PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: August 15, 2025

Y. Liu
X. Xu
China Mobile
C. Lin
New H3C Technologies

February 15, 2025

Multicast Source Discovery Protocol (MSDP) Send Hold Timer
draft-liu-pim-msdp-sendholdtimer-03

Abstract

This document defines the SendHoldTimer and the SendHoldTimer_Expires event in the MSDP protocol. The implementation of SendHoldTimer helps to address the situation where the local system detects that the remote system has not processed MSDP messages but has not terminated the MSDP session. According to this document, when the SendHoldTimer expires, the local system should close the MSDP session connection, rather than relying on the remote system to initiate the session closure. This document updates RFC3618.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 15, 2025.

Table of Contents

1. Introduction

As described in [I-D.ietf-idr-bgp-sendholdtimer], any upper-layer protocol that uses TCP for transport can encounter similar situations where the remote system is unable to read TCP data due to a failure, leading to the inability to terminate the BGP connection.

MSDP also requires a BGP-like send timeout mechanism [I-D.ietf-idr-bgp-sendholdtimer] to resolve this issue.

This document defines the SendHoldTimer and SendHoldTimer_Expires mechanisms for MSDP [RFC3618], as defined in Section 5 of that specification.

The failure to terminate blocked MSDP sessions may lead to Denial of Service (DoS) attacks, resulting in the inability to generate and convey multicast source information, thereby disrupting normal multicast forwarding. This specification aims to address this situation by requiring the termination of sessions when the local system detects that the remote system is unable to process any MSDP messages during the SendHoldTime.

With this specification, blocked connections can be terminated by the remote system and also by the local system between MSDP devices.

1.1. Conventions and Terminology

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.

2. Problem Statement

An example of a network failure scenario is when an established MSDP session, running over TCP [RFC9293], between MSDP peers is disrupted due to one peer declaring a zero-sized TCP receive window (RCV.WND) to the other peer. This zero-sized TCP receive window prevents the local system from sending critical MSDP messages such as KEEPALIVE and SA messages to the remote peer via the network socket.

In the absence of an implementation of the SendHoldTimer concept, a failed or overwhelmed remote peer can lead to a continuous accumulation of data on the MSDP socket of the local system, resulting in the inability to transmit multicast source information and consequently affecting the normal multicast forwarding.

Normally, MSDP implementations are unable to observe the current receive window size of the underlying subsystem (such as TCP) or the peer, and there are no MSDP mechanisms to terminate such blocked sessions. As a result, MSDP implementations are unable to handle this situation consistently.

This document provides a mechanism that allows MSDP implementations to detect whether the TCP socket between MSDP peers is making progress (data is being transmitted) or is stalled. In the case of a stall, the MSDP session can be restarted to restore the normal operation of the MSDP protocol.

3. SendHoldTimer - Changes to RFC 3618

3.1. Changes to Timers

In Chapter 5, timers for the MSDP protocol are defined. Add descriptions for [SendHoldTimer] and [SendHoldTime].

Next Text:

5.7 Send Hold Timer

Once the MSDP Peer transitions to the UP state, the MSDP Peer initiates the [SendHoldTimer] with a timeout value of [SendHoldTime]. When sending KeepAlive messages or SA messages to the peer, the [SendHoldTimer] is reset to [SendHoldTime]. It is recommended that [SendHoldTime] be consistent with [HoldTime-Period].

Upon expiration of the [SendHoldTimer], the MSDP session will be terminated.

3.2. Changes to MSDP Connection State Machine

Chapter 11 describes the finite state machine for MSDP. In this chapter, add the event "Send Hold Timer Expired" and specify that upon handling this event, the TCP connection should be closed.

The specific modification is as follows:

Old Text:

11.1. Events

...

E10) Any other error detected

Next Text:

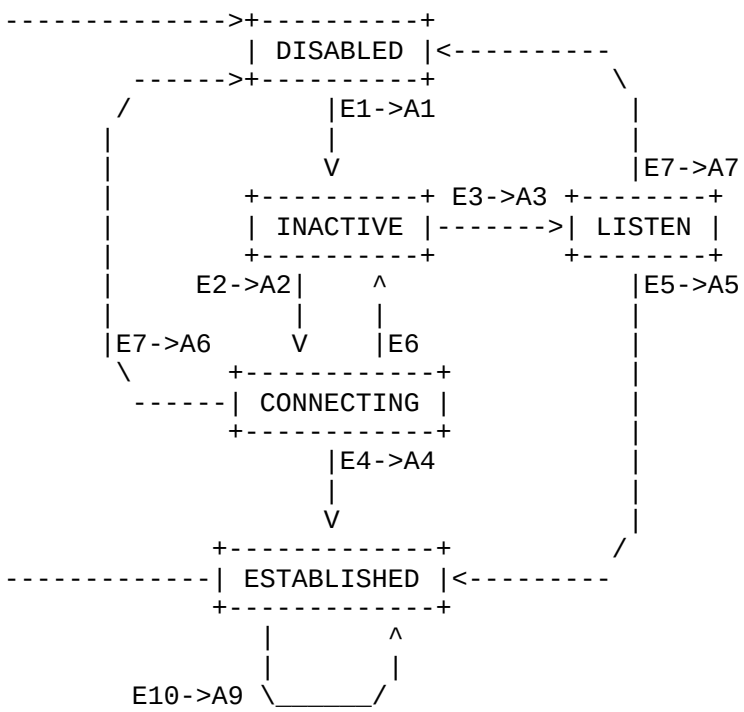
11.1. Events

...

E10) Any other error detected

E11) Send Hold Timer expired

Modify the state machine diagram to include the handling of the "Send Hold Timer Expired" event.



Modify the state machine diagram to include the handling of Send Hold Timer expiration in the "Establish" state.

Old Text:

E7->A8
E8->A8
E9->A8

Next Text:

E7->A8

E8->A8
E9->A8
E11->A8

Changes to Peer-specific Events: Add handling of the event for successful sending of KeepAlive or SA messages.

Old Text:

11.3. Peer-specific Events

...

*) KeepAlive timer expired:

-> Send KeepAlive TLV

-> Set KeepAlive timer to [KeepAlive-Period]

Next Text:

11.3. Peer-specific Events

...

*) KeepAlive timer expired:

-> Send KeepAlive TLV

-> Set KeepAlive timer to [KeepAlive-Period]

*) KeepAlive or SA TLV sended:

-> Set Send Hold Timer to [SendHoldTime]

4. Security Considerations

This specification addresses the potential vulnerability of MSDP to attacks where MSDP peers pretend to be unable to process MSDP messages, causing the MSDP protocol to malfunction.

In other aspects, this specification does not alter the security characteristics of MSDP.

5. IANA Considerations

This document does not request any IANA modifications.

6. Acknowledgements

The authors wish to thank Job Snijders, Ben Cartwright-Cox, and Yingzhen Qu for their work on the BGP Send Hold Timer concept.

7. References

7.1. Normative References

[I-D.ietf-idr-bgp-sendholdtimer] Snijders, J., Cartwright-Cox, B., and Y. Qu, "Border Gateway Protocol 4 (BGP-4) Send Hold Timer", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-sendholdtimer, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-sendholdtimer>

[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>.

[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>.

[RFC3618] D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003,

<https://www.rfc-editor.org/info/rfc3618>.

7.2. Informative References

Authors' Addresses

Yi Liu
China Mobile
China
Email: liuyi4@ha.chinamobile.com

Xiaolei Xu
China Mobile
China
Email: xuxiaolei@ha.chinamobile.com

Changwang Lin
New H3C Technologies
China
Email: linchangwang.04414@h3c.com