- AVC Home Page

Contents

Application Response Time (ART)

On ISR-G2, ART engine is available with Performance Agent (aka MACE). On ASR1k, ART engine is brought into Metric Mediation Agent (MMA) under performance-monitoring CLI.

Application Name

NBAR Application name provides the information regarding the L7 level information for a particular flow, e.g HTTP, FTP, SIP etc.. There is an ID exported by ART, which explains which application this flow belongs to. The Application-ID is divided into two parts: Engine-ID:Classification-id.

First 8 bits provides the information about engine, which classified this flow. For example: IANA-L4, CANA-L3 etc. The rest of the 24 bits provides information about the application, for example 80 (HTTP) etc. The CLI for this field is:

```
flow record type mace <pa-record>
 collect application name
```

It is exported against FNF field ID 95. It is supported in both the export formats i.e. netflow-v9 and IPFIX. With this CLI, only the application ID would be exported. The application-id would be a number which may not be understood by collector. To resolve this issue, there is an option to export the mapping table between the application ID and the application name. This option is configurable under flow exporter command. Here is the cli:

```
flow exporter <my-exporter>
```

```
 option application-table
```

If you want to get more information regarding the various attributes of a particular application, you can configure the following option under the flow exporter:

```
flow exporter <my-exporter>
 option application-attribute
```

The option command results in periodic export of Network Based Application Recognition (NBAR) application attributes to the collector. The following application attributes are sent to the collector per protocol:

Attribute	**Description**
Category	Provides first-level categorization for each application.
Sub-Category	Provides second-level categorization for each application.
Application-Group	Groups applications that belong to the same networking application.
P2P-Technology	Specifies whether the application is based on peer-to-peer technology or not.
Tunnel-Technology	Specifies whether the application tunnels the traffic of other protocols or not.
Encrypted	Specifies whether the application is an encrypted networking protocol or not. ? The optional timeout can alter the frequency at which reports are sent.

The following command can be used to display the option tables exported to the collector for application name mapping and attributes:

```
R9#show flow exporter templates
Flow Exporter MYEXPORTER:

[SNIP]

Client: Option options application-name
  Exporter Format: IPFIX (Version 10)
  Template ID    : 261
  Source ID      : 0
  Record Size    : 83
  Template layout
  _____________________________________________________________________________
  |                 Field                   |    ID | Ent.ID | Offset |  Size |
  -----------------------------------------------------------------------------
  | APPLICATION ID                          |    95 |        |      0 |     4 |
  | application name                        |    96 |        |      4 |    24 |
  | application description                 |    94 |        |     28 |    55 |
  -----------------------------------------------------------------------------
  Client: Option options application-attributes
  Exporter Format: IPFIX (Version 10)
  Template ID    : 262
  Source ID      : 0
  Record Size    : 130
  Template layout
  _____________________________________________________________________________
  |                 Field                   |    ID | Ent.ID | Offset |  Size |
```

```
  ------------------------------------------------------------------------------
  | APPLICATION ID                          |    95 |        |      0 |      4 |
  | application category name               | 12232 |      9 |      4 |     32 |
  | application sub category name           | 12233 |      9 |     36 |     32 |
  | application group name                  | 12234 |      9 |     68 |     32 |
  | p2p technology                          |   288 |        |    100 |     10 |
  | tunnel technology                       |   289 |        |    110 |     10 |
  | encrypted technology                    |   290 |        |    120 |     10 |
  ------------------------------------------------------------------------------

[SNIP]
```

The following show commands can be used to display NBAR option tables:

- display all the application ID and the name mapping

```
R9#show flow exporter option application table

Engine: prot (IANA_L3_STANDARD, ID: 1)

appID  Name                     Description
-----  ----                     -----------
1:8    egp                      Exterior Gateway Protocol
1:47   gre                      General Routing Encapsulation
1:1    icmp                     Internet Control Message Protocol
1:88   eigrp                    Enhanced Interior Gateway Routing Protocol

[SNIP]
```

- display the NBAR field extraction format which concatenates App ID|Sub-classification ID|Value

```
R9#show ip nbar parameter extraction

 Protocol        Parameter            ID
 --------        ---------            --
 smtp            sender               4666370
 smtp            server               4666369
 pop3            server               2176001
 sip             source               4273154
 sip             destination          4273153
 nntp            group-name           1848321
 http            referer              209924
 http            user-agent           209923
 http            host                 209922
 http            url                  209921
R9#
```

PA Metrics

PA provides the basic metrics for both TCP and UDP protocols and for both IPv4 and IPv6. Some of the metrics are dynamically exported in the form of the delta value for the interval. These include the client/server bytes and packets metrics. In addition, PA server/client bytes/packets metrics are for layer-3 measurements and in a TCP flow are counted up to the second FIN. The rest of the metrics are relatively static and remain the same across different export intervals.

