Workgroup: Network Working Group
Internet-Draft:
draft-ietf-lsr-isis-fast-flooding-03
Published: 13 March 2023
Intended Status: Experimental
Expires: 14 September 2023
Authors: B. Decraene L. Ginsberg T. Li
Orange Cisco Systems Juniper Networks, Inc.
G. Solignac M. Karasek C. Bowers
Cisco Systems Juniper Networks, Inc.
G. Van de Velde P. Psenak T. Przygienda
Nokia Cisco Systems Juniper

IS-IS Fast Flooding

Abstract

Current Link State Protocol Data Unit (PDU) flooding rates are much slower than what modern networks can support. The use of IS-IS at larger scale requires faster flooding rates to achieve desired convergence goals. This document discusses the need for faster flooding, the issues around faster flooding, and some example approaches to achieve faster flooding. It also defines protocol extensions relevant to faster flooding.

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 14 September 2023.

Table of Contents

1. Introduction

Link state IGPs such as Intermediate-System-to-Intermediate-System (IS-IS) depend upon having consistent Link State Databases (LSDB) on all Intermediate Systems (ISs) in the network in order to provide correct forwarding of data packets. When topology changes occur, new/updated Link State PDUs (LSPs) are propagated network-wide. The speed of propagation is a key contributor to convergence time.

Historically, flooding rates have been conservative - on the order of 10s of LSPs/second. This is the result of guidance in the base specification [ISO10589] and early deployments when both CPU speeds and interface speeds were much slower and the scale of an area was much smaller than they are today.

As IS-IS is deployed in greater scale both in the number of nodes in an area and in the number of neighbors per node, the impact of the historic flooding rates becomes more significant. Consider the bringup or failure of a node with 1000 neighbors. This will result in a minimum of 1000 LSP updates. At typical LSP flooding rates used today (33 LSPs/second), it would take 30+ seconds simply to send the updated LSPs to a given neighbor. Depending on the diameter of the network, achieving a consistent LSDB on all nodes in the network could easily take a minute or more.

Increasing the LSP flooding rate therefore becomes an essential element of supporting greater network scale.

Improving the LSP flooding rate is complementary to protocol extensions that reduce LSP flooding traffic by reducing the flooding topology such as Mesh Groups [RFC2973] or Dynamic Flooding [I-D.ietf-lsr-dynamic-flooding] . Reduction of the flooding topology does not alter the number of LSPs required to be exchanged between two nodes, so increasing the overall flooding speed is still beneficial when such extensions are in use. It is also possible that the flooding topology can be reduced in ways that prefer the use of neighbors that support improved flooding performance.

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. Historical Behavior

The base specification for IS-IS [ISO10589] was first published in 1992 and updated in 2002. The update made no changes in regards to suggested timer values. Convergence targets at the time were on the order of seconds and the specified timer values reflect that. Here are some examples:

minimumLSPGenerationInterval - This is the minimum time interval between generation of Link State PDUs. A source Intermediate system shall wait at least this long before re-generating one of its own Link State PDUs.

The recommended value is 30 seconds.

minimumLSPTransmissionInterval - This is the amount of time an Intermediate system shall wait before further propagating another Link State PDU from the same source system.

The recommended value is 5 seconds.

partialSNPInterval - This is the amount of time between periodic action for transmission of Partial Sequence Number PDUs. It shall be less than minimumLSPTransmission-Interval.

The recommended value is 2 seconds.

Most relevant to a discussion of the LSP flooding rate is the recommended interval between the transmission of two different LSPs on a given interface.

For broadcast interfaces, [ISO10589] defined:

minimumBroadcastLSPTransmissionInterval - the minimum interval between PDU arrivals which can be processed by the slowest Intermediate System on the LAN.

The default value was defined as 33 milliseconds. It is permitted to send multiple LSPs "back-to-back" as a burst, but this was limited to 10 LSPs in a one second period.

