

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2012

R. Huang  
R. Even  
Huawei  
March 5, 2012

**Web Real-Time Communication (RTCWEB): Monitoring Usage**  
**draft-huang-rtcweb-monitoring-00**

Abstract

This document describes the monitoring aspect usage in RTCWeb. It also gives some guidelines for which RTCP XR metrics and extensions need to be supported.

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 September 6, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Monitoring Issues for End Systems . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Monitoring Models for Service Provider . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	JS Dependent Monitoring . . . . .	<a href="#">4</a>
<a href="#">4.2.</a>	JS Independent Monitoring . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Guideline for Selecting Monitoring Metrics . . . . .	<a href="#">5</a>
<a href="#">5.1.</a>	Metrics from <a href="#">RFC3611</a> . . . . .	<a href="#">5</a>
<a href="#">5.2.</a>	Other Extended Metrics . . . . .	<a href="#">7</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">8</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">8</a>
<a href="#">9.</a>	References . . . . .	<a href="#">8</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">8</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">9</a>
	Authors' Addresses . . . . .	<a href="#">9</a>



## **1. Introduction**

RTP usage in RTCWEB framework is discussed in [[I-D.ietf-rtcweb-rtp-usage](#)]. It specifies how RTP is used in the RTCWEB context, and gives requirements of which RTP features and extensions should be included. It briefly describes the performance monitoring as one of RTP usages in RTCWEB framework, but it doesn't go into further discussion.

To help participants to know better the quality of RTCWEB services, it is required that RTCWEB framework should provide some means to estimate the services quality and to inform about network problems. This memo discusses the monitoring issues in RTCWEB framework, and makes recommendations about the selection of monitoring metrics in RTCWEB applications.

## **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](#)].

## **3. Monitoring Issues for End Systems**

Since the intention of monitoring is to reflect the service quality and the network conditions, some information, such as statistics of media packets and calculations of delay, should be provided to the participants, or even service providers. This kind of information could be presented from the signaling path (such as the total session setup time), and from the media path. Monitoring of the signal path is outside the scope of the RTCWEB standards suite, so only monitoring of the media path will be discussed in the document. This section mainly discusses the monitoring issues for participants, while [section 4](#) talks about the monitoring models for service provider.

As depicted in the RTCWEB architecture, the media path goes directly between the browsers, so browsers are the entities obtaining the monitoring information directly. That means some APIs must be provided for Javascript application to take advantage of this information. An undecided issue is what kind of monitoring information should be offered by Javascript APIs. There are two models can be applied. One model is that the browser collects the raw information from media path, does some calculation, and provides some new accumulated information to Javascript Applications through APIs. In this model, the accumulated information reflecting the



quality of media transmission informed from the APIs should be very common to any Javascript applications, so that each Javascript application would do nothing but convey the information to web server, or interpret it to user. Another alternative model is that the browser conveys the raw information collected from media path to the Javascript applications through APIs. As the browser is on the media path, it is possible for the browser to report all monitoring information, such as RTCP information, to the Javascript application. This model is more flexible than the previous one. Different Javascript applications can apply different calculations and statistics according to their respective demands. The only shortage is that it may be too much information which may be not required.

Another uninvestigated problem is what kind of monitoring information should be collected from the media path. In chapter 5, recommendations are provided to choose monitoring metrics from RTP layer.

#### **4. Monitoring Models for Service Provider**

From the perspective of service providers who wish to comply with service-level-agreements, they need to monitor the performance of the infrastructure and to provide a good service experience for their users, monitoring information should be passed to the providers' server (maybe web server). Based on the use-cases described in [[I-D.ietf-rtcweb-use-cases-and-requirements](#)], the monitoring implementation may vary in different scenarios. In this chapter, the monitoring models in RTCWEB will be discussed. Note in these models, no other new requirements for RTCWEB API are raised other than those discussed in chapter 3.

##### **4.1. JS Dependent Monitoring**

In this model, monitoring data goes from the browser to Javascript application through API, and then flows to Web Server through the signaling protocol. In this case, the Web Server is the monitoring server to collect all the monitoring information from all the end users for service provider. The browser-to-browser use-cases, telephony terminal and Fedex call of browser-GW/server use-cases specified in [[I-D.ietf-rtcweb-use-cases-and-requirements](#)] are all applicable to this model. However, there exists minor difference between these use-cases. For the cases of sharing one operator, two users have logged into the same web server. The web server can get monitoring data from both users, therefore the network condition between the two users can be inferred easily. While for the service with inter-operator calling, two users have logged into two different web servers provided by different service providers. Each web server



can only get its subscriber's monitoring data. If the monitoring information of the other user is needed, some additional agreements between two web servers will be needed, which will increase the complexity of the whole system. Actually, the media path in RTCWEB is a point-to-point unicast connection. Each side of the connection can obtain or calculate all the transport information from this unicast connection. So it is required that the browser can get monitoring data as much as it could in the JS dependent monitoring model.

