

Internet Engineering Task Force  
Internet Draft  
Intended status: Informational  
Expires: July 2016

P. Unbehagen  
D. Romascanu, Ed.  
J. Seligson  
C. Keene  
Avaya  
N. Bragg  
Ciena  
L. Beliveau  
Windriver  
January 2016

**Auto-attach using LLDP with IEEE 802.1aq SPBM networks  
draft-unbehagen-lldp-spb-02.txt**

Abstract

This informational document describes a method that allows for the automatic attachment of end stations and network devices to a core network based on the individual services that are run or configured on the stations or devices, and the mapping of the services to the managed paths in the network.

Specifically the document describes a compact method based on the IEEE802.1AB Link Layer Discovery Protocol (LLDP) to automatically attach network devices not supporting IEEE 802.1ah to individual services in an IEEE 802.1aq Shortest Path Bridging (SPB) network.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on July 26, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#) ..... [3](#)
- [2. Terminology](#) ..... [3](#)
- [3. Requirements Language](#)..... [4](#)
- [4. Auto Attachment Framework and Applicability](#) ..... [4](#)
- [5. Auto Attachment Architecture](#)..... [6](#)
- [5.1. Element Discovery](#)..... [6](#)
- [5.2. Service Requests](#)..... [7](#)
- [5.2.1. Element Inactivity Timeout](#)..... [7](#)
- [5.3. Server Mapping Request Processing](#) ..... [7](#)
- [5.4. Server Mapping Response Processing](#) ..... [8](#)
- [5.5. Service Mapping Timeout](#)..... [8](#)
- [6. Auto Attach LLDP Extensions](#)..... [9](#)
- [6.1. AA Element TLV](#) ..... [9](#)
- [6.2. I-SID/VLAN Assignment TLV](#)..... [12](#)
- [7. Security Considerations](#)..... [14](#)
- [7.1. TLV Security Considerations](#)..... [15](#)
- [8. IANA Considerations](#)..... [15](#)
- [9. Further and Related Work](#)..... [15](#)
- [10. Log of Changes](#) ..... [15](#)
- [10.1. Changes between \[draft-unbehagen-lldp-spb-01\]\(#\) and 02](#)..... [15](#)
- [11. Acknowledgments](#) ..... [16](#)
- [12. References](#) ..... [16](#)
- [12.1. Normative References](#)..... [16](#)
- [12.2. Informative References](#)..... [17](#)

## 1. Introduction

This informational document describes a compact method of using IEEE802.1AB Link Layer Discovery Protocol (LLDP) with IEEE 802.1aq Shortest Path Bridging (SPB) network to automatically attach network devices not supporting IEEE 802.1ah to individual services in a SPB network. These network devices typically do not support SPBM, MAC-in-MAC (802.1ah), nor I-SID usage and therefore cannot easily take advantage of the SPB infrastructure without manual configuration of attachment of VLANs to I-SIDs in multiple locations. A motivation for this draft is to suggest a useful means to simplify and automate connections to PBB L2VPN based service networks such as those defined in SPBM-EVPN.

## 2. Terminology

802.1aq - defines a technology for providing a link state protocol for the control of a common Ethernet switching layer.

802.1ah - Provider Backbone Bridges (PBBs), MAC-IN-MAC encapsulation

AAC - Auto Attach Client agent that resides on a non-SPB/PBB capable element that uses LLDPDUs to request I-SID assignment for the VLANs which have been configured on its network port.

