Internet Draft
Document: draft-hares-bgp-statemt-01.txt

Expires: December 2002

S. Hares
NextHop
Technologies,
Inc.
June 2002

BGP-4 Finite State Machine Table

Status of this Memo

Abstract

This document describes the BGP-4 Finite State Machine in detail. This Finite State machine (FSM) describes how the 27 events operate on the 6 states described in the BGP-4 standard. These 27 events include administrative events, timer events, TCP connectivity events and BGP message events.

This document is standards track to accompany the BGP-4 [1] standard as a standard document. This description augments the BGP-4 FSM description (section 8 of BGP-4 draft).

Table of Contents

Abstractàà.......1

1.0 BGP FSM Table Overviewàààààààààààààààààààààààààààà..3

2.0 BGP FSM Tableàààààààààààààààààààààààààààààààààààààà..3

2.1 Part A - Administrative eventsàààààààààààààààààààà..4

2.2 Part B - Timer eventsàààààààààààààààààààààààààààààà..5

2.3 Part C - TCP Eventsàààààààààààààààààààààààààààààààà..6

2.3.1 Transport Events.............................6
2.3.2 Sequences of TCP interchangesàààààààààààààà...7
2.3.3 Flags to track TCP exchangeààààààààààààààààà..8
2.3.4 Processing based on Transport flags in
Connect and Active states....................8

2.4 Part D - BGP Message related eventsààààààààààààààà..9

2.4.1 Collision Detect processing in Open Confirm..12
2.4.2 Collision Detect processing in Open Confirm..12

3.0 Actions in state table   àààààààààààààààààààààààààà.13

3.1 Actions for Initializationààààààààààààààààààààààààà.13

3.2 Actions for Disconnectionàààààààààààààààààààààààààà.14

4.0 Security Considerationsààààààààààààààààààààààààààà..17

5.0 Referencesààà.18

6.0 Author's Addressesàààààààààààààààààààààààààààààààààà.18

1.0 BGP FSM Table Overview

The table below gives the BGP State machine in table format. There are 6 states in the state table: Idle, Connect, Open Sent, Active, Open Confirm, and Established. The states are listed as columns in the table.

Each row of the table focuses on the state transitions when a single event occurs. The event is listed as the row header

In each box of the table, the first entry in the box is the new state. The second entry is the set of actions.   Each action is either specified (such as Ignore) or it given a Letter (such as B) that is described at the end of the table.

For example, if an manual start event occurs in the Idle state, the local system performs the actions listed under ôAö and goes to the Connect state.   The actions under the letter A are:

Action A

1) Initialize all BGP resources
2) ConnectRetryCnt set to 0
3) Start Connect retry timer with initial value
4) Initiate transport connection to BGP peer
5) Listen for connection set-up by remote BGP peer

If the bgp persistent peer oscillation protection is not enabled, the following events will not occur in this state machine:

Event6: Automatic start with bgp flap stop on
Event8: Idle Hold timer expires

Symbols within the Table:

(opt) - Denotes an optional event which MAY be implemented within a BGP implementation.

Collision Detect - Indicates the checks based on section 6.8 of the bgp-4 draft which lists the rules for detecting a collision of BGP connections. After these rules have been followed, if this connection is to be dumped then the "dumped" function is followed". If this connection is not to be dumped, then the Keep clause is followed.

See peer damp - Indicates that the operations in this state are covered in the BGP Peer Restart Backoff
Mechanisms
[draft-ietf-hares-backoff-01.txt]

TP - Transport
TP IND - Transport Indication
TP REQ - Transport Connection Request

TP REQ ACK - Transport Request Acknowledgement
Estab. - Establish state

Process based on - The exact processing of Transport messages in
starting
TP flags(2.3.4) BGP connection are described in section 2.3.4.

