Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2022

B. Decraene
Orange
C. Bowers
Jayesh. J
Juniper Networks, Inc.
T. Li
Arista Networks
G. Van de Velde
Nokia
G. Solignac
Orange

July 12, 2021

IS-IS Flooding Congestion Control
draft-decraene-lsr-isis-flooding-speed-07

Abstract

This document proposes a mechanism to adjust IS-IS flooding speed between two adjacent routers by adjusting the sender flooding speed to the capability of the receiver. This helps improving the flooding throughput, reducing LSPs losses and retransmissions due to receiver overload, and avoiding manual tuning of flooding parameters by the network operator. This document defines a new TLV for Hello messages. This TLV may carry a set of parameters indicating the performance capacity to receive LSPs.

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 January 13, 2022.

Table of Contents

1. Introduction

IGP flooding is paramount for Link State IGP as routing computations assume that the Link State DataBases (LSDBs) are always in sync across all nodes in the flooding domain.

Slow flooding directly translates to delayed network reaction to failure and LSDB inconsistencies across nodes. The former increases packet loss. The latter translates to routing inconsistencies and possibly micro-loops leading to packet loss, link overload, and jitter for all classes of service. Note that across the network, multiple links may be affected by these forwarding issues, even in the case of a single link failure.

In addition, one single event in the network can require the flooding of multiple LSPs. The typical case is a node failure which requires the flooding of at least one LSP per neighbor of the failed node. Hence, if a node has N IGP neighbors, the failure of this node requires the advertisement and flooding of at least N LSPs. The network won't be able to converge to the new topology until all N LSPs are received by all nodes. Hence there is a need to be able to quickly exchange N LSPs. This document addresses this requirement by allowing the fast flooding of a number of consecutive LSPs.

IGP flooding is hard. One would want fast flooding when the network is stable and slow enough flooding to not overload the neighbor(s) when the network is less stable. Since flooding is performed hop by hop, not overloading the adjacent receiver is sufficient.

Improving the communication speed and efficiency between IS-IS neighbors improves IS-IS scaling. These extensions do not compete with proposed extensions to reduce LSP flooding traffic by reducing the flooding topology such as [I-D.ietf-lsr-dynamic-flooding] . On the contrary, this extension complements those proposals. Indeed reducing the flooding topology does not reduce the size of the LSDB or the total number of LSPs to exchange between two nodes. So

increasing the overall flooding speed can be beneficial for nodes implementing dynamic flooding. The reverse is also true: as dynamic flooding reduces the number of neighbors with flooding enabled, this allows nodes implementing the flooding parameter extensions to focus their flooding resources on those neighbors by sending better parameters to the selected flooding nodes and worse parameters to non-selected flooding nodes.

1.1. 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 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview

Ensuring the goodput between two entities is a layer 4 responsibility as per the OSI model and a typical example is the TCP protocol defined in RFC 793 [RFC0793] . It typically relies on the following sub-functions: flow control, congestion control and reliability.

Flow control is about creating a control loop between a single transmitter and single receiver. TCP provides a mean for the receiver to govern the amount of data sent by the sender. This is achieved by advertising a "receive window", in units of octets, with every ACK. This document proposes to use the same mechanism by advertising a receive window, in units of LSP packets, in IS-IS Hello. The window indicates an allowed number of LSPs that the sender may transmit before receiving acknowledgment of those LSPs. There is an assumption that this is related to the currently available data buffer space available for this adjacency. Indicating a large window encourages transmissions.

Congestion control is about creating multiple interacting control loops between multiple transmitters and multiple receivers. Whereas

flow control prevents the sender from overwhelming the receiver, congestion control prevents senders from overwhelming the network. For an IS-IS adjacency, the network between two IS-IS neighbors is relatively limited in scope and consist in a link which is typically over-sized compared to the capability of the IS-IS speakers, but also includes components inside both routers such as a fabric switch, line card CPU and forwarding plane buffers which may experience congestion. This document proposes to use the AIMD (Additive Increase Multiplicative Decrease) algorithm to react to packet loss. Note that TCP Reno relies on the same algorithm.

Reliability relies on loss detection and recovery. IS-IS already has mechanisms to ensure the reliable transmission of LSPs. This is not changed by this document.

