Network Working Group A. Akhter
Internet-Draft Cisco Systems
Intended status: Standards Track March 27, 2012
Expires: September 28, 2012

# Methodology for Network Flow Performance Measurement
## draft-akhter-opsawg-perfmon-method-02.txt

Abstract

There is a need to be able to quantify and report the performance of network applications and the network service in handling user data. This performance data provides information essential in validating service level agreements, fault isolation as well as early warnings of network greater problems. This document describes a generic methodology for calculating metrics related to network based applications. In addition, to the performance metrics, several additional information elements are included to help provide greater context to the reports. The measurements use audio/video applications as base examples but are not restricted to these class of applications.

Status of this Memo



Copyright Notice

Table of Contents

## 1. Introduction

Today's networks support a multitude of highly demanding and sensitive network applications. Network issues are readily apparent by the users of these applications due to the sensitivity of these applications to impaired network conditions. Examples of these network applications include applications making use of IP based audio, video, database transactions, virtual desktop interface (VDI), online gaming, cloud services and many more. In some cases, the impaired application translates directly to loss of revenue. In other cases, there may be regulatory or contractual service level agreements that motivate the network operator. Due to the sensitivity of these types of applications to impaired service, it leaves a poor impression of the network service on the user-- regardless of the actual performance of the network itself. In the case of an actual problem within the network service, monitoring the performance may yield an early indicator of a much more serious problem.

Due to the demanding and sensitive nature of these applications, network operators have tried to engineer their networks towards wringing better and differentiated performance. However, that same differentiated design prevents network operators from extrapolating observational data from one application to another, or from one set of synthetic (active test) test traffic to actual application performance. This gap highlights the importance of generic measurements as well as the reliance on user traffic measurents-- rather than synthetic tests.

Performance measurements on user data provide greater visibility not only into the quality of experience of the end users but also visibility into network health. With regards to network health, as flow performance is being measured, there will be visibility into the end to end performance which means that not only visibility into local network health, but also viability into remote network health. If these measurements are made at multiple points within the network (or between the network and end device) then there is not only identification that there might be an issue, but a span of area can be established where the issue might be. The resolution of the fault increases with the number of measurement points along the flow path.

The IP Flow Information Export Protocol (IPFIX) [RFC5101] provides new levels of flexibility in reporting from measurement points across the life cycle of a network based application. IPFIX can provide granular results in terms of flow specificity as well as time granularity. At the same time, IPFIX allows for summarization of data along different types of boundaries for operators that are unconcerned about specific sessions but about health of a service or

a portion of the network. This document details the methodlogy of measurement, while an accompanying document describes the expression of the measurements in IPFIX format.

Where possible, an attempt has been made to make use of existing definitions of metrics ([RFC4710]) and if needed, clarify and expand on them to widen their usage with additional applications, and network devices. For example, the RTP measurments have generally defined from the prespective of end systems rather than intermdiate nodes which are not alwyas privy to the application context and may have limited scaling properties. The methodology described in [I-D.ietf-pmol-sip-perf-metrics] is used to describe the methodology of measurement.

There has been related work in this area such as [RFC2321]. [I-D.huici-ipfix-sipfix], and [VoIP-monitor]. This document is also an attempt to generalize as well as standardize the reporting formats and measurement methodology.

## 2. Terminology

Terms used in this document that are defined in the Terminology section of the IPFIX Protocol [RFC5101] document are to be interpreted as defined there.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In addition, the information element definitions use the following terms:

Name: Name of the information element

Description: Short description of what the information element is trying to convey.

Observation Point: Where the measurement is meant to be performed. Either at an intermediate system (for example, a router) or end system.

Use and Applications An explanation of how this particular information element would be used.

Calculation Method: In the case of metrics, this section describes how the metric is calculated, as well as any special conditions.

Units of Measurement: In the case of metrics, what are the units of measurement. The text here is expected to be wider and more descriptive than in the IPFIX Element Units section.

Measurement Timing: Discussion on the acceptable range of timing and sampling intervals.