Although this value was specific to LAN interfaces, this has commonly been applied by implementations to all interfaces though that was not the original intent of the base specification. In fact Section 12.1.2.4.3 states:

On point-to-point links the peak rate of arrival is limited only by the speed of the data link and the other traffic flowing on that link.

Although modern implementations have not strictly adhered to the 33 millisecond interval, it is commonplace for implementations to limit the flooding rate to an order of magnitude similar to the 33 ms value.

In the past 20 years, significant work on achieving faster convergence - more specifically sub-second convergence - has resulted in implementations modifying a number of the above timers in order to support faster signaling of topology changes. For example, minimumLSPGenerationInterval has been modified to support millisecond intervals, often with a backoff algorithm applied to prevent LSP generation storms in the event of a series of rapid oscillations.

However, the flooding rate has not been fundamentally altered.

4. Flooding Parameters TLV

This document defines a new Type-Length-Value tuple (TLV) called the "Flooding Parameters TLV" that may be included in IS to IS Hellos (IIH) or Partial Sequence Number PDUs (PSNPs). It allows IS-IS implementations to advertise flooding related parameters and capabilities which may be of use to the peer in support of faster flooding.

Type: 21

Length: variable, the size in octets of the Value field

Value: One or more sub-TLVs

Several sub-TLVs are defined in this document. The support of any sub-TLV is OPTIONAL.

For a given IS-IS adjacency, the Flooding Parameters TLV does not need to be advertised in each IIH or PSNP. An IS uses the latest received value for each parameter until a new value is advertised by the peer. However, as IIHs and PSNPs are not reliably exchanged, and may never be received, parameters SHOULD be sent even if there is no change in value since the last transmission. For a parameter which has never been advertised, an IS SHOULD use its local default value. That value SHOULD be configurable on a per node basis and MAY be configurable on a per interface basis.

4.1. LSP Burst Window sub-TLV

The LSP Burst Window sub-TLV advertises the maximum number of LSPs that the node can receive with no separation interval between LSPs.

Type: 1

Length: 4 octets

Value: number of LSPs that can be sent back to back.

4.2. LSP Transmission Interval sub-TLV

The LSP Transmission Interval sub-TLV advertises the minimum interval, in micro-seconds, between LSPs arrivals which can be 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 burst window has been used

The LSP Transmission Interval is an advertisement of the receiver's steady-state LSP reception rate.

4.3. LSPs Per PSNP sub-TLV

The LSP per PSNP (LPP) sub-TLV advertises the number of received LSPs that triggers the immediate sending of a PSNP to acknowledge them.

Type: 3

Length: 2 octets

Value: number of LSPs acknowledged per PSNP

A node advertising this sub-TLV with a value LPP MUST send a PSNP once LPP LSPs have been received and need to be acknowledged.

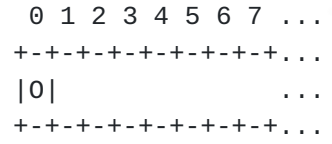
4.4. Flags sub-TLV

The sub-TLV Flags advertises a set of flags.

Type: 4

Length: Indicates the length in octets (1-8) of the Value field. The length SHOULD be the minimum required to send all bits that are set.

Value: List of flags.



When the O flag is set, the LSP will be acknowledged in the order they are received: a PSNP acknowledging N LSPs is acknowledging the N oldest LSPs received. The order inside the PSNP is meaningless. If the sender keeps track of the order of LSPs sent, this indication allows a fast detection of the loss of an LSP. This MUST NOT be used to trigger faster retransmission of LSP. This MAY be used to trigger a congestion signal.

4.5. Partial SNP Interval sub-TLV

The Partial SNP Interval sub-TLV advertises the amount of time in milliseconds between periodic action for transmission of Partial Sequence Number PDUs. This time will trigger the sending of a PSNP even if the number of unacknowledged LSPs received on a given interface does not exceed LPP (Section 4.3). The time is measured from the reception of the first unacknowldeged LSP.

