

Internet Engineering Task Force (IETF)
Request for Comments: 6035
Category: Standards Track
ISSN: 2070-1721

A. Pendleton
A. Clark
Telchemy Incorporated
A. Johnston
Avaya
H. Sinnreich
Unaffiliated
November 2010

Session Initiation Protocol Event Package for Voice Quality Reporting

Abstract

This document defines a Session Initiation Protocol (SIP) event package that enables the collection and reporting of metrics that measure the quality for Voice over Internet Protocol (VoIP) sessions. Voice call quality information derived from RTP Control Protocol Extended Reports (RTCP-XR) and call information from SIP is conveyed from a User Agent (UA) in a session, known as a reporter, to a third party, known as a collector. A registration for the application/ vq-rtcpxr media type is also included.

Status of This Memo

This is an Internet Standards Track document.

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). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6035>.

Copyright Notice

Copyright (c) 2010 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
1.1. Applicability Statement	4
1.2. Use of the Mechanism	4
2. Terminology	6
3. SIP Events for VoIP Quality Reporting	6
3.1. SUBSCRIBE NOTIFY Method	6
3.2. PUBLISH Method	7
3.3. Multi-Party and Multi-Segment Calls	7
3.4. Overload Avoidance	7
4. Event Package Formal Definition	8
4.1. Event Package Name	8
4.2. Event Package Parameters	8
4.3. SUBSCRIBE Bodies	8
4.4. Subscribe Duration	8
4.5. NOTIFY Bodies	8
4.6. Voice Quality Event and Semantics	10
4.6.1. ABNF Syntax Definition	10
4.6.2. Parameter Definitions and Mappings	21
4.7. Message Flow and Syntax Examples	29
4.7.1. End of Session Report Using NOTIFY	29
4.7.2. Midsession Threshold Violation Using NOTIFY	32
4.7.3. End of Session Report Using PUBLISH	35
4.7.4. Alert Report Using PUBLISH	37
4.8. Configuration Dataset for vq-rtcpxr Events	39
5. IANA Considerations	39
5.1. SIP Event Package Registration	39
5.2. application/vq-rtcpxr Media Type Registration	39
6. Security Considerations	40
7. Contributors	40
8. References	40
8.1. Normative References	40
8.2. Informative References	41

1. Introduction

Real-time communications over IP networks use SIP for signaling with RTP/RTCP for media transport and reporting, respectively. These protocols are very flexible and can support an extremely wide spectrum of usage scenarios. For this reason, extensions to these protocols must be specified in the context of a specific usage scenario. In this memo, extensions to SIP are proposed to support the reporting of RTP Control Protocol Extended Reports [4] metrics.

1.1. Applicability Statement

RTP is utilized in many different architectures and topologies. RFC 5117 [13] lists and describes the following topologies: point-to-point, point-to-multipoint using multicast, point-to-multipoint using the translator from RFC 3550, point-to-multipoint using the mixer model from RFC 3550, point-to-multipoint using video-switching Multipoint Control Units (MCUs), point-to-multipoint using RTCP-terminating MCU, and non-symmetric mixer/translators. As the Abstract of this document points out, this specification is for reporting quality of Voice over Internet Protocol (VoIP) sessions. As such, only the first topology, point to point, is currently supported by this specification. This reflects both current VoIP deployments, which are predominantly point to point using unicast, and the state of research in the area of quality.

How to accurately report the quality of a multipart conference or a session involving multiple hops through translators and mixers is currently an area of research in the industry. However, this mechanism can easily be used for centrally mixed conference calls, in which each leg of the conferences is just a point-to-point call. This mechanism could be extended to cover additional RTP topologies in the future once these topics progress out of the realm of research and into actual Internet deployments.

1.2. Use of the Mechanism

RTCP reports are usually sent to other participating endpoints in a session. This can make the collection of performance information by an administrator or management system quite complex to implement. In the usage scenarios addressed in this memo, the data contained in RTCP XR VoIP metrics reports (RFC 3611 [4]) are forwarded to a central collection server systems using SIP.

Applications residing in the server or elsewhere can aid in network management to alleviate bandwidth constraints and also to support customer service by identifying and acknowledging calls of poor quality. However, specifying such applications is beyond the scope of this paper.

There is a large portfolio of quality parameters that can be associated with VoIP, but only a minimal necessary number of parameters are included on the RTCP-XR reports:

1. The codec type, as resulting from the Session Description Protocol (SDP) offer-answer negotiation in SIP,
2. The burst gap loss density and max gap duration, since voice cut-outs are the most annoying quality impairment in VoIP,
3. Round-trip delay, because it is critical to conversational quality,
4. Conversational quality as a catch-all for other voice quality impairments, such as randomly distributed packet loss, jitter, annoying silent suppression effects, etc.

In specific usage scenarios where other parameters are required, designers can include other parameters beyond the scope of this paper.

RTCP reports are best effort only, and though they are very useful, they have a number of limitations as discussed in [3]. This must be considered when using RTCP reports in managed networks.

This document defines a new SIP event package, vq-rtcpxr, and a new MIME type, application/vq-rtcpxr, that enable the collection and reporting of metrics that measure quality for RTP [3] sessions. The definitions of the metrics used in the event package are based on RTCP Extended Reports [4] and RTCP [3]; a mapping between the SIP event parameters and the parameters within the aforementioned RFCs is defined within this document in Section 4.6.2.

Monitoring of voice quality is believed to be the highest priority for usage of this mechanism, and as such, the metrics in the event package are largely tailored for voice quality measurements. The event package is designed to be extensible. However, the negotiation of such extensions is not defined in this document.

The event package supports reporting the voice quality metrics for both the inbound and outbound directions. Voice quality metrics for the inbound direction can generally be computed locally by the

reporting endpoint; however, voice quality metrics for the outbound direction are computed by the remote endpoint and sent to the reporting endpoint using the RTCP Extended Reports [4].

The configuration of the usage of this event package is not covered in this document. It is the recommendation of this document that the SIP configuration framework [15] be used. This is discussed in Section 4.8.

The event package SHOULD be used with the SUBSCRIBE/NOTIFY method; however, it MAY also be used with the PUBLISH method [8] for backward compatibility with some existing implementations. Message flow examples for both methods are provided in this document.

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 BCP 14, RFC 2119 [1].

3. SIP Events for VoIP Quality Reporting

This document defines a SIP events package [5] for Voice over IP performance reporting. A SIP UA can send these events to an entity that can make the information available to other applications. For purposes of illustration, the entities involved in SIP vq-rtcpvr event reporting will be referred to as follows:

- o REPORTER: an entity involved in the measurement and reporting of media quality, i.e., the SIP UA involved in a media session.
- o COLLECTOR: an entity that receives SIP vq-rtcpvr events. A COLLECTOR may be a proxy server or another entity that is capable of supporting SIP vq-rtcpvr events.

3.1. SUBSCRIBE NOTIFY Method

The COLLECTOR SHALL send a SUBSCRIBE to the REPORTER to explicitly establish the relationship. The REPORTER SHOULD send the voice quality metric reports using the NOTIFY method. The REPORTER MUST NOT send any vq-rtcpvr events if a COLLECTOR address has not been configured. The REPORTER populates the Request-URI according to the rules for an in-dialog request. The COLLECTOR MAY send a SUBSCRIBE to a SIP Proxy acting on behalf of the reporting SIP UAs.

3.2. PUBLISH Method

A SIP UA that supports this specification MAY also send the service quality metric reports using the PUBLISH method [8]; however, this approach SHOULD NOT be used, in general, on the public Internet. The PUBLISH method MAY be supported for backward compatibility with existing implementations.

The REPORTER MAY therefore populate the Request-URI of the PUBLISH method with the address of the COLLECTOR. To ensure security of SIP proxies and the COLLECTOR, the REPORTER MUST be configured with the address of the COLLECTOR, preferably using the SIP UA configuration framework [15], as described in Section 5.8.

It is RECOMMENDED that the REPORTER send an OPTIONS message to the COLLECTOR to ensure support of the PUBLISH message.

If PUBLISH is not supported, then the REPORTER can only wait for a SUBSCRIBE request from the COLLECTOR and then deliver the information in NOTIFYs. If a REPORTER sends a PUBLISH to a COLLECTOR that does not support or allow this method, a 501 Not Implemented or a 405 Method Not Allowed response will be received, and the REPORTER will stop publication.

3.3. Multi-Party and Multi-Segment Calls

A voice quality metric report may be sent for each session terminating at the REPORTER, and it may contain multiple report bodies. For a multi-party call, the report MAY contain report bodies for the session between the reporting endpoint and each remote endpoint for which there was an RTP session during the call.

Multi-party services such as call hold and call transfer can result in the user participating in a series of concatenated sessions, potentially with different choices of codec or sample rate, although these may be perceived by the user as a single call. A REPORTER MAY send a voice quality metric report at the end of each session or MAY send a single voice quality metric report containing an application/vq-rtcp body for each segment of the call.

3.4. Overload Avoidance

Users of this extension should ensure that they implement general SIP mechanisms for avoiding overload. For instance, an overloaded proxy or COLLECTOR MUST send a 503 Service Unavailable or other 5xx response with an appropriate Retry-After time specified. REPORTERS MUST act on these responses and respect the Retry-After time

interval. In addition, future SIP extensions to better handle overload as covered in [14] should be followed as they are standardized.

To avoid overload of SIP Proxies or COLLECTORS, it is important to do capacity planning and to minimize the number of reports that are sent.

Approaches to avoiding overload include:

- a. Send only one report at the end of each call.
- b. Use interval reports only on "problem" calls that are being closely monitored.
- c. Limit the number of alerts that can be sent to a maximum of one per call.

