                  Real-time Application Quality-of-Service
                        Monitoring (RAQMON) Framework

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   There is a need to monitor end-devices such as IP phones, pagers,
   Instant Messaging clients, mobile phones, and various other handheld
   computing devices. This memo extends the remote network monitoring
   (RMON) family of specifications to allow real-time quality-of-service
   (QoS) monitoring of various applications that run on these devices
   and allows this information to be integrated with the RMON family
   using the Simple Network Management Protocol (SNMP). This memo
   defines the framework, architecture, relevant metrics, and transport
   requirements for real-time QoS monitoring of applications.

Table of Contents

1. Introduction

   With the growth of the Internet and advancements in embedded
   technologies, smart IP devices (such as IP phones, pagers, instant
   message clients, mobile phones, wireless handhelds, and various other
   computing devices) have become an integral part of our day-to-day
   operations. Enterprise operators, information technology (IT)
   managers, application service providers, network service providers,
   and so on, need to monitor these application and device types in
   order to ensure that end user quality-of-service (QoS) objectives are
   met. This memo describes a monitoring solution for these
   environments, extending the remote network monitoring (RMON) family
   of specifications [RFC2819]. These extensions support real-time QoS
   monitoring of typical applications that run on end-devices mentioned
   above, and they allow this information to be integrated using the
   familiar RMON family of specifications via SNMP [RFC3416].

   The Real-time Application QoS Monitoring Framework (RAQMON) allows
   end-devices and applications to report QoS statistics in real time.
   Many real-time applications (as well as non-real-time applications
   managed within the RMON family of specifications) can report
   application-level QoS statistics in real time using the RAQMON
   Framework outlined in this memo. Some possible applications
   scenarios include applications such as Voice over IP, Fax over IP,
   Video over IP, Instant Messaging (IM), Email, software download
   applications, e-business style transactions, web access from handheld
   computing devices, etc.

   The user experience of an application running on an IP end-device
   depends upon the type of application the user is running and the
   surrounding resources available to that application. An end-to-end
   application QoS experience is a compound effect of various
   application-level transactions and available network and host
   resources. For example, the end-to-end user experience of a Voice
   over IP (VoIP) call depends on the total time required to set up the
   call as much as on media-related performance parameters such as end-
   to-end network delay, jitter, packet loss, and the type of codec used
   in a call. The performance of a VoIP call is also influenced by
   behavior of network protocols like the Reservation Protocol (RSVP),
   explicit tags in differentiated services (DiffServ) [RFC2475] or IEEE
   802.1 [IEEE802.1D] along with available host resources such as device
   CPU or memory utilized by other applications while the call is
   ongoing.

   The end-to-end application quality of service (QoS) experience is
   application context sensitive. For example, the kinds of parameters
   reported by an IP telephony application may not really be needed for
   other applications such as Instant Messaging. The RAQMON Framework

offers a mechanism to report the end-to-end QoS experience
appropriate for a specific application context by providing
mechanisms to report a subset of metrics from a pre-defined list.

In order to facilitate a complete end-to-end view, RAQMON correlates
statistics that involve:

   i.   "User, Application, Session"-specific parameters (e.g.,
         session setup time, session duration parameters based on
         application context).

   ii.   "IP end-device"-specific parameters during a session (e.g.,
         CPU usage, memory usage).

   iii. "Transport network"-specific parameters during a session
         (e.g., end-to-end delay, one-way delay, jitter, packet loss
         etc).

At any given point, the applications at these devices can correlate
such diverse data and report end-to-end performance. The RAQMON
Framework specified in this memo offers a mechanism to report such
end-to-end QoS view and integrate such a view into the RMON family of
specifications. In particular, the RAQMON Framework specifies the
following:

   a. A set of basic metrics sent as reports between the RAQMON
      entities using for transport existing Internet Protocols such
      as TCP or SNMP.

   b. Requirements to be met by the underlying transport protocols
      that carry the RAQMON reports.

   c. A portion of the Management Information Base (MIB) as an
      extension of the RMON MIB Modules for use with network
      management protocols in the Internet community.

This memo provides the RAQMON functional architecture, RAQMON entity
definitions and requirements, requirements for the transport
protocols, a set of metrics, and an information model for the RAQMON
reports.

Supplementary memos will describe the mapping of the basic RAQMON
metrics onto different transport protocols. For example, the RAQMON
PDU [RFC4712] memo provides definitions of syntactical PDU structure
and use case scenarios of transmission of such PDUs over the
Transmission Control Protocol (TCP) and the Simple Network Management
Protocol (SNMP).

The RAQMON MIB [RFC4711] memo describes the Management Information
Base (MIB) for use with the SNMP protocol in the Internet community.
The document proposes an extension to the Remote Monitoring MIB
[RFC2819] to accommodate RAQMON solutions.

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

2. RAQMON Functional Architecture

The RAQMON Framework extends the architecture created in the RMON MIB
[RFC2819] by providing application performance information as
experienced by end-users. The RAQMON architecture is based on three
functional components named below:

    - RAQMON Data Source (RDS)

    - RAQMON Report Collector (RRC)

    - RAQMON MIB Structure

A RAQMON Data Source (RDS) is a functional component that acts as a
source of data for monitoring purposes. End-devices like IP phones,
cell phones, and pagers, and application clients like instant
messaging clients, soft phones in PCs, etc., are envisioned to act as
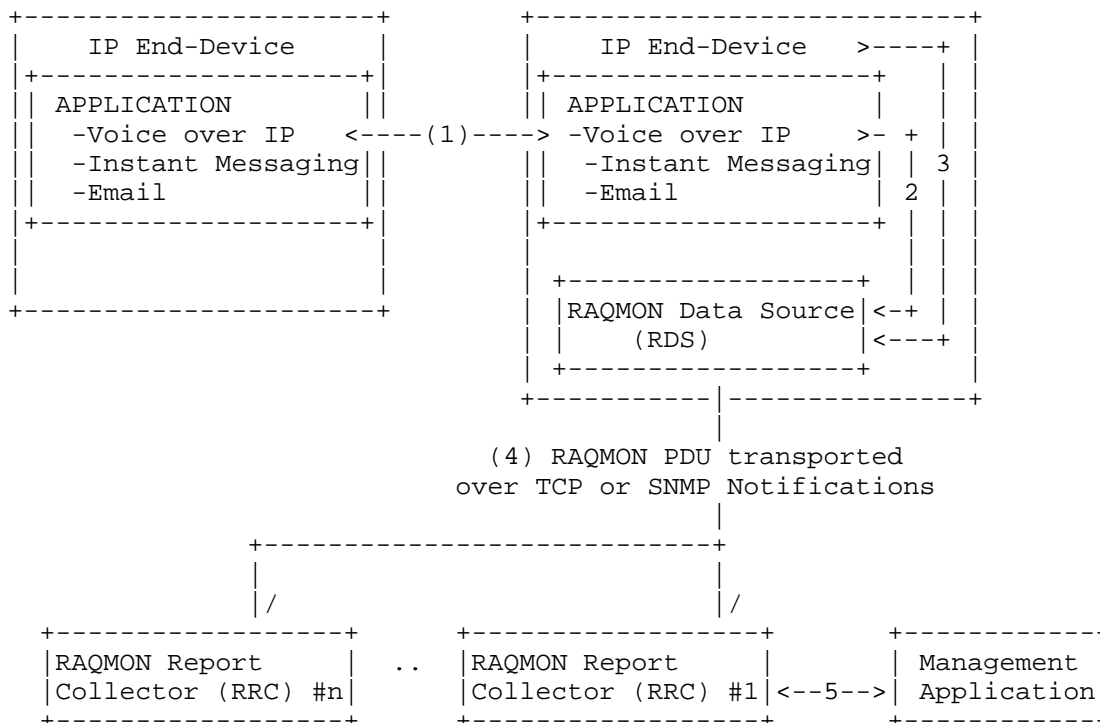RDSs within the RAQMON Framework.

```
+--------------------+        +-------------------------+
|     IP End-Device  |        |    IP End-Device  >----+ |
|+------------------+|        |+-----------------+   | |
|| APPLICATION      ||        || APPLICATION     |   | |
||  -Voice over IP  <----(1)----> -Voice over IP  >- + | |
||  -Instant Messaging||       ||  -Instant Messaging| | 3 |
||  -Email          ||        ||  -Email          | 2 | |
|+------------------+|        |+-----------------+   | | |
|                    |        |                   | | |
|                    |        | +---------------+  | | |
+--------------------+        | |RAQMON Data Source|<-+ | |
                              | |     (RDS)       |<---+ |
                              | +---------------+       |
                              +----------|--------------+
                                         |
                          (4) RAQMON PDU transported
                          over TCP or SNMP Notifications
                                         |
           +---------------------------+ |
           |                           | |
           |/                          |/
  +----------------+        +----------------+     +-----------+
  |RAQMON Report   |  ..    |RAQMON Report   |     | Management |
  |Collector (RRC) #n|      |Collector (RRC) #1|<--5-->| Application|
  +----------------+        +----------------+     +-----------+
```