2.0 BGP Finite State Machine

2.1 Part A - Administrative events

#	Event	Idle	Connect	Active	Open sent	Open Confirm	Estab.
1	Manual Start	Connect /A	Connect /Ignore	Active/ Ignore	Open Sent /Ignore	Open Confirm/ Ignore	Estab./ Ignore
2	Manual Stop	Idle/ Ignore	Idle/ Z	Idle/ Z	Idle/ S	Idle/ S	Idle/ C
3	Auto start (opt)	Connect/ A	Connect /Ignore	Active/ Ignore	Open Sent / Ignore	Open Confirm/ Ignore	Estab./ Ignore
4	Manual start & passive (opt)	Active/ B	Connect/ Ignore	Active/ Ignore	Open Sent/ Ignore	Open Confirm /Ignore	Estab./ Ignore
5	Auto Start & passive (opt)	Active/ B	Connect /Ignore	Active /Ignore	Open Sent/ Ignore	Open Confirm/ Ignore	Estab./ Ignore
6	Auto start & bgp flap stop on (opt) [note 1]	See peer damp draft/ F	Connect /Ignore	Active /Ignore	Open Sent/ Ignore	Open Confirm/ Ignore	Estab. /Ignore
7	Auto stop (opt)	Idle/ Ignore	Idle / D	Idle / D	Idle/ C	Idle/ C	Idle/ C

2.2 Part B - Timer events

#	Event	Idle	Connect	Active	Open	Open	Estab
					sent	Confirm	lished
8	Idle Hold timer expires (opt)	See bgp backoff draft	Idle / D	Idle / D	Idle / E	Idle / E	Idle / E
9	Connect Retry timer expires	Idle/ V	Connect/ X	Connect/ F	Idle / E	Idle / E	Idle/ E
10	Hold Timer expires	Idle/ V	Idle / D	Idle / D	Idle / K	Idle / K	Idle / M
11	Keep-alive Timer expires	Idle / V	Idle / D	Idle / D	Idle / E	Open Confirm/ Q	Established/ Q
12	Delay BGP Open timer (opt)	Idle/ V	Open Sent/ H	Open Sent/ H	Idle/ E	Idle/ E	Idle / E

2.3 Part C - Transport (TP) events

2.3.1) Transport Events

#	Event	Idle	CONNECT	ACTIVE	OPEN Sent	OPEN CONFIRM	Estab.
13	TP IND and valid peering	Idle/ V	Connect/ Process based on TP flags flags (2.3.4)	Active/ Process based on TP flags flags (2.3.4)	Open Sent/ Track 2nd TP session	Open Confirm/ Track 2nd TP session	Estab./ Track 2nd TP session
14	TP IND and invalid peering	Idle/ V	Connect/ L	Active/ L	Open Sent/ Ignore	Open confirm/ Ignore	Estab./ Ignore
15	TP REQ-ACK	Idle/ V	Process based on TP flags (2.3.4) If TP connects: Open Sent /H If TP connect timer delay / ZZ	Process based on TP flags & Open flags (2.3.4) If TP connects: Open Sent / H If TP connect & timer delay /ZZ	Open Sent/ Track 2nd TP session	Open Confirm/ Track 2nd TP session	Estab. / Track 2nd TP session
16	TCP Connect Confirm	Idle/ V	Process based on TP flags (2.3.4)	Process based on TP flags (2.3.4)	Open Sent/ Ignore	Open Confirm /Ignore	Estab. /Ignore

```
                    If Open    If open
                   timer delay timer delay
                    /ZZ        /ZZ
                    else:      else:
                    OpenSent   Open Sent
                     /H         /H
  -------------------------------------------------------------------

  17 TP        Idle/  Idle/    Idle /   Active/  Idle /   Idle /
     Connect    V      D        Y       O         Y        T
     FAILS
     (RCV TCP FIN
     or timeout)
 -------------------------------------------------------------------
```

2.3.2 Sequences of Transport

BGP can run over any Transport protocol. Examples of transport protocols BGP can run over is TCP or STCP[6]. The BGP FSM has the following transport events:

```
     Event 13: Transport Connection Indication and valid source and
               destination
     Event 14: Transport Connection INdication and invalid source and
               destination
     Event 15: Transport Connection Acknowledgemen (Received)
     Event 16: Transport Connection Confirmation (received)
     Event 17: Transport Connection Failed
```

For TCP the following messages cause each sequence:

```
    Event 13:   TCP SYN received by local peer and valid source IP
                address and Transport Port #, valid destination IP
                address and Transport Port #.

    Event 14:   TCP SYN received by local peer with either invalid
                source IP address or transport port # or invalid
destination
                IP address and Transport Port #.
```

```
        Event 15:   TCP SYN, ACK pair received by local peer from remote peer

        Event 16:   TCP ACK received from remote peer

        Event 17:   TCP FIN received from remote peer or an indication of
Transport
                    Timeout
```

Each of these events is illustrated in the following 4 examples for TCP.

