Network Working Group
Request for Comments: 3719
Category: Informational

J. Parker, Ed.
Axiowave Networks
February 2004

Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document discusses a number of differences between the Intermediate System to Intermediate System (IS-IS) protocol as described in ISO 10589 and the protocol as it is deployed today. These differences are discussed as a service to those implementing, testing, and deploying the IS-IS Protocol. A companion document discusses differences between the protocol described in RFC 1195 and the protocol as it is deployed today for routing IP traffic.

Table of Contents

1. Introduction

> In theory, there is no difference between theory and practice.
> But in practice, there is.
>
> Jan L.A. van de Snepscheut

Interior Gateway Protocols such as IS-IS are designed to provide timely information about the best routes in a routing domain. The original design of IS-IS, as described in ISO 10589 [1] has proved to be quite durable. However, a number of original design choices have been modified. This document addresses differences between the protocol described in ISO 10589 and the protocol that can be observed on the wire today. A companion document discusses differences between the protocol described in RFC 1195 [2] for routing IP traffic and current practice.

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT" and "MAY" in this document are to be interpreted as described in RFC 2119 [3].

2. Constants That Are Variable

Some parameters that were defined as constant in ISO 10589 are modified in practice. These include the following

(1) MaxAge - the lifetime of a Link State PDU (LSP)

(2) ISISHoldingMultiplier - a parameter used to describe the generation of hello packets

(3) ReceiveLSPBufferSize - discussed in a later section

2.1. MaxAge

Each LSP contains a RemainingLifetime field which is initially set to the MaxAge value on the generating IS. The value stored in this field is decremented to mark the passage of time and the number of times it has been forwarded. When the value of a foreign LSP becomes 0, an IS initiates a purging process which will flush the LSP from the network. This ensures that corrupted or otherwise invalid LSPs do not remain in the network indefinitely. The rate at which LSPs are regenerated by the originating IS is determined by the value of maximumLSPGenerationInterval.

MaxAge is defined in ISO 10589 as an Architectural constant of 20 minutes, and it is recommended that maximumLSPGenerationInterval be set to 15 minutes. These times have proven to be too short in some networks, as they result in a steady flow of LSP updates even when nothing is changing. To reduce the rate of generation, some implementations allow these times to be set by the network operator.

The relation between MaxAge and maximumLSPGenerationInterval is discussed in section 7.3.21 of ISO 10589. If MaxAge is smaller than maximumLSPGenerationInterval, then an LSP will expire before it is replaced. Further, as RemainingLifetime is decremented each time it is forwarded, an LSP far from its origin appears older and is removed sooner. To make sure that an LSP survives long enough to be replaced, MaxAge should exceed maximumLSPGenerationInterval by at least ZeroAgeLifetime + minimumLSPTransmissionInterval. The first term, ZeroAgeLifetime, is an estimate of how long it takes to flood an LSP through the network. The second term, minimumLSPTransmissionInterval, takes into account how long a router might delay before sending an LSP. The original recommendation was that MaxAge be at least 5 minutes larger than maximumLSPGenerationInterval, and that recommendation is still valid today.

An implementation MAY use a value of MaxAge that is greater than 1200 seconds. MaxAge SHOULD exceed maximumLSPGenerationInterval by at least 300 seconds. An implementation SHOULD NOT use its value of MaxAge to discard LSPs from peers, as discussed below.

An implementation is not required to coordinate the RemainingLifetime it assigns to LSPs to the RemainingLifetime values it accepts, and MUST ignore the following sentence from section 7.3.16.3. of ISO 10589.

> "If the value of Remaining Lifetime [of the received LSP] is greater than MaxAge, the LSP shall be processed as if there were a checksum error."

2.2. ISISHoldingMultiplier

An IS sends IS to IS Hello Protocol Data Units (IIHs) on a periodic basis over active circuits, allowing other attached routers to monitor their aliveness. The IIH includes a two byte field called the Holding Time which defines the time to live of an adjacency. If an IS does not receive a hello from an adjacent IS within this holding time, the adjacent IS is assumed to be no longer operational, and the adjacency is removed.

