Internet Architecture Board (IAB)
Request for Comments: 7295
Category: Informational
ISSN: 2070-1721

H. Tschofenig
L. Eggert
Z. Sarker

July 2014

Report from the IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication

Abstract

This document provides a summary of the IAB/IRTF Workshop on 'Congestion Control for Interactive Real-Time Communication', which took place in Vancouver, Canada, on July 28, 2012. The main goal of the workshop was to foster a discussion on congestion control mechanisms for interactive real-time communication. This report summarizes the discussions and lists recommendations to the Internet Engineering Task Force (IETF) community.

The views and positions in this report are those of the workshop participants and do not necessarily reflect the views and positions of the authors, the Internet Architecture Board (IAB), or the Internet Research Task Force (IRTF).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Architecture Board (IAB) and represents information that the IAB has deemed valuable to provide for permanent record. It represents the consensus of the Internet Architecture Board (IAB). Documents approved for publication by the IAB are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7295.

Table of Contents

1. Introduction

The Internet Architecture Board (IAB) holds occasional workshops designed to consider long-term issues and strategies for the Internet, and to suggest future directions for the Internet architecture. This long-term planning function of the IAB is complementary to the ongoing engineering efforts performed by working groups of the Internet Engineering Task Force (IETF), under the leadership of the Internet Engineering Steering Group (IESG) and area directorates.

Any application that sends significant amounts of data over the Internet is expected to implement reasonable congestion control behavior. The goals for congestion control are well understood and documented in RFC 2914 [2] and RFC 5405 [1]:

1. Preventing congestion collapse.

2. Allowing multiple flows to share the network fairly.

The Internet has been used for interactive real-time communication for decades, most of which is being transmitted using the Real-Time Transport Protocol (RTP) over UDP, often over provisioned capacity and/or using only rudimentary congestion control mechanisms. In 2004, the IAB raised concerns regarding possibilities of a congestion collapse due to a rapid growth in real-time voice traffic that does not practice end-to-end congestion control [17]. That congestion collapse did not happen, but concerns raised about both congestion collapse and fairness are still valid and have gained more relevance when applied to more bandwidth-hungry video applications. The development and upcoming widespread deployment of web-based real-time media communication -- where RTP is used to and from web browsers to transmit audio, video, and data -- will likely result in substantial new Internet traffic. Due to the projected volume of this traffic, as well as the fact that it is more likely to use unprovisioned capacity, it is essential that it is transmitted with robust and effective congestion control mechanisms.

Designing congestion control mechanisms that perform well under a wide variety of traffic mixes and over network paths with widely varying characteristics is not easy. Prevention of congestion collapse can be achieved through a "circuit breaker" mechanism (see, for example, [3]), but for media flows that are supposed to coexist with a user’s other ongoing communication sessions, a congestion control mechanism that shares capacity fairly in the presence of a mix of TCP, UDP, and other protocol flows is needed.

Many additional complications arise. Here are some examples:

1. Real-time interactive media sessions require low latencies, whereas streaming media can use large play-out buffers.

2. In an RTP session, feedback exchanged via the RTP Control Protocol (RTCP) typically arrives much less frequently than, for example, TCP ACKs for a given TCP connection. Theoretically, the RTP/RTCP control loop can lead to a longer reaction time.

3. Media codecs can usually only adjust their output rates in a much more coarse-grained fashion than, for example, TCP, and user experience suffers if encoding rates are switched too frequently. Codecs typically have a minimum sending rate as well.

4. Some bits of an encoded media stream are more important than others. For example, losing or dropping an I-frame of a video stream is more problematic than dropping a P-frame [40].

5. Ramping up the transmission rate can be problematic. Simply increasing the output rate of the codec without knowing whether the network path can sustain transmission at the increased rate runs the danger of incurring a significant amount of packet loss that can cause playback artifacts.

6. A congestion control scheme for interactive media needs to handle bundles of interrelated flows (audio, video, and data) in a way that accommodates the preferences of the application in the event of congestion.

7. The desire to provide a congestion control mechanism that can be efficiently implemented inside an application imposes additional restrictions. For example, a web browser is not able to take the protocol interactions of a software download happening in another application into account.

8. There are explicit congestion signals (such as Explicit Congestion Notification (ECN) [19]), and there are implicit indications of congestion (e.g., packet delay and loss). Care must be taken to account for each of these signals, particularly if various applications react on the same set of signals.

9. Large buffers are often used in network elements and end device operating systems to better support TCP-based applications. These buffers introduce additional communication delay, which harms the small delay budget available for interactive real-time applications.

2. Workshop Structure

The IETF has a long history of work on congestion control mechanisms. With ongoing standardization work on real-time interactive media communication on the web, new challenges have emerged that have refocused engineering attention on congestion control issues. To take a deeper look at congestion control in light of the growth of real-time traffic, workshop participants were invited to submit position papers that were then used to organize the workshop agenda into three principal components: a keynote talk given by Mark Handley describing the history of the work on congestion control for real-time media followed and his views of current problems; a presentation of simulations and data demonstrating current problems and solutions; and a discussion of desirable solution properties and challenges in deploying solutions.