1) Sequence 1: Local system initiates Transport connection

```
Received from                   Sent from
Remote BGP Peer                 Local BGP Peer
===============                ================
                        <----- Tranport Request
                               (within Action A or B)
                                [TCP: Syn]
Tranport
  Connection  ----->
  Acknowledgement
   (Event 15)
  [TCP: SYN, ACK]
                        <----  Transport Confirmed
                               [TCP: ACK]
```

2) Sequence 2: Remote system initiated Transport connection

```
Received from                  Sent from
Remote BGP Peer                Local BGP peer
================               ================
Transport
  connection
  Indication ---->
   (Event 13)           <----- Tranport
[TCP: SYN]                     connection
                               Acknowledgement
                               (Action
                                [TCP: Syn, ACK]
```

```
     Transport
      Connection--->
      Confirm [Event 16]
      [For TCP: ACK]
```

3) Sequence 3: Competing Transport Requests

```
     Received from                  Sent from
     Remote BGP Peer                Local BGP peer
     ================               ================
                              <----- Tranport Request
                                    (within Action A or B)
                                     [for TCP: Syn]
     Tranport
       Indication
       & valid port  ----->
        (Event 14)
       [for TCP: SYN]         <------ Transport Connection Acknowledgement
                                      [for TCP: Syn, Ack]

     Transport
       Confirm      ------>
        (Event 15)
        [TCP: ACK]
```

4) Sequence 4: Rejected Transport connection

```
     Received from                  Sent from
     Remote BGP Peer                Local BGP peer
     ================               ================
                          <----- Tranport Request
                                    (within Action A or B)
                                     [TCP: Syn]
     Tranport
       Connection  ----->
       Reject
        (Event 17)
       [TCP: FIN]
                             <----  Transport Rejection Confirmed
                                    [TCP: FIN-ACK]
```

```
5) Sequence 5: Failed Transport connection via timer
                        <----- Tranport Request
                               (within Action A or B)
                                [TCP: Syn]
          [Timeout]
```

2.3.3 Flags to track Transport exchange

The list below is a list of flags for sub-state processing in Connect and Active for the Transport. These substates may be implemented as flags or sub-states

Tranport processing sub-states
================================
1) Await Transport Indication (Await TP IND)
2) sent Transport Connection Request (Sent TP REQ)
3) Await Transport Connection Confirmed (Await TCP CONF)
4) Null - set if no sub-state processing

Transport Flags relating to the BGP Open status in Active and Connect

Configuration:

1) Passive TCP connection - wait for other side to start TCP connection
2) Delay Open by timer - After establishing a TCP sessino, delay sending open until timer expires

Status flag:

1) TCP connected

2.3.4 Processing based on Transport flags in Connect and Active state

This table has entries with 3 parts: State/ substate / action.
For example the entry:

```
 Connect
/RCV TP
 IND
/aa
```

has a transition to: Connect State, RCV TP IND substate, and action aa.
The sub-state processing is based on the sub-states of the

	Connect sub-states			
	#1	#2	#3	#4
# Event	Await TP IND [TCP SYN]	Sent TP REQ [TCP SYN]	Await TP CONF [TCP:ACK]	null
13 RCV TP IND & valid peering	Connect /Await TP CONF /aa	Connect /Await TP CONF /ab	Connect /Await TP CONF /aa	Connect /null /ignore
15 RCV TP CONACK [TCP: SYN,ACK]	Connect /Sent TP REQ /ab	Connect /null /ac	Connect /null /ac	Connect /null /ignore
16 Rcv TP Connect Confirm	Connect /Await TP IND /ignore	Connect /Sent TP REQ /ab	Connect /null /ac	Connect /null /ac