ISO 10589 defines ISISHoldingMultiplier to be 10, and states that the value of Holding Time should be ISISHoldingMultiplier multiplied by iSISHelloTimer for ordinary systems, and dRISISHelloTimer for a DIS. This implies that the neighbor must lose 10 IIHs before an adjacency times out.

In practice, a value of 10 for the ISISHoldingMultiplier has proven to be too large. DECnet PhaseV defined two related values. The variable holdingMultiplier, with a default value of 3, was used for point-to-point IIHs, while the variable ISISHoldingMultiplier, with a default value of 10, was used for LAN IIHs. Most implementations today set the default ISISHoldingMultiplier to 3 for both circuit types.

Note that adjacent systems may use different values for Holding Time and will form an adjacency with non-symmetric hold times.

An implementation MAY allow ISISHoldingMultiplier to be configurable. Values lower than 3 are unstable, and may cause adjacencies to flap.

3. Variables That Are Constant

Some values that were defined as variables in ISO 10589 do not vary in practice. These include

(1) ID Length - the length of the SystemID

(2) maximumAreaAddresses

(3) Protocol Version

3.1. ID Length

The ID Length is a field carried in all PDUs. The ID Length defines the length of the System ID, and is allowed to take values from 0 to 8. A value of 0 is interpreted to define a length of 6 bytes. As suggested in B.1.1.3 of [1], it is easy to use an Ethernet MAC address to generate a unique 6 byte System ID. Since the SystemID only has significance within the IGP Domain, 6 bytes has proved to be easy to use and ample in practice. There are also new IS-IS Traffic Engineering TLVs which assume a 6 byte System ID. Choices for the ID length other than 6 are difficult to support today. Implementations may interoperate without being able to deal with System IDs of any length other than 6.

An implementation MUST use an ID Length of 6, and MUST check the ID Length defined in the IS-IS PDUs it receives. If a router encounters a PDU with an ID Length different from 0 or 6, section 7.3.15.a.2

dictates that it MUST discard the PDU, and SHOULD generate an appropriate notification. ISO 10589 defines the notification iDFieldLengthMismatch, while the IS-IS MIB [7] defines the notification isisIDLenMismatch.

3.2. maximumAreaAddresses

The value of maximumAreaAddresses is defined to be an integer between 1 and 254, and defines the number of synonymous Area Addresses that can be in use in an L1 area. This value is advertised in the header of each IS-IS PDU.

Most deployed networks use one Area Address for an L1 area. When merging or splitting areas, a second address is required for seamless transition. The third area address was originally required to support DECnet PhaseIV addresses as well as OSI addresses during a transition.

ISO 10589 requires that all Intermediate Systems in an area or domain use a consistent value for maximumAreaAddresses. Common practice is for an implementation to use the value 3. Therefore an implementation that only supports 3 can expect to interoperate successfully with other conformant systems.

ISO 10589 specifies that an advertised value of 0 is treated as equivalent to 3, and that checking the value for consistency may be omitted if an implementation only supports the value 3.

An implementation SHOULD use the value 3, and it SHOULD check the value advertised in IS-IS PDUs it receives. If a router receives a PDU with maximumAreaAddresses that is not 0 or 3, it MUST discard the PDU, as described in section 7.3.15.a.3, and it SHOULD generate an appropriate notification. ISO 10589 defines the notification maximumAreaAddressMismatch, while the IS-IS MIB [7] defines the notification isisMaxAreaAddressesMismatch.

3.3. Protocol Version

IS-IS PDUs include two one-byte fields in the headers: "Version/Protocol ID Extension" and "Version".

An implementation SHOULD set both fields to 1, and it SHOULD check the values of these fields in IS-IS PDUs it receives. If a router receives a PDU with a value other than 1 for either field, it MUST drop the packet, and SHOULD generate the isisVersionSkew notification.