2.1. History and Current Challenges

Mark Handley argued that since 1988, the Internet has remained functional despite exponential growth, routers that are sometimes buggy or misconfigured, rapidly changing applications and usage patterns, and flash crowds. This is largely because most applications use TCP, and TCP implements end-to-end congestion control.

TCP’s congestion control adapts the window to fit the capacity available in the network and accomplishes approximate fairness between two competing flows over a period of time. Mark indicated that the provided level of fairness is not necessarily what we want: The 1/round-trip-time relationship in TCP is not ideal since it means that network operators can decide to lower packet loss by adding bigger buffers (which unfortunately leads to bufferbloat problems; see [31] and [39]). The 1/sqrt(packet drop rate) relationship is also not necessarily desirable since TCP initially did not work particularly well for high-speed flows (which had been the subject of much TCP research).

TCP controls the congestion window in bytes. For bulk transfer, usually this results in controlling the number of 1500-byte packets sent per second. Real-time media is different since it has its own time constraints. For audio, one wants to send one packet per 20 ms and for video, the ideal value would be 25 to 30 frames per second. One, therefore, wants to avoid additional sending delay.

As an example, in case of video, to relieve congestion one has to reduce the number of packets-per-second transmission rate rather than transmit smaller packets, since at higher bitrates on WiFi the time it takes to send a packet is almost negligible compared to the time

that is spent with Media Access Control (MAC) layer operations. Reducing the packet size makes little difference to the available capacity. For a serial line, it does not matter how big the packets are.

From a network point of view, the goals of congestion control therefore are:

1. Avoid congestion collapse

2. Avoid starvation of TCP flows

3. Avoid starvation of real-time flows, specifically in the case where TCP and real-time flows share the same FIFO queue.

From an application point of view, the goals of congestion control are different, namely:

1. Robust behavior. One wants to have a good throughput when the network is working well and passable performance when the network is working poorly.

2. Predictable behavior. This matters from a usability point of view since variable media creates a bad user experience.

3. Low latency. With large buffers along the end-to-end path, latency will increase when interactive real-time flows compete with TCP flows. This results in TCP filling up the buffers; increased buffering will lead to additional delays for the delivery of the interactive real-time media.

Attempts to provide congestion control for interactive real-time media have previously been made in the IETF, for example, with the work on TCP Friendly Rate Control (TFRC) [12]. TFRC illustrates the challenges quite well. TFRC tries to accomplish the same throughput as TCP, but with a smoother transmission rate. It measures the loss and the round-trip time but follows a similar model as TCP to determine the sending rate.

In a link with low statistical multiplexing, TCP can lead to bad oscillations. The sending rate hits the maximum rate of a bottleneck link, a lot of loss occurs, and then the sending rate peaks again. For very small buffers the result is acceptable, but bigger buffers lead to oscillations. The result is bad for networks and for applications. To deal with large buffers on these links, a short-term rate adaptation based on round-trip time (RTT) information is utilized in TRFC, but this requires good short-term RTT measurements.

TRFC works pretty well in theory. TFRC assumes the network is in charge of the codec and that the codec can produce data at the demanded rate. Modern video codecs inherently produce variable-bitrate video streams based on the content being encoded, and it is hard to produce data at exactly the desired bitrate without excessive buffering or ugly quality changes.

What if the codec is put in charge instead of the network? The network tells the codec the mean rate, but it does not worry about what happens in short time scales, and the codec matches the mean rate and does not worry whether it is over or under the rate for a relatively short time scale. This again leads to the low statistical multiplexing problem and leads to oscillations.

Known congestion control mechanisms work well if they can respond quickly enough to changes and if they do not bump into the low statistical multiplexing problem.

To avoid the low statistical multiplexing problem, techniques for inferring link speed are needed. The work from Van Jacobson's pathchar [37] (and successors) serve as valuable input. The idea is to send short packet trains, to measure timing accurately, and to infer the link speed from the relative delay. If we know the link speed, we can avoid exceeding it. Congestion control can give us an approximate rate, but we must not exceed link speed. This is a hybrid between codec being in charge (most of the time) and the network being in charge. These work well for some links, but not for others. Wireless links where speed can change in less than a single RTT because of fading, bitrate adaption, etc., cause problems. We would like to have the codec and the network be in charge. However, they both cannot be in charge at the same time.

Mark indicated that he is not entirely sure whether RTCP is suitable for congestion control. RTCP gives feedback, but it cannot send it often enough to avoid bumping into link speed. Circuit breakers [3], on the other hand, do not help to give good performance on an uncongested path. With circuit breakers, the sender measures the loss rate and RTT, and runs with a loose "cap."

In conclusion, Mark Handley claimed that we know how to do good congestion control, but only if congestion control is in charge, and that's not acceptable for real-time applications. We only know how to do good congestion control if we change the packet/sec rate and not the packet size.

