Internet Engineering Task Force (IETF)
Request for Comments: 8451
Category: Informational
ISSN: 2070-1721

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

September 2018

Considerations for Selecting RTP Control Protocol (RTCP) Extended Report (XR) Metrics for the WebRTC Statistics API

Abstract

This document describes monitoring features related to media streams in Web real-time communication (WebRTC). It provides a list of RTP Control Protocol (RTCP) Sender Report (SR), Receiver Report (RR), and Extended Report (XR) 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 document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8451.

Table of Contents

1. Introduction

Web real-time communication (WebRTC) [WebRTC-Overview] deployments are emerging, and applications need to be able to estimate the service quality. If sufficient information (metrics or statistics) is provided to the application, 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.webrtc-stats] currently lists metrics reported in the RTCP Sender Report 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.

Standards such as "RTP Control Protocol Extended Reports (RTCP XR)" [RFC3611] as well as other extensions standardized in the XRBLOCK Working Group, e.g., burst/gap loss metric reporting [RFC6958] and burst/gap discard metric reporting [RFC7003], have been produced for the purpose of collecting and reporting performance metrics from RTP endpoint devices that can be used to have end-to-end service visibility and to measure the delivery quality in various RTP services. These metrics are able to complement those in [RFC3550].

In this document, we provide rationale for choosing additional RTP metrics for the WebRTC getStats() API [W3C.webrtc]. All identifiers proposed in this document are recommended to be implemented by an WebRTC endpoint. An endpoint may choose not to expose an identifier if it does not implement the corresponding RTCP Report. This document only considers RTP-layer metrics. Other metrics, e.g., IP-layer metrics, are out of scope.

2. Terminology

In addition to the terminology from [RFC3550], [RFC3611], and [RFC7478], this document uses the following term.

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] expose the basic metrics for the local and remote media streams. However, these metrics provide only partial or limited information, which may not be sufficient for diagnosing problems or monitoring quality. 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 problems. 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.

The WebRTC application extracts statistics from the browser by querying the getStats() API [W3C.webrtc]. The browser can easily report the local variables, i.e., the statistics related to the outgoing and incoming RTP media streams. However, without the support of RTCP XRs or some other signaling mechanism, the WebRTC application cannot expose the remote endpoints' statistics. [WebRTC-RTP-USAGE] does not mandate the use of any RTCP XRs, and 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, it can report the information to an application server or a third-party monitoring system, which provides quality estimates or diagnostic 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 "Statistics Model" in [W3C.webrtc]), 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 that 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 previously, because 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 the 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, the following metrics can also be generated remotely and be sent to the local endpoint (that generated the media) via RTCP XR packets.

The metrics are classified into 3 categories as follows: 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 from the viewpoint of the application, e.g., bit rate, frame rate, or jitter buffers. Recovery metrics reflect how well the repair mechanisms perform, e.g., loss concealment, retransmission, or Forward Error Correction (FEC). All 3 types of metrics are useful for quality estimations of services in WebRTC implementations. WebRTC applications can use these metrics to calculate the estimated Mean Opinion Score (MOS) [ITU-T_P.800.1] values or Media Delivery Index (MDI) [RFC4445] for their services.

5.1. Network Impact Metrics

5.1.1. Loss and Discard 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 device breakdown, bit-error corruption, or network congestion (packets dropped by an intermediate router queue). Duplicate packets may be a result of network delays that cause 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. However, 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 a statistical point of view, while the users are actually 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 metrics for lost packets and duplicated packets defined in the 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 packet counts are useful for network problem diagnosis. It is better to use the packet loss metrics of [RFC3611] to indicate the lost packet 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 jitter. It augments the loss statistics metrics specified in standard RTCP SR/RR. For those WebRTC services with jitter buffers 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 the transitory nature of some 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. Bursts cause lower quality experience than the non-bursts 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 service quality, burst gap metrics provide more accurate information than RTCP SR/RR.

