Stream: Internet Engineering Task Force (IETF)
RFC: 9681
Category: Experimental
Published: November 2024
ISSN: 2070-1721
Authors: B. Decraene, *Orange*; L. Ginsberg, *Cisco Systems*; T. Li, *Juniper Networks, Inc.*; G. Solignac; M. Karasek, *Cisco Systems*; G. Van de Velde, *Nokia*; T. Przygienda, *Juniper*

RFC 9681
IS-IS Fast Flooding

Abstract

Current Link State 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

Copyright Notice

Table of Contents

1. Introduction

Link state IGPs such as Intermediate System to Intermediate System (IS-IS) depend upon having consistent Link State Databases (LSDBs) 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.

IS-IS base specification [ISO10589] does not use flow or congestion control but static flooding rates. Historically, flooding rates have been conservative -- on the order of tens of LSPs per second. This is the result of guidance in the base specification and early deployments when the CPU and interface speeds were much slower and the area scale 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 bring-up 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 per second), it would take more than 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.

Therefore, increasing the LSP flooding rate 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 [RFC9667]. 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.

With the goal of supporting faster flooding, this document introduces the signaling of additional flooding related parameters (Section 4), specifies some performance improvements on the receiver (Section 5) and introduces the use of flow and/or congestion control (Section 6).

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 regenerating one of its own Link State PDUs. [...]
>
> A reasonable value is 30 s.
>
> minimumLSPTransmissionInterval - This is the amount of time an Intermediate system shall wait before further propagating another Link State PDU from the same source system. [...]
>
> A reasonable value is 5 s.
>
> partialSNPInterval - This is the amount of time between periodic action for transmission of Partial Sequence Number PDUs. It shall be less than minimumLSPTransmissionInterval. [...]
>
> A reasonable value is 2 s.

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] states:

> minimumBroadcastLSPTransmissionInterval indicates 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 of [ISO10589] 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 the same order of magnitude: tens of milliseconds, and not the single digits or fractions of milliseconds that are needed today.

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 rapid successive oscillations.

However, the flooding rate has not been fundamentally altered.

4. Flooding Parameters TLV

This document defines a new Type-Length-Value (TLV) tuple called the "Flooding Parameters TLV" that may be included in IS-IS Hellos (IIHs) or Partial Sequence Number PDUs (PSNPs). It allows IS-IS implementations to advertise flooding-related parameters and capabilities that may be used by the peer to support 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 that has never been advertised, an IS uses 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 Size Sub-TLV

The LSP Burst Size sub-TLV advertises the maximum number of LSPs that the node can receive without an intervening delay between LSP transmissions.

Type: 1
Length: 4 octets
Value: number of LSPs that can be received back to back

4.2. LSP Transmission Interval Sub-TLV

The LSP Transmission Interval sub-TLV advertises the minimum interval, in microseconds, between LSPs arrivals that can be sustained on this receiving interface.

Type: 2
Length: 4 octets
Value: minimum interval, in microseconds, between two consecutive LSPs received after LSP Burst Size LSPs have been received

The LSP Transmission Interval is an advertisement of the receiver's sustainable LSP reception rate. This rate may be safely used by a sender that does not support the flow control or congestion algorithm. It may also be used as the minimal safe rate by flow control or congestion algorithms in unexpected cases, e.g., when the receiver is not acknowledging LSPs anymore.

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 for 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

```
 0 1 2 3 4 5 6 7 ...
+-+-+-+-+-+-+-+-+...
|O|             ...
+-+-+-+-+-+-+-+-+...
```

An LSP receiver sets the O-flag (Ordered acknowledgment) to indicate to the LSP sender that it will acknowledge the LSPs in the order as 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 for fast detection of the loss of an LSP. This **MUST NOT** be used to alter the retransmission timer for any LSP. This **MAY** be used to trigger a congestion signal.

4.5. PSNP Interval Sub-TLV

The PSNP Interval sub-TLV advertises the amount of time in milliseconds between periodic action for transmission of PSNPs. 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 unacknowledged LSP.