2.2. Simulations and Measurements

This second part of the workshop was focused on the presentation and the discussion of data gathered from simulations and real-world measurements.

Keith Winstein started the discussion with his presentation of measurements performed in cellular operator networks in the US [22]. The measurements indicate that the analyzed cellular networks showed varying RTT with transient latency spikes to hundreds of milliseconds, link speed that varies by a factor of 10 in a short time scale, and buffers that do not drop packets until they contain 5-10 seconds of data at bottleneck link speed.

Zaheduzzaman Sarker [21] presented results from real-time video communication in a Long Term Evolution (LTE) simulator utilizing ECN-based packet marking and adaptation using implicit methods like packet loss and delay. ECN marking provides ways for the network to explicitly signal congestion and hence distributes the cost of congestion well and helps achieve lower latency. However, although RFC 3168 [19] was finalized in 2001, the deployment of ECN is still lacking as investigated by Bauer, et al. [25]. A few participants noted that they believe that the deployment of LTE networks will also increase the deployment of ECN with the recent work on ECN for RTP over UDP [11].

Mo Zahaty [20] discussed TFRC [12] and TFRC with weighted fairness (MulTFRC) [4], which tunes TFRC to consider multiple flows, and showed the impact of RTT and loss rates on the type of video quality that can be achieved under those conditions. TFRC requires frequent feedback, which RTCP does not provide even when considering the extended RTP profile for RTCP-based feedback (RFC 4585 [5]). Mo argued that application-specified weighted fairness is important but while MulTFRC provides better performance than TFRC, it is not clear whether the added complexity over an n-times-TFRC approach is indeed worth the effort.

Markku Kojo shared analysis results of how real-time audio is affected by competing TCP flows. In the experiments shown in Figure 2 of [27], a real-time interactive audio stream had to compete against one TCP flow and, as a comparison, against six TCP flows. With one concurrent TCP flow, voice is impacted on startup and six TCP flows destroy the quality of the call. Two types of losses were analyzed, namely losses that result from a packet being dropped in the network (e.g., due to congestion or link errors) and losses that result from the delayed arrival of the packet (due to buffering) when the audio packet misses the deadline for the codec to decode and play the transmitted content. Consequently, even a moderate number of TCP

flows typically used by browsers to retrieve content on web pages in parallel causes irreparable harm for audio transfers. The size of the initial window (IW) also impacts interactive real-time communication since a larger TCP IW size (e.g., IW10 with ten segments, as proposed in [18], instead of three) leads to a bigger burst of packets because of the initial window transmission. Note that the study in [24] does not necessarily lead to the same conclusion. It claims that the increased initial window size leads to no impact or only modest impact for buffering in the majority of cases.

Cullen Jennings [28] presented measurement results showing interactions between RTP and TCP flows for several widely deployed video communication products: Apple FaceTime, Google Hangout, Cisco Movi, and Microsoft Skype. While all tested products implemented some form of congestion control, none of the applications did additive increase and multiplicative decrease (AIMD). In general, it was observable that video adapts more slowly than AIMD to changes in available bandwidth because most codecs cannot make small increases in sending rates when available bandwidth increases, and do not make large decreases in sending rates when available bandwidth decreases, in order to improve the user’s experience.

Stefan Holmer [43] investigated the difference between loss-based and delay-based congestion control algorithms. The suitability of loss-based congestion control schemes for interactive real-time communication systems heavily depends on buffer sizes and the deployment of active queue management mechanisms. If most routers are using tail-drop queuing, then loss-based congestion control cannot fulfill the requirements of interactive real-time applications since those flows will effectively increase the bitrate until a loss event is identified, which only happens when the bottleneck queue is full.

2.3. Design Aspects of Problems and Solutions

During the remaining part of the workshop, the participants discussed design aspects of both the problem and solution spaces. The discussions started with a presentation by Jim Gettys about problems related to bufferbloat [31][36]. Bufferbloat is "a phenomenon in packet-switched networks, in which excess buffering of packets causes high latency and packet delay variation (also known as jitter), as well as reducing the overall network throughput" [39]. A certain amount of buffering is helpful to improve the efficiency. Not dropping packets in the event of congestion leads to increasing delays for interactive real-time communication.

Packets may get buffered at various places along the end-to-end path including in the operating system/device drivers, customer premise equipment (such as cable modem and DSL routers), base stations, and routers. While the understanding of too large buffers has improved over the last few years, workshop participants were still concerned that many equipment manufacturers and network operators do not yet acknowledge the existence of the problem. This lack of understanding is caused by the strong focus on throughput network performance measurements that do not take latency into account. For example, only recently the Federal Communications Commission (FCC) has added latency tests to their test suites [41].

