



## CPS Statistics

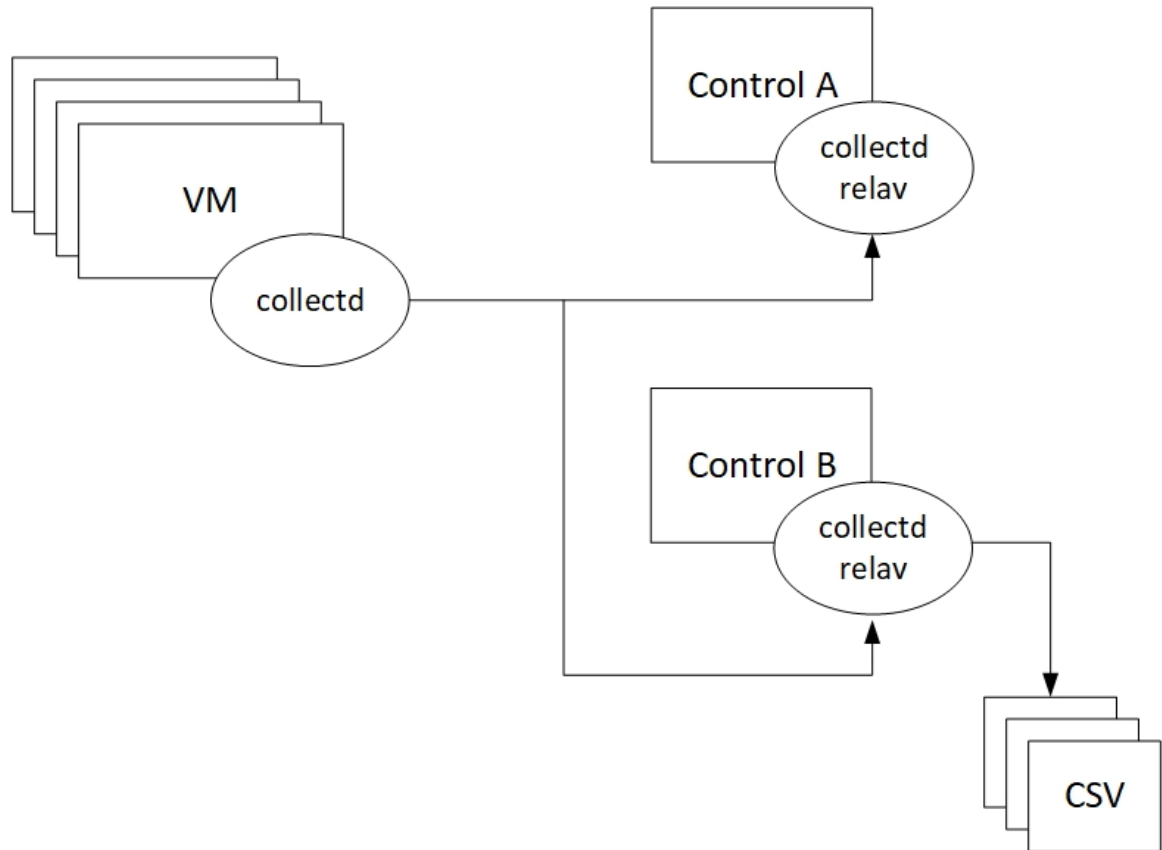
---

- [Bulk Statistics Overview](#), on page 1
- [CPS Statistics](#), on page 3
- [Bulk Statistics Collection](#), on page 3
- [Diameter Monitoring KPIs](#), on page 4
- [Example Statistics](#), on page 16

### Bulk Statistics Overview

Bulk Statistics are the statistics that are gathered over a given time period and written to a set of CSV files. These statistics can be used by external analytic processes and/or network management systems. The architecture of CPS bulk statistic collection is shown in the following illustration.

Figure 1: DRA Bulk Statistic Collection Architecture



The collection utility collectd is used for collecting and storing statistics from each VM. Detailed collectd documentation can be found on <http://collectd.org/>.

Collectd within CPS is deployed with nodes relaying data using the collectd network plug-in (<https://collectd.org/wiki/index.php/Plugin:Network>) to the centralized collection nodes on the control-A and control-B virtual machines. The centralized collector writes the collected data to output CSV files.




