Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 6, 2012

N. Akiya
M. Binderberger
Cisco Systems

June 4, 2012

Standardized interval support in BFD
draft-akiya-bfd-intervals-02

Abstract

This document defines a set of interval values that we call "Standard intervals". Values of this set must be supported for transmitting BFD control packets and for calculating the detection time in the receive direction when the value is equal or larger than the fastest, i.e. lowest, interval a particular BFD implementation supports.

This solves the problem of finding an interval value that both BFD speakers can support while allowing a simplified implementation as seen for hardware-based BFD.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 http://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 December 6, 2012.

Copyright Notice

Table of Contents

1. Introduction

The standard [RFC5880] describes how to calculate the transmission interval and the detection time. It does not make any statement though how so solve a situation where one BFD speaker cannot support the calculated value. In practise this may not been a problem as long as software-implemented timers have been used and as long as the granularity of such timers was small compared to the interval values being supported, i.e. as long as the error in the timer interval was small compared to 25 percent jitter.

In the meantime requests exist for very fast interval values, down to 3.3msec for MPLS-TP. At the same time the requested scale for the number of BFD sessions in increasing. Both requirements have driven vendors to use Network Processors (NP), FPGAs or other hardware-based solutions to offload the periodic packet transmission and the timeout detection in the receive direction. A potential problem with this hardware-based BFD is the granularity of the interval timers. Depending on the implementation only a few intervals may be supported, which can cause interoperability problems. This document proposes a set of interval values that must be supported by all implementations. Details are laid out in the following sections.

2. The problem with few supported intervals

Lets assume vendor "A" supports 10msec, 100msec and 1sec interval timers in hardware. Vendor "B" supports every value from 20msec onward, with a granularity of 1msec. For a BFD session "A" tries to set up the session with 10msec while "B" uses 20msec as the value for RequiredMinRxInterval and DesiredMinTxInterval. RFC5880 describes that the negotiated value for Rx and Tx is 20msec. But system "A" is not able to support this. Multiple ways exist to resolve the dilemma but none of them is without problems.

a. Realizing that it cannot support 20msec, system "A" sends out a new BFD packet, advertising the next larger interval of 100msec with RequiredMinRxInterval and DesiredMinTxInterval. The new negotiated interval between "A" and "B" then is 100msec, which is supported by both systems. The problem though is that we moved from the 10/20msec range to 100msec, which has far deviated from operator expectations.

b. System "A" could violate RFC5880 and use the 10msec interval for the Tx direction. In the receive direction it could use an adjusted multiplier value M' = 2 * M to match the correct detection time. Now beside the fact that we explicitly violate RFC5880 there may be the problem that system "B" drops up to 50%

of the packets; this could be the case when "B" uses an ingress rate policer to protect itself and the policer would be programmed with an expectation of 20msec receive intervals.

The example above could be worse when we assume that system "B" can only support a few timer values itself. Lets assume "B" supports "20msec", "300msec" and "1sec". If both systems would adjust their advertised intervals, then the adjustment ends at 1sec. The example above could even be worse when we assume that system "B" can only support "50msec", "500msec" and "2sec". Even if both systems walk their supported intervals, the two systems will never be able to agree on a interval to run any BFD sessions.

3. Well-defined, standardized intervals

The problem can be reduced by defining interval values that are supported by all implementations. Then the adjustment mechanism could find a commonly supported interval without deviating too much from the original request.

In technical terms the requirement is as follows: a BFD implementation MUST support all values in the set of Standard interval values which are equal to or larger than the fastest, i.e. lowest, interval the particular BFD implementation supports.

The proposed set of Standard interval values is: 3.3msec, 10msec, 20msec, 50msec, 300msec and 1sec.

This document is not adding new requirements with respect to how exact a timer value must be implemented. Supporting an interval value means to advertise this value in the DesiredMinTxInterval and/or RequiredMinRxInterval field of the BFD packets and to provide timers that are reasonably close. RFC5880 defines safety margins for the timers by defining a jitter range.

How is the "Standard interval set" used exactly? In the example above, vendor "A" has a fastest interval of 10msec and thus would be required to support all intervals in the standard set that are equal or larger than 10msec, i.e. it would support 10msec, 20msec, 50msec, 300msec, 1sec. Vendor "B" has a fastest interval of 20msec and thus would need to support 20msec, 50msec, 300msec and 1sec. As long as this requirement is met for the standard set of values, then both vendor "A" and "B" are free to support additional values outside of the standard set.