## 3. General Usage

### 3.1. Quality of Service (QoS) Monitoring

The network operator needs to be able to gauge the end user's satisfaction with the network service. While there are many components of the satisfaction such as pricing, packaging, offering, etc., a major component of satisfaction is delivering a consistent service. The user builds trust on this consistency of the network service and runs network applications with confidence-- which is of course the end goal. Without the ability to deliver a consistent service for end user network applications network operator will be left dealing with price sensitive disgruntled users with very low expectations and utilization (if they don't have choice of operator) or abandonment (if they have choice).

### 3.2. Service Level Agreemnt (SLA) Validation

Similar to QoS and QoE validation, there might be contractual or regulatory requirements that need to be met by the network operator. Monitoring the performance of the flows allows the application operator, network operator as well as the end user to validate if the target service is being delivered. While there is quite a diversity in the codification of network SLAs, the eventually involve some measurement of network uptime, end to end latency, end to end jitter and perhaps service response time. In the case of SLA violation, the start and end times, nature and network scope of the violation needs to be captured to allow for the most accurate settling of the SLA violation.

### 3.3. Fault Isolation and Troubleshooting

It has been generally easier to troubleshoot and fix problems that are binary in nature: it either works or does not work. The host is pingable or not pingable. However, it is the much more difficult to resolve transitory issues that move from location to location, are not complete faiures and sometimes with unverfifiable end user

reports as the only indication of a problem. It is these intermittent and seemingly inconsistent network impairments that performance metrics can be extremely helpful with. Just the basic timely detection that there is a problem (or an impending problem) can give the operator provider the confidence that there is a real problem that needs to be resolved. The next step would be to assist the operator in a speedy resolution by providing information regarding the network location and nature of the problem.

## 4. New Information Elements

The information elements are organized into two main groups:

Transport Layer: Metrics that might be calculated from observations at higher layers but essentially provide information about the network transport of user date. For example, the metrics related to packet loss, latency and jitter would be defined here.

User and Application Layer: Metrics that are might be affected by the network indirectly, but are ultimately related to user, end-system and session states. For example, session setup time, transaction rate and session duration would be defined here.

Contextual Elements: Information elements that provide further context to the metrics. For example, media type, codec type, and type of application would be defined here.

### 4.1. Transport Layer

#### 4.1.1. perfPacketLoss

Name: perfPacketLoss

Description: The packet loss metric reports the number of individual packets that were lost in the reporting interval.

Observation Point: The observation can be made anywhere along the media path or on the endpoints them selves. The observation is only relevant in a unidirectional sense.

Use and Applications The packet loss metric can be used to determine if there is a network impairment that is causing packet loss upstream of the measurement point. When there are observation points on either side of the impairment location it is possible to locate the impairment. With the location information the operator can is able to perform quicker fault-isolation as well as shorten time to resolution. Depending on implementation and operator

configuration, the granularity of contextual information can be very specific. For example, these traffic loss statistics when sent with IP subnet or DSCP information can provide visibilty into QoS specific or network topology issues.

