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
The Flow Control feature supports the following functionalities: • Different rate limit and queue size for different UPFs • Packet priority and differential treatment to packets based on packet type or DSCP value of the packet • FSOL token mechanism	2022.03.0
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.

The traffic flow between CP and UP must be regulated to ensure that the CP attends all the service requests without service interruption.

The following scenarios create burstiness or higher flow rates in the traffic flows:

- Power outage in a residential area
- Access network outage for a specific period
- UP catastrophic events such as process crash, route processor reboots, and chassis reload

These scenarios generate sudden spike in traffic going to the CP. To handle these traffic spikes, 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 ingress.

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:

- 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, on page 2
- Overload Control, on page 3

Dispatcher

In the dispatcher, if you configure the right dequeue rate, the packets do not pile up in the PFCP queue. The dequeue rate must be higher than the incoming rate from the UP.

All PFCP packets land into a single queue because there is no packet segregation. Any rate control that is 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 configuration handles the 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 control plane
- Configuration of queue size and flow control rate limits

Different Rate Limit and Queue Size Support

For N4 or GTPu dispatcher, the flow control feature supports different rate limits and queue sizes for different UPFs. Depending on the UPF's capability, the rates and queue sizes vary for all UPFs.

The rate limit and queue size are configurable for different UPFs. You can configure the flow control group under the endpoint udp-proxy configuration. You can associate the flow control group in the user-plane configuration.

Packet Priority and Differential Treatment to Packets

cnBNG supports packet priority and differential treatment to packets based on the packet type or DSCP value in the packet. When the system is congested or queues are full, certain incoming packets such as PFCP heartbeats, session reports, and subscriber redundancy group notification messages are treated with high priority. These packets will be delivered to the destination without delay and will not be dropped if the queues are full.

The PFCP Heartbeats, PFCP Association Update or Release, and PFCP Node Report packets that cannot be rate controlled and dropped will be bypassed from the dispatcher queues and overload control queues.

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 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 queues 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 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 each message type and treated based on the configured attributes for that group. For each queue, the size and rate limit can be configured.

For each message, the configured rate of packets are dequeued and sent to the CPF. For priority packets such as PFCP keepalives, dedicated queues are allocated so 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 the capacity.

FSOL Token Mechanism

The FSOL token mechanism is a protection method to control the maximum inflight transactions on the control plane at each FSOL pod. This mechanism addresses higher packet rates and works along with UDP proxy flow control to protect the control plane. The UDP proxy flow control configuration acts as the first-level check in the control plane to control incoming GTPU and PFCP message rates. Token mechanism does not support rate control.

To receover from token leakage or stale token usage, FSOL supports the use of the action command **subscriber reset-token { dhcp | pppoe }** to reset the in-use token to zero.

FSOL token mechanism supports the following functionalities:

- Token mechanism is applicable for IPoE, PPPoE, and L2TP session types.

  This feature does not support per UPF token.

- DHCP and PPPoE have separate token configurations. DHCPv6 over PPPoE consumes the DHCP token.
- Only transactions that deal with IPCs with multiple pods or that result in interaction with external entities consume the token. The external entities can be Policy plane interaction, N4 PFCP Establishment, N4 PFCP Modification. Handling of DHCPv4 Renew, DHCPv4Rebind, DHCPv4 Information and DHCPv6 Renew, DHCPv6 Rebind, DHCPv6 Information-Request packets will not consume the token.
- Acquires the token and releases across transactions.
- A Discover storm without any request packet for the offered IPs can consume all the tokens. Since the tokens that are acquired during Discover processing are released only after processing the request from CPE. If CPE does not send the request, the token is released only after the temporary lease expiry (around 1 minute).
- IPoE lease expiry handling consumes the token. If the token is not available, the lease expiry handling is postponed with an exponential back off up to 8 minutes.
- If the token is not available when the hold timer expires, then the hold timer is extended for a minute.
- DHCP admin clear consumes the token. If the token is not available, the session will not get cleared.
- Internal failure handling consumes the token. If the tokens are not available, then failure handling proceeds uninterrupted as failure cannot be cached.
- FSOL token will not be consumed in the following scenarios:
  - Session manager admin clear
  - Disconnect message from session manager (due to pod or PPPoE session disconnect)
  - Notify conflict handling, failure callback handling for CDL update, and CP reconciliation
- Caches packets if tokens are unavailable during release packet handling.

Limitations and Restrictions

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

- 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 DHCPv6 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 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 provide 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 feature on Control Plane (CP).

The configuration involves the following procedures:

- Configuring Dispatcher for GTPu Interface, on page 5
- Configuring Dispatcher for N4 Interface, on page 7
- Configuring Overload Control for Message Types, on page 9
- Associating Flow Control Group in User Plane, on page 10
- Configuring Token for FSOL Pod, on page 10

Configuring Dispatcher for GTPu Interface

To configure dispatcher for GTPu interface, use the following commands:

```
config
  instance instance-id instance_id
    endpoint udp-proxy
      interface gtpu dispatcher { cache { true | false } |
      capacity queue_capacity | count queue_count |
      flowctrl-group group_name { capacity inbound_queue_size |
```

```
outbound-capacity outbound_queue_size | outbound-rate-limit outbound_rate_limit
| rate-limit inbound_rate_limit } |
        outbound { true | false } | rate-limit rate_limit |
        threshold threshold_value }
        exit
```

