bring voice into the best-effort environment in a non-disruptive manner is to focus on the codec and look at rate adaptation measures that can successfully interoperate with existing transport protocols (e.g., TCP), while at the same time preserving the integrity of a real-time, analog voice signal; another way is to consider codecs with fixed sending rates. Whether the codec has a fixed or variable sending rate, we consider the appropriate response when the codec is at its minimum data rate, and the packet drop rate experienced by the flow remains high. This is the key issue addressed in this document.

3. Why are Persistent, High Drop Rates a Problem?

Persistent, high packet drop rates are rarely seen in the Internet today, in the absence of routing failures or other major disruptions. This happy situation is due primarily to low levels of link utilization in the core, with congestion typically found on lower-capacity access links, and to the use of end-to-end congestion control in TCP. Most of the traffic on the Internet today uses TCP, and TCP self-corrects so that the two ends of a connection reduce the rate of packet sending if congestion is detected. In the sections below, we discuss some of the problems caused by persistent, high packet drop rates.

3.1. Congestion Collapse

One possible problem caused by persistent, high packet drop rates is that of congestion collapse. Congestion collapse was first observed during the early growth phase of the Internet of the mid 1980s [RFC896], and the fix was provided by Van Jacobson, who developed the congestion control mechanisms that are now required in TCP implementations [Jacobson88, RFC2581].

As described in RFC 2914, congestion collapse occurs in networks with flows that traverse multiple congested links having persistent, high packet drop rates [RFC2914]. In particular, in this scenario packets that are injected onto congested links squander scarce bandwidth since these packets are only dropped later, on a downstream congested link. If congestion collapse occurs, all traffic slows to a crawl and nobody gets acceptable packet delivery or acceptable performance. Because congestion collapse of this form can occur only for flows that traverse multiple congested links, congestion collapse is a potential problem in VoIP networks when both ends of the VoIP call are on an congested broadband connection such as DSL, or when the call traverses a congested backbone or transoceanic link.

3.2. User Quality

A second problem with persistent, high packet drop rates concerns service quality seen by end users. Consider a network scenario where each flow traverses only one congested link, as could have been the case in the Nairobi demonstration above. For example, imagine N VoIP flows sharing a 128 Kbps link, with each flow sending at least 64 Kbps. For simplicity, suppose the 128 Kbps link is the only congested link, and there is no traffic on that link other than the N VoIP calls. We will also ignore for now the extra bandwidth used by the telephony traffic for FEC and packet headers, or the reduced bandwidth (often estimated as 70%) due to silence suppression. We also ignore the fact that the two streams composing a bidirectional VoIP call, one for each direction, can in practice add to the load on some links of the path. Given these simplified assumptions, the arrival rate to that link is at least N*64 Kbps. The traffic actually forwarded is at most 2*64 Kbps (the link bandwidth), so at least (N-2)*64 Kbps of the arriving traffic must be dropped. Thus, a fraction of at least (N-2)/N of the arriving traffic is dropped, and each flow receives on average a fraction 1/N of the link bandwidth. An important point to note is that the drops occur randomly, so that no one flow can be expected statistically to present better quality service to users than any other. Everybody's voice quality therefore suffers.

It seems clear from this simple example that the quality of best-effort VoIP traffic over congested links can be improved if each VoIP flow uses end-to-end congestion control, and has a codec that can adapt the bit rate to the bandwidth actually received by that flow. The overall effect of these measures is to reduce the aggregate packet drop rate, thus improving voice quality for all VoIP users on the link. Today, applications and popular codecs for Internet telephony attempt to compensate by using more FEC, but controlling the packet flow rate directly should result in less redundant FEC information, and thus less bandwidth, thereby improving throughput even further. The effect of delay and packet loss on VoIP in the presence of FEC has been investigated in detail in the literature [JS00, JS02, JS03, MTK03]. One rule of thumb is that when the packet loss rate exceeds 20%, the audio quality of VoIP is degraded beyond usefulness, in part due to the bursty nature of the losses [S03]. We are not aware of measurement studies of whether VoIP users in practice tend to hang up when packet loss rates exceed some limit.

The simple example in this section considered only voice flows, but in reality, VoIP traffic will compete with other flows, most likely TCP. The response of VoIP traffic to congestion works best by taking into account the congestion control response of TCP, as is discussed in the next subsection.

3.3. The Amorphous Problem of Fairness

A third problem with persistent, high packet drop rates is fairness. In this document we consider fairness with regard to best-effort VoIP traffic competing with other best-effort traffic in the Internet. That is, we are explicitly not addressing the issues raised by emergency services, or by QoS-enabled traffic that is known to be treated separately from best-effort traffic at a congested link.

While fairness is a bit difficult to quantify, we can illustrate the effect by adding TCP traffic to the congested link discussed in the previous section. In this case, the non-congestion-controlled traffic and congestion-controlled TCP traffic [RFC2914] share the link, with the congestion-controlled traffic's sending rate determined by the packet drop rate experienced by those flows. As in the previous section, the 128 Kbps link has N VoIP connections each sending 64 Kbps, resulting in packet drop rate of at least (N-2)/N on the congested link. Competing TCP flows will experience the same packet drop rates. However, a TCP flow experiencing the same packet drop rates will be sending considerably less than 64 Kbps. From the point of view of who gets what amount of bandwidth, the VoIP traffic is crowding out the TCP traffic.

Of course, this is only one way to look at fairness. The relative fairness between VoIP and TCP traffic can be viewed several different ways, depending on the assumptions that one makes on packet sizes and round-trip times. In the presence of a fixed packet drop rate, for example, a TCP flow with larger packets sends more (in Bps, bytes per second) than a TCP flow with smaller packets, and a TCP flow with a shorter round-trip time sends more (in Bps) than a TCP flow with a larger round-trip time. In environments with high packet drop rates, TCP's sending rate depends on the algorithm for setting the retransmit timer (RTO) as well, with a TCP implementation having a more aggressive RTO setting sending more than a TCP implementation having a less aggressive RTO setting.