Figure 1 - RAQMON Framework.

(1) Communication Session between real-time applications

(2) Context-Sensitive Metrics

(3) Device State Specific Metrics

(4) Reporting session - RAQMON metrics transmitted over specified
      interfaces (Specific Protocol Interface, IP Address, port)

(5) Management Application - RRC interaction using the RAQMON MIB

A RAQMON Report Collector (RRC) collects statistics from multiple
RDSs, analyzes them, and stores such information appropriately. RRC
is envisioned to be a network server, serving an administrative
domain defined by the network administrator. The RRC component of
the RAQMON architecture is envisioned to be computationally
resourceful. Only RRCs should implement the RAQMON MIB module.

The RAQMON Management Information Base (RAQMON MIB) extends the
Remote Monitoring MIB [RFC2819] to accommodate the RAQMON Framework
and exposes End-to-End Application QoS information to Network
Management Applications.

2.1. RAQMON Data Source (RDS)

2.1.1. RAQMON Data Source (RDS) Functional Architecture

A RAQMON Data Source (RDS) is a source of data for monitoring
purposes. The RDS monitoring function is performed in real time
during communication sessions. The RDS entities capture QoS
attributes of such communication sessions and report them within a
RAQMON "reporting session".

An RDS is primarily responsible for abstracting IP end-devices and
applications within the RAQMON architecture. It gathers the
parameters for a particular communication session and forwards them
to the appropriate RAQMON Report Collector (RRC). Since it is
envisioned that the RDS functionality will be realized by writing
firmware/software running on potentially small, low-powered end-
devices, the design of the RDS element is optimized towards that end.
Like the implementations of routing and management protocols, an
implementation of RDS in an end-device will typically execute in the
background, not in the data-forwarding path.

RDSs use a PUSH mechanism to report QoS parameters. While the
applications running on the RDS decide about the content of the PDU
appropriate for an application context, an RDS asynchronously sends
out reports to RRC.

The rate at which PDUs are sent from RDSs to RRCs is controlled by
the applications' administrative domain policy. While this mechanism
provides flexibility to gather a detailed end-to-end experience
required by IT managers and system administrators, certain steps
should be followed to operate RAQMON in congestion-safe manner.
Section 3 addresses steps required for congestion-safe operation.

An RDS reports QoS statistics for simplex flows. At a given
instance, a report from RDS is logically viewed as a collection of
QoS parameters associated with a communication session as perceived
by the reporting RDS. For example, if two IP phone users, Alice and
John, are involved in a communication session, the end-to-end delay
experienced by the IP phone user Alice could be different from the
one experienced by the IP phone user John for a variety of reasons.
Hence, a report from Alice's IP phone represents the QoS performance
of that call as perceived by the RDS that resides in Alice's IP
phone.

2.1.2. RAQMON Data Source (RDS) Requirements

    1. RAQMON Data Sources SHALL gather reports from multiple
       applications residing in that device and SHALL send out
       compound QoS reports associated with multiple communication
       sessions at a given moment.

       Examples include a conference bridge hosting several different
       conference calls or a two-party video call consisting of
       audio/video sessions. In each case an RDS could send out one
       single RAQMON report that consists of multiple sub-reports
       associated with audio and video sessions or sub-reports for
       each conference call.

    2. RAQMON Data Sources MUST implement the TCP transport and MAY
       implement the SNMP transport.

2.1.3. Configuring RAQMON Data Sources

   In order to report statistics to RAQMON Report Collectors, RDSs will
   need to be configured with the following parameters:

    1. The time interval between RAQMON PDUs. This parameter MUST be
       configured such that overflow of any RAQMON parameter within a
       PDU between consecutive transmissions is avoided.

    2. The IP address and port of target RRC.

   An RDS may use manual configuration for the RDS configuration
   parameters using command line interface (CLI), Telephone User
   Interface (TUI), etc.

   One of the following mechanisms to gain access to configuration
   parameters can also be considered:

    - RDS acts as a trivial file transfer protocol (TFTP) client and
       downloads text scripts to read the parameters.
    - RDS acts as a Dynamic Host Configuration Protocol (DHCP) Client
       and gets RRC addressing information as a DHCP option.
    - RDS acts as a DNS client and gets target collector information
       from a DNS Server.
    - RDS acts as a LDAP Client and uses directory look-ups.

   Identifying the DHCP option and structure to use, defining the
   structure of the configuration information in DNS, or defining a LDAP
   schema could be explored as items of future work.

Compliance to the RAQMON specification does not require usage of any specific configuration mechanisms mentioned above. It is left to the implementers to choose appropriate provisioning mechanisms for a system.

2.2. RAQMON Report Collector (RRC)

2.2.1. RAQMON Report Collector (RRC) Functional Architecture

A RAQMON Report Collector (RRC) receives RAQMON PDUs from multiple RDSs and analyzes and stores the information in the RAQMON MIB. The RRC is envisioned to be computationally resourceful, providing a storage and aggregation point for a set of RDSs.

Since RDSs can belong to separate administrative domains, the RAQMON Framework allows RDSs to report QoS parameters to separate RRCs. Vendors can develop a management application to correlate information residing in different RRCs across multiple administrative domains to represent one communication session. However, such an application-level specification is beyond the scope of this memo.

2.2.2. RAQMON Report Collector (RRC) Requirements

1. RAQMON Report Collectors MUST support the mandatory mapping over TCP of the RAQMON information model defined in [RFC4712] with the purpose of receiving RAQMON reports from RAQMON Data Sources (RDS).

2. RAQMON Report Collectors MAY support the optional mapping over SNMP notifications of the RAQMON information model defined in [RFC4712].

3. RAQMON Report Collectors MUST implement session timeout mechanisms to assume end of reporting for RDSs that have been out of reporting for a reasonable duration of time. Such timeout parameters SHOULD be configurable in vendor implementations, as programmable parameters at deployment.

4. RAQMON Report Collectors MUST support the RAQMON-MIB module and meet the compliance requirements of the raqmonCompliance MODULE-COMPLIANCE definition as described in [RFC4711]. The population of the RAQMON MIB with performance monitoring information is independent of the transport protocol, or protocols used to carry the information between RDSs and RRCs.

2.3. Information Model and RAQMON Protocol Data Unit (PDU)

2.3.1. RAQMON Information Model

   RAQMON defines a set of basic metrics that characterize the QoS of
   applications, as reported by RAQMON Data Sources. This basic set of
   metrics is defined in Section 5 of this memo. There is no minimal
   requirement for a mandatory set of metrics to be supported by an RDS.