	Active sub-states			
	#1	#2	#3	#4
# Event	Await TP IND [TCP SYN]	Sent TP REQ [TCP SYN]	Await TP CONF [TCP:ACK]	null
13 RCV TP IND & valid peering	Active /Await TP CONF /aa	Active /Await TP CONF /ab	Active /Await TP CONF /aa	Active /null /ignore
15 RCV	Active	Active	Active	Active

```
    TP CONACK|/Sent     /null    /null       /null
      [TCP:  |TP REQ
     SYN,ACK]|/ab       /ac      /ac         /ignore
    ---------------------------------------------------
    16 Rcv   |Connect  Connect   Connect      Connect
       TP    |/Await   /Sent     /null        /null
      Connect|TP IND   TP REQ
      Confirm|/ignore  /ab       /ac          /ac
    --------------------------------------------------
```

```
       Action aa)
           1) send TP Connect Acknowledge [TCP SYN, ACK]

        Action ab)
           1) send TP REQ [TCP SYN]

        Action ac)
           1) send TP Connection Confirmation [TCP ACK]
           2) if bgp open delay flag set:
                  -  clear connect retry timer
                  -  set BGP delay timer
              else:
                  - Set TCP Connected flag
```

2.4 Part D - BGP Message related events

```
 #  Event     Idle  CONNECT ACTIVE    OPEN     OPEN      Establish
                                      Sent     CONFIRM   ed
-------------------------------------------------------------------
 18 BGPOpen   Idle/  Idle/   Idle/    Open     Collision Collision
               V     D         D      Confirm/ Detect     detect
                                      N        (6.8)      (6.8)

                                               See 2.4.1 See 2.4.2
                                               below     below

                                               Discard:  Discard:
                                                R        R
-------------------------------------------------------------------
 19 BGP Open  Idle/  Open    Open     Open      Idle/    Idle/
    & delay    V    Confirm/ Confirm/ Confirm/  E        E
```

timer set		H	H	H		
20 BGPHeader err	Idle/ V	Idle/ D	Idle / D	Idle/ I	Idle / I	Idle / E
21 OpenMsg Err	Idle/ V	Idle/ D	Idle/ D	Idle/ J	Idle/ J	Idle / E
22 Open collision dump existing BGP session	Idle/ V	Idle/ D	Idle/ D	Idle/ R	Idle/ R	Idle / R
23 Notify with version error	Idle/ V	Idle/ D	Idle/ D	Idle/ Y	Idle/ Y	Idle/ T
24 Notification Received	Idle/ V	Idle/ D	Idle/ D	Idle/ Y	Idle/ Y	Idle/ T
25 Keepalive message	Idle/ V	Idle/ D	Idle/ D	Idle/ E	Established/ P	Established/ P
26 Update message	Idle/ V	Idle/ D	Idle/ D	Idle/ E	Idle/ E	Established/ W
27 Update message error	Idle/ V	Idle/ D	Idle/ D	Idle/ E	Idle/ E	Idle / U

2.4.1 Collision Detect processing in Open Confirm

Upon receipt of an OPEN message, the local system must check for any collision of BGP connections. Upon receiving an Open in Open Confirm state, the local system checks against any existing connections in the Open Confirm state, and connections in the Open Sent state if it knows the BGP identifier by means outside the protocol. The following rules from section 6.8 of [1] are followed:

1. The BGP identifier of the local system is compared to the BGP Identifier of the remote system (as specified in the Open message.)

2. If the value of the local BGP Identifier is less than the remote one, the local system:

   1) Sends an Open Message back to the remote peer

   2) closes the BGP connection that already exists via event 22 as a separate call to the state machine.

   3) Stays in Open Confirmed state

3. Otherwise,the local system closes the newly created BGP session by sending a TCP Fin on the new connection, and stays in Open Confirmed state and execute Action R

2.4.2 Collision Detect Processing in Established state

Collection detection processing depends if the local system is configured to process Open's in EStablished state. If the local system is not configured to process Open's in the Established state, the normal processing of Collision detect in Established state is to have the local system close the newly created BGP session and stay in Established state.

If the box has a configuration that indicates that the Collision should be detected, the following tests will be made:

1. The BGP identifier of the local system is compared to the BGP Identifier of the remote system (as specified in the Open message.)

