Network Working Group R. Huang
INTERNET-DRAFT R. Even
Intended Status: Informational Huawei
Expires: January 11, 2014 V. Singh
Aalto University
July 10, 2013

Consideration for Selecting RTCP Extended Report (XR) Metrics for RTCWEB Statistics API
draft-huang-xrblock-rtcweb-rtcp-xr-metrics-00

Abstract

This document describes monitoring features related to RTCWEB. It provides a list of RTCP XR metrics that are useful and may need to be supported in some RTCWEB implementations.

Status of this Memo

Copyright and License Notice

Table of Contents

1 Introduction

Web based real-time communication (WebRTC) is becoming prevalent. To help measure the quality of the WebRTC services better, applications need the ability to estimate the service quality and to inform about network problems. If sufficient information (metrics or statistics) are provided to the applications, it can function better in providing better media quality. [RTCWEB-REQ] specifies a requirement for statistics, which is listed below for convenient reading.

"F38   The browser MUST be able to collect statistics, related to the transport of audio and video between peers, needed to estimate quality of service."

[RTCWEB-STAT] describes a registration procedure for choosing metrics reported by the Javascript API. It also identifies basic metrics reported in the RTCP Sender and Receiver Report (SR/RR) to fulfill this requirement. However, the basic metrics from RTCP SR/RR may not be sufficient for precise quality monitoring or troubleshooting. For example, the metrics from SR/RR with respect to packet loss are controversial and could be misleading (look at section 3.2.3 for discussion). They’re better to be complemented with correspondent metrics defined in RTCP XR. . Thus, indicating a minimal useful statistic metrics would be helpful.

There are two ways to convey these metrics in WebRTC. One way is the Javascript application extracts the local statistics from the browser’s internals using an API. Since the media path goes directly between browsers, the application is able to query the statistics information directly from the local RTP stack. The remote-side information could be fetched by some other means outside the scope of WebRTC.

Another possible method is to use RTCP XRs, implemented in the browser to allow sending the statistics information directly between endpoints. Since RTP is used as the media transport protocol for RTCWEB and RTCP XR can provide more useful statistics information than RTCP SR/RR concerning media quality. However, no RTCP report extensions are currently mandated in RTCWEB for monitoring because the chosen metrics usually depend on the application. It is agreed that these RTP monitoring extensions could be supported later by SDP negotiation between browsers. So at the current stage, we should only consider those metrics that can be measured at a local endpoint and useful for WebRTC.

2 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3 Considerations for Selecting Monitoring Metrics

Since RTP is the media transport protocol in RTCWEB, we mainly discuss the RTP based monitoring metrics. RTCP SR/RR collects information and reports it periodically. However, it only provides partial information, which means that one can use it in a limited way to identify network problems. This may not be sufficient for diagnosing problems or performance monitoring. RTP Control Protocol Extended Reports [RFC3611] and other extensions discussed in XRBLOCK working group have defined many monitoring metrics that complement the RTCP SR/RR. These metrics are useful for a range of RTP applications. In this chapter some recommendations are provided to help RTCWEB applications choose the metrics specified in RTCP XR and other extensions defined in XRBLOCK working.

3.1 Consideration for Impact of Measurement Interval

RTCP extensions like RTCP XR usually share the same timing interval with RTCP SR/RR, i.e., they are sent as compound packets, together with the RTCP SR/RR. Alternatively, the RTCP XR can use a different measurement interval and all XRs using the same measurement interval are compounded together and the measurement interval is indicated in a specific measurement information block [RFC6776].

When using WebRTC Statistics APIs (see section 7 of [WebRTCAPI]), Javascript applications can query this information at arbitrary intervals. Some applications may choose 1 second or another interval. Currently, RTCP XRs are not required to be implemented by the browsers; the endpoint can query only local statistics. In the following sections, we only discuss about the metrics, which are mainly local reception statistics.

3.2 Candidate Metrics

3.2.1 Retransmitted Packet Count Metric

RTP retransmission is not required to be implemented in RTCWEB. As depicted in [RTCWEB-RTPUSAGE], NACKs may be sent by receivers to indicate missing RTP packets and senders may send retransmission packets in response to these NACKs. In low delay networks with low loss rates, retransmissions have great value without incurring additional complexity. Providing some retransmission statistic information in such applications could help to provide a more accurate quality evaluation since retransmission could greatly reduce the impact of packet loss.