AAS - Auto Attach Server agent that processes VLAN to I-SID requests from AAC elements that are connected to a SPB BEB

BCB - Backbone Core Bridge

BEB - Backbone Edge Bridge

B-TAG - Backbone VLAN Tag

C-TAG - Customer VLAN Tag

Element - Any end device or network node that may implement the auto attach functionality

I-SID - Backbone Service Instance Identifier

IS-IS - Intermediate System to Intermediate System Protocol

LAN - Local Area Network

LLDP - IEEE 802.1AB Link Layer Discovery Protocol

SPB - IEEE 802.1aq Shortest Path Bridging

SPBM - Shortest Path Bridging, MAC mode

VLAN - Virtual Local Area Network

### 3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

### 4. Auto Attachment Framework and Applicability

This section provides an overview of the behavior of auto attachment functionality into a SPBM [802.1aq] environment and is not intended to be interpreted as normative text. Note that while the scheme proposed in this document is applicable at the link layer and is the subject of proposed standardization work in the IEEE 802.1 Working Group [[AA](#)], it is consistent with work proposed in the IETF in the Network Virtualization Overlays (nvo3) and Interface to the Routing System (i2rs) Working Groups.

The purpose of Auto Attach is to allow a non-SPB device to connect to an SPB capable networking device. The non-SPB device is called an AA client (AAC) and the SPB capable networking device is called the AA server (AAS). An AA Client is a non-SPBM device that supports some form of I-SID/VLAN binding definition and, if connectivity permits, has the ability to advertise this data to a directly connected AA Server. An AA Server is a SPBM device that potentially accepts externally generated I-SID/VLAN assignments that can be used for automated configuration purposes. The client identifies itself to the server and then requests VLAN ID to SPB ISID binding(s). The server will either accept or reject each binding request. If accepted, any traffic on the (locally significant) VLAN is forwarded through the SPB cloud on the specified ISID.

A prototype of the extension proposed in the memo was successfully implemented and tested with Open vSwitch. IEEE 802.1aq SPB software is available from multiple vendors of Ethernet switches to connect end devices and non-SPB compliant switches to the SPB enabled backbone network. Edge switches in SPBM that utilize the 802.1ah PBB encapsulation are referred to as Backbone Edge Bridges (BEB). In



support of SPBM, these bridges map a VLAN ID on the UNI to an I-SID (Individual Service ID), as defined in IEEE 802.1ah. In order to facilitate an automatic way in which a AAC can request individual service connectivity from an SPBM Backbone Edge Bridge BEB acting as a AAS, this method of using IEEE 802.1AB Link Layer Discovery Protocol (LLDP) with IEEE 802.1aq Shortest Path Bridging network can be used. These widely deployed client devices typically do not support SPBM, IEEE 802.1ah and therefore cannot easily take advantage of the SPB infrastructure without manual configuration of attachment of VLANs to I-SIDs in multiple locations.

Elements that utilize this automated method for service assignment pass this data to attached SPBM capable BEB nodes where the mappings are processed and approved or rejected. Specific actions are taken on the non-SPBM devices, referred to as Auto Attach Clients (AAC), as well as the SPBM device, referred to as Auto-Attach Server (AAS), based on the outcome of the mapping request.

