IEEE P802.11
Wireless LANs

|  |
| --- |
| TGaq – LB208 D1.0 - Editor’s comments |
| Date: 2015-02-23 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Dan Gal | Alcatel-Lucent | 600 Mountain Ave, Murray Hill,NJ 07974, United States | +1 407 416 7435 | ddrgal@gmail.com ;dan.gal@alcatel-Lucent.com |

Abstract

This document proposes changes to TGaq Draft D1.0’s section, table and figure numbering to align it with those of draft 802.11 REVmc D4.0, as well as other editorial changes.

This submission supports a general LB 208 D1.0 ballot comment by the author.

# 8. Frame Formats

**8.4.2 Elements**

**8.4.2.1 General**

*<Insert the following rows (ignoring the header row) in* [Table 8-74](#Table_8_83) *after the preceding amendment’s last entry >*

**Table 8-74 – Element IDs**

|  |  |  |
| --- | --- | --- |
| Element | Element ID | Extensible |
| Service Hint ( see [8.4.2.171](#section_8_4_2_171)) | <ANA> |  |
| Service Advertisement ( see [8.4.2.172](#section_8_4_2_172)) | <ANA> |  |
| Service Hash (see [8.4.2.173](#section_8_4_2_173)) | <ANA> |  |
| Supported ULP (see [8.4.2.174](#section_8_4_2_174)) | <ANA> |  |

**8.4.2.26 Extended Capabilities element**

*<Insert the following new row (ignoring the header row) in* **Table 8-132** *after the preceding amendment’s last entry>*

|  |
| --- |
| **Table 8-132 – Capabilities field** |
| **Bit** | **Information** | **Notes** |
| <ANA>\* | Pre-association Discovery (PAD) | When dot11UnsolicitedPADActivated, dot11SolicitedPADActivated or dot11EncapsulatedPADActivated is true, the PAD field is set to 1 to indicate the STA supports the PAD service as described in 10.25.3When dot11UnsolicitedPADActivated and dot11SolicitedPADActivated and dot11EncapsulatedPADActivated are false, the PAD field is set to 0 to indicate the STA does not support this capability. |

*\* Editor’s Note: The <ANA> flag will be replaced by a value assigned by the IEEE 802.11 Assigned Numbers Authority (ANA) before completion of sponsor ballot.*

**8.4.2.92 Advertisement Protocol element**

*<Insert a new row in* **[Table 8-210](#Table_8_219)***, after the ‘Registered location query protocol (RLQP)’ table entry, and change the* ***Reserved*** *values accordingly>*

|  |
| --- |
| **Table 8-210 - Advertisement protocol ID definitions** |
| **Name** | **Value** |
| Access Network Query Protocol for service discovery (ANQP-SD) |  5  |

*<Insert a new dashed-list item (shown in red) after ‘The RLQP support information…’>*

—The ANQP-SD supports service information retrieval using ANQP-elements. It is used by a requesting STA to query another STA (i.e., the receiving STA can respond to queries with and without proxying the query to a server in an external network). The use of an alternative Advertisement protocol ID allows the receiving STA to proxy the query to an alternative server in an external network. See clause [10.25.3.2.11](#section_10_25_3_2_11_ANQP_SD_procedures) for information on ANQP-SD procedures.

*<Insert the following four new subclauses, at the end of clause 8.4.2.* >

**8.4.2.171 Service Hint element**

The Service Hint element contains information identifying services that are supported by an AP. The Service Hint element is transmitted in beacons.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length | Bloom Filter Information |  *m*-bit Service Hint Map |
| Octets | 1 | 2 | 2 |  variable |

**Figure 8-577a – Service Hint element format**

The Element ID field and Length field are defined in [8.4.2.1](#Section_8_4_2_1) (General).

The value of the Length field is variable and is 2 plus the variable-length *m*-bit Service Hint Map field.

The Bloom Filter Information field is a 2-octet field, representing the settings of the Bloom filter. The format of the Bloom Filter Information field is shown in Figure 8-577b.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Number of services | Number of Hash functions | Reserved |
| Bits | 0-8 | 9-12 | 13-15 |

**Figure 8-577b – Bloom Filter Information field format**

The Number of services field is used to indicate the maximum number of services, *n, that* can be supported by the AP. The maximum number of services is 512.

The Number of Hash functions field is used to indicate the number of hash functions, *k*, (out of the maximum of 16) used by the Bloom filter.

The m-bit Service Hint Map (in Figure 8-577a) provides an indication about the services offered by the AP, using the Bloom filter. For more information on the operation of the Bloom filter Hash function, see section [10.25.3.4.5](#section_10_25_3_4_5), as well as Annex [Za.4](#Annex_Za_4_Bloom_Filter)

**8.4.2.172 Service Advertisement element**

The Service Advertisement element identifies a service, advertised by an AP.