Calculation Method: This metric requires that each IP packet be individually marked with a monotonically incrementing sequence number. A number of encapsulations support this type of sequencing: IPSec ESP [RFC4303], GRE [RFC2890] and RTP [RFC3550]. An analysis of the sequence number field can yield the lost number of packets. In certain cases, there might be an element of discovery and synchronization of the flow itself before the measurement can be made. An example of this can be found for RTP flows running on ephemeral UDP port numbers. In these cases, reporting 0 as packet loss would be misleading and the value 0xFFFFFFFF MUST be used in cases where the packet loss value cannot be determined. In the case of a monitor interval where synchronization was achieved mid-interval, the loss packet counter MAY be used to represent the remainder of the interval. As this metric is a deltaCounter, the number of loss packets only represent the observation within the reporting interval. Due to the dependency on the arrival of a packet with a sequence number to calculate loss, the loss calculation may be indefinitely delayed if no more packets arrive at all. For the case of RTP, in addition to the 16 bit sequence number field in RFC3550, there is also the additional 16-bit high-order sequence number field (for a total of 32-bit seq number space) that is used in RFC3497 [RFC3497]. RFC3497 traffic runs at a very high rate and the 32-bit field allow for additional time for wrapping (21 seconds). So, a loss span of greater than 21 seconds measured only by the 16-bit field will lead to inaccurate reporting. In the case of secure RTP [RFC3711], the relevant portion of the RTP header is in the clear and lost packet counting can still be performed. It is important to note that the sequence number space is unique per RTP SSRC. Therefore it is important to track the high sequence number seen on a per SSRC-5-tuple basis. There may be multiple SSRCs in a single 5-tuple. Certain applications inject non-RTP traffic into the same 5-tuple as the media stream. RTCP packets may be seen in the same 5-tuple as the RTP stream [RFC5761], and STUN [RFC5389] packets may also be seen. The loss detection should ignore these packets. There may be spans within the network where header compression schemes such as [RFC2508] are used. In cases where the measurement device is terminating the compression, and the measurement implementation does not support calculation of the metric the value 0xFFFFFFFF MUST be reported. In other cases the measurement point may be at a midpoint of the header compression network span. Depending on the mechanics of header compression, sequencing information may be present and it is possible to

calculate the metric. In such cases the implementation SHOULD perform the calculation and report the metric.

Units of Measurement: packets

Measurment Timing To be able to calculate this metric a continuous set of the flow's packets (as each would have an incrementing sequence number) needs to be monitored. Therefore, per-packet sampling would prevent this metric from being calculated. However, there are other sampling methodologies that might be usable. It is possible to generate sampled metrics by sampling spans of continuous packets, however a portion of the span may have to be utilized for resynchronization of the sequence number. Another form of acceptable sampling would be at the flow level.

### 4.1.2. perfPacketExpected

Name: perfPacketExpected

Description: The number of packets there were expected within a monitoring interval.

Observation Point: The observation can be made anywhere along the media path or on the endpoints them selves. The observation is only relevant in a unidirectional sense.

Use and Applications The perfPacketExpected is a mid-calculation metric used in the generation of perfPacketLossRate. It is equivilent to the highest received packet sequence number at the time of measurement. As the value only increments when packets are received, packet loss may be occouring at the time of measurement but perfPacketExpected remains constant.

Calculation Method: The subtraction of the last sequence number from the first sequence number in monitoring interval yields the expected count. As discussed with perfPacketLost, there might be a delay due to synchronization with the flow's sequence numbers and in such times the value of the metric should be set to 0xFFFFFFFE. Care has to be taken to account for cases where the packet's sequence number field wraps. For RTP, the expected count calculation formula can be found in Appendix A.3 of [RFC3550]. Refer to the perfPacketLoss metric regarding considerations for header compression. The value 0xFFFF is used to represent cases where the metric could not be calculated.

Units of Measurement: packets

Measurment Timing Same considerations as perfPacketLoss

### 4.1.3. perfPacketLossRate

Name: perfPacketLossPercentage

Description: Percentage of number of packets lost out of the total set of packets sent.

Observation Point: The observation can be made anywhere along the media path or on the endpoints them selves. The observation is only relevant in a unidirectional sense.

Use and Applications The perfPacketLossRate metric can be used to normalize the perfPacketLoss metric to handle cases where different flows are running at different packet per second (PPS) rates. Due to the normalization, comparisons can now be made against thresholds (for creating alerts, etc.). In addition, the percentage form of the metric allows for comparisons against other flows at the same observation point to determine if there is an equal bias for drops between the flows. Otherwise, the perfoPacketLossRate is used in same way as perfPacketLoss. This value can be derived from perfPacketExpected and perfPacketLoss and is offered as a convenience to ease functions such as thresholding, and pre-computed reporting. It shoudl be noted that for large values of perfPacketExpected and perfPacketLoss it might be preferable and more accurate for the conversion to percentage to occour at a later stage where the accuracy can be controlled.