NOTES:

- **instance** *instance_id*: Configure multiple instances for the specified instance and enters the instance sub-mode.
- **endpoint udp-proxy**: Configure parameters for the UDP-proxy endpoint and enters the endpoint sub-mode.
- **interface gtpu dispatcher { cache { true | false } | capacity** *queue_capacity* **| count** *count* **| flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **} | outbound { true | false } | rate-limit** *value* **| threshold** *threshold_value* **}**: Specify the dispatcher parameters for the GTPu interface.
  - **cache { true | false }**: Enable (false) or disable (true) cache retransmission support. The default value **false** indicates that the cache retransmission support is enabled.
  - **capacity** *queue_capacity*: Specify the number of packets that this queue holds.

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

  - **count** *queue_count*: Specify 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.
  - **flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **}**: Specify the queue size and rate limit for the specified flow control group.

Note The flow control group name must be the same for N4 and GTPu interfaces for a given group.

    - **capacity** *inbound_queue_size*: Specify the capacity of inbound queue.
    - **outbound-capacity** *outbound_queue_size*: Specify the capacity of outbound queue.
    - **outbound-rate-limit** *outbound_rate_limit*: Specify the rate limit for outbound queue.
    - **rate-limit** *inbound_rate_limit*: Specify the rate limit for inbound queue.

  - **outbound { true | false }**: Enable (true) or disable (false) queue support for outbound messages. Default value: **false**.

Note Outbound flow control for BNG is not supported.

- **rate-limit** *rate_limit*: Specify the rate limit for each queue, that is, when packets are dequeued. The rate limit is defined in seconds.
- **threshold** *threshold_value*: Specify the queue size 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
    flowctrl-group group1
     capacity 2000
     rate-limit 200
    exit
   exit
  exit
exit
exit
```

Configuring Dispatcher for N4 Interface

To configure dispatcher for N4 interface, use the following commands:

```
config
   instance instance_id
      endpoint udp-proxy
         interface n4 dispatcher { cache { true | false } |
         capacity queue_capacity | count queue_count |
         flowctrl-group group_name { capacity inbound_queue_size |
outbound-capacity outbound_queue_size | outbound-rate-limit outbound_rate_limit
| rate-limit inbound_rate_limit } |
         outbound { true | false } | rate-limit rate_limit |
         threshold threshold_value }
         exit
```

NOTES:

- **instance** *instance_id*: Configure multiple instances for the specified instance and enters the instance sub-mode.
- **endpoint udp-proxy**: Configure parameters for the UDP-proxy endpoint and enters the endpoint sub-mode.
- **interface n4 dispatcher { cache { true | false } | capacity** *queue_capacity* **| count** *count* **flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **} | outbound { true | false } | rate-limit** *value* **| threshold** *threshold_value* **}**: Specify dispatcher parameters for the N4 interface.
  - **cache { true | false }**: Enable (false) or disable (true) cache retransmission support. The default value **false** indicates that the cache retransmission support is enabled.

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

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

- **count** *queue_count*: Specify 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.
- **flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **}**: Specify the queue size and rate limit for the specified flow control group.

Note The flow control group name must be the same for N4 and GTPu interfaces for a given group.

  - **capacity** *inbound_queue_size*: Specify the capacity of inbound queue.
  - **outbound-capacity** *outbound_queue_size*: Specify the capacity of outbound queue.
  - **outbound-rate-limit** *outbound_rate_limit*: Specify the rate limit for outbound queue.
  - **rate-limit** *inbound_rate_limit*: Specify the rate limit for inbound queue.

- **outbound { true | false }**: Enable (true) or disable (false) queue support for outbound messages. Default value: **false**.

Note Outbound flow control for BNG is not supported.

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

Example

The following is an example configuration.

```
endpoint udp-proxy
  replicas 1
  nodes 2
  vip-ip 201.201.201.51
  vip-ipv6 2001::10:1:39.191
  interface n4
   sla response 150000
   dispatcher
    count 1
    capacity 500000
    outbound true
    rate-limit 300
    cache false
```

```
    threshold 950000
    flowctrl-group group1
     capacity 1000
     rate-limit 100
    exit
   exit
```

Configuring Overload Control for Message Types

To configure overload control for all message types, use the following commands:

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

NOTES:

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

Associating Flow Control Group in User Plane

To associate the flow control group in the user plane, use the following sample configuration:

```
config
   user-plane user_plane_name
      flowctrl-group group_name
      exit
```

NOTES:

- **user-plane** *user_plane_name*: Specify the User Plane (UP) name and enter UP Configuration mode.
- **flowctrl-group** *group_name*: Specify the name of the flow control group to be associated in the user plane.

Configuring Token for FSOL Pod

To configure the FSOL token mechanism, use the following sample configuration:

```
config
   subscriber token { dhcp | pppoe } token_count
   exit
```

NOTES:

- **subscriber token { dhcp | pppoe }** *token_count*: Set the maximum token available for the FSOL pod. *token_count* is cumulative across instances. For example, if there are 4 DHCP pods and DHCP token is set as 2000, then 500 tokens will be assigned for each pod.
  - **dhcp**: Set the DHCP pod token count.
  - **pppoe**: Set the PPPoE pod token count.