---

**Note** Control A and Control B collect bulk statistics independently. As a result, it is normal to have slight differences between the two files. For example, control-A will generate a file at time t and control-B will generate a file at time t +/- the clock drift between the two machines.

---

As a best practice, always use the bulk statistics collected from Control-A. Control-B can be used as a backup in the event of failure of control-A.

In the event that Control-A becomes unavailable, statistics will still be gathered on Control-B. Statistics data is not synchronized between Control-A and Control-B, so a gap would exist in the collected statistics while control-A is down.




---

**Note** The collectd collection mechanism are separate from the Prometheus / Grafana Monitoring.

---

# CPS Statistics

The list of statistics available in CPS is consolidated in an Excel spreadsheet. After CPS is installed, this spreadsheet can be downloaded from the Bulk Stats link available on below URL:

<https://<master ip>/central/dra/#!/dra/docs/stats>

## Bulk Statistics Collection

By default, CPS outputs a bulk statistics CSV file to the `/var/broadhop/stats/` directory on the control-A and control-B VMs in five-minute intervals.

An `scp / sftp` daemon running on port 2026 retrieves all statistics within the `/var/broadhop/stats` directory. Only locally defined users within the scheduling application associated to the “bulkstats” or “admin” group are able to retrieve statistics.

You can also retrieve statistics by logging into the virtual machine directly and retrieving the statistics from the `/data/stats` directory.

The default naming standard is `bulk-hostname-YYYY-MM-DD-HH-MI.csv`

These CSV files include all statistics collected from all VMs during the five-minute interval.



---

**Note** If a statistic is generated by the system multiple times within the five-minute interval, only the last measured statistics is collected in the CSV file.

---

The following list is a sample of the file names created in the `/var/broadhop/stats/` directory on the control-A VM:

```
[root@control-1 stats]# pwd
/data/stats-relay-s1/var/broadhop/stats [root@control-A stats]# ls
bulk-control-A-201510131350.csv
bulk-control-A-201510131355.csv
bulk-control-A-201510131400.csv
bulk-control-A-201510131405.csv
bulk-control-A-201510131410.csv
bulk-control-A-201510131415.csv
bulk-control-A-201510131420.csv
bulk-control-A-201510131425.csv
bulk-control-A-201510131430.csv
bulk-control-A-201510131435.csv
bulk-control-A-201510131440.csv
bulk-control-A-201510131445.csv
bulk-control-A-201510131450.csv
bulk-control-A-201510131455.csv
bulk-control-A-201510131500.csv
bulk-control-A-201510131505.csv
bulk-control-A-201510131510.csv
bulk-control-A-201510131515.csv
bulk-control-A-201510131520.csv
bulk-control-A-201510131525.csv
bulk-control-A-201510131530.csv
bulk-control-A-201510131535.csv
bulk-control-A-201510131540.csv
```

```

bulk-control-A-201510131545.csv
bulk-control-A-201510131550.csv
bulk-control-A-201510131555.csv
bulk-control-A-201510131600.csv
bulk-control-A-201510131605.csv
bulk-control-A-201510131610.csv
bulk-control-A-201510131615.csv
bulk-control-A-201510131620.csv
bulk-control-A-201510131625.csv
bulk-control-A-201510131630.csv

```

## Retention of CSV Files

CPS retains each bulk statistics CSV file on the control-A/B VM for two days; after which the file is automatically removed.

If you need to preserve these CSV files, you must back up the files or move them to an alternate system.

## Diameter Monitoring KPIs

The following table describes CPS KPIs that are useful for monitoring Diameter message traffic.



**Note** As each deployment is unique, no recommended ranges are provided. Cisco recommends monitoring these KPIs for a period of time (1-3 months) to establish a baseline. Deviations can then be monitored from the baseline values.

**Table 1: Diameter Monitoring KPIs**

| Appld/<br>Monitoring<br>Area | Category            | Statistic  | Description  | Availability/Node |
|------------------------------|---------------------|--|--|-------------------|
| Gx/F                         | Diameter Round Trip | node[x].messages.<br>e2e_<domain>_<br>[realm_] Gx_CCR-I_2001.<br>qns_stat.success              | Success message count for<br>return code 2001                                  | Policy Director   |
| Gx/F                         | Diameter Round Trip | node[x].messages.<br>e2e_<domain>_<br>[realm_] Gx_CCR-I_2001.<br>qns_stat.total<br>_time_in_ms | Total milliseconds of<br>successful messages with<br>return code matching 2001 | Policy Director   |