3. Flooding Parameters TLV

This document defines a new TLV called "Flooding Parameters TLV" that may be included in IIH PDUs. It allows the LSP receiver to advertise receiver related parameters and capabilities which allows the LSP sender to better adapt to the receiver.

Type: TBD1.

Length: variable, the size in octet of the Value field.

Value: a list of sub-TLVs.

Two sub-TLVs are defined in this document.

3.1. InterfaceLSPReceiveWindow sub-TLV

The sub-TLV InterfaceLSPReceiveWindow advertises the maximum number of un-acknowledged LSPs that the node can receive/process with no separation interval between LSPs.

Type: 1.

Length: 4 octets.

Value: number of un-acknowledged LSPs which can be sent back to back.

Note that if an LSP has not been acknowledged and is sent again, it does not count twice. The reason is that this LSP is assumed to be lost and hence not in a buffer at the receiver.

3.2. minimumInterfaceLSPTransmissionInterval sub-TLV

The sub-TLV minimumInterfaceLSPTransmissionInterval advertises the minimum interval, in micro-seconds, between LSPs arrivals which can be processed/received on this interface, after the maximum number of un-acknowledged LSPs has been sent.

Type: 2.

Length: 4 octets.

Value: minimum interval, in micro-seconds, between two consecutive LSPs sent after the receive window has been used.

4. Flow control

Flow control is about creating a control loop between a single transmitter and single receiver. This document proposes to use a mechanism similar to the TCP receive window to allow the receiver to govern the amount of data sent by the sender. This receive window indicates an allowed number of LSPs that the sender may transmit before receiving acknowledgment of those LSPs. This receive window, in units of LSPs, is advertised in the sub-TLV InterfaceLSPReceiveWindow.

4.1. Operation on a point to point interface

By sending the InterfaceLSPReceiveWindow sub-TLV with a value N, the node advertises to its IS-IS neighbor, its ability to receive a

maximum of N un-acknowledged LSPs from this neighbor, with no separation interval. This is akin to a reception window or sliding window in flow control. This value typically reflects the socket buffer size. Special care must be taken to let space for Hello and SNP PDUs if they share the same socket. In this case, this document suggests to advertise a Receive Window corresponding to half the size of the socket buffer.

By sending the minimumInterfaceLSPTransmissionInterval sub-TLV with a value T, the node advertises to its IS-IS neighbor, its ability to receive, after the receive window is full, LSPs separated by at least T micro-seconds from this neighbor.

The IS transmitter MUST NOT exceed these parameters. After having sent N un-acknowledged LSPs, it MUST send the following LSPs with an interval of at least T micro-seconds between each LSP.

Note however that if either the LSP transmitter or receiver does not adhere to these parameters, for example because of transient conditions, this causes no fatal condition to the operation of IS-IS. The worst case, the loss of LSP on the IS receiver, is already accounted for in [ISO10589] . As per [ISO10589] , after a few seconds, respectively 2 and 10 by default in [ISO10589] , neighbors will exchange PSNP (for point to point interface) or CSNP (for broadcast interface) and recover from the lost LSPs. This worst case (overrunning the receiver) should however be avoided as those additional seconds are impacting the network and the traffic as the LSDB is not fully synchronized. Hence it is better to err on the conservative side and to under-run the receiver rather than over-run it.

For a given IS-IS adjacency, the Flooding Parameters TLV does not need to be advertised in each IIH. The IS transmitter uses the

latest received value for each parameter (sub-TLV) until a new value is advertised by the IS receiver. Note however that IIH are not reliably exchanged, hence may never be received. For a parameter which has never been advertised, the IS transmitter use its local default value. That value SHOULD be configurable on a per node basis and MAY be configurable on a per interface basis.

4.2. Faster acknowledgments of LSPs

As per [ISO10589] , on point to point interfaces, the LSP receiver dynamically acknowledges the received LSPs by sending PSNP messages.

By acknowledging the LSPs before the InterfaceLSPReceiveWindow is exhausted, the receiver can achieve dynamic flow control and increase the flooding throughput without risking overloading any IS-IS router. If the InterfaceLSPReceiveWindow is large enough, the downstream flooding node can acknowledge a set of multiple LSPs up to the maximum size of a PSNP (90 LSPs) which allows dynamic flow control with limited or even no increase in the number of sent PSNPs.