Unfortunately, there is no obvious canonical round-trip time for judging relative fairness of flows in the network. Agreement in the literature is that the majority of packets on most links in the network experience round-trip times between 10 and 500 ms [RTTWeb].

(This does not include satellite links.) As a result, if there was a canonical round-trip for judging relative fairness, it would have to be within that range. In the absence of a single representative round-trip time, the assumption of this paper is that it is reasonable to consider fairness between a VoIP connection and a TCP connection with the same round-trip time.

Similarly, there is no canonical packet size for judging relative fairness between TCP connections. However, because the most common packet size for TCP data packets is 1460 bytes [Measurement], we assume that it is reasonable to consider fairness between a VoIP connection, and a TCP connection sending 1460-byte data packets. Note that 1460 bytes is considerably larger than is typically used for VoIP packets.

In the same way, while RFC 2988 specifies TCP's algorithm for setting TCP's RTO, there is no canonical value for the minimum RTO, and the minimum RTO heavily affects TCP's sending rate in times of high congestion [RFC2988]. RFC 2988 specifies that TCP's RTO must be set to SRTT + 4*RTTVAR, for SRTT the smoothed round-trip time, and for RTTVAR the mean deviation of recent round-trip time measurements. RFC 2988 further states that the RTO "SHOULD" have a minimum value of 1 second. However, it is not uncommon in practice for TCP implementations to have a minimum RTO as low as 100 ms. For the purposes of this document, in considering relative fairness, we will assume a minimum RTO of 100 ms.

As an additional complication, TCP connections that use fine-grained timestamps can have considerably higher sending rates than TCP connections that do not use timestamps, in environments with high packet drop rates. For TCP connections with fine-grained timestamps, a valid round-trip time measurement is obtained when a retransmitted packet is successfully received and acknowledged by the receiver; in this case a backed-off retransmit timer can be un-backed-off as well. For TCP connections without timestamps, a valid round-trip time measurement is only obtained when the transmission of a new packet is received and acknowledged by the receiver. This limits the opportunities for the un-backing-off of a backed-off retransmit timer. In this document, in considering relative fairness, we use a TCP connection without timestamps, since this is the dominant use of TCP in the Internet.

A separate claim that has sometimes been raised in terms of fairness is that best-effort VoIP traffic is inherently more important that other best-effort traffic (e.g., web surfing, peer-to-peer traffic, or multi-player games), and therefore merits a larger share of the bandwidth in times of high congestion. Our assumption in this document is that TCP traffic includes pressing email messages, business documents, and emergency information downloaded from web pages, as well as the more recreational uses cited above. Thus, we do not agree that best-effort VoIP traffic should be exempt from end-to-end congestion control due to any claims of inherently more valuable content. (One could equally logically argue that because email and instant messaging are more efficient forms of communication than VoIP in terms of bandwidth usage, as a result email and instant messaging are more valuable uses of scarce bandwidth in times of high congestion.) In fact, the network is incapable of making a judgment about the relative user value of traffic. The default assumption is that all best-effort traffic has equal value to the network provider and to the user.

We note that this discussion of relative fairness does not in any way challenge the right of ISPs to allocate bandwidth on congested links to classes of traffic in any way that they choose. (For example, administrators rate-limit the bandwidth used by peer-to-peer traffic on some links in the network, to ensure that bandwidth is also available for other classes of traffic.) This discussion merely argues that there is no reason for entire classes of best-effort traffic to be exempt from end-to-end congestion control.

4. Current efforts in the IETF

There are four efforts currently underway in IETF to address issues of congestion control for real time traffic: an upgrade of the RTP specification, TFRC, DCCP, and work on audio codecs.

4.1. RTP

RFC 1890, the original RTP Profile for Audio and Video Control, does not discuss congestion control [RFC1890]. The revised document on "RTP Profile for Audio and Video Conferences with Minimal Control" [RFC3551] discusses congestion control in Section 2. [RFC3551] says the following:

> "If best-effort service is being used, RTP receivers SHOULD monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve an average throughput, measured on a reasonable timescale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast session), or by arranging for a receiver to leave the session if the loss rate is unacceptably high."

> "The comparison to TCP cannot be specified exactly, but is intended as an "order-of-magnitude" comparison in timescale and throughput. The timescale on which TCP throughput is measured is the round-trip time of the connection. In essence, this requirement states that it is not acceptable to deploy an application (using RTP or any other transport protocol) on the best-effort Internet which consumes bandwidth arbitrarily and does not compete fairly with TCP within an order of magnitude."

Note that [RFC3551] says that receivers "SHOULD" monitor packet loss. [RFC3551] does not explicitly say that the RTP senders and receivers "MUST" detect and respond to a persistent high loss rate. Since congestion collapse can be considered a "danger to the Internet" the use of "MUST" would be appropriate for RTP traffic in the best-effort Internet, where the VoIP traffic shares a link with other traffic, since "danger to the Internet" is one of two criteria given in RFC 2119 for the use of "MUST" [RFC2119]. Different requirements may hold for a private best-effort IP network provisioned solely for VoIP, where the VoIP traffic does not interact with the wider Internet.

