

Audio/Video Transport Working Group	G. Hunt
Internet-Draft	P.J. Arden
Intended status: Informational	BT
Expires: September 29, 2011	Q. Wu, Ed.
	Huawei
	March 28, 2011

Monitoring Architectures for RTP
draft-hunt-avtcore-monarch-02.txt

[Abstract](#)

This memo proposes an architecture for extending RTCP with a new RTCP XR (RFC3611) block type to report new metrics regarding media transmission or reception quality, as proposed in RFC5968. This memo suggests that a new block should contain a single metric or a small number of metrics relevant to a single parameter of interest or concern, rather than containing a number of metrics which attempt to provide full coverage of all those parameters of concern to a specific application. Applications may then "mix and match" to create a set of blocks which covers their set of concerns. Where possible, a specific block should be designed to be re-usable across more than one application, for example, for all of voice, streaming audio and video.

[Status of this Memo](#)

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

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

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

This Internet-Draft will expire on September 29, 2011.

[Copyright Notice](#)

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as

described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

[Table of Contents](#)

- *1. [Introduction](#)
- *2. [Requirements notation](#)
- *3. [RTP monitoring architecture](#)
- *4. [RTCP Metric Block Report and associated parameters](#)
 - *4.1. [Classification of RTCP Metric Block parameters](#)
 - *4.1.1. [Application level parameters](#)
 - *4.1.2. [Transport level parameters](#)
 - *4.1.3. [End system parameters](#)
- *5. [Monitoring Methodology](#)
 - *5.1. [Option 1 - Monitoring every packet](#)
 - *5.2. [Option 2 - Real-time histogram methods](#)
 - *5.3. [Option 3 - Monitoring by exception](#)
 - *5.4. [Option 4 - Application-specific monitoring](#)
- *6. [Issues with RTCP XR extension](#)
- *7. [Guideline for reporting block format using RTCP XR](#)
 - *7.1. [Using small blocks](#)
 - *7.2. [Sharing the identity block](#)
 - *7.3. [Expanding the RTCP XR block namespace](#)
- *8. [An example of a metric block](#)
- *9. [Application to RFC 5117 topologies](#)
 - *9.1. [Applicability to MCU](#)
 - *9.2. [Application to translators](#)
- *10. [IANA Considerations](#)

- *11. [Security Considerations](#)
- *12. [Acknowledgement](#)
- *13. [Change Log](#)
- *13.1. [draft-hunt-avtcore-monarch-00](#)
- *13.2. [draft-hunt-avtcore-monarch-01](#)
- *13.3. [draft-hunt-avtcore-monarch-02](#)
- *14. [References](#)
- *[Authors' Addresses](#)

1. Introduction

Service providers and network providers today suffer from lack of good service that can monitor the performance at the user's home, handset or remote office. Without service performance metrics, it is difficult for network operators to properly locate the problem and solve service issues before problems impact subscriber/end user. The resolution generally involves deploying costly field network technician to conduct on-site troubleshooting and diagnostics. By reducing the expensive deployments with more automated remote monitoring capabilities, network operators can save significant costs, reduce mean time to repair and provide a better service offering.

As more users and subscribers rely on real time application services, uncertainties in the performance and availability of these services are driving the need to support new standard methods for gathering performance metrics from RTP applications. These rapidly emerging standards, such as RTCP XR [\[RFC3611\]](#) and other RTCP extension to Sender Reports(SR), Receiver Reports (RR) [\[RFC3550\]](#) are being developed for the purpose of collecting and reporting performance metrics from endpoint devices that can be used to correlate the metrics, provide end to end service visibility and measure and monitor QoE.

However the proliferation of RTP/RTCP specific metrics for transport and application quality monitoring has been identified as a potential problem for RTP/RTCP interoperability, which attempt to provide full coverage of all those parameters of concern to a specific application. Since different applications layered on RTP may have some monitoring requirements in common, therefore these metrics should be satisfied by a common design.