The Service Advertisement element is included in the Probe Response returned by the AP in response to a Probe Request from a non-AP STA that has one or more matching Service Hashes.
For each matching Service Hash, the AP includes a corresponding Basic Service Information Descriptor.

The format of the Service Advertisement element is shown in Figure 8-577c.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | Basic Service Information Descriptors |
| Octets | 1 | 1 | Variable |

**Figure 8-577c – Service Advertisement element format**

The Element ID and Length fields are defined in [8.4.2.1](#Section_8_4_2_1) (General).

The Basic Service Information Descriptors field contains one or more Basic Service Information Descriptor sub-fields. The format of the Basic Service Information Descriptor sub-field is shown in Figure 8-577d.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AdvertisementID | Service Name Length | Service Name | ServiceStatus |
| Octets | 4 | 1 | variable | 1 |

**Figure 8-577d – Basic Service Information Descriptor sub-field format**

The Advertisement ID field is a 4-octet unsigned integer assigned by the AP when advertising a service.

The Service Name Length field is the length of the Service Name field.

The Service Name field is a UTF-8 encoded string with a maximum length of 63 bytes. It may be an official IANA registered name, as defined in RFC 6335, or a developer-specified name.

The Service Status sub-field is a 1-octet long, indicating the current status of the service as shown in Table 8-248a.

**Table 8-248a – Service Status sub-field value**

|  |  |
| --- | --- |
| **Service Status value** | **Description** |
| 0 | not available |
| 1 | available |
| 2-255 | reserved |

**8.4.2.173 Service Hash element**

The Service Hash element consists of Service Hash Values. The Service Hash element may be included in the Beacon and the Probe Request frames.

The format of the Service Hash element is shown in Figure 8-577e.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | Service Hash Value |
| Octets | 1 | 1 | Multiples of 6-octets with each 6-octet representing a service  |

**Figure 8-577e – Service Hash element format**

The Element ID field and Length fields are defined in [8.4.2.1](#Section_8_4_2_1) (General).

The Service Hash Value field contains one or more Service Hash Values. The Service Hash Value is formed from the value of a service name by using the first 6 octets of the SHA-256 hashing algorithm of the service name value.

**8.4.2.174 Supported ULP element**

The Supported Upper Layer Protocol (ULP) element is used to indicate the ULP supported by the AP. The Supported ULP element may be included in the Beacon frame and the Probe Response frame. The format of the Supported ULP element is shown in Figure 8-577f:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | Supported ULP Bitmap  |
| Octets | 1 | 1 | 4 |

**Figure 8-577f – Supported ULP element format**

The Element ID field and Length fields are defined in [8.4.2.1](#Section_8_4_2_1) (General).

The Supported ULP Bitmap field is a 4-octet field that represents the set of ULPs supported by the AP. The bitmap encoding of this field is shown in Table 8-248b. A bit value of “1” indicates “supported” and a bit value of “0” indicates “not supported”, or “reserved”.