Active queue management (AQM) aims to prevent queues from growing too large. This is accomplished by monitoring queue length and informing the sender by dropping or marking packets to lower their transmission rate. Random Early Detection (RED) [9] is one such AQM algorithm, but it has not been widely deployed in routers largely because of challenges to configure it correctly [32]. According to [23], RED does not work with the default settings as it is "too "gentle" to handle fast changes due to TCP slow start, when the aggregate traffic is limited." There may also be a lack of incentives to deploy AQM algorithms. Participants speculated about the time it takes to update network equipment (to support AQM algorithms), considering the different replacement cycles of these devices.

One outcome of that discussion on AQM at the workshop was a Birds of a Feather ("BoF") meeting on "Active Queue Management and Packet Scheduling" at IETF 87 (July 28 - August 5, 2013, Berlin, Germany). The AQM WG [35] was chartered a few weeks later and is now designing AQM and network infrastructure improvements to deal with bufferbloat and related issues.

Measurement tools that allow an end user to determine the performance of his or her network, including latency, is seen as a promising approach to motivate network operators to upgrade their equipment and to make use of AQM algorithms. Measurement tools would allow users to determine how bad their networks perform and to complain to their ISP, thereby creating a market force. As to what the right performance measurement metrics are, it was noted that the intent of the IETF IP Performance Metrics (IPPM) working group [33] was to develop such metrics to qualify networks. That work may have begun before its time, but there have been recent attempts to revisit the measurement work and an effort by the FCC has gotten a lot of attention recently (see [7] and [42]).

Matt Mathis and others argued that the traffic of throughput-maximizing and delay-minimizing applications need to be in separate queues (segregated queuing). Requiring segregated queues assumes you

are sharing the network with other greedy traffic. Quality-of-Service (QoS) signaling is a way to deploy segregated queuing, but there are several simpler alternatives, such as Stochastic Fair Queuing [38]. The Controlled Delay (CoDel) AQM algorithm [6] can also be used in combination with stochastic fair queuing. Note that queue segregation is not necessary for every router to implement; using it at the edge of a network where bottleneck links are located is already sufficient.

It was noted that current interactive voice usage over the Internet works most of the time satisfactorily. In typical networks, the reason voice works is because networks are underloaded. As long as there is idle capacity and the queue is empty when packets arrive, traffic does not need to be separated into distinct queues. Further explanations were offered as to why many networks work surprisingly well: Low Extra Delay Background Transport (LEDBAT) [8] is used for the download of software updates, voice traffic contributes only a small percentage of the overall Internet traffic, and users employ "human protocols" (e.g., parents asking their kids to get off the network during the time of a conference call).

Cullen Jennings raised a concern that although interactive voice may be functional without a congestion control mechanism, the potentially large uptake of interactive video spurred on by Real-Time Communications on the Web (RTCWEB) could create substantially more significant problems. In the class of space where voice is currently working, video may fail. Ted Hardie countered by saying that RTCWEB is trying to replace existing proprietary technologies. It may ramp up the amount of use we are expecting, but it is not doing much that was not being done by Adobe Flash or Skype. RTCWEB is not a totally novel context of Internet usage. Magnus Westerlund added that RTCWEB might be the driver for the moment, but web browsers are not the only consumers of such congestion control algorithm.

Furthermore, Ted Hardie noted that applications will not produce media streams that grow to 10 Mbps because their sending rate is auto rate limited by the production of the video. He suggested to ask ourselves if we are trying to get TCP to be friendly to media streams that are already rate limited or if we are asking media streams that are already rate limited to be TCP friendly. To quote Andrew McGregor: "It’s really not good to be TCP friendly because it’s not going to return the favor." If the desired properties we want are no starvation, fairness, and effective goodput for the offered loads, are we only willing to consider changes in RTP control, or are we willing to consider changes in TCP congestion control?

This led to a discussion about whether the development of a congestion control algorithm for interactive real-time applications provides any value if network equipment suffers from bufferbloat. Is there something that can be done today to help interactive real-time media or do we have to wait to get the network updated first? Replacing home routers and updating routers with modern AQM algorithms was seen as a longer-term effort. Also, the time scale for changing TCP's congestion control is on the same time scale as deploying ECN [19]. Colin Perkins noted that we cannot change TCP quickly; the way TCP is being used is changing quickly, and we can impact the way TCP is used. When TCP is used for file transfer, it will send data as fast as it can, but when TCP is used for WebSockets, the dynamics are different. WebSockets and SPDY are clearly changing the behavior of TCP. Also, Netflix-style video-streaming applications are huge users of TCP and those applications can change rather quickly. Matt Mathis added that real-time videoconferencing almost always produces video streams at a lower bitrate than downloading equivalent-sized stored video using best-effort file-sharing.