4.2. TFRC

As mentioned in RFC 3267, equation-based congestion control is one of the possibilities for VoIP. TCP Friendly Rate Control (TFRC) is the equation-based congestion control mechanism that has been standardized in the IETF. The TFRC specification, "TCP Friendly Rate Control (TFRC): Protocol Specification" [RFC3448], says the following:

> "TFRC ... is reasonably fair when competing for bandwidth with TCP flows, but has a much lower variation of throughput over time compared with TCP, making it more suitable for applications such as telephony or streaming media where a relatively smooth sending rate is of importance. ... TFRC is designed for applications that use a fixed packet size, and vary their sending rate in packets per second in response to congestion. Some audio applications require a fixed interval of time between packets and vary their packet size instead of their packet rate in response to congestion. The congestion control mechanism in this document cannot be used by those applications; TFRC-PS (for TFRC-PacketSize) is a variant of TFRC for applications that have a fixed sending rate but vary their packet size in response to congestion. TFRC-PS will be specified in a later document."

There is no draft available for TFRC-PS yet, unfortunately, but several researchers are still working on these issues.

4.3. DCCP

The Datagram Congestion Control Protocol (DCCP) is a transport protocol being standardized in the IETF for unreliable flows, with the application being able to specify either TCP-like or TFRC congestion control [DCCP03].

DCCP currently has two Congestion Control IDentifiers or CCIDs; these are CCID 2 for TCP-like congestion control and CCID 3 for TFRC congestion control. As TFRC-PS becomes available and goes through the standards process, we would expect DCCP to create a new CCID, CCID 4, for use with TFRC-PS congestion control.

4.4. Adaptive Rate Audio Codecs

A critical component in the design of any real-time application is the selection of appropriate codecs, specifically codecs that operate at a low sending rate, or that will reduce the sending rate as throughput decreases and/or packet loss increases. Absent this, and in the absence of the response to congestion recommended in this document, the real-time application is likely to significantly increase the risk of Internet congestion collapse, thereby adversely impacting the health of the deployed Internet. If the codec is capable of reducing its bit rate in response to congestion, this improves the scaling of the number of VoIP or TCP sessions capable of sharing a congested link while still providing acceptable performance to users. Many current audio codecs are capable of sending at a low bit rate, in some cases adapting their sending rate in response to congestion indications from the network.

RFC 3267 describes RTP payload formats for use with the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) audio codecs [RFC 3267]. The AMR codec supports eight speech encoding modes having bit rates between 4.75 and 12.2 kbps, with the speech encoding performed on 20 ms speech frames, and is able to reduce the transmission rate during silence periods. The payload format specified in RFC 3267 includes forward error correction (FEC) and frame interleaving to increase robustness against packet loss somewhat. The AMR codec was chosen by the Third Generation Partnership Project (3GPP) as the mandatory codec for third generation (3G) cellular systems, and RFC 3267 recommends that AMR or AMR-WB applications using the RTP payload format specified in RFC 3267 use congestion control, though no specific mechanism is recommended. RFC 3267 gives "Equation-Based Congestion Control for Unicast Applications" as an example of a congestion control mechanism suitable for real-time flows [FHPW00].

The "Internet Low Bit Rate Codec", iLBC, is an IETF effort to develop an IPR-free codec for robust voice communication over IP [ILBRC]. The codec is designed for graceful speech quality degradation in the case of lost packets, and has a payload bit rate of 13.33 kbps for 30 ms frames or 15.20 kbps for 20 ms frames.

There are several unencumbered low-rate codec algorithms in Ivox (the Interactive VOice eXchange) [IVOX], with plans to add additional variable rate codecs. For example, LPC2400 (a.k.a. LQ2400) is a 2400 bps LPC based codec with an enhancement to permit "silence detection". The 2400 bps codec is reported to have a "slight robotic quality" [A03] (even without the additional complications of packet loss). The older multirate codec described in [KFK79, KF82] is an LPC codec that works at two rates, 2.4 kbps and 9.6 kbps, and can optionally send additional "residual" bits for enhanced quality at a higher bit rate.

Off-the-shelf ITU-T vocoders such as G.711 were generally designed explicitly for circuit-switched networks and are not as well-adapted for Internet use, even with the addition of FEC on top.

4.5. Differentiated Services and Related Topics

The Differentiated Services Working Group [DIFFSERV], which concluded in 2003, completed standards for the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers [RFC2474], including several per-hop forwarding behaviors [RFC2597, RFC3246]. The Next Steps in Signaling Working Group [NSIS] is developing an optimized signalling protocol for QoS, based in part on earlier work of the Resource Reservation Setup Protocol Working Group [RSVP]. We do not discuss these and related efforts further in this document, since this document concerns only that VoIP traffic that might be carried as best-effort traffic over some congested link in the Internet.

5. Assessing Minimum Acceptable Sending Rates

Current IETF work in the DCCP and AVT working groups does not consider the problem of applications that have a minimum sending rate and are not able to go below that sending rate. This clearly must be addressed in the TFRC-PS draft. As suggested in the RTP document, if the loss rate is persistently unacceptably high relative to the current sending rate, and the best-effort application is unable to lower its sending rate, then the only acceptable answer is for that flow to discontinue sending on that link. For a multicast session, this could be accomplished by the receiver withdrawing from the multicast group. For a unicast session, this could be accomplished by the unicast connection terminating, at least for a period of time.

We can formulate a problem statement for the minimum sending rate in the following way. Consider a best-effort, adaptive audio application that is able to adapt down to a minimum sending rate of N Bps (bytes per second) of application data, sending M packets per second. Is this a sufficiently low sending rate that the best-effort flow is never required to terminate due to congestion, or to reduce its sending rate in packets per second still further? In other words, is N Bps an acceptable minimum sending rate for the application, which can be continued in the face of congestion without terminating or suspending the application?

