Network Working Group
Request for Comments: 4712
Category: Standards Track

A. Siddiqui
D. Romascanu
Avaya
E. Golovinsky
Alert Logic
M. Rahman
Samsung Information Systems America
Y. Kim
Broadcom

October 2006

Transport Mappings for Real-time Application Quality-of-Service Monitoring (RAQMON) Protocol Data Unit (PDU)

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

Copyright (C) The Internet Society (2006).

This memo specifies two transport mappings of the Real-Time Application Quality-of-Service Monitoring (RAQMON) information model defined in RFC 4710 using TCP as a native transport and the Simple Network Management Protocol (SNMP) to carry the RAQMON information from a RAQMON Data Source (RDS) to a RAQMON Report Collector (RRC).

Table of Contents

1. Introduction

The Real-Time Application QoS Monitoring (RAQMON) Framework, as outlined by [RFC4710], extends the Remote Monitoring family of protocols (RMON) by defining entities such as RAQMON Data Sources RDS) and RAQMON Report Collectors (RRC) to perform various application monitoring in real time. [RFC4710] defines the relevant metrics for RAQMON monitoring carried by the common protocol data unit (PDU) used between a RDS and RRC to report QoS statistics. This memo contains a syntactical description of the RAQMON PDU structure.

The following sections of this memo contain detailed specifications for the usage of TCP and SNMP to carry RAQMON information.

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. Transporting RAQMON Protocol Data Units

The RAQMON Protocol Data Unit (PDU) utilizes a common data format understood by the RDS and the RRC. 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. As part of the specification, this memo also specifies the usage of TCP and SNMP as underlying transport protocols to carry RAQMON PDUs between RDSs and RRCs. While two transport protocol choices have been provided as options to chose from for RDS implementers, RRCs MUST implement the TCP transport and MAY implement the SNMP transport.

2.1. TCP as an RDS/RRC Network Transport Protocol

A transport binding using TCP is included within the RAQMON specification to facilitate reporting from various types of embedded devices that run applications such as Voice over IP, Voice over Wi-Fi, Fax over IP, Video over IP, Instant Messaging (IM), E-mail, software download applications, e-business style transactions, web access from wired or wireless computing devices etc. For many of these devices, PDUs and a TCP-based transport fit the deployment needs.

The RAQMON transport requirements for end-to-end congestion control and reliability are inherently built into TCP as a transport protocol [RFC793].

To use TCP to transport RAQMON PDUs, it is sufficient to send the PDUs as TCP data. As each PDU carries its length, the receiver can determine the PDU boundaries.

The following section details the RAQMON PDU specifications. Though transmitted as one Protocol Data Unit, a RAQMON PDU is functionally divided into two different parts: the BASIC part and application extensions required for vendor-specific extension [RFC4710]. Both functional parts follow a field carrying a SMI Network Management Private Enterprise code currently maintained by IANA http://www.iana.org/assignments/enterprise-numbers, which is used to identify the organization that defined the information carried in the PDU.

A RAQMON PDU in the current version is marked as PDU Type (PDT) = 1. The parameters carried by RAQMON PDUs are shown in Figure 1 and are defined in section 5 of [RFC4710].

Vendors MUST use the BASIC part of the PDU to report parameters pre-listed here in the specification for interoperability, as opposed to using the application-specific portion. Vendors MAY also use application-specific extensions to convey application-, vendor-, or device-specific parameters not included in the BASIC part of the specification and explicitly publish such data externally to attain extended interoperability.

2.1.1. The RAQMON PDU

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PDT = 1  |B|  T  |P|S|R|  RC   |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              DSRC                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  SMI Enterprise Code = 0      |Report Type = 0|     RC_N      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |flag
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Data Source Address {DA}                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Receiver's Address (RA)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                NTP Timestamp, most significant word           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                NTP Timestamp, least significant word          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length       |   Application Name (AN)  ...                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length       |   Data Source Name (DN)  ...                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length       |    Receiver's Name (RN)  ...                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length       |    Session State          ...                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Session Duration                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Round-Trip End-to-End Network Delay              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              One-Way End-to-End Network Delay                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Cumulative Packet Loss                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Cumulative Application Packet Discard            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Total # Application Packets sent                 |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Total # Application Packets received                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Total # Application Octets sent                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Total # Application Octets received               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Data Source Device Port Used |  Receiver Device Port Used    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    S_Layer2   |   S_Layer3    |   S_Layer2    |   S_Layer3    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Source Payload |Receiver       | CPU           | Memory        |
|Type           |Payload Type   | Utilization   | Utilization   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Session Setup Delay        |     Application Delay         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IP Packet Delay Variation     |   Inter arrival Jitter        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Packet Discrd |  Packet loss  |          Padding              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SMI Enterprise Code = "xxx"                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Report Type = "yyy"       | Length of Application Part    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               application/vendor specific extension           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...............                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...............                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...............                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SMI Enterprise Code = "abc"                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Report Type = "zzz"       | Length of Application Part    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               application/vendor specific extension           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...............                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: RAQMON Protocol Data Unit