4. Event Package Formal Definition

4.1. Event Package Name

This document defines a SIP Event Package. SIP Event Packages were originally defined in RFC 3265 [5].

4.2. Event Package Parameters

No event package parameters are defined.

4.3. SUBSCRIBE Bodies

SUBSCRIBE bodies are described by this specification.

4.4. Subscribe Duration

Subscriptions to this event package MAY range from minutes to weeks. Subscriptions in hours or days are more typical and are RECOMMENDED. The default subscription duration for this event package is one hour.

4.5. NOTIFY Bodies

There are three notify bodies: a Session report, an Interval report, and an Alert report.

The Session report SHOULD be used for reporting when a voice media session terminates, when a media change occurs, such as a codec change or a session fork, or when a session terminates due to no media packets being received and MUST NOT be used for reporting at

arbitrary points in time. This report MUST be used for cumulative metric reporting and the report timestamps MUST be from the start of a media session to the time at which the report is generated.

The Interval report SHOULD be used for periodic or interval reporting and MUST NOT be used for reporting of the complete media session. This report is intended to capture short duration metric reporting and the report intervals SHOULD be non-overlapping time windows.

The Alert report MAY be used when voice quality degrades during a session. The time window to which an Alert report relates MAY be a short time interval or from the start of the call to the point the alert is generated; this time window SHOULD be selected to provide the most useful information to support problem diagnosis.

Session, Interval, and Alert reports MUST populate the metrics with values that are measured over the interval explicitly defined by the "start" and "stop" timestamps.

Voice quality summary reports reference only one codec (payload type). This payload type SHOULD be the main voice payload, not comfort noise or telephone event payloads. For applications that consistently and rapidly switch codecs, the most used codec should be reported. All values in the report, such as IP addresses, synchronization source (SSRC), etc., represent those values as received by the REPORTER. In some scenarios, these may not be the same on either end of the session -- the COLLECTOR will need logic to be able to put these sessions together. The values of parameters such as sample rate, frame duration, frame octets, packets per second, round-trip delay, etc., depend on the type of report in which they are present. If present in a Session or an Interval report, they represent average values over the session or interval. If present in an Alert report, they represent instantaneous values.

The REPORTER always includes local quality reporting information and should, if possible, share remote quality reporting information to the COLLECTOR. This remote quality could be available from received RTCP-XR reports or other sources. Reporting this is useful in cases where the other end might support RTCP-XR but not this voice quality reporting.

This specification defines a new MIME type, application/vq-rtcpxr, which is a text encoding of the RTCP and RTCP-XR statistics with some additional metrics and correlation information.

4.6. Voice Quality Event and Semantics

This section describes the syntax extensions required for event publication in SIP. The formal syntax definitions described in this section are expressed in the Augmented BNF [6] format used in SIP [2] and contain references to elements defined therein.

Additionally, the definition of the timestamp format is provided in [7]. Note that most of the parameters are optional. In practice, most implementations will send a subset of the parameters. It is not the intention of this document to define what parameters may or may not be useful for monitoring the quality of a voice session, but to enable reporting of voice quality. As such, the syntax allows the implementer to choose which metrics are most appropriate for their solution. As there are no "invalid", "unknown", or "not applicable" values in the syntax, the intention is to exclude any parameters for which values are not available, not applicable, or unknown.

The authors recognize that implementers may need to add new parameter lines to the reports and new metrics to the existing parameter lines. The extension tokens are intended to fulfill this need.

4.6.1. ABNF Syntax Definition

```
VQReportEvent = AlertReport / SessionReport / IntervalReport
```

```
SessionReport = "VQSessionReport" [ HCOLON "CallTerm" ] CRLF
                SessionInfo CRLF
                LocalMetrics [ CRLF RemoteMetrics ]
                [ CRLF DialogID ]
```

; CallTerm indicates the final report of a session.

```
IntervalReport = "VQIntervalReport" [ HCOLON "CallTerm" ] CRLF
                SessionInfo CRLF
                LocalMetrics [ CRLF RemoteMetrics ]
                [ CRLF DialogID ]
```

```
LocalMetrics = "LocalMetrics" HCOLON CRLF Metrics
```

```
RemoteMetrics = "RemoteMetrics" HCOLON CRLF Metrics
```

```
AlertReport = "VQAlertReport" HCOLON
              MetricType WSP Severity WSP Direction CRLF
              SessionInfo CRLF
              LocalMetrics [ CRLF RemoteMetrics ]
              [ DialogID ]
```

```
SessionInfo =
  CallID CRLF
  LocalID CRLF
  RemoteID CRLF
  OrigID CRLF
  LocalAddr CRLF
  RemoteAddr CRLF
  LocalGroupID CRLF
  RemoteGroupID CRLF
  [ LocalMACAddr CRLF ]
  [ RemoteMACAddr CRLF ]

Metrics = TimeStamps CRLF
  [ SessionDescription CRLF ]
  [ JitterBuffer CRLF ]
  [ PacketLoss CRLF ]
  [ BurstGapLoss CRLF ]
  [ Delay CRLF ]
  [ Signal CRLF ]
  [ QualityEstimates CRLF ]
  *(Extension CRLF)

; Timestamps are provided in Coordinated Universal Time (UTC)
; using the ABNF format provided in RFC 3339,
; "Date and Time on the Internet: Timestamps"
; These timestamps SHOULD reflect, as closely as
; possible, the actual time during which the media session
; was running to enable correlation to events occurring
; in the network infrastructure and to accounting records.
; Time zones other than "Z" are not allowed.

TimeStamps = "Timestamps" HCOLON StartTime WSP StopTime
StartTime = "START" EQUAL date-time
StopTime = "STOP" EQUAL date-time

; SessionDescription provides a shortened version of the
; session SDP but contains only the relevant parameters for
; session quality reporting purposes.

SessionDescription = "SessionDesc" HCOLON
  [ PayloadType WSP ]
  [ PayloadDesc WSP ]
  [ SampleRate WSP ]
  [ PacketsPerSecond WSP ]
  [ FrameDuration WSP ]
  [ FrameOctets WSP ]
  [ FramesPerPacket WSP ]
  [ FmtpOptions WSP ]
```

```
[ PacketLossConcealment WSP ]
[ SilenceSuppressionState ]
*(WSP Extension)

; PayloadType provides the PT parameter used in the RTP packets.

PayloadType = "PT" EQUAL (1*3DIGIT)

; PayloadDesc provides a text description of the codec.
; This parameter SHOULD use the IANA registry for
; media-type names defined by RFC 4855 where it unambiguously
; defines the codec. Refer to the "Audio Media Types"
; registry on http://www.iana.org.

PayloadDesc = "PD" EQUAL (word / DQUOTE word-plus DQUOTE)

; SampleRate reports the rate at which a voice was sampled
; in the case of narrowband codecs, this value will typically
; be 8000.
; For codecs that are able to change sample rates, the lowest and
; highest sample rates MUST be reported (e.g., 8000;16000).

SampleRate = "SR" EQUAL (1*6DIGIT) *(SEMI (1*66DIGIT))

; FrameDuration can be combined with the FramesPerPacket
; to determine the packetization rate; the units for
; FrameDuration are milliseconds. NOTE: for frame-based codecs,
; each frame constitutes a single frame; for sample-based codecs,
; a "frame" refers to the set of samples carried in an RTP packet.

FrameDuration = "FD" EQUAL (1*4DIGIT)

; FrameOctets provides the number of octets in each frame
; at the time the report is generated (i.e., last value).
; This MAY be used where FrameDuration is not available.
; NOTE: for frame-based codecs, each frame constitutes a single frame;
; for sample-based codecs, a "frame" refers to the set of samples
; carried in an RTP packet.

FrameOctets = "FO" EQUAL (1*5DIGIT)

; FramesPerPacket provides the number of frames in each RTP
; packet at the time the report is generated.
; NOTE: for frame-based codecs, each frame constitutes a single frame;
; for sample-based codecs, a "frame" refers to the set of samples
; carried in an RTP packet.

FramesPerPacket = "FPP" EQUAL (1*2DIGIT)
```

; Packets per second provides the average number of packets
; that are transmitted per second, as at the time the report is
; generated.

PacketsPerSecond = "PPS" EQUAL (1*5DIGIT)

; FMTP options from SDP. Note that the parameter is delineated
; by " " to avoid parsing issues in transitioning between SDP
; and SIP parsing.

FmtpOptions = "FMTP" EQUAL DQUOTE word-plus DQUOTE

; PacketLossConcealment indicates whether a PLC algorithm was
; or is being used for the session. The values follow the same
; numbering convention as RFC 3611 [4].
; 0 - unspecified
; 1 - disabled
; 2 - enhanced
; 3 - standard

PacketLossConcealment = "PLC" EQUAL ("0" / "1" / "2" / "3")

; SilenceSuppressionState indicates whether silence suppression,
; also known as Voice Activity Detection (VAD) is enabled.

SilenceSuppressionState = "SSUP" EQUAL ("on" / "off")

; CallId provides the call id from the SIP dialog.

CallID = "CallID" HCOLON Call-ID-Parm

; LocalID identifies the reporting endpoint for the media session [2].

LocalID = "LocalID" HCOLON (name-addr/addr-spec)

; RemoteID identifies the remote endpoint of the media session [2].

RemoteID = "RemoteID" HCOLON (name-addr/addr-spec)

; OrigID identifies the endpoint which originated the session.

OrigID = "OrigID" HCOLON (name-addr/addr-spec)

; LocalAddr provides the IP address, port, and SSRC of the
; endpoint/UA, which is the receiving end of the stream being
; measured.