We assume, generously for VoIP, that the limitation of the network is in bandwidth in bytes per second (Bps), and not in CPU cycles or in packets per second (pps). If the limitation in the network is in bandwidth, this is a limitation in Bps, while if the limitation is in router processing capacity in packets, this would be a limitation in pps. We note that TCP sends fixed-size data packets, and reduces its sending rate in pps when it adapts to network congestion, thus reducing the load on the forward path both in Bps and in pps. In contrast, for adaptive VoIP applications, the adaption is sometimes to keep the same sending rate in pps, but to reduce the packet size, reducing the sending rate in Bps. This fits the needs of audio as an application, and is a good response on a network path where the limitation is in Bps. Such behavior would be a less appropriate response for a network path where the limitation is in pps.

If the network limitation in fact is in Bps, then all that matters in terms of congestion is a flow's sending rate on the wire in Bps. If this assumption of a network limitation in Bps is false, then the sending rate in pps could contribute to congestion even when the sending rate in Bps is quite moderate. While the ideal would be to have a transport protocol that is able to detect whether the bottleneck links along the path are limited in Bps or in pps, and to respond appropriately when the limitation is in pps, such an ideal is hard to achieve. We would not want to delay the deployment of congestion control for telephony traffic until such an ideal could be accomplished. In addition, we note that the current TCP congestion control mechanisms are themselves not very effective in an environment where there is a limitation along the reverse path in pps. While the TCP mechanisms do provide an incentive to use large data packets, TCP does not include any effective congestion control mechanisms for the stream of small acknowledgement packets on the reverse path. Given the arguments above, it seems acceptable to us to assume a network limitation in Bps rather than in pps in considering the minimum sending rate of telephony traffic.

Assuming 40-byte packet headers (IP, RTP, and UDP or DCCP), the application data sending rate of N Bps and M pps translates to a sending rate on the wire of B = N+40M Bps. If the application uses additional FEC (Forward Error Correction), the FEC bits must be added in as well. In our example, we ignore bandwidth adjustments that are needed to take into account the additional overhead for FEC or the reduced sending rate for silence periods. We also are not taking into account the possible role of header compression on congested edge links, which can reduce significantly the number of bytes used for headers on those links.

Now, consider an equivalent-rate TCP connection with data packets of P bytes and a round-trip time of R seconds. Taking into account header size, such a TCP connection with a sending rate on the wire of B Bps is sending B/(P+40) pps, or, equivalently, BR/(P+40) ppr (packets per round-trip time).

Restating the question in terms of the above expressions for VoIP and TCP: if the best-effort VoIP connection is experiencing a persistent packet drop rate of D, and is at its minimum sending rate on the wire of B Bps, when should the application or transport protocol terminate or suspend the VoIP connection?

One answer to this question is to find the sending rate in ppr for a TCP connection sending at the same rate on the wire in Bps, and to use the TCP response function to determine whether a conformant TCP connection would be able to maintain a sending rate close to that sending rate with the same persistent drop rate D. If the sending rate of the VoIP connection is significantly higher than the sending rate of a conformant TCP connection under the same conditions, and the VoIP connection is unable to reduce its sending rate on the wire, then the VoIP connection should terminate or suspend.

As discussed above, there are two reasons for requiring the application to terminate:

1) Avoiding congestion collapse, given the possibility of multiple congested links,

2) Fairness for congestion-controlled TCP traffic sharing the link.

In addition, if an application requires a minimum service level from the network in order to operate, and that service level is consistently not achieved, then the application should terminate or suspend sending.

One counter-argument is that users will just hang up anyway with a high packet drop rate so there is no point in enforcing a minimum acceptable rate. Users might hang up, but they might also just keep on talking, with the occasional noise getting though, for minutes or longer waiting for a short period of clarity. Another counter-argument is that nobody really benefits from VoIP connections being terminated or suspended when persistent packet drop rates exceed the allowable packet drop rate for the configured minimum sending rate. This is untrue, since the termination of these VoIP connections could allow competing TCP and VoIP traffic to make some progress.

In the next section, we illustrate the approach outlined above for VoIP flows with minimum sending rates of 4.75 and 64 kbps respectively, and show that in practice such an approach would not seem too burdensome for VoIP traffic. This approach implies that the VoIP traffic would terminate or suspend when the packet drop rate significantly exceeds 40% for a VoIP flow with a minimum sending rate of 4.75 kbps. If VoIP is to deliver "carrier quality" or even near "carrier quality" on best-effort links, conditioning deployment on the ability to maintain maximum sending rates during periods of persistent packet drops rates exceeding 40% does not suggest a service model that will see widespread acceptance among consumers, no matter what the price differential. Good packet throughput is vital for the delivery of acceptable VoIP service.

For a VoIP flow that stops sending because its minimum sending rate is too high for the steady-state packet drop rate, we have not addressed the question of when a VoIP flow might be able to start sending again, to see if the congestion on the end-to-end path has changed. This issue has been addressed in a proposal for Probabilistic Congestion Control [PCC].

We note that if the congestion indications are in the form of ECN-marked packets (Explicit Congestion Notification), as opposed to dropped packets, then the answers about when a flow with a minimum sending rate would have to stop sending are somewhat different. ECN allows routers to explicitly notify end-nodes of congestion by ECN-marking instead of dropping packets [RFC3168]. If packets are ECN-marked instead of dropped in the network, then there are no concerns of congestion collapse or of user quality (for the ECN-capable traffic, at any rate), and what remains are concerns of fairness with competing flows. Second, in regimes with very high congestion, TCP has a higher sending rate with ECN-marked than with dropped packets, in part because of different dynamics in terms of un-backing-off a backed-off retransmit timer.

5.1. Drop Rates at 4.75 kbps Minimum Sending Rate

Consider an adaptive audio application with an RTT of R=0.1 seconds that is able to adapt down to a minimum sending rate of 4.75 kbps application data, sending M=20 packets per second. This sending rate translates to N=593 Bps of application data, for a sending rate on the wire of B=1393 Bps. An equivalent-rate TCP connection with data packets of P=1460 bytes and a round-trip time of R=0.1 seconds would be sending BR/(P+40) = 0.09 ppr.