Type: 5

Length: 2 octets

Value: partialSNPInterval in milliseconds

A node advertising this sub-TLV SHOULD send a PSNP at least once per Partial SNP Interval if one or more unacknowledged LSPs have been received on a given interface.

4.6. Receive Window sub-TLV

The Receive Window (RWIN) sub-TLV advertises the maximum number of unacknowledged LSPs that the node can receive.

Type: 6

Length: 2 octets

Value: maximum number of unacknowledged LSPs

4.7. Operation on a LAN interface

On a LAN interface, all LSPs are link-level multicasts. Each LSP sent will be received by all ISs on the LAN and each IS will receive LSPs from all transmitters. In this section, we clarify how the flooding parameters should be interpreted in the context of a LAN.

An LSP receiver on a LAN will communicate its desired flooding parameters using a single Flooding Parameters TLV, copies of which will be received by all transmitters. The flooding parameters sent by the LSP 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 multiple transmitters that can send LSPs to the receiver LAN interface. The receiver might want to take that into account by advertising more conservative values, e.g. a higher LSP Transmission Interval. When the transmitters receive the LSP Transmission Interval value advertised by a LSP 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 LSP transmitter will receive multiple flooding parameter advertisements from different receivers that may carry different flooding parameter values. A given transmitter SHOULD use the most convervative value on a per parameter basis. For example, if the transmitter receives multiple LSP Burst Window values, it should use the smallest value.

The Designated Intermediate System (DIS) plays a special role in the operation of flooding on the LAN as it is responsible for responding to PSNPs sent on the LAN circuit which are used to request LSPs that the sender of the PSNP does not have. If the DIS does not support faster flooding this will impact the maximum flooding speed which could occur on a LAN. Use of LAN priority to prefer a node which supports faster flooding in the DIS election may be useful.

NOTE: The focus of work used to develop the example algorithms discussed later in this document focused on operation over point to point interfaces. A full discussion of how best to do faster flooding on a LAN interface is therefore out of scope for this document.

5. Performance improvement on the receiver

This section defines two behaviors that SHOULD be implemented on the receiver.

5.1. Rate of LSP Acknowledgments

On point-to-point networks, PSNP PDUs provide acknowledgments for received LSPs. [ISO10589] suggests that some delay be used when sending PSNPs. This provides some optimization as multiple LSPs can be acknowledged in a single PSNP.

Faster LSP flooding benefits from a faster feedback loop. This requires a reduction in the delay in sending PSNPs.

The receiver SHOULD reduce its 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, and the requirement to synchronize the LSDB more quickly. 200 ms seems to be 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 of LSPs 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, giving faster feedback to the sender. Outside of the burst case, the usual time-based PSNP approach comes into effect. The LPP SHOULD also be less than or equal to 90 as this is the maximum number of LSPs that can be acknowledged in a PSNP at common MTU sizes, hence waiting longer would not reduce the number of PSNPs sent but would delay the acknowledgements. Based on experimental evidence, 15 unacknowledged LSPs is a good value assuming that the Receive Window is at least 30 and reasonably fast CPUs for both the transmitter and receiver. More frequent PSNPs gives the transmitter more feedback on receiver progress, allowing the transmitter to continue transmitting while not burdening the receiver with undue overhead.

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.

As PSNPs also consume link bandwidth, packet queue space, and protocol processing time on receipt, the increased sending of PSNPs should be taken into account when considering the rate at which LSPs can be sent on an interface.

5.2. Packet Prioritization on Receive

There are three classes of PDUs sent by IS-IS:

- *Hellos
- *LSPs
- *Complete Sequence Number PDUs (CSNPs) and PSNPs

Implementations today may prioritize the reception of Hellos over LSPs and SNPs in order to prevent a burst of LSP updates from triggering an adjacency timeout which in turn would require additional LSPs to be updated.