Conceptual SPB Auto Attach Model

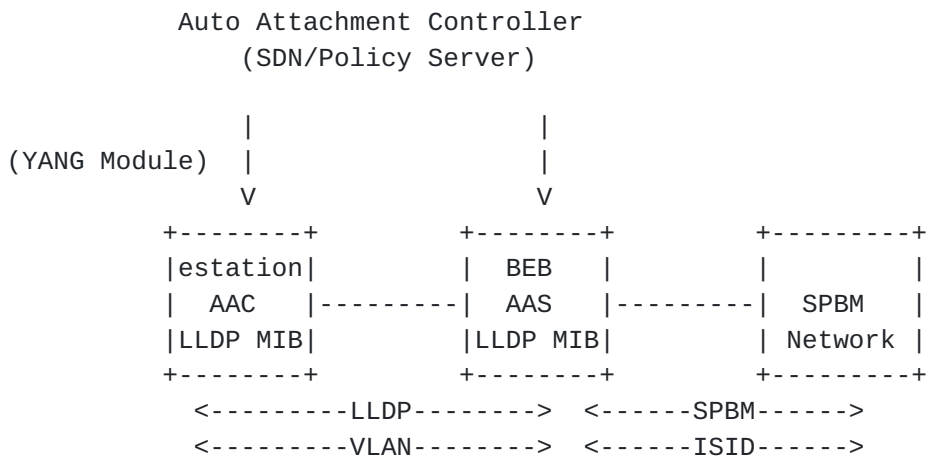


Figure 1: LLDP exchanges trigger IS-IS SPBM announcements

Figure 1 depicts a conceptual example of the process where an AAC can use LLDP to communicate the need to connect a VLAN to the appropriate I-SID on the SPB BEB it is attached to on its network uplink port. The IEEE 802.1AB Link Layer Discovery Protocol (LLDP) and the LLDP MIB are part of the Auto Attachment Framework and will be implemented in the Backbone Edge Bridges (BEB) and the Backbone Core Bridges (BCB).

An Auto Attach Client (AAC) can run in any device (end station - estation) that connects to an SBPM network. One field of applications can be for example Internet of Things (IoT) devices that connect to a network. The service association is automated with relative low resources, allowing connection of the devices to the appropriate network services and applications.

The different classes of devices that run AACs may be configured by means that are specific for the respective classes of applications or services. A YANG module will be defined as part of the Auto Attachment framework allowing for standard-based interaction with the Auto Attachment Controllers, which are instantiated as policy or Software Define Networks (SDN) servers.

## **5. Auto Attachment Architecture**

### **5.1. Element Discovery**

The first stage of establishing AA connectivity involves element discovery. An Auto Attach agent resides on all capable elements. Server agents control the Auto Attach (AA) of VLANs to I-SIDs on themselves when enabled to accept and process such requests from AAC elements. Typically this is done through a global service setting and through per-port settings that control the transmission of information in LLDPDUs on the appropriate links that interface AAC's and AAS's.

Once the required AA settings are enabled on the elements (e.g., the AA service and the per-port AA settings) the AA agent on each element type, both AAC and AAS, advertises its capabilities (i.e., server/client) through LLDPDU packets to each other.

Following discovery of AA capabilities by both the AAC and the AAS, the AA agent on each element is aware of all AA services currently provided by the network elements to which it is directly connected. Based on this information, an AAC agent can determine whether Auto Attach data, namely locally administered I-SID/VLAN assignments, should be exported to the AAS that is associated with an SBPM BEB to which it is attached to on its network uplink ports.

Initial Auto Attach functionality, when enabled, can be used to extract management VLAN data from the primary AA server advertisements and can use this data to update the in-band management VLAN and initiate IP address acquisition using techniques such as DHCP.





## **5.2. Service Requests**

Service mappings can be established when two criteria are met:

1. AA Server found during discovery

Assuming that an administrator has defined one or more ports for auto attach mode a discovery message is sent out each port defined using LLDP. Element information is forwarded using LLDP TLV extensions defined in [section 6.1](#).

2. I-SID/VLAN bindings are defined locally

Assuming that an administrator has defined one or more I-SID/VLAN assignments (or AAC bindings have been received for processing), an AAC sends the I-SID/VLAN assignment list to the discovered AAS. I-SID/VLAN data is exported using LLDP TLV extensions defined in [section 6.2](#).

### **5.2.1. Element Inactivity Timeout**

An AAC must handle primary AAS loss and this requires maintenance of a server's inactivity timer. If primary AAS advertisements are not received for a pre-determined amount of time, the I-SID/VLAN assignments accepted by the server are considered rejected. I-SID/VLAN assignment data is then defaulted (reverts to the 'pending' state) and the AA agent, which resides on the AAC, removes related settings.

### **5.3. Server Mapping Request Processing**

Each I-SID/VLAN assignment in an AA request received by the AAS is processed individually and can be accepted or rejected. An assignment may be rejected for a number of reasons, such as server resource limitations or, for example, restrictions related only to the source AAC. Rejected assignments are passed back to the originating AAC with a rejected state and, if appropriate, an indication as to why the rejection occurred. Limited state information may be maintained on the server related to rejected I-SID/VLAN assignments.