2.1.2. The BASIC Part of the RAQMON Protocol Data Unit

A RAQMON PDU must contain the following BASIC part fields at all times:

PDU type (PDT): 5 bits - This indicates the type of RAQMON PDU being sent. PDT = 1 is used for the current RAQMON PDU version defined in this document.

basic (B): 1 bit - While set to 1, the basic flag indicates that the PDU has BASIC part of the RAQMON PDU. A value of zero is considered valid and indicates a RAQMON NULL PDU.

trailer (T): 3 bits - Total number of Application-Specific Extensions that follow the BASIC part of RAQMON PDU. A value of zero is considered valid as many times as there is no application-specific information to add to the basic information.

padding (P): 1 bit - If the padding bit is set, the BASIC part of the RAQMON PDU contains some additional padding octets at the end of the BASIC part of the PDU that are not part of the monitoring information. Padding may be needed in some cases, as reporting is based on the intent of a RDS to report certain parameters. Also, some parameters may be reported only once at the beginning of the reporting session, e.g., Data Source Name, Receiver Name, payload type, etc. Actual padding at the end of the BASIC part of the PDU is 0, 8, 16, or 24 bits to make the length of the BASIC part of the PDU a multiple of 32 bits

Source IP version Flag (S): 1 bit - While set to 1, the source IP version flag indicates that the Source IP address contained in the PDU is an IPv6 address.

Receiver IP version Flag (R): 1 bit - While set to 1, the receiver IP version flag indicates that the receiver IP address contained in the PDU is an IPv6 address.

record count (RC): 4 bits - Total number of application records contained in the BASIC part of the PDU. A value of zero is considered valid but useless, with the exception of the case of a NULL PDU indicating the end of a RDS reporting session.

length: 16 bits (unsigned integer) - The length of the BASIC part of the RAQMON PDU in units of 32-bit words minus one; this count includes the header and any padding.

DSRC: 32 bits - Data Source identifier represents a unique RAQMON reporting session descriptor that points to a specific reporting session between RDS and RRC. Uniqueness of DSRC is valid only within a reporting session. DSRC values should be randomly generated using vendor-chosen algorithms for each communication session. It is not sufficient to obtain a DSRC simply by calling random() without carefully initializing the state. One could use an algorithm like the one defined in Appendix A.6 in [RFC3550] to create a DSRC. Depending on the choice of algorithm, there is a finite probability that two DSRCs from two different RDSs may be the same. To further reduce the probability that two RDSs pick the same DSRC for two different reporting sessions, it is recommended that an RRC use parameters like Data Source Address (DA), Data Source Name (DN), and layer 2 Media Access Control (MAC) Address in the PDU in conjunction with a DSRC value. It is not mandatory for RDSs to send parameters like Data Source Address (DA), Data Source Name (DN), and MAC Address in every PDU sent to RRC, but occasionally sending these parameters will reduce the probability of DSRC collision drastically. However, this will cause an additional overhead per PDU.

A value of zero for basic (B) bit and trailer (T) bits constitutes a RAQMON NULL PDU (i.e., nothing to report). RDSs MUST send a RAQMON NULL PDU to RRC to indicate the end of the RDS reporting session. A NULL PDU ends with the DSRC field.

SMI Enterprise Code: 16 bits. A value of SMI Enterprise Code = 0 is used to indicate the RMON-WG-compliant BASIC part of the RAQMON PDU format.

Report Type: 8 bits - These bits are reserved by the IETF RMON Working Group. A value of 0 within SMI Enterprise Code = 0 is used for the version of the PDU defined by this document.

The BASIC part of each RAQMON PDU consists of Record Count Number (RC_N) and RAQMON Parameter Presence Flags (RPPF) to indicate the presence of appropriate RAQMON parameters within a record, as defined in Table 1.

