

eVIP Network Counters Measurements

Evolved Virtual IP

InterWork Description

Copyright

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 2 | Performance Measurement Groups | 3 |
| 3 | Measurements of EvipNetstat PM Group | 5 |
| 4 | Performance Measurement Jobs | 9 |
| 5 | Measurement Job Results | 11 |
| 5.1 | Example ROID or measObjLdn | 11 |





1 Introduction

This document describes the PM Counters interface provided by eVIP. A number of measurements are available to report network statistics provided by eVIP.

eVIP PM counters are only available when eVIP is run in a Core MW environment and cannot be used by eVIP for LSB.

For more information about Performance Management, refer to [Core MW Performance Management Description](#).





2 Performance Measurement Groups

eVIP has one PM Group that applies to both real and virtualized hardware.

Table 1 PM Group Names

| pmGroupId | Properties |
|-------------|---|
| EvipNetstat | <ul style="list-style-type: none">• description: Network statistics for a particular FEE or ALB. Syntax for ROID or measObjLdn (Measured Object Local Distinguished Name) of related measurementType: Alb=<alb_name> or Alb=<alb_name>,Fee=<fee_name>. See Section 5.1 on page 11• pmGroupVersion: 1 |

Note: Providing the FEE name is optional. When FEE is omitted, aggregated values of all FEEs within the ALB are provided.





3 Measurements of EvipNetstat PM Group

PM producer daemons (apart of evip_amf_proxy) on each eVIP node collect statistical information which is reported via the Core MW Performance Management Interface. For more details on this interface, refer to [Core MW Performance Management Description](#). All network measurements are reported as delta (change in value) from previous sample.

Table 2 eVIP Measurement Instances

| measurementTypeId | Properties |
|-------------------|---|
| ip4OutOctets | <ul style="list-style-type: none"> • description: IPv4 Octets Sent. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic. • collectionMethod: CC • aggregation: SUM |
| ip6OutOctets | <ul style="list-style-type: none"> • description: IPv6 Octets Sent. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic. • collectionMethod: CC • aggregation:: SUM |
| ip4OutDatagrams | <ul style="list-style-type: none"> • description: IPv4 Datagrams Sent. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic. • collectionMethod: CC • aggregation:SUM |
| ip6OutDatagrams | <ul style="list-style-type: none"> • description: IPv6 Datagrams Sent. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic. • collectionMethod: CC • aggregation:: SUM |



| | |
|----------------|--|
| ip4InOctets | <ul style="list-style-type: none">• description: IPv4 Octets Received. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic.• collectionMethod: CC• aggregation: SUM |
| ip6InOctets | <ul style="list-style-type: none">• description: IPv6 Octets Received. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic.• collectionMethod: CC• aggregation: SUM |
| ip4InDatagrams | <ul style="list-style-type: none">• description: IPv4 Datagrams Received. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic.• collectionMethod: CC• aggregation: SUM |
| ip6InDatagrams | <ul style="list-style-type: none">• description: IPv6 Datagrams Received. The measured value includes data passing through FEE elements and does not include data related to routing protocols and intra-cluster traffic.• collectionMethod: CC• aggregation: SUM |
| I2RxDropped | <ul style="list-style-type: none">• description: Number of received ethernet frames with wrong VLAN or proto id unsupported/not configured on this interface• collectionMethod: CC• aggregation: SUM |
| I2RxErrors | <ul style="list-style-type: none">• description: Number of received ethernet frames with size not divisible by 8• collectionMethod: CC• aggregation: SUM |