Table 1 in the Appendix looks at the packet drop rate experienced by a TCP connection with the RTO set to twice the RTT, and gives the corresponding sending rate of the TCP connection in ppr. The second column gives the sending rate estimated by the standard analytical approach, and the third, fourth, and fifth columns give the average sending rate from simulations with random packet drops or marks. The sixth column gives the sending rates from experiments on a 4.8-RELEASE FreeBSD machine. The analytical approaches require an RTO expressed as a multiple of the RTT, and Table 1 shows the results for the RTO set to 2 RTT. In the simulations, the minimum RTO is set to twice the RTT. See the Appendix for more details.

For a sending rate of 0.09 ppr and an RTO set to 2 RTT, Table 1 shows that the analytical approach gives a corresponding packet drop rate of roughly 50%, while the simulations in the fifth column and the experiments in the sixth column give a packet drop rate of between 35% and 40% to maintain a sending rate of 0.09 ppr. (For a reference TCP connection using timestamps, shown in the fourth column, the simulations give a packet drop rate of 55% to maintain a sending rate of 0.09 ppr.) Of the two approaches for determining TCP's relationship between the sending rate and the packet drop rate, the analytic approach and the use of simulations, we consider the simulations to be the most realistic, for reasons discussed in the Appendix. This suggests a packet drop rate of 40% would be reasonable for a TCP connection with an average sending rate of 0.09 ppr. As a result, a VoIP connection with an RTT of 0.1 sec and a minimum sending rate of 4.75 kbps would be required to terminate or suspend when the persistent packet drop rate significantly exceeds 40%.

These estimates are sensitive to the assumed round-trip time of the TCP connection. If we assumed instead that the equivalent-rate TCP connection had a round-trip time of R=0.01 seconds, the equivalent-rate TCP connection would be sending BR/(P+40) = 0.009 ppr. However, we have also assumed a minimum RTO for TCP connections of 0.1 seconds, which in this case would mean an RTO of at least 10 RTT. For this setting of the RTO, we would use Table 2 from the appendix to determine the average TCP sending rate for a particular packet drop rate. The simulations in the fifth column of Table 2 suggest that a TCP connection with an RTT of 0.01 sec and an RTO of 10 RTT would be able to send 0.009 ppr with a packet drop rate of 45%. (For the same TCP connection using timestamps, shown in the fourth column, the simulations give a packet drop rate of 60-65% to maintain a sending rate of 0.009 ppr.)

Thus, for a VoIP connection with an RTT of 0.01 sec and a minimum sending rate of 4.75 kbps, the VoIP connection would be required to terminate or suspend when the persistent packet drop rate exceeded 45%.

5.2. Drop Rates at 64 kbps Minimum Sending Rate

The effect of increasing the minimum acceptable sending rate to 64 kbps is effectively to decrease the packet drop rate at which the application should terminate or suspend sending. For this section, consider a codec with a minimum sending rate of 64 kbps, or N=8000 Bps, and a packet sending rate of M=50 pps. (This would be equivalent to 160-byte data packets, with 20 ms. per packet.) The sending rate on the wire is B = N+40M Bps, including headers, or 10000 Bps. A TCP connection having that sending rate, with packets of size P=1460 bytes and a round-trip time of R=0.1 seconds, sends BR/(P+40) = 0.66 ppr. From the fifth column of Table 1, for an RTO of 2 RTT, this corresponds to a packet drop rate between 20 and 25%. [For a TCP connection using fine-grained timestamps, as shown in the fourth column of Table 1, this sending rate corresponds to a packet drop rate between 25% and 35%.] As a result, a VoIP connection with an RTT of 0.1 sec and a minimum sending rate of 64 kbps would be required to terminate or suspend when the persistent packet drop rate significantly exceeds 25%.

For an equivalent-rate TCP connection with a round-trip time of R=0.01 seconds and a minimum RTO of 0.1 seconds (giving an RTO of 10 RTT), we use the fifth column of Table 2, which shows that a sending rate of 0.066 ppr corresponds to a packet drop rate of roughly 30%. [For a TCP connection using fine-grained timestamps, as shown in the fourth column of Table 2, this sending rate corresponds to a packet drop rate of roughly 45%.] Thus, for a VoIP connection with an RTT of 0.01 sec and a minimum sending rate of 64 kbps, the VoIP connection would be required to terminate or suspend when the persistent packet drop rate exceeded 30%.

5.3. Open Issues

This document does not attempt to specify a complete protocol. For example, this document does not specify the definition of a persistent packet drop rate. The assumption would be that a "persistent packet drop rate" would refer to the packet drop rate over a significant number of round-trip times, e.g., at least five seconds. Another possibility would be that the time interval for measuring the persistent drop rate is a function of the lifetime of the connection, with longer-lived connections using longer time intervals for measuring the persistent drop rate.

The time period for detecting persistent congestion also affects the potential synchronization of VoIP sessions all terminating or suspending at the same time in response to shared congestion. If flows use some randomization in setting the time interval for detecting persistent congestion, or use a time interval that is a function of the connection lifetime, this could help to prevent all VoIP flows from terminating at the same time.

Another design issue for a complete protocol concerns whether a flow terminates when the packet drop rate is too high, or only suspends temporarily. For a flow that suspends temporarily, there is an issue of how long it should wait before resuming transmission. At the very least, the sender should wait long enough so that the flow's overall sending rate doesn't exceed the allowed sending rate for that packet drop rate.