Calculation Method: The number of lost packets divided by the number of expected packets in an interval period multiplied by 100. In cases where perfPacketLoss is unknown (for example due to synchronization issues), the perfPacketLossRate would also be unknown. If there are multiple flows whose loss rate is being aggregated, then the average of the individual flows is used. Refer to the perfPacketLoss metric regarding considerations for header compression.

Units of Measurement: percentage

Measurment Timing Same notes as perfPacketLossPercentage

#### 4.1.4. perfPacketLossEvent

Name: perfPacketLossEvent

Description: The packet loss event metric reports the number of continuous sets of packets that were lost in the reporting interval.

Observation Point: The observation can be made anywhere along the media path or on the endpoints them selves. The observation is only relevant in a unidirectional sense.

Use and Applications The perfPacketLossEvent metric can provide loss information for protocols that do not implement per packet sequencing. Similarly to the perfPacketLoss metric, the packet loss event metric can be used to determine if there is a network impairment that is causing packet loss upstream of the measurement point. In cases where both the perfPacketLoss and perfPacketLossEvent metric are available, the ratio between the packet loss and packet event count can provide the average loss length. The average loss length provides additional information regarding the cause of the loss. For example, a dirty fiber connection might have a low average loss length, while a routing protocol convergence will have a high loss length.

Calculation Method: This data value is a simplified version of the Lost Packets metric. Whereas Lost Packets counts individual packet loss, the 'loss event count' metric counts sets of packets that are lost. For example, in the case of a sequence of packets: 1,3,6,7,10 the packets marked 2,4,5,8 and 9 are lost. So, a total of 5 packets are lost. This same sequence translates to 3 loss events: (2), (4,5) and (8,9). In the case of RTP, the sequence number in the RTP header can be used to identify loss events. Certain protocols such as TCP and UDP+MPEG2-TS encapsulation in IP have sequencing information, but the sequence field is incremented by individual IP packets. As a side note, in the case of UDP+ MPEG2-TS encapsulation the simple use of RTP+MPEG2-TS via [RFC2250] results in the avaliability of the more granular perfPacketLoss metrics. In these cases, the perfPacketLoss metric cannot be calculated but the perfPacketLossEvent can be calculated and can provide detection of loss. The value 0xFFFFFFFF is used to represent non-applicable cases such as lack of sequence number synchronization. Many of the same considerations as for perfPacketLoss apply to perfPacketLoss event. Please refer to the Calculation Method section of the perfPacketLoss.

Units of Measurement: event counts

Measurment Timing Please refer to the measurement timing section of perfPacketLoss.

### 4.1.5. perfPacketInterArrivalJitterAvg

Name: perfPacketInterArrivalJitterAvg

Description: This metric measures the absolute deviation of the difference in packet spacing at the measurement point compared to the packet spacing at the sender.

Observation Point: The observation can be made anywhere along the media path or on the receiver. The observation is only relevant in a unidirectional sense.

Use and Applications The inter arrival jitter data value can be used be network operator to determine the network's impact to the spacing in between a media stream's packets as they traverse the network. For example, in the case of media applications, the receiving end system is expecting these packets to come in at a particular periodicity and large deviations may result in de-jitter buffers adding excessive delay, or the media packets being discarded. When the data is reported from multiple intermediate nodes, the area of the network that is having a detrimental contribution can be identified. On a non-media application level, the inter arrival jitter metrics can be used for early indication queuing contention within the network (which could lead to packet loss).