PA keeps exporting the measurements as long as the flow stays active. As a consequence, the collector might occasionally observe zero values for dynamic metrics such as the client/server bytes/packets. For all the UDP flows, the TCP related metrics such as ART metrics would be zero. Another note for the Input/Output interface metrics is that these are corresponding to the interface from which the flow enters/leaves the box.

All the PA metrics can be exported either through Netflow v9 or IPFIX protocol. PA metrics are summarized below.

Field Name	Export ID	CLI	Description
Application ID	95	collect application name	exports application ID field (coming from NBAR2) to reporting tool.
Client Bytes	1	collect counter client bytes	Total bytes sent by initiator of the connection. Counted up to the second FIN if for a TCP flow.
Client Packets	2	collect counter client packets	Total packets sent by initiator of the connection. Counted up to the second FIN if for a TCP flow.
Interface Input	10	collect interface input	Interface name from which flow is entering the box.
Interface Output	14	collect interface output	Interface name from which flow is exiting out the box.
Server Bytes	23	collect counter server bytes	Total bytes sent by responder of the connection. Counted up to the second FIN if for a TCP flow.
Server Packets	24	collect counter server packets	Total packets sent by responder of the connection. Counted up to the second FIN if for a TCP flow.
Datalink Mac Source Address Input	56	collect datalink mac source address input	MAC address of source device from Input side
IPv4 DSCP	195	collect ipv4 dscp	IPv4 DSCP value
IPv6 DSCP	195	collect ipv6 dscp	IPv6 DSCP value

ART Metrics

- Available on ISR-G2 with PA
- Available on ASR1k with MMA, starting from IOS-XE 3.8

Field Name	Export ID	CLI	Description
Client Network Time [sum/min/max]	• 42084 (sum) • 42085 (max) • 42086 (min)	• collect art client network time sum • collect art client network time minimum • collect art client network time maximum	The round trip time between SYN-ACK & ACK and also called Client Network Delay (CND). CND = T8 ? T5
Server Network Time [sum/min/max]	• 42087(sum) • 42088(max) • 42089(min)	• collect art server network time sum • collect art server network time minimum • collect art server network time maximum	The round trip time between SYN & SYN-ACK and also called Server Network Delay (SND). SND = T5 - T2
Network Time [sum/min/max]	• 42081(sum) • 42082(max) • 42083(min)	• collect art network time sum • collect art network time minimum • collect art network	The round trip time that is the summation of CND and SND. It is also called Network Delay (ND).

		time maximum	
Server Response Time [sum/min/max]	• 42074(sum) • 42075(max) • 42076(min)	• collect art server response time sum • collect art server response time minimum • collect art server response time maximum	The time taken by an application to respond to a request. It is also called Application Delay (AD) or Application Response Time. • AD = RT ? SND • min_AD = min_RT ? sum_SND/no. of sessions • max_AD = max_RT ? sum_SND/no. of sessions • sum_AD = sum_RT ? (sum_SND*no. of responses)/no. of sessions
Response Time [sum/min/max]	• 42071(sum) • 42072(max) • 42073(min)	• collect art response time sum • collect art response time minimum • collect art response time maximum	The amount of time between the Client REQ and the 1st Server RESP. The Client request could contain multiple packets and we consider the time of last received client packet.
Total Response Time [sum/min/max]	• 42077(sum) • 42078(max) • 42079(min)	• collect art total response time sum • collect art total response time minimum • collect art total response time maximum	The total time taken from the moment the client sends the request until the 1st response packet from the server is delivered to the client. It is also known as Total Delay (TD). • TD = RT + CND • min_totalDelay = min(min_RT + sum_CND/no. of sessions, responses + min_CND) • max_totalDelay = max(max_RT + sum_CND/no. of sessions, sum_RT/no. of responses + max_CND) • sum_totalDelay = sum_RT + (sum_CND* No of responses) /no. of sessions.
Total Transaction Time [sum/min/max]	• 42041(sum) • 42042(max) • 42043(min)	• collect art total transaction	The amount of time between the client request and the final response packet from the server. It is measured and exported on receiving either a new request from client (which indicates end of current