LocalAddr = "LocalAddr" HCOLON IPAddress WSP Port WSP Ssrc

```
; RemoteAddr provides the IP address, port, and SSRC of the
; the source of the stream being measured.

RemoteAddr = "RemoteAddr" HCOLON IPAddress WSP Port WSP Ssrc

; LocalMACAddr provides the Media Access Control (MAC) address
; of the local SIP device.

LocalMACAddr = "LocalMAC" HCOLON hex2 *(":" hex2)

; RemoteMACAddr provides the MAC address
; of the remote SIP device.

RemoteMACAddr = "RemoteMAC" HCOLON hex2 *(":" hex2)

; LocalGroupID provides the identification for the purposes
; of aggregation for the local endpoint.

LocalGroupID = "LocalGroup" HCOLON word-plus

; RemoteGroupID provides the identification for the purposes
; of aggregation for the remote endpoint.

RemoteGroupID = "RemoteGroup" HCOLON word-plus

; For clarification, the LocalAddr in the LocalMetrics report
; MUST be the RemoteAddr in the RemoteMetrics report.

IPAddress = "IP" EQUAL IPv6address / IPv4address
Port = "PORT" EQUAL 1*DIGIT
Ssrc = "SSRC" EQUAL ( %x30.78 1*8HEXDIG)

JitterBuffer = "JitterBuffer" HCOLON
[ JitterBufferAdaptive WSP ]
[ JitterBufferRate WSP ]
[ JitterBufferNominal WSP ]
[ JitterBufferMax WSP ]
[ JitterBufferAbsMax ]
*(WSP Extension)

; JitterBufferAdaptive indicates whether the jitter buffer in
; the endpoint is adaptive, static, or unknown.
; The values follow the same numbering convention as RFC 3611 [4].
; For more details, please refer to that document.
; 0 - unknown
; 1 - reserved
; 2 - non-adaptive
; 3 - adaptive
```

```
JitterBufferAdaptive = "JBA" EQUAL ("0" / "1" / "2" / "3")

; JitterBuffer metric definitions are provided in RFC 3611 [4].

JitterBufferRate      = "JBR" EQUAL (1*2DIGIT) ;0-15
JitterBufferNominal   = "JBN" EQUAL (1*5DIGIT) ;0-65535
JitterBufferMax       = "JBM" EQUAL (1*5DIGIT) ;0-65535
JitterBufferAbsMax    = "JBX" EQUAL (1*5DIGIT) ;0-65535

; PacketLoss metric definitions are provided in RFC 3611 [4].

PacketLoss = "PacketLoss" HCOLON
    [ NetworkPacketLossRate WSP ]
    [ JitterBufferDiscardRate ]
    *(WSP Extension)

NetworkPacketLossRate =
    "NLR" EQUAL (1*3DIGIT [ "." 1*2DIGIT ]) ;percentage

JitterBufferDiscardRate =
    "JDR" EQUAL (1*3DIGIT [ "." 1*2DIGIT ]) ;percentage

; BurstGapLoss metric definitions are provided in RFC 3611 [4].

BurstGapLoss = "BurstGapLoss" HCOLON
    [ BurstLossDensity WSP ]
    [ BurstDuration WSP ]
    [ GapLossDensity WSP ]
    [ GapDuration WSP ]
    [ MinimumGapThreshold ]
    *(WSP Extension)

BurstLossDensity =
    "BLD" EQUAL (1*3DIGIT [ "." 1*2DIGIT ]) ;percentage

BurstDuration =
    "BD" EQUAL (1*7DIGIT) ;0-3,600,000 -- milliseconds

GapLossDensity =
    "GLD" EQUAL (1*3DIGIT [ "." 1*2DIGIT ]) ;percentage

GapDuration =
    "GD" EQUAL (1*7DIGIT) ;0-3,600,000 -- milliseconds
```

```
MinimumGapThreshold =
  "GMIN" EQUAL (1*3DIGIT) ;1-255

Delay = "Delay" HCOLON
  [ RoundTripDelay WSP ]
  [ EndSystemDelay WSP ]
  [ OneWayDelay WSP ]
  [ SymmOneWayDelay WSP ]
  [ InterarrivalJitter WSP ]
  [ MeanAbsoluteJitter ]
  *(WSP Extension)

; RoundTripDelay SHALL be measured as defined in RFC 3550 [3].

RoundTripDelay = "RTD" EQUAL (1*5DIGIT) ;0-65535

; EndSystemDelay metric is defined in RFC 3611 [4].

EndSystemDelay = "ESD" EQUAL (1*5DIGIT) ;0-65535

; OneWayDelay is defined in RFC 2679 [12].

OneWayDelay = "OWD" EQUAL (1*5DIGIT) ;0-65535

; SymmOneWayDelay is defined as half the sum of RoundTripDelay
; and the EndSystemDelay values for both endpoints.

SymmOneWayDelay = "SOWD" EQUAL (1*5DIGIT); 0-65535

; Interarrival Jitter is calculated as defined RFC 3550 [3]
; and converted into milliseconds.

InterarrivalJitter = "IAJ" EQUAL (1*5DIGIT) ;0-65535 ms

; Mean Absolute Jitter is measured as defined
; by ITU-T G.1020 [9] where it is known as MAPDV.

MeanAbsoluteJitter = "MAJ" EQUAL (1*5DIGIT);0-65535

; Signal metrics definitions are provided in RFC 3611 [4].

Signal = "Signal" HCOLON
  [ SignalLevel WSP ]
  [ NoiseLevel WSP ]
  [ ResidualEchoReturnLoss ]
  *(WSP Extension)

; SignalLevel will normally be a negative value.
```



```
; The absence of the negative sign indicates a positive value.  
; Where the signal level is negative, the sign MUST be  
; included. This metric applies to the speech signal decoded  
; from the received packet stream.
```

```
SignalLevel = "SL" EQUAL ([ "-" ] 1*2DIGIT)
```

```
; NoiseLevel will normally be negative and the sign MUST be  
; explicitly included.  
; The absence of a sign indicates a positive value.  
; This metric applies to the speech signal decoded from the  
; received packet stream.
```

```
NoiseLevel = "NL" EQUAL ([ "-" ] 1*2DIGIT)
```

```
; Residual Echo Return Loss (RERL) is the ratio between  
; the original signal and the echo level as measured after  
; echo cancellation or suppression has been applied.  
; Expressed in decibels (dB). This is typically a positive  
; value.  
; This metric relates to the proportion of the speech signal  
; decoded from the received packet stream that is reflected  
; back in the encoded speech signal output in the transmitted  
; packet stream (i.e., will affect the REMOTE user's  
; conversational quality). To support the diagnosis of echo-  
; related problems experienced by the local user of the device  
; generating a report according to this document, the value of  
; RERL reported via the RTCP XR VoIP Metrics payload SHOULD be  
; reported in the RemoteMetrics set of data.
```

```
ResidualEchoReturnLoss = "RERL" EQUAL (1*3DIGIT)
```

```
; Voice Quality estimation metrics.  
; Each quality estimate has an optional associated algorithm.  
; These fields permit the implementation to use a variety  
; of different calculation methods for each type of metric.
```

```
QualityEstimates = "QualityEst" HCOLON
```

```
  [ ListeningQualityR WSP ]  
  [ RLQEstAlg WSP ]  
  [ ConversationalQualityR WSP ]  
  [ RCQEstAlg WSP ]  
  [ ExternalR-In WSP ]  
  [ ExtRInEstAlg WSP ]  
  [ ExternalR-Out WSP ]  
  [ ExtROutEstAlg WSP ]  
  [ MOS-LQ WSP ]  
  [ MOSLQEstAlg WSP ]
```

```
[ MOS-CQ WSP ]
[ MOSCQEstAlg WSP ]
[ QoEEstAlg ]
*(WSP Extension)

ListeningQualityR = "RLQ" EQUAL (1*3DIGIT) ; 0 - 120

RLQEstAlg = "RLQEstAlg" EQUAL word ; "P.564" [10], or other

ConversationalQualityR = "RCQ" EQUAL (1*3DIGIT) ; 0 - 120

RCQEstAlg = "RCQEstAlg" EQUAL word ; "P.564", or other

; ExternalR-In is measured by the local endpoint for incoming
; connection on the "other" side of this endpoint.  For example,
;   Phone A <---> Bridge <----> Phone B
;   ListeningQualityR = quality for Phone A ----> Bridge path
;   ExternalR-In = quality for Bridge <----> Phone B path

ExternalR-In = "EXTRI" EQUAL (1*3DIGIT) ; 0 - 120

ExtRIInEstAlg = "ExtRIInEstAlg" EQUAL word ; "P.564" or other

; ExternalR-Out is copied from the RTCP XR message received from the
; remote endpoint on the "other" side of this endpoint.  For example,
;   Phone A <---> Bridge <----> Phone B
;   ExternalR-Out = quality for Bridge <----> Phone B path

ExternalR-Out = "EXTRO" EQUAL (1*3DIGIT) ; 0 - 120

ExtROOutEstAlg = "ExtROOutEstAlg" EQUAL word ; "P.564" or other

MOS-LQ = "MOSLQ" EQUAL (DIGIT [ "." 1*3DIGIT ]) ; 0.0 - 4.9

MOSLQEstAlg = "MOSLQEstAlg" EQUAL word ; "P.564" or other

MOS-CQ = "MOSCQ" EQUAL (DIGIT [ "." 1*3DIGIT ]) ; 0.0 - 4.9

MOSCQEstAlg = "MOSCQEstAlg" EQUAL word ; "P.564" or other

; QoEEstAlg provides an alternative to the separate
; estimation algorithms for use when the same algorithm
; is used for all measurements.
```

```
QoEEstAlg = "QoEEstAlg" EQUAL word ; "P.564" or other

; DialogID provides the identification of the dialog with
; which the media session is related. This value is taken
; from the SIP header.

DialogID = "DialogID" COLON Call-ID-Parm *(SEMI did-parm)

did-parm = to-tag / from-tag / word

to-tag = "to-tag" EQUAL token

from-tag = "from-tag" EQUAL token

; MetricType provides the metric on which a notification of
; threshold violation was based. The more commonly used metrics
; for alerting purposes are included here explicitly, using the
; character encoding that represents the parameter in
; this ABNF. The Extension parameter can be used to provide
; metrics that are not defined by this document.

MetricType = "Type" EQUAL "RLQ" / "RCQ" / "EXTR" /
  "MOSLQ" / "MOSQC" /
  "BD" / "NLR" / "JDR" /
  "RTD" / "ESD" / "IAJ" /
  "RERL" / "SL" / "NL" / Extension

Direction = "Dir" EQUAL "local" / "remote"
Severity = "Severity" EQUAL "Warning" / "Critical" /
  "Clear"

Call-ID-Parm = word [ "@" word ]

; General ABNF notation from RFC 5234.

CRLF = %x0D.0A
DIGIT = %x30-39
WSP = SP / HTAB ; white space
SP = " "
HTAB = %x09 ; horizontal tab
HEXDIG = DIGIT / "A" / "B" / "C" / "D" / "E" / "F" /
  "a" / "b" / "c" / "d" / "e" / "f"
DQUOTE = %x22 ; " (Double Quote)
ALPHA = %x41-5A / %x61-7A ; A-Z / a-z
```