**Table 8-248b – Supported ULP Bitmap**

|  |  |  |
| --- | --- | --- |
| **ULP name** | **ULP Abbreviation** | **Bit** |
| DNS Service Discovery, part of Apple’s Bonjour technology | **DNS-SD, Bonjour**See: IETF, RFC 6763, DNS-Based Service Discovery, February, 2013<https://www.ietf.org/rfc/rfc6763.txt><https://developer.apple.com/bonjour/index.html> | 0 |
| Service Location Protocol | **SLP**See: IETF, RFC 2609, Service Location Protocol, June, 1999. <http://www.ietf.org/rfc/rfc2609.txt> | 1 |
| Simple Service Discovery Protocol (as used in Universal Plug and Play) | **SSDP, UPnP**See: UPnP FORUM, UPnP Device Architecture 1.0, October, 2008. <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf> | 2 |
| Universal Description Discovery and Integration for web services | **UDDI**See: OASIS, UDDI Version 3.0, October, 2004. <http://www.uddi.org/pubs/uddi_v3.htm> | 3 |
| Jini for Java objects. | **JINI**See: <http://river.apache.org/doc/specs/html/jini-spec.htm> | 4 |
| Bluetooth Service Discovery Protocol | **SDP**See: Bluetooth Special Interest Group, “Bluetooth Specification Version 4.1, Vol. 3: Core System Package, Part B: Service Discovery Protocol Specification,” December, 2013. <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=282159> | 5 |
| Salutation | **Salutation**See: The Salutation Consortium, Salutation Architecture Specification Version 2.0c, June, 1999. <http://salutation.org> | 6 |
| XMPP Service Discovery | **XEP-0030**See: XMPP Standard Foundation, XEP-0030: Service Discovery, Version 2.4, June, 2008. <http://xmpp.org/extensions/xep-0030.html> | 7 |
| Web Services Dynamic Discovery | **WS-Discovery**See: OASIS, Web Service Dynamic Discovery Version 1.1, July, 2009. <http://docs.oasis-open.org/ws-dd/discovery/1.1/os/wsdd-discovery-1.1-spec-os.html> | 8 |
| multicast DHCP | **MDHCP**See: IETF, RFC 2131, Dynamic Host Configuration Protocol, March, 1997. <https://www.ietf.org/rfc/rfc2131.txt> | 9 |
| Internet Storage Name Service | **iSNS**See: IETF, RFC 4171, Internet Storage Name Service, September, 2005. <https://www.ietf.org/rfc/rfc4171.txt> | 10 |
| Web Proxy Autodiscovery Protocol | **WPAD**See: See: IETF, Internet-draft, Web Proxy Auto-Discovery Protocol, December, 1999. <http://tools.ietf.org/id/draft-ietf-wrec-wpad-01.txt> | 11 |
| Dynamic Host Configuration Protocol | **DHCP**See: IETF, RFC 2131, Dynamic Host Configuration Protocol, March, 1997. <http://www.ietf.org/rfc/rfc2131.txt> | 12 |
| eXtensible Resource Descriptor Sequence | **XRDS**See: OASIS, Extensible Resource Descriptor (XRD) Version 1.0, November, 2010. <http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.html> | 13 |
| e911 (Emergency Service) | **e911**See: <http://www.nena.org/?page=Standards> | 14 |
| Next Generation 911 (Emergency Service) | **NG911**See: <http://www.nena.org/?page=Standards> | 15 |
| Location Service | **Location** | 16 |
| MQTT |  | 17 |
| Reserved |  | 18-30 |
| Vendor-specific |  | 31 |

*<Insert three new entries (shown in red**) in Section 8.4.5, Table 8-257>*

**8.4.5 Access network query protocol (ANQP) elements**

|  |
| --- |
| **Table 8-257 – ANQP-element definitions**  |
| **ANQP-element name** | **Info ID** | **ANQP- (Ed)element (subclause)** |
| Service Information Request | <ANA> | 8.4.5.20 (Service Information Request ANQP-element) |
| Service Information Response | <ANA> | 8.4.5.21 (Service Information Response ANQP-element) |
| ULP Encapsulation | <ANA> | 8.4.5.22 (ULP Encapsulation ANQP-element) |
| Reserved | <ANA> – 56797 | 8.4.5.8 (VendorSpecific ANQP element) |

*<Insert the following* ***three*** *new subclauses following subclause 8.4.5.19 >*

**8.4.5.20 Service Information Request ANQP-element**

The Service Information Request ANQP-element is sent by the non-AP STA to the AP and used to request service information. It is included in a GAS Query Request.

The format of the Service Information Request ANQP-element is shown in Figure 8-607a.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Info ID | Length | Service Name Length | Service Name | Service Information Query Request Length | Service Information Query Request |
| Octets: | 2 | 2 | 1 | variable | 1 | variable |

**Figure 8-607a – Service Information Request ANQP-element format**

The Info ID and Length fields are defined in 8.4.4.1

The Service Name Length and Service Name fields are defined in [8.4.2.172](#section_8_4_2_172).

The Service information Query Request Length contains the length of the Service Information Query Request field.

The Service Information Query Request field contains service-specific query, such as key-value query.

The procedure used for this element is described in clause [10.25.3.2.11.1](#_10.25.3.2.11.1_ANQP-SD_Service).

**8.4.5.21 Service Information Response ANQP-element**

The Service Information Response ANQP-element is used to provide detailed service information between STAs, using the GAS protocol, in response to a Service Information Request ANQP-element. The Service Information Response ANQP-element is included in a GAS Query Response, sent by the AP to the non-AP STA.

The format of the Service Information Response ANQP-element is shown in Figure 8-607b.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Info ID | Length | Detailed Service Information Descriptors |
| Octets | 2 | 2 | variable |

**Figure 8-607b - Service Information Response ANQP-element format**

The Info ID and Length fields are defined in 8.4.4.1

The Detailed Service Information Descriptors field contains one or more Detailed Service Information Descriptor sub-fields (Figure 8-607c).

