BGP over QUIC

draft-retana-idr-bgp-quic-05

Alvaro Retana, Yingzhen Qu, Jeff Haas
Shuanglong Chen, Jeff Tantsura
Memory Refresh – Why BGP over QUIC

• High Resiliency
  With multi-channel channel BGP support:
  each channel can be reset independently without impacting other channels. draft-ietf-idr-bgp-multisession defines BGP over multiple TCP sessions but achieved very limited implementation and deployment.
  Allowing each channel to transmit data concurrently. This helps avoid head-of-line blocking and improve overall performance.

• Enhanced Security
  QUIC provides transport security with build-in encryption and authentication. This helps protect against spoofing attacks etc.

• Connection Migration
  QUIC supports seamless migration of connections between different network interfaces or IP addresses.
Terminologies

- **BoQ, Multi-channel BGP using QUIC**: Running the BGP protocol over multiple QUIC streams as defined in this document.
- **QUIC connection**: A transport-layer connection between two endpoints using QUIC [RFC9000].
- **QUIC streams**: A bidirectional or unidirectional byte stream provided by the QUIC transport [RFC9000].
- **BGP channel**: Instance of BGP protocol state machine mapped to specific QUIC stream.
- **BGP control channel**: A channel dedicated to transmitting the session control data, which is implemented as a bidirectional stream.
- **BGP function channel**: BGP per AFI/SAFI channel, which is implemented asymmetrically as unidirectional streams.
Multi-Channel BGP over QUIC

A
- A0 Control
- A1 IPv4 Unicast Send
- A IPv4 Unicast Rcv
- A3 BGP-LS Send

B
- B0 Control
- B1 IPv4 Unicast Send
- B IPv4 Unicast Rcv
- B BGP-LS Rcv

OPEN [0]  
OPEN [0]  
Keepalive [0]  
Keepalive 0  
OPEN [1]  
OPEN [1]  
Keepalive [1]  
OPEN [3]  
Keepalive [3]  
Established

update

update

update

update

OPEN

OPEN

OPEN

OPEN

IETF 117, San Francisco, July 2023
BoQ FSM
BGP FSM – RFC 4271

Control Channel – FSM
Control Channel – Connect State

Text used to generate the UML diagram:

```plantuml
state Connect {
  state DelayOpen <<choice>>
  connect --> DelayOpen
  DelayOpen --> delayopen_false
  DelayOpen --> delayopen_true
  delayopen_true --> delayopen_timer:16,17,29 - [delayopen]
  delayopen_false --> delayopen_timer:16,17,29

  state holdtimer <<choice>>
  delayopen_timer --> holdtimer: 20 - [no_delayopen][stopdelaytimer][open][keepalive]
  holdtimer --> holdtimer_zero
  holdtimer --> holdtimer_nonzero

  delayopen_timer --> OpenSent: 12 - [send_open]
  delayopen_false --> OpenSent: 16,17,29 - [nodelayopen][send_open]
  delayopen_timer --> Active: 18 - [connectretry]
  holdtimer_zero --> OpenConfirm : [holdtimer_zero]
  holdtimer_nonzero --> OpenConfirm : [holdtimer_nonzero]
  Connect --> Idle: 30 - [restart]
  Connect --> Idle: 21,22 - [sendnotificationwithoutopen][restart][reconnect]
  delayopen_timer --> Idle: 24 - [restart]
  delayopen_false --> Idle: 24 - [restart][reconnect]
  Connect --> Connect: 1,3-7,14,15
  Connect --> Idle: 2 - [stop]
  Connect --> Connect: 9 - [retry]
  Connect --> Idle: 8,10,11,13,19,23,25-28 [restart]
}
```

Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams (plantuml.com)
Control Channel – Established State
Function Channel

Sending FSM:

- Connect
- OpenSent
- OpenConfirm
- Established
- Terminating

Receiving FSM:

- Active
- OpenConfirm
- Established
- Terminating
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Initializes all BGP resources for the peer connection, sets ConnectRetryCounter to zero, starts the ConnectRetryTimer with the initial value, listens for a connection that may be initiated by the remote BGP peer</td>
</tr>
<tr>
<td>initiate_connection</td>
<td>Initiates a QUIC connection to the other BGP peer if the local system is configured as BoQ client or any</td>
</tr>
<tr>
<td>stop</td>
<td>Drops the QUIC connection, releases all BGP resources, sets ConnectRetryCounter to zero, stops the ConnectRetryTimer and sets ConnectRetryTimer to zero</td>
</tr>
<tr>
<td>retry</td>
<td>Drops the QUIC connection, restarts the ConnectRetryTimer, stops the DelayOpenTimer and resets the timer to zero, initiates a QUIC connection to the other BGP peer, continues to listen for a connection that may be initiated by the remote BGP peer</td>
</tr>
<tr>
<td>send_open</td>
<td>Sends an OPEN message to its peer, sets the HoldTimer to a large value, stops the ConnectRetryTimer (if running) and sets the ConnectRetryTimer to zero</td>
</tr>
<tr>
<td>delayopen</td>
<td>Stops the ConnectRetryTimer (if running) and sets the ConnectRetryTimer to zero, sets the DelayOpenTimer to the initial value</td>
</tr>
<tr>
<td>no_delayopen</td>
<td>Stops the ConnectRetryTimer (if running) and sets the ConnectRetryTimer to zero, completes BGP initialization</td>
</tr>
<tr>
<td>connectretry</td>
<td>Restarts the ConnectRetryTimer with the initial value, stops the DelayOpenTimer and resets its value to zero, continues to listen for a connection that may be initiated by the remote BGP peer</td>
</tr>
<tr>
<td>restart</td>
<td>Stops the ConnectRetryTimer to zero, if the DelayOpenTimer is running, stops and resets the DelayOpenTimer (sets to zero) drops the QUIC connection, releases all BGP resources</td>
</tr>
<tr>
<td>stopdelaytimer</td>
<td>Stops and clears the DelayOpenTimer (sets the value to zero)</td>
</tr>
<tr>
<td>open</td>
<td>Sends an OPEN message, keeps a KEEPALIVE message</td>
</tr>
<tr>
<td>keepalive</td>
<td>Sends a KEEPALIVE message, starts the KeepaliveTimer with the initial value, resets the HoldTimer to the negotiated value</td>
</tr>
<tr>
<td>holdtimer_nonzero</td>
<td>Starts the KeepaliveTimer with the initial value, resets the HoldTimer to zero, stops the KeepaliveTimer, resets the HoldTimer value to zero</td>
</tr>
<tr>
<td>holdtimer_zero</td>
<td>Resets the KeepaliveTimer, resets the HoldTimer value to zero</td>
</tr>
<tr>
<td>snotifyWithoutOpen</td>
<td>(Optionally) If the snotifyWithoutOpen attribute is set to TRUE, then the local system first sends a NOTIFICATION message with the appropriate error code</td>
</tr>
<tr>
<td>reconnect</td>
<td>Increments the ConnectRetryCounter by 1 (Optionally) performs peer oscillation damping if the DampPeerOscillations attribute is set to TRUE</td>
</tr>
<tr>
<td>clear_connectretrycounter</td>
<td>Sets ConnectRetryCounter to zero, sets ConnectRetryTimer with initial value, initiates a QUIC connection to the other BGP peer, continues to listen for a QUIC connection that may be initiated by a remote BGP peer</td>
</tr>
<tr>
<td>connectretrytimer</td>
<td>Restarts the ConnectRetryTimer (with initial value), initiates a QUIC connection to the other BGP peer, continues to listen for a QUIC connection that may be initiated by a remote BGP peer, completes the BGP initialization</td>
</tr>
<tr>
<td>delayopen</td>
<td>Sets the ConnectRetryTimer to zero, stops and clears the DelayOpenTimer (set to zero), completes the BGP initialization</td>
</tr>
<tr>
<td>cease</td>
<td>Sends the NOTIFICATION with a Cease</td>
</tr>
<tr>
<td>holdtimer</td>
<td>Sends a NOTIFICATION with the error code Hold Timer Expired</td>
</tr>
<tr>
<td>receive_open</td>
<td>Resets the DelayOpenTimer to zero, sets the BGP ConnectRetryTimer to zero</td>
</tr>
<tr>
<td>notification</td>
<td>Sends a NOTIFICATION message with the appropriate error code</td>
</tr>
<tr>
<td>keepaliveTimer</td>
<td>Sends a KEEPALIVE message, restarts the KeepaliveTimer</td>
</tr>
<tr>
<td>restart_holidtimer</td>
<td>Restarts the HoldTimer</td>
</tr>
<tr>
<td>deleteroutes</td>
<td>Deletes all routes associated with the connection</td>
</tr>
</tbody>
</table>
Marker for End-of-RIB

• The End-of-RIB marker was defined in RFC4724 for the purpose of BGP graceful restart, and it was recommended because it was helpful for routing convergence in general.

• Partial deployment defeats the purpose of this marker.

• A BoQ speaker SHOULD always send the End-of-RIB marker to its peer to indicate the completion of the initial routing routing update.
Next Steps

• GitHub Repository: 
  https://github.com/Yingzhen-ietf/BGP-QUIC
• Reviews and comments are welcome
• Request WG adoption?