   Specific applications, new types of network appliances or new methods
   to measure and characterize the QoS of applications lead to the
   requirement for the information model to be extensible. To answer
   this need, the information model is designed so that vendors can
   extend it by adding new metrics.

   Although NOT REQUIRED for RAQMON conformance, extensions of the
   information model can offer useful information for specific
   applications. An example of metrics that can extend the basic RAQMON
   information model are the detailed metrics for VoIP media monitoring
   and call quality included in the VoIP Metrics Report Block defined in
   [RFC3611].

   The RAQMON Information model is expressed by defining a conceptual
   RAQMON Protocol Data Unit (PDU).

2.3.2. RAQMON Protocol Data Unit

   A RAQMON Protocol Data Unit (PDU) is a common data format understood
   by RDSs and RRCs. A RAQMON PDU does not transport application data
   but rather occupies the place of a payload specification at the
   application layer of the protocol stack. Different transport
   mappings may be used to carry RAQMON PDU between RDSs and RRCs.
   Transport protocol requirements are being defined in Section 2.4 of
   this memo.

   Though architected conceptually as a single PDU, the RAQMON PDU is
   functionally divided into two different parts. They are the BASIC
   part, and the Application-Specific Extensions, required for
   application-, vendor-, and device-specific extensions.

   The BASIC part of the RAQMON PDU:
      The BASIC part of the RAQMON PDU follows the SMI Network
      Management Private Enterprise Code 0, indicating an IETF standard
      construct. The RAQMON PDU BASIC part offers an entry-type from a
      pre-defined list of QoS parameters defined in Section 5 and allows
      applications to fill in appropriate values for those parameters.
      Application developers also have the flexibility to make an RDS
      report built only of a subset of the parameters listed in

Section 5. There is no need to carry all metrics in every PDU;
moreover, it is RECOMMENDED that static or pseudo-static metrics
that do not change or seldom change for a given session or
application will be send only when the session or application are
initiated, and then at large time intervals.

The Application part of RAQMON PDU:
Since it is difficult to structure a BASIC part that meets the
needs of all applications, RAQMON provides extension capabilities
to convey application-, vendor-, and device-specific parameters
for future use. Additional parameters can be defined within
payload of the APP part of the PDU by the application developers
or vendors. The owner of the definition of the application part
of the RAQMON PDU is indicated by a vendor's SMI Network
Management Private Enterprise Code defined in
http://www.iana.org/assignments/enterprise-numbers. Such
application-specific extensions should be maintained and published
by the application vendor.

Though RDSs and RRCs are designed to be stateless for an entire
reporting session, the framework requires an indication for the end
of the reporting. For this purpose, an RDS MUST send a RAQMON NULL
PDU. A NULL PDU is a RAQMON PDU containing ALL NULL values (i.e.,
nothing to report).

2.4. RDS/RRC Network Transport Protocol Requirements

The RAQMON PDUs rely on the underlying protocol(s) to provide
transport functionalities and other attributes of a transport
protocol, e.g., transport reliability, re-transmission, error
correction, length indication, congestion safety,
fragmentation/defragmentation, etc. The maximum length of the RAQMON
data packet is limited only by the underlying protocols.

The following requirements MUST be met by the transport protocols:

   1. The transport protocol SHOULD allow for RDS lightweight
      implementations. RDSs will be implemented on low-powered
      embedded devices with limited device resources.

   2. Scalability - Since RRCs need to interact with a very large
      number (many tens, many hundreds, or more) of RDSs, scalability
      of the transport protocol is REQUIRED.

   3. Congestion safety - as per [RFC2914]. See also Section 3.

   4. Security - Since RAQMON statistics may carry sensitive system
      information requiring protection from unauthorized disclosure
      and modification in transit, a transport protocol that provides
      strong secure modes or allows for data encryption and integrity
      to be applied is REQUIRED.

   5. NAT-Friendly - The transport protocol SHOULD comply with
      [RFC3235], so that an RDS could communicate with an RRC through
      a Firewall/Network Address Translation device.

   6. The transport protocol MAY implement session timeout mechanisms
      to assume end of reporting for RDSs that have been out of
      reporting for a reasonable duration of time. Such timeout
      parameters SHOULD be configurable in vendor implementations,
      programmable at deployment.

   7. Reliability - The RAQMON Framework expects PDUs to operate in
      lossy networks. However, retransmission is not included in the
      RAQMON framework, in order to keep the design simple. If
      retransmission is a necessity, RAQMON MAY operate over
      transport protocols, such as TCP.

   In the future, if RAQMON PDUs are to be carried in an underlying
   protocol that provides the abstraction of a continuous octet stream
   rather than messages (packets), an encapsulation for the RAQMON
   packets must be defined to provide a framing mechanism. Framing is
   also needed if the underlying protocol contains padding so that the
   extent of the RAQMON payload cannot be determined. No framing
   mechanism is defined in this document. Carrying several RAQMON
   packets in one network or transport packet reduces header overhead.

   Further memos like [RFC4712] describe how the PDU is transported over
   existing protocols like the Transmission Control Protocol (TCP) or
   the Simple Network Management Protocol (SNMP).

3. RAQMON Operation in Congestion-Safe Mode

   RAQMON PDUs can be transmitted over multiple transport protocols.
   The RAQMON Framework will be congestion safe, if a RAQMON PDU is
   transported over TCP.

   One solution to the congestion awareness problem could have been to
   discourage the use of UDP entirely for RAQMON. Though RAQMON PDUs
   can be transported over TCP, some transports like SNMP over TCP are
   not commonly practiced in practical deployments.

The use of UDP inherently increases the risks of network congestion
problems, as UDP itself does not define congestion prevention,
avoidance, detection, or correction mechanisms. The fundamental
problem with UDP is that it provides no feedback mechanism to allow a
sender to pace its transmissions against the real performance of the
network. While this tends to have no significant effect on extremely
low-volume sender-receiver pairs, the impact of high-volume
relationships on the network can be severe. This problem could be
further aggravated by large RAQMON PDUs fragmented at the UDP level.
Transport protocols such as DCCP can also be used as underlying
RAQMON PDU transport, which provides flexibility of UDP style
datagram transmission with congestion control.

It should be noted that the congestion problem is not just between
RDS and RRC pairs, but whenever there is a high fan-in ratio,
congestion could occur (e.g., many RDSs reporting to an RRC). Within
the RAQMON Framework using UDP as a transport, congestion safety can
be achieved in following ways:

   1. Constant Transmission Rate: In a well-managed network, a
      constant transmission rate policy (e.g., 1 RAQMON PDU per
      device every N seconds) will ensure congestion safety as
      devices are introduced into the network in a controlled manner.
      For example, in an enterprise network, IP Phones are added in a
      controlled manner, and a constant transmission rate policy can
      be sufficient to ensure congestion-safe operation. The
      configured rate needs to be related to the expected peak number
      of devices. As a worst-case scenario, if the RDSs enforce an
      administrative policy where the maximum PDU transmission rate
      is no more than one RAQMON PDU every two minutes, a UDP-based
      implementation can be as congestion safe as a TCP-based
      implementation. Such policies can be enforced while
      configuring RDSs, and the timers for the constant rate need to
      be randomly jittered.

   2. Single outstanding requests: This approach requires that a
      request be sent at the application level, then there is a wait
      for some sort of response indicating that the request was
      received before sending anything else. This produces an effect
      described by some as "ping-ponging": traffic bounces back and
      forth between two nodes like a ping-pong ball in a match.
      Since there's only one ball in play between any two players at
      any given time, most of the potential for congestion cascades
      is eliminated. For reliability and efficiency reasons, this
      technique must include backed-off retransmissions. For
      example, if RAQMON PDUs are transported using SNMP INFORM PDUs
      over UDP, a SNMP response from the RRC SHOULD be processed by
      the RDS to implement this mechanism. [RFC4712] specifies that

if the SNMP notifications transport mapping mechanism is
implemented, it is RECOMMENDED to use INFORM PDUs, and it is
NOT RECOMMENDED to use Trap PDUs.