CSNPs and PSNPs serve to trigger or acknowledge the transmission of specified LSPs. On a point-to-point link, PSNPs acknowledge the receipt of one or more LSPs. For this reason, [ISO10589] specifies a delay (partialSNPInterval) before sending a PSNP so that the number of PSNPs required to be sent is reduced. On receipt of a PSNP, the set of LSPs acknowledged by that PSNP can be marked so that they do not need to be retransmitted.

If a PSNP is dropped on reception, the set of LSPs advertised in the PSNP cannot be marked as acknowledged and this results in needless retransmissions that will further delay transmission of other LSPs that have yet to be transmitted. It may also make it more likely that a receiver becomes overwhelmed by LSP transmissions.

It is therefore RECOMMENDED that implementations prioritize the receipt of Hellos and then SNPs over LSPs. Implementations MAY also prioritize IS-IS packets over other less critical protocols.

6. Congestion and Flow Control

6.1. 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 [RFC9293] and relies on the flow control, congestion control, and reliability mechanisms of the protocol.

Flow control creates a control loop between a transmiter and a receiver so that the transmitter does not overwhelm the receiver. TCP provides a mean for the receiver to govern the amount of data sent by the sender through the use of a sliding window.

Congestion control creates multiple interacting control loops between multiple transmitters and multiple receivers to prevent the transmitters from overwhelming the overall network. For an IS-IS adjacency, the network between two IS-IS neighbors is relatively limited in scope and consist of a link that is typically over-sized compared to the capability of the IS-IS speakers, but may also includes components inside both routers such as a switching fabric, line card CPU, and forwarding plane buffers that may experience congestion. These resources may be shared across multiple IS-IS adjacencies for the system and it is the responsibility of congestion control to ensure that these are shared reasonably.

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

The following two sections provides examples of Flow and/or Congestion control algorithms as examples that may be implemented by taking advantage of the extensions defined in this document. They are non-normative. The IS-IS extensions defined in Section 4 and Section 5 are generic and are designed to support different sender-side algorithms. A sender can unilaterally choose a different algorithm to use.

6.2. Congestion and Flow Control algorithm 1

6.2.1. Flow control

A flow control mechanism creates a control loop between a single instance of a transmitter and a single receiver. This section uses 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

('rwin') indicates an allowed number of LSPs that the sender may transmit before waiting for an acknowledgment. The size of the receive window, in units of LSPs, is initialized with the value advertised by the receiver in the Receive Window sub-TLV. If no value is advertised, the transmitter should initialize rwin with its own local value.

When the transmitter sends a set of LSPs to the receiver, it subtracts the number of LSPs sent from rwin. If the transmitter receives a PSNP, then rwin is incremented for each acknowledged LSP. The transmitter must ensure that the value of rwin never goes negative.

6.2.1.1. Operation on a point to point interface

By sending the LSP Burst Window sub-TLV, a node advertises to its neighbor its ability to receive that many un-acknowledged LSPs from the neighbor, with no separation interval. This is akin to a receive window or sliding window in flow control. In some implementations, this value should reflect the IS-IS socket buffer size. Special care must be taken to leave space for CSNP and PSNP (SNP) PDUs and IIHs if they share the same input queue. In this case, this document suggests advertising an LSP Burst Window corresponding to half the size of the IS-IS input queue.

By advertising an LSP Transmission Interval sub-TLV, a node advertises its ability to receive LSPs separated by at least the advertised value, outside of LSP bursts.

The LSP transmitter MUST NOT exceed these parameters. After having sent a full burst of un-acknowledged LSPs, it MUST send the following LSPs with an LSP Transmission Interval between LSP arrivals. For CPU scheduling reasons, this rate may be averaged over a small period e.g. 10 to 30ms.