Each VLAN that is associated with an accepted I-SID/VLAN assignment is instantiated on the AAS bridge if it does not already exist. These VLANs are designated SPBM UNI VLANs on a BEB. The port through which the AA I-SID/VLAN assignment list was received (i.e., the AAS downlink) must be a member of the VLAN(s) in the I-SID/VLAN assignment list that are accepted by the AAS. Port membership is



automatically updated when the UNI service (I-SID/VLAN/port) is created. To ensure that VLAN markings are maintained between switches, traffic on the downlink port MUST be tagged. The AA agent on the serving BEB handles all of these tasks automatically. No administrator intervention is required.

The AAS agent is responsible for tracking which, if any, of these actions are performed so that settings can be cleared when they are no longer needed. This can occur, for example, when configuration changes on an AAC updates the received I-SID/VLAN assignment list when an AAC associated with a downlink port changes or an AAC connection disappears entirely. Specifically, when an SPBM switched UNI-based VLAN and a switched UNI have been created on a downlink port because of an accepted AA I-SID/VLAN assignment (and not because of an explicit administrator port action), then the UNI and associated VLAN SHOULD be deleted when the related I-SID/VLAN assignment is cleared by the AAS.

#### **5.4. Server Mapping Response Processing**

Each VLAN that is associated with an AAC I-SID/VLAN assignment must be defined on the client's device. The port associated with the uplink connecting the AAC to the AAS must be a member of the VLAN(s) in the I-SID/VLAN assignment list that are sent to and accepted by the AAS. This allows traffic on these VLANs to pass through the switch into the SPB fabric when required. To ensure that VLAN markings are maintained between devices, traffic on the uplink port MUST be tagged. If a VLAN has not been created before the I-SID/VLAN assignment itself, it is automatically created by the AAC agent when a proposed assignment is accepted. Port tagging and the port VLAN membership update are also performed by the AAC automatically based on assignment acceptance. To ensure consistency, VLANs SHOULD NOT be deleted while they are referenced in any I-SID/VLAN assignments on the device.

#### **5.5. Service Mapping Timeout**

A "last updated" timestamp is associated with all active assignments on the AAS. When this value is not updated for a pre-determined amount of time, the I-SID/VLAN assignment is considered obsolete. Obsolete assignment data and related settings are removed by the AAS, subject to the constraints imposed by [section 4.3](#).

The current I-SID/VLAN assignment list is advertised by an AAC at regular intervals (dictated by LLDP operation). During processing of this data, an AAS must handle list updates and delete assignments from previous advertisements that are no longer present. Though



these entries would be processed appropriately when they timeout, the AAS attempts to update the data in real-time and SHOULD initiate deletion immediately upon detection of this condition.

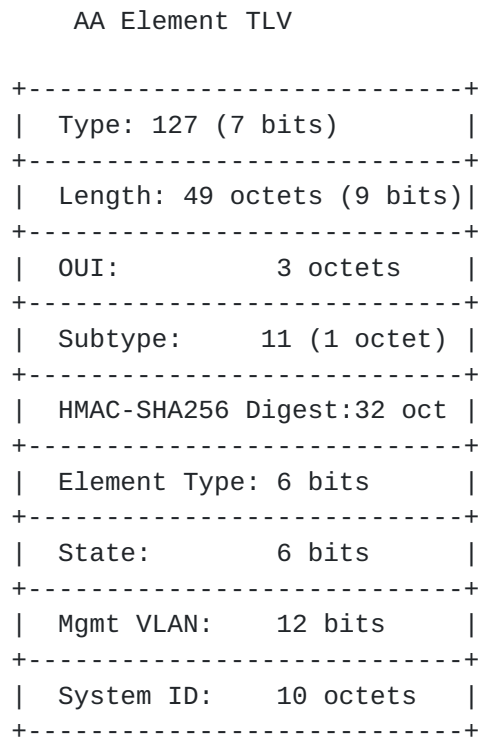
6. Auto Attach LLDP Extensions

The text in this section is not normative. The complete definition of the Auto Attach TLVs is provided in the IEEE 802.1Qcj [AA] Amendment of the IEEE 802.1Q standard.