#### **4.2. JS Independent Monitoring**

In this model, monitoring data may not go through Javascript applications. It flows in the media path, directly collected by some central server provided by service providers. In this case, the central server is the monitoring server. Note that the central server is not some new entity that will be added to the media path just for monitoring purpose. The central server is a participant in the RTCWEB session. Use-cases, which the browser of each participant establishes connections with a central server, such as the video conferencing system with central server specified in [\[I-D.ietf-rtcweb-use-cases-and-requirements\]](#) belong to this model. In this case, the central server is the other side of the media path. Monitoring data can be obtained directly from the media path by the central server. As long as adequate information is collected, the central server can easily give service quality evaluation and network condition diagnosis. There is no additional APIs required. All can be done according to the current RTP monitoring activities. The communication from the central server to the web server or other servers is based on monitoring implementation of service operator which is out of scope here.

### **5. Guideline 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. But it only provides partial information, which means that you can use it in a limited way to solve issues such as congestion. It's not sufficient for problem diagnosis or performance monitoring. In this chapter some guidelines are provided for RTCWEB to choose the statistics metrics specified in RTCP XR and other extensions defined in xrblock working group.

#### **5.1. Metrics from [RFC3611](#)**

RTP Control Protocol Extended Reports [[RFC3611](#)] extends the statistics specified in the RTCP SR/RR. It defines a set of metrics that





contain information for assessing media quality, especially VoIP, and diagnosing problems. The information includes 4 kinds of metrics: packet-by-packet metrics, reference time related metrics, statistics summary metrics and VoIP metrics. Packet-by-packet metrics are usually applied to some network tomography applications, such as multicast inference of network characteristics (MINC). VoIP metrics are used for VoIP performance monitoring and diagnosis. These two kinds of metrics are not generic enough to be implemented in RTCWEB. While the other two kinds of metrics, reference time related metrics and statistics summary metrics, are relatively useful for RTCWEB services.

Reference time related metrics include two report blocks defined in [RFC3611](#). They are receiver reference time report block and DLRR report block. These two report blocks enable the receivers the ability to calculate the round-trip time (RTT) between sender and itself. Reference time related metrics are very useful for the receiver to report the transmission time. In RTCWEB services, each browser participating in the session may have the requirement to measure or calculate the media path transmission time. It is recommended that the reference time related metrics SHOULD be implemented in the browser so that each side either receiver or sender can calculate the transmission time in the media path and other related .

Statistics summary report block is defined to contain statistics summary metrics. The report block records statistics about lost packets, duplicated packet, jitter information and TTL or Hop limit values. Lost packets and duplicated packets give more precise statistics than the loss statistics specified in [\[RFC3550\]](#). Jitter information metrics includes minimum jitter, max jitter, mean jitter and deviation jitter values. These metrics evaluate the transmission quality during an interval. In RTCWEB services, each browser participating in the session acts as both sender and receiver. When the browser acts as a receiver, it can calculate these metrics by itself. While when browser is a sender, it could get this kind of information by itself. One of the resolutions is that the receiver receiving the media issued from the browser sends these metrics by RTCP XR block. As what is discussed in chapter 4, each user in RTCWEB service SHOULD try to collect enough monitoring data from both sides. So if you have the requirements to present detail statistics information of the media transmission, you SHOULD implement these metrics in your browsers. TTL or Hop limit values convey the information about the path characteristics change during an interval. The change may lead to the increasing or decreasing of delay and packet losses. But in RTCWEB services, the transport of media is point-to-point and best-effort. The variety of hops in media path isn't the most direct information for user or service provider. It



is not required to be implemented in the browsers. While TTL or Hop limit values are included in the one statistics summary report block. It is recommended to set the TTL/Hop limit related fields to 0.

Note that the measurement intervals may be different from the RTCP SR/RR transmission interval specified in [\[RFC3550\]](#). It is recommended that the RTCP XR report blocks are compounded with RTCP SR/RR and sharing the same interval for simplification.

## 5.2. Other Extended Metrics

New report blocks containing new metrics are being discussed in the XRBLOCK working group. Some of them may be useful to RTCWEB applications.