The recommendation of this document is that VoIP flows with minimum sending rates should have corresponding configured packet drop rates, such that the flow terminates or suspends when the persistent packet drop rate of the flow exceeds the configured rate. If the persistent packet drop rate increases over time, flows with higher minimum sending rates would have to suspend sending before flows with lower minimum sending rates. If VoIP flows terminate when the persistent packet drop rate is too high, this could lead to scenarios where VoIP flows with lower minimum sending rates essentially receive all of the link bandwidth, while the VoIP flows with higher minimum sending rates are required to terminate. However, if VoIP flows suspend sending for a time when the persistent packet drop rate is too high, instead of terminating entirely, then the bandwidth could end up being shared reasonably fairly between VoIP flows with different minimum sending rates.

5.4. A Simple Heuristic

One simple heuristic for estimating congestion would be to use the RTCP reported loss rate as an indicator. For example, if the RTCP-reported lost rate is greater than 30%, or N back-to-back RTCP reports are missing, the application could assume that the network is too congested, and terminate or suspend sending.

6. Constraints on VoIP Systems

Ultimately, attempting to run VoIP on congested links, even with adaptive rate codecs and minimum packet rates, is likely to run into hard constraints due to the nature of real time traffic in heavily congested scenarios. VoIP systems exhibit a limited ability to scale their packet rate. If the number of packets decreases, the amount of audio per packet is greater and error concealment at the receiver becomes harder. Any error longer than phoneme length, which is typically 40 to 100 ms depending on the phoneme and speaker, is unrecoverable. Ideally, applications want sub 30ms packets and this is what most voice codecs provide. In addition, voice media streams exhibit greater loss sensitivity at lower data rates. Lower-data rate codecs maintain more end-to-end state and as a result are generally more sensitive to loss.

We note that very-low-bit-rate codecs have proved useful, although with some performance degradation, in very low bandwidth, high noise environments (e.g., 2.4 kbps HF radio). For example, 2.4 kbps codecs "produce speech which although intelligible is far from natural sounding" [W98]. Figure 5 of [W98] shows how the speech quality with several forms of codecs varies with the bit rate of the codec.

7. Conclusions and Recommendations

In the near term, VoIP services are likely to be deployed, at least in part, over broadband best-effort connections. Current real time media encoding and transmission practice ignores congestion considerations, resulting in the potential for trouble should VoIP become a broadly deployed service in the near to intermediate term. Poor user quality, unfairness to other VoIP and TCP users, and the possibility of sporadic episodes of congestion collapse are some of the potential problems in this scenario.

These problems can be mitigated in applications that use fixed-rate codecs by requiring the best-effort VoIP application to specify its minimum bit throughput rate. This minimum bit rate can be used to estimate a packet drop rate at which the application would terminate.

This document specifically recommends the following:

(1) In IETF standards for protocols regarding best-effort flows with a minimum sending rate, a packet drop rate must be specified, such that the best-effort flow terminates, or suspends sending temporarily, when the steady-state packet drop rate significantly exceeds the specified drop rate.

(2) The specified drop rate for the minimum sending rate should be consistent with the use of Tables 1 and 2 as illustrated in this document.

We note that this is a recommendation to the IETF community, as a specific follow-up to RFC 2914 on Congestion Control Principles.

This is not a specific or complete protocol specification.

Codecs that are able to vary their bit rate depending on estimates of congestion can be even more effective in providing good quality service while maintaining network efficiency under high load conditions. Adaptive variable-bit-rate codecs are therefore preferable as a means of supporting VOIP sessions on shared use Internet environments.

Real-time traffic such as VoIP could derive significant benefits from the use of ECN, where routers may indicate congestion to end-nodes by marking packets instead of dropping them. However, ECN is only standardized to be used with transport protocols that react appropriately to marked packets as indications of congestion. VoIP traffic that follows the recommendations in this document could satisfy the congestion-control requirements for using ECN, while VoIP traffic with no mechanism for terminating or suspending when the packet dropping and marking rate was too high would not. However, we repeat that this document is not a complete protocol specification. In particular, additional mechanisms would be required before it was safe for applications running over UDP to use ECN. For example, before using ECN, the sending application would have to ensure that the receiving application was capable of receiving ECN-related information from the lower-layer UDP stack, and of interpreting this ECN information as a congestion indication.

8. Acknowledgements

We thank Brian Adamson, Ran Atkinson, Fred Baker, Jon Crowcroft, Christophe Diot, Alan Duric, Jeremy George, Mark Handley, Orion Hodson, Geoff Huston, Eddie Kohler, Simon Leinen, David Meyer, Jean-Francois Mule, Colin Perkins, Jon Peterson, Mike Pierce, Cyrus Shaoul, and Henning Schulzrinne for feedback on this document. (Of course, these people do not necessarily agree with all of the document.) Ran Atkinson and Geoff Huston contributed to the text of the document.

The analysis in Section 6.0 resulted from a session at the whiteboard with Mark Handley. We also thank Alberto Medina for the FreeBSD experiments showing TCP's sending rate as a function of the packet drop rate.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

[RFC3267] Sjoberg, J., Westerlund, M., Lakaniemi, A. and Q. Xie, "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 3267, June 2002.

9.2. Informative References

[A02] Ran Atkinson, An ISP Reality Check, Presentation to ieprep, 55th IETF Meeting, November 2002. URL "http://www.ietf.cnri.reston.va.us/proceedings/02nov/219.htm#slides".

[A03] Brian Adamson, private communication, June 2003.

[BBFS01] Deepak Bansal, Hari Balakrishnan, Sally Floyd, and Scott Shenker, Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms, SIGCOMM 2001.

[COPS] Durham, D., Ed., Boyle, J., Cohen, R., Herzog, S., Rajan, R. and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[DCCP03] Eddie Kohler, Mark Handley, Sally Floyd, and Jitendra Padhye, Datagram Congestion Control Protocol (DCCP), internet-draft Work in Progress, March 2003. URL "http://www.icir.org/kohler/dcp/".