4. Alternative Metrics

Section 7.2.2, ISO 10589 describes four metrics: Default Metric, Delay Metric, Expense Metric, and Error Metric. None but the Default Metric are used in deployed networks, and most implementations only consider the Default Metric. In ISO 10589, the most significant bit of the 8 bit metrics was the field S (Supported), used to define if the metric was meaningful.

> If this IS does not support this metric it shall set bit S to 1 to indicate that the metric is unsupported.

The Supported bit was always 0 for the Default Metric, which must always be supported. However, RFC 2966 [5] uses this bit in the Default Metric to mark L1 routes that have been leaked from L1 to L2 and back down into L1 again.

Implementations MUST generate the Default Metric when using narrow metrics, and SHOULD ignore the other three metrics when using narrow metrics. Implementations MUST assume that the Default Metric is supported, even if the S bit is set. RFC 2966 describes restrictions on leaking such routes learned from L1 into L2.

5. ReceiveLSPBufferSize

Since IS-IS does not allow segmentation of protocol PDUs, Link State PDUs (LSPs) must be propagated without modification on all IS-IS enabled links throughout the area/domain. Thus it is essential to configure a maximum size that all routers can forward, receive, and store.

This affects three aspects, which we discuss in turn:

(1) The largest LSP we can receive (ReceiveLSPBufferSize)

(2) The size of the largest LSP we can generate (originatingL1LSPBufferSize and originatingL2LSPBufferSize)

(3) Available Link MTU for supported Circuits (MTU). Note this often differs from the MTU available to IP clients.

ISO 10589 defines the architectural constant ReceiveLSPBufferSize with value 1492 bytes, and two private management parameters, originatingL1LSPBufferSize for level 1 PDUs and originatingL2LSPBufferSize for level 2 PDUs. The originating buffer size parameters define the maximum size of an LSP that a router can generate. ISO 10589 directs the implementor to treat a PDU larger than ReceiveLSPBufferSize as an error.

It is crucial that

originatingL1LSPBufferSize <= ReceiveLSPBufferSize

originatingL2LSPBufferSize <= ReceiveLSPBufferSize

and that for all L1 links in the area

originatingL1LSPBufferSize <= MTU

and for all L2 links in the domain

originatingL2LSPBufferSize <= MTU

The original thought was that operators could decrease the originating Buffer size when dealing with smaller MTUs, but would not need to increase ReceiveLSPBufferSize beyond 1492.

With the definition of new information to be advertised in LSPs, such as the Traffic Engineering TLVs, the limited space of the LSP database which may be generated by each router (256 * 1492 bytes at each level) has become an issue. Given that modern networks with MTUs larger than 1492 on all links are not uncommon, one method which can be used to expand the LSP database size is to allow values of ReceiveLSPBufferSize greater than 1492.

Allowing ReceiveLSPBUfferSize to become a configurable parameter rather than an architectural constant must be done with care: if any system in the network does not support values larger than 1492 or one or more link MTUs used by IS-IS anywhere in the area/domain is smaller than the largest LSP which may be generated by any router, then full propagation of all LSPs may not be possible, resulting in routing loops and black holes.

The steps below are recommended when changing ReceiveLSPBufferSize.

(1) Set the ReceiveLSPBufferSize to a consistent value throughout the network.

(2) The implementation MUST not enable IS-IS on circuits which do not support an MTU at least as large as the originating BufferSize at the appropriate level.

(3) Include an originatingLSPBufferSize TLV when generating LSPs, introduced in section 9.8 of ISO 10589:2002 [1].

(4) When receiving LSPs, check for an originatingLSPBufferSize TLV, and report the receipt of values larger than the local value of ReceiveLSPBufferSize through the defined Notifications and Alarms.

(5) Report the receipt of a PDU larger than the local ReceiveLSPBufferSize through the defined Notifications and Alarms.