The format of the Detailed Service Information Descriptor sub-field is shown in Figure 8-607c

|  |  |  |  |
| --- | --- | --- | --- |
|  | Basic Service Information Descriptor | Service Information Query Response Length | Service Information Query Response |
| Octets | variable  | 2 | variable |

**Figure 8-607c – Detailed Service Information Descriptor sub-field format**

The Basic Service Information Descriptor field is defined in [8.4.2.172](#section_8_4_2_172)

The Service Information Query Response field is a variable length field. The format of the Service Information Query Response is service-specific that contains requested service information.

The procedure used for this element is described in clause [10.25.3.2.11.2](#_10.25.3.2.11.2_ANQP-SD_Service)

**8.4.5.22 ULP Encapsulation ANQP-element**

ULP Encapsulation ANQP-element is used to exchange upper layer protocol (ULP) frames between STAs, using the GAS protocol. ULP Encapsulation ANQP-element is used as a request, included in a GAS Query Request, or returned as a response in the GAS Query Response.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Info ID | Length | Payload |
| Octets: | 2 | 2 | variable |

**Figure 8-607d – ULP Encapsulation ANQP-element format**

The Info ID and Length fields are defined in 8.4.4.1

The format of the Payload sub-field is an encapsulated upper layer protocol (ULP) frame.
For more information on the ULPs, see [Table 8-257b](file:///C%3A%5C%23%20MY%20FOLDERS%5C---%20IEEE%20802.11aq%20Drafts%20%26amp%3B%20Ballots%5C%5E%20D1.0%20Letter%20Ballot%5CP802.11aq%20-%20Draft%20D1.0a_YY.docx#Table_8_257b).

The procedure used for this element is described in clause [10.25.3.2.11.3](#_10.25.3.2.11.3_ANQP-SD_ULP).

**10. MLME**

**10.25 WLAN interworking with external networks procedures**

**10.25.3 Interworking procedures: generic advertisement service (GAS)**

*<insert the following (red) text in the beginning of section 10.25.3.2.1>*

**10.25.3.2 ANQP procedures**

**10.25.3.2.1 General(Ed)**

In this clause, ANQP refers to the Advertisement Protocols indicated by the Advertisement Protocol IDs 0 and 5.

*<add a new column and new elements in Table 10-16, as shown>*

|  |  |
| --- | --- |
| **Table 10-16 - ANQP usage (11u)** |  |
|  |  | **BSS** | **IBSS** | **Advertisement Protocol ID** |
| **ANQP-element Name** | **ANQP-element (subclause)(Ed)** | **ANQP-element Type** | **AP** | **Non-AP STA** | **STA** |  |
| Query List | 8.4.5.2 (Query List ANQP-element) | Q | T, R | T, R | T, R | 0 |
| Capability List | 8.4.5.3 (Capability List ANQP-element) | S | T, R | T, R | T, R | 0 |
| Venue Name | 8.4.5.4 (Venue Name ANQP-element ) | S | T | R | — | 0 |
| Emergency Call Number | 8.4.5.5 (Emergency Call Number ANQP-element ) | S | T | R | — | 0 |
| Network Authentication Type | 8.4.5.6 (Network Authentication Type ANQP-element) | S | T | R | — | 0 |
| Roaming Consortium | 8.4.5.7 (Roaming Consortium ANQP- element) | S | T | R | — | 0 |
| Vendor Specific | 8.4.5.8 (Vendor Specific ANQP-element) | Q, S | T, R | T, R | T, R | 0 |
| IP Address Type Availability | 8.4.5.9 (IP Address Type Availability ANQP-element ) | S | T, R | T, R | T, R | 0 |
| NAI Realm | 8.4.5.10 (NAI Realm ANQP-element) | S | T | R | T, R | 0 |
| 3GPP Cellular Network | 8.4.5.11 (3GPP Cellular Network ANQP-element) | S | T | R | — | 0 |
| AP Geospatial Location | 8.4.5.12 (AP Geospatial Location ANQP-element) | S | T | R | T, R | 0 |
| AP Civic Location | 8.4.5.13 (AP Civic Location ANQP-element)  | S | T | R | T, R | 0 |
| (#13006)AP Location Public Identifier URI | 8.4.5.14 (AP Location Public Identifier URI ANQP-element)  | S | T | R | T, R | 0 |
| Domain Name | 8.4.5.15 (Domain Name ANQP-element) | S | T | R | — | 0 |
| Emergency Alert Identifier URI | 8.4.5.16 (Emergency Alert URI ANQP-element) | S | T | R | T, R | 0 |
| TDLS Capability (#13018) | 8.4.5.18 (TDLS Capability ANQP-element) | Q, S | T,R | T,R | T, R | 0 |
| Emergency NAI | 8.4.5.17 (Emergency NAI ANQP-element) | S | T | R | — | 0 |
| Neighbor Report | 8.4.5.19 (Neighbor Report ANQP-element) | S | T | R | - | 0 |
| Service Information Request | 8.4.5.20 (Service Information Request ANQP-element) | Q | T, R | T,R | T, R | 5 |
| Service Information Response | 8.4.5.21 (Service Information Response ANQP-element) | S | T, R | T, R | T, R | 5 |
| ULP Encapsulation  | 8.4.5.22 (ULP Encapsulation ANQP-element | Q, S | T,R | T,R | T, R | 5 |
| **Symbols**Q element is an ANQP queryS element is an ANQP responseT ANQP-element may be transmitted by MAC entityR ANQP-element may be received by MAC entity— ANQP-element is neither transmitted nor received by MAC entity |  |