Calculation Method: The inter arrival jitter value makes use of the association of sending time with an IP packets and comparison of the arrival time on the monitoring point. In certain protocols, a representation of sending time is encoded into the header itself. For example, in the case of RTP packets, the RTP header's timestamps field represents encoder clock ticks-- which are representations of time. Similarly, in the case of TCP options encode absolute timestamps values. For RTP the calculation method can be found in Appendix A of [RFC3550]. It should be noted that the RFC3550 calculation is on the last 16 packets measured. The most recent value calculated SHOULD be reported at the end of the monitoring interval. The range of the jitter values during the monitoring interval can be reported using perfPacketInterArrivalJitterMin and perfPacketInterArrivalJitterMax. Similarly to the perfPacketLoss case there may be periods of time where the jitter value cannot be calculated. In these cases, the 0xFFFFFFFF value should be used

to convey the lack of availability of the metric. As mentioned earlier, the RTP header timestamps is actually a 'sample-stamp' (ie clicks) from the encoder's clock. The frequency of the clock is dependent on the codec. Some codecs (eg AAC-LD) support multiple possible frequencies one of which is then selected for the media-stream. The mapping to clock rate can be performed via mapping from the static RTP payload type (RTP-PT), but newer codecs are make use of the dynamic payload type range and the RTP-PT (in the dynamic case) cannot be used to determine the clock frequency. There are various methods by which the clock frequency (deep packet inspection of the signalling, manual configuration, etc.) can be associated to the calculation method. The frequency should be locked in the metering layer to a unique combination of the IP source, IP destination, IP protocol layer-4 ports, RTP-PT and SSRC. By strict RFC3550 definition, the SSRC is set to a specific encoder clock and it is the SSRC that should be tracked rather than payload type. However, in recent discussions it has been noted that there are RTP implementations that might change the encoder clock frequency while maintaining the SSRC value. An encoder frequency change will be accompanied by a different RTP-PT.

Units of Measurement: microseconds

Measurment Timing Please refer to the measurement timing section of perfPacketLoss.

#### 4.1.6. perfPacketInterArrivalJitterMin

Name: perfPacketInterArrivalJitterMin

Description: This metric measures the minimum value the calculation used for perfPacketInterArrivalJitterAvg within the monitoring interval.

Observation Point: The observation can be made anywhere along the media path or on the receiver. The observation is only relevant in a unidirectional sense.

Use and Applications Please refer to the 'Use and Applications' section of perfPacketInterArrivalJitterAvg. This specific metric, along with perfPacketInterArrivalJitterMax, is to capture the range of measurements observed within a monitoring interval as the average function may hide extremes.

Calculation Method: Please see the perfPacketInterArrivalJitterAvg section for general calculation section. The average calculation is evaluated on a running basis over the last 16 packets and the entire monitoring interval is not covered. In this metric, the minimum value is taken over the entire monitoring interval.

Units of Measurement: microseconds

Measurment Timing Please refer to the measurement timing section of perfPacketLoss.

### 4.1.7. perfPacketInterArrivalJitterMax

Name: perfPacketInterArrivalJitterMax

Description: This metric measures the maximum value the calculation used for perfPacketInterArrivalJitterAvg within the monitoring interval.

Observation Point: The observation can be made anywhere along the media path or on the receiver. The observation is only relevant in a unidirectional sense.

Use and Applications Please refer to the 'Use and Applications' section of perfPacketInterArrivalJitterAvg. This specific metric, along with perfPacketInterArrivalJitterMin, is to capture the range of measurements observed within a monitoring interval as the average function may hide extremes.

Calculation Method: Please see the perfPacketInterArrivalJitterAvg section for general calculation section. The average calculation is evaluated on a running basis over the last 16 packets and the entire monitoring interval is not covered. In this metric, the maximum value is taken over the entire monitoring interval.

Units of Measurement: microseconds

Measurment Timing Please refer to the measurement timing section of perfPacketLoss.

### 4.1.8. perfRoundTripNetworkDelay

Name: perfRoundTripNetworkDelay

Description: This metric measures the network round trip time between end stations for a flow.

Observation Point: The observation can be made anywhere along the flow path as long as the bidirectional network delay is accounted for.

Use and Applications The perfRoundTripNetworkDelay metric can be used in multiple ways. If the applicaiton being monitored provides interactive feedback to the user the perfRoundTripNetworkDelay can be used to judge the 'liveliness' of the application experience. Other use cases may involve troubleshooting throughput issues where the transport protocol's throughtput is affected by network delay.

