Internet Engineering Task Force (IETF)
Request for Comments: 7212
Category: Standards Track
ISSN: 2070-1721

D. Frost
Blue Sun
S. Bryant
Cisco Systems
M. Bocci
Alcatel-Lucent

June 2014

MPLS Generic Associated Channel (G-ACh) Advertisement Protocol

Abstract

The MPLS Generic Associated Channel (G-ACh) provides an auxiliary logical data channel associated with a Label Switched Path (LSP), a pseudowire, or a section (link) over which a variety of protocols may flow. These protocols are commonly used to provide Operations, Administration, and Maintenance (OAM) mechanisms associated with the primary data channel. This document specifies simple procedures by which an endpoint of an LSP, pseudowire, or section may inform the other endpoints of its capabilities and configuration parameters, or other application-specific information. This information may then be used by the receiver to validate or adjust its local configuration, and by the network operator for diagnostic purposes.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7212.

Table of Contents

1. Introduction

The MPLS Generic Associated Channel (G-ACh) is defined and described in [RFC5586]. It provides an auxiliary logical data channel over which a variety of protocols may flow. Each such data channel is associated with an MPLS Label Switched Path (LSP), a pseudowire, or a section (link). An important use of the G-ACh and the protocols it supports is to provide Operations, Administration, and Maintenance (OAM) [RFC6291] capabilities for the associated LSP, pseudowire, or section. Examples of such capabilities include Pseudowire Virtual Circuit Connectivity Verification (VCCV) [RFC5085]; Bidirectional Forwarding Detection (BFD) for MPLS [RFC5884]; and MPLS packet loss, delay, and throughput measurement [RFC6374]; as well as OAM functions developed for the MPLS Transport Profile (MPLS-TP) [RFC5921].

This document specifies procedures for an MPLS Label Switching Router (LSR) to advertise its capabilities and configuration parameters, or other application-specific information, to its peers over LSPs, pseudowires, and sections. Receivers can then make use of this information to validate or adjust their own configurations, and network operators can make use of it to diagnose faults and configuration inconsistencies between endpoints. Note that in this document the term "application" refers to an application that uses the protocol defined herein (and hence operates over the G-ACh), and it should not be confused with an end-user application.

The main principle guiding the design of the MPLS G-ACh Advertisement Protocol (GAP) is simplicity. The protocol provides a one-way method of distributing information about the sender. How this information is used by a given receiver is a local matter. The data elements distributed by the GAP are application specific and, except for those associated with the GAP itself, are outside the scope of this document. An IANA registry has been created to allow GAP applications to be defined as needed.

The assignment of application identifiers and associated GAP parameters for protocols other than the GAP itself is outside the scope of this document. Such assignments can be made in subsequent documents according to the IANA considerations specified here.

1.1. Motivation

It is frequently useful in a network for a node to have general information about its adjacent nodes, i.e., those nodes to which it has links. At a minimum, this allows a human operator or management application with access to the node to determine which adjacent nodes this node can see; this is helpful when troubleshooting connectivity problems. A typical example of an "adjacency awareness protocol" is

the Link Layer Discovery Protocol [LLDP], which can provide various pieces of information about adjacent nodes in Ethernet networks, such as system name, basic functional capabilities, link speed/duplex settings, and maximum supported frame size. Such data is useful both for human diagnostics and for automated detection of configuration inconsistencies.

In MPLS networks, the G-ACh provides a convenient link-layer-agnostic means for communication between LSRs that are adjacent at the link layer. The G-ACh advertisement protocol presented in this document thus allows LSRs to exchange information of a similar sort to that supported by LLDP for Ethernet links. The GAP, however, does not depend on the specific link-layer protocol in use, and it can be used to advertise information on behalf of any MPLS application.

In networks based on the MPLS Transport Profile (MPLS-TP) [RFC5921] that do not also support IP, the normal protocols used to determine the Ethernet address of an adjacent MPLS node, such as the Address Resolution Protocol [RFC0826] and IP version 6 Neighbor Discovery [RFC4861], are not available. One possible use of the G-ACh advertisement protocol is to discover the Ethernet media access control addresses of MPLS-TP nodes lacking IP capability [RFC7213]. However, where it is anticipated that the only data that needs to be exchanged between LSRs over an Ethernet link are their Ethernet addresses, then the operator may instead choose to use LLDP for that purpose.

