XR Block Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 26, 2017

V. Singh
callstats.io
R. Huang
R. Even
Huawei
D. Romascanu
Avaya
L. Deng
China Mobile

September 22, 2016

Considerations for Selecting RTCP Extended Report (XR) Metrics for the WebRTC Statistics API
draft-ietf-xrblock-rtcweb-rtcp-xr-metrics-04

Abstract

This document describes monitoring features related to media streams in Web real-time communication (WebRTC). It provides a list of RTCP Sender Report, Receiver Report and Extended Report metrics, which may need to be supported by RTP implementations in some diverse environments. It lists a set of identifiers for the WebRTC's statistics API. These identifiers are a set of RTCP SR, RR, and XR metrics related to the transport of multimedia flows.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 26, 2017.

Copyright Notice

Table of Contents

1. Introduction

Web real-time communication (WebRTC) deployments are emerging and applications need to be able to estimate the service quality. If sufficient information (metrics or statistics) are provided to the applications, it can attempt to improve the media quality. [RFC7478] specifies a requirement for statistics:

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

The WebRTC Stats API [W3C.WD-webrtc-stats-20150206] currently lists metrics reported in the RTCP Sender and Receiver Report (SR/RR) [RFC3550] to fulfill this requirement. However, the basic metrics from RTCP SR/RR are not sufficient for precise quality monitoring, or diagnosing potential issues.

In this document, we provide rationale for choosing additional RTP metrics for the WebRTC getStats() API [W3C.WD-webrtc-20150210]. The document also creates a registry containing identifiers from the metrics reported in the RTCP Sender, Receiver, and Extended Reports. All identifiers proposed in this document are RECOMMENDED to be implemented by an endpoint. An endpoint MAY choose not to expose an identifier if it does not implement the corresponding RTCP Report.

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 [RFC2119].

ReportGroup: It is a set of metrics identified by a common Synchronization source (SSRC).

3. RTP Statistics in WebRTC Implementations

The RTCP Sender Reports (SRs) and Receiver Reports (RRs) [RFC3550] exposes the basic metrics for the local and remote media streams. However, these metrics provides only partial or limited information, which may not be sufficient for diagnosing problems or quality monitoring. For example, it may be useful to distinguish between packets lost and packets discarded due to late arrival, even though

they have the same impact on the multimedia quality, it helps in identifying and diagnosing issues.

RTP Control Protocol Extended Reports (XRs) [RFC3611] and other extensions discussed in the XRBLOCK working group provide more detailed statistics, which complement the basic metrics reported in the RTCP SR and RRs. Section Section 5 discusses the use of XR metrics that may be useful for monitoring the performance of WebRTC applications. Sections Section 6 proposes a set of candidate metrics.

The WebRTC application extracts the statistic from the browser by querying the getStats() API [W3C.WD-webrtc-20150210], but the browser currently only reports the local variables i.e., the statistics related to the outgoing RTP media streams and the incoming RTP media streams. Without the support of RTCP XRs or some other signaling mechianism, the WebRTC application cannot expose the remote endpoints' statistics. At the moment [I-D.ietf-rtcweb-rtp-usage] does not mandate the use of any RTCP XRs and since their usage is optional. If the use of RTCP XRs is successfully negotiated between endpoints (via SDP), thereafter the application has access to both local and remote statistics. Alternatively, once the WebRTC application gets the local information, they can report it to an application server or a third-party monitoring system, which provides quality estimations or diagnosis services for application developers. The exchange of statistics between endpoints or between a monitoring server and an endpoint is outside the scope of this document.

4. Considerations for Impact of Measurement Interval

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

When using WebRTC getStats() APIs (see section 7 of [W3C.WD-webrtc-20150210]), the applications can query this information at arbitrary intervals. For the statistics reported by the remote endpoint, e.g., those conveyed in an RTCP SR/RR/XR, these will not change until the next RTCP report is received. However, statistics generated by the local endpoint have no such restrictions as long as the endpoint is sending and receiving media. For example, an application may choose to poll the stack for statistics every 1 second, in this case the underlying stack local will return the current snapshot of the local statistics (for incoming and outgoing

media streams). However it may return the same remote statistics as before for the remote statistics, as no new RTCP reports may have been received in the past 1 second. This can occur when the polling interval is shorter than the average RTCP reporting interval.