| Appld/<br>Monitoring<br>Area | Category             | Statistic   | Description  | Availability/Node   |
|------------------------------|----------------------|---|--|---------------------|
| Gx/F                         | Diameter Round Trip  | node[x].messages.e2e_<br>_<domain>_[realm_]<br>Gx_CCR-I_3xxx.<br>qns_stat.success         | Success count of messages with return code matching 3XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-I_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-I_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                 | Policy Director     |
| Gx/A                         | Diameter Input Queue | node1.counters.[realm_]<br>Gx_CCR-I.qns_count   | Count of messages successfully sent to the policy engine                 | Policy Server (qns) |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-U_2001.<br>qns_stat.success          | Success message count for return code 2001                               | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-U_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching 2001 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-U_3xxx.<br>qns_stat.success          | Success count of messages with return code matching 3XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-U_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-U_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                 | Policy Director     |

| Appld/<br>Monitoring<br>Area | Category             | Statistic  | Description  | Availability/Node   |
|------------------------------|----------------------|--|--|---------------------|
| Gx/A                         | Diameter Input Queue | node1.counters. [realm_] Gx_CCR-U.<br>qns_count                                      | Count of messages successfully sent to the policy engine                 | Policy Server (qns) |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-U_2001.<br>qns_stat.success          | Success message count for return code 2001                               | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-U_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching 2001 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-U_3xxx.<br>qns_stat.success          | Success count of messages with return code matching 3XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-U_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-U_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                 | Policy Director     |
| Gx/A                         | Diameter Input Queue | node1.counters. [realm_] Gx_CCR-U.<br>qns_count                                      | Count of messages successfully sent to the policy engine                 | Policy Server (qns) |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-T_2001.<br>qns_stat.success          | Success message count for return code 2001                               | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages. e2e_<domain>_ [realm_] Gx_CCR-T_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching 2001 | Policy Director     |

| Appld/<br>Monitoring<br>Area | Category             | Statistic   | Description  | Availability/Node   |
|------------------------------|----------------------|---|--|---------------------|
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-T_3xxx.<br>qns_stat.success        | Success count of messages with return code matching 3XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-T_4xxx.<br>qns_stat.success        | Success count of messages with return code matching 4XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_CCR-T_5xxx.<br>qns_stat.success        | Success count of messages with return code matching 5XXX                 | Policy Director     |
| Gx/A                         | Diameter Input Queue | node1.counters. [realm_]<br>Gx_CCR-T.qns_count  | Count of messages successfully sent to the policy engine                 | Policy Server (qns) |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_RAR_2001.<br>qns_stat.success          | Success message count for return code 2001                               | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_RAR_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching 2001 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_RAR_3xxx.<br>qns_stat.success          | Success count of messages with return code matching 3XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_RAR_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                 | Policy Director     |
| Gx/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_RAR_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                 | Policy Director     |

| ApplId/<br>Monitoring<br>Area | Category              | Statistic  | Description  | Availability/Node   |
|-------------------------------|-----------------------|--|--|---------------------|
| Gx/F                          | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_]<br>Gx_RAR_timeout.<br>qns_stat.success    | Success timeout count for RAR message                                    | Policy Director     |
| Gx/A                          | Diameter Input Queue  | node1.counters. [realm_] Gx_RAA.qns_count  | Count of all messages sent to the policy engine                          | Policy Server (qns) |
| Gx/A                          | Diameter Input Queue  | node1.messages.<br>in_q_Gx_RAA.<br>qns_stat.error                                    | Count of messages failed to be sent to the policy engine                 | Policy Server (qns) |
| Gx/A                          | Diameter Input Queue  | node1.messages.<br>in_q_Gx_RAA.<br>qns_stat.success                                  | Count of messages successfully sent to the policy engine                 | Policy Server (qns) |
| Gx/E                          | Diameter Output Queue | node1.counters. [realm_] Gx_RAR.qns_count  | Count of messages successful sent to the Policy Director (LB)            | Policy Server (qns) |
| Rx/F                          | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_AAR_2001.<br>qns_stat.success          | Success message count for return code 2001                               | Policy Director     |
| Rx/F                          | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_AAR_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching 2001 | Policy Director     |
| Rx/F                          | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_AAR_3xxx.<br>qns_stat.success          | Success count of messages with return code matching 3XXX                 | Policy Director     |
| Rx/F                          | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_AAR_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                 | Policy Director     |
| Rx/F                          | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_AAR_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                 | Policy Director     |



