Stream: Internet Engineering Task Force (IETF)
RFC: 9687
Updates: 4271
Category: Standards Track
Published: October 2024
ISSN: 2070-1721
Authors: J. Snijders, *Fastly*; B. Cartwright-Cox, *Port 179*; Y. Qu, *Futurewei*

RFC 9687
Border Gateway Protocol 4 (BGP-4) Send Hold Timer

Abstract

This document defines the SendHoldTimer, along with the SendHoldTimer_Expires event, for the Border Gateway Protocol (BGP) Finite State Machine (FSM). Implementation of the SendHoldTimer helps overcome situations where a BGP connection is not terminated after the local system detects that the remote system is not processing BGP messages. This document specifies that the local system should close the BGP connection and not solely rely on the remote system for connection closure when the SendHoldTimer expires. This document updates RFC 4271.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9687.

Copyright Notice

Table of Contents

1. Introduction

This document defines the `SendHoldTimer`, along with the `SendHoldTimer_Expires` event, for the Border Gateway Protocol (BGP) Finite State Machine (FSM) defined in Section 8 of [RFC4271].

Failure to terminate a blocked BGP connection can result in network reachability issues, and the subsequent failure to generate and deliver BGP UPDATE messages to another BGP speaker of the local system is detrimental to all participants of the inter-domain routing system. This phenomena is thought to have contributed to IP traffic blackholing events in the global Internet routing system [bgpzombies].

This specification intends to improve this situation by requiring that BGP connections be terminated if the local system has detected that the remote system cannot possibly have processed any BGP messages for the duration of the `SendHoldTime`. Through standardization of the aforementioned requirement, operators will benefit from consistent behavior across different BGP implementations.

BGP speakers following this specification do not rely exclusively on remote systems closing blocked connections; they also locally close blocked connections.

2. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Example of a Problematic Scenario

In implementations lacking the concept of a `SendHoldTimer`, a malfunctioning or overwhelmed remote speaker may cause data on the BGP socket in the local system to accumulate ad infinitum. This could result in forwarding failure and traffic loss, as the overwhelmed speaker continues to utilize stale routes.

An example fault state: as BGP runs over TCP [RFC9293], it is possible for a BGP speaker in the Established state to encounter a BGP speaker that is advertising a TCP Receive Window (RCV.WND) of size zero. This 0 window prevents the local system from sending KEEPALIVE, UPDATE, or any other critical BGP messages across the network socket to the remote speaker.

Generally BGP implementations have no visibility into lower-layer subsystems such as TCP or the speaker's current Receive Window size, and there is no existing BGP mechanism for such a blocked connection to be recognized. Hence BGP implementations are not able to handle this situation in a consistent fashion.

The major issue arising from a BGP speaker being unable to send a BGP message to a given remote speaker is that as a result that speaker subsequently is operating based on stale routing information. Failure of the BGP speaker to send (and thus the remote speaker to receive) BGP messages on a single BGP session can negatively impact the ability of an entire autonomous system (or even a group of autonomous systems) to converge.

4. Changes to RFC 4271 - SendHoldTimer

BGP speakers are implemented following a conceptual model "BGP Finite State Machine" (FSM), which is outlined in Section 8 of [RFC4271]. This specification adds a BGP timer, `SendHoldTimer`, and updates the BGP FSM as indicated in the following subsections.

4.1. Session Attributes

The following optional session attributes for each connection are added to the list in Section 8 of [RFC4271] appearing just prior to "The optional session attributes support different features of the BGP functionality that have implications for the BGP FSM state transitions":

NEW

> 14) SendHoldTimer
>
> 15) SendHoldTime

`SendHoldTime` determines how long a BGP speaker will stay in the Established state before the TCP connection is dropped because no BGP messages can be transmitted to its peer. A BGP speaker can configure the value of the `SendHoldTime` for each peer independently.

4.2. Timer Event: SendHoldTimer_Expires

Another timer event is added to Section 8.1.3 of [RFC4271] as follows:

NEW

> Event 29: SendHoldTimer_Expires
>
> Definition: An event generated when the SendHoldTimer expires.
>
> Status: Optional

4.3. Changes to the FSM

The following changes are made to Section 8.2.2 of [RFC4271].

In "OpenConfirm State", the handling of Event 26 is revised as follows:

OLD

> If the local system receives a KEEPALIVE message (KeepAliveMsg (Event 26)), the local system:
>
> - restarts the HoldTimer and
> - changes its state to Established.

NEW

> If the local system receives a KEEPALIVE message (KeepAliveMsg (Event 26)), the local system:
>
> - restarts the HoldTimer,
> - starts the SendHoldTimer if the SendHoldTime is non-zero, and
> - changes its state to Established.

