

# BGP Send Hold Timer – April 2023 IDR Interim

Job Snijders, Fastly, [job@fastly.com](mailto:job@fastly.com)

Ben Cartwright-Cox, Port 179, [ben@benjojo.co.uk](mailto:ben@benjojo.co.uk)

# The problem the `SendHoldTimer` tries to improve

In an ideal situation - when the remote peer is overwhelmed, remote side fails to generate `KEEPALIVES`, causing the local system to close the connection and remove routes pointing to (broken) remote peer.

If the remote peer signals a *TCP Receive Window of 0* for a period of time, local system cannot deliver `KEEPALIVE` messages, and if the (broken) remote peer doesn't close the connection because of the missing `KEEPALIVES`, on many systems the connection stays open, leading to operational problems.

# Operational consequences of stuck BGP sessions

- **Failure to deliver BGP messages to the remote peer implies the remote peer is operating based on stale information.**
  - The remote peer did not receive the latest UPDATES/WITHDRAWs
- **Failure to disconnect from a stuck peer means the local system is operating based on stale information.**
  - If the remote system is not progressing incoming BGP messages, any routes pointing at the stuck peer are now suspect.
- **Failure to disconnect from a stuck peer hinders the local system's ability to inform its other peers.**
  - Might poison the whole network

## Criticism of the internet-draft contents:

- Changes to the FSM (“the surgery”) not complete enough
- Changes to the FSM too invasive

## Criticism of the concept:

- It might not solve all corner cases of remote peer being broken

# Actionable suggestions for next I-D version

- **Ensure default SendHoldTimer provides sufficient time**

- [Jgs@Juniper](#) suggested “exactly as long as a piece of string”
- At least twice the HoldTimer
- 8, 20, or 60 minutes

*All in principle agreeable – the objective is that it fires at some point in time*

- **Change tone of voice of the I-D to be less alarmist**

- **Don't update RFC 4271 FSM text, but rather provide descriptive text outlining desired outcome**

- **Document what the idea does *not* solve**

# Implementation status

- OpenBGPD
- FRRouting
- neo-bgp

# Implementation details for OpenBGPD

Provided that:

- Remote system's receive buffer clogs up to the point it starts signaling TCP Receive Window = 0
- AND Local system's outbound write buffer fills up
- AND msgbuf\_write()→sendmsg() didn't happen for duration of SendHoldTimer (caused by socket being unwritable)

→ only then, connection is closed (as if HoldTimer expired)

(So in practise it takes a bit longer than SendHoldTimer for SendHoldTimer mechanism to kick in)

**Very coarse mechanism, really an option of *last resort***

# Re-iteration of the goal

This mechanism really is intended to be a *backstop of last resort*

Intended to address pathological cases where for long periods of time the remote side is not progressing inbound information.



# Hopes for continuation of the work

- Adoption as Working Group Document
- Gain more implementation experience
- IANA registration for *Send Hold Timer Expired* in the *BGP Error Codes* sub-registry

# References

**BGP Zero-Window test:**

<https://github.com/benjojo/bgp-zerowindow-test>

**Hunting Down Stuck Routes (TCP Zero Window suspected)**

<https://blog.benjojo.co.uk/post/bgp-stuck-routes-tcp-zero-window>

**Draft-spaghetti-idr-bgp-sendholdtimer**

<https://datatracker.ietf.org/doc/html/draft-spaghetti-idr-bgp-sendholdtimer>