The Auto Attach TLVs are implemented as extensions to the LLDP standard, using its flexible extension mechanism. They SHOULD be implemented as vendor-specific TLVs using TLV type 127 as described in the 802.1AB (LLDP) standard. TLVs supporting the exchange of AA element data and I-SID/VLAN assignment data have been defined below.

6.1. AA Element TLV

The Element TLV is used by an AA device to announce its capabilities to its LLDP peer on a given interface. Use of the Auto Attach functionality is encoded in to the 802.1AB LLDP Custom Element TLV as follows:



Subtype = 11 for AA Element TLV

HMAC-256 Digest:

The Element TLV data integrity and source validation is supported through the use of the HMAC-SHA256 message authentication algorithm. The HMAC-SHA256 generated digest size is 32 octets and the Element TLV includes a field to support the digest exchange between source and destination parties. Symmetric private keys are used for digest generation.

The HMAC-SHA256 data digest computation starts at [0-based] byte 38 of the TLV. The digest is then placed in the HMAC-SHA256 Digest field in the TLV prior to transmission. Upon receipt, the digest is again computed and the resulting digest is compared against the received digest. If the received digest is the same as the newly computed digest, the TLV is considered valid and processing can commence. If the comparison fails, the TLV is discarded and processing is terminated.

Element Type:

The element type identifies the capability of the advertising AA node. The AA Server describes an AAS capable device that can map incoming VLAN to I-SID and announce I-SID connectivity to the SPB network. AA Clients may operate in either tagged or untagged modes. If an AA client announces untagged, then the entire port MUST be mapped to the I-SID on the BEB.

The AA Element TLV can only exist once in a LLDPDU. It is included in all LLDPDUs when the Auto Attach service is enabled and when the per-port transmission flags associated with this TLV, as required by the 802.1AB standard, are enabled.

A number of AA Element Type values, including the AA Server and several AA Client element types, are currently defined. The list of supported element types will expand as additional devices incorporate AA signaling.

Currently supported Auto Attach Element Type values:

AA Element Type - Other (1)

AA Server (2)

AA Server No Authentication (3)

AA Client - Wireless Access Point Type 1 (4) [wireless clients get direct network attachment]

AA Client - Wireless Access Point Type 2 (5) [wireless clients get tunneled to a controller]

AA Client - Switch (6)

AA Client - Router (7)

AA Client - IP Phone (8)

AA Client - IP Camera (9)

AA Client - IP Video (10)

AA Client - Security Device (11) [FW, IPS/IDS, etc.]

AA Client - Virtual Switch (12)

AA Client - Server/Endpoint (13)

AA Client - SDN Controller (14)

AA Client - SPB-over-IP Network Device (15)

State:

The AA Element TLV State field settings indicate AA Client link tagging requirements in AA Client-sourced frames and current provisioning mode information (bits are numbered left to right):

Link VLAN Tagging Requirements (bit 1)

0 - All traffic tagged on link

1 - Tagged and untagged traffic on link

Automatic Provisioning Mode (bits 2/3)

0 - Automatic provisioning disabled

- 1 - SPB provisioning
- 2 - VLAN provisioning

System ID: conveys information that the TLV recipient can use to enforce connectivity restrictions. It includes System MAC Address, connection type and identifiers. Detailed specification of the System ID sub-fields is TBD.

**6.2. I-SID/VLAN Assignment TLV**

The AA I-SID/VLAN Assignment TLV is used by the AAC to announce I-SID/VLAN assignments that it would like supported by a directly connected AAS. It is also used by the AAS to announce that that I-SID/VLAN bindings processed by the AAS are active or rejected.

The AA I-SID/VLAN Assignment TLV can only exist once in a LLDPDU. It is only included in a LLDPDU when complementary AA element (i.e., AA server/ client) devices are directly connected. Data integrity and source validation is supported through the use of the HMAC-SHA256 message authentication algorithm. The HMAC-SHA256 generated digest size is 32 octets and the AA I-SID/VLAN Assignment TLV includes a field to support the digest exchange between source and destination parties.