; ABNF notation from RFC 3261.

```

alphanum = ALPHA / DIGIT
LWS      = [ *WSP CRLF ] 1*WSP ; linear whitespace
SWS      = [ LWS ] ; sep whitespace
SEMI     = SWS ";" SWS ; semicolon
EQUAL    = SWS "=" SWS ; equal
COLON    = SWS ":" SWS ; colon
HCOLON   = *( SP / HTAB ) ":" SWS

token     = 1*(alphanum / "-" / "." / "!" / "%" / "*" /
              / "_" / "+" / "\" / "'" / "~" )

IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6address = hexpart [ ":" IPv4address ]
hexpart     = hexseq / hexseq ":" [ hexseq ] / ":"
              [ hexseq ]
hexseq      = hex4 *( ":" hex4 )
hex4        = 1*4HEXDIG
hex2        = 2HEXDIG

```

; ABNF notation from RFC 3339.

```

date-fullyear = 4DIGIT ; e.g. 2006
date-month   = 2DIGIT ; e.g. 01 or 11
date-mday    = 2DIGIT ; e.g. 02 or 22
time-hour    = 2DIGIT ; e.g. 01 or 13
time-minute  = 2DIGIT ; e.g. 03 or 55
time-second  = 2DIGIT ; e.g. 01 or 59
time-secfrac = "." 1*DIGIT
time-numoffset = ("+" / "-") time-hour ":" time-minute
time-offset   = "Z" / time-numoffset
partial-time  = time-hour ":" time-minute ":" time-second [ time-secfrac ]
full-date     = date-fullyear "-" date-month "-" date-mday
full-time     = partial-time time-offset
date-time     = full-date "T" full-time

```

; Miscellaneous definitions
;

Extension = word-plus

```

word = 1*(alphanum / "-" / "." / "!" / "%" / "*" /
          / "_" / "+" / "\" / "'" / "~" /
          / "(" / ")" / "<" / ">" /
          / ":" / "\" / DQUOTE /
          / "/" / "[" / "]" / "?" )

```

```

word-plus = 1*(alphanum / "-" / "." / "!" / "%" / "*" /
  "_" / "+" / "\" / "'" / "~" /
  "(" / ")" / "<" / ">" / ":" /
  "\" / "/" / "[" / "]" / "?" /
  "{" / "}" / "=" / " ")

```

4.6.2. Parameter Definitions and Mappings

Parameter values, codec types, and other aspects of the endpoints may change dynamically during a session. The reported values of metrics and configuration parameters SHALL be the current value at the time the report is generated.

The Packet Loss Rate and Packet Discard Rate parameters are calculated over the period between the starting and ending timestamps for the report. These are normally calculated from a count of the number of lost or discarded packets divided by the count of the number of packets, and hence are based on the current values of these counters at the time the report was generated.

Packet delay variation, signal level, noise level, and echo level are computed as running or interval averages, based on the appropriate standard, e.g., RFC 3550 for Packet Delay Variation (PDV), and the sampled value of these running averages is reported. Delay, packet size, jitter buffer size, and codec-related data may change during a session and the current value of these parameters is reported as sampled at the time the report is generated.

4.6.2.1. General Mapping Percentages from 8-bit, Fixed-Point Numbers

RFC 3611 uses an 8-bit, fixed-point number with the binary point at the left edge of the field. This value is calculated by dividing the total number of packets lost by the total number of packets expected and multiplying the result by 256, and then taking the integer part.

For any RTCP XR parameter in this format, to map into the equivalent SIP vq-rtcpxr parameter, simply reverse the equation, i.e., divide by 256 and take the integer part.

4.6.2.2. Timestamps

Following SIP and other IETF conventions, timestamps are provided in Coordinated Universal Time (UTC) using the ABNF format provided in RFC 3339 [7]. These timestamps SHOULD reflect, as closely as possible, the actual time during which the media session was running to enable correlation to related events occurring in the network and to accounting or billing records.

4.6.2.3. SessionDescription

The parameters in this field provide a shortened version of the session SDP(s), containing only the relevant parameters for session quality reporting purposes. Where values may change during a session, for example, a codec may change rate, then the most-recent value of the parameter is reported.

4.6.2.3.1. Payload Type

This is the "payload type" parameter used in the RTP packets, i.e., the codec. This field can also be mapped from the SDP "rtpmap" attribute field "payload type". IANA-registered types SHOULD be used.

4.6.2.3.2. Payload Desc

This parameter is a text description of the codec. This parameter SHOULD use the IANA registry for media-type names where it unambiguously defines the codec. Refer to the "Audio Media Types" registry on <http://www.iana.org>.

4.6.2.3.3. Sample Rate

This parameter is mapped from the SDP "rtpmap" attribute field "clock rate". The field provides the rate at which a voice was sampled, measured in Hertz (Hz).

4.6.2.3.4. Packets Per Second

This parameter is not contained in RTP or SDP but can usually be obtained from the device codec. Packets per second provides the (rounded) number of RTP packets that are transmitted per second.

4.6.2.3.5. Frame Duration

This parameter is not contained in RTP or SDP but can usually be obtained from the device codec. The field reflects the amount of voice content in each frame within the RTP payload, measured in milliseconds. Note that this value can be combined with the FramesPerPacket to determine the packetization rate. Also, where a sample-based codec is used, a "frame" refers to the set of samples carried in an RTP packet.

4.6.2.3.6. Frame Octets

This parameter is not contained in RTP or SDP but is usually provided by the device codec. The field provides the number of octets in each frame within the RTP payload. This field is usually not provided when the FrameDuration is provided. Also, where a sample-based codec is used, a "frame" refers to the set of samples carried in an RTP packet.

4.6.2.3.7. Frames Per Packet

This parameter is not contained in RTP or SDP but can usually be obtained from the device codec. This field provides the number of frames in each RTP packet. Note that this value can be combined with the FrameDuration to determine the packetization rate. Also, where a sample-based codec is used, a "frame" refers to the set of samples carried in an RTP packet.

4.6.2.3.8. FMTP Options

This parameter is taken directly from the SDP attribute "fmtp" defined in RFC 4566.

4.6.2.3.9. Silence Suppression State

This parameter does not correspond to SDP, RTP, or RTCP XR. It indicates whether silence suppression, also known as Voice Activity Detection (VAD), is enabled for the identified session.

4.6.2.3.10. Packet Loss Concealment

This value corresponds to "PLC" in RFC 3611 in the VoIP Metrics Report Block. The values defined by RFC 3611 are reused by this recommendation and therefore no mapping is required.

4.6.2.4. LocalAddr

This field provides the IP address, port, and synchronization source (SSRC) for the session from the perspective of the endpoint that is measuring performance. The IPAddress MAY be in IPv4 or IPv6 format. The SSRC is taken from SDP, RTCP, or RTCP XR input parameters.

In the presence of NAT and where a NAT-traversal mechanism such as Session Traversal Utilities for NAT (STUN) [16] is used, the external IP address can be reported, since the internal IP address is not visible to the network operator.

4.6.2.5. RemoteAddr

This field provides the IP address, port, and SSRC of the session peer from the perspective of the remote endpoint measuring performance. In the presence of NAT and where a NAT-traversal mechanism such as STUN [16] is used, the external IP address can be reported, since the internal IP address is not visible to the network operator.

4.6.2.6. Jitter Buffer Parameters

4.6.2.6.1. Jitter Buffer Adaptive

This value corresponds to "JBA" in RFC 3611 in the VoIP Metrics Report Block. The values defined by RFC 3611 are unchanged and therefore no mapping is required.

4.6.2.6.2. Jitter Buffer Rate

This value corresponds to "JB rate" in RFC 3611 in the VoIP Metrics Report Block. The parameter does not require any conversion.

4.6.2.6.3. Jitter Buffer Nominal

This value corresponds to "JB nominal" in RFC 3611 in the VoIP Metrics Report Block. The parameter does not require any conversion.