(6) Do not discard large PDUs by default. Storing and processing them as normal PDUs may help maintain coherence in a misconfigured network.

Steps 1 and 2 are enough by themselves, but the consequences of mismatch are serious enough and difficult enough to detect, that steps 3-6 are recommended to help track down and correct problems.

6. Padding Hello PDUs

To prevent the establishment of adjacencies between systems which may not be able to successfully receive and propagate IS-IS PDUs due to inconsistent settings for originatingLSPBufferSize and ReceiveLSPBufferSize, section 8.2.3 of [1] requires padding on point-to-point links.

On point-to-point links, the initial IIH is to be padded to the maximum of

(1) Link MTU

(2) originatingL1LSPBufferSize if the link is to be used for L1 traffic

(3) originatingL2LSPBufferSize if the link is to be used for L2 traffic

In section 6.7.2 e) ISO 10589 assumes

> Provision that failure to deliver a specific subnetwork SDU will result in the timely disconnection of the subnetwork connection in both directions and that this failure will be reported to both systems

With this service provided by the link layer, the requirement that only the initial IIH be padded was sufficient to check the consistency of the MTU on the two sides. If the PDU was too big to be received, the link would be reset. However, link layer protocols in use on point-to-point circuits today often lack this service, and the initial padded PDU might be silently dropped without resetting the circuit. Therefore, the requirement that only the initial IIH be padded does not provide the guarantees anticipated in ISO 10589.

If an implementation is using padding to detect problems, point-to-point IIH PDUs SHOULD be padded until the sender declares an adjacency on the link to be in state Up. If the implementation implements RFC 3373 [4], "Three-Way Handshake for IS-IS Point-to-Point Adjacencies" then this is when the three-way state is Up: if the implementation use the "classic" algorithm described in ISO 10589, this is when adjacencyState is Up. Transmission of padded IIH PDUs SHOULD be resumed whenever the adjacency is torn down, and SHOULD continue until the sender declares the adjacency to be in state Up again.

If an implementation is using padding, and originatingL1LSPBUfferSize or originatingL2LSPBUfferSize is modified, adjacencies SHOULD be brought down and reestablished so the protection provided by padding IIH PDUs is performed consistent with the modified values.

Some implementations choose not to pad. Padding does not solve all problems of misconfigured systems. In particular, it does not provide a transitive relation. Assume that A, B, and C all pad IIH PDUs, that A and B can establish an adjacency, and that B and C can establish an adjacency. We still cannot conclude that A and C could establish an adjacency, if they were neighbors.

The presence or absence of padding TLVs MUST NOT be one of the acceptance tests applied to a received IIH regardless of the state of the adjacency.

7. Zero Checksum

A checksum of 0 is impossible if the checksum is computed according to the rules of ISO 8473 [8].

ISO 10589, section 7.3.14.2(i), states:

> A Link State PDU received with a zero checksum shall be treated as if the Remaining Lifetime were zero. The age, if not zero, shall be overwritten with zero.

That is, ISO 10589 directs the receiver to purge the LSP. This has proved to be disruptive in practice. An implementation SHOULD treat all LSPs with a zero checksum and a non-zero remaining lifetime as if they had as checksum error. Such packets SHOULD be discarded.

8. Purging Corrupted PDUs

While ISO 10589 requires in section 7.3.14.2 e) that any LSP received with an invalid PDU checksum should be purged, this has been found to be disruptive. Most implementations today follow the revised specification, and simply drop the LSP.

In ISO 10589:2002 [1], Section 7.3.14.2, it states:

> (e) An Intermediate system receiving a Link State PDU with an incorrect LSP Checksum or with an invalid PDU syntax SHOULD
>
> 1) generate a corruptedLSPReceived circuit event,
>
> 2) discard the PDU.