Number of retransmission packets metric counts the retransmitted packets that are successfully received by receivers. It could be used for quality evaluation in RTCWEB systems that has negotiated to support the transmission mechanism. The number of retransmitted packets is subtracted from the number of lost packets, which indicates the residual lost packets.

3.2.3 Loss, Discard and Duplicate Packet Count Metric

In multimedia transport, packets that are received abnormally are classified into 3 types: lost, discarded and duplicate packets. Packet loss may be caused by network devices breakdown, bit-error corruption or serious congestions (packets dropped by an intermediate router queue). Duplicated data packets may be due to a slight long network delay which causes the sender to retransmit the original packets. Discarded packets are packets that have been delayed long enough and are considered useless by the receiver. All the 3 cases cause problems for multimedia services, as missing data and long delay can cause degradation in service quality, e.g., large blocks of packets that are missing (lost or discarded) at once may cause choppy audio, and long network transmission time may cause audio or video buffering. RTCP SR/RR defines a metric for counting the total number of RTP data packets that have been lost since the beginning of reception. But this statistic doesn’t distinguish lost packets from discarded and duplicate packets. Packets that arrive late and are discarded are not treated as lost, and duplicate packets will be regarded as a normally received packet. This metric is misleading if many duplicated or discarded packets are received, which causes the quality of media transport to look okay from the statistic while actually users are experiencing bad service quality, because packets are still missing. So in such cases, it’s better to use more accurate metrics in addition to those defined in RTCP SR/RR.

The lost packets and duplicated packets metrics defined in Statistics Summary Report Block of [RFC3611] extend the information of loss carried in standard RTCP SR/RR. They explicitly give an account of lost and duplicated packets, which are useful for network problem diagnosis. Note that the distinction between loss and duplication is unnecessary in applications that do not apply RTP retransmissions.

Using loss metrics without considering discard metrics may result in inaccurate quality evaluation, as packet discard due to jitter is often more prevalent than packet loss in modern IP networks. The discarded metric specified in [XRBLOCK-DISCARD] counts the number of packets discarded due to the jitter. It augments the loss statistics metrics specified in standard RTCP SR/RR. For those RTCWEB services with jitter buffer requiring precise quality evaluation and accurate troubleshooting, this metric is useful as a complement to the metrics of RTCP SR/RR.

3.2.4 Discard Octets Metric

The metric reports the cumulative size of the packets discarded in the interval, it is complementary to number of discarded packets. An application measures sent octets and received octets to calculate sending rate and receiving rate, respectively. The application can calculate goodput in a particular interval by subtracting the discarded octets from the received octets.

For WebRTC, discarded octets supplements the sent and received octets and provides an accurate method for calculating goodput.

3.2.5 Run Length Encoded Metrics for Loss, Discard and Post-repair

Run-length encoding uses a bit vector to encode information about the packet. Each bit in the vector represents a packet and depending on the signaled metric it defines if the packet was lost, duplicated, discarded, or repaired. An endpoint typically uses the run length encoding to accurately communicate the status of each packet in the interval to the other endpoint. [RFC3611], [XRBLOCK-DISCARDRLE] and [RFC5725] define run-length encoding for lost and duplicate packets, discarded packets and post-repair packets.

For WebRTC, the application may benefit from the additional information. If losses occur after discards, an endpoint may be able to correlate the two run length vectors to identify congestion-related losses, i.e., a router queue became overloaded causing delays and then overflowed. If the losses are independent, it may indicate bit-error corruption. Lastly, when using error-resilience mechanisms like Forward Error Correction (FEC) or NACK, the post-repair metric block indicates the success of the error-resilience mechanism to the monitoring application or the sending endpoint. The endpoint can correlate the loss and post-repair run lengths to ascertain the ratio of repaired packets to lost packets and where the losses occurred in the interval. For example, consecutive losses are likely not to be repaired by a simple FEC scheme.