Per-port TLV transmission flags must be enabled on the communicating devices as well. The AA Element TLV must also be present in the LLDPDU for the AA I-SID/VLAN Assignment TLV to be processed. The TLV cannot exceed the LLDP 512 byte TLV size limit, which implies a maximum of 94 I-SID/VLAN assignments in a LLDPDU

The format of the TLV is as follows:

Service Assignment TLV

```

+-----+
| Type:  127 (7 bits)      |
+-----+
| Length: 41-506 octets (9 bits)|
+-----+
| OUI:                3 octets  |
+-----+

```



```

| Subtype:          12 (1 octet) |
+-----+
| HMAC-SHA256 Digest: 32 octets |
+-----+
| Assignment Status: 4 bits      |
+-----+
| VLAN:             12 bits      |
+-----+
| I-SID:            3 octets     |
+-----+

```

The HMAC-SHA256 digest is computed for the series (1-94) of I-SID/VLAN assignments (i.e. data for the digest computation starts at [0-based] byte 38 of the TLV). The digest is then placed in the HMAC-SHA256 Digest field in the TLV prior to transmission. Upon receipt, the digest is again computed for the series (1-94) of I-SID/VLAN assignments in the received TLV and the resulting digest is compared against the received digest. If the received digest is the same as the newly computed digest, the TLV is considered valid and processing can commence. If the comparison fails, the TLV is discarded and processing is terminated. Additionally the value for the I-SID in the incoming LLDP exchanges SHOULD trigger an IS-IS SPBM announcement using normal IEEE 802.1aq mechanisms if not already being announced by the BEB.

The assignment status data is returned by the AA Server for each pending I-SID/VLAN assignment request. Assignment rejections may include information to indicate the reason for the rejection. A limited number of detailed rejection error codes will initially be supported.

Assignment Pending(1)

Assignment Accepted(2)

Rejection: Generic(3)

Rejection: AA resources unavailable(4) - the resources that are required for the Auto Attach agent to support additional I-SID/VLAN assignments are currently exhausted. The maximum number of assignments that can be supported has been reached.

Rejection: Duplicate(5)

Rejection: VLAN invalid(6) - the specified VLAN can't be used to create a switched UNI at this time. The VLAN already exists and is either inactive or has an incorrect type for this application.

Rejection: VLAN unknown(7)

Rejection: VLAN resources unavailable(8) - the maximum number of VLANs that can be supported by the device has been reached.

Rejection: Application interaction issue(9) - a failure has been detected during AA interactions with the VLAN and/or the SPBM applications. The VLAN operations to create the required SPBM switched UNI VLAN or enable port tagging may have failed or the SPBM operation to create the switched UNI may have failed

Please note that the status field is only valid when generated by an AA Server. Any Assignment TLVs which are received by an AA server are assumed to be requests. It is recommended that the status field of assignments generated by AA clients be set to 0 or 1.

VLAN: A VLAN value of 0 may indicate that AAC traffic is untagged.

## 7. Security Considerations

It is important to provide an option to ensure that the aforementioned Auto Attach communication is secure in terms of data integrity (i.e., the data has not been altered in transit) and authenticity (i.e., the data source is valid).

If communication is occurring between non-secure systems, the HMAC-SHA256 Digest data should always be zero and the digest data, regardless of the value, is ignored. A misconfiguration can occur with one system operating in secure mode and the other operating in non-secure mode. In this scenario, the Element TLV or the I-SID/VLAN Assignment TLV will always be discarded prior to processing by the system operating in secure mode.

These security requirements are satisfied by using an optional keyed-hash message authentication code (HMAC) to protect the AAC/AAS Element Discovery and I-SID/VLAN assignment exchanges. This type of message authentication allows communicating parties to verify that the contents of the message have not been altered and that the source is authentic. Use of this mechanism is optional and is controlled through a user-configurable attribute.



