End-to-End Flow Control

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2021.04.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

The Cloud Native Broadband Network Gateway (cnBNG) manages residential subscribers from different access planes in a centralized way. It accepts and identifies subscriber Control Plane (CP) traffic coming from multiple User Planes (UPs) associated with the CP. When the number of UPs scale, the amount of CP traffic coming from each UP, multiplies.

There are various scenarios where the traffic flow between the CP and UP must be regulated. This is to ensure that the CP attends all the service requests without service interruption. The scenarios that create burstiness or higher flow rates in the traffic flows are as follows:

- Power outage in a residential area
- Access network outage for a specific period
- UP catastrophic events like process crash, route processor reboots, chassis reload are some of the examples.

These scenarios generate sudden spike in traffic going to the CP. To handle these spikes in traffic, it is necessary to flow control and rate limit the CP ingress to ensure that service applications are not overwhelmed with these bursts. The End-to-End Flow Control feature optimizes flow control and rate limit of the traffic toward the CP ingres.

How it Works

This section describes how End-to-End Flow Control works in cnBNG.

There are two types of traffic that enter or exit the CP:. The control traffic that is responsible for subscriber session creation and the other type of traffic is , control traffic on already provisioned subscriber session.

- Control traffic that is responsible for subscriber session creation
- Control traffic that is already provisioned for a subscriber session

The following application infrastructure (App-Infra) features facilitates the cnBNG CP ingress packet flow control:

- Dispatcher
- Overload Control

Dispatcher

In the dispatcher, if the right dequeue rate is configured, the packets do not pile up in the PFCP queue. Also, the dequeue rate should be higher than the incoming rate from the UP. Because there is no packet segregation, all PFCP packets land into a single queue. Any rate control applied on this queue is per UPF PFCP packet rate control. It is not possible to control a particular type of packet per UPF. For example, DHCP release, PPPoE PADT, or keepalive failure notification packet cannot be controlled per UPF at the dispatcher queue.

The dispatcher queue size is configured to handle burst of packets. This functionality supports the following:

- Dedicated queue for each PFCP or N4 interface and GTPu interface for each UPF connected to the CP
- Configuration of size and flow control rate limits

Overload Control

Overload control is applied to a packet after it is released from a dispatcher. This creates a queue based on the packet type at the aggregate level across all UPF data. Because the overload control enables packet type based queues, rate control is applied for that type of packet at the aggregate level of all UPFs.

Special treatment of the packet is indirectly achieved by having different queue for a packet at overload control feature and aggregate of all UPF level.

The dispatcher supports the following catogories of virtual message groups:

- PFCP Keepalive messages between CP and UPF
- PFCP LCP keepalive failure notification messages
- PFCP response messages
- Session Report messages
- Other message types, which includes all GTPu and other messages which are not listed in different categories

The Overload Control feature provides aggregate queues for a message type coming from UPF functions. Group ids are supported for each message group and the message type is configurable for each group. When configured, a virtual queue is created for that message and treated based on the configured attributes for that group.

Example configuration

```
bng(config)# endpoint udp-proxy interface n4 overload-control msg-type ?
```

Possible completions:

all | lcpkeepalive | pfcpkeepalive | pfcpresponse | sessionreport

In the preceding configuration, for each message type, one virtual message queue is supported. For each queue, the size and rate limit can be configured. Based on the configuration, queue sizes and threshold is allocated for each queue.

For each message, the configured rate of packets are dequeued and sent to the CPF. For priority packets like, PFCP keepalives, dedicated queues are allocated such that they are not impacted with other queue sizes.

Based on the cluster capacity, specific values for each queue and message type must be configured. The values are adjusted based on capacity.

Limitations and Restrictions

The End-to-End Flow Control feature has the following limitations and restrictions:

- Flow control on the CP cannot prioritise packets. Therefore, if there is a congestion, critical packets like PFCP keepalive processing can get delayed.
- Session bring-down rate (DHCP release, PPPOE PADT, L2TP, CDN rate control) cannot be enforced using the CP flow control configuration. Also, UP does not have flow control of these packets. Therefore, solution level flow control for session disconnect triggers for all session types is not supported.
- Packet level flow control for DHCPv4 and DHCP/v6 Renew and DHCPv6 Relay forwarded messages is not supported.

- L2TP LAC and LNS FSOL rate control are not supported on the ASR 9000 UP in this release. The CP does not have rate control based on FSOL. Because PPPoE bring-up controls LAC, PPPoE FSOL rate control on ASR 9000 can be used to control LAC session bring-up.
- Dispatcher configuration changes implementation would require restarting of the CP.
- Flow control must be configured at the UP level for the following packets at the UPF. This ensures that the packet rate from UP to CP is controlled because CP cannot proivde per packet rate control, per UPF.
  - FSOL
  - Session delete notifications
  - LCP keepalive failure notifications
  - Session statistics report

Configuring End-to-End Flow Control

This section describes how to configure the End-to-End Flow Control features on Control Plane (CP).

The configuration involves the following procedures:

- Configuring Dispatcher for GTPu Interface, on page 5
- Configuring Dispatcher for N4 Interfaces, on page 4
- Configuring Overload Control for Message Types, on page 6

Configuring Dispatcher for N4 Interfaces

Use the following commands to configure the Dispatcher feature for N4 interfaces.

```
config
  instance instance_id
    endpoint udp-proxy
      interface n4 dispatcher { cache { true | false } |
      capactiy queue_capacity | count count |
      outbound { true | false } | rate-limit value |
      threshold value }
      commit
```

NOTES:

- **instance** *instance_id* : Configures multiple instances for the specified instance and enters instance sub-mode.
- **endpoint udp-proxy** : Configures parameters for the UDP-proxy endpoint and enters endpoint sub-mode.
- **interface n4 dispatcher { cache { true | false } | capactiy** *queue_capacity* **| count** *count* **| outbound { true | false } | rate-limit** *value*| **threshold** *threshold* **}** : Specifies dispatcher parameters for the N4 interface.
  - **cache { true | false }**: Enables (false) or disables (true) cache retransmission support. The default is **false**, which indicates that the cache retransmission support is enabled.