5. Candidate Metrics

Since following metrics are all defined in RTCP XR which is not mandated in WebRTC, all of them are local. However, if RTCP XR is supported by negotiation between two browsers, following metrics can also be generated remotely and be sent to local by RTCP XR packets.

Following metrics are classified into 3 categories: network impact metrics, application impact metrics and recovery metrics. Network impact metrics are the statistics recording the information only for network transmission. They are useful for network problem diagnosis. Application impact metrics mainly collect the information in the viewpoint of application, e.g., bit rate, frames rate or jitter buffers. Recovery metrics reflect how well the repair mechanisms perform, e.g. loss concealment, retransmission or FEC. All of the 3 types of metrics are useful for quality estimations of services in WebRTC implementations. WebRTC application can use these metrics to better calculate MoS values or Media Delivery Index (MDI) for their services.

5.1. Network Impact Metrics

5.1.1. Loss and Discard Packet Count Metric

In multimedia transport, packets which are received abnormally are classified into 3 types: lost, discarded and duplicate packets. Packet loss may be caused by network device breakdown, bit-error corruption or network congestion (packets dropped by an intermediate router queue). Duplicate packets may be a result of network delays, which causes the sender to retransmit the original packets. Discarded packets are packets that have been delayed long enough (perhaps they missed the playout time) and are considered useless by the receiver. Lost and discarded packets cause problems for multimedia services, as missing data and long delays can cause degradation in service quality, e.g., missing large blocks of contiguous packets (lost or discarded) may cause choppy audio, and long network transmission delay time may cause audio or video buffering. The 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 does not distinguish lost packets from discarded and duplicate packets. Packets that arrive late will be discarded and are not reported as lost, and duplicate packets will be regarded as a normally received packet. Hence, the loss metric can

be misleading if many duplicate packets are received or packets are discarded, which causes the quality of the media transport to appear okay from the statistic point of view, but meanwhile the users may actually be experiencing bad service quality. So in such cases, it is 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. Lost packets counts are useful for network problem diagnosis. It is better to use the loss packets metrics of [RFC3611] to indicate the packet lost count instead of the cumulative number of packets lost metric of [RFC3550]. Duplicated packets are usually rare and have little effect on QoS evaluation. So it may not be suitable for use in WebRTC.

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 [RFC7002] 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.

5.1.2. 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 are 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 may occur during short dense periods, resulting in short periods of degraded 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 the WebRTC application needs 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

[RFC7003] 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. Additionally, the block reports the frequency of burst events which is useful information for evaluating the quality of experience. Hence, if WebRTC application need to do quality evaluation and observe when and why quality degrades, these metrics should be considered.

5.1.3. Run Length Encoded Metrics for Loss, Discard

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], [RFC7097] define run-length encoding for lost and duplicate packets, and discarded packets, respectively.

The WebRTC application could 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. For the WebRTC Stats API [W3C.WD-webrtc-stats-20150206], these types of metrics are not recommended for use due to the large amount of data and the computation involved.

5.2. Application Impact Metrics

5.2.1. 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 the actual bit rate 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 the actual bit rate which is an important parameter to reflect the quality of the media. The discarded bytes metric is defined in [RFC7243].

5.2.2. 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. Video is typically encoded using two frame types i.e., key frames and derived frames. Key frames are normally just spatially compressed, i.e., without prediction from other pictures. The derived frames are temporally compressed, i.e., depend on the key frame for decoding. Hence, key frames are much larger in size than derived frames. The loss of these key frames results in a substantial reduction in video quality. Thus it is reasonable 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.

The following metrics can also be considered for WebRTC's Statistics API: number of discarded key frames, number of lost key frames, number of discarded derived frames, number of lost derived frames. These metrics can be used to calculate Media Loss Rate (MLR) of MDI. Details of the definition of these metrics are described in [RFC7003]. Additionally, the metric provides the rendered frame rate, an important parameter for quality estimation.

5.2.3. Jitter Buffer Metrics

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 endpoint. They are useful for providing better end-user quality of experience (QoE) when jitter buffer factors are used as inputs to calculate MoS values. Thus for those cases, jitter buffer metrics should be considered. The definition of these metrics is provided in [RFC7005].