Bill Ver Steeg suggested to consider three different deployment environments, namely:

1. Flows competing with flows from the host ("self-inflicted queuing delay")

2. Flows competing with flows in the same subnetwork (e.g., home network)

3. Flows competing with flows from other networks (e.g., traffic from different households that utilize the same DSL provider)

The narrowest problem domain that makes sense is to avoid self-inflicted queuing delay. Michael Welzl indicated that this requires an information exchange (called flow-state exchange) inside a browser (at the level of the same host or even beyond, as described in [29]) to synchronize congestion control of different audio, video, and data flows. Although it would provide great benefits if one could share information about a bottleneck with all the flows sharing that bottleneck, this is considered challenging even within a single host. John Leslie [30] also noted: "We're acting as if we believe congestion will magically be solved by a new transport algorithm. It won't." Instead, an interaction between the network layer, transport layer, and the application layer is needed whereby the application layer is the only practical place to balance what piece(s) to constrain to lower bandwidths. All flows relating to a user session

should have a common congestion controller. For many applications, audio is much more critical than video. In those cases, the video may back off, but the audio transmission remains unchanged.

Mo Zanaty pointed to the importance of the media start-up behavior, which is an area where the exchange of real-time interactive media is different from a TCP-based file transfer. The instantaneous experience in the first part of a video call is highly determinative of people's perception of the call quality. Vendors are using vague heuristics, for example, data from the last call to figure out what to do on the next call. Lars Eggert highlighted that the start-up behavior of an application affects ongoing performance of other flows if, for example, an application blasts at line rate at the beginning of a video stream. You need to start slow enough to not cause congestion to others. Randell Jesup argued that for an interactive real-time video application, you really need to have most of your bandwidth right away. Colin Perkins agreed and added that on startup you need good quality video quickly, but perhaps not as quickly as voice. The requirements are likely going to be different from audio to video and maybe even vary between different applications. Various protocol exchanges take place before media is exchanged between endpoints (such as Session Traversal Utilities for NAT (STUN) packets [13] as part of the Interactive Connectivity Establishment (ICE) [15] or a Datagram Transport Layer Security (DTLS) handshake [14]) and may be used to obtain simple start-up measurements.

The group agreed that it is feasible to design a congestion control algorithm that works on mostly idle networks. In the view of the participants, upgrades of the network infrastructure can happen in parallel. This view was later confirmed at the RTP Media Congestion Avoidance Techniques (RMCAT) BoF meeting at IETF 84 (July 29 - August 3, 2012, Vancouver, BC, Canada) that led to the formation of the RMCAT working group [34].

3. Recommendations

The participants suggested to explore two primary solution tracks: changes to network infrastructure and the development of algorithms to avoid self-inflicted queuing. These are discussed below. A third approach recommended by some participants was to change the way TCP is used in browsers and other HTTP-based applications. For example, by not opening too many concurrent TCP connections, and by improving the interaction with other non-real-time applications (such as video streaming and file sharing), additional improvements can be made. The work on HTTP 2.0 with SPDY [16] is already a step in the right direction since SPDY makes use of a more aggressive form of multiplexing instead of opening a larger number of TCP connections.

3.1. Changes to Network Infrastructure

As for all other traffic on the network, better data plane infrastructure improves the perceived quality of the best-effort service that the Internet provides for RTCWEB flows. The IETF has already developed several technologies that would be of immediate usefulness if they were to be deployed. The workshop participants expressed the hope that due to the volume and importance of RTCWEB traffic, some of these technologies might finally see widespread use.

The first and by far most important improvement is traffic segregation: the ability to use different queues for different traffic types. Specifically, jitter- and delay-sensitive protocols would benefit from being in different queues from throughput-maximizing protocols. It is not possible for a single queue/AQM to be optimal for both.

Furthermore, ECN allows routers along the end-to-end path to signal the onset of congestion and allows applications to respond early, avoiding losses and keeping queue sizes short and, therefore, end-to-end delay low. ECN is implemented on some end system stacks and routers, but is frequently not enabled. The participants expressed the importance of increasing the deployment of ECN, even if used initially only in closed environments, such as data centers (as with Data Center TCP (DCTCP) [26]).

Different mechanisms have been developed to facilitate traffic segregation. Differentiated Services [10] is one possibility in this space. If applications start to mark outgoing traffic appropriately and routers segregate traffic accordingly, browsers could more directly control the relative importance of their various flows and avoid self-competition. Compared to ECN, however, DiffServ is far more difficult to deploy meaningfully end to end, especially given that Differentiated Services Code Points (DSCPs) have no defined end-to-end meaning and packets can be re-marked.

QoS signaling together with resource reservation facilities would enable a fine-grained and flexible way to indicate resource needs to network elements, but it is also by far the most heavyweight proposal, and unlikely to be viable in the global Internet. However, as mentioned in Section 2.3, QoS signaling is not the only way to accomplish traffic segregation. Further investigations regarding stochastic fair queuing and new AQM algorithms are seen as desirable.

In any case, network infrastructure updates will take time, particularly if the interest of the involved stakeholders is not aligned (as is often the case for network operators when dealing with

over-the-top real-time traffic). It is, therefore, imperative that RTCWEB congestion control provides adequate improvement in the absence of any of the aforementioned schemes.

3.2. Avoiding Self-Inflicted Queuing

This approach tries to ensure that the network does not suffer from congestion collapse and that one data flow from a single host does not harm another data flow from the same host. A single congestion manager within the end host or the browser could help to coordinate various congestion control activities and to ensure a more coordinated approach between different applications and different flows.