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. In the worst case, an LSP is lost at the receiver and this situation is already remedied by mechanisms in [ISO10589] . After a few seconds, neighbors will exchange PSNPs (for point to point interfaces) or CSNPs (for broadcast interfaces) and recover from the lost LSPs. This worst case should be avoided as those additional seconds impact convergence time 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.

6.2.1.2. Operation on a broadcast LAN interface

In order for the LSP Burst Window to be a useful parameter, an LSP transmitter needs to be able to keep track of the number of un-acknowledged LSPs it has sent to a given LSP receiver. On a LAN there is no explicit acknowledgment of the receipt of LSPs between a given LSP transmitter and a given LSP receiver. However, an LSP transmitter on a LAN can infer whether any LSP 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 LSP 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 LSPs defined in Section 5.1, the sustainable transmission rate of LSPs will be much slower on a LAN interface than on a point to point interface. The LSP Burst Window is still very useful for the first burst of LSPs sent, especially in the case of a single node failure that requires the flooding of a relatively small number of LSPs.

6.2.2. 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.

This section describes one sender-side congestion control algorithm largely inspired by the TCP congestion control algorithm [RFC5681].

The proposed algorithm uses a variable congestion window 'cwin'. It plays a role similar to the receive window described above. The main difference is that cwin is dynamically changed according to various events described below.

6.2.2.1. Core algorithm

In its simplest form, the congestion control algorithm looks like the following:

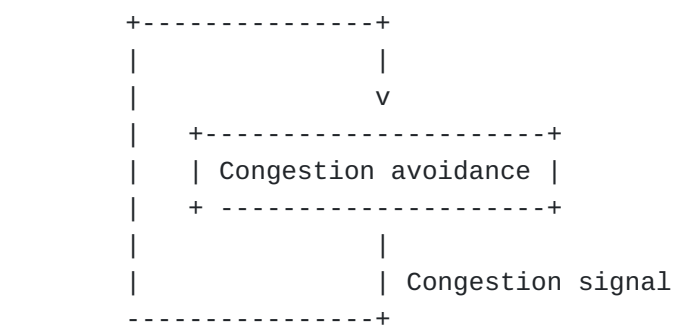


Figure 1

The algorithm starts with cwin := LPP + 1. In the congestion avoidance phase, cwin increases as LSPs are acked: for every acked LSP, cwin += 1 / cwin. Thus, the sending rate roughly increases linearly with the RTT. Since the RTT is low in many IS-IS deployments, the sending rate can reach fast rates in short periods of time.

When updating cwin, it must not become higher than the number of LSPs waiting to be sent, otherwise the sending will not be paced by the receiving of acks. Said differently, tx pressure is needed to maintain and increase cwin.

When the congestion signal is triggered, cwin is set back to its initial value and the congestion avoidance phase starts again.

6.2.2.2. Congestion signals

The congestion signal can take various forms. The more reactive the congestion signals, the less LSPs will be lost due to congestion. However, congestion signals too aggressive will cause a sender to keep a very low sending rate even without actual congestion on the path.

Two practical signals are given hereafter.

Timers: when receiving acknowledgements, a sender estimates the acknowledgement time of the receiver. Based on this estimation, it can infer that a packet was lost, and infer congestion on the path.

There can be a timer per LSP, but this can become costly for implementations. It is possible to use only a single timer t1 for every LSPs: during t1, 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 t1 period, 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.

There are multiple strategies to set the timeout value t1. It should be based on measures of the maximum acknowledgement time (MAT) of each PSNPs. The simplest one is to use a exponential moving average of the MATs, like [RFC6298]. A more elaborate one is to take a running maximum of the MATs over a period of time of a few seconds. This value should include a margin of error to avoid false positives (e.g. estimated MAT measure variance) which would have a significant impact on performance.

Reordering: a sender can record its sending order and check that acknowledgements arrive on the same order than LSPs. This makes an additional assumption and should ideally be backed up by a confirmation by the receiver that this assumption stands. The O flag defined in Section 4.4 serves this purpose.