4.6.2.6.4. Jitter Buffer Max

This value corresponds to "JB maximum" in RFC 3611 in the VoIP Metrics Report Block. The parameter does not require any conversion.

4.6.2.6.5. Jitter Buffer Abs Max

This value corresponds to "JB abs max" in RFC 3611 in the VoIP Metrics Report Block. The parameter does not require any conversion.

4.6.2.7. Packet Loss Parameters

4.6.2.7.1. Network Loss Rate

This value corresponds to "loss rate" in RFC 3611 in the VoIP Metrics Report Block. For conversion, see Section 4.6.2.1. A loss rate of 100% MAY be reported if media packets were expected but none had been received at the time of session termination.

4.6.2.7.2. Jitter Buffer Discard Rate

This value corresponds to "discard rate" in RFC 3611 in the VoIP Metrics Report Block. For conversion, see Section 4.6.2.1.

4.6.2.8. Burst/Gap Parameters

4.6.2.8.1. Burst Loss Density

This value corresponds to "burst density" in RFC 3611 in the VoIP Metrics Report Block. For conversion, see Section 4.6.2.1.

4.6.2.8.2. Burst Duration

This value corresponds to "burst duration" in RFC 3611 in the VoIP Metrics Report Block. This value requires no conversion; the exact value sent in an RTCP XR VoIP Metrics Report Block can be included in the SIP vq-rtcpxr parameter.

4.6.2.8.3. Gap Loss Density

This value corresponds to "gap density" in RFC 3611 in the VoIP metrics Report Block.

4.6.2.8.4. Gap Duration

This value corresponds to "gap duration" in RFC 3611 in the VoIP Metrics Report Block. This value requires no conversion; the exact value sent in an RTCP XR VoIP Metrics Report Block can be reported.

4.6.2.8.5. Minimum Gap Threshold

This value corresponds to "Gmin" in RFC 3611 in the VoIP Metrics Report Block. This value requires no conversion; the exact value sent in an RTCP XR VoIP Metrics Report Block can be reported.

4.6.2.9. Delay Parameters

4.6.2.9.1. Round-Trip Delay

This value corresponds to "round trip delay" in RFC 3611 in the VoIP Metrics Report Block and may be measured using the method defined in RFC 3550. The parameter is expressed in milliseconds.

4.6.2.9.2. End System Delay

This value corresponds to "end system delay" in RFC 3611 in the VoIP Metrics Report Block. This parameter does not require any conversion. The parameter is expressed in milliseconds.

4.6.2.9.3. Symmetric One-Way Delay

This value is computed by adding Round-Trip Delay to the local and remote End System Delay and dividing by two.

4.6.2.9.4. One-Way Delay

This value SHOULD be measured using the methods defined in IETF RFC 2679 [12]. The parameter is expressed in milliseconds.

4.6.2.9.5. Inter-Arrival Jitter

Inter-arrival jitter is calculated as defined in RFC 3550 and converted into milliseconds.

4.6.2.9.6. Mean Absolute Jitter

It is recommended that MAJ be measured as defined in ITU-T G.1020 [9]. This parameter is often referred to as MAPDV (Mean Absolute Packet Delay Variation). The parameter is expressed in milliseconds.

4.6.2.10. Signal-Related Parameters

4.6.2.10.1. Signal Level

This field corresponds to "signal level" in RFC 3611 in the VoIP Metrics Report Block. This field provides the voice signal relative level is defined as the ratio of the signal level to a 0 dBm0 reference, expressed in decibels. This value can be used directly without extra conversion.

4.6.2.10.2. Noise Level

This field corresponds to "noise level" in RFC 3611 in the VoIP Metrics Report Block. This field provides the ratio of the silent period background noise level to a 0 dBm0 reference, expressed in decibels. This value can be used directly without extra conversion.

4.6.2.10.3. Residual Echo Return Loss (RERL)

This field corresponds to "RERL" in RFC 3611 in the VoIP Metrics Report Block. This field provides the ratio between the original signal and the echo level in decibels, as measured after echo cancellation or suppression has been applied. This value can be used directly without extra conversion.

4.6.2.11. Quality Scores

4.6.2.11.1. ListeningQualityR

This field reports the listening quality expressed as an R factor (per G.107). This does not include the effects of echo or delay. The range of R is 0-95 for narrowband calls and 0-120 for wideband calls. Algorithms for computing this value SHOULD be compliant with ITU-T Recommendations P.564 [10] and G.107 [11].

4.6.2.11.2. RLQEstAlg

This field provides a text name for the algorithm used to estimate ListeningQualityR. This field will be free form text and not necessarily reflective of any standards or recommendations.

4.6.2.11.3. ConversationalQualityR

This field corresponds to "R factor" in RFC 3611 in the VoIP Metrics Report Block. This parameter provides a cumulative measurement of voice quality from the start of the session to the reporting time. The range of R is 0-95 for narrowband calls and 0-120 for wideband calls. Algorithms for computing this value SHOULD be compliant with ITU-T Recommendations P.564 and G.107. Within RFC 3611, a reported R factor of 127 indicates that this parameter is unavailable; in this case, the ConversationalQualityR parameter MUST be omitted from the vq-rtcpxr event.

4.6.2.11.4. RCQEstAlg

This field provides a text name for the algorithm used to estimate ConversationalQualityR. This field will be free form text and not necessarily reflective of any standards or recommendations.

4.6.2.11.5. ExternalR-In

This field corresponds to "ext. R factor" in RFC 3611 in the VoIP Metrics Report Block. This parameter reflects voice quality as measured by the local endpoint for incoming connection on "other" side (refer to RFC 3611 for a more-detailed explanation). The range

of R is 0-95 for narrowband calls and 0-120 for wideband calls. Algorithms for computing this value SHOULD be compliant with ITU-T Recommendations P.564 and G.107. Within RFC 3611, a reported R factor of 127 indicates that this parameter is unavailable; in this case, the ConversationalQualityR parameter MUST be omitted from the vq-rtcpxr event.

4.6.2.11.6. ExtRInEstAlg

This field provides a text name for the algorithm used to estimate ExternalR-In. This field will be free-form text and not necessarily reflective of any standards or recommendations.

4.6.2.11.7. ExternalR-Out

This field corresponds to "ext. R factor" in RFC 3611 in the VoIP Metrics Report Block. Here, the value is copied from RTCP XR message received from the remote endpoint on the "other" side of this endpoint; refer to RFC 3611 for a more detailed explanation). The range of R is 0-95 for narrowband calls and 0-120 for wideband calls. Algorithms for computing this value SHOULD be compliant with ITU-T Recommendations P.564 and G.107. Within RFC 3611, a reported R factor of 127 indicates that this parameter is unavailable; in this case, the ConversationalQualityR parameter SHALL be omitted from the vq-rtcpxr event.

4.6.2.11.8. ExtROutEstAlg

This field provides a text name for the algorithm used to estimate ExternalR-Out. This field will be free-form text and not necessarily reflective of any standards or recommendations.

4.6.2.11.9. MOS Reporting

Conversion of RFC 3611 reported mean opinion scores (MOSs) for use in reporting MOS-LQ and MOS-CQ MUST be performed by dividing the RFC 3611 reported value by 10 if this value is less than or equal to 50 or omitting the MOS-xQ parameter if the RFC 3611 reported value is 127 (which indicates unavailable).

4.6.2.11.9.1. MOS-LQ

This field corresponds to "MOSLQ" in RFC 3611 in the VoIP Metrics Report Block. This parameter is the estimated mean opinion score for listening voice quality on a scale from 1 to 5, in which 5 represents "Excellent" and 1 represents "Unacceptable". Algorithms for

computing this value SHOULD be compliant with ITU-T Recommendation P.564 [10]. This field provides a text name for the algorithm used to estimate MOS-LQ.

4.6.2.11.9.2. MOS-CQ

This field corresponds to "MOSCQ" in RFC 3611 in the VoIP Metrics Report Block. This parameter is the estimated mean opinion score for conversation voice quality on a scale from 1 to 5, in which 5 represents excellent and 1 represents unacceptable. Algorithms for computing this value SHOULD be compliant with ITU-T Recommendation P.564 with regard to the listening quality element of the computed MOS score.

4.6.2.11.9.3. MOSCQEstAlg

This field provides a text name for the algorithm used to estimate MOS-CQ. This field will be free-form text and not necessarily reflective of any standards or recommendations.

4.6.2.11.10. QoEEstAlg

This field provides a text description of the algorithm used to estimate all voice quality metrics. This parameter is provided as an alternative to the separate estimation algorithms for use when the same algorithm is used for all measurements. This field will be free-form text and not necessarily reflective of any standards or recommendations.

4.7. Message Flow and Syntax Examples

This section shows a number of message flow examples showing how the event package works.