The applicability of the G-ACh advertisement protocol is not limited to link-layer adjacency, either in terms of message distribution or message content. The G-ACh exists for any MPLS LSP or pseudowire, so GAP messages can be exchanged with remote LSP or pseudowire endpoints. The content of GAP messages is extensible in a simple manner and can include any kind of information that might be useful to MPLS LSRs connected by links, LSPs, or pseudowires. For example, in networks that rely on the G-ACh for OAM functions, GAP messages might be used to inform adjacent LSRs of a node’s OAM capabilities and configuration parameters.

1.2. Terminology

Term	Definition
G-ACh	Generic Associated Channel
GAL	G-ACh Label
GAP	G-ACh Advertisement Protocol
LSP	Label Switched Path
OAM	Operations, Administration, and Maintenance

1.3. 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 [RFC2119].

2. Overview

The G-ACh Advertisement Protocol has a simple one-way mode of operation: a device configured to send information for a particular data channel (MPLS LSP, pseudowire, or section) transmits GAP messages over the G-ACh associated with the data channel. The payload of a GAP message is a collection of Type-Length-Value (TLV) objects, organized on a per-application basis. An IANA registry has been created to identify specific applications. Application TLV objects primarily contain static data that the receiver is meant to retain for a period of time, but they may also represent metadata or special processing instructions.

Each GAP message can contain data for several applications. A sender may transmit a targeted update that refreshes the data for a subset of applications without affecting the data of other applications sent in a previous message. GAP messages are processed in the order in which they are received.

For example, a GAP message might be sent containing the following data:

Application A: A-TLV4, A-TLV15, A-TLV9

Application B: B-TLV1, B-TLV3

Application C: C-TLV6,

where the TLVx refers to an example GAP TLV.

A second message might then be sent containing:

Application B: B-TLV7, B-TLV3

Upon receiving the second message, the receiver retains B-TLV1 from the first message and adds B-TLV7 to its B-database. How it handles the new B-TLV3 depends on the rules B has specified for this object type; this object could replace the old one or be combined with it in some way. The second message has no effect on the databases maintained by the receiver for Applications A and C.

The rate at which GAP messages are transmitted is at the discretion of the sender and may fluctuate over time as well as differ per application. Each message contains, for each application it describes, a lifetime that informs the receiver how long to wait before discarding the data for that application.

The GAP itself provides no fragmentation and reassembly mechanisms. In the event that an application wishes to send larger chunks of data via GAP messages than fall within the limits of packet size, it is the responsibility of the application to fragment its data accordingly. It is the responsibility of the application and the network operator to ensure that the use of the GAP does not congest the link to the peer.

The GAP is designed to run over a unidirectional channel. However, where the channel is bidirectional, communication may be optimized through the use of a number of messages defined for transmission from the receiver back to the sender. These are optimizations and are not required for protocol operation.

3. Message Format

An Associated Channel Header (ACH) Channel Type has been allocated for the GAP as follows:

Protocol	Channel Type
G-ACh Advertisement Protocol	0x0059

For this Channel Type, as noted in [RFC7026], the ACH SHALL NOT be followed by the ACH TLV Header defined in [RFC5586].

Fields in this document shown as Reserved or Resv are reserved for future specification and MUST be set to zero. All integer values for fields defined in this document SHALL be encoded in network byte order.

A GAP message consists of a fixed header followed by a GAP payload. The payload of a GAP message is an Application Data Block (ADB) consisting of one or more block elements. Each block element contains an application identifier, a lifetime, and a series of zero or more TLV objects for the application it describes.

Malformed GAP messages MUST be discarded by the receiver, although an error MAY be logged. If the error is logged remotely, a suitable form of rate limiting SHOULD be used to prevent excessive logging messages being transmitted over the network.

Implementations of this protocol version MUST set reserved fields in the message formats that follow to all zero bits when sending and ignore any value when receiving messages.

3.1. GAP Message Format

The following figure shows the format of a G-ACh Advertisement Protocol message, which follows the Associated Channel Header (ACH):

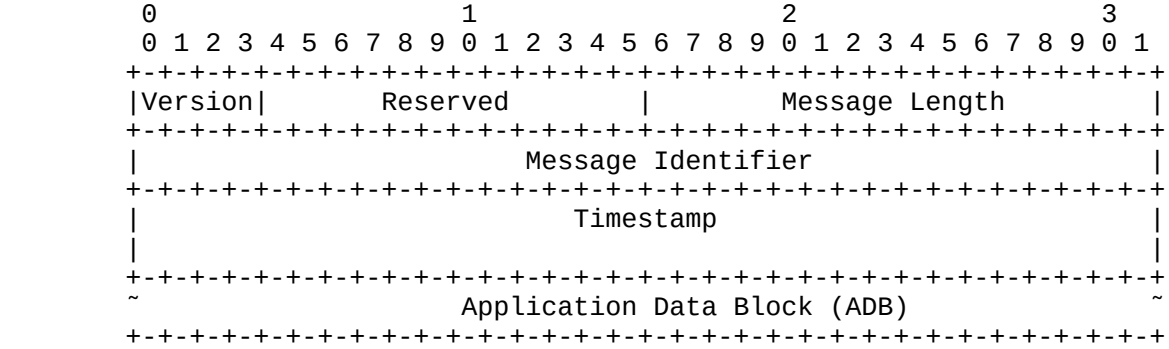

Figure 1: GAP Message Format

The meanings of the fields are:

Version (4 bits): Protocol version. This is set to zero.

Reserved (12 bits): MUST be sent as zero.

Message Length (16 bits): Size in octets of this message, i.e., of the portion of the packet following the Associated Channel Header.

Message Identifier (MI) (32 bits): Unique identifier of this message. For disambiguation, a sender MUST NOT reuse an MI over a given channel until it is confident that all ADBs associated with it have been expired by the receiver. The sole purpose of this field is duplicate detection in the event of a message burst (Section 5.1).

Timestamp: 64-bit Network Time Protocol (NTP) transmit timestamp, as specified in Section 6 of [RFC5905].

3.2. Applications Data Block

An ADB consists of one or more elements of the following format:

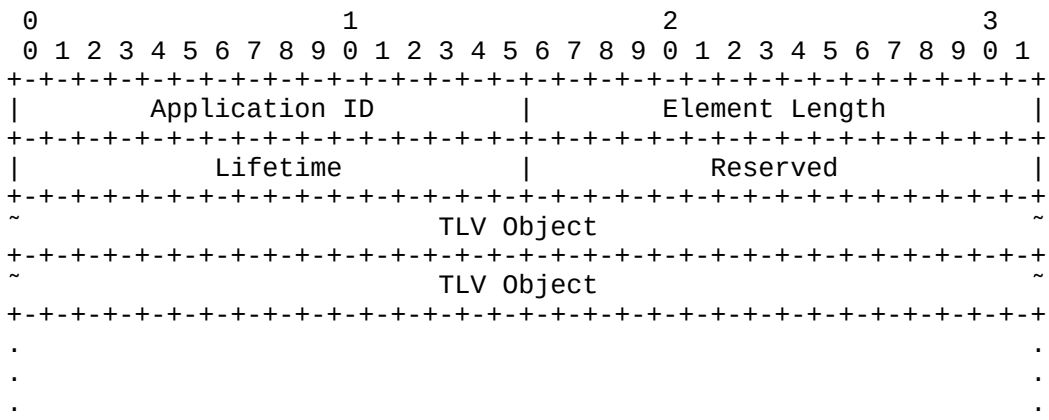

Figure 2: Application Data Block Element

Application ID (16 bits): Identifies the application this element describes; an IANA registry has been created to track the values for this field. More than one block element with the same Application ID may be present in the same ADB, and block elements with different Application IDs may also be present in the same ADB. The protocol rules for the mechanism, including what ADB elements are present and which TLVs are contained in an ADB element, are to be defined in the document that specifies the application-specific usage.

Element Length (16 bits): Specifies the total length in octets of this block element (including the Application ID and Element Length fields).

Lifetime field (16 bits): Specifies how long, in seconds, the receiver should retain the data in this message (i.e., it specifies the lifetime of the static data carried in the TLV set of this ADB). For TLVs not carrying static data, the Lifetime is of no significance. The sender of a GAP message indicates this by setting the Lifetime field to zero. If the Lifetime is zero, TLVs in this ADB are processed by the receiver, and the data associated with these TLV types is immediately marked as expired. If the ADB contains no TLVs, the receiver expires all data associated with TLVs previously sent to this application.

The remainder of the Application Data Block element consists of a sequence of zero or more TLV objects that use the format defined in Section 3.3.

The scope of an ADB element is an application instance attached to a specific channel between a specific source-destination pair, and the

Lifetime field specifies the lifetime of the ADB element data in that specific context.