*<Insert the following new clause and subclauses after 10.25.3.2.10>*

**10.25.3.2.11 ANQP-SD procedures**

ANQP-SD uses an alternative Advertisement Protocol ID (ID=5) as opposed to the non-service discovery ANQP (Advertisement Protocol ID=0). This is to allow the receiving STA to proxy ANQP-SD queries to an alternative server in an external network, if required. The receiving STA may also directly respond to ANQP-SD queries.

Since a GAS query only has a single Advertisement Protocol ID, a requesting STA shall not send a mixture of ANQP and ANQP-SD queries simultaneously. If the receiving STA or server in an external network receives an ANQP-element that is not supported, it is discarded.

**10.25.3.2.11.1 Service Information Request procedure**

The Service Information Request ANQP-element (see 8.4.5.20) is used by a requesting STA to perform an ANQP-SD request using the procedures defined in 10.25.3.2.1.

The Service Information Request ANQP-element is used to discover available services within the BSS. A Service Name may be placed within the request. The Service Name is used within the BSS to assist with discovering services, as described in Annex Za.

The Service Discovery Information Request ANQP element is re-directed to the proxy as described in Annex Za, as this query is directed to the Service Information Server, as opposed to an ANQP Advertisement Server.

If no Service Name value is present, the BSS will return all known services within the response.

**10.25.3.2.11.2 Service Information Response procedure**

The Service Information Response ANQP-element is returned in response to a Service Information Request ANQP-element. It contains a list of Service Information Descriptors resulting from the service discovery as described in Annex AQ

**10.25.3.2.11.3 ULP Encapsulation procedure**

The ULP Encapsulation element (see 8.4.5.22) is used by STAs to allow the transmission of upper layer protocol frames using ANQP-SD request and responses using the procedures defined in [10.25.3.2.11](#section_10_25_3_2_11_ANQP_SD_procedures).

The ULP Encapsulation ANQP-element is re-directed to the proxy as described in Annex Za, as this query is directed to the Service Information Server, as opposed to an ANQP Advertisement Server.

The ULP Encapsulation ANQP-element provides a means to exchange service discovery information between STAs. The elements support multiple service discovery protocols.

*<Insert the following new clause and subclauses after clause 10.25.3.3 >*

**10.25.3.4 Pre-association discovery (PAD) protocol procedures**

**10.25.3.4.1 General**

PAD provides functionality that enables STAs, to discover the availability of services offered by an AP, before they associate with the WLAN. While the specification of service-specific information is outside the scope of this standard, the AP can act as a proxy to the services, offered by external network or services offered by non-AP STAs associated with the AP.

There are two types of PAD, unsolicited and solicited:

In the unsolicited PAD, basic service advertisement information is included in the Beacons transmitted by the AP (see 8.4.2.171-173). Upon receiving the Beacon frames, the non-AP STAs can make an informed decision to associate with the AP, or query for more detailed service information, using ANQP-SD as described in ([8.4.5.20-21](#section_8_4_4_20_Service_info_request)) before association.

In the solicited PAD, basic service information is included in the Probe Request transmitted by the non-AP STA. Upon receiving the Probe Request, the AP responds with a Probe Response only if there is a service match between the non-AP STA and the AP. The non-AP STAs can make an informed decision to associate with the AP, or query for more detailed service information, using ANQP as described in ([8.4.5.20-21](#section_8_4_4_20_Service_info_request)) before association.

**10.25.3.4.2 Unsolicited PAD procedure**

An AP having dot11UnsolictedPADActivated equals to true shall include a Service Hint element or Service Hash element or both in Beacon frames.

The non-AP STA may associate with the AP based on the received Service Hint element, or may use a Service Information Request ANQP-element to request more detailed information as defined in Table 8-184 prior to association. The receiving AP shall respond to the Service Information Request ANQP-element with a Service Information Response ANQP-element.