This pacing or serialization approach has the side-effect of
significantly reducing the maximum throughput, as transmission
occurs in only one direction at a time and there is at least a
2xRTT (round-trip time) delay between transmissions. More
sophisticated algorithms (such as those in TCP and Stream
Control Transmission Protocol (SCTP)) have been developed to
address this, and it would be inappropriate to duplicate that
work at the application level. Consequently, if greater
efficiency is required than that provided by this simple
approach, implementers SHOULD use TCP, SCTP, or another such
protocol. But if one absolutely must use UDP, this approach
works. It has been also used in other application scenarios
like SIP over UDP.

3. By restricting transmission to a maximum transmission unit
   (MTU) size: An RDS may be faced with a request to deliver a
   large message using UDP as a transport. Fragmentation of such
   messages is problematic in several ways. Loss of any fragment
   requires time-out and retransmission of the message. The
   fragments are commonly transmitted out of the interface at
   local interface (usually LAN) rates, without awareness of the
   intervening network conditions. For these reasons, it is
   generally considered a bad practice to send large PDUs over
   UDP. If the MTU size is known, as an implementation, an RDS
   should not allow an application to send more information by
   limiting the size of transmissions over UDP to reduce the
   effects of fragmentation. As an alternate, an RDS MAY also
   send parameters to RRC over multiple RAQMON PDUs but identify
   them as part of the same RAQMON reporting session with exactly
   the same Network Time Protocol (NTP) [RFC1305] time stamp.

   While the actual MTU of a link may not be known, common
   practice seems to indicate that the RDS local interface MTU is
   likely to be a reasonable "approximation". Where the actual
   path MTU is known, that value SHOULD be used instead.

4. Irrespective of choice of transport protocol, it is also
   RECOMMENDED that no more than 10% network bandwidth be used for
   RDS/RRC reporting. More frequent reports from an RDS to RRC
   would imply requirements for higher network bandwidth usage.

4. Measurement Methodology

   It is not the intent of this document to recommend a methodology to
   measure any of the QoS parameters defined in Section 5. Measurement
   algorithms are left to the implementers and equipment vendors to
   choose. There are many different measurement methodologies available
   for measuring application performance. These include probe-based,
   client-based, synthetic-transaction, and other approaches. This
   specification does not mandate a particular methodology and is open
   to any methodology that meets the minimum requirements. For
   conformance to this specification, it is REQUIRED that the collected
   data match the semantics described herein. However, it is
   RECOMMENDED that vendors use IETF-defined and International
   Telecommunication Union (ITU)-specified methodologies to measure
   parameters when possible.

5. Metrics Pre-Defined for the BASIC Part of the RAQMON PDU

   The BASIC part of the RAQMON PDU provides for a list of pre-defined
   parameters frequently used by applications to characterize end-to-end
   application Quality of Service. This section defines a set of simple
   metrics to be contained in the BASIC part of the RAQMON PDU, through
   reference to existing IETF, ITU, and other standards organizations'
   documents. Appropriate IETF or ITU references are included in the
   metrics definitions.

   As mentioned earlier, the RAQMON PDU also contains an application-
   specific part, where application- and vendor-specific information not
   included in BASIC part can be added as <Name, Value> pairs, or as a
   variable binding list. These extensions, managed independently by
   vendors or other organizations, should be published for wider
   interoperability.

   Applications are not required to report all the parameters mentioned
   in this section, but should have the flexibility to report a subset
   of these parameters appropriate to an application context. The memo
   further identifies the parameters that RDSs are required to include
   in all PDUs for compliance, as well as optional parameters that RDSs
   may report as needed. The definitions presented here are meant to
   provide guidance to implementers, and IETF metric definition
   references are provided for each metric. Application developers
   should choose the metrics appropriate to their applications' needs.
   Syntactical representations of the parameters identified here are
   provided in the [RFC4712] specification.

5.1. Data Source Address (DA)

   The Data Source Address (DA) is the address of the data source. This
   could be either a globally unique IPv4 or IPv6 address, or a
   privately IPv4 allocated address as defined in [RFC1918].

   It is expected that the DA would remain constant within a given
   communication session. RDSs SHOULD avoid sending these parameters
   within RAQMON reports too often to ensure an efficient usage of
   network resources.

5.2. Receiver Address (RA)

   The Receiver Address (RA) takes the same form as the Data Source
   Address (DA) but represents the Receiver's Address. In a
   communication session, the reporting RDSs SHOULD fill in the other
   party's address as a Receiver Address. Like the Data Source Address,
   this could be either a globally unique IPv4 or IPv6 address, or a
   privately allocated IPv4 address as defined in [RFC1918].

   It is expected that the Receiver Address (RA) would remain constant
   within a given communication session. RDSs SHOULD avoid sending
   these parameters within RAQMON reports too often in order to ensure
   an efficient usage of network resources.

5.3. Data Source Name (DN)

   The Data Source Name (DN) item could be of various formats as needed
   by the application. Forms the DN could take include, but are not
   restricted to:

      - "user@host", or "host" if a user name is not available as on
        single-user systems. For both of these formats, "host" is the
        fully qualified domain name of the host from which the payload
        originates, formatted according to the rules specified in
        [RFC1034], [RFC1035], and Section 2.1 of [RFC1123]. Use example
        names are "big-guy@example.com" or "big-guy@192.0.2.178" for a
        multi-user system. On a system with no user name, an example
        would be "ip-phone4630.example.com". It is RECOMMENDED that the
        standard host's numeric address not be reported via the DN
        parameter, as the DA parameter is used for that purpose.

      - Another instance of a DN could be a valid E.164 phone number, a
        SIP URI, or any other form of telephone or pager number. The
        phone number SHOULD be formatted with a plus sign replacing the
        international access code. Example: "+44-116-496-0348" for a
        number in the UK.

The DN value is expected to remain constant for the duration of a
session. RDSs SHOULD avoid sending these parameters within RAQMON
reports too often in order to ensure an efficient usage of network
resources.

5.4. Receiver Name (RN)

The Receiver Name (RN) takes the same form as DN, but represents the
Receiver's name. In a communication session, an application SHOULD
supply as an RN the name of the other party with which it is
communicating.

The RN value is expected to remain constant for the duration of a
session. RDSs SHOULD avoid sending these parameters within RAQMON
reports too often in order to ensure an efficient usage of network
resources.

5.5. Data Source Device Port Used

This parameter indicates the source port used by the application for
a particular session or sub-session in communication. Examples of
ports include TCP Ports or UDP Ports, as used by communication
application protocols such as Session Initiation Protocol (SIP), SIP
for Instant Messaging and Presence Leveraging Extensions (SIMPLE),
H.323, RTP, HyperText Transport Protocol (HTTP), and so on.

This parameter MUST be sent in the first RAQMON PDU.

5.6. Receiver Device Port Used

This parameter indicates the receiver port used by the application
for a particular session or sub-session. Examples of ports include
TCP Ports, or UDP Ports used by communication application protocols
such as SIP, SIMPLE, H.323, RTP, HTTP, etc.

This parameter MUST be sent in the first RAQMON PDU.

5.7. Session Setup Date/Time

This parameter gives the time when the setup was initiated, if the
application has a setup phase, or when the session was started, if
such a setup phase does not exist. The time is represented using the
timestamp format of the Network Time Protocol (NTP), which is in
seconds relative to 0h UTC (Coordinated Universal Time) on 1 January
1900 [RFC1305].

This parameter SHOULD be sent only in the first RAQMON PDU, after the
session setup is completed.