2.) If the value of the local BGP Identifier is less than the remote one, the local system

   1) Sends an Open statement back to the remote peer
   2) Closes the BGP connection via event 22 as a separate call to the state machine

   3) transitions to the Open Confirmed state from Established state

3.) Otherwise, the local system closes the new BGP connection via action R

3.0 Actions in state table

3.1 Actions for Initialization

Action A

1) Initialize all BGP resources
2) ConnectRetryCnt set to 0
3) Start Connect retry timer with initial value
4) Initiate transport connection to the BGP peer
   (if using TCP, send a TCP
5) Listen for connection set-up by the remote BGP peer
   (that is, listen for TCP syn, ack pair)

Action B

1) Initialize all BGP resources
2) ConnectRetryCnt set to 0
3) Start connect retry timer with initial value
4) Listen for connection set-up by remote BGP peer
   [TCP syn]

3.2 Actions for Disconnection

Action C

1) Set Administrative stop in MIB code reason
2) Send Notification with Cease
3) Connect retry timer reset (set to zero)
4) If any routes are associated with BGP session, delete these routes
5) Release all BGP resources
6) Drop TCP connection
   (Send FIN, and await FIN-ACK)
7) Increment the ConnectRetryCnt by 1
8) BGP peer oscillation damping process [2]

Action D

1) Set FSM error in MIB reason code

```
      2) Connect retry timer reset (set to zero)
      3) If any routes are associated with the BGP session, delete
         the routes associated with this BGP session.
      4) Release all BGP resources
            (including clear BGP Delay timer)
      5) Drop TCP connection
            [send TCP FIN and Await FIN-ACK)

      6) Increment the connect retry count by 1
      7) BGP peer oscillation damping process [2]

    Action E
      1) Set FSM error in MIB reason code
      2) Send Notification with FSM error
      3) Connect retry timer reset (set to zero)
      4) If any routes are associated with the BGP session, delete
         the routes associated with the BGP session
      5) Release all BGP resources
      6) Drop the TCP connection
            [send TCP Fin and Await FIN-ACK]
      7) Increment the ConnectRetryCnt by 1
      8) BGP peer oscillation damping process [2]

    Action I
      1) Set BGP Header error in MIB reason code
      2) Send Notification with BGP Header Error
      3) Connect retry timer reset (set to zero)
      4) Drop TCP connection
            (Send a FIN, Await a FIN-ACK)
      5) Release all BGP resources
      6) Increment the ConnectRetryCnt by 1
      7) BGP peer oscillation damping process [2]

    Action J
      1) Set Open message error in MIB reason code
      2) Send Notification with Open message error
      3) Connect retry timer reset (set to zero)
      4) Release all BGP resources
      5) Drop TCP connection
            (Send a FIN, await a FIN-ACK)
      6) Increment the ConnectRetryCnt by 1
      7) BGP peer oscillation damping process [2]
```

```
    Action K
```

```
  1) Set Hold Timer expired in MIB Reason code
  2) Send Notification with Hold timer expired
  3) Connect retry timer reset (set to zero)
  4) Release all BGP resources
  5) Drop TCP connection
        (Send a Fin, await a FIN-ACK)
  6) Increment the ConnectRetryCnt by 1
  7) BGP peer oscillation damping process [2]


Action M
  1) set Hold timer expired in MIB reason code
  2) send Notification with Hold Timer expired
  3) Delete Routes
  4) Connect retry timer reset (set to zero)
  5) Release all BGP resources
  6) Drop TCP connection
        (send a Fin, await a FIN-ACK)
  7) Increment the ConnectRetryCnt by 1
  8) BGP Peer oscillation damping process [2]


Action R
  If Collision Detected (section 6.8) and this bgp session is
  the sesion to be dumped,
   1) set the Call Collision cease in MIB reason code
   2) Send Notification with a Cease
   3) reset Connect timer (set to zero)
   4) Release all BGP resources
   5) Drop TCP connection (Send TCP FIN, await TCP FIN ACK)
   6) Increment ConnectRetryCnt by 1
   7) BGP peer oscillation damping process [2]

Action S
   1) Set Administrative down in MIB Reason code
   2) Send Notificationwith Cease
   3) Drop TCP connection [Send TCP FIN, await TCP FIN ACK)
   4) set ConnectRetryCnt to 0
   5) reset Connect retry timer (set to zero)


Action T
   1) set appropriate code in MIB reason code
      - either Notification, or TCP disconnect or
        TCP failure

   2) delete routes
   3) Reset Connect retry timer (set to zero)
   4) Release all BGP resources
   5) Drop TCP connection
   6) Incremetn the ConnectRetryCnt by 1
```