4.7.1. End of Session Report Using NOTIFY

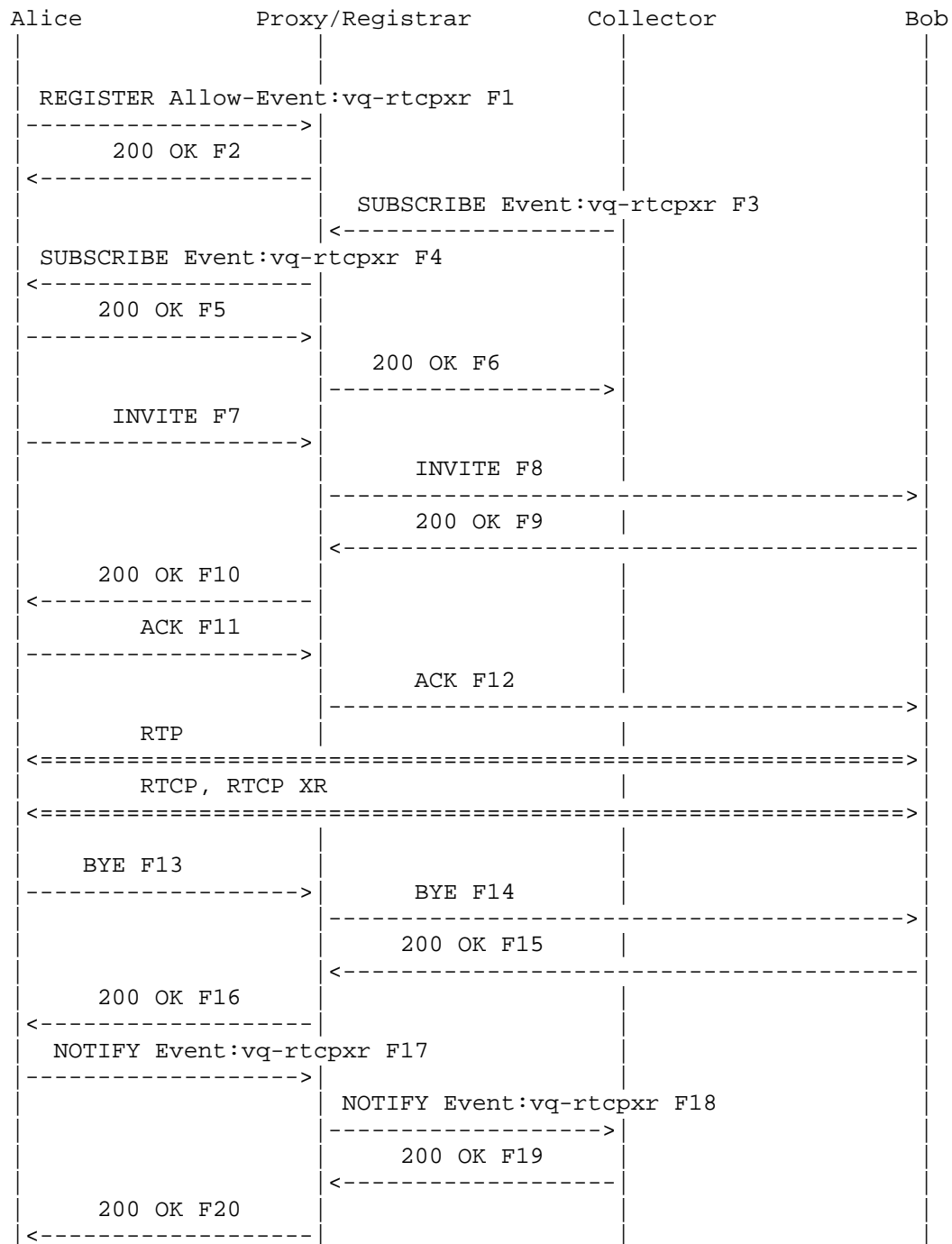


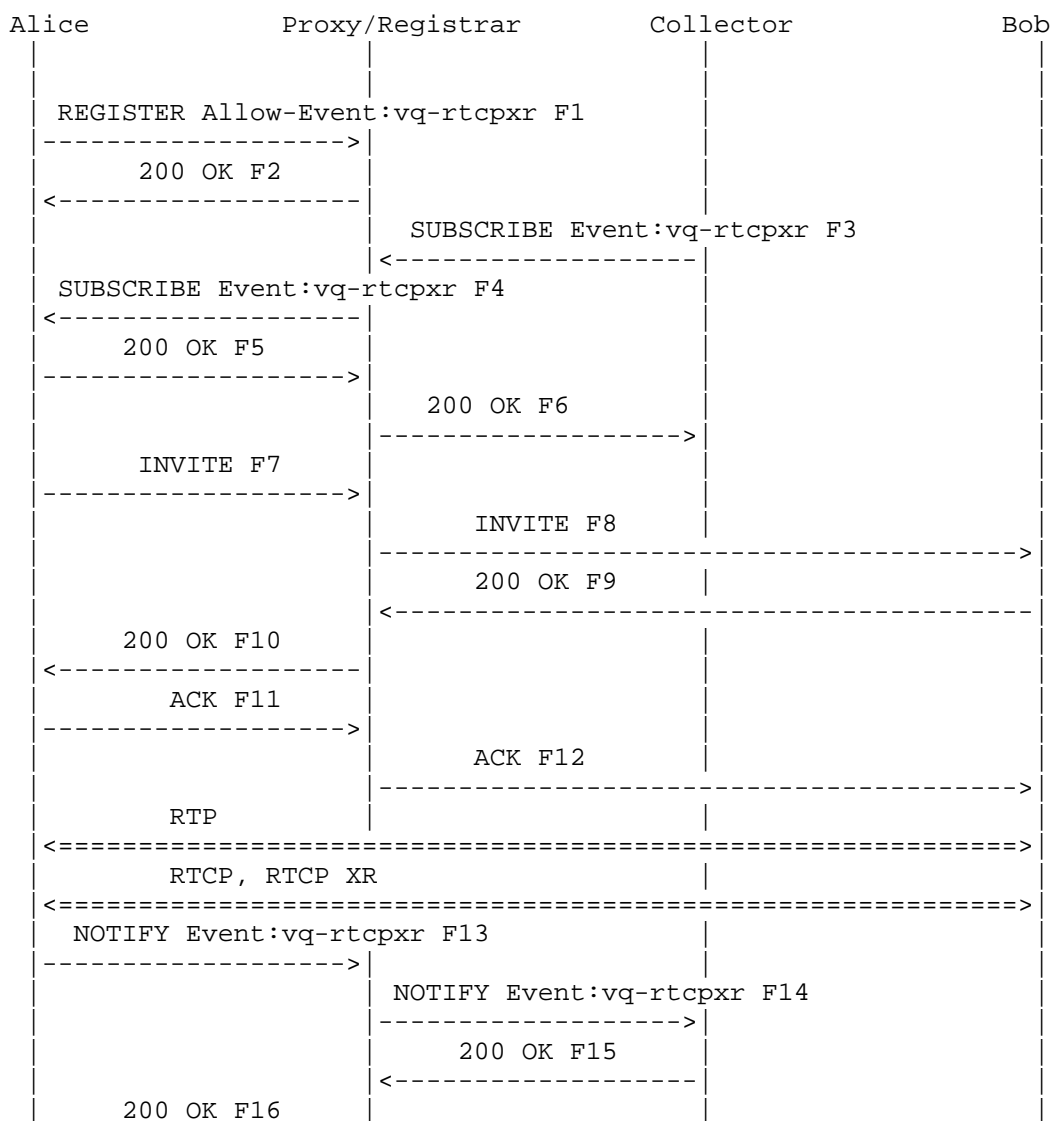
Figure 1. Summary report with NOTIFY sent after session termination. In the call flow depicted in Figure 1, the following message format is sent in F17:

```
NOTIFY sip:collector@example.org SIP/2.0
Via: SIP/2.0/UDP pc22.example.org;branch=z9hG4bK3343d7
Max-Forwards: 70
To: <sip:collector@example.org>;tag=43524545
From: Alice <sip:alice@example.org>;tag=a3343df32
Call-ID: 1890463548
CSeq: 4321 NOTIFY
Contact: <sip:alice@pc22.example.org>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
      SUBSCRIBE, NOTIFY
Event: vq-rtcpvr
Accept: application/sdp, message/sipfrag
Subscription-State: active;expires=3600
Content-Type: application/vq-rtcpvr
Content-Length: ...

VQSessionReport: CallTerm
CallID: 6dg37f1890463
LocalID: Alice <sip:alice@example.org>
RemoteID: Bill <sip:bill@example.net>
OrigID: Alice <sip:alice@example.org>
LocalGroup: example-phone-55671
RemoteGroup: example-gateway-09871
LocalAddr: IP=10.10.1.100 PORT=5000 SSRC=1a3b5c7d
LocalMAC: 00:1f:5b:cc:21:0f
RemoteAddr: IP=11.1.1.150 PORT=5002 SSRC=0x2468abcd
RemoteMAC: 00:26:08:8e:95:02
LocalMetrics:
Timestamps: START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc: PT=0 PD=PCMU SR=8000 FD=20 FO=160 FPP=1 PPS=50
             PLC=3 SSUP=on
JitterBuffer: JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss: NLR=5.0 JDR=2.0
BurstGapLoss: BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay: RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal: SL=-18 NL=-50 RERL=55
QualityEst: RLQ=88 RCQ=85 EXTRI=90 MOSLQ=4.1 MOSCQ=4.0
           QoEEstAlg=P.564
RemoteMetrics:
Timestamps: START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc: PT=0 PD=PCMU SR=8000 FD=20 FO=160 FPP=1 PPS=50
             PLC=3 SSUP=on
JitterBuffer: JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss: NLR=5.0 JDR=2.0
```

BurstGapLoss:BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
 Delay:RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
 Signal:SL=-21 NL=-45 RERL=55
 QualityEst:RLQ=90 RCQ=85 EXTRI=90 MOSLQ=4.3 MOSCQ=4.2
 QoEEstAlg=P.564
 DialogID:1890463548@alice.example.org;to-tag=8472761;
 from-tag=9123dh311