		time sum • collect art total transaction time minimum • collect art total transaction time maximum	transaction) or the first FIN packet.
ART Client Bytes / Packets	• 231(bytes) • 42033(packets)	• collect art client bytes • collect art client packets	Byte & Packet count for all the client packets. • ART client/server bytes/packets will be reported when the flow is completed or the first server response packet is received. • For long-lived flows, e.g. if flow duration is longer than 2 export cycles, the following behavior is expected • client/server bytes/packets will be reported when first server response packet is received; • those metrics may not be updated during the intermediate export cycle before flow is completed or transaction ends. • During the export cycle when flow is completed, client/server bytes/packets will be updated and reported.
ART Server Bytes / Packets	• 232(bytes) • 42034(packets)	• collect art server bytes • collect art server packets	Byte & Packet count for all the server packets. • ART client/server bytes/packets will be reported when the flow is completed or the first server response packet is received • For long-lived flows, e.g. if flow duration is longer than 2 export cycles, the following behavior is expected • client/server bytes/packets will be reported when first server response packet is received; • those metrics may not be updated during the intermediate export cycle before flow is completed. • During the export cycle when flow is completed, client/server bytes/packets will be updated and reported.
ART Count New Connections	• 42050	• collect art count new connections	Number of TCP sessions established (3-way handshake). It is also called Number of connections (sessions).
ART Count Responses	• 42060	• collect art count	Number of Req-Rsp pair received within the monitoring interval

		responses	
Responses histogram buckets (7-bucket histogram))	• 42061-42067	• collect art count responses histogram	Number of responses by response time in 7-bucket histogram. • Threshold values for 7 buckets are 2, 5, 10, 50, 100, 500, 1000 milliseconds; • Bucket 1, response time < 2 milliseconds; • Bucket 2, response time is between 2-5 milliseconds; • Bucket 3, response time is between 5-10 milliseconds; • Bucket 4, response time is between 10-50 milliseconds; • Bucket 5, response time is between 50-100 milliseconds; • Bucket 6, response time is between 100-500 milliseconds; • Bucket 7, response time is between 500 - 1000 milliseconds; • If response time is great than 1000 milliseconds, it will be considered as timeout. This response is not considered towards the min/max/sum response time calculation. Nor is it considered for ART packets/bytes metrics calculation. • For example, if response time is equal to 9 milliseconds, it will goes to bucket 3;
Art Count Late Responses	• 42068	• collect art count late responses	Number of responses received after the max Response Time. Current threshold of timeout is 1 second. Also called Number of late responses (timeouts)
Art Count Transactions	• 42040	• collect art count transactions	Total number of Transactions for all TCP connections. • A new transaction is counted under any one of the following 3 conditions: 1. Receiving a data packet from client request while the previous packet state is server response; 2. Receiving a client FIN packet while the previous packet state is server response; 3. Receiving a server FIN packet while the previous packet state is server response;
Art Count Retransmissions	• 42036	• collect art count retransmissions	Packet count for possible retransmitted packets with the same sequence number as the last received packet. The metric is for client retransmission only.
Art All Metrics	• N/A		Single CLI to collect all the ART related metrics in mace. This CLI works as a replacement of all the

		• collect art all	ART related collect statements in a flow record.

Top Domain, URL Hit Count Report

Supported on ISR-G2 with Performance Agent, starting from IOS 15.2(4)M2. The list of new NBAR related fields are:

- HTTP Host
- URI and URI Statistics

If the user configures one or more of the above fields, PA control plane will explicitly activate NBAR. The HTTP fields can be only exported through IPFIX protocol (NetFlow version9 doesn't support variable field length).

Field Name	Export ID	CLI	Description
application http host	• 45003	• collect application http host	Host name
application http uri statistics	• 42125	• collect application http uri statistics	URI Statistics
art count new connections	• 42050	• collect art count new connections	Number of TCP sessions established (3-way handshake). It is also called Number of connections (sessions).