In order to avoid reducing the throughput, the receiver should avoid letting the receive window exhaust. Therefore, the receiver SHOULD acknowledge the LSP more quickly than the default specified in [ISO10589] . This is beneficial both to the LSP sender which receives faster feedback and to the LSP receiver which has more time to acknowledge many LSPs before the sender times out and resend the LSP.

Receiver SHOULD reduce partialSNPInterval. The choice of this lower value is a local choice. It may depend on the (available) processing power of the node, the number of adjacencies, the requirement to synchronize the LSDB more quickly. 200 ms seems a reasonable value.

In addition to the timer based partialSNPInterval, the receiver SHOULD keep track of the number of unacknowledged LSPs per circuit and level. When this number exceeds a preset threshold LSP per PSNP (LPP), the receiver SHOULD immediately send a PSNP without waiting for the PSNP timer to expire. In case of a burst of LSPs, this allows for more frequent PSNPs, hence a faster feedback loop to the sender. In the absence of burst, the usual time-based PSNP approach comes into effect. This number SHOULD be lower than the advertised receive window InterfaceLSPReceiveWindow, e.g., InterfaceLSPReceiveWindow/2. This number SHOULD also be lower or equal to 90 as this is the maximum number of LSPs that can be acknowledged in a PSNP, hence waiting longer would not reduce the number of PSNPs sent but would delay the acknowledgements. Best performance is achieved when this number is an integer fraction of InterfaceLSPReceiveWindow. Based on experimental evidence, 15

unacknowledged LSPs is a right value assuming that

InterfaceLSPReceiveWindow is at least twice bigger (>30).

By deploying both the time-based and the threshold-based PSNP approaches, the receiver can be adaptive to both LSP bursts and infrequent LSP updates.

4.3. Operation on a LAN interface

On a LAN interface, an IS receiver will generally receive LSPs from multiple IS transmitters. Also the LSPs sent by a given IS transmitter is received by all of the IS receivers on that LAN. In this section, we clarify how the flooding parameters should be interpreted in the context of a LAN.

An IS receiver on a LAN will communicate its desired flooding parameters using a single Flooding Parameters TLV, copies of which will be received by all N transmitters. The flooding parameters sent by the IS receiver MUST be understood as instructions from the receiver to each transmitter about the desired maximum transmit characteristics of each transmitter. The receiver is aware that there are N transmitters that can send LSPs to the receiver LAN interface. The receiver might want to take that into account by advertising a higher value of InterfaceLSPTransmissionInterval on this LAN interface than what it would advertise on a point to point interface. When the transmitters receive the InterfaceLSPTransmissionInterval value advertised by the DIS receiver, the transmitters should rate limit LSPs according to the advertised flooding parameters. They should not apply any further interpretation to the flooding parameters advertised by the receiver.

A given IS transmitter will receive flooding parameter advertisements from N different Flooding Parameters TLVs, which may carry different flooding parameter values. A given transmitter SHOULD use the most convervative value on a per Flooding parameter basis. For example, if the transmitter receivers InterfaceLSPReceiveWindow from N IS-Is nodes on the LAN, it should use the smallest value.

In order for the InterfaceLSPReceiveWindow to be a useful parameter, an IS transmitter needs to be able to keep track of the number of un-acknowledged LSPs it has sent to a given IS receiver. On a LAN there is no explicit acknowledgment of the receipt of LSPs between a given IS transmitter and a given IS receiver. However, an IS transmitter on a LAN can infer whether any IS receiver on the LAN has requested retransmission of LSPs from the DIS, by monitoring PSNPs generated on the LAN. If no PSNPs have been generated on the LAN for a suitable period of time, then an IS transmitter can safely set the number of un-acknowledged LSPs to zero. Since this suitable period of time is

much higher than the fast acknowledgment of LSP defined in Section 4.2 , the substainable sending rate of LSP will be much slower on a LAN interface compared to a point to point interface. However, InterfaceLSPReceiveWindow is still very useful for the first LSPs sent and hence usefull for the faster flooding in case of a single node failure which requires to flood a relatively small number of LSPs.