3.3. TLV Object Format

GAP TLV objects use the following format:

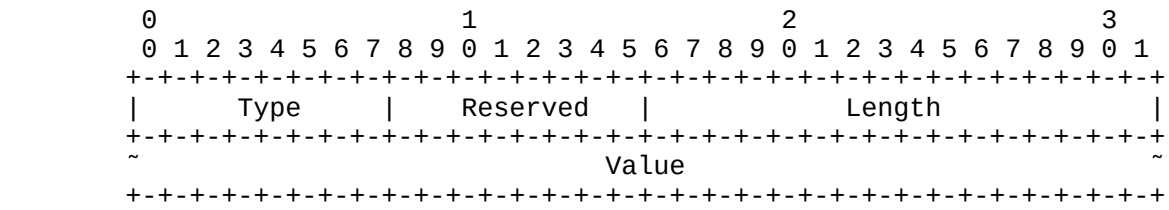

Figure 3: TLV Object Format

Type (8 bits): Identifies the TLV Object and is scoped to a specific application; each application creates an IANA registry to track its Type values.

Reserved (8 bits): MUST be sent as zero.

Length (16 bits): The length in octets of the Value field. The Value field need not be padded to provide alignment.

GAP messages do not contain a checksum. If validation of message integrity is desired, the authentication procedures in Section 6 should be used.

4. G-ACh Advertisement Protocol TLVs

The GAP supports several TLV objects related to its own operation via the Application ID 0x0000. These objects represent metadata and processing instructions rather than static data that is meant to be retained. When an ADB element for the GAP is present in a GAP message, it MUST precede other elements. This is particularly important for the correct operation of the Flush message (Section 4.3).

Any application using the GAP inherits the ability to use facilities provided by Application 0x0000.

Application 0x0000 GAP messages MUST be processed in the order in which they are received.

4.1. Source Address TLV

The Source Address object identifies the sending device and possibly the transmitting interface and the channel; it has the following format:

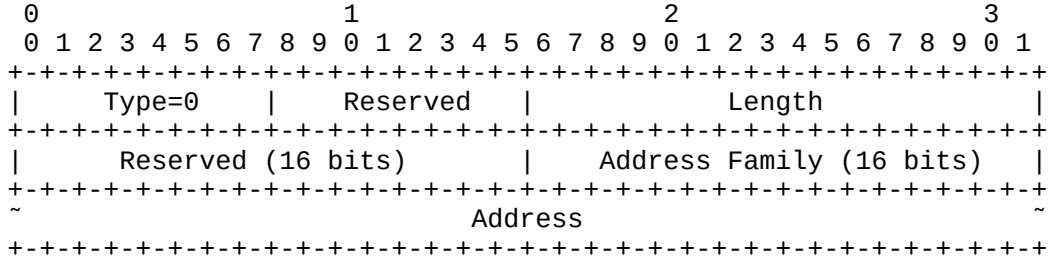

Figure 4: Source Address TLV Format

The Address Family field indicates the type of the address; it SHALL be set to one of the assigned values in the IANA "Address Family Numbers" registry.

In IP networks, a Source Address SHOULD be included in GAP messages and set to an IP address of the sending device; when the channel is a link, this address SHOULD be an address of the transmitting interface.

In non-IP MPLS-TP networks, a Source Address SHOULD be included in GAP messages and set to the endpoint identifier of the channel. The formats of these channel identifiers SHALL be as given in Sections 3.5.1, 3.5.2, and 3.5.3 of [RFC6428] (excluding the initial Type and Length fields shown in those sections). IANA has allocated Address Family Numbers for these identifiers; see Section 10.2.

On multipoint channels, a Source Address TLV is REQUIRED.

4.2. GAP Request TLV

This object is a request by the sender for the receiver to transmit an immediate unicast GAP update to the sender. If the Length field is zero, this signifies that an update for all applications is

requested. Otherwise, the Value field specifies the applications for which an update is requested, in the form of a sequence of Application IDs:

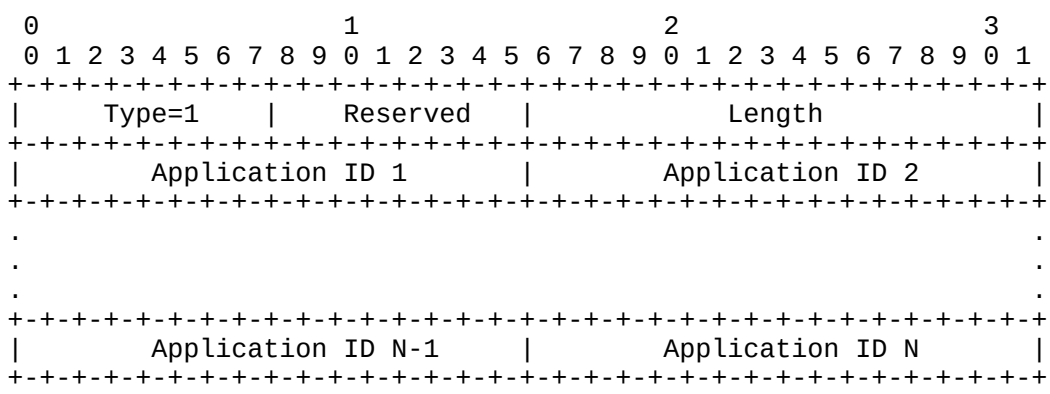

Figure 5: GAP Request TLV Format

The intent of this TLV is to request the immediate transmission of data following a local event such as a restart, rather than waiting for a periodic update. Applications need to determine what information is meaningful to send in response to such a request. The inclusion of an Application ID in a Request TLV does not guarantee that the response will provide information for that application. The responder may also include information for applications not included in the request. A receiver SHOULD discard GAP Request messages that arrive at a rate in excess of that which is considered reasonable for the application.

For an Application ID 0x0000 GAP Request, it is meaningful to respond with the Source Address.

This TLV is considered to be part of the GAP and thus does not need to be retained. The reception of the TLV may however be recorded for management purposes.

4.3. GAP Flush TLV

This object is an instruction to the receiver to flush the GAP data for all applications associated with this (sender, channel) pair. It is a null object, i.e., its Length is set to zero.

The GAP Flush instruction does not apply to data contained in the message carrying the GAP Flush TLV object itself. Any application data contained in the same message SHALL be processed and retained by the receiver as usual.

The Flush TLV type is 2.

This TLV is considered to be part of the GAP and thus does not need to be retained. The reception of the TLV may however be recorded for management purposes.

4.4. GAP Suppress TLV

This object is a request to the receiver to cease sending GAP updates to the transmitter over the current channel for the specified duration. Duration is a 16-bit non-negative integer in units of seconds. The receiver MAY accept and act on the request, MAY ignore the request, or MAY resume transmissions at any time according to implementation or configuration choices, and depending on local pragmatics. The format of this object is as follows:

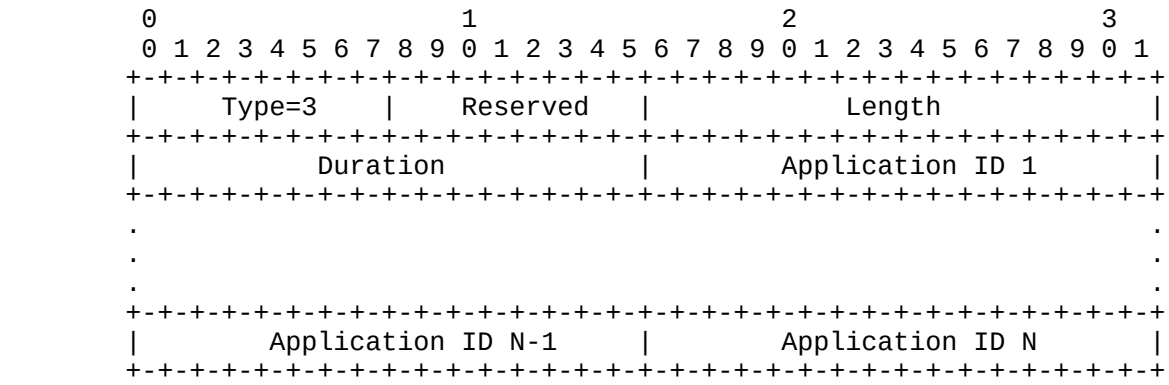

Figure 6: GAP Suppress TLV Format

If the Length is set to 2, i.e., if the list of Application IDs is empty, then suppression of all GAP messages is requested; otherwise, suppression of only those updates pertaining to the listed applications is requested. A duration of zero cancels any existing suppress requests for the listed applications.

This object makes sense only for point-to-point channels or when the sender is receiving unicast GAP updates.

4.5. GAP Authentication TLV

This object is used to provide authentication and integrity validation for a GAP message. It has the following format:

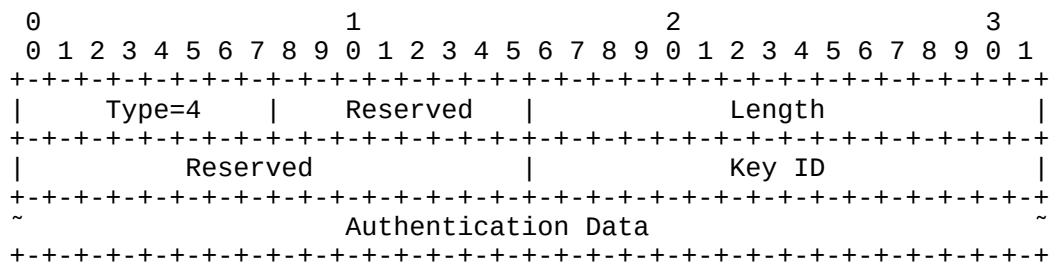

Figure 7: GAP Authentication TLV Format

The data and procedures associated with this object are explained in Section 6.