The objective of this document is to define an extensible RTP monitoring framework to provide a small number of re-usable QoS/QoE metrics which facilitate reduced implementation costs and help maximize inter-operability. [\[RFC5968\]](#) has stated that, where RTCP is to be extended with a new metric, the preferred mechanism is by the addition of a new RTCP XR [\[RFC3611\]](#) block. This memo assumes that any

requirement for a new metric to be transported in RTCP will use a new RTCP XR block.

2. Requirements notation

This memo is informative and as such contains no normative requirements.

3. RTP monitoring architecture

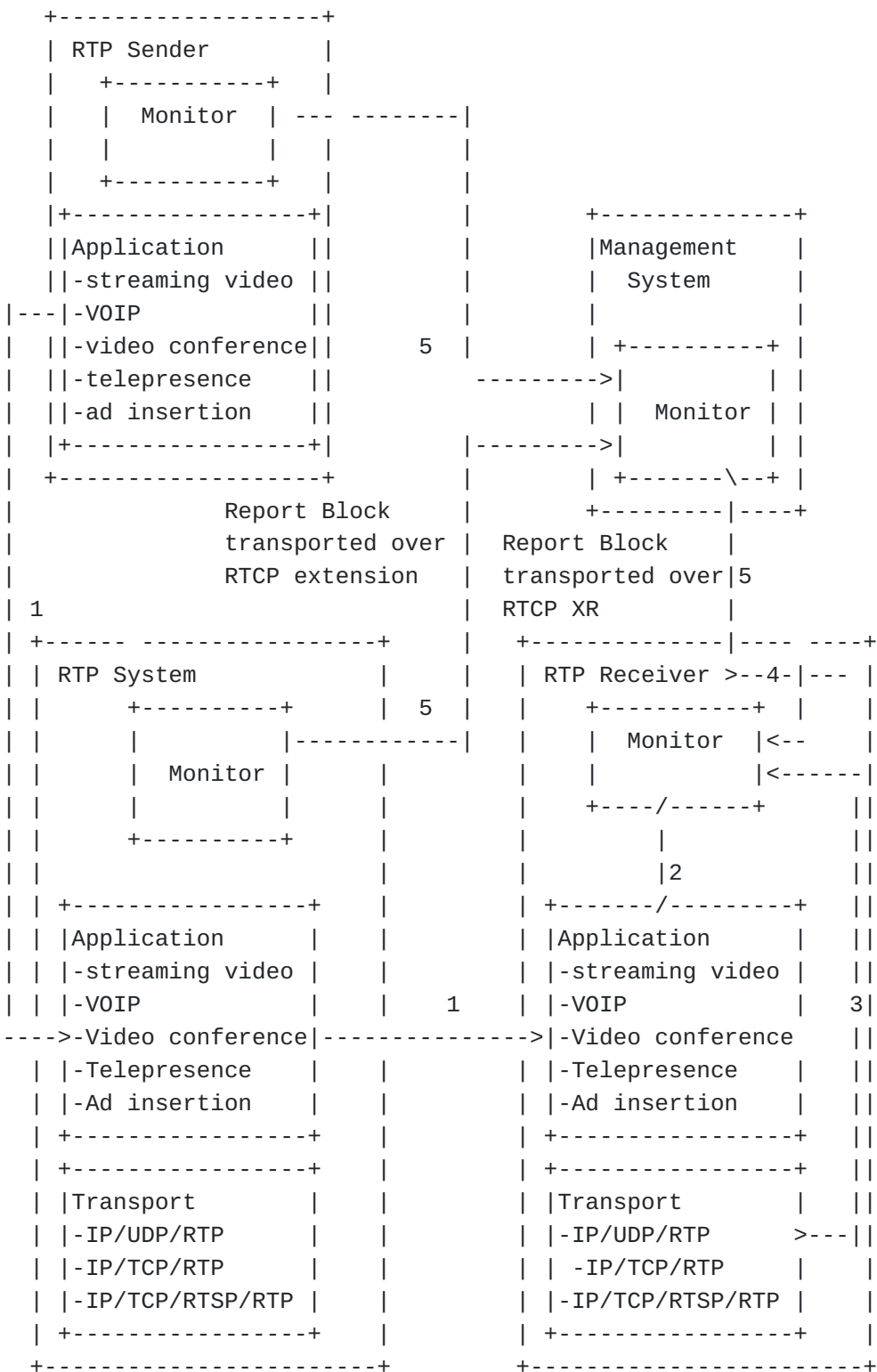
The RTP monitoring architecture comprises the following two functional components shown below:

*Monitor

*Metric Block Structure

Monitor is a functional component that acts as a source of information gathered for monitoring purposes. It may also collect statistics from multiple source, stores such information reported by RTCP XR or other RTCP extension appropriately as base metric or calculates composite metric. The end system that source RTP streams, or an intermediate-system that forwards RTP packets to End-devices that send or receive metric block can be envisioned to act as Monitor within the RTP monitoring architecture.

The Metric Block exposes real time Application Quality information in the report block format to monitor within the Network Management System. The RTCP or RTCP XR can be extended to convey such information.



1. RTP communication between real time applications

2. Application layer metrics

3. Transport layer metrics

4. End System metrics
5. Reporting Session- metrics transmitted over specified interfaces
6. Management application- monitor interaction using northbound interface. - Monitor outputs reports to the management application. The management application collects raw data from monitor, organizes database, conducts data analysis and creates alerts to the users. Note that Management application interaction with Monitor is out of scope of this document.

4. RTCP Metric Block Report and associated parameters

The basic RTCP Reception Report (RR) conveys reception statistics in metric block report format for multiple RTP media streams including [\[RFC3611\]](#) supplement the existing RTCP packets and provide more detailed feedback on reception quality in several categories:

- *transport level statistics
- *the fraction of packet lost since the last report
- *the cumulative number of packets lost
- *the highest sequence number received
- *an estimate of the inter-arrival jitter
- *and information to allow senders to calculate the network round trip time.

The RTCP XRs

- *Loss and duplicate RLE reports
- *Packet-receipt times reports
- *Round-trip time reports
- *Statistics Summary Reports

There are also various other scenarios in which it is desirable to send RTCP Metric reports more frequently. The Audio/Video Profile with Feedback [\[RFC4585\]](#) extends the standard A/V Profile [\[RFC3551\]](#) to allow RTCP reports to be sent early provided RTCP bandwidth allocation is respected. There are four use cases but are not limited to:

- *RTCP NACK is used to provide feedback on the RTP sequence number of the lost packets.

*RTCP XR is extended to provide feedback on multicast acquisition statistics information and parameters.

*RTCP is extended to convey requests for full intra-coded frames or select the reference picture, and signal changes in the desired temporal/spatial trade-off and maximum media bit rate.

*RTCP or RTCP XR is extended to provide feedback on ECN statistics information.

4.1. Classification of RTCP Metric Block parameters

4.1.1. Application level parameters

Measured data at the application level, i.e., QoE related parameters which focus on quality of content rather than network parameters. These include but are not limited to:

*Sound/Noise Level

*Echo return lost

*Statistics Summary Info, e.g., key frame lost key frame lost rate/ discard rate, key frame burst severity

*Codec Control

*Estimated Mean Opinion Score (MOS)

4.1.2. Transport level parameters

Measured data at the transport level. These include but are not limited to:

*Lost packets

*Round trip delay

*Jitter

*Congestion info

*FEC

*Codec Control

*Media Synchronization info

*Retransmission Info

*RAMS info

4.1.3. End system parameters

Measured data from application residing in that device. These include but are not limited to:

- *Error Concealment
- *FEC
- *Media Synchronization info
- *Jitter Buffer Lost
- *Jitter Buffer Delay

5. Monitoring Methodology

5.1. Option 1 - Monitoring every packet

The aim of "monitoring every packet" is to ensure that the information reported is not dependent on the application. In this scheme, RTP systems will report arrival data for each individual RTP packet. RTP (or other) systems receiving this "raw" data may use it to calculate any preferred heuristic metrics, but such calculations and the reporting of the results (e.g. to a session control layer or a management layer) are outside the scope of RTP and RTCP.