Type: 5
Length: 2 octets
Value: partialSNPInterval in milliseconds

A node advertising this sub-TLV **SHOULD** send a PSNP at least once per PSNP 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 for a given adjacency.

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, which will be received by all LSP transmitters. The flooding parameters sent by the LSP receiver **MUST** be understood as instructions from the LSP receiver to each LSP 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 an 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 include different flooding parameter values. A given transmitter **SHOULD** use the most conservative value on a per-parameter basis. For example, if the transmitter receives multiple LSP Burst Size 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 that 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 that could occur on a LAN. Use of LAN priority to prefer a node that 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, PSNPs provide acknowledgments for received LSPs. [ISO10589] suggests using some delay when sending PSNPs. This provides some optimization as multiple LSPs can be acknowledged by a single PSNP.

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

For the generation of PSNPs, the receiver **SHOULD** use a partialSNPInterval smaller than the one defined in [ISO10589]. 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 the case of a burst of LSPs, this allows more frequent PSNPs, giving faster feedback to the sender. Outside of the burst case, the usual timer-based PSNP approach comes into effect.

The smaller the LPP is, the faster the feedback to the sender and possibly the higher the rate if the rate is limited by the end-to-end RTT (link RTT + time to acknowledge). This may result in an increase in the number of PSNPs sent, which may increase CPU and IO load on both the sender and receiver. The LPP should 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 acknowledgments. LPP should not be chosen too high as the congestion control starts with a congestion window of LPP + 1. Based on experimental evidence, 15 unacknowledged LSPs is a good value, assuming that the Receive Window is at least 30. More frequent PSNPs give 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 timer-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
- SNPs (Complete Sequence Number PDUs (CSNPs) and PSNPs)

Implementations today may prioritize the reception of Hellos over LSPs and Sequence Number PDUs (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 further delay transmission of other LSPs that are yet to be transmitted. It may also make it more likely that a receiver becomes overwhelmed by LSP transmissions.

Therefore, implementations **SHOULD** prioritize IS-IS PDUs on the way from the incoming interface to the IS-IS process. The relative priority of packets in decreasing order **SHOULD** be: Hellos, SNPs, and LSPs. Implementations **MAY** also prioritize IS-IS packets over other protocols, which are less critical for the router or network, less sensitive to delay, or more bursty (e.g., BGP).

6. Congestion and Flow Control

6.1. Overview

Ensuring the goodput between two entities is a Layer 4 responsibility as per the OSI model. A typical example is the TCP protocol defined in [RFC9293] that provides flow control, congestion control, and reliability.

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

Congestion control prevents the set of transmitters from overwhelming the path of the packets between two IS-IS implementations. This path typically includes a point-to-point link between two IS-IS neighbors, which is usually oversized compared to the capability of the IS-IS speakers, but potentially also includes some internal elements inside each neighbor such as 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.

Sections 6.2 and 6.3 provide two flow and/or congestion control algorithms that may be implemented by taking advantage of the extensions defined in this document. The signal that these IS-IS extensions (defined in Sections 4 and 5) provide is generic and is designed to support different sender-side algorithms. A sender can unilaterally choose a different algorithm to use.

6.2. Congestion and Flow Control Algorithm

6.2.1. Flow Control

A flow control mechanism creates a control loop between a single 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 locally configured value for this receiver.

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.

The RWIN value is of importance when the RTT is the limiting factor for the throughput. In this case, the optimal size is the desired LSP rate multiplied by the RTT. The RTT is the addition of the link RTT plus the time taken by the receiver to acknowledge the first received LSP in its PSNP.

The values 50 or 100 may be reasonable default numbers for RWIN. As an example, an RWIN of 100 requires a control plane input buffer of 150 kbytes per neighbor (assuming an IS-IS MTU of 1500 octets) and limits the throughput to 10000 LSPs per second and per neighbor for a link RTT of 10 ms. With the same RWIN, the throughput limitation is 2000 LSPs per second when the RTT is 50 ms. That's the maximum throughput assuming no other limitations such as CPU limitations.