The following paragraph is added to Section 8.2.2 of [RFC4271] in "Established State", after the paragraph that ends "unless the negotiated HoldTime value is zero":

NEW

> If the SendHoldTimer_Expires (Event 29) occurs, the local system:
>
> - (optionally) sends a NOTIFICATION message with the BGP Error Code "Send Hold Timer Expired" if the local system can determine that doing so will not delay the following actions in this paragraph,
> - logs an error message in the local system with the BGP Error Code "Send Hold Timer Expired",
> - releases all BGP resources,
> - sets the ConnectRetryTimer to zero,
> - drops the TCP connection,
> - increments the ConnectRetryCounter by 1,
> - (optionally) performs peer oscillation damping if the DampPeerOscillations attribute is set to TRUE, and
> - changes its state to Idle.
>
> Each time the local system sends a BGP message, it restarts the SendHoldTimer unless the SendHoldTime value is zero or the negotiated HoldTime value is zero, in which case the SendHoldTimer is stopped.

> The SendHoldTimer is stopped following any transition out of the Established state as part of the "release all BGP resources" action.

4.4. Changes to BGP Timers

Section 10 of [RFC4271] summarizes BGP timers. This document adds another optional BGP timer: `SendHoldTimer`.

NEW

> SendHoldTime is an FSM attribute that stores the initial value for the SendHoldTimer. If SendHoldTime is non-zero, then it **MUST** be greater than the value of HoldTime; see Section 6 of [RFC9687] for suggested default values.

5. Send Hold Timer Expired Error Handling

If the local system does not send any BGP messages within the period specified in `SendHoldTime`, then a NOTIFICATION message with the "Send Hold Timer Expired" Error Code **MAY** be sent and the BGP connection **MUST** be closed. Additionally, an error **MUST** be logged in the local system, indicating the "Send Hold Timer Expired" Error Code.

6. Implementation Considerations

Due to the relative rarity of the failure mode that this specification is designed to address, and also the fact that network operators may be unfamiliar with the formal specification of BGP fault detection mechanisms such as `HoldTimer`, it is likely that a large number of operators will be unaware of the need for an additional mechanism such as `SendHoldTimer`.

Accordingly, it is **RECOMMENDED** that implementations of this specification enable `SendHoldTimer` by default, without requiring additional configuration of the BGP-speaking device.

The default value of `SendHoldTime` for a BGP connection **SHOULD** be the greater of:

- 8 minutes or
- 2 times the negotiated HoldTime

Implementations **MAY** make the value of `SendHoldTime` configurable, either globally or on a per-peer basis, within the constraints set out in Section 4.4.

The subcode for NOTIFICATION message "Send Hold Timer Expired" is set to 0 and is not used; no additional data is to be appended to the end of a "Send Hold Timer Expired" NOTIFICATION message.

7. Operational Considerations

When the local system recognizes that a remote speaker has not processed any BGP messages for the duration of the `SendHoldTime`, it is likely that the local system will not be able to inform the remote peer through a NOTIFICATION message as to why the connection is being closed. This document suggests that an attempt to send a NOTIFICATION message with the "Send Hold Timer Expired" Error Code still be made, if doing so will not delay closing the BGP connection. Meanwhile, an error message is logged into the local system.

Other mechanisms can be used as well, for example, BGP speakers **SHOULD** provide this reason as part of their operational state (for example, bgpPeerLastError in the BGP MIB [RFC4273]).

8. Security Considerations

This specification does not change BGP's security characteristics. Implementing the BGP `SendHoldTimer` as specified in this document will enhance network resilience by terminating connections with malfunctioning or overwhelmed remote peers.

9. IANA Considerations

IANA has registered value 8 for "Send Hold Timer Expired" in the "BGP Error (Notification) Codes" registry within the "Border Gateway Protocol (BGP) Parameters" registry group.

10. Acknowledgements

The authors would like to thank William McCall, Theo de Raadt, John Heasley, Nick Hilliard, Jeffrey Haas, Tom Petch, Susan Hares, Keyur Patel, Ben Maddison, Claudio Jeker, and John Scudder for their helpful review of this document.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/info/rfc9293>.

11.2. Informative References

[bgpzombies] Fontugne, R., "BGP Zombies", April 2019, <https://labs.ripe.net/author/romain_fontugne/bgp-zombies/>.

[RFC4273] Haas, J., Ed. and S. Hares, Ed., "Definitions of Managed Objects for BGP-4", RFC 4273, DOI 10.17487/RFC4273, January 2006, <https://www.rfc-editor.org/info/rfc4273>.

Authors' Addresses

Job Snijders
Fastly
Amsterdam
Netherlands
Email: job@fastly.com

Ben Cartwright-Cox
Port 179 Ltd
London
United Kingdom
Email: ben@benjojo.co.uk

Yingzhen Qu
Futurewei Technologies
Santa Clara,
United States of America
Email: yingzhen.ietf@gmail.com