5.2. Option 2 - Real-time histogram methods

There are several potentially useful metrics which rely on the accumulation of a histogram in real time, so that a packet arrival results in a counter being incremented rather than in the creation of a new data item. These metrics may be gathered with a low and predictable storage requirement. Each counter corresponds to a single class interval or "bin" of the histogram. Examples of metrics which may be accumulated in this way include the observed distribution of packet delay variation, and the number of packets lost per unit time interval. Different networks may have very different expected and achieved levels of performance, but it may be useful to fix the number of class intervals in the reported histogram to give a predictable volume of data. This can be achieved by starting with small class intervals ("bin widths") and automatically increasing the width (e.g. by factors of two) if outliers are seen beyond the current upper limit of the histogram. Data already accumulated may be assigned unambiguously to the new set of bins, given some simple conditions on the relationship between the old and new origins and bin widths.

A significant disadvantage of the histogram method is the loss of any information about time-domain correlations between the samples which build the histogram. For example, a histogram of packet delay variation provides no indication of whether successive samples of packet delay

variation were uncorrelated, or alternatively that the packet delay variation showed a highly-correlated low-frequency wander.

5.3. Option 3 - Monitoring by exception

An entity which both monitors the packet stream, and has sufficient knowledge of the application to know when transport impairments may have degraded the application's performance, may choose to send exception reports containing details of the transport impairments to a receiving system. The crossing of a transport impairment threshold, or some application-layer event, would trigger such reports. RTP end systems and mixers are likely to contain application implementations which may, in principle, identify this type of exception.

It is likely that RTP translators will not contain suitable implementations which could identify such exceptions.

On-path devices such as routers and switches are not likely to be aware of RTP at all. Even if they are aware of RTP, they are unlikely to be aware of the RTP-level performance required by specific applications, and hence they are unlikely to be able to identify the level of impairment at which exceptional transport conditions may start to affect application performance.

This type of monitoring typically requires the storage of recent data in a FIFO (e.g. a circular buffer) so that data relevant to the period just before and just after the exception may be reported. It is not usually helpful to report transport data only from the period following an exception event detected by an application. This imposes some storage requirement (though less than needed for Option 1). It also implies the existence of additional cross-layer primitives or APIs to trigger the transport layer to generate and send its exception report. Such a capability might be considered architecturally undesirable, in that it complicates one or more interfaces above the RTP layer.

5.4. Option 4 - Application-specific monitoring

This is a business-as-usual option which suggests that the current approach should not be changed, based on the idea that previous application-specific approaches such as that of [RFC3611] were valid. If a large category of RTP applications (such as VoIP) has a requirement for a unique set of transport metrics, arising from its different requirements of the transport, then it seems reasonable for each application category to define its preferred set of metrics to describe transport impairments. We expect that there will be few such categories, probably less than 10.

It may be easier to achieve interworking for a well-defined set of application-specific metrics than it would be in the case that applications select a profile from a palette of many independent reusable metrics.

6. Issues with RTCP XR extension

Issues that have come up in the past with extensions to RTCP or RTCP XR include (but are probably not limited to) the following:

- *Repeating this information in a number of metrics blocks within the same RTCP packet may bring inefficiency when multiple small blocks in the same RTCP XR packet contain measurement data for the same stream and period.
- *RFC 3611 [\[RFC3611\]](#) defines seven report block formats for network management and quality monitoring. However some of these block types defined in [\[RFC3611\]](#) are only specifically designed for conveying multicast inference of network characteristics(MINC) or voice over IP (VoIP) monitoring.
- *Designing a single report block or metric containing a large number of parameters in different classes for a specific application may increase implementation cost and minimize interoperability.
- *The RTCP XR block namespace is limited by the 8-bit block type field in the RTCP XR header Under current allocation pressure, we expect that the RTCP XR Block Type space will be exhausted soon. We therefore need a way to extend the block type space, so that new specifications may continue to be developed.