5.8. Session Setup Delay

   The Session Setup Delay metric reports the time taken from an
   origination request being initiated by a host/endpoint to the media
   path being established (or a session progress indication being
   received from the remote host/endpoint), expressed in milliseconds.
   For example, in VoIP systems, a session setup time can be measured as
   the interval from the last DTMF (dual-tone multi-frequency) button
   pushed to the first ring-back tone that indicates that the far end is
   ringing. Another example would be the Session Setup Delay of a SIP
   call, which is measured as the elapsed time between when an INVITE is
   generated by a User Agent and when the 200 OK is received.

   This parameter SHOULD be sent only in the first RAQMON PDU, after the
   session setup is completed.

5.9. Session Duration

   The Session Duration metric reports how long a session or a sub-
   session lasted. This metric is application context sensitive. For
   example, a VoIP Call Session Duration can be measured as the elapsed
   time between call pickup and call termination, including session
   setup time.

   This parameter SHOULD be sent only in the first RAQMON PDU, after the
   session is terminated.

5.10. Session Setup Status

   The Session Setup Status metric is intended to report the
   communication status of a session. Its values identify appropriate
   communication session states, such as Call Progressing, Call
   Established successfully, "trying", "ringing", "re-trying", "RSVP
   reservation failed", and so on.

   Session setup status is meaningful in the context of applications.
   For this reason, applications SHOULD use this metric together with
   the application/name metrics defined in Section 5.32.

   This information could be used by network management systems to
   calculate parameters such as call success rate, call failure rate,
   etc., or by a debugging tool that captures the status of a call's
   setup phase as soon as a call is established.

   This parameter SHOULD be sent after each change in the session
   status.

5.11. Round-Trip End-to-End Network Delay

   The Round-Trip End-to-End Network Delay, defined in [RFC3550] for
   applications running over RTP and in [RFC2681] for all other IP
   applications, is a key metric for Application QoS Monitoring. Some
   applications do not perform well (or at all) if the end-to-end delay
   between hosts is large relative to some threshold value. Erratic
   variation in delay values makes it difficult (or impossible) to
   support many real-time applications such as Voice over IP, Video over
   IP, Fax over IP etc.

   The Round-Trip End-to-End Network delay of the underlying transport
   network is measured using methodologies described in [RFC3550] for
   RTP and in [RFC2681] for other IP applications.

   Note that the packets used for measurement in some methodologies may
   be of a different type from those used for media (e.g., ICMP instead
   of RTP) and hence may differ in terms of route and queue priority.
   This may result in measured delays being different from those
   experienced on the media path. Conformance for this metric requires
   that actual application packets, or packets of the same application
   type, be used.

   Support for RTP can be determined by the support of the RTP MIB
   [RFC2959] in the hosts running the applications or by inclusion of
   the string 'RTP' at the beginning of the Application Name (Section
   5.32).

   This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
   capability of determining its value and if the parameter is relevant
   for the application.

5.12. One-Way End-to-End Network Delay

   The One-Way End-to-End Network Delay [RFC2679] metric reports the
   One-Way End-to-End delay encountered by traffic from the source to
   the destination network interface. One-Way Delay measurements
   identified by the IP Performance Metrics (IPPM) Working Group
   [RFC2679] will be used to measure one-way end-to-end network delay.

   The need for such a metric is derived from the fact that the path
   from a source to a destination may be different from the path from
   the destination back to the source ("asymmetric paths"), such that
   different sequences of routers are used for the forward and reverse
   paths. Therefore, round-trip measurements actually measure the
   performance of two distinct paths together.

Measuring each path independently highlights the performance
difference between the two paths that may traverse different Internet
service providers, and even radically different types of networks
(for example, research versus commodity networks, or ATM
(Asynchronous Transfer Mode) versus Packet-over-SONET (Synchronous
Optical) transport networks).

Even when the two paths are symmetric, they may have radically
different performance characteristics due to asymmetric queuing.
Performance of an application may depend mostly on the performance in
one direction. For example, a file transfer using TCP may depend
more on the performance in the direction that data flows than on the
direction in which acknowledgements travel.

In QoS-enabled networks, provisioning in one direction may be
radically different from provisioning in the reverse direction, and
thus the QoS guarantees differ. Measuring the paths independently
allows the verification of both guarantees.

RAQMON SHOULD NOT derive One-Way End-to-End Network Delay by assuming
Internet paths are symmetric (i.e., dividing Round-Trip Delay by
two).

Note that the packets used for measurement in some methodologies may
be of a different type from those used for media (e.g., ICMP instead
of RTP) and hence may differ in terms of route and queue priority.
This may result in measured delays being different from those
experienced on the media path. Conformance for this metric requires
that actual application packets, or packets of the same application
type, be used.

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.13. Application Delay

Various Network Delay versions, as outlined in Sections 5.11 and
5.12, do not include delays associated with buffering, play-out,
packet-sequencing, coding/decoding, etc., in the end-devices. The
Application Delay metric defined in this section is targeted to
capture all such delay parameters, providing a total application
endpoint delay.

Application delay can be expressed as the time delay introduced
between the network interface and the application-level presentation.
Since it is difficult to envision usage of all sorts of applications,

the following guidance is provided to the implementers to measure the
application delay:

   - The sending end contribution to application delay is defined as the
     sum of sample sequencing, accumulation, and encoding delay.

   - The receiving end contribution to application delay is calculated
     as the sum of delays associated with buffering, play-out, packet-
     sequencing, and decoding associated with the receiving direction,
     if relevant.

   The endpoint application delay is defined as the sum of the receiving
   and sending contributions to delay measured or estimated within the
   endpoint that is generating this report.

   It is easy to recognize that applications running on an IP device can
   experience same network delay but have different application-
   associated delay values. As such, the user experience associated
   with specific applications may vary while the network delay value
   remains same for both the applications.

   Having network delay and application delay measurements available, a
   management application can represent the delay experienced by the end
   user at the application level as a sum of network delay and the
   application delays reported from the endpoints. However, the
   specification of such a management application is outside the scope
   of the RAQMON specification.

   This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
   capability of determining its value and if the parameter is relevant
   for the application.

5.14. Inter-Arrival Jitter

   The Inter-Arrival Jitter metric provides a short-term measure of
   network congestion [RFC3550]. The jitter measure may indicate
   congestion before it leads to packet loss. The inter-arrival jitter
   field is only a snapshot of the jitter at the time when a RAQMON PDU
   is generated and is not intended to be taken quantitatively as
   indicated in [RFC3550]. Rather, it is intended for comparison of
   inter-arrival jitter from one receiver over time. Such inter-arrival
   jitter information is extremely useful to understand the behavior of
   certain applications such as Voice over IP, Video over IP, etc.
   Inter-arrival jitter information is also used in the sizing of play-
   out buffers for applications requiring the regular delivery of
   packets (for example, voice or video play-out).

In [RFC3550], the selection function is implicitly applied to
consecutive packet pairs, and the "jitter estimate" is computed by
applying an exponential filter with parameter 1/16 to generate the
estimate (i.e., j_new = 15/16* j_old + 1/16*j_new).

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.15. IP Packet Delay Variation

[RFC3393] provides guidance to several absolute jitter parameters.
RAQMON uses the [RFC3393] definition of the IP Packet Delay Variation
(ipdv) for packets inside a stream of packets. The IP Delay
Variation metric is used to determine the dynamics of queues within a
network (or router) where the changes in delay variation can be
linked to changes in the queue length processes at a given link or a
combination of links. Such a parameter provides visibility within an
IP Network and a better understanding of application-level
performance problems as it relates to IP Network performance.

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.16. Total Number of Application Packets Received

This metric reports the number of application payload packets
received by the RDS as part of this session since the last RAQMON PDU
was sent up until the time this RAQMON PDU was generated.