Equally, RTT is of importance for the performance. That is why the performance improvements on the receiver specified in Section 5 are important to achieve good throughput. If the receiver does not support those performance improvements, in the worst case (small RWIN and high RTT) the throughput will be limited by the LSP Transmission Interval as defined in Section 4.2.

6.2.1.1. Operation on a Point-to-Point Interface

By sending the Receive Window sub-TLV, a node advertises to its neighbor its ability to receive that many unacknowledged LSPs from the neighbor. 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 CSNPs, PSNPs, and IIHs if they share the same input queue. In this case, this document suggests advertising an LSP Receive 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.

By advertising an LSP Burst Size sub-TLV, a node advertises its ability to receive that number of LSPs back to back.

The LSP transmitter **MUST NOT** exceed these parameters. After having sent a full burst of LSPs, it **MUST** send the subsequent LSPs with a minimum of LSP Transmission Interval between LSP transmissions. For CPU scheduling reasons, this rate **MAY** be averaged over a small period, e.g., 10-30 ms.

If either the LSP transmitter or receiver does not adhere to these parameters, for example, because of transient conditions, this doesn't result in a fatal condition for IS-IS operation. 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 since 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

Flow and congestion control on a LAN interface is out of scope for this document.

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 that is typically oversized compared to the capability of the IS-IS speakers. In situations where the probability of LSP drop is low, flow control (Section 6.2.1) is expected to give good results, without the need to implement congestion control. Otherwise, adding congestion control will help handling congestion of LSPs in the receiver.

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 adjusted dynamically according to various events described below.

6.2.2.1. Core Algorithm

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

```
+---------------+
|               |
|               v
|   +----------------------+
|   | Congestion avoidance |
|   + ---------------------+
|               |
|               | Congestion signal
----------------+
```

Figure 1

The algorithm starts with cwin = cwin0 = LPP + 1. In the congestion avoidance phase, cwin increases as LSPs are acked: for every acked LSP, cwin += 1 / cwin without exceeding RWIN. When LSPs are exchanged, cwin LSPs will be acknowledged in 1 RTT, meaning cwin(t) = t/RTT + cwin0. 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, transmission 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 fewer LSPs will be lost due to congestion. However, overly aggressive congestion signals will cause a sender to keep a very low sending rate even without actual congestion on the path.

Two practical signals are given below.

1. Delay: When receiving acknowledgments, a sender estimates the acknowledgment time of the receiver. Based on this estimation, it can infer that a packet was lost and that the path is congested.

   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 all 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 measurements of the maximum acknowledgment time (MAT) of each PSNP. Using three times the RTT is the simplest strategy; alternatively, an exponential moving average of the MATs, as described in [RFC6298], can be used. A more elaborate one is to take a running maximum of the MATs over a period 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.
2. Loss: if the receiver has signaled the O-flag (see Section 4.4), a sender **MAY** record its sending order and check that acknowledgments arrive in the same order. If not, some LSPs are missing, and this **MAY** be used to trigger a congestion signal.

6.2.2.3. Refinement

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 and 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.

```
+---------------+
|               |
|               v
|    +----------------------+
|    | Congestion avoidance |
|    + ---------------------+
|               |
|               | Congestion signal
|               |
|    +----------------------+
|    |     Fast recovery    |
|    +----------------------+
|               |
|               | frthresh reached
----------------+
```

Figure 2

6.2.2.4. Remarks

This algorithm's performance is dependent on the LPP value. Indeed, the smaller the 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 trade-off must be made. This document recommends using an LPP of 15 or less. If a Receive Window is advertised, LPP **SHOULD** be lower, and the best performance is achieved when LPP is an integer fraction of the Receive 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 LSPs 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 LSP Burst Size.

Senders can implement pacing as they choose. A perfectly paced sender spreads packets 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 = (SRTT / cwin) / N

SRTT is the Smoothed Round-Trip Time [RFC6298].