Host

HTTP Host field is exported with Netflow export id 45003. It would be encoded as:

```
0                    31 32                47 48
+--------------------+--------------------+----------------
| NBAR Application ID | Sub-Application ID | Value (host)
+--------------------+--------------------+----------------
```

Collectors will be able to identify the extracted field name and type based on the application ID and sub-application ID that are embedded in it.

- For HTTP Host, the application-id is generally 0x03000050 (see previous chapter on application ID to know how to get this value). The first byte (03) specifies engine-id (i.e. IANA-L4), next 3 bytes (000050) are for selector-id (i.e. decimal 80 for HTTP).
- Sub- application-id for host is 0x0002. From show ip nbar parameters extraction and sub-application-table option template. Only take the last two bytes, 0x3402 = HTTP Host
- Value is the host string.

PA exports one host-name for each L7 flow. The CLI for host name is:

```
flow record type mace <pa-record>
 collect application http host
```

URI and Count

PA will collect and export URI and hit-count in the format ?uri:count::uri:count.....?. The delimiters colon (:) and double colon (::) are written here just for the demonstration of the format. The actual delimiter would be NULL (\0). URI and count is always represented in binary format using fixed length 2bytes. The collector has to parse the URI by parsing on the basis of delimiters i.e. NULL (\0). URI count is read as a 2byte binary number and there is no '\0' delimiter after the count. A special PA specific FNF export field (42125) would be used to export the list of URIs and the corresponding hit-counts. The encoding would be done as follows:

```
{URI\0countURI\0count}
```

Please note that the URI being collected and exported is limited to the first ?/?. For example, if the URL is http://www.cisco.com/router/isr/g2, the URI collected is ?router?. Example: The following flows are collected on the router: 10.1.1.1 to 10.2.2.2, destination port 80, protocol TCP www.cisco.com/US - 2 flows www.cisco.com/WORLD - 1 flow

The data would then be exported as: [10.1.1.1, 10.2.2.2, 80, TCP, www.cisco.com, US\02WORLD\01]

The CLI to collect uri and its hit-count (statistics) is:

```
flow record type mace <pa-record>
 collect application http uri statistics
```

QoS Class-ID, Queue Drops and Queue Hierarchy

Supported on ISR-G2 with Performance Agent, starting from IOS 15.2(4)M2. The list of new QoS related fields are:

- The QoS classification hierarchy
- The QoS queue drops

Field Name	Export ID	CLI	Description
QoS Policy Classification Hierarchy	• 41000	• collect policy qos classification hierarchy	Report application class of service hierarchy
QoS Queue Index	• 42128	• collect policy qos classification hierarchy	Report Queue Index
QoS Queue Drops	• 42129	• collect policy qos queue drops	Number of drops in the queue
Timestamp absolute	• 65501		Timestamp when monitoring interval expires. Added automatically with ?collect policy qos queue

monitoring-interval		• No CLI is needed.	drops'

QoS policy Classification Hierarchy

To get the QoS queue for a particular flow, PA will export the hierarchy of the class that the flow matched on. This hierarchy will be exported in the flow record as a list of IDs: {Policy ID, Class ID 1, Class ID 2, Class ID 3, Class ID 4, Class ID 5}. Each of these IDs is a 4-byte integer representing a C3PL policy-map or class-map, with any missing or unnecessary fields being 0. Total length of the field is 24 bytes. The ID to name mapping will be exported as an option template under a flow exporter.

FNF export ID 41000 would be used for this metric. It is supported by netflow-v9 as well as IPFIX. Here is the CLI:

```
flow record type mace <pa-record>
 collect policy qos classification hierarchy
```

Class hierarchy will be collected in configurations where PA and QoS are applied on the same L3 interface or in configurations where PA is applied on a dialer interface.