4.7.2. Midsession Threshold Violation Using NOTIFY



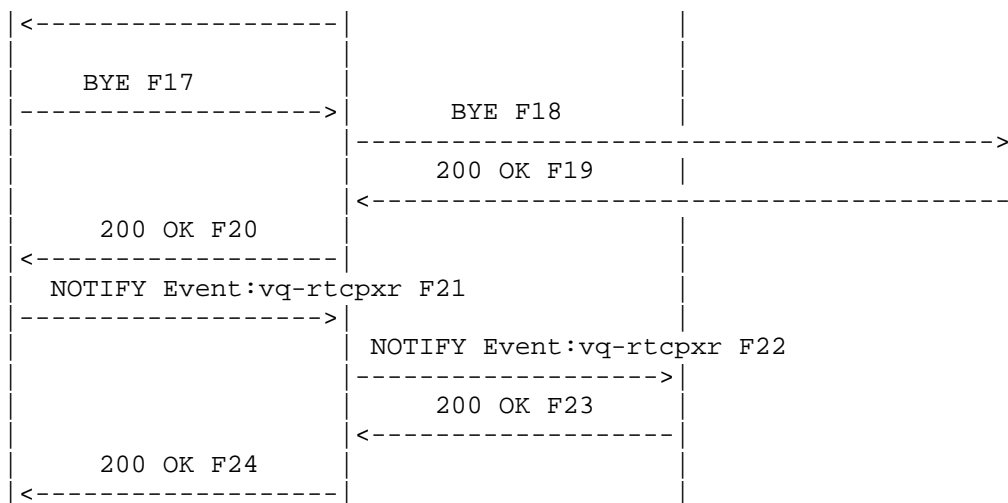


Figure 2. An alert report is sent during the session.
In the call flow depicted in Figure 2, the following message format is sent in F13:

```

NOTIFY sip:collector@example.org SIP/2.0
Via: SIP/2.0/UDP pc22.example.org;branch=z9hG4bK3343d7
Max-Forwards: 70
To: <sip:proxy@example.org>
From: Alice <sip:alice@example.org>;tag=a3343df32
Call-ID: 1890463548
CSeq: 4331 PUBLISH
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
      SUBSCRIBE, NOTIFY
Event: vq-rtcprr
Accept: application/sdp, message/sipfrag
Content-Type: application/vq-rtcprr
Content-Length: ...
  
```

```

VQAlertReport: Type=NLR Severity=Critical Dir=local
CallID: 6dg37f1890463
LocalID: Alice <sip:alice@example.org>
RemoteID: Bill <sip:bill@example.org>
OrigID: Alice <sip:alice@example.org>
LocalGroup: example-phone-55671
RemoteGroup: example-gateway-09871
LocalAddr: IP=10.10.1.100 PORT=5000 SSRC=0x2468abcd
LocalMAC: 00:1f:5b:cc:21:0f
  
```

```
RemoteAddr:IP=11.1.1.150 PORT=5002 SSRC=1357efff
RemoteMAC: 00:26:08:8e:95:02
LocalMetrics:
Timestamps:START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc:PT=18 PD=G729 SR=8000 FD=20 FO=20 FPP=2 PPS=50
             FMTP="annexb=no" PLC=3 SSUP=on
JitterBuffer:JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss:NLR=10.0 JDR=2.0
BurstGapLoss:BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay:RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal:SL=-21 NL=-50 RERL=55
QualityEst:RLQ=80 RCQ=85 EXTRI=90 MOSLQ=3.5 MOSCQ=3.7
             QoEEstAlg=P.564
RemoteMetrics:
Timestamps:START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc:PT=18 PD=G729 SR=8000 FD=20 FO=20 FPP=2 PPS=50
             FMTP="annexb=no" PLC=3 SSUP=on
JitterBuffer:JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss:NLR=5.0 JDR=2.0
BurstGapLoss:BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay:RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal:SL=-21 NL=-45 RERL=55
QualityEst:RLQ=90 RCQ=85 MOSLQ=4.3 MOSCQ=4.2 QoEEstAlg=P.564
DialogID:1890463548@alice.example.org;to-tag=8472761;
        from-tag=9123dh311
```

4.7.3. End of Session Report Using PUBLISH

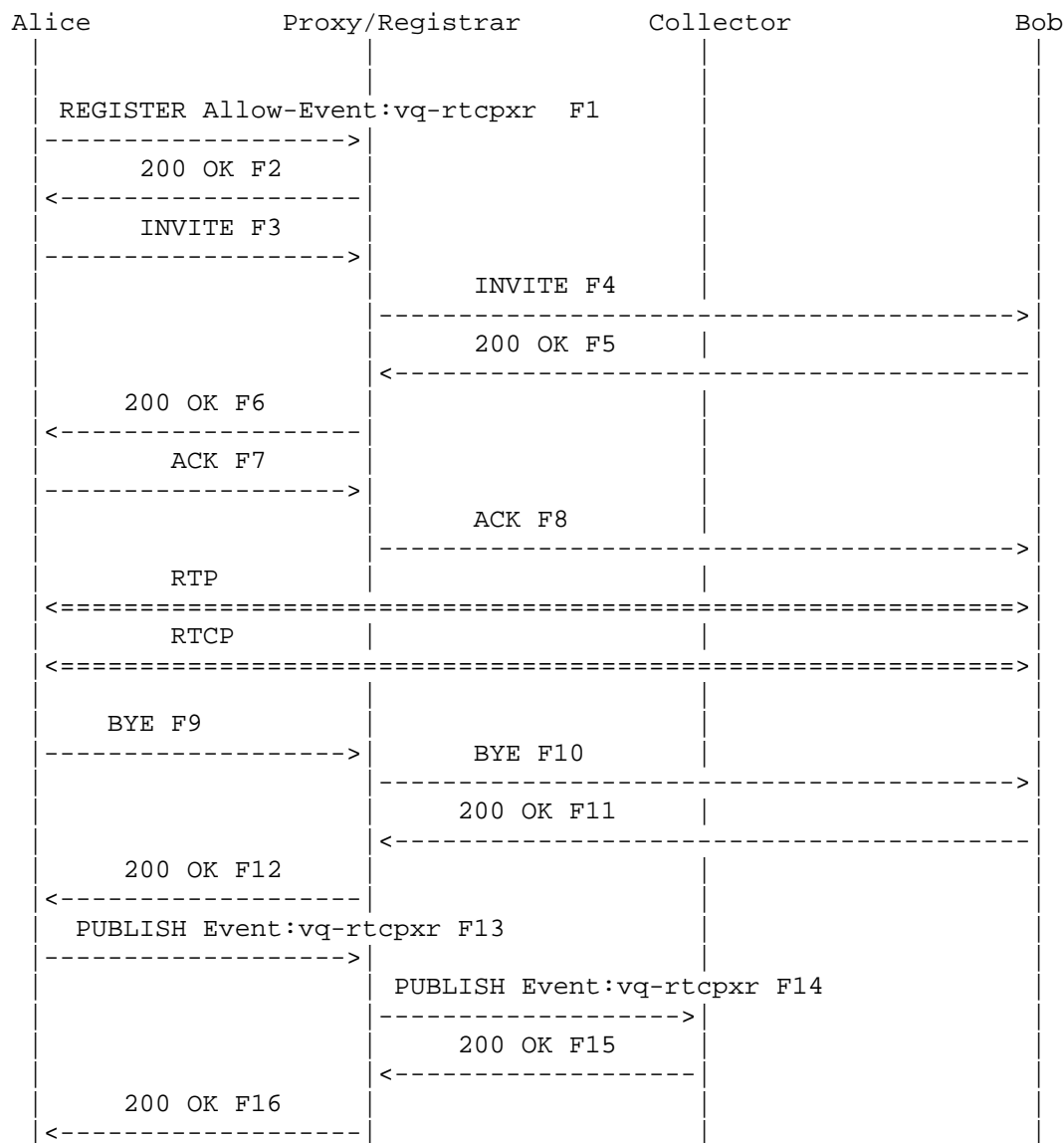


Figure 3. End of session report sent after session termination.
 In the message flow depicted in Figure 3, the following message is sent in F13.

```
PUBLISH sip:collector@example.org SIP/2.0
Via: SIP/2.0/UDP pc22.example.org;branch=z9hG4bK3343d7
Max-Forwards: 70
To: <sip:proxy@example.org>
From: Alice <sip:alice@example.org>;tag=a3343df32
Call-ID: 1890463548
CSeq: 4331 PUBLISH
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
      SUBSCRIBE, NOTIFY
Event: vq-rtcpxr
Accept: application/sdp, message/sipfrag
Content-Type: application/vq-rtcpxr
Content-Length: ...

VQSessionReport: CallTerm
CallID: 6dg37f1890463
LocalID: Alice <sip:alice@example.org>
RemoteID: Bill <sip:bill@example.net>
OrigID: Alice <sip:alice@example.org>
LocalGroup: example-phone-55671
RemoteGroup: example-gateway-09871
LocalAddr: IP=10.10.1.100 PORT=5000 SSRC=1a3b5c7d
LocalMAC: 00:1f:5b:cc:21:0f
RemoteAddr: IP=11.1.1.150 PORT=5002 SSRC=0x2468abcd
RemoteMAC: 00:26:08:8e:95:02
LocalMetrics:
Timestamps: START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc: PT=18 PD=G729 SR=8000 FD=20 FO=20 FPP=2 PPS=50
              FMTP="annexb=no" PLC=3 SSUP=on
JitterBuffer: JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss: NLR=5.0 JDR=2.0
BurstGapLoss: BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay: RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal: SL=-21 NL=-50 RERL=55
QualityEst: RLQ=90 RCQ=85 EXTRI=90 MOSLQ=4.2 MOSCQ=4.3
           QoEEstAlg=P.564
RemoteMetrics:
Timestamps: START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc: PT=18 PD=G729 SR=8000 FD=20 FO=20 FPP=2 PPS=50
              FMTP="annexb=no" PLC=3 SSUP=on
JitterBuffer: JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss: NLR=5.0 JDR=2.0
BurstGapLoss: BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay: RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal: SL=-21 NL=-45 RERL=55
QualityEst: RLQ=90 RCQ=85 MOSLQ=4.3 MOSCQ=4.2 QoEEstAlg=P.564
DialogID: 1890463548@alice.example.org;to-tag=8472761;
         from-tag=9123dh311
```