**10.25.3.4.3 Solicited PAD procedure**

A non-AP STA having dot11PADActivated equals to true may send Probe Request containing Service Hash element ([8.4.2.173](file:///C%3A%5C%23%20MY%20FOLDERS%5C---%20IEEE%20802.11aq%20Drafts%20%26amp%3B%20Ballots%5C%5E%20D1.0%20Letter%20Ballot%5CP802.11aq%20-%20Draft%20D1.0a_YY.docx#section_8_4_2_173)), to the AP, in pre-association state.

An AP having dot11SolictedPADActivated equals to true shall include Service Advertisement element ([8.4.2.172](file:///C%3A%5C%23%20MY%20FOLDERS%5C---%20IEEE%20802.11aq%20Drafts%20%26amp%3B%20Ballots%5C%5E%20D1.0%20Letter%20Ballot%5CP802.11aq%20-%20Draft%20D1.0a_YY.docx#section_8_4_2_172)) in Probe Response frame, if there is one or more Service Hashes ([8.4.2.173](file:///C%3A%5C%23%20MY%20FOLDERS%5C---%20IEEE%20802.11aq%20Drafts%20%26amp%3B%20Ballots%5C%5E%20D1.0%20Letter%20Ballot%5CP802.11aq%20-%20Draft%20D1.0a_YY.docx#section_8_4_2_173)) matching with the received Probe Request containing the Service Hash element sent by the non-AP STA.

The non-AP STA may associate with the AP based on the received Service Advertisement element or may use a Service Information Request ANQP-element ([8.4.5.20](#section_8_4_4_20_Service_info_request)) to request more detailed information as defined in Table 8-257, prior to association. The receiving AP shall respond to the Service Information Request ANQP-element with a Service Information Response ANQP-element.

**10.25.3.4.4 PAD encapsulation protocol procedures**

{\*\*LB D1.0 comment: please add text to this subclause}

**10.25.3.4.5 Bloom filter Hash function operation**

The Bloom filter Hash function works as follows:

Each Service Hash is hashed to *j* bit positions in the *m*-bit Service Hint Map, using *j* hash functions. A total of 16 Hash functions are defined and are constructed as follows:

Let H(*j*,X,*m*) denote the Hash function, with:

1. *j* - Bloom filter Hash Function pre-pend parameter used in the computation.
*j* ranges from 0x00 to 0x0F, in hexadecimal notation.
2. X - is the Service Hash that is mapped to *j*-bits of the *m*-bits Service Hint Map
3. *m* - is the size of Bloom filter to be indicated in bits

The *H*(*j*,X,*m*) is computed as follows:

1. Compute A(*j*,X) = [*j* || X], where || denotes an append operation
2. Compute B(*j*,X) = CRC32(A(*j*,X)) & 0x0000FFFF. i.e., obtain the last 2 bytes 32 bit CRC of A(*j*,X), where the CRC operation is seeded with 0xFFFFFFFF.
3. H(*j*,X,*m*) = B(*j*,X) mod *m*

The full set of Hash functions is shown in Table 10-16a.

**Table 10-16a – Hash functions for the Bloom filter**

|  |  |
| --- | --- |
| Hash Function Index, *j* (in Hexadecimal)  | Hash Function  |
| 0x00 | H(0x00,X,*m*) |
| 0x01 | H(0x01,X,*m*) |
| 0x02 | H(0x02,X,*m*) |
| 0x03 | H(0x03,X,*m*) |
| 0x04 | H(0x04,X,*m*) |
| 0x05 | H(0x05,X,*m*) |
| 0x06 | H(0x06,X,*m*) |
| 0x07 | H(0x07,X,*m*) |
| 0x08 | H(0x08,X,*m*) |
| 0x09 | H(0x09,X,*m*) |
| 0x0A | H(0x0A,X,*m*) |
| 0x0B | H(0x0B,X,*m*) |
| 0x0C | H(0x0C,X,*m*) |
| 0x0D | H(0x0D,X,*m*) |
| 0x0E | H(0x0E,X,*m*) |
| 0x0F | H(0x0F,X,*m*) |

The Number of Hash functions field is used to indicate the number of hash functions, *k*, (out of the maximum of 16) used by the Bloom filter. For example, 0001 means the first 2 Hash functions are used (denoted by Hash function index 0x00 and 0x01 as shown in Table 10-16a - Hash Functions).