5.3. Recovery metrics

This document does not consider concealment metrics as part of recovery metrics.

5.3.1. Post-repair Packet Count Metrics

Error-resilience mechanisms, like RTP retransmission or FEC, are optional in RTCWEB because the overhead of the repair bits adding to the original streams. But they do help to greatly reduce the impact of packet loss and enhance the quality of transmission. Web applications could support certain repair mechanism after negotiation between both sides of browsers when needed. For these web applications using repair mechanisms, providing some statistic information for the performance of their repair mechanisms could help to have a more accurate quality evaluation.

The un-repaired packets count and repaired loss count defined in [RFC7509] provide the recovery information of the error-resilience mechanisms to the monitoring application or the sending endpoint. The endpoint can use these metrics to ascertain the ratio of repaired packets to lost packets. Including this kind of metrics helps the application evaluate the effectiveness of the applied repair mechanisms.

5.3.2. Run Length Encoded Metric for Post-repair

[RFC5725] defines run-length encoding for post-repair packets. When using error-resilience mechanisms, the endpoint can correlate the loss run length with this metric to ascertain where the losses and repairs occurred in the interval. This provides more accurate information for recovery mechanisms evaluation than those in Section 5.3.1. However, it is not suggested to use due to their enormous amount of data when RTCP XR are supported.

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, the endpoint can correlate the loss and post-repair run lengths to ascertain where the losses and repairs occurred in the interval. For example, consecutive losses are likely not to be repaired by a simple FEC scheme.

6. Identifiers from Sender, Receiver, and Extended Report Blocks

This document describes a list of metrics and corresponding identifiers relevant to RTP media in WebRTC. These group of identifiers are defined on a ReportGroup corresponding to an Synchronization source (SSRC). In practice the application MUST be able to query the statistic identifiers on both an incoming (remote) and outgoing (local) media stream. Since sending and receiving SR and RR are mandatory, the metrics defined in the SR and RR report blocks are always available. For XR metrics, it depends on two factors: 1) if it measured at the endpoint, 2) if it reported by the endpoint in an XR report. If a metric is only measured by the endpoint and not reported, the metrics will only be available for the incoming (remote) media stream. Alternatively, if the corresponding metric is also reported in an XR report, it will be available for both the incoming (remote) and outgoing (local) media stream.

For a remote statistic, the timestamp represents the timestamp from an incoming SR/RR/XR packet. Conversely, for a local statistic, it refers to the current timestamp generated by the local clock (typically the POSIX timestamp, i.e., milliseconds since Jan 1, 1970).

As per [RFC3550], the octets metrics represent the payload size (i.e., not including header or padding).

6.1. Cumulative Number of Packets and Octets Sent

Name: packetsSent

Definition: section 6.4.1 in [RFC3550].

Name: bytesSent

Definition: section 6.4.1 in [RFC3550].

6.2. Cumulative Number of Packets and Octets Received

Name: packetsReceived

Definition: section 6.4.1 in [RFC3550].

Name: bytesReceived

Definition: section 6.4.1 in [RFC3550].

6.3. Cumulative Number of Packets Lost

Name: packetsLost

Definition: section 6.4.1 in [RFC3550].

6.4. Interval Packet Loss and Jitter

Name: jitter

Definition: section 6.4.1 in [RFC3550].

Name: fractionLost

Definition: section 6.4.1 in [RFC3550].

6.5. Cumulative Number of Packets and Octets Discarded

Name: packetsDiscarded

Definition: The cumulative number of RTP packets discarded due to late or early-arrival, Appendix A (a) of [RFC7002].

Name: bytesDiscarded

Definition: The cumulative number of octets discarded due to late or early-arrival, Appendix A of [RFC7243].

6.6. Cumulative Number of Packets Repaired

Name: packetsRepaired

Definition: The cumulative number of lost RTP packets repaired after applying a error-resilience mechanism, Appendix A (b) of [RFC7509]. To clarify, the value is upper bound to the cumulative number of lost packets.

6.7. Burst Packet Loss and Burst Discards

Name: burstPacketsLost

Definition: The cumulative number of RTP packets lost during loss bursts, Appendix A (c) of [RFC6958].