A compliant implementation may choose to not support this operation on a LAN interface.

5. Congestion control

Whereas flow control prevents the sender from overwhelming the receiver, congestion control prevents senders from overwhelming the network. For an IS-IS adjacency, the network between two IS-IS neighbors is relatively limited in scope and includes a single link which is typically over-sized compared to the capability of the IS-IS speakers. It also includes components inside both routers such as a fabric switch, line cards CPU and forwarding plane buffers which may experience congestion. This document proposes one optional congestion control algorithm but implementations may choose a different one or none.

The congestion control algorithm defined in this document is largely inspired by the TCP congestion control algorithm RFC 5681 [RFC5681]. A congestion control algorithm is comprised of three elements : a slow start phase, a congestion avoidance phase, and a transition between the two.

The proposed algorithm uses a variable Congestion window 'cwin'. It plays the same role as Receive Window described before. The main difference is that CWin is dynamically changed according to the feedback obtained by the PSNPs.

5.1. Slow start

The goal of the slow start phase is to grow fast and try to estimate the effective link capacity.

The algorithm is circuit scoped. At the beginning of the slow start, the sender starts with:

- a congestion window (cwin) set to one. cwin := 1;

- a number of acked LSPs. acked_lsps := 0;

- a max seen bandwidth. max_bw := 0;

- a current rtt estimate. cur_rtt := NA;

Upon LSP sending, a sender records for said LSP the current time in time_sent and acked_lsps in acked_lsps_sent. This data is tied to each LSP.

Upon PSNP reception, a sender does the following:

```
cwin := min(cwin + nb_of_lsp_entries, rwin)
acked_lsps += nb_of_lsp_entries
max_diff := 0
max_bw := 0
for every LSP entry:
    time_to_ack := time_now - time_sent
    nb_acked := acked_lsps - acked_lsps_sent
    bw_est := nb_acked/time_to_ack
    max_bw := max(max_bw, bw_est)
    max_diff := max(max_diff, time_to_ack)

if cur_rtt == NA then cur_rtt = max_diff
else cur_rtt := 7/8 * cur_rtt + 1/8 * max_diff
```

Figure 1

Starting with the first PSNP, max_bw is checked every cur_rtt. Once it has stalled for 3 consecutive times, the congestion control algorithm transitions from slow start to congestion avoidance. There is bandwidth stalling when the bandwidth has not increased by at least 25% compared the last RTT. Note that this is similar to Google's BBR ([I-D.cardwell-iccrg-bbr-congestion-control]) slow start phase.

5.2. Congestion avoidance

The goal of the congestion avoidance phase is to try to stay close to the effective capacity of the link. For this, the algorithm estimates the maximum time taken by the receiver to acknowledge a LSP. If an LSP arrives slower than this delay, congestion is inferred and cwin is decreased.

Upon PSNP reception, a sender does the following:


```
cwin = min(cwin + N/congestion window, rwin)
rtt_est := 0
for every LSP entry:
    time_to_ack = time_now - time_sent
    rtt_est = max(rtt_est, time_to_ack)

if rtt_var == NA then rtt_var = rtt_est / 2
else rtt_var = 3/4 * rtt_var + 1/4 * abs(cur_rtt - rtt_est)

cur_rtt = 7/8 * cur_rtt + 1/8 * rtt_est
```

Figure 2

Every LSP is checked to be acked within cur_rtt + rtt_var. If an LSP arrives late, cwin is divided by two. This behaviour is similar to TCP retransmission timer defined in RFC 6298 [RFC6298]

Note: there is no need for a timer per LSP. A timer per RTT is enough. During an RTT, sent LSPs are recorded in a list list_1. Once the RTT is over, list_1 is kept and another list list_2 is used to store the next LSPs. LSPs are removed from the lists when acked. At the end of the second RTT, every LSP in list_1 should have been acked, so list_1 is checked to be empty. List_1 can then be reused for the next RTT.

If there is no transmitted LSP for a fixed period of time, e.g. 2 seconds, the sender switches back to the slow start phase.