RC_N: 8 bits - The Record Count number indicates a sub-session within a communication session. A value of zero is a valid record number. The maximum number of records that can be described in one RAQMON Packet is 256.

RAQMON Parameter Presence Flags (RPPF): 32 bits

Each of these flags, while set, represents that this RAQMON PDU contains corresponding parameters as specified in Table 1.

Bit Sequence Number	Presence/Absence of corresponding Parameter within this RAQMON PDU
0	Data Source Address (DA)
1	Receiver Address (RA)
2	NTP Timestamp
3	Application Name
4	Data Source Name (DN)
5	Receiver Name (RN)
6	Session Setup Status
7	Session Duration
8	Round-Trip End-to-End Net Delay (RTT)
9	One-Way End-to-End Network Delay (OWD)
10	Cumulative Packets Loss
11	Cumulative Packets Discards
12	Total number of App Packets sent
13	Total number of App Packets received
14	Total number of App Octets sent
15	Total number of App Octets received
16	Data Source Device Port Used
17	Receiver Device Port Used
18	Source Layer 2 Priority
19	Source Layer 3 Priority
20	Destination Layer 2 Priority
21	Destination Layer 3 Priority

22	Source Payload Type
23	Receiver Payload Type
24	CPU Utilization
25	Memory Utilization
26	Session Setup Delay
27	Application Delay
28	IP Packet Delay Variation
29	Inter arrival Jitter
30	Packet Discard (in fraction)
31	Packet Loss (in fraction)

Table 1: RAQMON Parameters and Corresponding RPPF

Data Source Address (DA): 32 bits or 160 bits in binary representation - This parameter is defined in section 5.1 of [RFC4710]. IPv6 addresses are incorporated in Data Source Address by setting the source IP version flag (S bit) of the RAQMON PDU header to 1.

Receiver Address (RA): 32 bits or 160 bits - This parameter is defined in section 5.2 of [RFC4710]. It follows the exact same syntax as Data Source Address but is used to indicate a Receiver Address. IPv6 addresses are incorporated in Receiver Address by setting the receiver IP version flag (R bit) of the RAQMON PDU header to 1.

Session Setup Date/Time (NTP timestamp): 64 bits - This parameter is defined in section 5.7 of [RFC4710] and represented using the timestamp format of the Network Time Protocol (NTP), which is in seconds [RFC1305]. The full resolution NTP timestamp is a 64-bit unsigned fixed-point number with the integer part in the first 32 bits and the fractional part in the last 32 bits.

Application Name: This parameter is defined in section 5.32 of [RFC4710]. The Application Name field starts with an 8-bit octet count describing the length of the text followed by the text itself using UTF-8 encoding. Application Name field is a multiple of 32 bits, and padding will be used if necessary.

A Data Source that does not support NTP SHOULD set the appropriate RAQMON flag to 0 to avoid wasting 64 bits in the PDU. Since the NTP time stamp is intended to provide the setup Date/Time of a session, it is RECOMMENDED that the NTP Timestamp be used only in the first RAQMON PDU after sub-session RC_N setup is completed, in order to use network resources efficiently.

Data Source Name (DN): Defined in section 5.3 of [RFC4710]. The Data Source Name field starts with an 8-bit octet count describing the length of the text followed by the text itself. Padding is used to ensure that the length and text encoding occupy a multiple of 32 bits in the DN field of the PDU. The text MUST NOT be longer than 255 octets. The text is encoded according to the UTF-8 encoding specified in [RFC3629]. Applications SHOULD instruct RDSs to send out the Data Source Name infrequently to ensure efficient usage of network resources as this parameter is expected to remain constant for the duration of the reporting session.

Receiver Name (RN): This metric is defined in section 5.4 of [RFC4710]. Like Data Source Name, the Receiver Name field starts with an 8-bit octet count describing the length of the text, followed by the text itself. The Receiver Name, including the length field encoding, is a multiple of 32 bits and follows the same padding rules as applied to the Data Source Name. Since the Receiver Name is expected to remain constant during the entire reporting session, this information SHOULD be sent out occasionally over random time intervals to maximize success of reaching a RRC and also conserve network bandwidth.

Session Setup Status: The Session (sub-session) Setup Status is defined in section 5.10 of [RFC4710]. This field starts with an 8-bit length field followed by the text itself. Session Setup Status is a multiple of 32 bits.