Name: burstLossCount

Definition: The cumulative number of bursts of lost RTP packets, Appendix A (e) of [RFC6958].

Name: burstPacketsDiscarded

Definition: The cumulative number of RTP packets discarded during discard bursts, Appendix A (b) of [RFC7003].

Name: burstDiscardCount

Definition: The cumulative number of bursts of discarded RTP packets, Appendix A (e) of [I-D.ietf-xrblock-independent-burst-gap-discard].

[RFC3611] recommends a Gmin (threshold) value of 16 for classifying packet loss or discard burst.

6.8. Burst/Gap Rates

Name: burstLossRate

Definition: The fraction of RTP packets lost during bursts, Appendix A (a) of [RFC7004].

Name: gapLossRate

Definition: The fraction of RTP packets lost during gaps, Appendix A (b) of [RFC7004].

Name: burstDiscardRate

Definition: The fraction of RTP packets discarded during bursts, Appendix A (e) of [RFC7004].

Name: gapDiscardRate

Definition: The fraction of RTP packets discarded during gaps, Appendix A (f) of [RFC7004].

6.9. Frame Impairment Metrics

Name: framesLost

Definition: The cumulative number of full frames lost, Appendix A (i) of [RFC7004].

Name: framesCorrupted

Definition: The cumulative number of frames partially lost, Appendix A (j) of [RFC7004].

Name: framesDropped

Definition: The cumulative number of full frames discarded, Appendix A (g) of [RFC7004].

Name: framesSent

Definition: The cumulative number of frames sent.

Name: framesReceived

Definition: The cumulative number of partial or full frames received.

7. Adding new metrics to WebRTC Statistics API

The metrics defined in this draft have already been added to the W3C WebRTC specification. The current working process to add new metrics is, create an issue or pull request on the repository of the W3C WebRTC specification (https://github.com/w3c/webrtc-stats).

8. Security Considerations

The monitoring activities are implemented between two browsers or between a browser and a server. Therefore encryption procedures, such as the ones suggested for a Secure RTCP (SRTCP), need to be used. Currently, the monitoring in RTCWEB introduces no new security considerations beyond those described in [I-D.ietf-rtcweb-rtp-usage], [I-D.ietf-rtcweb-security].

9. Acknowledgements

The authors would like to thank Bernard Aboba, Harald Alvestrand, Al Morton, Colin Perkins, and Shida Schubert for their valuable comments and suggestions on earlier version of this document.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <http://www.rfc-editor.org/info/rfc3611>.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <http://www.rfc-editor.org/info/rfc4588>.

[RFC5725] Begen, A., Hsu, D., and M. Lague, "Post-Repair Loss RLE Report Block Type for RTP Control Protocol (RTCP) Extended Reports (XRs)", RFC 5725, DOI 10.17487/RFC5725, February 2010, <http://www.rfc-editor.org/info/rfc5725>.

[RFC6776] Clark, A. and Q. Wu, "Measurement Identity and Information Reporting Using a Source Description (SDES) Item and an RTCP Extended Report (XR) Block", RFC 6776, DOI 10.17487/RFC6776, October 2012, <http://www.rfc-editor.org/info/rfc6776>.

[RFC6958] Clark, A., Zhang, S., Zhao, J., and Q. Wu, Ed., "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Loss Metric Reporting", RFC 6958, DOI 10.17487/RFC6958, May 2013, <http://www.rfc-editor.org/info/rfc6958>.

[RFC7002] Clark, A., Zorn, G., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Discard Count Metric Reporting", RFC 7002, DOI 10.17487/RFC7002, September 2013, <http://www.rfc-editor.org/info/rfc7002>.

[RFC7003] Clark, A., Huang, R., and Q. Wu, Ed., "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Discard Metric Reporting", RFC 7003, DOI 10.17487/RFC7003, September 2013, <http://www.rfc-editor.org/info/rfc7003>.

[RFC7004] Zorn, G., Schott, R., Wu, Q., Ed., and R. Huang, "RTP Control Protocol (RTCP) Extended Report (XR) Blocks for Summary Statistics Metrics Reporting", RFC 7004, DOI 10.17487/RFC7004, September 2013, <http://www.rfc-editor.org/info/rfc7004>.