| Appld/<br>Monitoring<br>Area | Category              | Statistic  | Description  | Availability/Node   |
|------------------------------|-----------------------|--|--|---------------------|
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_]<br>Rx_AAR_timeout.<br>qns_stat.success    | Success count of messages with return code matching 5XXX                                   | Policy Director     |
| Rx/A                         | Diameter Input Queue  | node1.counters. [realm_] Rx_RAA.qns_count  | Count of messages successful sent to the Policy Director (LB)                              | Policy Server (qns) |
| Rx/A                         | Diameter Input Queue  | node1.counters. [realm_] Rx_AAR_drop. qns_count                                      | Count of messages dropped due to exceedingSLA  | Policy Server (qns) |
| Rx/E                         | Diameter Output Queue | node1.counters. [realm_] Rx_AAA_2001.<br>qns_count                                   | Count of AAA messages with result-code = 2001 sent successfully to the PolicyDirector (LB) | Policy Server (qns) |
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_ASR_2001.<br>qns_stat.success          | Success message count for return code 2001   | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_ASR_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching2001                    | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_ASR_3xxx.<br>qns_stat.success          | Success count of messages with return code matching 3XXX                                   | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_ASR_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                                   | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_ASR_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                                   | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.<br>e2e_<domain>_[realm_] Rx_ASR_retry. qns_count                   | Retry count for ASR message  | Policy Server (qns) |

| Appld/<br>Monitoring<br>Area | Category              | Statistic   | Description   | Availability/Node   |
|------------------------------|-----------------------|---|---|---------------------|
| Rx/A                         | Diameter Input Queue  | node1.counters.[realm_]Rx_ASA_bypass.qns_count                              | Count of message that do not require processing by the policy engine    | Policy Server (qns) |
| Rx/A                         | Diameter Input Queue  | node1.counters.[realm_]Rx_ASA.qns_count                                     | Count of messages successfully sent to the policy engine                | Policy Server (qns) |
| Rx/A                         | Diameter Input Queue  | node1.counters.[realm_]Rx_ASA_drop.qns_count                                | Count of messages dropped due to exceedingSLA                           | Policy Server (qns) |
| Rx/F                         | Diameter Round Trip   | node[x].messages.e2e_<domain>_[realm_]Rx_RAR_2001.qns_stat.success          | Success message count for return code 2001                              | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.e2e_<domain>_[realm_]Rx_RAR_2001.qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching2001 | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.e2e_<domain>_[realm_]Rx_RAR_3xxx.qns_stat.success          | Success count of messages with return code matching 3XXX                | Policy Director     |
| Rx/F                         | Diameter Round Trip   | node[x].messages.e2e_<domain>_[realm_]Rx_RAR_4xxx.qns_stat.success          | Success count of messages with return code matching 4XXX                | Policy Director     |
| Rx/F                         | Diameter Input Queue  | node[x].messages.e2e_<domain>_[realm_]Rx_RAR_5xxx.qns_stat.success          | Success count of messages with return code matching 5XXX                | Policy Director     |
| Rx/A                         | Diameter Input Queue  | node1.counters.[realm_]Rx_RAA_bypass.qns_count                              | Count of message that do not require processing by the policy engine    | Policy Server (qns) |
| Rx/A                         | Diameter Output Queue | node1.counters.[realm_]Rx_RAA.qns_count                                     | Count of messages successfully sent to the policy engine                | Policy Server (qns) |