3.2.6 Burst/Gap Pattern Metrics for Loss and Discard

RTCP SR/RR defines coarse metrics regarding loss statistics, the metrics are all about per call statistics and not detailed enough to capture some transitory nature of the impairments like bursty packet loss. Even if the average packet loss rate is low, the lost packets are likely to occur during short dense periods, resulting in short periods of degraded quality. Thus, bursty packet loss has a severe impact on media quality. Distributed burst provides a higher subjective quality than a non burst distribution for low packet loss rates whereas for high packet loss rates the converse is true. So capturing burst gap information is very helpful for quality evaluation and locating impairments. If RTCWEB services have the requirement to evaluate the services quality, burst gap metrics provides more accurate information than RTCP SR/RR.

[RFC3611] introduces burst gap metrics in VoIP report block. These metrics record the density and duration of burst and gap periods, which are helpful in isolating network problems since bursts correspond to periods of time during which the packet loss/discard rate is high enough to produce noticeable degradation in audio or video quality. Burst gap related metrics are also introduced in [XRBLOCK-BURSTGAPDISCARD] and [RFC6958] which define two new report blocks for usage in a range of RTP applications beyond those described in RFC3611. These metrics distinguish discarded packets from loss packets that occur in the bursts period and provides more information for diagnosing network problems. Besides that, the metric number of bursts counts the burst events which could provide useful information to evaluate the frequency of burst occurrences. So if WebRTC services have the requirement to do quality evaluation and observe when and why quality degrades, these metrics should be considered.

3.2.7 Jitter and Jitter Buffer Metrics

Although RFC3550 has provided a simple running average of the packet to packet delay variation, this metric with a scaling factor of 16 is most strongly affected by the delay variation of the most recent 10-20 packets. If the interval is big enough, e.g., the RTCP SR/RR interval which is usually at least 5 seconds, this metric may not be able to reflect the variation of the whole interval. So if we want to have better information about the impact of jitter on applications, inter arrival jitter of RTCP SR/RR is not enough.

Jitter information metrics defined in statistics summary report block of [RFC3611] introduce four statistics: minimum jitter, max jitter, mean jitter and deviation jitter values. These jitter information metrics measure jitter in a way that correlates better with the impact of jitter. They could be used to determine whether jitter level is acceptable or not to increase the jitter buffer size in browsers or to enable adaptive jitter buffer operation. Even in cases where jitter buffer size couldn’t be adjusted better tolerate the jitters, this information may be used to decide whether steps should be taken to isolate the source of jitter and eliminate the problem at source. [RFC6798] specified some statistics which calculate the percentile of jitter exceeding a certain threshold, which give a deeper analysis than the jitter metrics of [RFC3611] do. They are Positive PDV Threshold, Positive PDV Percentile, Negative PDV Threshold, Negative PDV Percentile. Thus, these statistic metrics should be considered in the WebRTC implementations which employ jitter buffer to temporarily store arriving packets in order to minimize jitters.

The size of the jitter buffer affects the end-to-end delay on the network and also the packet discard rate. When the buffer size is too small, slower packets are not played out and dropped, while when the buffer size is too large, packets are held longer than necessary and consequently reduce conversational quality. Measurement of jitter buffer should not be ignored in the evaluation of end user perception of conversational quality. Jitter buffer related metrics, such as maximum and nominal jitter buffer, could be used to show how the jitter buffer behaves at the receiving end of RTP stream. They are useful for providing better end-user quality of experience (QoE) when jitter buffer factors are used as inputs to calculate QoE metrics. RTCWEB services are point-to-point connections. Usually, senders don’t care what the perception quality of the remote end is. But in some cases, it may be meaningful for receivers to send this kind of information to senders telling what the media quality the receiver is being through. For example, senders who have the ability to adjust the media codecs may require to know the quality of the receivers so that they can switch to a lower bandwidth usage codec when service degradation happens. Thus for those cases, jitter buffer metrics could be considered. The definition of these metrics could be found in [XRBLOCK-JITTERBUFFER].