6.2.2.3. Refinement 1

With the algorithm presented above, if congestion is detected, cwin goes back to its initial value, and does not use the information gathered in previous congestion avoidance phases.

It is possible to use a fast recovery phase once congestion is detected, to avoid going through this linear rate of growth from scratch. When congestion is detected, a fast recovery threshold frthresh is set to frthresh := cwin / 2. In this fast recovery phase, for every acked LSP, cwin += 1. Once cwin reaches frthresh, the algorithm goes back to the congestion avoidance phase.

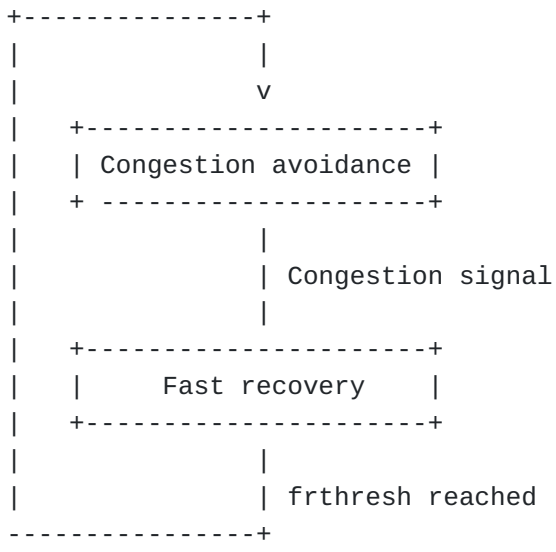


Figure 2

6.2.2.4. Refinement 2

The rates of increase were inspired from TCP [RFC5681], but it is possible that a different rate of increase for cwin in the congestion avoidance phase actually yields better results due to the low RTT values in most IS-IS deployments.

6.2.2.5. 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. If an LSP Burst Window is advertised, LPP SHOULD be lower and the best performance is achieved when LPP is an integer fraction of the LSP Burst Window.

Note that this congestion control algorithm benefits from the extensions proposed in this document. The advertisement of a receive window from the receiver (Section 6.2.1) avoids the use of an arbitrary maximum value by the sender. The faster acknowledgment of LSPs (Section 5.1) allows for a faster control loop and hence a faster increase of the congestion window in the absence of congestion.

6.2.3. Pacing

As discussed in [RFC9002], Section 7.7 a sender SHOULD pace sending of all in-flight LSP based on input from the congestion controller.

Sending multiple packets without any delay between them creates a packet burst that might cause short-term congestion and losses. Senders MUST either use pacing or limit such bursts. Senders SHOULD limit bursts to the initial congestion window. A sender with knowledge that the receiver can absorb larger bursts, such as by receiving the LSP Burst Window sub-TLV from this receiver MAY use a higher limit.

Senders can implement pacing as they choose. A perfectly paced sender spreads packets exactly evenly over time. For a window-based congestion controller, such as the one in this section, that rate can be computed by averaging the congestion window over the RTT. Expressed as an inter-packet interval in units of time:

```
interval = ( smoothed_rtt / congestion_window ) / N
```

Using a value for N that is small, but at least 1 (for example, 1.25) ensures that variations in RTT do not result in underutilization of the congestion window.

Practical considerations, such as scheduling delays and computational efficiency, can cause a sender to deviate from this rate over time periods that are much shorter than an RTT.

One possible implementation strategy for pacing uses a leaky bucket algorithm, where the capacity of the "bucket" is limited to the maximum burst size and the rate the "bucket" fills is determined by the above function.

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

The values that a receiver advertises do not need to be perfect. If the values are too low then the transmitter will not use the full bandwidth or available CPU resources. If the values are too high then the receiver may drop some LSPs during the first RTT and this loss will reduce the usable receive window and the protocol mechanisms will allow the adjacency to recover. Flooding several orders of magnitude slower than both nodes can achieve will hurt performance, as will consistently overloading the receiver.