5.3. Remarks

This algorithm's performance is dependent on the LPP value. Indeed, the smaller LPP is, the more information is available for the

congestion control algorithm to perform well. However, it also increases the resources spent on sending PSNPs, so a tradeoff must be made. This document recommends to use an LPP of 15 or less.

Note that this congestion control algorithm benefits from the extensions proposed in this document. The advertisement of a receive window from the receiver (Section 4) avoids the use of an arbitrary maximum value by the sender. The faster acknowledgment of LSP (Section 4.2) allows for a faster control loop and hence a faster increase of the congestion window in the absence of congestion.

6. Interaction with other LSP rate limiting mechanisms

[ISO10589] describes a mechanism that limits the rate at which LSPs from the same source system are sent out on interfaces. (See the description of the parameter

minimumBroadcastLSPTranLSPTransmissionInterval in section 7.3.15.6 of [ISO10589]). In practice, however, router vendors have implemented mechanisms that limit the rate of LSPs sent on a given interface. This is often configurable on a per-interface basis using 'lsp-interval' or 'lsp-pacing-interval' CLI configuration). The mechanism described in the current document extends the practice of limiting the rate of LSPs sent on a given interface, by using parameters advertised by the LSP receiver. When the mechanism described in the current document is used, the mechanism described in section 7.3.15.6 of [ISO10589] is not used.

7. Determining values to be advertised in the Flooding Parameters TLV

The values that a receiving IS advertises do not need to be close to perfection. It is OK to be too low and hence not to use the full bandwidth or CPU resources. It is OK to be too high during some situation and hence have the receiver drop some LSPs as the IS-IS protocol has mechanisms to recover. What is not OK is to flood multiple order of magnitudes slower than both nodes can achieve, or to consistently overload the receiver.

The values may not need to be dynamic as a form of dynamic is provided by the dynamic acknowledgment of LSPs in SNP messages. Acknowledgments provides a feedback loop on how fast/slower the LSPs are processed by the receiver. They also signal that the LSPs have

been processed by the receiver hence removed from receive window, explicitly signaling to the sender that more LSPs may be sent. By advertising relatively static parameters, we expect to produce overall flooding behavior similar to what might be achieved by manually configuring per-interface LSP rate limiting on all interfaces in the network. The advertised values may be based, for example, on an off line tests of the overall LSP processing speed for a particular set of hardware and the number of interfaces configured for IS-IS. With such a formula, the values advertised in the Flooding Parameters TLV would only change when additional IS-IS interfaces are configured.

The values may be updated dynamically, to reflect the relative change of load of the receiver, by improving the values when the receiver load is getting lower and degrading the values when the receiver load is getting higher. For example, if LSPs are regularly dropped, or the queue regularly comes close to being filled, then values may be too high. On the other hand, if the queue is barely used (by IS-IS), then values may be too low.

The values may also be absolute value reflecting relevant (averaged) hardware resources that are been monitored, typically the amount of buffer space used by incoming LSPs. In this case, care must be taken

when choosing the parameters influencing the values, in order to avoid undesirable or instable feedback loops. It would be undesirable to use a formula that depends, for example, on an active measurement of the instantaneous CPU load to modify the values advertised in the Flooding Parameters TLV. This could introduce feedback into the IGP flooding process that could produce unexpected behavior.

8. Operation considerations

As discussed in Section 4.3 , the solution is more effective on point to point adjacencies. Hence a broadcast interface (e.g. Ethernet) only shared by two IS-IS neighbhors should be configured as point to point in order to have a more effective flooding.

9. IANA Considerations

IANA is requested to allocate one TLV from the IS-IS TLV codepoint

registry.

Type	Description	IIH	LSP	SNP	Purge
TBD1	Flooding Parameters TLV	y	n	n	n

Figure 3

This document creates the following sub-TLV Registry:

Name: Sub-TLVs for TLV TBD1 (Flooding Parameters TLV).

Registration Procedure: Expert Review [RFC8126] .

Type	Description
0	Reserved
1	InterfaceLSPReceiveWindow
2	minimumInterfaceLSPTransmissionInterval
3-255	Unassigned

Table 1: Initial allocations

10. Security Considerations

Any new security issues raised by the procedures in this document depend upon the ability of an attacker to inject a false but

apparently valid SNP or IIH, the ease/difficulty of which has not been altered.