Calculation Method: perfRoundTripNetworkDelay can estimated by accounting for the network flight time between a transport protocol request and response. In the case of TCP, this would the time difference between the TCP SYN and ACK packets in the TCP handshake. It should be noted that at times other than the TCP handshake the time difference between TCP end station packet. For RTP (RFC3550) based applications, the network round trip can be calculated by analysis of hte RTCP sending and receive times.

Units of Measurement: microseconds

Measurment Timing Depending on the method used to calculate the round trip time, the measurment may only be possible at specific times during the session lifecycle. In time periods where the metric is not current 'not calculated' SHOULD be reported.

## 4.2. User and Application Layer

### 4.2.1. perfSessionSetupDelay

Name: perfSessionSetupDelay

Description: The Session Setup Delay metric reports the time taken from a request being initiated by a host/endpoint to the response (or request indicator) to the request being observed. This metric is defined in [RFC4710], however the units have been updated to microseconds.

Observation Point: This metric needs to be calculated where both request and response can be observed. This could be at network choke points, application proxies, or within the end systems themselves.

Use and Applications The session setup delay metric can measure the end user initial wait experience as seen from the network transaction level. The value will not only include the network flight time, but also includes the server response time and may be used to alert the operator in cases where the overall service is overloaded and thus sluggish, or within normal operating values.

Calculation Method: Measure distance in time between the first bit of request and the first bit of the response. For the case of SIP, please see Section 4.3.1 of [I-D.ietf-pmol-sip-perf-metrics]

Units of Measurement: microseconds

Measurment Timing This measurement can be sampled on a session by session basis. It may be advisable to set sample targets on a per source range - to destination basis. Due to the nature of measurement intervals, there may be a period of time (and thus measurement reports) in which the perfSessionSetupDelay value has not been calculated. In these cases the value 0xFFFFFFFE MUST be used and can be interpreted to mean not applicable. For measurement intervals after perfSessionSetupDelay has been calculated and the existing calculated perfSessionSetupDelay value SHOULD be sent if reporting only on that single session. However, if multiple sessions are summarized in the report then the average for perfSessionSetupDelay values calculated in the most recent interval SHOULD be used. The intention with this behavior is to acknowledge that the value has not bee calculated, and when it has provide the freshest values available.

## 4.3. Contextual Elements

### 4.3.1. mediaRTPSSRC

Name: mediaRTPSSRC

Description: Value of the synchronization source (SSRC) field in the RTP header of the flow. This field is defined in [RFC3550]

Observation Point: This metric can be gleaned from the RTP packets directly, so the observation point needs to on the flow path or within the endpoints.

Use and Applications The RTP SSRC value denotes a specific media stream. As such when trying to differentiate media stream problems between session participants the SSRC field is needed.

Calculation Method: Copy from RTP header's SSRC field as defined in [RFC3550]. In the case of a non-RTP flow, or the time period in which the flow has not been verified to be a RTP flow the value 0xFFFFFFFE MUST be reported.

Units of Measurement: identifier

Measurment Timing It is possible that the SSRC may have be renegotiated mid-session due to collisions with other RTP senders.

### 4.3.2. mediaRTPPayloadType

Name: mediaRTPPayloadType

Description: The value of the RTP Payload Type Field as seen in the RTP header of the flow. This field is defined in [RFC3550]

Observation Point: This metric can be gleaned from the RTP packets directly, so the observation point needs to on the flow path or within the endpoints.

Use and Applications The RTP PT conveys the payload format and media encoding used in the RTP payload. For simple cases, where the RTP PT is from the statically defined range this can lead to an understanding of type of media codec used. With the knowledge of the codec being used the degree of media impairment (given loss values and jitter) can be estimated better. However, for more recent codecs, the RTP dynamic range is used. In these cases the RTP payload values are dynamically negotiated. In the case of a non-RTP flow, or the time period in which the flow has not been verified to be a RTP flow, the value 0xFFFF MUST be reported.