This parameter represents a very simple incremental counter that
counts the number of "application" packets that an RDS has received.
Application packets MAY include signaling packets. Since this count
is a snapshot in time, depending on application type, it also varies
based on the application states, e.g., an RDS within an application
session will report the aggregated number of application packets that
were sent out during signaling setup, media packets received, session
termination, etc.

For example, during Voice over IP or Video over IP sessions setup,
this counter represents the number of signaling-session-related
packets that have been received that will be derived from the
relevant application signaling protocol stack such as SIP or H.323,
SIMPLE, and various other signaling protocols used by the application
to establish the communication session.

However, during a period when media is established between the
communicating entities, this counter will be indicative of the number
of RTP Frames that have been sent out to the communicating party
since last PDU was sent out. The methodology described within RTCP
SR/RR reports [RFC3550] to count RTP frames will be applied wherever
applications use RTP. This being a cumulative counter, applications
need to take into consideration the possibility of the counter
overflowing and restarting counting from zero.

Support for RTP can be determined by the support of the RTP MIB
[RFC2959] in the hosts running the applications or by inclusion of
the string 'RTP' at the beginning of the Application Name (Section
5.32).

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.17. Total Number of Application Packets Sent

This metric reports the number of signaling and payload packets sent
by the RDS as part of this session since the last RAQMON PDU was sent
until the time this RAQMON PDU was generated. Applications packets
MAY include signaling packets. Similar to the total number of
application packets received parameter in Section 5.16, this count is
a snapshot in time. Depending on the application type, the counter
also varies based on various application states, including packet
counts for signaling setup, media establishment, session termination
states, and so on. This being a cumulative counter, applications
need to take into consideration the possibility of the counter
overflowing and restarting counting from zero.

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.18. Total Number of Application Octets Received

This metric reports the total number of signaling and payload octets
received in packets by the RDS as part of this session since the last
RAQMON PDU was sent, up until the time this RAQMON packet was
generated. Applications octets MAY include signaling octets. The
methodology described by [RFC3550] will be applied wherever
applications use RTP. This being a cumulative counter, applications
need to take into consideration the possibility of the counter
overflowing and restarting counting from zero.

   Support for RTP can be determined by the support of the RTP MIB
   [RFC2959] in the hosts running the applications or by inclusion of
   the string 'RTP' at the beginning of the Application Name (Section
   5.32).

   This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
   capability of determining its value and if the parameter is relevant
   for the application.

5.19. Total Number of Application Octets Sent

   This metric reports the total number of signaling and payload octets
   received in packets by the RDS as part of this session since the last
   RAQMON PDU was sent, up until the time this RAQMON packet was
   generated. This is similar to the Total Number of Application Octets
   Received metric. Applications octets MAY include signaling octets.
   The methodology described by [RFC3550] will be applied wherever
   applications use RTP. This being a cumulative counter, applications
   need to take into consideration the possibility of the counter
   overflowing and restarting counting from zero.

   Support for RTP can be determined by the support of the RTP MIB
   [RFC2959] in the hosts running the applications or by inclusion of
   the string 'RTP' at the beginning of the Application Name (Section
   5.32).

   This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
   capability of determining its value and if the parameter is relevant
   for the application.

5.20. Cumulative Packet Loss

   The cumulative packet loss metric indicates the loss associated with
   the network as well as local device losses over time. This parameter
   is counted as the total number of application packets from the source
   that have been lost since the beginning of the session. This number
   is defined to be the number of packets expected less the number of
   packets actually received, where the number of packets received
   includes the count of packets that are late or duplicates. If a
   packet is discarded due to late arrival, then it MUST be counted as
   either lost or discarded but MUST NOT be counted as both.

   Packet loss by the underlying transport network SHALL be measured
   using the methodologies described in [RFC3550] for RTP traffic and
   [RFC2680] for other IP traffic. The number of packets expected is
   defined to be the extended last sequence number received, as defined

next, less the initial sequence number received. For RTP traffic,
this may be calculated using techniques such as those shown in
Appendix A.3 of [RFC3550].

Packet loss by the underlying transport network SHALL be measured
using the methodologies described in [RFC3550] for RTP traffic and
[RFC2680] for other IP traffic. The number of packets expected is
defined to be the extended last sequence number received, as defined
next, less the initial sequence number received. For RTP traffic,
this may be calculated using techniques such as those shown in
Appendix A.3 of [RFC3550].

Support for RTP can be determined by the support of the RTP MIB
[RFC2959] in the hosts running the applications or by inclusion of
the string 'RTP' at the beginning of the Application Name (Section
5.32).

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.21. Packet Loss in Fraction

The Packet Loss in Fraction metric represents the packet loss as
defined above, but expressed as a fraction of the total traffic over
time.

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.22. Cumulative Application Packet Discards

The RAQMON Framework allows applications to distinguish between
packets lost by the network and those discarded due to jitter and
other application-level errors. Though packet loss and discards have
an equal effect on the quality of the application, having separate
counts for packet loss and discards helps identify the source of
quality degradation.

The packet discard metric indicates packets discarded locally by the
device over time. Local device-level packet discard is captured as
the total number of application-level packets from the source that
have been discarded since the beginning of reception, due to late or
early arrival, under-run or overflow at the receiving jitter buffer,
or any other application-specific reasons.

If the RDS cannot tell the difference between discards and lost
packets, then it MUST report only lost packets and MUST NOT report
discards.

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.23. Packet Discards in Fraction

The packet discards in fraction metric represents packets from the
source that have been discarded since the beginning of the reception
but expressed as a fraction of the total traffic over time.

This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
capability of determining its value and if the parameter is relevant
for the application.

5.24. Source Payload Type

The source payload type reports payload formats (e.g., media
encoding) as sent by the data source, e.g., ITU G.711, ITU G.729B,
H.263, MPEG-2, ASCII, etc. This memo follows the definition of
Payload Type (PT) in [RFC3551]. For example, to indicate that the
source payload type used for a session is PCMA (pulse-code modulation
with A-law scaling), the value of the source payload field for the
respective session will be 8.

The source payload type value is expected to remain constant for the
duration of a session, with the exception of events like dynamic
codec changes. RDSs SHOULD avoid sending these parameters within
RAQMON reports more often than necessary (e.g., at dynamic codec
changes) to ensure an efficient usage of network resources.

If dynamic types (values 96 to 127, according to [RFC3551]) are being
used to identify the source payload type, a RAQMON extension
parameter MAY be defined to indicate the MIME subtypes. In the case
where the RDS does send reports noting dynamic codec changes, there
may be instances where this extension parameter is used only before
or after the codec change, as the source payload may shift between
the dynamic and static types.

5.25. Receiver Payload Type

The receiver payload type reports payload formats (e.g., media
encodings) as sent by the other communicating party back to the
source, e.g., ITU G.711, ITU G.729B, H.263, MPEG-2, ASCII, etc. This
document follows the definition of payload type (PT) in [RFC3551].

For example, to indicate that the destination payload type used for a session is PCMA, the destination payload type field for the respective session will be 8.

The destination payload type value is expected to remain constant for the duration of a session, with the exception of events like dynamic codec changes. RDSs SHOULD avoid sending these parameters within RAQMON reports more often than necessary (e.g., at dynamic codec changes) to ensure an efficient usage of network resources.

If dynamic types (values 96 to 127, according to [RFC3551]) are being used to identify the destination payload type, a RAQMON extension parameter MAY be defined to indicate the MIME subtypes. In the case where the RDS does send reports noting dynamic codec changes, there may be instances where this extension parameter is used only before or after the codec change, as the destination payload may shift between the dynamic and static types.