***<add the following new entries to existing annexes A, B and C and a new Annex Za>***

**Annex A**(informative) **Bibliography**

*<insert the following bibliography items>*

[B66] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, “Theory and Practice of Bloom Filters for Distributed Systems,” *IEEE Communications Surveys and Tutorials,* vol. 14, no. 1, pp. 131-155, Feb 2011

**Annex B**

(informative)

**Protocol Implementation Conformance Statement (PICS) - proforma**

**B.2 Abbreviations and special symbols**

* + 1. **General abbreviations for Item and Support columns**

*<Insert the following new list item, at the end of* ***B.2.2****>*

PAD pre-association discovery

* 1. PICS proforma - IEEE Std 802.11-<year>[[1]](#footnote-2)
		1. IUT configuration (Continued)

*<Insert the following entry at the end of the IUT configuration table>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Item** | **IUT configuration** | **References** | **Status** | **Support** |
| \*CFaq |  Pre-Association Discovery |  | O | Yes  No  N/A  |
|  |  |  |  |  |

*<Insert the following new subclause at the end of clause B.4>*

**B.4.27 Pre-Association Discovery Extensions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol capability | References | Status | Support |
| PAD1 | Advertisement Protocol element | 8.4.2.92 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD2 | Service Hint element | 8.4.2.171 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD3 | Service Advertisement element | 8.4.2.172 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD4 | Service Hash element | 8.4.2.173 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD5 | Supported ULP element | 8.4.2.174 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD6 | Unsolicited PAD | 10.25.3.4.2 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD7 | Solicited PAD | 10.25.3. 4.3 | CFaq:TBD | Yes □ No □ N/A □ |
| PAD8 | Encapsulation PAD | 10.25.3. 4.4 | CFaq:TBD | Yes □ No □ N/A □ |

**Annex C**

**(normative)**

**ASN.1 encoding of the MAC and PHY MIB**

**C.3 MIB Detail**

***<Insert new MIB values as follows:>***

dot11PADImplemented OBJECT-TYPE

 SYNTAX TruthValue

 MAX-ACCESS read-only

 STATUS current

 DESCRIPTION

"This is a capability variable. Its value is determined by device capabilities.

This attribute when true, indicates the STA is capable of pre-association discovery(PAD)with external networks. A STA setting this to true, implements PAD. When this is false, the STA does not implement PAD.”

 DEFVAL {false}

::= { dot11StationConfigEntry *<ANA>*}

dot11SolictedPADActivated OBJECT-TYPE

 SYNTAX TruthValue

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

 "This is a control variable.

It is written by an external management entity or the SME. Changes take effect as soon as practical in the implementation.

This attribute when true, indicates that the capability of the STA to operate Solicited PAD with external networks is enabled. The capability is disabled otherwise."

 DEFVAL {false}

::= { dot11StationConfigEntry *<ANA>*}

dot11UnsolicitedPADActivated OBJECT-TYPE

 SYNTAX TruthValue

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

 "This is a control variable.

It is written by an external management entity or the SME. Changes take effect as soon as practical in the implementation.

This attribute when true, indicates that the capability of the STA to operate Unsolicited PAD with external networks is enabled. The capability is disabled otherwise."

 DEFVAL {false}

::= { dot11StationConfigEntry *<ANA>*}

dot11EncapsulatedPADActivated OBJECT-TYPE

 SYNTAX TruthValue

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

 "This is a control variable.

It is written by an external management entity or the SME. Changes take effect as soon as practical in the implementation.

This attribute when true, indicates that the capability of the STA to operate Encapsulated PAD with external networks is enabled. The capability is disabled otherwise."

 DEFVAL {false}

::= { dot11StationConfigEntry *<ANA>*}

**Annex Za (aq)**

(normative)

**Pre-association Discovery Protocol (PAD) Additional Information**

**Za.1 High-level Functional Diagram**

The PAD protocol can help the user (STA), in pre-association state, obtain useful information from the AP about available services and access networks. This information can help the user decide whether or not to associate with the WLAN behind the AP.

Figure Za-1 describes the functional flow of MAC messaging and service information between the STA and the AP. The PAD Proxy entities, shown in Figure Za-1, are logical caches of information about network services and access network available for the STA to connect to, after it associates with the network.

****

**Figure Za-1 – PAD protocol stack functional diagram**

**Za.2 PAD Proxy Entity**

This is required to pass information up to higher layer applications and for the PAD protocol to work.

It is assumed that there is proxy function in the network that maintains a list of services. The upper layer protocols are not exposed to un-associated STAs. The proxy is used to encapsulate the service identifiers and exchange that information to the STA. Therefore, PAD is opaque to the service definition and is handled by the proxy and the end STA itself.

**Za.3 Pre-association service discovery usage scenarios**

The Pre-association service discovery (PAD) protocol supports alternative usages, depending on the deployment scenario, for obtaining service information. In the following sub-clauses, two usage scenarios are described: background search and immediate search.