Calculation Method: Copy from RTP header's RTP-PT field as defined in [RFC3550]

Units of Measurement: identifier

Measurment Timing

### 4.3.3. mimeType

Name: mimeType

Description: The mime type describes the content of the flow.

Observation Point: The ideal location of this metric is on the application generators and consumers. However, given application signalling inspection or static configuration it is possible that intermediate nodes are able to generate mime type (eg. codec name) information.

Use and Applications The mime type value conveys information regarding the content of a flow. For example, in the case of Audio/Video applications the name of the codec used to encode the media in the flow. Simply reporting loss and jitter measurements are useful for detection of network problems. However, judging the degree of the impact on the audio/video experience needs additional information. The most basic information is the codec being used which when coupled with per-codec knowledge of sensitivity to the transport metrics a better idea of the experience can be gained.

Calculation Method: The valid values for the mime type are listed on the IANA mime type registry. For Audio/Video codecs, there is a specific media-types registry. Analysis of the RTP payload type may lead to the determination of the media codec. However, with the use of the RTP dynamic payload type range the media information is not encoded into the data packet. For these cases, intermediate nodes may need to perform inspection of the signalling (SIP, H.323, RTSP, etc.). In cases where the mediaCodec cannot be determined, the value 'unknown' MUST be used.

Units of Measurement: identifier

Measurment Timing

## 5. Security Considerations

The recommendations in this document do not introduce any additional security issues to those already mentioned in [RFC5101] and [RFC5477]

## 6. Acknowledgements

The authors would like to thank Rahul Patel, Jan Novak, Al Morton, Brad Fawcett, Doug Manley and Shingo Kashima for their invaluable review and comments. Thank-you.

## 7. References

### 7.1. Normative References


[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5610] Boschi, E., Trammell, B., Mark, L., and T. Zseby, "Exporting Type Information for IP Flow Information Export (IPFIX) Information Elements", RFC 5610, July 2009.

[RFC4710] Siddiqui, A., Romascanu, D., and E. Golovinsky, "Real-time Application Quality-of-Service Monitoring (RAQMON) Framework", RFC 4710, October 2006.

[RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.

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

[RFC3497] Gharai, L., Perkins, C., Goncher, G., and A. Mankin, "RTP Payload Format for Society of Motion Picture and Television Engineers (SMPTE) 292M Video", RFC 3497, March 2003.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[I-D.ietf-pmol-sip-perf-metrics]
Malas, D. and A. Morton, "Basic Telephony SIP End-to-End Performance Metrics", draft-ietf-pmol-sip-perf-metrics-07 (work in progress), September 2010.

[iana-ipfix-assignments]
Internet Assigned Numbers Authority, "IP Flow Information Export Information Elements (http://www.iana.org/assignments/ipfix/ipfix.xml)".


### 7.2. Informative References


[I-D.ietf-pmol-metrics-framework]
Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", draft-ietf-pmol-metrics-framework-12 (work in progress), July 2011.

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC2250] Hoffman, D., Fernando, G., Goyal, V., and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250, January 1998.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.

[I-D.huici-ipfix-sipfix] Huici, F., Niccolini, S., and S. Anderson, "SIPFIX: Use Cases and Problem Statement for VoIP Monitoring and Exporting", draft-huici-ipfix-sipfix-00 (work in progress), June 2009.

[nProbe] "nProbe - NetFlow/IPFIX Network Probe (http://www.ntop.org/nProbe.html)".

[RFC2321] Bressen, A., "RITA -- The Reliable Internetwork Troubleshooting Agent", RFC 2321, April 1998.

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

[RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.

[VoIP-monitor] L. Chang-Yong, H. Kim, K. Ko, J. Jim, and H. Jeong, "A VoIP Traffic Monitoring System based on NetFlow v9, International Journal of Advanced Science and Technology, vol. 4, Mar. 2009".

## Author's Address

Aamer Akhter
Cisco Systems, Inc.
7025 Kit Creek Road
RTP, NC 27709
USA

Email: aakhter@cisco.com