Example:

```
class-map match-all C1
 match any
class-map match-all C11
 match ip dscp ef
class-map match-all C12
 match ip dscp cs2
!
policy-map Child-P11
class child-class-C11
 bandwidth remaining percent 10
class child-class-C12
 bandwidth remaining percent 70
class class-default
 bandwidth remaining percent 20
!
policy-map Parent-P1
 class child-class-C1
  shaping average 16000000
  service-policy Child-P11
!
interface e0/0
 service-policy out Parent-P1
!
```

PA will export the QoS class hierarchy information as follows:

Flow-ID	Class Hierarchy (41000)	Queue id (42128)
Flow-1	P1, C1, C11, 0, 0, 0	1
Flow-2	P1, C1, C11, 0, 0, 0	1
Flow-3	P1, C1, C12, 0, 0, 0	2

Two option templates would be used to export the class and policy information. One template would be for class-id and class-name mapping, a second one is for policy-id/policy-name mapping..The sample CLI is as follows:

```
flow exporter <my-exporter>
 option c3pl-class-table
 option c3pl-policy-table
```

The following command can be used in exec mode to display the option tables exported to the collector for class and policy mapping:

```
show flow exporter <my-exporter> templates
```

Example:

- Class-map Option Template

```
router#show flow exporter MYEXPORTER templates

[SNIP]

Flow Exporter MYEXPORTER:
Client: Option classmap option table
  Exporter Format: NetFlow Version 9
  Template ID    : 261
  Source ID      : 0
  Record Size    : 304
  Template layout
  _____________________________________________________________________
  |                Field                    |  Type | Offset |  Size  |
  ---------------------------------------------------------------------
  | v9-scope system                         |     1 |     0  |     4  |
  | c3pl class cce-id                       | 41001 |     4  |     4  |
  | c3pl class name                         | 41002 |     8  |    40  |
  | c3pl class type                         | 41003 |    48  |   256  |
  ---------------------------------------------------------------------

[SNIP]
```

- Policy-map Option Template

```
router#show flow exporter MYEXPORTER templates

[SNIP]

Flow Exporter MYEXPORTER:
 Client: Option policymap option table
  Exporter Format: NetFlow Version 9
  Template ID    : 262
  Source ID      : 0
  Record Size    : 304
  Template layout
  _____________________________________________________________________
  |                Field                    |  Type | Offset |  Size  |
  ---------------------------------------------------------------------
  | v9-scope system                         |     1 |     0  |     4  |
  | c3pl policy cce-id                      | 41004 |     4  |     4  |
  | c3pl policy name                        | 41005 |     8  |    40  |
  | c3pl policy type                        | 41006 |    48  |   256  |
  ---------------------------------------------------------------------
```

```
[SNIP]
```

QoS Queue drops

The queue statistics that are exported will be the drops seen due to the queuing action configured under a class-map. These drops will be collected per export interval from the queuing feature and will be exported in a separate option template table. This option template table will identify the drops seen for a particular queue id, and the flow record will identify the id of the queue that a flow was queued in. As we?re exporting the drops at a queue level, the drops seen under a queue will be a summation of the drops experienced by multiple flows that matched on the same class-map hierarchy.

FNF export ID 42129 would be used for this metric. It is supported by netflow-v9 as well as IPFIX. Here is the CLI:

```
flow record type mace <pa-record>
 collect policy qos queue drops
```

As a result of above command, two sets of export will happen.

- The first one is the data that is exported as part of each flow entry. In addition to all other metrics, queue-id and timestamp will be exported per flow entry.

Here is the example:

Flow-ID	Queue id (42128)	Timestamp (65501)
Flow-1	1	50000
Flow-2	2	50000

- The second part is the data which is exported as part of option template. It is exported once when the PA timer expires. Queue-id, timestamp and queue-drops are included in the template. Here is the example:

Queue-id (42128)	Timestamp (65501)	Packet Drops (42129)
1	50000	100
2	50000	20

From the above two tables, user can match queue-id and timestamp and figure out the queue-drops from the queue to which a particular FLOW belongs to.

In configurations where there are no explicit queues specified under a class-map, the drop count collected for this class-map will be the drop count from the default queue. This behavior follows the QoS behavior of directing flows without a queuing action to the default queue. As such, if there are multiple classes without a queuing action, flows that match any of these class-maps will all see the same drop count (drops seen in the default queue).

The following command can be used in exec mode to check PA templates exported and especially the template for MACE (IPv4 and.or IPv6) and Queue Flow:

```
# show flow exporter <my-exporter> templates
```

Example:

Please note that the export protocol is IPFIX in the example below. And there is a difference between NFv9 and IPFIX IDs.