7. Guideline for reporting block format using RTCP XR

7.1. Using small blocks

Different applications using RTP for media transport certainly have differing requirements for metrics transported in RTCP to support their operation. For many applications, the basic metrics for transport impairments provided in RTCP SR and RR packets [\[RFC3550\]](#) (together with source identification provided in RTCP SDES packets) are sufficient. For other applications additional metrics may be required or at least sufficiently useful to justify the overheads, both of processing in endpoints and of increased session bandwidth. For example an IPTV application using Forward Error Correction (FEC) might use either a metric of post-repair loss or a metric giving detailed information about pre-repair loss bursts to optimise payload bandwidth and the strength of FEC required for changing network conditions. However there are many metrics available. It is likely that different applications or classes of applications will wish to use different metrics. Any one application is likely to require metrics for more than one parameter but if this is the case, different applications will almost certainly require different combinations of metrics. If larger blocks are defined containing multiple metrics to address the needs of each application,

it becomes likely that many different such larger blocks are defined, which becomes a danger to interoperability.

To avoid this pitfall, this memo proposes the use of small RTCP XR metrics blocks each containing a very small number of individual metrics characterising only one parameter of interest to an application running over RTP. For example, at the RTP transport layer, the parameter of interest might be packet delay variation, and specifically the metric "IPDV" defined by [\[Y1540\]](#). See [Section 8](#) for architectural considerations for a metrics block, using as an example a metrics block to report packet delay variation.

[7.2. Sharing the identity block](#)

Any measurement must be identified. However if metrics are delivered in small blocks there is a danger of inefficiency arising from repeating this information in a number of metrics blocks within the same RTCP packet, in cases where the same identification information applies to multiple metrics blocks.

An instance of a metric must be identified using information which is likely to include most of the following:

- *the node at which it was measured,
- *the source of the measured stream (for example, its CNAME),
- *the SSRC of the measured stream,
- *the sequence number of the first packet of the RTP session,
- *the extended sequence numbers of the first packet of the current measurement interval, and the last packet included in the measurement,
- *the duration of the most recent measurement interval and
- *the duration of the interval applicable to cumulative measurements (which may be the duration of the RTP session to date).

Note that this set of information may overlap with, but is more extensive than, that in the union of similar information in RTCP RR packets. However we can not assume that RR information is always present when XR is sent, since they may have different measurement intervals. Also the reason for the additional information carried in the XR is the perceived difficulty of "locating" the *start* of the RTP session (sequence number of 1st packet, duration of interval applicable to cumulative measurements) using only RR. However when an RTCP XR packet containing more than two metrics blocks, reporting on the same streams from the same source, each metric block should have the same measurement identify, if each metric block carry the duplicated data

for the measurement identity ,it leads to redundant information in this design since equivalent information is provided multiple times, once in *every* identification packet. Though this ensures immunity to packet loss, the design bring more complexity and the overhead is not completely trivial.

This section proposes an approach to minimise the inefficiency of providing this identification information, assuming that an architecture based on small blocks means that a typical RTCP packet will contain more than one metrics block needing the same identification. The choice of identification information to be provided is discussed in [\[IDENTITY\]](#) (work in progress).

The approach is to define a stand-alone block containing only identification information, and to tag this identification block with a number which is unique within the scope of the containing RTCP XR packet. The "containing RTCP XR packet" is defined here as the RTCP XR header with PT=XR=207 defined in Section 2 of [\[RFC3611\]](#) and the associated payload defined by the length field of this RTCP XR header. The RTCP XR header itself includes the SSRC of the node at which all of the contained metrics were measured, hence this SSRC need not be repeated in the stand-alone identification block. A single containing RTCP XR packet may contain multiple identification blocks limited by the range of the tag field. Typically there will be one identification block per monitored source SSRC, but the use of more than one identification block for a single monitored source SSRC within a single containing RTCP XR packet is not ruled out.

There will be zero or more metrics blocks dependent on each identification block. The dependence of an instance of a metrics block on an identification block is established by the metrics block's having the same numeric value of the tag field as its identification block (in the same containing RTCP XR packet).

[Figure 2](#) below illustrates this principle using as an example an RTCP XR packet containing four metrics blocks, reporting on streams from two sources. The measurement identity information is provided in two blocks with Block Type NMI, and tag values 0 and 1 respectively.

