

# BGP FSM Discussion

IDR Chairs, IETF 120

# Brief History for BGP Finite State Machine

- Most of the states we recognize in the BGP FSM started in BGP-1, RFC 1105.
- RFC 1771 gave us what we recognize as the current BGP-4 FSM with six states.
- draft-ietf-idr-bgp4-17 is where we start seeing the updates to the FSM where actions and additional events are accounted for.
  - This was motivated by Area Director feedback that we had gaps vs. reality
- RFC 4271 gave us the current text

# A segue on terminology

- The original BGP FSM was intended to be in the style of a Moore machine where the outputs depend on current state and its inputs.  
[https://en.wikipedia.org/wiki/Moore\\_machine](https://en.wikipedia.org/wiki/Moore_machine)
- In RFC 4271:
  - RFC 4271 does not formally have sub-states.
  - We have six states: Idle, Active, Connect, OpenSent, OpenConfig, Established. Idle is our “start” state. We don’t have a formal “finish” state.
  - A table of “Events”, some of which may be “Optional”. Some events may be “event while variable is true”.
  - Session attributes (i.e., variables) that represent provisioning state, or potentially track sub-state behaviors, and timers. May cause conditional branching in a state transition.

# Things that could have been tracked using sub-states

- Timers running. However, simply handling their expirations modeled better.
- Delay open
- TCP state tracking. But this gets messy in interesting ways when talking about graceful restart!
- BGP session collisions. “Outside the scope of this document.” Really about the fact that multiple FSM instances between two endpoints may be running at the same time and this is an event occurring between those two instances.
  - However, TCP connection tracking is part of this machinery!
- Version negotiation. Really about an instance of a FSM has been started for that version of BGP.

# Missing stuff in RFC 4271

- The transmission of Update messages isn't described in the FSM, only that receiving an Update outside of Established is a FSM error.

# State transitions

- For the current state, an event (potentially a “decorated” event) occurs.
- For that event dispatch, a next state is specified.
- As part of next state transition, a set of “actions” take place:
  - Sending of PDUs
  - Setting/Clearing timers
  - Updating/Clearing session attributes
  - The state change.

# Actions

- The text of RFC 4271 has significant redundancy in the text covering actions as part of state transitions. This was a style choice at the time the text was being worked on.
- This redundant text wasn't fully consistent due to size and number of iterations while working on the text.

# RFC 4271 choices, maintaining the original states

- When viewed in a black-box format, the BGP FSM appears to move through its states by the messages it sends on the wire: OPEN, NOTIFICATION, KEEPALIVE.
- The internal changes in RFC 4271 were done to try to preserve the existing big-picture FSM.
- These states are made visible in management mechanisms like SNMP and YANG.
  - If we consider sub-states, we'll want to consider what we expose.

# Maintaining FSM text

- The style has historically been to copy “old text”, especially for actions, and update with “new text”.
- This makes the “diff” hard to read.
- It doesn't deal with multiple RFCs “patching” the same event well.

# Consistency in FSM updates across new RFCs

- RFC 4724 - patching in the GR retention with the FSM. Consider also end-of-rib behaviors in the Established state.
- RFC 5291 - additional machinery on IMMEDIATE for ORFs in a refresh.
- RFC 7313 - enhanced route refresh machinery in Established.

# in-progress drafts impacting FSM

- draft-ietf-idr-bgp-sendholdtimer
- draft-ietf-idr-bgp-bfd-strict-mode
- draft-retana-idr-bgp-quic

We're also considering moving BGP to full standard and this may involve cleaning up RFC 4271 FSM to address points raised in these discussions.

# Making things maintainable

- Even though we're doing "Moore" style state machines, it's possible for us to impact things "on input" as part of our events.
- When actions are common, we can use text that impacts event transition as a whole without needing to update every single bit of event text. This aids comprehension of the change.
  - sendholdtimer did this for resetting the new timer on state transition.
- Long term, we should consider "appending" actions to transitions rather than copy-and-paste of sections.
  - Avoids "conflicts on merge" when multiple drafts change same section at once.

# Making things maintainable (2)

- Stylistically, leverage RFC document sections to cleanly identify what state and what action is being maintained.
  - style used in bgp-bfd-strict mode.
- Leverage xml2rfc v3 markup to identify blocks better (blockquote) and emphasize changed text.
- Include SVG diagrams in the appendices as non-normative references for clarity.
  - UML gives us a way to consistently represent our state machines once we design a style-guide for BGP.
  - Can we embed the SVG contents from the UML in a human-readable plain-text format in the document normatively? See later presentations.