Session Duration: 32 bits - The Session (sub-session) Duration metric is defined in section 5.9 of [RFC4710]. Session Duration is an unsigned integer expressed in seconds.

Round-Trip End-to-End Network Delay: 32 bits - The Round-Trip End-to-End Network Delay is defined in section 5.11 of [RFC4710]. This field represents the Round-Trip End-to-End Delay of sub-session RC_N, which is an unsigned integer expressed in milliseconds.

One-Way End-to-End Network Delay: 32 bits - The One-Way End-to-End Network Delay is defined in section 5.12 of [RFC4710]. This field represents the One-Way End-to-End Delay of sub-session RC_N, which is an unsigned integer expressed in milliseconds.

Cumulative Application Packet Loss: 32 bits - This parameter is defined in section 5.20 of [RFC4710] as an unsigned integer, representing the total number of packets from sub-session RC_N that have been lost while this RAQMON PDU was generated.

Cumulative Application Packet Discards: 32 bits - This parameter is defined in section 5.22 of [RFC4710] as an unsigned integer representing the total number of packets from sub-session RC_N that have been discarded while this RAQMON PDU was generated.

Total number of Application Packets sent: 32 bits - This parameter is defined in section 5.17 of [RFC4710] as an unsigned integer, representing the total number of packets transmitted within sub-session RC_N by the sender.

Total number of Application Packets received: 32 bits - This parameter is defined in section 5.16 of [RFC4710] and is represented as an unsigned integer representing the total number of packets transmitted within sub-session RC_N by the receiver.

Total number of Application Octets sent: 32 bits - This parameter is defined in section 5.19 of [RFC4710] as an unsigned integer, representing the total number of payload octets (i.e., not including header or padding) transmitted in packets by the sender within sub-session RC_N.

Total number of Application Octets received: 32 bits - This parameter is defined in section 5.18 of [RFC4710] as an unsigned integer representing the total number of payload octets (i.e., not including header or padding) transmitted in packets by the receiver within sub-session RC_N.

Data Source Device Port Used: 16 bits - This parameter is defined in section 5.5 of [RFC4710] and describes the port number used by the Data Source as used by the application in RC_N session while this RAQMON PDU was generated.

Receiver Device Port Used: 16 bits - This parameter is defined in section 5.6 of [RFC4710] and describes the receiver port used by the application to communicate to the receiver. It follows same syntax as Source Device Port Used.

S_Layer2: 8 bits - This parameter, defined in section 5.26 of [RFC4710], is associated to the source's IEEE 802.1D [IEEE802.1D] priority tagging of traffic in the communication sub-session RC_N. Since IEEE 802.1 priority tags are 3 bits long, the first 3 bits of this parameter represent the IEEE 802.1 tag value, and the last 5 bits are padded to 0.

S_Layer3: 8 bits - This parameter, defined in section 5.27 of [RFC4710], represents the layer 3 QoS marking used to send packets to the receiver by this data source during sub-session RC_N.

D_Layer2: 8 bits - This parameter, defined in section 5.28 of [RFC4710], represents layer 2 IEEE 802.1D priority tags used by the receiver to send packets to the data source during sub-session RC_N session if the Data Source can learn such information. Since IEEE 802.1 priority tags are 3 bits long, the first 3 bits of this parameter represent the IEEE 802.1 priority tag value, and the last 5 bits are padded to 0.

D_Layer3: 8 bits - This parameter is defined in section 5.29 of [RFC4710] and represents the layer 3 QoS marking used by the receiver to send packets to the data source during sub-session RC_N, if the Data Source can learn such information.

Source Payload Type: 8 bits - This parameter is defined in section 5.24 of [RFC4710] and specifies the payload type of the data source of the communication sub-session RC_N as defined in [RFC3551].

Receiver Payload Type: 8 bits - This parameter is defined in section 5.25 of [RFC4710] and specifies the receiver payload type of the communication sub-session RC_N as defined in [RFC3551].

CPU Utilization: 8 bits - This parameter, defined in section 5.30 of [RFC4710], represents the percentage of CPU used during session RC_N from the last report until the time this RAQMON PDU was generated. The CPU Utilization is expressed in percents in the range 0 to 100. The value should indicate not only CPU utilization associated to a session RC_N but also actual CPU Utilization, to indicate a snapshot of the CPU utilization of the host running the RDS while session RC_N in progress.