The following design and testing aspects were considered relevant to this approach:

Reacting to All Congestion Signals:

To initiate the congestion control process, it is important to detect congestion in the communication path. Congestion can be detected using either an explicit mechanism or an implicit mechanism. An explicit mechanism involves direct congestion signaling usually from the congested network node, such as ECN. In case of an implicit mechanism, packet-loss events or observed delay increases are used as an indication for congestion. These measurements can also be made available in a variety of different protocols, such as RTCP reports or transport protocols. It is recommended for applications to take all available congestion signals into account and to couple the congestion control algorithm, the codec, and the application so that better information exchange between these components is possible since there are constraints on how quickly a codec can adapt to a specific sending rate.

Delay- and Loss-Based Algorithms:

The main goal of designing a congestion control algorithm for real-time conversational media is to achieve low latency. Explicit congestion signals provide the most reliable way for applications to react, but due to the lack of ECN deployment, delay-based algorithms are needed. Since there is large delay variation in wireless networks (even in a non-congested network), the workshop participants recommended that more research should be done to better understand non-congestion-related delay variation in the network. General consensus among the workshop participants was that latency-based congestion control algorithms are needed

due to the lack of loss indications caused by large buffers, even though loss-based techniques dominate latency-based techniques when the two are competing for bandwidth.

Algorithm Evaluation:

The Internet consists of heterogeneous networks, which include misconfigured and unmanaged network nodes. Bandwidth and latency vary a lot. Different services deployed using RTP/UDP have different requirements in terms of media quality. A congestion control algorithm needs to perform well not only in simulators but also in the deployed Internet. To achieve this, it is recommended to test the algorithms with real-world loss and delay figures to ensure that the desired audio/video rates are attainable using the proposed algorithms for the desired services.

Media Characteristics:

Interactive real-time voice and video data are inherently variable. Usually the content of the media and service requirements dictate the media coding. The codec may be bursty and not all frames are equally important (e.g., I-frames are more important than P-frames). Thus, codecs have limited room for adaptation. Congestion control for audio and video codecs is, therefore, different from congestion control applied to bulk file transfers where buffering is not a problem and the transmission rate can be changed to any rate suitable for the congestion control algorithm. In the workshop, these limitations were brought up and the workshop participants recommended that a congestion controller needs to be aware of these constraints. However, further investigation is needed to decide what information needs to be exchanged between a codec and the congestion manager.

Start-up Behavior:

The start-up media quality is very important for real-time interactive applications and for user-perceived application performance. The start-up behavior of these is also different from other traffic. By nature, real-time interactive communication applications want to provide a smooth user experience and maintain the best media quality possible to ease the interaction. While it may be desirable from a user-experience point of view to immediately start streaming video with high-definition quality and audio of a wideband codec, this will have impacts on the bandwidth of the already ongoing flows. As such,

it would be ideal to start slow enough to avoid causing excessive congestion to other flows but fast enough to offer a good user experience. The sweet spot, however, yet has to be found.

4. Security Considerations

Two position papers focused on security, but these papers were not discussed during the workshop. As such, nothing beyond the material contained in those position papers can be reported.

5. Acknowledgments

We would like to thank the participants and the paper authors of the position papers for their input.

Additionally, we would like to thank the following persons for their review comments: Michael Welzl, John Leslie, Mirja Kuehlewind, Matt Mathis, Mary Barnes, Spencer Dawkins, Dave Thaler, and Alissa Cooper.

6. Informative References

[1] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

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

[3] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", Work in Progress, February 2014.

[4] Welzl, M., Damjanovic, D., and S. Gjessing, "MulTFRC: TFRC with weighted fairness", Work in Progress, July 2010.

[5] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[6] Nichols, K. and V. Jacobson, "Controlled Delay Active Queue Management", Work in Progress, March 2014.

[7] Schulzrinne, H., Johnston, W., and J. Miller, "Large-Scale Measurement of Broadband Performance: Use Cases, Architecture and Protocol Requirements", Work in Progress, September 2012.

[8] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, December 2012.

[9] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[10] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[11] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.

[12] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

[13] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[14] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[15] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[16] Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", Work in Progress, June 2014.

[17] Floyd, S. and J. Kempf, "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004.

[18] Chu, J., Dukkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, April 2013.

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

[20] Zanaty, M., "Fairness Considerations for Congestion Control for Interactive Real-Time Communication (IRTC)", IAB/ RTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[21] Sarker, Z. and I. Johansson, "Improving the Interactive Real-Time Video Communication with Network Provided Congestion Notification", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[22] Winstein, K., Sivaraman, A., and H. Balakrishnan, "Congestion Control for Interactive Real-Time Flows on Today's Internet", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[23] Jarvinen, I., Ding, A., Nyrhinen, A., and M. Kojo, "Harsh RED: Improving RED for Limited Aggregate Traffic", In Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA 2012), March 2012.