[DIFFSERV] Differentiated Services (diffserv), Concluded Working Group, URL "http://www.ietf.cnri.reston.va.us/html.charters/OLD/diffserv-charter.html".

[E2E] The end2end-interest mailing list, URL "http://www.postel.org/mailman/listinfo/end2end-interest".

[FHPW00] S. Floyd, M. Handley, J. Padhye, J. Widmer, "Equation-Based Congestion Control for Unicast Applications", ACM SIGCOMM 2000.

[FM03] S. Floyd and R. Mahajan, Router Primitives for Protection Against High-Bandwidth Flows and Aggregates, internet draft (not yet submitted).

[FWD] Free World Dialup, URL "www.pulver.com/fwd/".

[IEPREP02] Internet Emergency Preparedness (ieprep), Minutes, 55th IETF Meeting, November 2002. URL "http://www.ietf.cnri.reston.va.us/proceedings/02nov/219.htm#cmr".

[ILBRC] S.V. Andersen, et. al., Internet Low Bit Rate Codec, Work in Progress, March 2003.

[G.114] Recommendation G.114 - One-way Transmission Time, ITU, May 2003. URL "http://www.itu.int/itudoc/itu-t/aap/sg12aap/recaap/g.114/".

[IVOX] The Interactive VOice eXchange, URL "http://manimac.itd.nrl.navy.mil/IVOX/".

[Jacobson88] V. Jacobson, Congestion Avoidance and Control, ACM SIGCOMM '88, August 1988.

[AUT] The maximum feasible drop rate for VoIP traffic depends on the codec. These numbers are a range for a variety of codecs; voice quality begins to deteriorate for many codecs around a 10% drop rate. Note from authors.

[JS00] Wenyu Jiang and Henning Schulzrinne, Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality, NOSSDAV, 2000. URL "http://citeseer.nj.nec.com/jiang00modeling.html".

[JS02] Wenyu Jiang and Henning Schulzrinne, Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss, NOSSDAV, 2002. URL "http://www1.cs.columbia.edu/~wenyu/".

[JS03] Wenyu Jiang, Kazummi Koguchi, and Henning Schulzrinne, QoS Evaluation of VoIP End-points, ICC 2003. URL "http://www1.cs.columbia.edu/~wenyu/".

[KFK79] G.S. Kang, L.J. Fransen, and E.L. Kline, "Multirate Processor (MRP) for Digital Voice Communications", NRL Report 8295, Naval Research Laboratory, Washington DC, March 1979.

[KF82] G.S. Kang and L.J. Fransen, "Second Report of the Multirate Processor (MRP) for Digital Voice Communications", NRL Report 8614, Naval Research Laboratory, Washington DC, September 1982.

[Measurement] Web page on "Measurement Studies of End-to-End Congestion Control in the Internet", URL "http://www.icir.org/floyd/ccmeasure.html". The section on "Network Measurements at Specific Sites" includes measurement data about the distribution of packet sizes on various links in the Internet.

[MTK03] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam, "Assessing the Quality of Voice Communications Over Internet Backbones", IEEE/ACM Transactions on Networking, V. 11 N. 5, October 2003.

[NSIS] Next Steps in Signaling (nsis), IETF Working Group, URL "http://www.ietf.cnri.reston.va.us/html.charters/nsis-charter.html".

[PCC] Joerg Widmer, Martin Mauve, and Jan Peter Damm. Probabilistic Congestion Control for Non-Adaptable Flows. Technical Report 3/2001, Department of Mathematics and Computer Science, University of Mannheim. URL "http://www.informatik.uni-mannheim.de/informatik/pi4/projects/CongCtrl/pcc/index.html".

[PFTK98] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP Throughput: A Simple Model and its Empirical Validation, Tech Report TF 98-008, U. Mass, February 1998.

[RFC896] Nagle, J., "Congestion Control in IP/TCP", RFC 896, January 1984.

[RFC1890] Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 1890, January 1996.

[RFC2474] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2581] Allman, M., Paxson, V. and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

[RFC2597] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group, RFC 2597, June 1999.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC2990] Huston, G., "Next Steps for the IP QoS Architecture", RFC 2990, November 2000.

[RFC3042] Allman, M., Balakrishnan, H. and S., Floyd, "Enhancing TCP's Loss Recovery Using Limited Transmit", RFC 3042, January 2001.

[RFC3168] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3246] Davie, B., Charny, A., Bennet, J.C.R., Benson, K., Le Boudec, J.Y., Courtney, W., Davari, S., Firoiu, V. and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[RFC3448] Handley, M., Floyd, S., Pahdye, J. and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

[RSVP] Resource Reservation Setup Protocol (rsvp), Concluded Working Group, URL "http://www.ietf.cnri.reston.va.us/html.charters/OLD/rsvp-charter.html".

[RTTWeb] Web Page on Round-Trip Times in the Internet, URL "http://www.icir.org/floyd/rtt-questions.html"

[S03] H. Schulzrinne, private communication, 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 3551, July 2003.

[Vonage]   Vonage, URL "www.vonage.com".

[W98]   J. Woodward, Speech Coding, Communications Research Group, University of Southampton, 1998. URL "http://www-mobile.ecs.soton.ac.uk/speech_codecs/",

10. Appendix - Sending Rates with Packet Drops

The standard way to estimate TCP's average sending rate S in packets per round-trip as a function of the packet drop rate would be to use the TCP response function estimated in [PFTK98]:

$$S = 1/(\mathrm{sqrt}(2p/3) + K \min(1,3\ \mathrm{sqrt}(3p/8))\ p\ (1 + 32\ p^2)) \qquad (1)$$

for acks sent for every data packet, and the RTO set to K*RTT.