Note: in this example, RTCP XR block type values for four proposed new block types (work in progress) are given as NMI, NPDV, NBGL and NDEL. These represent numeric block type codepoints to be allocated by IANA at the conclusion of the work.

Each of these two identity blocks will specify the SSRC of one of the monitored streams, as well as information about the span of the measurement. There are two metrics blocks with tag=0 indicating their association with the measurement identity block which also has tag=0. These are the two blocks following the identity block with tag=0, though this positioning is not mandatory. There are also two metrics blocks with tag=1 indicating their association with the measurement identity block which also has tag=1, and these are the two blocks following the identity block with tag=1.

Note that if metrics blocks associated with an identity block must always follow the identity block, we could save the tag field and

possibly simplify processing. However depending on ordering of metric block and identity block may bring inefficiency since you do not know which block is the last metric block associated with identity block. Hence it is more desirable to cross-referencing with a numeric tag, i.e., using tag field to associated metric block with identity block.

In the example, the block types of the metrics blocks associated with tag=0 are BT=NPDV (a PDV metrics block) and BT=NBGL (a burst and gap loss metrics block). The block types of the metrics blocks associated with tag=1 are BT=NPDV (a second PDV metrics block) and BT=NDEL (a delay metrics block). This illustrates that:

- *multiple instances of the same metrics block may occur within a containing RTCP XR packet, associated with different identification information, and

- *differing measurements may be made, and reported, for the different streams arriving at an RTP system.

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|V=2|P|reserved |   PT=XR=207   |           length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SSRC of RTCP XR packet sender           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BT=NMI   |0|tag=0| resv |           block length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SSRC of stream source 1           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.       ...measurement identity information, source 1...       .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BT=NPDV   |I|tag=0|pdvtyp |           block length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.       ...PDV information for source 1...       .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BT=NBGL   |I|tag=0| resv |           block length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.       ...burst-gap-loss information for source 1...       .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BT=NMI   |0|tag=1| resv |           block length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SSRC of stream source 2           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.       ...measurement identity information, source 2...       .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BT=NPDV   |I|tag=1|pdvtyp |           block length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.       ...PDV information for source 2...       .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BT=NDEL   |I|tag=1| resv |           block length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.       ...delay information for source 2...       .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

This approach of separating the identification information is more costly than providing identification in each metrics block if only a single metrics block is sent in an RTCP packet, but becomes beneficial as soon as more than one metrics block shares common identification.

[7.3. Expanding the RTCP XR block namespace](#)

The RTCP XR block namespace is limited by the 8-bit block type field in the RTCP XR header (Section 3 of [\[RFC3611\]](#)). IESG have noted that this is potentially restrictive. It would be possible to standardize an expansion mechanism, probably based on use of a new field near the start of the variable-length "type-specific block contents" field. Clearly this could apply only to new block types, so might be

standardized to apply to some subrange of the current 8-bit range, for example the range 128 through 191 might be used. At time of writing, block types 12 to 254 are unassigned and 255 is reserved for future expansion. One potential use is through hierarchical control, where one or a few codepoints at the top level are given to other SDOs who may then define a number of metrics distinguished by values in the (so far hypothetical) new field.

8. An example of a metric block

This section uses the example of an existing proposed metrics block to illustrate the application of the principles set out in [Section 7.1](#). The example [\[PDV\]](#) (work in progress) is a block to convey information about packet delay variation (PDV) only, consistent with the principle that a metrics block should address only one parameter of interest. One simple metric of PDV is available in the RTCP RR packet as the "jit" field. There are other PDV metrics which may be more useful to certain applications. Two such metrics are the IPDV metric ([\[Y1540\]](#), [\[RFC3393\]](#)) and the MAPDV2 metric [\[G1020\]](#). Use of these metrics is consistent with the principle in Section 5 of [\[RFC5968\]](#) that metrics should usually be defined elsewhere, so that RTCP standards define only the transport of the metric rather than its nature. The purpose of this section is to illustrate the architecture using the example of [\[PDV\]](#) (work in progress) rather than to document the design of the PDV metrics block or to provide a tutorial on PDV in general. Given the availability of at least three metrics for PDV, there are design options for the allocation of metrics to RTCP XR blocks:

- *provide an RTCP XR block per metric
- *provide a single RTCP XR block which contains all three metrics
- *provide a single RTCP block to convey any one of the three metrics, together with a identifier to inform the receiving RTP system of the specific metric being conveyed

In choosing between these options, extensibility is important, because additional metrics of PDV may well be standardised and require inclusion in this framework. The first option is extensible but only by use of additional RTCP XR blocks, which may consume the limited namespace for RTCP XR blocks at an unacceptable rate. The second option is not extensible, so could be rejected on that basis, but in any case a single application is quite unlikely to require transport of more than one metric for PDV. Hence the third option was chosen. This implies the creation of a subsidiary namespace to enumerate the PDV metrics which may be transported by this block, as discussed further in [\[PDV\]](#) (work in progress).

9. Application to RFC 5117 topologies

An RTP system (end system, mixer or translator) which originates, terminates or forwards RTCP XR blocks is expected to handle RTCP, including RTCP XR, as specified in [\[RFC3550\]](#) for that class of RTP systems. Provided this expectation is met, an RTP system using RTCP XR is architecturally no different from an RTP system of the same class (end system, mixer, or translator) which does not use RTCP XR. This statement applies to the topologies investigated in [\[RFC5117\]](#), where they use RTP end systems, RTP mixers and RTP translators as these classes are defined in [\[RFC3550\]](#).

These topologies are specifically Topo-Point-to-Point, Topo-Multicast, Topo-Translator (both variants, Topo-Trn-Translator and Topo-Media-Translator, and combinations of the two), and Topo-Mixer.

9.1. Applicability to MCU

The topologies based on systems which do not behave according to [\[RFC3550\]](#), that is Topo-Video-Switch-MCU and Topo-RTCP-terminating-MCU, suffer from the difficulties described in [\[RFC5117\]](#). These difficulties apply to systems sending, and expecting to receive, RTCP XR blocks as much as to systems using other RTCP packet types. For example, a participant RTP end system may send media to a video switch MCU. If the media stream is not selected for forwarding by the switch, neither RTCP RR packets nor RTCP XR blocks referring to the end system's generated stream will be received at the RTP end system. Strictly the RTP end system can only conclude that its RTP has been lost in the network, though an RTP end system complying with the robustness principle of [\[RFC1122\]](#) should survive with essential functions unimpaired.

9.2. Application to translators

Section 7.2 of [\[RFC3550\]](#) describes processing of RTCP by translators. RTCP XR is within the scope of the recommendations of [\[RFC3550\]](#). Some RTCP XR metrics blocks may usefully be measured at, and reported by, translators. As described in [\[RFC3550\]](#) this creates a requirement for the translator to allocate an SSRC for itself so that it may populate the SSRC in the RTCP XR packet header (although the translator is not a Synchronisation Source in the sense of originating RTP media packets). It must also supply this SSRC and the corresponding CNAME in RTCP SDES packets.

In RTP sessions where one or more translators generate any RTCP traffic towards their next-neighbour RTP system, other translators in the session have a choice as to whether they forward a translator's RTCP packets. Forwarding may provide additional information to other RTP systems in the connection but increases RTCP bandwidth and may in some cases present a security risk. RTP translators may have forwarding behaviour based on local policy, which might differ between different interfaces of the same translator.

For bidirectional unicast, an RTP system may usually detect RTCP XR from a translator by noting that the sending SSRC is not present in any RTP media packet. However even for bidirectional unicast there is a possibility of a source sending RTCP XR before it has sent any RTP media (leading to transient mis-categorisation of an RTP end system or RTP mixer as a translator), and for multicast sessions - or unidirectional/streaming unicast - there is a possibility of a receive-only end system being permanently mis-categorised as a translator. Hence it is desirable for a translator that send XR to have a way to declare itself explicitly.

10. IANA Considerations

None.