The values advertised need not be dynamic as feedback is provided by the acknowledgment of LSPs in SNP messages. Acknowledgments provide a feedback loop on how fast the LSPs are processed by the receiver. They also signal that the LSPs can be 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 offline 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 if the queue regularly comes close to being filled, then the 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 average 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.

6.2.5. Operation considerations

As discussed in Section 4.7, 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.

6.3. Congestion Control algorithm 2

This section describes a congestion control algorithm based on performance measured by the transmitter without dependance on signaling from the receiver.

6.3.1. Router Architecture Discussion

(The following description is an abstraction - implementation details vary.)

Existing router architectures may utilize multiple input queues. On a given line card, IS-IS PDUs from multiple interfaces may be placed in a rate limited input queue. This queue may be dedicated to IS-IS PDUs or may be shared with other routing related packets.

The input queue may then pass IS-IS PDUs to a "punt queue" which is used to pass PDUs from the data plane to the control plane. The punt queue typically also has controls on its size and the rate at which packets will be punted.

An input queue in the control plane may then be used to assemble PDUs from multiple linecards, separate the IS-ISs PDU from other types of packets, and place the IS-IS PDUs in an input queue dedicated to the IS-IS protocol.

The IS-IS input queue then separates the IS-IS PDUs and directs them to an instance specific processing queue. The instance specififc processing queue may then further separate the IS-IS PDUs by type (IIHs, SNPs, and LSPs) so that separate processing threads with varying priorities may be employed to process the incoming PDUs.

In such an architecture, it may be difficult for IS-IS in the control plane to accurately track the state of the various input queues and determine what value should be advertised as a current receive window.

The following section describes a congestion control algorithm based on performance measured by the transmitter without dependance on signaling from the receiver.

6.3.2. Transmitter Based Flow Control

The congestion control algorithm described in this section does not depend upon direct signaling from the receiver. Instead it adapts the tranmsmission rate based on measurement of the actual rate of acknowledgments received.

When flow control is necessary, it can be implemented based on knowledge of the current flooding rate and the current acknowledgement rate. Such an algorithm is a local matter and there is no requirement or intent to standardize an algorithm. There are a number of aspects which serve as guidelines which can be described.

A maximum target LSP transmission rate (LSPTxMax) SHOULD be configurable. This represents the fastest LSP transmission rate which will be attempted. This value SHOULD be applicable to all interfaces and SHOULD be consistent network wide.

When the current rate of LSP transmission (LSPTxRate) exceeds the capabilities of the receiver, the flow control algorithm needs to quickly and aggressively reduce the LSPTxRate. Slower responsiveness is likely to result in a larger number of retransmissions which can introduce much larger delays in convergence.

Dynamic adjustment of the rate of LSP transmission (LSPTxRate) upwards (i.e., faster) SHOULD be done less aggressively and only be done when the neighbor has demonstrated its ability to sustain the current LSPTxRate.

The flow control algorithm MUST NOT assume the receive performance of a neighbor are static, i.e., it MUST handle transient conditions which result in a slower or faster receive rate on the part of a neighbor.

The flow control algorithm SHOULD consider the expected delay time in receiving an acknowledgment. It therefore incorporates the neighbor partialSNPInterval(Section 4.5) to help determine whether acknowlegments are keeping pace with the rate of LSPs transmitted. In the absence of an advertisement of partialSNPInterval a locally configured value can be used.

7. IANA Considerations

7.1. Flooding Parameters TLV

IANA has made the following temporary allocation from the IS-IS TLV codepoint registry.

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

Figure 3

7.2. Registry: Sub-TLVs for TLV 21 (Flooding Parameters TLV)

This document creates the following sub-TLV Registry:

Name: Sub-TLVs for TLV 21 (Flooding Parameters TLV).