As with others TLV advertisements, the use of a cryptographic authentication as defined in [RFC5304] or [RFC5310] allows the authentication of the peer and the integrity of the message. As this document defines a TLV for SNP or IIH message, the relevant cryptographic authentication is for SNP and IIH message.

In the absence of cryptographic authentication, as IS-IS does not run over IP but directly over the link layer, it's considered difficult

to inject false SNP/IHH without having access to the link layer.

If a false SNP/IIH is sent with a Flooding Parameters TLV set to conservative values, the attacker can reduce the flooding speed between the two adjacent neighbors which can result in LSDB inconsistencies and transient forwarding loops. However, it is not significantly different than filtering or altering LSPDUs which would also be possible with access to the link layer. In addition, if the downstream flooding neighbor has multiple IGP neighbors, which is typically the case for reliability or topological reasons, it would receive LSPs at a regular speed from its other neighbors and hence would maintain LSDB consistency.

If a false SNP/IIH is sent with a Flooding Parameters TLV set to aggressive values, the attacker can increase the flooding speed which can either overload a node or more likely generate loss of LSPs. However, it is not significantly different than sending many LSPs which would also be possible with access to the link layer, even with cryptographic authentication enabled. In addition, IS-IS has procedures to detect the loss of LSPs and recover.

This TLV advertisement is not flooded across the network but only sent between adjacent IS-IS neighbors. This would limit the consequences in case of forged messages, and also limits the dissemination of such information.

11. Acknowledgments

The authors would like to thank Henk Smit, Sarah Chen, Xuesong Geng, Pierre Francois and Hannes Gredler for their reviews, comments and suggestions.

The authors would like to thank David Jacquet, Sarah Chen, and Qiangzhou Gao for the tests performed on commercial implementations and their identification of some limiting factors.

12. References

12.1. Normative References

[ISO10589]
International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, Nov 2002.

[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>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <https://www.rfc-editor.org/info/rfc6298>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

12.2. Informative References

[I-D.cardwell-iccrg-bbr-congestion-control]
Cardwell, N., Cheng, Y., Yeganeh, S. H., and V. Jacobson, "BBR Congestion Control", draft-cardwell-iccrg-bbr-congestion-control-00 (work in progress), July 2017.

[I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., Dontula, S., and G. S. Mishra, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-08 (work in progress), December 2020.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

Appendix A. Changes / Author Notes

[RFC Editor: Please remove this section before publication]

00: Initial version.

01: Two notes added in section 3 "Operation".

02: Refresh, no technical change.

03:

- Flooding Parameters TLV: name changed, advertised in both Hello and SNP rather than just Hello, contains sub-TLVs, parameters encoded in 4 octets.
- Terminology: upstream/downstream terms removed, in favor of terms from ISO specification (transmitter, receiver); burst-size rename to receive-window.
- Significant editorials changes.
- New section on the faster acknowledgment of LSPs.
- New section on the faster retransmission of lost LSPs.

04:

- Adding general introduction on flow control, congestion control, loss detection and recovery.
- Reorganizing sections as per the high level functions: flow control, congestion control, loss detection and recovery.

- Adding a section on congestion control.

05:

- Some editorials changes.
- Updating section "Faster acknowledgments of LSPs" following the IS-IS flooding performance tests presented during IETF 108.
- Updated IANA section (new registry).

06: Refresh, no technical change.

07:

- Precision that if a LSP is lost and resent, it does not count twice in the InterfaceLSPReceiveWindow.
- Title changed.
- Removed fast retransmissions of LSPs.
- Changed congestion control algorithm.
- Removed support of TLV in SNP.

Authors' Addresses

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Chris Bowers
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: cbowers@juniper.net

Jayesh J
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: jayeshj@juniper.net

Tony Li
Arista Networks
5453 Great America Parkway
Santa Clara, California 95054
USA

Email: tony.li@tony.li

Gunter Van de Velde
Nokia
Copernicuslaan 50
Antwerp 2018
Belgium

Email: gunter.van_de_velde@nokia.com

Guillaume Solignac
Orange

Email: guillaume.solignac@orange.com