5.26. Source Layer 2 Priority

Many devices use Layer 2 technologies to prioritize certain types of traffic in the Local Area Network environment. For example, the 1998 Edition of IEEE 802.1D [IEEE802.1D], "Media Access Control Bridges", contains expedited traffic capabilities to support transmission of time-critical information. Many devices use that standard to mark Ethernet frames according to IEEE P802.1p standard. Details on these can be found in [IEEE802.1D], which incorporates P802.1p. The Source Layer 2 Priority RAQMON field indicates what Layer 2 values were used by the host running the RDS to prioritize these packets in the Local Area Network environment.

The Source Layer 2 Priority value is expected to remain constant for the duration of a session. Hosts running the RDSs SHOULD avoid sending these parameters within RAQMON reports too often in order to ensure an efficient usage of network resources.

5.27. Source TOS/DSCP Value

Various Layer 3 technologies are in place to prioritize traffic in the Internet. For example, the traditional IP Precedence [RFC791] and Type of Service (TOS) [RFC1812], or more recent technologies like Differentiated Services [RFC2474] [RFC2475], use the TOS octet in IPv4, whereas the traffic class octet is used to prioritize traffic in IPv6. Source Layer TOS/DCP RAQMON field reports the appropriate Layer 3 values used by the Data Source to prioritize these packets.

   The Source TOS/DSCP value is expected to remain constant for the
   duration of a session. Hosts running the RDSs SHOULD avoid sending
   these parameters within RAQMON reports too often in order to ensure
   an efficient usage of network resources.

5.28. Destination Layer 2 Priority

   The Destination Layer 2 Priority reports the Layer 2 value used by
   the communication receiver to prioritize packets while sending
   traffic to the data source in the Local Area Networks environment.
   Like Source Layer 2 Priority, Destination Layer 2 Priority could
   indicate whether the destination has used Layer 2 technologies like
   IEEE P802.1p for priority queuing.

   The Destination Layer 2 Priority value is expected to remain constant
   for the duration of a session. Hosts running the RDSs SHOULD avoid
   sending these parameters within RAQMON reports too often in order to
   ensure an efficient usage of network resources.

5.29. Destination TOS/DSCP Value

   The Destination TOS/DSCP RAQMON field reports the values used by the
   Data Receiver to prioritize these packets received by the source.
   Similar to Source Layer 3 Priority, Destination Layer 3 Priority
   indicates whether the destination has used any Layer 3 technologies
   like IP Precedence [RFC791] and Type of Service (TOS) [RFC1812], or
   more recent technologies like Differentiated Service [RFC2474]
   [RFC2475].

   The Destination TOS/DSCP value is expected to remain constant for the
   duration of a session. Hosts running the RDSs SHOULD avoid sending
   these parameters within RAQMON reports too often in order to ensure
   an efficient usage of network resources.

5.30. CPU Utilization in Fraction

   This parameter captures the CPU usage of the hosts running the RDSs
   that may have very critical implications for QoS of an end-device.
   It is computed as an average since the last reporting interval, and
   corresponds to the percentage of that time that the CPU was busy.

   In the case of multiple CPU hosts, the maximum utilization among the
   different CPUs MUST be reported.

   This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
   capability of determining its value and if the parameter is relevant
   for the application.

5.31. Memory Utilization in Fraction

   This parameter captures the memory usage of the hosts running the
   RDSs that may have very critical implications for QoS of an end-
   device. It is computed as an average since the last reporting
   interval and corresponds to the average percentage of the total
   memory space critical for the applications in use during that time
   interval (e.g., primary CPU RAM, buffers).

   In the case of multiple CPU hosts, the maximum memory utilization
   among the different CPUs MUST be reported.

   This parameter SHOULD be sent in each RAQMON PDU, if the RDS has the
   capability of determining its value and if the parameter is relevant
   for the application.

5.32. Application Name/Version

   The Application Name/Version parameter gives the name and,
   optionally, the version of the application associated with that
   session or sub-session, e.g., "XYZ VoIP Agent 1.2". This information
   may be useful for scenarios where the end-device is running multiple
   applications with various priorities and could be very handy for
   debugging purposes.

   If the application is using RTP [RFC3550], the Application Name
   SHOULD begin with the string 'RTP'.

   This parameter MUST be sent in the first RAQMON PDU.

6. Report Aggregation and Statistical Data processing

   Within the RAQMON Framework, RRCs are expected to have significantly
   greater computational resources than RDSs. Consequently, various
   aggregation functions are performed by the RRCs, while RDSs are not
   burdened by statistical data processing such as computation of
   minima, maxima, averages, standard deviations, etc.

   The RAQMON MIB provides minimal aggregation of the RAQMON parameters
   defined above. The RAQMON MIB is not designed to provide extensive
   aggregation like the Application Performance Measurement (APM) MIB
   [RFC3729] or the Transport Performance Metrics (TPM) MIB [RFC4150].
   One should use APM and TPM MIBs to aggregate parameters based on
   protocols (e.g., performance of HTTP, RTP) or applications (e.g.,
   performance of VoIP, Video Applications).

In the RAQMON MIB, aggregation can be performed only on specific
RAQMON metric parameters. Aggregation always results in statistical
Mean/Min/Max values, according to these definitions:

Mean: Mean is defined as the statistical average of a metric over
      the duration of a communication session. For example, if an
      RDS reported End-to-End delay metric N times within a
      communication session, then the Mean End-to-End Delay can be
      computed by summing of these N reported values, and then
      dividing by N.

Min: Min is defined as the statistical minimum of a metric over
      the duration of a communication session. For example, if
      the end-to-end delay metric of an end-device within a
      communication session is reported N times by the RDS, then
      the Min end-to-end delay is the smallest of the N end-to-end
      delay metric values reported.

Max: Max is defined as the statistical maximum of a metric over
      the duration of a communication session. For example, if
      the end-to-end delay metric of an end-device within a
      communication session is reported N times by the RDS, then
      the Max End-to-End Delay is the largest of the N End-to-End
      Delay metric values reported.

7. Keeping Historical Data and Storage

It is evident from the document that the RAQMON MIB data need to be
managed to optimize storage space. The large volume of data gathered
in a communication session could be optimized for storage space by
performing and storing only aggregated RAQMON metrics for history if
required.

Examples of how such storage space optimization can be performed
include:

1. Make data available through the MIB only at the end of a
   communication session, i.e., upon receipt of a NULL PDU. The
   aggregated data could be made available using the RAQMON MIB as
   Mean, Max, or Min entries and saved for historical purposes.

2. Use a time-based algorithm that aggregates data over a specific
   period of time within a communication session, thus requiring
   fewer entries, to reduce storage space requirements. For
   example, if an RDS sends data out every 10 seconds and the RRC
   updates the RAQMON MIB once every minute, for every 6 data
   points there would be one MIB entry.

3. Periodically delete historical data in accordance with an
   administrative policy. An example of such a policy would be to
   delete historical data older than 60 days. The implementation
   of such policies is left to the application developer's
   discretion, and their use is an operational concern.

8. Security Considerations

   Security considerations associated with the RAQMON Framework are
   discussed below, and in greater detail in other RAQMON memos as is
   appropriate.

8.1. The RAQMON Threat Model

   The vulnerabilities associated with the RAQMON Framework are a
   combination of those associated with the underlying layers up to the
   transport layer, and of possible exploits of RAQMON payload.
   Possible exploits of RAQMON payloads fall within these classes:

   1. Unauthorized examination of sensitive information in the
      payload in transit.

   2. Unauthorized modification of payload contents in transit,
      leading to:

      a. Mis-identification of information from one RAQMON reporting
         session as belonging to another destined to the same RRC;

      b. Mismapping of RAQMON sessions;

      c. Various forms of session-level denial-of-service (DoS)
         attacks;

      d. DoS through modification of RAQMON parameter values and
         statistics;

      e. Invalid timestamps, leading to false interpretation of the
         monitored data, affecting call records information, and
         making difficult to place monitoring events in their
         appropriate temporal context.