The results from Equation (1) are given in the second column in Tables 1 and 2 below. However, Equation (1) overestimates TCP's sending rate in the regime with heavy packet drop rates (e.g., of 30% or more). The analysis behind Equation (1) assumes that once a single packet is successfully transmitted, TCP's retransmit timer is no longer backed-off. This might be appropriate for an environment with ECN, or for a TCP connection using fine-grained timestamps, but this is not necessarily the case for a non-ECN-capable TCP connection without timestamps. As specified in [RFC2988], if TCP's retransmit timer is backed-off, this back-off should only be removed when TCP successfully transmits a new packet (as opposed to a retransmitted packet), in the absence of timestamps.

When the packet drop rate is 50% or higher, for example, many of the successful packet transmissions can be of retransmitted packets, and the retransmit timer can remain backed-off for significant periods of time, in the absence of timestamps. In this case, TCP's throughput is determined largely by the maximum backoff of the retransmit timer. For example, in the NS simulator the maximum backoff of the retransmit timer is 64 times the un-backed-off value. RFC 2988 specifies that "a maximum value MAY be placed on RTO provided it is at least 60 seconds." [Although TCP implementations vary, many TCP implementations have a maximum of 45 seconds for the backed-off RTO after dropped SYN packets.]

Another limitation of Equation (1) is that it models Reno TCP, and therefore underestimates the sending rate of a modern TCP connection that used SACK and Limited Transmit.

The table below shows estimates of the average sending rate S in packets per RTT, for TCP connections with the RTO set to 2 RTT for Equation (1).

These estimates are compared with simulations in the third, fourth, and fifth columns, with ECN, packet drops for TCP with fine-grained timestamps, and packet drops for TCP without timestamps respectively. (The simulation scripts are available from http://www.icir.org/floyd/VoIP/sims.) Each simulation computes the average sending rate over the second half of a 10,000-second simulation, and for each packet drop rate, the average is given over 50 simulations. For the simulations with very high packet drop rates, it is sometimes the case that the SYN packet is repeatedly dropped, and the TCP sender never successfully transmits a packet. In this case, the TCP sender also never gets a measurement of the round-trip time.

The sixth column of Table 1 shows the average sending rate S in packets per RTT for an experiment using a 4.8-RELEASE FreeBSD machine. For the low packet drop rates of 0.1 and 0.2, the sending rate in the simulations is higher than the sending rate in the experiments; this is probably because the TCP implementation in the simulations uses Limited Transmit [RFC3042]. With Limited Transmit, the TCP sender can sometimes avoid a retransmit timeout when a packet is dropped and the congestion window is small. With high packet drop rates of 0.65 and 0.7, the sending rate in the simulations is somewhat lower than the sending rate in the experiments. For these high packet drop rates, the TCP connections in the experiments would often abort prematurely, after a sufficient number of successive packet drops.

We note that if the ECN marking rate exceeds a locally-configured threshold, then a router is advised to switch from marking to dropping. As a result, we do not expect to see high steady-state marking rates in the Internet, even if ECN is in fact deployed.

Drop					
Rate p	Eq(1)	Sims:ECN	Sims:TimeStamp	Sims:Drops	Experiments
0.1	2.42	2.92	2.38	2.32	0.72
0.2	.89	1.82	1.26	0.82	0.29
0.25	.55	1.52	.94	.44	0.22
0.35	.23	.99	.51	.11	0.10
0.4	.16	.75	.36	.054	0.068
0.45	.11	.55	.24	.029	0.050
0.5	.10	.37	.16	.018	0.036
0.55	.060	.25	.10	.011	0.024
0.6	.045	.15	.057	.0068	0.006
0.65	.051	.	.033	.0034	0.008
0.7	.041	.06	.018	.0022	0.007
0.75	.034	.04	.0099	.0011	
0p.8	.028	.027	.0052	.00072	
0.85	.023	.015	.0021	.00034	
0.9	.020	.011	.0011	.00010	
0.95	.017	.0079	.00021	.000037	

Table 1: Sending Rate S as a Function of the Packet Drop Rate p, for RTO set to 2 RTT, and S in packets per RTT.

The table below shows the average sending rate S, for TCP connections with the RTO set to 10 RTT.

Drop Rate p	Eq(1)	Sims:ECN	Sims:TimeStamp	Sims:Drops
0.1	0.97	2.92	1.67	1.64
0.2	0.23	1.82	.56	.31
0.25	0.13	.88	.36	.13
0.3	0.08	.61	.23	.059
0.35	0.056	.41	.15	.029
0.4	0.040	.28	.094	.014
0.45	0.029	.18	.061	.0080
0.5	0.021	.11	.038	.0053
0.55	0.016	.077	.022	.0030
0.6	0.013	.045	.013	.0018
0.65	0.010	.	.0082	.0013
0.7	0.0085	.018	.0042	
0.75	0.0069	.012	.0025	.00071
0.8	0.0057	.0082	.0014	.00030
0.85	0.0046	.0047	.00057	.00014
0.9	0.0041	.0034	.00026	.000025
0.95	0.0035	.0024	.000074	.000013

Table 2: Sending Rate as a Function of the Packet Drop Rate, for RTO set to 10 RTT, and S in packets per RTT.

11. Security Considerations

This document does not itself create any new security issues for the Internet community.

12. IANA Considerations

There are no IANA considerations regarding this document.

13. Authors' Addresses

Internet Architecture Board
EMail: iab@iab.org

Internet Architecture Board Members
at the time this document was published were:

Bernard Aboba
Harald Alvestrand (IETF chair)
Rob Austein
Leslie Daigle (IAB chair)
Patrik Faltstrom
Sally Floyd
Jun-ichiro Itojun Hagino
Mark Handley
Geoff Huston (IAB Executive Director)
Charlie Kaufman
James Kempf
Eric Rescorla
Mike St. Johns

This document was created in January 2004.

14. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.