3.2.8 Frame Impairment Summary Metrics

RTP has different framing mechanisms for different payload types. For audio streams, a single RTP packet may contain one or multiple audio frames each of which has a fixed length. On the other hand, in video streams, a single video frame may be transmitted in multiple RTP packets. The size of each packet is limited by the Maximum Transmission Unit (MTU) of the underlying network. However, statistics from standard SR/RR only collect information from transport layer, which may not fully reflect the quality observed by the application. For example, two frame types used in different video algorithms are key frames and derived frames. Key frames are usually independently coded without prediction from other pictures and used as a reference frame for predicting other pictures, key frames are always much larger in size than derived frames. The loss of these key frames results in a substantial reduction in video quality. Thus it is meaningful to consider this application layer information in WebRTC implementations, which influence sender strategies to mitigate the problem or require the accurate assessment of users' quality of experience.

[XRBLOCK-SUMMARY] defines metrics conveyed in RTCP XR by receivers reporting to senders or other monitor devices. The following metrics can also be considered for WebRTC's Statistics API: number of discarded key frames, number of duplicated key frames, number of fully lost key frames, number of partial lost key frames, number of discarded derived frames, number of duplicated derived frames, number of full lost derived frames, number of partial lost derived frames. Details of the definition of these metrics are in [XRBLOCK-SUMMARY].

4 Security Considerations

The monitoring activities are implemented between two browsers or browser-to-server. Also encryption procedures, such as those being suggested for a Secure RTCP (SRTCP), can be used. It is believed that monitoring in RTCWEB introduces no new security considerations beyond those described in [RTCWEB-RTPUSAGE] and [RTCWEB-SECURITY].

5 IANA Considerations

There is no IANA action in this document.

6. Acknowledgement

The authors would like to thank Dan Romascanu and Colin Perkins for their valuable comments and suggestions on this document.

7 References

7.1 Normative References

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[RTCWEB-REQ] Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use-cases and Requirements", December 2012.

[RTCWEB-STAT] Alvestrand, H., "A Registry for WebRTC statistics identifiers", September 24, 2012.

[RTCWEB-RTPUSAGE] Perkins, C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", February 2013.

[RTCWEB-SECURITY] Rescorla, E., "Security Considerations for RTC-Web", January 2013.

[RFC6792] Wu, Q., Ed., Hunt, G., and P. Arden, "Monitoring Architecture for RTP", RFC 6792, November 2012.

[XRBLOCK-BURSTGAPDISCARD] Hunt, G., Clark, A., and Q. Wu, Ed., "RTCP XR Report Block for Burst/Gap Discard Metric Reporting", April 2013.

[RFC6958] Hunt, G., Clark, A., Zhang, S., Ed., "RTCP XR Report Block for Burst/Gap Loss Metric Reporting", April 2013.

[XRBLOCK-JITTERBUFFER] Clark, A., Singh, V., and Q. Wu, "RTCP XR Report Block for Jitter Buffer Metric Reporting", April 2013

[XRBLOCK-DISCARD] Hunt, G., Clark, A., Zorn, G., and Q. Wu, "RTCP XR Report Block for Discard Metric Reporting", April 2013.

[RFC6798] Clark, A., Wu, Q., Ed., "RTCP Control Protocol (RTCP) Extended Report (XR) Block for Packet Delay Variation

Metric Reporting", November 2012.

[RFC5725] Begen, A., Hsu, D., Lague, M., "Post-Repair Loss RLE Report Block Type for RTP Control Protocol (RTCP) Extended Reports (XRs)", February 2010

[XRBLOCK-DISCARDRLE] Singh, V., Ott, J., Curcio, I.D.D., "RTP Control Protocol (RTCP) Extended Reports (XR) for Run Length Encoding (RLE) of Discarded Packets", July 2013

[WebRTCAPI] Bergkvist, A., Burnett, D., Jennings, C., Ed., http://dev.w3.org/2011/webrtc/editor/webrtc.html, June 2013

Authors' Addresses

Rachel Huang
Huawei
101 Software Avenue, Yuhua District
Nanjing 210012
China

EMail: rachel.huang@huawei.com

Roni Even
Huawei
14 David Hamelech
Tel Aviv 64953
Israel

EMail: roni.even@mail01.huawei.com

Varun Singh
Aalto University
School of Electrical Engineering
Otakaari 5 A
Espoo, FIN 02150
Finland

Email: varun@comnet.tkk.fi
URI: http://www.netlab.tkk.fi/~varun/