7) BGP Peer oscillation dampoing process [2]

Action U

1) Set Update error code in MIB Reason code

Action V

1) set FSM error in MIB reason code

Action X

1) set value of Notfication received in MIB reason code
2) Drop TCP connection (Send TCP FIN)
3) Restart Connection Retry Timer (with initial value)
4) Send TCP Sync
5) Listen for TCP Sync

Action y

1) Set TCP disconnect in MIB reason code
2) Restart ConnectRetry timer (with initial value)
3) Release all BGP resources
4) Drop TCP connection (send FIN ACK)
5) Increment ConnectRetryCnt by 1
6) BGP peer oscillation damping process [2]

Action Z

1) Set Manual disconnect in MIB reason code
2) Drop TCP connection [Send FIN, await FIN Ack]
3) Release all BGP
4) Set ConnectRetryCnt to zero
5) Reset ConnectRetry timer

3.3 TCP Connection Actions

Action F

1) Restart ConnectRetry timer (with initial value)
2) Initiates a transport connection to the other bgp peer
   [Send a TCP SYN]
3) Listen for remote transport connection that
   may be initiated by the remote BGP peer (TCP connection)

Action G

1) Restart ConnectRetry timer (with initial value)
2) Listen for BGP peer connection

Action L
1) Reject TCP connection [send TCP FIN]

Action O
1) Clear BGP resources
2) Restart ConnectRetry Timer
3) Listen for TCP connection (Listen for TCP Sync)

Action ZZ:
1) Set the BGP Open delay timer to initial value

3.4) Sending the Open message

Action H
1) Clear ConnectRetry timer (clear to zero)
2) Complete BGP initialization
3) Clear BGP Open Delay timer
3) Send Open message
4) Hold timer set to Large value (4 minutes)

3.5) Keepalive message related

Action N
1) Clear BGP delay timer
2) Reset BGP Connect Timer
3) Send KEEPALIVE message
4a) If negotiated Hold time value is non-zero,
- Set Keepalive timer
- Hold timer reset to negotiated Hold time value [section 4.2 of bgp-4 draft [1]]

4b) If negotiated Hold time value is zero,
- Reset Keepalive timer
- Reset Hold timer

5) If value of Autonomous System field is the same as the local Autonomous system number, set the connection is an internal connection; otherwise it is ôexternalö. (This impacts Update processing)

```
      Action P
        1) Restart Hold timer

      Action Q
        1) Send Keepalive message
        2) Restart KEEPALIVE timer


   3.6) Update packet related

      Action W
        1) Process update packet
        2) Restart Hold timer
```

4.0 Security Considerations

Security concerns for BGP-4 are addressed in the BGP-4 specification, and accompanying specifications on TCP MD5 [3] and IP Security[4]. No additional considerations need to be made for the BGP-4 state machine description.

6.0 References

[1] "A Border Gateway Protocol 4 (BGP-4)" Y. Rekhter, T. Li Editors
http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-17.txt

[2] "BGP Peer Restart Backoff Mechanisms", S. Hares
draft-skh-bgp-backoff-00.txt

[3] "Protection of BGP Sessions via the TCP MD5 Signature Option"
A. Heffernan, rfc2385.txt

[4] Securing BGPv4 using Ipsec", D. Ward,
draft-ward-bgp-ipsec-00.txt

[5] TCP, RFC 793

[6] SCTP, RFC 2960

7.0 Author's Addresses

Susan Hares
NextHop Technologies, Inc

825 Victors Way   Phone: 1-734-222-1610
Ann Arbor, MI USA   Email: skh@nexthop.com