- **capacity** *queue_capacity* : Specifies the number of packets this queue holds.

Note Ensure that there is sufficient memory when configuring higher capactity queues.

- **count** *count* : Specifies the number of N4 queues to be created. Each queue is associated or dedicated to an UPF. For example, if the count is 2, two N4 queues are created and two UPs can be connected. One UP per queue.
- **outbound { true | false }**: Enables (true) or disables (false) queue support for outbound messages. The default is **false** for BNG.

Note Outbound flow control for BNG is not supported.

- **rate-limit** *per_second* : Specifies rate limit for each queue, that is, when packets are dequeued. The rate limit is defined in seconds.
- **threshold** *threshold* : Specifies size of the queue before packets are dropped.

Example

The following is a configuration example.

```
endpoint udp-proxy
  replicas 1
  nodes    2
  vip-ip 201.201.201.51
  interface n4
   sla response 150000
   dispatcher
    count      1
    capacity   500000
    outbound   true
    rate-limit 300
    cache      false
    threshold  950000
   exit
```

Configuring Dispatcher for GTPu Interface

Use the following commands to configure the Dispatcher feature for GTPu interfaces.

```
config
   instance instance_id
      endpoint udp-proxy
         interface gtpu dispatcher { cache { true | false } |
         capactiy queue_capacity | count count |
         outbound { true | false } | rate-limit value |
         threshold value }
         commit
```

NOTES:

- **instance** *instance_id* : Configures multiple instances for the specified instance and enters instance sub-mode.
- **endpoint udp-proxy** : Configures parameters for the UDP-proxy endpoint and enters endpoint sub-mode.
- **interface gtpu dispatcher { cache { true | false } | capactiy** *queue_capacity* **| count** *count* **| outbound { true | false } | rate-limit** *value*| **threshold** *threshold* **}**: Specifies dispatcher parameters for the GTPu interface.
  - **cache { true | false }**: Enables (false) or disables (true) cache retransmission support. The default is **false**, which indicates that the cache retransmission support is enabled.
  - **capacity** *queue_capacity* : Specifies the number of packets this queue holds.

  Note Ensure that there is sufficient memory when configuring higher capactity queues.

  - **count** *count* : Specifies the number of N4 queues to be created. Each queue is associated or dedicated to an UPF. For example, if the count is 2, two N4 queues are created and two UPs can be connected. One UP per queue.
  - **outbound { true | false }**: Enables (true) or disables (false) queue support for outbound messages. The default is **false** for BNG.

  Note Outbound flow control for BNG is not supported.

  - **rate-limit** *per_second* : Specifies rate limit for each queue, that is, when packets are dequeued. The rate limit is defined in seconds.
  - **threshold** *threshold* : Specifies size of the queue before packets are dropped.

Example

The following is a configuration example.

```
interface gtpu
   sla response 150000
   dispatcher
    count      1
    capacity   1000000
    outbound   true
    rate-limit 500
    cache      true
    threshold  950000
   exit
  exit
exit
exit
```

Configuring Overload Control for Message Types

Use the following commands to configure the Overload Control feature for all message types.

```
config
     overload-control msg-type { all | lcpkeepalive | pfcpkeepalive |
     pfcpresponse | sessionreport}
        msg-priority msg_priority  | rate-limit value|
        queue-size queue_size  | reject-threshold reject_threshold |
        pending-request pending_request  | discard-behavior { drop | true }
        commit
```

NOTES:

- **msg-type { all | lcpkeepalive | pfcpkeepalive | pfcpresponse | sessionreport }**: Configures overload control for the specified message type. The valid values are all, lcpkeepalive, pfcpkeepalive, and pfcpresponse, and sessionreport.
- **msg-priority** *msg_priority* : Specifies message priority. This keyword is not applicable in the BNG context.
- **rate-limit** *per_second*: Specifies rate limit for each queue, that is, when packets are dequeued.. The rate limit is defined in seconds.
- **queue-size** *queue_size* : Specifies the size of the queue to be created.
- **reject-threshold** *threshold_limit* : Specifies the percentage of the pending-request value.
- **pending-request** *pending_request* : Specifies the number of packets present in the queue at any time.
- **discard-behavior { drop | true }** : Specifies whether to drop or process the packets. In BNG, the default is **drop**.

Example

The following is a configuration example.

```
overload-control msg-type all
    rate-limit 13000 queue-size 200000 reject-threshold 95 pending-request 200000
   exit
   overload-control msg-type lcpkeepalive
    rate-limit 1100 queue-size 25000 reject-threshold 95 pending-request 25000
   exit
   overload-control msg-type sessionreport
    rate-limit 1000 queue-size 25000 reject-threshold 95 pending-request 25000
   exit
   overload-control msg-type pfcpkeepalive
    rate-limit 100 queue-size 1000 reject-threshold 95 pending-request 1000
   exit
   overload-control msg-type pfcpresponse
    rate-limit 4000 queue-size 25000 reject-threshold 95 pending-request 25000
   exit
  exit
```