[RFC3611] introduces burst gap metrics in the 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. Metrics related to the burst gap are also introduced in [RFC7003] and [RFC6958], which define two new report blocks for use in a range of RTP applications beyond those described in [RFC3611]. These metrics distinguish discarded packets from loss packets that occur in the burst period and provide 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 applications 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 and Discard

Run-length encoding uses a bit vector to encode information about the packet. Each bit in the vector represents a packet; 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] and [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, e.g., 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.webrtc-stats], 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. Discarded Octets Metric

The metric reports the cumulative size of the packets discarded in the interval. It is complementary to the number of discarded packets. An application measures sent octets and received octets to calculate the 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, the discarded octets metric supplements the metrics on 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 Bytes Discarded 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. 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, the statistics from standard SR/RR only collect information from the transport layer, so they 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 metrics in this category include: number of discarded key frames, number of lost key frames, number of discarded derived frames, and number of lost derived frames. These metrics can be used to calculate the Media Loss Rate (MLR) of the MDI [RFC4445]. 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, late-arriving packets are not played out and are 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. Metrics related to the jitter buffer, 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 estimated 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 [RFC7294] as part of recovery metrics.

5.3.1. Post-Repair Packet Count Metrics

Web applications can support certain RTP error-resilience mechanisms following the recommendations specified in [WebRTC-RTP-USAGE]. For these web applications using repair mechanisms, providing some statistics about the performance of their repair mechanisms could help provide a more accurate quality evaluation.

The unrepaired packet 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 post-repair packet count 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, when RTCP XR metrics are supported, using run-length encoded metrics is not suggested because the per-packet information yields an enormous amount of data that is not required in this case.

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, e.g., 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. This group of identifiers are defined on a ReportGroup corresponding to a synchronization source (SSRC). In practice, the application needs to be able to query the statistic identifiers on both an incoming (remote) and outgoing (local) media stream. Since sending and receiving SRs and RRs are mandatory, the metrics defined in the SRs and RRs are always available. For XR metrics, it depends on two factors: 1) if it is measured at the endpoint and 2) if it is reported by the endpoint in an XR block. 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 block, 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, or 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 January 1, 1970).

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

6.1. Cumulative Number of Packets and Octets Sent

Name: packetsSent
Definition: Section 6.4.1 of [RFC3550].

Name: bytesSent
Definition: Section 6.4.1 of [RFC3550].

6.2. Cumulative Number of Packets and Octets Received

Name: packetsReceived
Definition: Section 6.4.1 of [RFC3550].

Name: bytesReceived
Definition: Section 6.4.1 of [RFC3550].

6.3. Cumulative Number of Packets Lost

Name: packetsLost
Definition: Section 6.4.1 of [RFC3550].

6.4. Interval Packet Loss and Jitter

Name: jitter
Definition: Section 6.4.1 of [RFC3550].

Name: fractionLost
Definition: Section 6.4.1 of [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; see item a of Appendix A of [RFC7002].

Name: bytesDiscarded
Definition: The cumulative number of octets discarded due to late or early arrival; see 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; see item b of Appendix A of [RFC7509]. To clarify, the value is the upper bound on 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; see item c of Appendix A of [RFC6958].

Name: burstLossCount
Definition: The cumulative number of bursts of lost RTP packets; see item d of Appendix A of [RFC6958].

Name: burstPacketsDiscarded
Definition: The cumulative number of RTP packets discarded during discard bursts; see item b of Appendix A of [RFC7003].

Name: burstDiscardCount
Definition: The cumulative number of bursts of discarded RTP packets; see item e of Appendix A of [RFC8015].

[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; see item a of Appendix A of [RFC7004].

Name: gapLossRate
Definition: The fraction of RTP packets lost during gaps; see item b of Appendix A of [RFC7004].

Name: burstDiscardRate
Definition: The fraction of RTP packets discarded during bursts; see item e of Appendix A of [RFC7004].

Name: gapDiscardRate
Definition: The fraction of RTP packets discarded during gaps; see item f of Appendix A of [RFC7004].