5. Operation

5.1. Message Transmission

G-ACh Advertisement Protocol message transmission SHALL operate on a per-data-channel basis and be configurable by the operator accordingly.

Because GAP message transmission may be active for many logical channels on the same physical interface, message transmission timers SHOULD be randomized across the channels supported by a given interface so as to reduce the likelihood of large synchronized message bursts.

The Message Identifier (MI) uniquely identifies this message and its value is set at the sender’s discretion. It MUST NOT be assumed to be a sequence number. The scope of an MI is a channel between a specific source-destination pair.

The Timestamp field SHALL be set to the time at which this message is transmitted.

The Lifetime field of each Application Data Block element SHALL be set to the number of seconds the receiver is advised to retain the data associated with this message and application.

When the transmitter wishes the data previously sent in an ADB element to persist, then it must refresh the ADB element by sending another update. Refresh times SHOULD be set in such a way that at least three updates will be sent prior to Lifetime expiration. For example, if the Lifetime is set to 210 seconds, then updates should be sent at least once every 60 seconds.

A sender may signal that previously sent data SHOULD be marked as expired by setting the ADB element lifetime to zero as previously described in Section 3.

In some cases, an application may desire additional reliability for the delivery of some of its data. When this is the case, the transmitter MAY send several (for example, three) instances of the message in succession, separated by a delay appropriate to, or specified by, the application. For example, this procedure might be invoked when sending a Flush instruction following device reset. The expectation is that the receiver will detect duplicate messages using the MI.

5.2. Message Reception

G-ACh Advertisement Protocol message reception SHALL operate on a per-data-channel basis and be configurable by the operator accordingly.

Upon receiving a G-ACh Advertisement Protocol message that contains data for some application X, the receiver determines whether it can interpret X-data. If it cannot, then the receiver MAY retain this data for the number of seconds specified by the Lifetime field; although it cannot parse this data, it may still be of use to the operator.

If the receiver can interpret X-data, then it processes the data objects accordingly, retaining the data associated with those that represent static data for the number of seconds specified by the Lifetime field. If the Lifetime is zero, such data is immediately marked as expired, and, if no TLVs are specified, all data associated with previously received TLVs is marked as expired (Section 3). If one of the received TLV objects has the same Type as a previously received TLV, then the data from the new object SHALL replace the data associated with that Type unless the X specification dictates a different behavior.

The received data is made available to local applications that require it and are locally authorized to view it. The method for doing this is local to the receiver and outside the scope of this document.

The receiver MAY make use of the application data contained in a GAP message to perform some level of auto-configuration, for example, if the application is an OAM protocol. The application SHOULD, however, take care to prevent cases of oscillation resulting from each endpoint attempting to adjust its configuration to match the other. Any such auto-configuration based on GAP information MUST be disabled by default.

The MI may be used to detect and discard duplicate messages.

6. Message Authentication

The GAP provides a means of authenticating messages and ensuring their integrity. This is accomplished by attaching a GAP Authentication TLV and including, in the Authentication Data field, the output of a cryptographic hash function (known as a Message Authentication Code (MAC)), the input to which is the message together with a secret key known only to the sender and receiver. Upon receipt of the message, the receiver computes the same MAC and compares the result with the MAC in the message; if the MACs are not equal, the message is discarded. Use of GAP message authentication is RECOMMENDED.

The remainder of this section gives the details of this procedure, which is based on the procedures for generic cryptographic authentication for the Intermediate System to Intermediate System (IS-IS) routing protocol as described in [RFC5310].

6.1. Authentication Key Identifiers

An Authentication Key Identifier (Key ID) is a 16-bit tag shared by the sender and receiver that identifies a set of authentication parameters. These parameters are not sent over the wire; they are assumed to be associated, on each node, with the Key ID by external means, such as via explicit operator configuration or a separate key-exchange protocol. Multiple Key IDs may be active on the sending and receiving nodes simultaneously, in which case the sender locally selects a Key ID from this set to use in an outbound message. This capability facilitates key migration in the network.

The parameters associated with a Key ID are:

- Authentication Algorithm: This signifies the authentication algorithm to use to generate or interpret authentication data. At present, the following values MAY be supported: HMAC-SHA-1, HMAC-SHA-256. HMAC-SHA-1 MUST be supported.