4.7.4. Alert Report Using PUBLISH

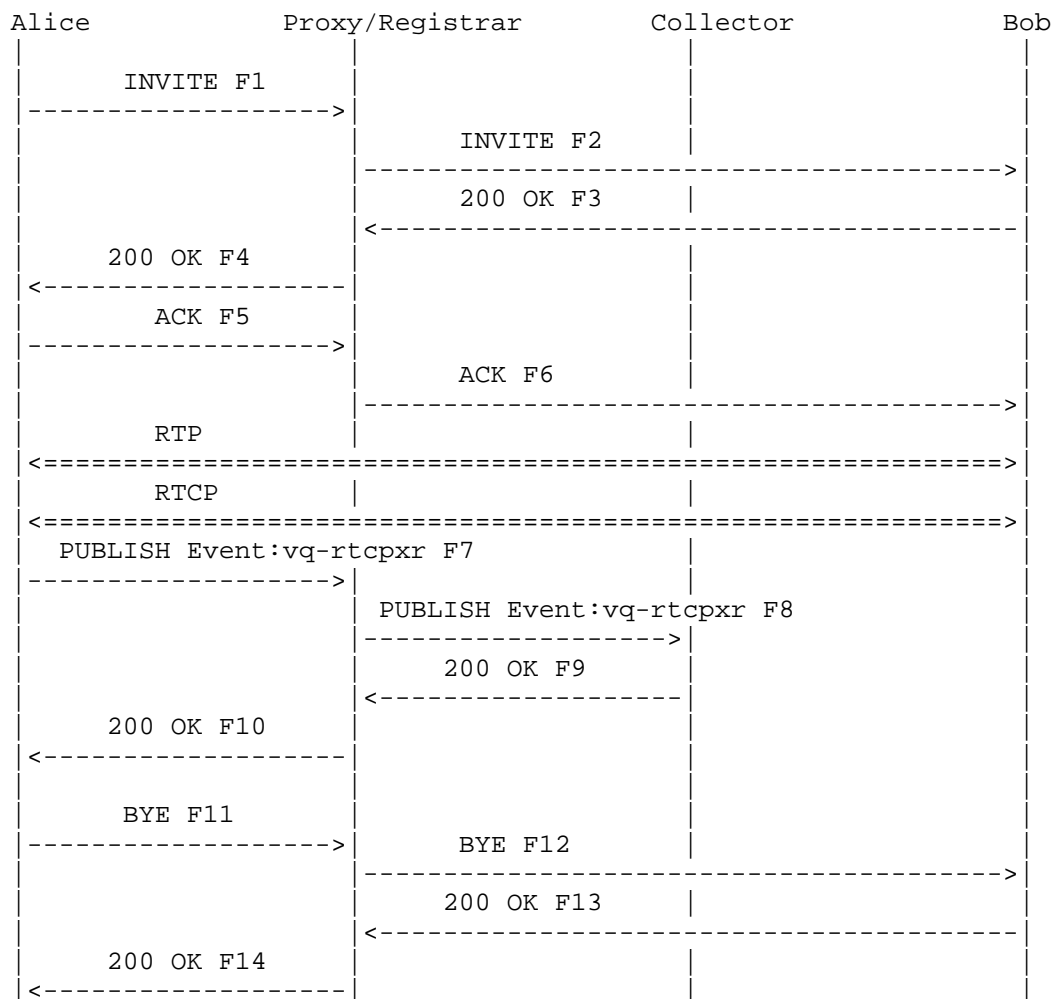


Figure 4. Alert report message flow

In the message flow depicted in Figure 4, the following message is sent in F7:

```

PUBLISH sip:collector@example.org SIP/2.0
Via: SIP/2.0/UDP pc22.example.org;branch=z9hG4bK3343d7
Max-Forwards: 70
To: <sip:collector@example.org>
From: Alice <sip:alice@example.org>;tag=a3343df32
Call-ID: 1890463548
  
```

CSeq: 4321 PUBLISH
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
SUBSCRIBE, NOTIFY
Event: vq-rtcpxr
Accept: application/sdp, message/sipfrag
Content-Type: application/vq-rtcpxr
Content-Length: ...

VQAlertReport: Type=RLQ Severity=Warning Dir=local
CallID: 6dg37f1890463
LocalID: Alice <sip:alice@example.org>
RemoteID: Bill <sip:bill@example.org>
OrigID: Alice <sip:alice@example.org>
LocalGroup: example-phone-55671
RemoteGroup: example-gateway-09871
LocalAddr: IP=10.10.1.100 PORT=5000 SSRC=1a3b5c7d
LocalMAC: 00:1f:5b:cc:21:0f
RemoteAddr: IP=11.1.1.150 PORT=5002 SSRC=0x2468abcd
RemoteMAC: 00:26:08:8e:95:02
Metrics:
Timestamps: START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc: PT=0 PD=PCMU SR=8000 FD=20 FO=160 FPP=1 PPS=50
PLC=3 SSUP=on
JitterBuffer: JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss: NLR=5.0 JDR=2.0
BurstGapLoss: BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay: RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal: SL=-12 NL=-30 RERL=55
QualityEst: RLQ=60 RCQ=55 EXTR=90 MOSLQ=2.4 MOSCQ=2.3
QoEEstAlg=P.564
RemoteMetrics:
Timestamps: START=2004-10-10T18:23:43Z STOP=2004-10-01T18:26:02Z
SessionDesc: PT=0 PD=PCMU SR=8000 FD=20 FO=160 FPP=1 PPS=50
PLC=3 SSUP=on
JitterBuffer: JBA=3 JBR=2 JBN=40 JBM=80 JBX=120
PacketLoss: NLR=5.0 JDR=2.0
BurstGapLoss: BLD=0 BD=0 GLD=2.0 GD=500 GMIN=16
Delay: RTD=200 ESD=140 SOWD=200 IAJ=2 MAJ=10
Signal: SL=-23 NL=-60 RERL=55
QualityEst: RLQ=90 RCQ=85 EXTRI=90 MOSLQ=4.2 MOSCQ=4.3
QoEEstAlg=P.564
DialogID: 1890463548@alice.example.org;to-tag=8472761;
from-tag=9123dh3111

4.8. Configuration Dataset for vq-rtcprr Events

It is the suggestion of the authors that the SIP configuration framework [15] be used to establish the necessary parameters for usage of vq-rtcprr events. A dataset for this purpose should be designed and documented in a separate document upon completion of the framework.

5. IANA Considerations

This document registers a new SIP Event Package and a new media type.

5.1. SIP Event Package Registration

Package name: vq-rtcprr
Type: package
Contact: Amy Pendleton <aspen@telchemy.com>
Published Specification: This document

5.2. application/vq-rtcprr Media Type Registration

Type name: application
Subtype name: vq-rtcprr
Required parameters: none
Optional parameters: none
Encoding considerations: 7 bit
Security considerations: See next section.
Interoperability considerations: none.
Published specification: This document.

Applications that use this media type: This document type is being used in notifications of VoIP quality reports.

Additional Information:

Magic Number: None
File Extension: None
Macintosh file type code: "TEXT"

Person and email address for further information: Amy Pendleton
<aspen@telchemy.com>

Intended usage: COMMON

Author / Change controller: The IETF.

6. Security Considerations

RTCP reports can contain sensitive information since they can provide information about the nature and duration of a session established between two or more endpoints. As a result, any third party wishing to obtain this information SHOULD be properly authenticated by the SIP UA using standard SIP mechanisms and according to the recommendations in [5]. Additionally, the event content MAY be encrypted to ensure confidentiality; the mechanisms for providing confidentiality are detailed in [2].

7. Contributors

The authors would like to thank Rajesh Kumar, Dave Oran, Tom Redman, Shane Holthaus, and Jack Ford for their comments and input.

8. References

8.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [3] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [4] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [5] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.
- [6] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [7] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [8] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.
- [9] ITU-T G.1020, "Performance parameter definitions for quality of speech and other voiceband applications utilizing IP networks".

- [10] ITU-T P.564, "Conformance testing for voice over IP transmission quality assessment models".
- [11] ITU-T G.107, "The E-model, a computational model for use in transmission planning".
- [12] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

8.2. Informative References

- [13] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.
- [14] Hilt, V., Noel, E., Shen, C., and A. Abdelal, "Design Considerations for Session Initiation Protocol (SIP) Overload Control", Work in Progress, July 2009.
- [15] Petrie, D. and S. Channabasappa, "A Framework for Session Initiation Protocol User Agent Profile Delivery", Work in Progress, October 2010.
- [16] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

Authors' Addresses

Amy Pendleton
Telchemy Incorporated
EMail: aspen@telchemy.com

Alan Clark
Telchemy Incorporated
EMail: alan.d.clark@telchemy.com

Alan Johnston
Avaya
St. Louis, MO 63124
EMail: alan.b.johnston@gmail.com

Henry Sinnreich
Unaffiliated
EMail: henry.sinnreich@gmail.com