### **7.1. TLV Security Considerations**

A HMAC-SHA256 digest is computed for Element TLV or for the series of I-SID/VLAN assignments, where the digest computation starts [0 based] at byte 38 of the TLV. The resulting digest is then placed in the TLV prior to sending. Where upon receipt of the digest, the contents are again computed in the same manner and the digests are compared, if the comparison fails then the TLV is discarded, otherwise if both digests are the same the TLV is considered valid and processed appropriately.

### **8. IANA Considerations**

This memo includes no request to IANA.

Note: the section will be removed during conversion into an RFC by the RFC Editor.

### **9. Further and Related Work**

The standard extensions to the IEEE 802.1AB (LLDP) [[LLDP](#)] protocol are developed by the IEEE 802.1 Working Group. The relevant project is IEEE 802.1Qcj [[AA](#)] for 'Automatic Attachment to Provider Backbone Bridges (PBB) services'.

Current open issues:

- Define whether an AA Proxy needs to be made part of the architecture and if yes, define its role
- Further details on the two AA TLVs fields
- Define semantics of I-SID value of 0
- Alignment with the (normative) definitions in IEEE 802.1Qcj (as they progress)

The Auto Attachment YANG data model is developed as [TBA1].

The Auto Attachment Controller logic (defining the VLAN/ISID mapping logic) is developed as [TBA2].

### **10. Log of Changes**

(this section to be removed when RFC is published)

#### **10.1. Changes between [draft-unbehagen-lldp-spb-01](#) and 02**

- Added definition of the Auto Attach framework and principal components

- Updated Figure 1
- Updated the AA Element TLV fields
- Added several Element Type values
- Added Rejection error codes in I-SID/VLAN Assignment TLV
- Defined references to the IEEE 802.1Qcj approved project
- Various editorial improvements and corrections

## **11. Acknowledgments**

We would like to thank the following people (in no particular order) for their contributions:

Zenon Kuc

Cristian Mema

Roger Lapuh

Craig Griffin

Chris Buerger

Keith Krajewski

This document was prepared using 2-Word-v2.0.template.dot.

## **12. References**

### **12.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [LLDP] IEEE STD 802.1AB, "IEEE Standard for Local and Metropolitan Area Networks-- Station and Media Access Control Connectivity Discovery", 2005.
- [PBB] IEEE STD 802.1ah, "IEEE Standard for Local and Metropolitan Area Networks / Virtual Bridged Local Area Networks / Amendment 7: Provider Backbone Bridges", 2008.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC6329] Fedyk, D., Ashwood-Smith, P., Allan, D., Bragg, A. and P. Unbehagen, "IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging", [RFC 6329](#), April 2012.



[SPB] IEEE STD 802.1aq, "IEEE Standard for Local and metropolitan area networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment 20: Shortest Path Bridging", 2012.

## 12.2. Informative References

[AA] IEEE 802.1Qcj, "Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Automatic Attachment to Provider Backbone Bridging (PBB) services"

### Authors' Addresses

Paul Unbehagen Jr, editor  
Avaya  
1300 W. 120th Avenue  
Westminster, CO 80234  
US  
Email: unbehagen@avaya.com

Dan Romascanu  
Avaya  
Azrieli Center Holon  
26, HaRokhmim Str., Bldg. D  
Holon, 5885849  
Israel  
Phone: +972-3-645-8414  
Email: dromasca@avaya.com

John Seligson  
Avaya  
4655 Great America Parkway  
Santa Clara, CA 95054  
US  
Email: jseligso@avaya.com

Carl Keene  
Avaya  
600 Technology Park Dr  
Boston, MA 01821  
US  
Email: ckeene@avaya.com

Nigel Bragg  
Ciena  
Ciena House  
43-51 Worship Street  
London, EC2A 2DX  
Uk  
Email: nbragg@ciena.com

Ludovic Beliveau  
Wind River  
Email: Ludovic.Beliveau@windriver.com