11. Security Considerations

This document itself contains no normative text and hence should not give rise to any new security considerations, to be confirmed.

12. Acknowledgement

The authors would like to thank Colin Perkins, Graeme Gibbs, Debbie Greenstreet, Keith Drage, Dan Romascanu, Ali C. Begen, Roni Even for their valuable comments and suggestions on the early version of this document.

13. Change Log

13.1. draft-hunt-avtcore-monarch-00

The following are the major changes compared to previous version:

- *Provide some background texts and related work into Introduction section.
- *Add a new section 3 to describe RTP monitoring architecture.
- *Add a new section 4 to describe RTCP Metric Block Report and associated parameters.
- *Move section 3.1, 3.2, 3.3 and 3.4 in draft-hunt-avt-monarch-00 to this version as section 5 to describe Monitoring Methodology.
- *Add a new section 6 to describe Issues with RTCP XR extension.
- *Merge section 3, 4, 8 in previous version into one new section 9 to describe Guideline for reporting block format using RTCP XR.

*Merge section 6,7 in previous version into one new section 9 to describe Application to RFC 5117 topologies.

13.2. draft-hunt-avtcore-monarch-01

The following are the major changes compared to previous version:

*Update figure 1 to describe the interface between RTP Sender and Report Collector in precise granularity.

*Add some texts to define the role of RRW and RRC.

*Correct the order of the second figure in the document.

*Other editorial changes.

13.3. draft-hunt-avtcore-monarch-02

The following are the major changes compared to previous version:

*Substitute the two components in the monitoring architecture with monitor functionality to get alignment with RFC3550 terminology.

*Clear up all the Editor Notes in the document.

*Add one more issue with RTCP XR extension.

*Add some texts to clarify the RRC interface with Network Management Application.

14. References

[RFC1122]	Braden, R., " Requirements for Internet Hosts -- Communication Layers ", RFC 1122, October 1989.
[RFC3393]	Demichelis, C., " IP Packet Delay Variation Metric for IP Performance Metrics (IPPM) ", RFC 3393, November 2002.
[RFC3550]	Schulzrinne, H., " RTP: A Transport Protocol for Real-Time Applications ", RFC 3550, July 2003.
[RFC3611]	Friedman, T. (Ed), " RTP Control Protocol Extended Reports (RTCP XR) ", RFC 3611, November 2003.
[RFC5117]	Westerlund, M., " RTP Topologies ", RFC 5117, January 2008.
[RFC5968]	Ott, J. and C. Perkins, " Guidelines for Extending the RTP Control Protocol (RTCP) ", RFC 5968, September 2010.
[RFC4585]	Ott, J. and S. Wenger, " Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF) ", RFC 4585, July 2006.
[RFC3551]	

	Schulzrinne , H. and S. Casner, " Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF) ", RFC 3551, July 2003.
[IDENTITY]	Hunt, G., "RTCP XR Report Block for Measurement Identity", ID draft-ietf-avt-rtcp-xr-meas-identity-02, May 2009.
[PDV]	Hunt, G., "RTCP XR Report Block for Packet Delay Variation Metric Reporting", ID draft-ietf-avt-rtcp-xr-pdv-03, May 2009.
[G1020]	, , "ITU-T Rec. G.1020, Performance parameter definitions for quality of speech and other voiceband applications utilizing IP networks", July 2006.
[Y1540]	, , "ITU-T Rec. Y.1540, IP packet transfer and availability performance parameters", November 2007.

Authors' Addresses

Geoff Hunt
 Hunt BT Orion 1 PP2 Adastral Park Martlesham Heath
 Ipswich, Suffolk IP5 3RE United Kingdom Phone: +44 1473 651704
 EMail: geoff.hunt@bt.com

Philip Arden
 Arden BT Orion 3/7 PP4 Adastral Park Martlesham Heath
 Ipswich, Suffolk IP5 3RE United Kingdom Phone: +44 1473 644192
 EMail: philip.arden@bt.com

Qin Wu editor
 Wu Huawei 101 Software Avenue, Yuhua District Nanjing,
 Jiangsu 210012 China EMail: sunseawq@huawei.com