- IPFIX fields consist of an E (Enterprise) bit, followed by a 15-bit ID.

If topmost bit = "E" then the Field Id is enterprise-specific versus IANA standard.

- FNF treats this as a single 16-bit field ID.

policy qos classification hierarchy = IPFIX 8232 + Enterprise ID 9 In NFv9, it would be 0x8000 (32768) + 8232 = 41000

Although it looks different, it is exactly the same value.

```
R9#sh flow exporter MYEXPORTER templates

[SNIP]

  Client: MACE EXPORTER GROUP MACE-EXP-1
  Exporter Format: IPFIX (Version 10)
  Template ID    : 264
  Source ID      : 0
  Record Size    : 263 + var
  Template layout
  _____________________________________________________________________________
  |                 Field                    |    ID | Ent.ID | Offset |  Size |
  -----------------------------------------------------------------------------
  | ipv4 source address                      |     8 |        |      0 |     4 |
  | ipv4 destination address                 |    12 |        |      4 |     4 |
  | transport destination-port               |    11 |        |      8 |     2 |
  | ip protocol                              |     4 |        |     10 |     1 |
  | waas optimization segment                |  9252 |      9 |     11 |     1 |
  | application http uri statistics          |  9357 |      9 |     13 |   var |
  | application http host                    | 12235 |      9 |     16 |   var |
  | ip dscp                                  |   195 |        |     18 |     1 |
  | application id                           |    95 |        |     19 |     4 |
  | art response time sum                    |  9303 |      9 |     23 |     4 |
  | art response time minimum                |  9305 |      9 |     27 |     4 |
  | art response time maximum                |  9304 |      9 |     31 |     4 |
  | art server response time sum             |  9306 |      9 |     35 |     4 |
  | art server response time minimum         |  9308 |      9 |     39 |     4 |
  | art server response time maximum         |  9307 |      9 |     43 |     4 |
  | art network time sum                     |  9313 |      9 |     47 |     4 |
  | art network time minimum                 |  9315 |      9 |     51 |     4 |
  | art network time maximum                 |  9314 |      9 |     55 |     4 |
  | art client network time sum              |  9316 |      9 |     59 |     4 |
  | art client network time minimum          |  9318 |      9 |     63 |     4 |
  | art client network time maximum          |  9317 |      9 |     67 |     4 |
  | art server network time sum              |  9319 |      9 |     71 |     4 |
  | art server network time minimum          |  9321 |      9 |     75 |     4 |
  | art server network time maximum          |  9320 |      9 |     79 |     4 |
```

art total response time sum	9309	9	83	4
art total response time minimum	9311	9	87	4
art total response time maximum	9310	9	91	4
art total transaction time sum	9273	9	95	4
art total transaction time minimum	9275	9	99	4
art total transaction time maximum	9274	9	103	4
art count transactions	9272	9	107	4
art count retransmissions	9268	9	111	4
art count new connections	9282	9	115	4
art count responses	9292	9	119	4
art count late responses	9300	9	123	4
art count responses histogram bucket1	9293	9	127	4
art count responses histogram bucket2	9294	9	131	4
art count responses histogram bucket3	9295	9	135	4
art count responses histogram bucket4	9296	9	139	4
art count responses histogram bucket5	9297	9	143	4
art count responses histogram bucket6	9298	9	147	4
art count responses histogram bucket7	9299	9	151	4
policy qos queue index	9360	9	155	4
interface input snmp	10		159	4
interface output snmp	14		163	4
counter server bytes	23		167	8
counter server packets	24		175	8
policy qos classification hierarchy	8232	9	183	24
art server packets	9266	9	207	8
art server bytes	232		215	8
art client packets	9265	9	223	8
art client bytes	231		231	8
timestamp absolute monitoring-interval	32733	9	239	8
counter bytes long	1		247	8
counter packets long	2		255	8

[SNIP]

Client: MACE EXPORTER GROUP MACE Queue Flow
Exporter Format: IPFIX (Version 10)
Template ID : 263
Source ID : 0
Record Size : 20
Template layout

Field	ID	Ent.ID	Offset	Size
POLICY QOS QUEUE INDEX	9360	9	0	4
policy qos queue drops	9361	9	4	8
timestamp absolute monitoring-interval	32733	9	12	8