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 that 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 slower than both nodes can support will hurt performance as will consistently overloading the receiver.

6.2.4.1. Static Values

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 the 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 could be based, for example, on 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.

Static values are dependent on the CPU generation, class of router, and network scaling, typically the number of adjacent neighbors. Examples at the time of publication are provided below. The LSP Burst Size could be in the range 5 to 20. From a router perspective, this value typically depends on the queue(s) size(s) on the I/O path from the packet forwarding engine to the control plane, which is very platform-dependent. It also depends upon how many IS-IS neighbors share this I/O path, as typically all neighbors will send the same LSPs at the same time. It may also depend on other incoming control plane traffic that is sharing that I/O path, how bursty they are, and how many incoming IS-IS packets are prioritized over other incoming control plane traffic. As indicated in Section 3, the historical behavior from [ISO10589] allows a value of 10; hence, 10 seems conservative. From a network operation perspective, it would be beneficial for the burst size to be equal to or higher than the number of LSPs that may be originated by a single failure. For a node failure, this is equal to the number of IS-IS neighbors of the failed node. The LSP Transmission Interval could be in the range of 1 ms to 33 ms. As indicated in Section 3, the historical behavior from [ISO10589] is 33 ms; hence, 33 ms is conservative. The LSP Transmission Interval is an advertisement of the receiver's sustainable LSP reception rate taking into account all aspects and particularly the control plane CPU and the I/O bandwidth. It's expected to improve (hence, decrease) as hardware and software naturally improve over time. It should be chosen conservatively, as this rate may be used by the sender in all conditions -- including the worst conditions. It's also not a bottleneck as the flow control algorithm may use a higher rate in good conditions, particularly when the receiver acknowledges quickly, and the receive window is large enough compared to the RTT. LPP could be in the range of 5 to 90 with a proposed 15. A smaller value provides faster feedback at the cost of the small overhead of more PSNP messages. PartialSNPInterval could be in the range 50 to 500 ms with a proposed value of 200 ms. One may distinguish the value used locally from the value signaled to the sender. The value used locally benefits from being small but is not expected to be the main parameter to improve performance. It depends on how fast the IS-IS flooding process may be scheduled by the CPU. Even when the receiver CPU is busy, it's safe because it will naturally delay its acknowledgments, which provides a negative feedback loop. The value advertised to the sender should be conservative (high enough) as this value could be used by the sender to send some LSPs rather than keep waiting for acknowledgments. Receive Window could be in the range of 30 to 200 with a proposed value of 60. In general, the larger the better the performance on links with high RTT. The higher that number and the higher the number of IS-IS neighbors, the higher the use of control plane memory, so it's mostly dependent on the amount of memory, which may be dedicated to IS-IS flooding and the number of IS-IS neighbors. From a memory usage perspective (a priori), one could use the same value as the TCP receive window, but the value advertised should not be higher than the buffer of the "socket" used.

6.2.4.2. Dynamic Values

To reflect the relative change of load on the receiver, the values may be updated dynamically by improving the values when the receiver load is getting lower and by 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 the values may be too low.

Alternatively, the values may be computed to reflect the relevant average hardware resources, e.g., 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 unstable feedback loops. For example, it would be undesirable to use a formula that depends 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. Operational 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 neighbors should be configured as point-to-point in order to have more effective flooding.

6.3. Transmitter-Based Congestion Control Approach

This section describes an approach to the congestion control algorithm based on performance measured by the transmitter without dependence on signaling from the receiver.

6.3.1. Router Architecture Discussion

Note that 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 line cards, separate the IS-IS PDUs 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-specific 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 determine what value should be advertised as a receive window.

The following section describes an approach to congestion control based on performance measured by the transmitter without dependence on signaling from the receiver.

6.3.2. Guidelines for Transmitter-Side Congestion Controls

The approach described in this section does not depend upon direct signaling from the receiver. Instead, it adapts the transmission rate based on measurement of the actual rate of acknowledgments received.