4. IANA Considerations

No request to IANA.

5. Security Considerations

This document does not introduce any additional security issues and the security mechanisms defined in [RFC5880] apply in this document.

6. Acknowledgements

We would like to thank Sylvain Masse and Anca Zamfir for bringing up the discussion about the Poll sequence. Jeffrey Haas helped finding the fine line between "exact" and "pedantic".

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

Appendix A. Why some intervals are in the standard set

The list of standard interval values is trying to balance various objectives. The list should not contain too many values as more timers may increase the implementation costs. On the other hand less values produces larger gaps and adjustment jumps. Larger values in the lower interval range may be easier to support, potentially even in software instead of hardware.

- 3.3msec: required by MPLS-TP
- 10msec and 20msec: both values allow to detect faster than 50msec, when used with a multiplier of 2 or 3 (for 10msec). A compromise could be a single interval of 15msec.
- 50msec: this seems an interval often supported by software implementations, so the assumption here is that for convenience this value should be supported.
- 300msec: this would support large scale of 3 x 300msec setups used by customers to have a detection time slightly below 1sec for VoIP

setups.

- 1sec: as mentioned in RFC5880

With that stated, one of the primary intention of this first draft is to seek feedback on the number of interval values in the standard set as well as each value.

Appendix B. Timer adjustment with non-identical interval sets

RFC5880 implicitly assumes that a BFD implementation can support any timer value equal or above the advertised value. When a BFD speaker starts a poll sequence then the peer must reply with the Final (F) bit set and adjust the transmit and detection timers accordingly. With contiguous software-based timers this is a valid assumption. Even in the case of a small number of supported interval values this assumption holds when both BFD speakers support exactly the same interval values.

But what happens when both speakers support intervals that are not supported by the peer? An example is router "A" supporting the standard interval set plus 100 msec while router "B" support the standard intervals plus 200 msec. Assume both routers are configured and run at 50msec. Now router A is configured for 100msec. We know the result must be that both BFD speaker use 300 msec timers but how do they reach this endpoint?

First router A is sending a packet with 100msec. The P bit is set according to RFC5880. The Tx timer stays at 50msec, the detection timer is 3 * 100msec:

(A) DesiredTx: 100msec, MinimumRx: 100msec, P-bit
Tx: 50msec , Detect: 3 * 100msec

Router B now must reply with an F bit. The problem is B is confirming timer values which it cannot support. The only setting to avoid a session flap would be

(B) DesiredTx: 200msec, MinimumRx: 200msec, F-bit
Tx: 50msec , Detect: 3 * 200msec

immediately followed by a P-bit packet as the asvertised timer values have been changed:

(B) DesiredTx: 200msec, MinimumRx: 200msec, P-bit
Tx: 50msec , Detect: 3 * 200msec

This is not exactly what RFC5880 states in section 6.8.7 about the transmission rate. On the other hand as we will see this state does not last for long. Router A would adjust it's timers based on the received Final bit

(A) Tx: 100msec , Detect: 3 * 300msec

Router A is not supporting the proposed 200msec and would use 300msec instead for the detection time. It would then respond to the received Poll sequence from router B, using 300msec as router A does not support the Max(100msec, 200msec):

(A) DesiredTx: 300msec, MinimumRx: 300msec, F-bit
Tx: 100msec , Detect: 3 * 300msec

followed by it's own Poll sequence as the advertised timer values have been changed:

(A) DesiredTx: 300msec, MinimumRx: 300msec, P-bit
Tx: 100msec , Detect: 3 * 300msec

Router B would adjust it's timers based on the received Final

(B) Tx: 200msec , Detect: 3 * 300msec

and would then reply to the Poll sequence from router A:

(B) DesiredTx: 200msec, MinimumRx: 200msec, F-bit
Tx: 300msec , Detect: 3 * 300msec

which finally makes router A adjusting it's timers:

(A) Tx: 300msec , Detect: 3 * 100msec

In other words router A and B go through multiple poll sequences until they reach a commonly supported interval value. Reaching such a value is guaranteed by this draft.

Authors' Addresses

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

Marc Binderberger
Cisco Systems

Email: mbinderb@cisco.com