9. Checking System ID in Received point-to-point IIH PDUs

In section 8.2.4.2, ISO 10589 does not explicitly require comparison of the source ID of a received IIH with the neighbourSystemID associated with an existing adjacency on a point-to-point link.

To address this omission, implementations receiving an IIH PDU on a point to point circuit with an established adjacency SHOULD check the Source ID field and compare that with the neighbourSystemID of the adjacency. If these differ, an implementation SHOULD delete the adjacency.

Given that IIH PDUs as specified in ISO 10589 do not include a check-sum, it is possible that a corrupted IIH may falsely indicate a change in the neighbor's System ID. The required subnetwork guarantees for point-to-point links, as described in 6.7.2 g) 1) assume that undetected corrupted PDUs are very rare (one event per four years). A link with frequent errors that produce corrupted data could lead to flapping an adjacency. Inclusion of an optional checksum TLV as specified in "Optional Checksums in IS-IS" [6], may be used to detect such corruption. Hello packets carrying this TLV that are corrupted PDUs SHOULD be silently dropped, rather than dropping the adjacency.

Some implementations have chosen to discard received IIHs where the source ID differs from the neighbourSystemID. This may prevent needless flapping caused by undetected PDU corruption. If an actual administrative change to the neighbor's system ID has occurred, using this strategy may require the existing adjacency to timeout before an adjacency with the new neighbor can be established. This is expedited if the neighbor resets the circuit as anticipated in 10589 after a System ID change, or resets the 3-way adjacency state, as anticipated in RFC 3373.

10. Doppelganger LSPs

When an Intermediate System shuts down, it may leave old LSPs in the network. In the normal course of events, a rebooting system flushes out these old LSPs by reissuing those fragments with a higher sequence number, or by purging fragments that it is not currently generating.

In the case where a received LSP or SNP entry and an LSP in the local database have the same LSP ID, same sequence number, non-zero remaining lifetimes, but different non-zero checksums, the rules defined in [1] cannot determine which of the two is "newer". In this case, an implementation may opt to perform an additional test as a tie breaker by comparing the checksums. Implementations that elect to use this method MUST consider the LSP/SNP entry with the higher checksum as newer. When comparing the checksums the checksum field is treated as a 16 bit unsigned integer in network byte order (i.e., most significant byte first).

The choice of higher checksum, rather than lower, while arbitrary, aligns with existing implementations and ensures compatibility.

Note that a purged LSP (i.e., an LSP with remaining lifetime set to 0) is always considered newer than a non-purged copy of the same LSP.

11. Generating a Complete Set of CSNPs

There are a number of cases in which a complete set of CSNPs must be generated. The DIS on a LAN, two IS’s peering over a P2P link, and an IS helping another IS perform graceful restart must generate a complete set of CSNPs to assure consistent LSP Databases throughout. Section 7.3.15.3 of [1] defines a complete set of CSNPs to be:

> "A complete set of CSNPs is a set whose Start LSPID and End LSPID ranges cover the complete possible range of LSPIDs. (i.e., there is no possible LSPID value which does not appear within the range of one of the CSNPs in the set). "

Strict adherence to this definition is required to ensure the reliability of the update process. Deviation can lead to subtle and hard to detect defects. It is not sufficient to send a set of CSNPs which merely cover the range of LSPIDs which are in the local database. The set of CSNPs must cover the complete possible range of LSPIDs.

Consider the following example:

If the current Level 1 LSP database on a router consists of the following non pseudo-node LSPs:

From system 1111.1111.1111 LSPs numbered 0-89(59H)
From system 2222.2222.2222 LSPs numbered 0-89(59H)

If the maximum size of a CSNP is 1492 bytes, then 90 CSNP entries can fit into a single CSNP PDU. The following set of CSNP start/end LSPIDs constitute a correctly formatted complete set:

Start LSPID	End LSPID
0000.0000.0000.00-00	1111.1111.1111.00-59
1111.1111.1111.00-5A	FFFF.FFFF.FFFF.FF-FF

The following are examples of incomplete sets of CSNPS:

Start LSPID	End LSPID
0000.0000.0000.00-00	1111.1111.1111.00-59
1111.1111.1111.00-5A	2222.2222.2222.00-59

The sequence above has a gap after the second entry.