Registration Procedure(s): Expert Review

Expert(s): TBD

Reference: TBD

Type	Description
0	Reserved
1	LSP Burst Window
2	LSP Transmission Interval
3	LSPs Per PSNP
4	Flags
5	Partial SNP Interval
6	Receive Window
7-255	Unassigned

Table 1: Initial allocations

7.3. Registry: Flooding Parameters Flags Bits

This document also requests IANA to create a new registry for assigning Flag bits advertised in the Flags sub-TLV.

Name: Flooding Parameters Flags Bits.

Registration Procedure:

Expert Review Expert(s): TBD

Bit #	Description
0	O Flag

8. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589] , [RFC5304] , and [RFC5310] . These documents describe mechanisms that provide the authentication and integrity of IS-IS PDUs, including SNPs and IIHs. These authentication mechanisms are not altered by this document.

With the cryptographic mechanisms described in [RFC5304] and [RFC5310] , an attacker wanting to advertise an incorrect Flooding Parameters TLV would have to first defeat these mechanisms.

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 LSPs 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.

9. Contributors

The following people gave a substantial contribution to the content of this document and should be considered as coauthors:

*Acee Lindem, Cisco Systems, acee@cisco.com

*Jayesh J, Juniper Networks, jayeshj@juniper.net

10. 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.

11. References

11.1. Normative References

[ISO10589] ISO, "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, November 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>.

[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>.

11.2. Informative References

[I-D.ietf-lsr-dynamic-flooding]
Li, T., Przygienda, T., Psenak, P., Ginsberg, L., Chen, H., Jalil, L., Dontula, S., and G. S. Mishra, "Dynamic Flooding on Dense Graphs", Work in Progress, Internet-Draft, draft-ietf-lsr-dynamic-flooding-12, 24 February

2023, <https://datatracker.ietf.org/doc/html/draft-ietf-lsr-dynamic-flooding-12>.

[RFC2973] Balay, R., Katz, D., and J. Parker, "IS-IS Mesh Groups", RFC 2973, DOI 10.17487/RFC2973, October 2000, <https://www.rfc-editor.org/info/rfc2973>.

[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>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/info/rfc9002>.

[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>.

Appendix A. Changes / Author Notes

[RFC Editor: Please remove this section before publication]

IND 00: Initial version.

WG 00: No change.

WG 01: IANA allocated code point.

WG 02: No change.

WG 03:

- *Pacing section added (taken from RFC 9002).
- *Some text borrowed from RFC 9002 (QUIC Loss Detection and Congestion Control).
- *Considerations on the special role of the DIS.
- *Editorial changes.

Appendix B. Issues for Further Discussion

[RFC Editor: Please remove this section before publication]

This section captures issues which the authors either have not yet had time to address or on which the authors have not yet reached consensus. Future revisions of this document may include new/altered text relevant to these issues.

There are no open issues at this time.

Authors' Addresses

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Les Ginsberg
Cisco Systems
821 Alder Drive
Milpitas, CA 95035
United States of America

Email: ginsberg@cisco.com

Tony Li
Juniper Networks, Inc.

Email: tony.li@tony.li

Guillaume Solignac

Email: gsoligna@protonmail.com

Marek Karasek
Cisco Systems
Pujmanove 1753/10a, Prague 4 - Nusle
10 14000 Prague
Czech Republic

Email: mkarasek@cisco.com

Chris Bowers
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
United States of America

Email: cbowers@juniper.net

Gunter Van de Velde
Nokia
Copernicuslaan 50
2018 Antwerp
Belgium

Email: gunter.van_de_velde@nokia.com

Peter Psenak
Cisco Systems
Apollo Business Center Mlynske nivy 43
821 09 Bratislava
Slovakia

Email: ppsenak@cisco.com

Tony Przygienda
Juniper
1137 Innovation Way
Sunnyvale, Ca
United States of America

Email: prz@juniper.net