| | |
|-------------------|--|
| I2RxFifoErrors | <ul style="list-style-type: none"> • description: Number of received frames which were received when queue from device driver to IP stack in kernel was full i.e. no of ingress overruns • collectionMethod: CC • aggregation: SUM |
| I2RxFrameErrors | <ul style="list-style-type: none"> • description: Number of frames received with CRC errors, too-short frames and too-long frames • collectionMethod: CC • aggregation: SUM |
| I2TxCarrierErrors | <ul style="list-style-type: none"> • description: Number of carrier sense errors detected on wire • collectionMethod: CC • aggregation: SUM |
| I2TxCollisions | <ul style="list-style-type: none"> • description: Number of times CSMA/CD collisions were detected on wire • collectionMethod: CC • aggregation: SUM |
| I2TxDropped | <ul style="list-style-type: none"> • description: Number of frames dropped by device driver • collectionMethod: CC • aggregation: SUM |
| I2TxErrors | <ul style="list-style-type: none"> • description: Number of ethernet frame sending errors reported by device driver • collectionMethod: CC • aggregation: SUM |
| I2TxFifoErrors | <ul style="list-style-type: none"> • description: Number of received frames which were received when queue from device driver to IP stack in kernel was full i.e. no of ingress overruns • collectionMethod: CC • aggregation: SUM |





4 Performance Measurement Jobs

Measurement jobs can be created as [ECIM PM, Use Case Description](#). The COM CLI can be used to create, activate, modify, and delete PM jobs as per examples below.

```
configure
ManagedElement=1,SystemFunctions=1,Pm=1
PmJob=MyEvipJob
requestedJobState=STOPPED
granularityPeriod=ONE_MIN
reportingPeriod=ONE_MIN
MeasurementReader=1
measurementSpecification,groupRef="ManagedElement=1,SystemFunctions=1,Pm=1,PmGroup=EvipNetstat"
commit
show
exit
```

Example 1 Example via ECIM, to set up a new measurement job:

```
configure
ManagedElement=1,SystemFunctions=1,Pm=1
PmJob=MyEvipJob
requestedJobState=ACTIVE
commit
show
exit
```

Example 2 Example via ECIM, to activate the measurement job:

```
configure
ManagedElement=1,SystemFunctions=1,Pm=1
PmJob=MyEvipJob
granularityPeriod=FIVE_MIN
reportingPeriod=FIVE_MIN
commit
exit
```

Example 3 Example via ECIM, to change the reporting and granularity period to five minutes:

```
configure
ManagedElement=1,SystemFunctions=1,Pm=1
PmJob=MyEvipJob
requestedJobState=STOPPED
commit
up
configure
no PmJob=MyEvipJob
commit
exit
```

Example 4 Example via ECIM, to delete a measurement job:

Alternatively, Core MW command line tools can be used to create a job as per examples below.

```
cmw-pmjob-create MyEvipJob -u EvipNetstat -p3 -t1 -q1 -g3 -r3
cmw-pm-show-counters Alb=alb1,Fee=fee1
```

See further relevant command line tools provided by Core MW:
cmw-pmjob-create, cmw-pmjob-delete, cmw-pmjob-list, cmw-pmjob-modify, cmw-pmjob-start, cmw-pmjob-status, cmw-pmjob-stop, cmw-pm-show-counters.

Example 5 Example via Core MW commands:



Note:

- Due to Core MW PM architecture and its method of collection, changes to FEE allocation, such as reboot, shutdown, scaling-in, can cause up to one granularity period of packet counter data to be lost.
- When adding a job, the granularity_period should be equal to reporting_period. Periods below one minute are not possible.



5 Measurement Job Results

- At the end of the configured granularity period, a report with counter values is output to the folder
 - /storage/no-backup/com-apr9010443/PerformanceManagementReportFiles
- Alternatively, the helper command "cmw-pm-show-counters" can display counters at runtime, with the following syntax.
 - cmw-pm-show-counters <measObjLdn>

For measObjLdn, see Section 5.1 on page 11.

Here are some examples:

- Show counter values per FEE:
 - cmw-pm-show-counters Alb=alb1,Fee=fee1
- Show the aggregated values of counters per ALB:
 - cmw-pm-show-counters Alb=alb1

5.1 Example ROID or measObjLdn

The ROID or measObjLdn (Measured Object Local Distinguished Name) can be constructed for each measurement instance by using the correct identifiers after the appropriate PM Group.

For example

- Alb=alb1
- Alb=alb1,Fee=fee1
- Alb=alb1,Fee=fee2
- Alb=alb2
- Alb=alb2,Fee=fee1
- Alb=alb2,Fee=fee2