[RFC7005] Clark, A., Singh, V., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for De-Jitter Buffer Metric Reporting", RFC 7005, DOI 10.17487/RFC7005, September 2013, <http://www.rfc-editor.org/info/rfc7005>.

[RFC7097] Ott, J., Singh, V., Ed., and I. Curcio, "RTP Control Protocol (RTCP) Extended Report (XR) for RLE of Discarded Packets", RFC 7097, DOI 10.17487/RFC7097, January 2014, <http://www.rfc-editor.org/info/rfc7097>.

[RFC7243] Singh, V., Ed., Ott, J., and I. Curcio, "RTP Control Protocol (RTCP) Extended Report (XR) Block for the Bytes Discarded Metric", RFC 7243, DOI 10.17487/RFC7243, May 2014, <http://www.rfc-editor.org/info/rfc7243>.

[RFC7509] Huang, R. and V. Singh, "RTP Control Protocol (RTCP) Extended Report (XR) for Post-Repair Loss Count Metrics", RFC 7509, DOI 10.17487/RFC7509, May 2015, <http://www.rfc-editor.org/info/rfc7509>.

[I-D.ietf-xrblock-independent-burst-gap-discard] Singh, V., Perkins, C., Clark, A., and R. Huang, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Independent Reporting of Burst/Gap Discard Metric", draft-ietf-xrblock-independent-burst-gap-discard-00 (work in progress), December 2015.

10.2. Informative References

[RFC7478] Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use Cases and Requirements", RFC 7478, DOI 10.17487/RFC7478, March 2015, <http://www.rfc-editor.org/info/rfc7478>.

[W3C.WD-webrtc-20150210] Bergkvist, A., Burnett, D., Jennings, C., and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20150210, February 2015, <http://www.w3.org/TR/2015/WD-webrtc-20150210>.

[I-D.ietf-rtcweb-rtp-usage] Perkins, C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", draft-ietf-rtcweb-rtp-usage-26 (work in progress), March 2016.

[I-D.ietf-rtcweb-security] Rescorla, E., "Security Considerations for WebRTC", draft-ietf-rtcweb-security-08 (work in progress), February 2015.

[W3C.WD-webrtc-stats-20150206]
Alvestrand, H. and V. Singh, "Identifiers for WebRTC's Statistics API", World Wide Web Consortium WD WD-webrtc-stats-20150206, February 2015, <http://www.w3.org/TR/2015/WD-webrtc-stats-20150206>.

[RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <http://www.rfc-editor.org/info/rfc6390>.

Appendix A. Change Log

Note to the RFC-Editor: please remove this section prior to publication as an RFC.

A.1. changes in draft-ietf-xrblock-rtcweb-rtcp-xr-metrics-04

- Removed IANA registry.

A.2. changes in draft-ietf-xrblock-rtcweb-rtcp-xr-metrics-02, -03

- Keep-alive versions, updates to references.

A.3. changes in draft-ietf-xrblock-rtcweb-rtcp-xr-metrics-01

- Create new registry for WebRTC media metrics.
- Using camelCase instead of TitleCase for identifier names.
- Imported RTCP SR and RR metrics from the registry in alvestrand-rtcweb-stats-registry.
- Added Burst/Gap rate metrics.
- Added Frames sent and received metrics.

A.4. changes in draft-ietf-xrblock-rtcweb-rtcp-xr-metrics-00

- Submitted as WG Draft.

A.5. changes in draft-huang-xrblock-rtcweb-rtcp-xr-metrics-04

- Addressed comments from the London IETF meeting:
- Removed ECN metrics.
- Merged draft-singh-xrblock-webrtc-additional-stats-01

Authors' Addresses

Varun Singh
CALLSTATS I/O Oy
Runeberginkatu 4c A 4
Helsinki 00100
Finland

Email: varun@callstats.io
URI: https://www.callstats.io/about

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

Email: rachel.huang@huawei.com

Roni Even
Huawei
14 David Hamelech
Tel Aviv 64953
Israel

Email: roni.even@mail01.huawei.com

Dan Romascanu
Avaya
Park Atidim, Bldg. #3
Tel Aviv 61581
Israel

Email: dromasca@avaya.com

Lingli Deng
China Mobile

Email: denglingli@chinamobile.com