   3. Malformed payloads, permitting the exploitation of potential
      implementation weaknesses to compromise an RRC.

   4. Unauthorized disclosure of sensitive data carried by
      application PDUs, leading to a breach of confidentiality.

Consequently, threats based on unauthorized disclosure or
modification of payloads or headers will have to be assumed.

8.2. The RAQMON Security Requirements and Assumptions

In order to preserve integrity of the RAQMON PDU against these
threats, the RAQMON model must provide for cryptographically strong
security services.

Consequently, the RAQMON framework must be able to provide for the
following protections:

   1. Authentication - the RRC should be able to verify that a RAQMON
      PDU was in fact originated by the RDS that claims to have sent
      it.

   2. Privacy - Since RAQMON information includes identification of
      the parties participating in a communication session, the
      RAQMON framework should be able to provide for protection from
      eavesdropping, to prevent an unauthorized third party from
      gathering potentially sensitive information. This can be
      achieved by using various payload encryption technologies, such
      as Data Encryption Standard (DES), 3-DES, Advanced Encryption
      Standard (AES), etc.

   3. Protection from DoS attacks directed at the RRC - RDSs send
      RAQMON reports as a side effect of an external event (for
      example, a phone call is being received). An attacker can try
      to overwhelm the RRC (or the network) by initiating a large
      number of events (i.e., calls) for the purpose of swamping the
      RRC with too many RAQMON PDUs.

      To prevent DoS attacks against RRC, the RDS will send the first
      report for a session only after the session has been in
      progress for the five-second reporting interval. Sessions
      shorter than that should be stored in the RDS and will be
      reported only after that interval has expired.

8.3. RAQMON Security Model

The RAQMON architecture permits the use of multiple transport
protocols. Most of these support a secure mode of operation. There
are advantages to relying on the security provided at the transport
protocol layer.

   1. Transport-protocol-level security can generally protect the
      payload with end-to-end authentication, confidentiality,
      message integrity, and replay protection services.

   2. A good cryptographic security protocol always has an associated
      key management protocol. Use of transport protocol security
      relies on its key management and does not require development
      of another mechanism.

   3. When transport protocol security is already enabled between the
      RDS and RRC, additional encryption and message authentication
      at the application level is avoided.

   However, there are also shortcomings to be noted in relying on
   transport protocol security.

   1. When session-level isolation of the different RAQMON sessions
      of an RDS-RRC pair is required, it will be necessary to open
      separate transport protocol instances. Such cases, however,
      may be rare.

   2. Since security services are not provided by the RAQMON
      framework, the absence of transport or lower protocol security
      implies the absence of RAQMON security.

9. Acknowledgements

   The authors would like to thank Andy Bierman, Alan Clark, Mahalingam
   Mani, Colin Perkins, Steve Waldbusser, Magnus Westerlund, and Itai
   Zilbershtein for the precious advices and real contributions brought
   to this document. The authors would also like to extend special
   thanks to Randy Presuhn, who reviewed this document for spelling and
   formatting purposes, and who provided a deep review of the technical
   content. We also would like to thank Bert Wijnen for the permanent
   coaching during the evolution of this document and the detailed
   review of its final versions.

10. Normative References

   [RFC791]   Postel, J., "Internet Protocol", STD 5, RFC 791,
                   September 1981.

   [RFC1812]   Baker, F., "Requirements for IP Version 4 Routers", RFC
                   1812, June 1995.

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

   [RFC2474]   Nichols, K., Blake, S., Baker, F., and D. Black,
                   "Definition of the Differentiated Services Field (DS
                   Field) in the IPv4 and IPv6 Headers", RFC 2474, December
                   1998.

   [RFC2475]   Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.,
                   and W. Weiss, "An Architecture for Differentiated
                   Service", RFC 2475, December 1998.

   [RFC2679]   Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way
                   Delay Metric for IPPM", RFC 2679, September 1999.

   [RFC2680]   Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way
                   Packet Loss Metric for IPPM", RFC 2680, September 1999.

   [RFC2681]   Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-
                   trip Delay Metric for IPPM", RFC 2681, September 1999.

   [RFC2819]   Waldbusser, S., "Remote Network Monitoring Management
                   Information Base", STD 59, RFC 2819, May 2000.

   [RFC2959]   Baugher, M., Strahm, B., and I. Suconick, "Real-Time
                   Transport Protocol Management Information Base", RFC
                   2959, October 2000.

   [RFC3393]   Demichelis, C. and P. Chimento, "IP Packet Delay
                   Variation Metric for IP Performance Metrics (IPPM)", RFC
                   3393, November 2002.

   [RFC3416]   Presuhn, R., Ed., "Version 2 of the Protocol Operations
                   for the Simple Network Management Protocol (SNMP)", STD
                   62, RFC 3416, December 2002.

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

   [RFC3551]   Schulzrinne, H. and S. Casner, "RTP Profile for Audio
                and Video Conferences with Minimal Control", STD 65, RFC
                3551, July 2003.

11. Informative References

   [RFC1034]   Mockapetris, P., "Domain names - concepts and
                facilities", STD 13, RFC 1034, November 1987.

   [RFC1035]   Mockapetris, P., "Domain names - implementation and
                specification", STD 13, RFC 1035, November 1987.

   [RFC1123]   Braden, R., "Requirements for Internet Hosts -
                Application and Support", STD 3, RFC 1123, October 1989.

   [RFC1305]   Mills, D., "Network Time Protocol (Version 3)
                Specification, Implementation and Analysis", RFC 1305,
                March 1992.

   [RFC1918]   Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot,
                G., and E. Lear, "Address Allocation for Private
                Internets", BCP 5, RFC 1918, February 1996.

   [RFC2914]   Floyd, S., "Congestion Control Principles", BCP 41, RFC
                2914, September 2000.

   [RFC3235]   Senie, D., "Network Address Translator (NAT)-Friendly
                Application Design Guidelines", RFC 3235, January 2002.

   [RFC3611]   Friedman, T., Caceres, R., and A. Clark, "RTP Control
                Protocol Extended Reports (RTCP XR)", RFC 3611, November
                2003.

   [RFC3729]   Waldbusser, S., "Application Performance Measurement
                MIB", RFC 3729, March 2004.

   [RFC4150]   Dietz, R. and R. Cole, "Transport Performance Metrics
                MIB", RFC 4150, August 2005.

   [RFC4711]   Siddiqui, A., Romascanu, D., and E. Golovinsky, "Real-
                time Application Quality-of-Service Monitoring (RAQMON)
                MIB", RFC 4711, October 2006.

   [RFC4712]   Siddiqui, A., Romascanu, D., Golovinsky, E., Ramhman,
                M., and Y. Kim, "Transport Mappings for Real-time
                Application Quality-of-Service Monitoring (RAQMON)
                Protocol Data Unit (PDU)", RFC 4712, October 2006.

    [IEEE802.1D] Information technology - Telecommunications and
                 information exchange between systems - Local and
                 metropolitan area networks - Common Specification a -
                 Media access control (MAC) bridges:15802-3: 1998
                 (ISO/IEC). Revision. This is a revision of ISO/IEC
                 10038: 1993, 802.1j-1992 and 802.6k-1992. It
                 incorporates P802.11c, P802.1p and P802.12e [ANSI/IEEE
                 Std 802.1D, 1998 Edition]

Authors' Addresses

    Anwar A. Siddiqui
    Avaya Labs
    307 Middletown Lincroft Road
    Lincroft, New Jersey 07738
    USA

    Phone: +1 732 852-3200
    EMail: anwars@avaya.com

    Dan Romascanu
    Avaya
    Atidim Technology Park, Building #3
    Tel Aviv, 61131
    Israel

    Phone: +972-3-645-8414
    EMail: dromasca@avaya.com

    Eugene Golovinsky

    EMail: gene@alertlogic.net