Flow control is not used by this approach. When congestion control is necessary, it can be implemented based on knowledge of the current flooding rate and the current acknowledgment rate. The algorithm used is a local matter. There is no requirement to standardize it, but there are a number of aspects that serve as guidelines that can be described. Algorithms based on this approach should follow the recommendations described below.

A maximum LSP transmission rate (LSPTxMax) should be configurable. This represents the fastest LSP transmission rate that 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 congestion 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 longer delays in convergence.

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

The congestion control algorithm should not assume that the receive performance of a neighbor is static, i.e., it should handle transient conditions that result in a slower or faster receive rate on the part of a neighbor.

The congestion control algorithm should consider the expected delay time in receiving an acknowledgment. Therefore, it incorporates the neighbor partialSNPInterval (Section 4.5) to help determine whether acknowledgments 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 allocation in the "IS-IS Top-Level TLV Codepoints" registry.

Value	Name	IIH	LSP	SNP	Purge
21	Flooding Parameters TLV	y	n	y	n

Table 1

7.2. Registry: IS-IS Sub-TLV for Flooding Parameters TLV

IANA has created the following sub-TLV registry in the "IS-IS TLV Codepoints" registry group.

Name: IS-IS Sub-TLVs for Flooding Parameters TLV
Registration Procedure(s): Expert Review
Description: This registry defines sub-TLVs for the Flooding Parameters TLV (21).
Reference: RFC 9681

Type	Description
0	Reserved
1	LSP Burst Size
2	LSP Transmission Interval
3	LSPs per PSNP
4	Flags
5	PSNP Interval

Type	Description
6	Receive Window
7-255	Unassigned

Table 2: Initial Sub-TLV Allocations for Flooding Parameters TLV

7.3. Registry: IS-IS Bit Values for Flooding Parameters Flags Sub-TLV

IANA has created a new registry, in the "IS-IS TLV Codepoints" registry group, for assigning Flag bits advertised in the Flags sub-TLV.

Name: IS-IS Bit Values for Flooding Parameters Flags Sub-TLV
Registration Procedure: Expert Review
Description: This registry defines bit values for the Flags sub-TLV (4) advertised in the Flooding Parameters TLV (21).

Note: In order to minimize encoding space, a new allocation should pick the smallest available value.

Reference: RFC 9681

Bit #	Description
0	Ordered acknowledgment (O-flag)
1-63	Unassigned

Table 3: Initial Bit Allocations for Flags Sub-TLV

8. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589], [RFC5304], and [RFC5310]. These documents describe mechanisms that provide for 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 a false SNP or IIH without having access to the link layer.

If a false SNP or 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 or 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 cause 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 limit the dissemination of such information.

9. References

9.1. Normative References

[ISO10589] ISO/IEC, "Information technology - Telecommunications and information exchange between systems - 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)", Second Edition, ISO/IEC 10589:2002, November 2002, <https://www.iso.org/standard/30932.html>.

[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>.

9.2. Informative References

[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>.

[RFC9667] Li, T., Ed., Psenak, P., Ed., Chen, H., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", RFC 9667, DOI 10.17487/RFC9667, October 2024, <https://www.rfc-editor.org/info/rfc9667>.

Acknowledgments

The authors would like to thank Henk Smit, Sarah Chen, Xuesong Geng, Pierre Francois, Hannes Gredler, Acee Lindem, Mirja Kühlewind, Zaheduzzaman Sarker, and John Scudder 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 for their identification of some limiting factors.

Contributors

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

Jayesh J
Ciena
Email: jayesh.ietf@gmail.com

Chris Bowers
Juniper Networks
Email: cbowers@juniper.net

Peter Psenak
Cisco Systems
Email: ppsenak@cisco.com

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

Gunter Van de Velde
Nokia
Copernicuslaan 50
2018 Antwerp
Belgium
Email: gunter.van_de_velde@nokia.com

Tony Przygienda
Juniper
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: prz@juniper.net