Memory Utilization: 8 bits - This parameter, defined in section 5.31 of [RFC4710], represents the percentage of total memory used during session RC_N up until the time this RAQMON PDU was generated. The memory utilization is expressed in percents 0 to 100. The Memory Utilization value should indicate not only the memory utilization associated to a session RC_N but the total memory utilization, to indicate a snapshot of end-device memory utilization while session RC_N is in progress.

Session Setup Delay: 16 bits - The Session (sub-session) Setup Delay metric is defined in section 5.8 of [RFC4710] and expressed in milliseconds.

Application Delay: 16 bits - The Application Delay is defined in section 5.13 of [RFC4710] and is represented as an unsigned integer expressed in milliseconds.

IP Packet Delay Variation: 16 bits - The IP Packet Delay Variation is defined in section 5.15 of [RFC4710] and is represented as an unsigned integer expressed in milliseconds.

Inter-Arrival Jitter: 16 bits - The Inter-Arrival Jitter is defined in section 5.14 of [RFC4710] and is represented as an unsigned integer expressed in milliseconds.

Packet Discard in Fraction: 8 bits - This parameter is defined in section 5.23 of [RFC4710] and is expressed as a fixed-point number with the binary point at the left edge of the field. (That is equivalent to taking the integer part after multiplying the discard fraction by 256.) This metric is defined to be the number of packets discarded, divided by the total number of packets.

Packet Loss in Fraction: 8 bits - This parameter is defined in section 5.21 of [RFC4710] and is expressed as a fixed-point number, with the binary point at the left edge of the field. The metric is defined to be the number of packets lost divided by the number of packets expected. The value is calculated by dividing the total number of packets lost (after the effects of applying any error protection, such as Forward Error Correction (FEC)) by the total number of packets expected, multiplying the result of the division by 256, limiting the maximum value to 255 (to avoid overflow), and taking the integer part.

padding: 0, 8, 16, or 24 bits - If the padding bit (P) is set, then this field may be present. The actual padding at the end of the BASIC part of the PDU is 0, 8, 16, or 24 bits to make the length of the BASIC part of the PDU a multiple of 32 bits.

2.1.3. APP Part of the RAQMON Protocol Data Unit

The APP part of the RAQMON PDU is intended to accommodate extensions for new applications in a modular manner and without requiring a PDU type value registration.

Vendors may design and publish application-specific extensions. Any RAQMON-compliant RRC MUST be able to recognize vendors' SMI Enterprise Codes and MUST recognize the presence of application-specific extensions identified by using Report Type fields. As represented in Figure 1, the Report Type and Application Length fields are always located at a fixed offset relative to the start of the extension fields. There is no need for the RRC to understand the semantics of the enterprise-specific parts of the PDU.

SMI Enterprise Code: 32 bits - Vendors and application developers should fill in appropriate SMI Enterprise IDs available at http://www.iana.org/assignments/enterprise-numbers. A non-zero SMI Enterprise Code indicates a vendor- or application-specific extension.

RAQMON PDUs are capable of carrying multiple Application Parts within a PDU.

Report Type: 16 bits - Vendors and application developers should fill in the appropriate report type within a specified SMI Enterprise Code. It is RECOMMENDED that vendors publish application-specific extensions and maintain such report types for better interoperability.

Length of the Application Part: 16 bits (unsigned integer) - The length of the Application Part of the RAQMON PDU in 32-bit words minus one, which includes the header of the Application Part.

Application-dependent data: variable length - Application/ vendor-dependent data is defined by the application developers. It is interpreted by the vendor-specific application and not by the RRC itself. Its length must be a multiple of 32 bits and will be padded if necessary.

2.1.4. Byte Order, Alignment, and Time Format of RAQMON PDUs

All integer fields are carried in network byte order, that is, most significant byte (octet) first. This byte order is commonly known as big-endian. The transmission order is described in detail in [RFC791]. Unless otherwise noted, numeric constants are in decimal (base 10).

All header data is aligned to its natural length, i.e., 16-bit fields are aligned on even offsets, 32-bit fields are aligned at offsets divisible by four, etc. Octets designated as padding have the value zero.

2.2. Securing RAQMON Session

The RAQMON session, initiated over TCP transport, between an RDS and an RRC carries monitoring information from an RDS client to the RRC, the collector. The RRC distinguishes between clients based on various identifiers used by the RDS to identify itself to the RRC

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/048041106126007004