- Authentication Keystring: A secret octet string that forms the basis for the cryptographic key used by the Authentication Algorithm. It SHOULD NOT be a human-memorable string. Implementations MUST be able to use random binary values of the appropriate length as a keystring.

Implementers SHOULD consider the use of [RFC7210] for key management. If used, authenticated information sent over the GAP MUST only considered valid if it was sent during the Keying and Authentication for Routing Protocols (KARP) interval between SendLifetimeStart and SendLifeTimeEnd. However, if the GAP TLV used to send it expires before the KARP SendLifetimeStart, then information is never used; if it expires before KARP SendNotAfter, the key becomes invalid on expiry of the GAP TLV.

At the time of this writing, mechanisms for dynamic key management in the absence of IP are not available. Key management in such environments therefore needs to take place via the equipment management system or some other out-of-band service. The MPLS layer in a network is normally isolated from direct access by users and thus is a relatively protected environment. Therefore, key turnover is expected to be a relatively infrequent event.

6.2. Authentication Process

The authentication process for GAP messages is straightforward. First, a Key ID is associated on both the sending and receiving nodes with a set of authentication parameters. Following this, when the sender generates a GAP message, it sets the Key ID field of the GAP Authentication TLV accordingly. (The length of the Authentication Data field is also known at this point because it is a function of the Authentication Algorithm.) The sender then computes a MAC for the message as described in Section 6.3 and fills the Authentication Data field of the GAP Authentication TLV with the MAC, overwriting the zeros used in computation. The message is then sent.

When the message is received, the receiver computes a MAC for it as described below, again setting the Authentication Data field of the GAP Authentication TLV to all zeros before computing the MAC. The receiver compares its computed MAC to the MAC received in the Authentication Data field. If the two MACs are equal, authentication of the message is considered to have succeeded; otherwise, it is considered to have failed.

This process suffices to ensure the authenticity and integrity of messages but is still vulnerable to a replay attack, in which a third party captures a message and sends it on to the receiver at some later time. The GAP message header contains a Timestamp field, which

can be used to protect against replay attacks. To achieve this protection, the receiver checks that the time recorded in the Timestamp field of a received and authenticated GAP message corresponds to the current time, within a reasonable tolerance that allows for message propagation delay, and it accepts or rejects the message accordingly. Clock corrections SHOULD be monotonic to avoid replay attacks, unless operator intervention overrides the monotonic configuration setting to achieve a faster convergence with current time.

If the clocks of the sender and receiver are not synchronized with one another, then the receiver must perform the replay check against its best estimate of the current time according to the sender's clock. The timestamps that appear in GAP messages can be used to infer the approximate clock offsets of senders, and, while this does not yield high-precision clock synchronization, it suffices for purposes of the replay check with an appropriately chosen tolerance.

6.3. MAC Computation

The HMAC procedure described in [RFC2104] is used to compute the MAC.

The Authentication Data field of the GAP Authentication TLV is set to all zeros. The MAC is then computed over the entire GAP message as shown in Figure 1.

Where there is less data than is needed for the MAC computation, a value of zero MUST be used.

The length of the Authentication Data field is always less than or equal to the message digest size of the specific hash function that is being used. However, the implementer needs to consider that although MAC truncation decreases the size of the message, it results in a corresponding reduction in the strength of the assurance provided.

MAC truncation is NOT RECOMMENDED.

7. Link-Layer Considerations

When the GAP is used to support device discovery on a data link, GAP messages must be sent in such a way that they can be received by other listeners on the link without the sender first knowing the link-layer addresses of the listeners. In short, they must be multicast. Considerations for multicast MPLS encapsulation are discussed in [RFC5332]. For example, Section 8 of [RFC5332] describes how destination Ethernet MAC addresses are selected for multicast MPLS packets. Since a GAP packet transmitted over a data

link contains just one label, the G-ACh Label (GAL) with label value 13, the correct destination Ethernet address for frames carrying GAP packets intended for device discovery, according to these selection procedures, is 01-00-5e-80-00-0d.

8. Manageability Considerations

The data sent and received by this protocol MUST be made accessible for inspection by network operators, and where local configuration is updated by the received information, it MUST be clear why the configured value has been changed. This allows the operator to determine the operational parameters currently in use and to understand when local configuration has been superseded by inbound parameters received from its peer.

In the event of a system restart, any GAP application data and peer state data that has been retained as a consequence of prior advertisements from GAP peers MUST be discarded; this prevents incorrect operation on the basis of stale data.

All GAP applications MUST be disabled by default and need to be enabled by the operator if required.