| Appld/<br>Monitoring<br>Area | Category             | Statistic  | Description   | Availability/Node   |
|------------------------------|----------------------|--|---|---------------------|
| Rx/A                         | Diameter Round Trip  | node1.counters.[realm_] Rx_RAA_drop.qns_count                                    | Count of messages dropped due to exceedingSLA                           | Policy Server (qns) |
| Rx/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Rx_STR_2001.<br>qns_stat.success          | Success message count for return code 2001                              | Policy Director     |
| Rx/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Rx_STR_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching2001 | Policy Director     |
| Rx/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Rx_STR_3xxx.<br>qns_stat.success          | Success count of messages with return code matching 3XXX                | Policy Director     |
| Rx/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Rx_STR_4xxx.<br>qns_stat.success          | Success count of messages with return code matching 4XXX                | Policy Director     |
| Rx/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Rx_STR_5xxx.<br>qns_stat.success          | Success count of messages with return code matching 5XXX                | Policy Director     |
| Rx/A                         | Diameter Input Queue | node1.counters.[realm_] Rx_STR.qns_count   | Count of messages successful sent to the policy engine                  | Policy Server (qns) |
| Rx/A                         | Diameter Input Queue | node1.counters.[realm_] Rx_STR_drop.qns_count                                    | Count of messages dropped due to exceedingSLA                           | Policy Server (qns) |
| Rx/A                         | Diameter Input Queue | node1.messages.in_q_Rx_STR.<br>qns_stat.success                                  | Count of messages successful sent to the policy engine                  | Policy Server (qns) |
| Rx/A                         | Diameter Input Queue | node1.messages.in_q_Rx_STR.qns_stat.<br>total_time_in_ms                         | Total milliseconds of messages successfully sent to the policy engine   | Policy Server (qns) |

| Appld/<br>Monitoring<br>Area | Category                | Statistic  | Description   | Availability/Node      |
|------------------------------|-------------------------|--|---|------------------------|
| Rx/D                         | Engine Message          | node1.messages.<br>diameter_Rx_STR.<br>qns_stat.success                                  | Success message count   | Policy Server<br>(qns) |
| Rx/D                         | Engine Message          | node1.messages.<br>diameter_Rx_STR.qns_stat.<br>total_time_in_ms                         | Total milliseconds of<br>successful messages  | Policy Server<br>(qns) |
| Rx/E                         | Diameter Input<br>Queue | node1.counters. [realm_]<br>Rx_STA_2001.<br>qns_count                                    | Count of STA messages<br>with result-code = 2001<br>sent successfully to the<br>PolicyDirector (LB) | Policy Server<br>(qns) |
| Sy/F                         | Diameter Round<br>Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SLR_2001.<br>qns_stat.success          | Success message count for<br>return code 2001   | Policy Director        |
| Sy/F                         | Diameter Round<br>Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SLR_2001.<br>qns_stat.total_time_in_ms | Total milliseconds of<br>successful messages with<br>return code matching 2001                      | Policy Director        |
| Sy/F                         | Diameter Round<br>Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SLR_3xxx.<br>qns_stat.success          | Success count of messages<br>with return code matching<br>3XXX                                      | Policy Director        |
| Sy/F                         | Diameter Round<br>Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SLR_4xxx.<br>qns_stat.success          | Success count of messages<br>with return code matching<br>4XXX                                      | Policy Director        |
| Sy/F                         | Diameter Round<br>Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SLR_5xxx.<br>qns_stat.success          | Success count of messages<br>with return code matching<br>5XXX                                      | Policy Director        |
| Sy/A                         | Diameter Input<br>Queue | node1.counters.<br>[realm_] Sy_SLR_bypass.<br>qns_count                                  | Count of message that do<br>not require processing by<br>the policy engine                          | Policy Server<br>(qns) |

| Appld/<br>Monitoring<br>Area | Category             | Statistic  | Description   | Availability/Node   |
|------------------------------|----------------------|--|---|---------------------|
| Sy/A                         | Diameter Input Queue | node1.counters.[realm_] Sy_SLR.qns_count                                     | Count of messages successful sent to the policy engine                  | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.counters.[realm_] Sy_SLR_drop.qns_count                                | Count of messages dropped due to exceedingSLA                           | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.messages.in_q_Sy_SLA.qns_stat.success                                  | Count of messages successfully sent to the policy engine                | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.messages.in_q_Sy_SLA.qns_stat.total_time_in_ms                         | Total milliseconds of messages successfully sent to the policy engine   | Policy Server (qns) |
| Sy/D                         | Engine Message       | node1.messages.diameter_Sy_SLA.qns_stat.success                              | Success message count   | Policy Server (qns) |
| Sy/D                         | Engine Message       | node1.messages.diameter_Sy_SLA.qns_stat.total_time_in_ms                     | Total milliseconds of successful messages                               | Policy Server (qns) |
| Sy/B                         | Diameter Action      | node1.actions.send.diameter_Sy_SLR.qns_stat.success                          | Success actions count   | Policy Server (qns) |
| Sy/B                         | Diameter Action      | node1.actions.send.diameter_Sy_SLR.qns_stat.total_time_in_ms                 | Total milliseconds of successful messages                               | Policy Server (qns) |
| Sy/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Sy_SNR_2001.qns_stat.success          | Success message count for return code 2001                              | Policy Director     |
| Sy/F                         | Diameter Round Trip  | node[x].messages.e2e_<domain>_[realm_] Sy_SNR_2001.qns_stat.total_time_in_ms | Total milliseconds of successful messages with return code matching2001 | Policy Director     |