RTCP SR/RR and RTCP XR have defined some metrics regarding loss/jitter statistics and calculations. These metrics are all about per call statistics and average packet loss rates which are too coarse, not detailed enough to capture some transitory nature of the impairments, such as transient network congestions. [\[I.D-ietf-xrblock-rtcp-xr-burst-gap-discard\]](#) and [\[I.D-ietf-xrblock-rtcp-xr-burst-gap-loss\]](#) define new report blocks beyond the information in VoIP metrics of [RFC3611](#) for usage in a range of RTP applications. More burst-related metrics are reported in either cumulative or interval reports, which gives a flexible algorithm usage. If RTCWEB services have the requirement to see the transient problems, it is recommended to implement the metrics defined [\[I.D-ietf-xrblock-rtcp-xr-burst-gap-discard\]](#) and [\[I.D-ietf-xrblock-rtcp-xr-burst-gap-loss\]](#) in the browsers. If both of them are applied, it is recommended to compound them together. Note the use of these two report blocks in RTCWeb applications must follow the framework defined in [\[I.D-ietf-avtcore-monarch\]](#).

[\[I.D-ietf-xrblock-rtcp-xr-discard\]](#) defines a report block to report the number of packets discarded due to the jitter. It augments the statistics summary metrics specified in [RFC3611](#). Using this metric, discard rate can be calculated. This kind of granular statistics supply accurate metrics for troubleshooting and SLA delivery. It is applicable to the applications with jitter buffer, which RTCWEB end users definitely have. If you have the requirement to get accurate monitoring information, you SHOULD implement the metrics defined in [\[I.D-ietf-xrblock-rtcp-xr-discard\]](#) along with the statistics summary report block of [RFC3611](#), and keep the report interval of this metric accordance with the interval of the statistics summary report block. Note the use of the report block in RTCWEB applications must follow the framework defined in [\[I.D-ietf-avtcore-monarch\]](#).

[\[I.D-ietf-xrblock-rtcp-xr-delay\]](#) defines a new report block including



metrics beyond the delay metrics of VoIP metrics report block specified in [RFC3611](#). It augments the information with the mean, minimum and maximum values of round-trip delay, which can be calculated by RTCP SR/RR or RTCP XR. These metrics can be used by the receivers to report the delays the receivers apperceive to the senders. In RTCWEB services, each browser participating in the session acts as both sender and receiver. When it acts as a sender, it has the requirement to know the delay information of the media stream sent by itself. As the browser of RTCWEB services all have the requirement to obtain the delay of bidirectional connections, it is recommended to implement the delay metrics defined in [\[I.D-ietf-xrblock-rtcp-xr-delay\]](#) in browsers. Note that the interval of this report block may be consistent with the RTCP SR/RR or RTCP XR. If they use the same interval with RTCP SR/RR or RTCP XR, the mean, minimum and maximum values of round-trip will have the same value. The one way symmetric media path delay can be calculated from the round trip and end system delay. The use of the report block in RTCWEB applications must follow the framework defined in [\[I.D-ietf-avtcore-monarch\]](#).

## **6. IANA Considerations**

There is no IANA action in this document.

## **7. 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 [\[I-D.ietf-rtcweb-rtp-usage\]](#) and [\[I-D.ietf-rtcweb-security\]](#).

## **8. Acknowledgements**

TBD.

## **9. References**

### **9.1. Normative References**

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



- [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., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", [RFC 3611](#), November 2003.

## **[9.2.](#) Informative References**

- [I-D.ietf-rtcweb-rtp-usage]  
    , C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", October 2011.
- [I-D.ietf-rtcweb-security]  
    Rescorla, E., "Security Considerations for RTC-Web", October 2011.
- [I-D.ietf-rtcweb-use-cases-and-requirements]  
    Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use-cases and Requirements", November 2011.
- [I.D-ietf-avtcore-monarch]  
    Wu, Q., Ed., Hunt, G., and P. Arden, "Monitoring Architecture for RTP", February 2012.
- [I.D-ietf-xrblock-rtcp-xr-burst-gap-discard]  
    Hunt, G., Clark, A., and Q. Wu, Ed., "RTCP XR Report Block for Burst/Gap Discard Metric Reporting", January 2012.
- [I.D-ietf-xrblock-rtcp-xr-burst-gap-loss]  
    Hunt, G., Clark, Alan., Zhang, S., Ed., Zhao, J., and Q. Wu, "RTCP XR Report Block for Burst/Gap Loss Metric Reporting", January 2012.
- [I.D-ietf-xrblock-rtcp-xr-delay]  
    Hunt, G., Clark, A., Gross, K., and Q. Wu, "RTCP XR Report Block for Delay Metric Reporting".
- [I.D-ietf-xrblock-rtcp-xr-discard]  
    Hunt, G., Clark, A., Zorn, G., and Q. Wu, "RTCP XR Report Block for Discard Metric Reporting".





Authors' Addresses

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

Email: Rachel@huawei.com

Roni Even  
Huawei  
14 David Hamelech  
Tel Aviv 64953  
Israel

Email: roni.even@huawei.com