9. Security Considerations

G-ACh Advertisement Protocol messages contain information about the sending device and its configuration, which is sent in cleartext over the wire. If an unauthorized third party gains access to the MPLS data plane or the lower network layers between the sender and receiver, it can observe this information. In general, however, the information contained in GAP messages is no more sensitive than that contained in other protocol messages, such as routing updates, which are commonly sent in cleartext. No attempt is therefore made to guarantee confidentiality of GAP messages. Therefore, the GAP MUST NOT be used to send TLVs in cleartext where the value concerned requires confidentiality, for example, GAP or application TLVs containing 'bare' cryptographic keying material. Applications that require confidentiality will need to implement a suitable confidentiality method.

A more significant potential threat is the transmission of GAP messages by unauthorized sources, or the unauthorized manipulation of messages in transit; this can disrupt the information receivers hold about legitimate senders. To protect against this threat, message authentication procedures (specified in Section 6) enable receivers to ensure the authenticity and integrity of GAP messages. These

procedures include the means to protect against replay attacks in which a third party captures a legitimate message and "replays" it to a receiver at some later time.

10. IANA Considerations

10.1. Associated Channel Type Allocation

IANA has allocated an entry in the "MPLS Generalized Associated Channel (G-ACh) Types (including Pseudowire Associated Channel Types)" registry for the "G-ACh Advertisement Protocol", as follows:

Value	Description	Reference
0x0059	G-ACh Advertisement Protocol	This RFC

The reader should note that the "TLV Follows" column in the registry has been deleted [RFC7026].

10.2. Allocation of Address Family Numbers

IANA has allocated three entries from the Standards Track range in the "Address Family Numbers" registry for MPLS-TP Section, LSP, and Pseudowire endpoint identifiers, per Section 4.1. The allocations are:

Number	Description	Reference
26	MPLS-TP Section Endpoint Identifier	This RFC
27	MPLS-TP LSP Endpoint Identifier	This RFC
28	MPLS-TP Pseudowire Endpoint Identifier	This RFC

10.3. Creation of G-ACh Advertisement Protocol Application Registry

IANA has created a new registry, "G-ACh Advertisement Protocol Application Registry" in the "Generic Associated Channel (G-ACh) Parameters" registry, with fields and initial allocations as follows:

Application ID	Description	Reference
0x0000	G-ACh Advertisement Protocol	This RFC

The range of the Application ID field is 0x0000 - 0xFFFF.

The allocation policy for this registry is IETF Review.

10.4. Creation of G-ACh Advertisement Protocol TLV Registry

IANA has created a new registry, "G-ACh Advertisement Protocol: GAP TLV Objects (Application ID 0)" in the "Generic Associated Channel (G-ACh) Parameters" registry, with fields and initial allocations as follows:

Type Name	Type ID	Reference
Source Address	0	This RFC
GAP Request	1	This RFC
GAP Flush	2	This RFC
GAP Suppress	3	This RFC
GAP Authentication	4	This RFC

The range of the Type ID field is 0 - 255.

The allocation policy for this registry is IETF Review.

11. Acknowledgements

We thank Adrian Farrel for his valuable review comments on this document.

12. References

12.1. Normative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

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

[RFC5332] Eckert, T., Rosen, E., Aggarwal, R., and Y. Rekhter, "MPLS Multicast Encapsulations", RFC 5332, August 2008.

[RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", RFC 5586, June 2009.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6428] Allan, D., Swallow Ed. , G., and J. Drake Ed. , "Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile", RFC 6428, November 2011.

[RFC7210] Housley, R., Polk, T., Hartman, S., and D. Zhang, "Database of Long-Lived Symmetric Cryptographic Keys", RFC 7210, April 2014.

12.2. Informative References

[LLDP] IEEE, "Station and Media Access Control Connectivity Discovery", IEEE 802.1AB, September 2009.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5085] Nadeau, T. and C. Pignataro, "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.

[RFC5921] Bocci, M., Bryant, S., Frost, D., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, July 2010.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, June 2011.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.

[RFC7026] Farrel, A. and S. Bryant, "Retiring TLVs from the Associated Channel Header of the MPLS Generic Associated Channel", RFC 7026, September 2013.

[RFC7213] Frost, D., Bryant, S., and M. Bocci, "MPLS-TP Next-Hop Ethernet Addressing", RFC 7213, June 2014.

Authors' Addresses

Dan Frost
Blue Sun

EMail: frost@mm.st

Stewart Bryant
Cisco Systems

EMail: stbryant@cisco.com

Matthew Bocci
Alcatel-Lucent

EMail: matthew.bocci@alcatel-lucent.com