[24] Allman, M., "Comments on Bufferbloat", In ACM SIGCOMM Computer Communication Review, Volume 43, Issue 1, pp. 30-37, January 2013, <http://dl.acm.org/citation.cfm?doid=2427036.2427041>.

[25] Bauer, S., Beverly, R., and A. Berger, "Measuring the state of ECN readiness in servers, clients,and routers", In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference (IMC '11), New York, NY, USA, pp. 171-180, February 2011, <http://dl.acm.org/citation.cfm?doid=2068816.2068833>.

[26] Bauer, S., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and M. Sridharan, "Data center TCP (DCTCP)", In Proceedings of the ACM SIGCOMM 2010 conference (SIGCOMM '10), New York, NY, USA, pp. 63-74, August 2010, <http://dl.acm.org/citation.cfm?doid=1851182.1851192>.

[27] Jarvinen, I., Chemmagate, B., Daniel, L., Ding, A., Kojo, M., and M. Isomaki, "Impact of TCP on Interactive Real- Time Communication", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[28] Jennings, C., Nandakumar, S., and H. Phan, "Vendors Considered Harmfull", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[29] Welzl, M., "One control to rule them all", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[30] Leslie, J., "There is No Magic Transport Wand", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

[31] Gettys, J. and J. Gettys, "Bufferbloat: Dark Buffers in the Internet", IEEE Internet Computing, Volume 15, Issue 3, pp. 95-96, May/June 2011.

[32] Feng, W., Shin, K., Kandlur, D., and D. Saha, "The BLUE active queue management algorithms", In IEEE/ACM Transactions on Networking, Volume 10, Issue 4, pp. 513-528, August 2002.

[33] IETF, "IP Performance Metrics (ippm) Working Group", January 2012, <http://datatracker.ietf.org/wg/ippm/charter/>.

[34] IETF, "RTP Media Congestion Avoidance Techniques (rmcat) Working Group", January 2012, <http://datatracker.ietf.org/wg/rmcat/charter/>.

[35] IETF, "Active Queue Management and Packet Scheduling (aqm) Working Group", September 2013, <http://datatracker.ietf.org/wg/aqm/charter/>.

[36] Gettys, J. and K. Nichols, "Bufferbloat: Dark Buffers in the Internet", Communications of the ACM, Vol. 55, No. 1, pp. 57-65, January 2012, <http://cacm.acm.org/magazines/2012/1/144810-bufferbloat/>.

[37] Jacobson, V., "pathchar - a tool to infer characteristics of Internet paths", Presented at the Mathematical Sciences Research Institute, April 1997, <ftp://ftp.ee.lbl.gov/pathchar/msri-talk.pdf>.

[38] McKenney, P., "Stochastic Fairness Queuing", In IEEE INFOCOM'90 Proceedings, Volume 2, pp. 733-740, June 1990.

[39] Wikipedia, "Bufferbloat", May 2014, <http://en.wikipedia.org/w/index.php?title=Bufferbloat&oldid=608805474>.

[40] Wikipedia, "Video compression picture types", March 2014, <http://en.wikipedia.org/w/index.php?title=Video_compression_picture_types&oldid=602183340>.

[41] FCC, "Methodology - Measuring Broadband America July Report 2012", July 2012, <http://www.fcc.gov/measuring-broadband-america/2012/methodology-july-report-2012>.

[42] IETF, "lmap -- Large Scale Measurement of Access network Performance Mailing List", 2012, <https://www.ietf.org/mailman/listinfo/lmap>.

[43] Holmer, S., "On Fairness, Delay and Signaling of Different Approaches to Real-time Congestion Control", IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication, July 2012.

Appendix A. Program Committee

This workshop was organized by Harald Alvestrand, Bernard Aboba, Mary Barnes, Gonzalo Camarillo, Spencer Dawkins, Lars Eggert, Matthew Ford, Randell Jesup, Cullen Jennings, Jon Peterson, Robert Sparks, and Hannes Tschofenig.

Appendix B. Workshop Material

- Main Workshop Page:
  http://www.iab.org/activities/workshops/cc-workshop/

- Position Papers:
  http://www.iab.org/activities/workshops/cc-workshop/papers/

- Slides:
  http://www.iab.org/activities/workshops/cc-workshop/slides/

Appendix C. Accepted Position Papers

1. "One control to rule them all" by Michael Welzl

2. "Congestion Avoidance Through Deterministic" by Pier Luca Montessoro, Riccardo Bernardini, Franco Blanchini, Daniele Casagrande, Mirko Loghi, and Stefan Wieser

3. "Congestion Control in Real Time Media - Context" by Harald Alvestrand

4. "Improving the Interactive Real-Time Video Communication with Network Provided Congestion Notification" by ANM Zaheduzzaman Sarker and Ingemar Johansson

5. "Multiparty Requirements in Congestion Control for Real-Time Interactive Media" by Magnus Westerlund