| Appld/<br>Monitoring<br>Area | Category             | Statistic   | Description   | Availability/Node      |
|------------------------------|----------------------|---|---|------------------------|
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SNR_3xxx.<br>qns_stat.success           | Success count of messages<br>with return code matching<br>3XXX                | Policy Director        |
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SNR_4xxx.<br>qns_stat.success           | Success count of messages<br>with return code matching<br>4XXX                | Policy Director        |
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_SNR_5xxx.<br>qns_stat.success           | Success count of messages<br>with return code matching<br>5XXX                | Policy Director        |
| Sy/A                         | Diameter Input Queue | node1.counters. [realm_] Sy_SNR.qns_count   | Count of messages<br>successful sent to the<br>policy engine                  | Policy Server<br>(qns) |
| Sy/A                         | Diameter Input Queue | node1.counters.<br>[realm_] Sy_SNR_drop.<br>qns_count                                     | Count of messages<br>dropped due to<br>exceedingSLA                           | Policy Server<br>(qns) |
| Sy/A                         | Diameter Input Queue | node1.messages. in_q_<br>Sy_SNR. qns_stat.success   | Count of messages<br>successfully sent to the<br>policy engine                | Policy Server<br>(qns) |
| Sy/A                         | Diameter Input Queue | node1.messages.<br>in_q_Sy_SNR. qns_stat.<br>total_time_in_ms                             | Total milliseconds of<br>messages successfully sent<br>to the policy engine   | Policy Server<br>(qns) |
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_[realm_] Sy_STR_2001.<br>qns_stat.success               | Success message count for<br>return code 2001                                 | Policy Director        |
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_STR_2001.<br>qns_stat. total_time_in_ms | Total milliseconds of<br>successful messages with<br>return code matching2001 | Policy Director        |

| Appld/<br>Monitoring<br>Area | Category             | Statistic   | Description   | Availability/Node   |
|------------------------------|----------------------|---|---|---------------------|
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_STR_3xxx.<br>qns_stat.success | Success count of messages with return code matching 3XXX              | Policy Director     |
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_STR_4xxx.<br>qns_stat.success | Success count of messages with return code matching 4XXX              | Policy Director     |
| Sy/F                         | Diameter Round Trip  | node[x].messages.<br>e2e_<domain>_<br>[realm_] Sy_STR_5xxx.<br>qns_stat.success | Success count of messages with return code matching 5XXX              | Policy Director     |
| Sy/A                         | Diameter Input Queue | node1.counters.<br>[realm_] Sy_STA_bypass.<br>qns_count                         | Count of message that do not require processing by the policy engine  | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.counters. [realm_] Sy_STA.qns_count                                       | Count of messages successful sent to the policy engine                | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.counters. [realm_] Sy_STA_drop.qns_count                                  | Count of messages dropped due to exceedingSLA                         | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.messages.<br>in_q_Sy_STA.<br>qns_stat.success                             | Count of messages successfully sent to the policy engine              | Policy Server (qns) |
| Sy/A                         | Diameter Input Queue | node1.messages.<br>in_q_Sy_STA.<br>qns_stat.total_time_in_ms                    | Total milliseconds of messages successfully sent to the policy engine | Policy Server (qns) |
| Sy/D                         | Engine Message       | node1.messages.<br>diameter_Sy_STA.<br>qns_stat.success                         | Success message count   | Policy Server (qns) |
| Sy/D                         | Engine Message       | node1.messages.<br>diameter_Sy_STA.<br>qns_stat.total_time_in_ms                | Total milliseconds of successful messages                             | Policy Server (qns) |

| Appld/<br>Monitoring<br>Area | Category              | Statistic  | Description   | Availability/Node   |
|------------------------------|-----------------------|--|---|---------------------|
| Sy/B                         | Diameter Action       | node1.actions.send.diameter_Sy_STR.qns_stat.success          | Success actions count   | Policy Server (qns) |
| Sy/B                         | Diameter Action       | node1.actions.send.diameter_Sy_STR.qns_stat.total_time_in_ms | Total milliseconds of successful actions                        | Policy Server (qns) |
| Sy/E                         | Diameter Output Queue | node1.counters. [realm_] Sy_STR.qns_count                    | Count of messages successfully sent to the Policy Director (LB) | Policy Server (qns) |