6.9. Frame Impairment Metrics

Name: framesLost
Definition: The cumulative number of full frames lost; see item i of Appendix A of [RFC7004].

Name: framesCorrupted
Definition: The cumulative number of frames partially lost; see item j of Appendix A of [RFC7004].

Name: framesDropped
Definition: The cumulative number of full frames discarded; see item g of Appendix A 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

While this document was being drafted, the metrics defined herein were added to the W3C WebRTC specification. The process to add new metrics in the future is to create an issue or pull request on the repository of the W3C WebRTC specification (https://github.com/w3c/webrtc-stats).

8. Security Considerations

This document focuses on listing the RTCP XR metrics defined in the corresponding RTCP reporting extensions and does not give rise to any security vulnerabilities beyond those described in [RFC3611] and [RFC6792].

The overall security considerations for RTP used in WebRTC applications is described in [WebRTC-RTP-USAGE] and [WebRTC-Sec], which also apply to this memo.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[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, <https://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, <https://www.rfc-editor.org/info/rfc3611>.

[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, <https://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, <https://www.rfc-editor.org/info/rfc6776>.

[RFC6792] Wu, Q., Ed., Hunt, G., and P. Arden, "Guidelines for Use of the RTP Monitoring Framework", RFC 6792, DOI 10.17487/RFC6792, November 2012, <https://www.rfc-editor.org/info/rfc6792>.

[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, <https://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, <https://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, <https://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, <https://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>.

[RFC8015] Singh, V., Perkins, C., Clark, A., and R. Huang, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Independent Reporting of Burst/Gap Discard Metrics", RFC 8015, DOI 10.17487/RFC8015, November 2016, <http://www.rfc-editor.org/info/rfc8015>.

10.2. Informative References

[ITU-T_P.800.1]
ITU-T, "Mean Opinion Score (MOS) terminology", ITU-T P.800.1, July 2016, <https://www.itu.int/rec/T-REC-P.800.1-201607-I>.

[RFC4445] Welch, J. and J. Clark, "A Proposed Media Delivery Index (MDI)", RFC 4445, DOI 10.17487/RFC4445, April 2006, <https://www.rfc-editor.org/info/rfc4445>.

[WebRTC-Overview]
Alverstrand, H., "Overview: Real Time Protocols for Browser-based Applications", Work in Progress, draft-ietf-rtcweb-overview-19, November 2017.

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

[WebRTC-Sec]
Rescorla, E., "Security Considerations for WebRTC", Work in Progress, draft-ietf-rtcweb-security-10, January 2018.

[RFC7294] Clark, A., Zorn, G., Bi, C., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Blocks for Concealment Metrics Reporting on Audio Applications", RFC 7294, DOI 10.17487/RFC7294, July 2014, <https://www.rfc-editor.org/info/rfc7294>.

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

[W3C.webrtc]
Bergkvist, A., Burnett, C., Jennings, C., Narayanan, A., Aboba, B., Brandstetter, T., and J. Bruaroey, "WebRTC 1.0: Real-time Communication Between Browsers", W3C Candidate Recommendation, June 2018, <https://www.w3.org/TR/2018/CR-webrtc-20180621/>. Latest version available at <https://www.w3.org/TR/webrtc/>.

[W3C.webrtc-stats]
Alvestrand, H. and V. Singh, "Identifiers for WebRTC's Statistics API", W3C Candidate Recommendation, July 2018, <https://www.w3.org/TR/2018/CR-webrtc-stats-20180703/>. Latest version available at <https://www.w3.org/TR/webrtc-stats/>.

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 draft versions of this document.

Authors' Addresses

Varun Singh
CALLSTATS I/O Oy
Annankatu 31-33 C 42
Helsinki 00100
Finland

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

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@huawei.com

Dan Romascanu

Email: dromasca@gmail.com

Lingli Deng
China Mobile

Email: denglingli@chinamobile.com