6. "On Fairness, Delay and Signaling of Different Approaches to Real-time Congestion Control" by Stefan Holmer

7. "RTP Congestion Control and RTCWEB Application Feedback" by Ted Hardie

8. "Issues with Using Packet Delays and Inter-arrival Times for Inference of Internet Congestion" by Wesley M. Eddy

9. "Impact of TCP on Interactive Real-Time Communication" by Ilpo Jarvinen, Binoy Chemmagate, Laila Daniel, Aaron Yi Ding, Markku Kojo, and Markus Isomaki

10. "Security Concerns For RTCWEB Congestion Control" by Dan York

11. "Vendors Considered Harmfull" by Cullen Jennings, Suhas Nandakumar, and Hein Phan

12. "Network-Assisted Dynamic Adaptation" by Xiaoqing Zhu and Rong Pan

13. "Congestion Control for Interactive Real-Time Applications" by Sanjeev Mehrotra and Jin Li

14. "There is No Magic Transport Wand" by John Leslie

15. "Towards Adaptive Congestion Management for Interactive Real-Time Communications" by Dirk Kutscher and Miriam Kuehlewind

16. "Enlarge the pre-congestion spectrum usage?" by Xavier Marjou and Emile Stephan

17. "Congestion control for users who don’t have first-class internet access" by Maire Reavy

18. "Realtime Congestion Challenges" by Randell Jesup

19. "Congestion Control for Interactive Media: Control Loops & APIs" by Varun Singh, Joerg Ott, and Colin Perkins

20. "Some Notes on Threat Modelling Congestion Management" by Eric Rescorla

21. "Timely Detection of Lost Packets" by Ali C. Begen

22. "Congestion Control Considerations for Data Channels" by Michael Tuexen

23. "Position paper on CC for Interactive RT" by Matt Mathis

24. "Overall Considerations for Congestion Control" by Mo Zanaty, Bill VerSteeg, Bent Christensen, David Benham, and Allyn Romanow

25. "Fairness Considerations for Congestion Control" by Mo Zanaty

26. "Media is not Data: The Meaning of Fairness for Competing Multimedia Flows" by Timothy B. Terriberry

27. "Thoughts on Real-Time Congestion Control" by Murari Sridharan

28. "Congestion Control for Interactive Real-Time Flows on Today's Internet" by Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan

29. "Congestion Control Principles for CoAP" by Carsten Bormann and Klaus Hartke

30. "Erasure Coding and Congestion Control for Interactive Real-Time Communication" by Pierre-Ugo Tournoux, Tuan Tran Thai, Emmanuel Lochin, Jerome Lacan, and Vincent Roca

31. "Video Conferencing Specific Considerations for RTP Congestion Control" by Stephen Botzko and Mary Barnes

32. "The Internet is Broken, and How to Fix It" by Jim Gettys

33. "Deployment Considerations for Congestion Control in Real-Time Interactive Media Systems" by Jari Arkko

Appendix D. Workshop Participants

We would like to thank the following workshop participants for attending the workshop:

- Mat Ford
- Bernard Aboba
- Alissa Cooper
- Mary Barnes
- Lars Eggert
- Harald Alvestrand
- Gonzalo Camarillo
- Robert Sparks
- Cullen Jennings
- Dirk Kutscher
- Carsten Bormann
- Michael Welzl
- Magnus Westerlund
- Colin Perkins
- Murari Sridharan
- Klaus Hartke
- Pier Luca Montessoro
- Xavier Marjou
- Vincent Roca
- Wes Eddy
- Ali C. Begen
- Mo Zanaty
- Jin Li
- Dave Thaler

- Bob Briscoe
- Barry Leiba
- Jari Arkko
- Stewart Bryant
- Martin Stiemerling
- Russ Housley
- Marc Blanchet
- Zaheduzzaman Sarker
- Xiaoqing Zhu
- Randell Jesup
- Eric Rescorla
- Suhas Nandakumar
- Hannes Tschofenig
- Bill VerSteeg
- Sean Turner
- Keith Winstein
- Jon Peterson
- Maire Reavy
- Michael Tuexen
- Stefan Holmer
- Joerg Ott
- Timothy Terriberry
- Benoit Claise
- Ted Hardie
- Stephen Botzko
- Matt Mathis
- David Benham
- Jim Gettys
- Spencer Dawkins
- Sanjeev Mehrotra
- Adrian Farrel
- Greg White
- Markku Kojo

We also had remote participants, namely:

- Emmanuel Lochin
- Mark Handley
- Anirudh Sivaraman
- John Leslie
- Varun Singh

Authors' Addresses

Hannes Tschofenig
Hall, Tirol 6060
Austria

EMail: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

Lars Eggert
Sonnenallee 1
Kirchheim 85551
Germany

Phone: +49 151 12055791
EMail: lars@netapp.com
URI: http://eggert.org/

Zaheduzzaman Sarker
Lulea SE-971 28
Sweden

Phone: +46 10 717 37 43
EMail: zaheduzzaman.sarker@ericsson.com