## Example Statistics

### Sample CSV Files

The following list is a sample of the file names created in the /var/broadhop/stats directory on the control-A VM.

```
[root@control-A stats]# pwd
/var/broadhop/stats [root@control-A stats]# ls
bulk-control-A-201510131350.csv
bulk-control-A-201510131355.csv
bulk-control-A-201510131400.csv
bulk-control-A-201510131405.csv
bulk-control-A-201510131410.csv
bulk-control-A-201510131415.csv
bulk-control-A-201510131420.csv
bulk-control-A-201510131425.csv
bulk-control-A-201510131430.csv
bulk-control-A-201510131435.csv
bulk-control-A-201510131440.csv
bulk-control-A-201510131445.csv
bulk-control-A-201510131450.csv
bulk-control-A-201510131455.csv
bulk-control-A-201510131500.csv
bulk-control-A-201510131505.csv
bulk-control-A-201510131510.csv
bulk-control-A-201510131515.csv
bulk-control-A-201510131520.csv
bulk-control-A-201510131525.csv
bulk-control-A-201510131530.csv
bulk-control-A-201510131535.csv
bulk-control-A-201510131540.csv
bulk-control-A-201510131545.csv
bulk-control-A-201510131550.csv
bulk-control-A-201510131555.csv
bulk-control-A-201510131600.csv
bulk-control-A-201510131605.csv
bulk-control-A-201510131610.csv
bulk-control-A-201510131615.csv
```



```
bulk-control-A-201510131620.csv  
bulk-control-A-201510131625.csv  
bulk-control-A-201510131630.csv
```

## Sample Output

C,<VM\_name>,node1.actions.send.diameter\_Gx\_CCA-I.qns\_stat.success,19 where the <VM\_Name> indicates the VM where statistics has been collected.

A sample bulk statistics.csv file is shown below:

```
C,qns01,node1.actions.SaveSubscriberActionImpl.qns_stat.error,0  
C,qns01,node1.actions.SaveSubscriberActionImpl.qns_stat.success,6  
C,qns01,node1.actions.send.diameter_Gx_CCA-I.qns_stat.error,0  
C,qns01,node1.actions.send.diameter_Gx_CCA-I.qns_stat.success,19  
C,qns01,node1.actions.send.diameter_Gx_CCA-T.qns_stat.error,0  
C,qns01,node1.actions.send.diameter_Gx_CCA-T.qns_stat.success,9  
D,qns01,node1.messages.in_q_Gx_CCR-I.qns_stat.total_time_in_ms,14  
D,qns01,node1.messages.in_q_Gx_CCR-T.qns_stat.total_time_in_ms,2  
D,qns01,node1.messages.in_q_Gx_CCR-U.qns_stat.total_time_in_ms,1  
D,qns01,node1.messages.in_q_Gx_RAA.qns_stat.total_time_in_ms,0  
D,qns01,node1.messages.in_q_Sh_SNA.qns_stat.total_time_in_ms,2  
D,qns01,node1.messages.in_q_Sh_UDA.qns_stat.total_time_in_ms,0  
D,qns01,node1.messages.TimerExpired.qns_stat.total_time_in_ms,7244  
D,qns01,node1.spr.createSubscriber.qns_stat.total_time_in_ms,29  
D,qns01,node1.spr.deleteSubscriber.qns_stat.total_time_in_ms,40  
D,qns01,node1.spr.getSubscriber.qns_stat.total_time_in_ms,44  
D,qns01,node1.spr.updateSubscriber.qns_stat.total_time_in_ms,21  
G,lb02,node1.ldap.SITELDAP.qns_ldap_connection.MaximumAvailableConnections,10.0  
G,lb02,node1.ldap.SITELDAP.qns_ldap_connection.NumAvailableConnections,0.0  
G,lb02,node1.thread.gauge.daemon_thread_count,80.0  
G,lb02,node1.thread.gauge.live_thread_count,184.0
```