Start LSPID	End LSPID
0000.0000.0000.00-00	1111.1111.1111.00-59
2222.2222.2222.00-00	FFFF.FFFF.FFFF.FF-FF

The sequence above has a gap between the first and second entry.

Although it is legal to send a CSNP which contains no actual LSP entry TLVs, it should never be necessary to do so in order to conform to the specification.

12. Overload Bit

To deal with transient problems that prevent an IS from storing all the LSPs it receives, ISO 10589 defines an LSP Database Overload condition in section 7.3.19. When an IS is in Database Overload condition, it sets a flag called the Overload Bit in the non-pseudonode LSP number Zero that it generates. Section 7.2.8.1 of ISO 10589 instructs other systems not to use the overloaded IS as a transit router. Since the overloaded IS does not have complete information, it may not be able to compute the right routes, and routing loops could develop.

An overloaded router might become the DIS. An implementation SHOULD not set the Overload bit in PseudoNode LSPs that it generates, and Overload bits seen in PseudoNode LSPs SHOULD be ignored.

13. Security Considerations

The clarifications in this document do not raise any new security concerns, as there is no change in the underlying protocol described in ISO 10589 [1].

14. References

14.1. Normative References

[1] ISO, "Intermediate system to Intermediate system routeing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)," ISO/IEC 10589:2002.

[2] Callon, R., "OSI IS-IS for IP and Dual Environment", RFC 1195, December 1990.

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

[4] Katz, D. and Saluja, R., " Three-Way Handshake for Intermediate System to Intermediate System (IS-IS) Point-to-Point Adjacencies", RFC 3373, September 2002.

[5] Li, T., Przygienda, T. and H. Smit, "Domain-wide Prefix Distribution with Two-Level IS-IS", RFC 2966, October 2000.

[6] Koodli, R. and R. Ravikanth, "Optional Checksums in Intermediate System to Intermediate System (ISIS)", RFC 3358, August 2002.

14.2. Informative References

[7] Parker, J., "Management Information Base for IS-IS", Work in Progress, January 2004.

[8] ITU, "Information technology - Protocol for providing the connectionless-mode network service", ISO/IEC 8473-1, 1998.

15. Acknowledgments

This document is the work of many people, and is the distillation of over a thousand mail messages. Thanks to Vishwas Manral, who pushed to create such a document. Thanks to Danny McPherson, the original editor, for kicking things off. Thanks to Mike Shand, for his work in creating the protocol, and his uncanny ability to remember what everything is for. Thanks to Micah Bartell and Philip Christian, who showed us how to document difference without displaying discord. Thanks to Les Ginsberg, Neal Castagnoli, Jeff Learman, and Dave Katz, who spent many hours educating the editor. Thanks to Radia Perlman, who is always ready to explain anything. Thanks to Satish Dattatri, who was tenacious in seeing things written up correctly. Thanks to Russ White, whose writing improved the treatment of every topic he touched. Thanks to Shankar Vemulapalli, who read several drafts with close attention. Thanks to Don Goodspeed, for his close reading of the text. Thanks to Aravind Ravikumar, who pointed out that we should check Source ID on point-to-point IIH packets. Thanks to Michael Coyle for identifying the quotation from Jan L.A. van de Snepscheut. Thanks for Alex Zinin's ministrations behind the scenes. Thanks to Tony Li and Tony Przygienda, who kept us on track as the discussions veered into the weeds. And thanks to all those who have contributed, but whose names I have carelessly left from this list.

16. Author's Address

Jeff Parker
Axiowave Networks
200 Nickerson Road
Marlborough, Mass 01752
USA

EMail: jparker@axiowave.com

17. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.