**Za.3.1 Background Search**

Applications that run in the background (e.g. automatically receiving sales coupons that a user has previously signed up for) may not require immediate discovery results to be presented to the user. It may be appropriate to prevent non-AP STAs, running such background applications, from performing a Solicited PAD search. Furthermore, Solicited PAD search in a dense WLAN environment can cause network congestion. In such a scenario, it is more effective to perform Unsolicited PAD search, whereby an AP advertises multiple services it offers, while non-AP STAs need respond only if there is a matched service.

The AP may elect to advertise several typical services using Service Hash element, and advertise the remaining services using Service Hint element, in the Beacon frames[[2]](#footnote-3). Upon receiving the Beacon frame, a non-AP STA processes the Service Hash and Service Hint elements to verify if there are any potential matching services. Figures Za-2 and Figure Za-3 show two cases where there is a matching Service Hint.

If the probability of false positives in the Service Hint element is relatively high (see Figure Za-2), the non-AP STA may send a Probe Request with the Service Hash to confirm the service is indeed offered by the AP. The AP then responds with a Probe Response with Service Advertisement element that containing the corresponding Service Name. The non-AP STA may then send a PAD Service Information Request containing the Service Name and specific Service Information Query Request to obtain more information about the service from the AP. The AP responds to the PAD Service Information Request with the PAD Service Information Response containing the Service Name and specific Service Information Query Response. After the PAD Service Information Request and Response exchange, the non-AP STA should be able to make an informed decision about choosing to associate to the AP.



**Figure Za-2: Example of a message exchange for background search with high probability of false positive**

If the probability of false positive is relatively low (see Figure Za-3), the non-AP STA may directly send a PAD Service Information Request frame containing the Service Name and specific Service Information Query Request to obtain more information about the service from the AP.



**Figure Za-3: Example of a message exchange for background search with low probability of false positive**

In a scenario where there is a matching Service Hash element, the non-AP STA may directly send a PAD Service Information Request frame containing the Service Name and specific Service Information Query Request to obtain more information about the service from the AP as shown in Figure Za-4. Alternatively, the non-AP STA may choose to associate based on the matching Service Hash element.



**Figure Za-4: Example of message exchange for background search with matching Hash element**

**Za.3.2 Immediate Search**

Applications that are initiated by users (e.g. a user is looking for a fast movie download service) require immediate discovery results to be presented to the user. In this scenario, a non-AP STA should perform a Solicited PAD, whereby the non-AP STA sends Probe Request frames to query specific services immediately after user initiation of the service/application, and the AP responds with a Probe Response frame accordingly, if there is a matched service (Figure Za-5). The Probe Request frame contains the Service Hash element of the search service. The AP responds with a Probe Response frame with a Service Advertisement element containing the corresponding Service Name. The non-AP STA then may perform a PAD Service Information Request and Response exchange with the AP as shown in Figure Za-5, to obtain more information about the service.



**Figure Za-5: Example of a message exchange for immediate search**

**Za.4: Bloom Filter – definitions and application to the PAD protocol**

**Za.4.1: Determining the Bloom Filter Size, *m***

*Reference: Annex A* [B56]

A Bloom filter is a space-efficient probabilistic data structure used to test if an element (i.e. service) is a member of a set. A Bloom filter is an array of *m* bits, representing a set of *n* services S={x1, x2, …, x*n*}. These *m*-bits are initially set to all zero. A service x, is mapped to a random number uniformly between 1, …, *m* by using *k* hash functions, *hi*(*k*), for 1≤ *i* ≤ *k*.

A service y is reported as a member of S, if the bits *hi*(*y*) are set to all ones, and is guaranteed to not to be a member of S if any bit, *hi*(*y*), is set to zero.

*p* is the probability of the false positive event (lower bound)*,* which occurs when y is actually not a member of S, but reported as being in the set, is given by formula (1) and is dependent on the parameters *n*, *m* and *k.*

The variables *p*, *n*, *m* and *k* are related to each other with the following approximation, formula (1):

(1)

The optimal value of *k,* is given by:

(2)

Substituting k from (2) in (1) and reordering terms, the value of *m,* rounded to the nearest multiple of 8, is given by:

 (3)

For example, for *n*=25 services and *p*=0.01, the size of the Bloom filter *m* is 240 bits and the required number of hash function is 7.

1. *1Copyright release for PICS proforma:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS. [↑](#footnote-ref-2)
2. Alternatively, the AP may elect to advertise all of the services using either the Service Hash or Service Hint element in the Beacon frames [↑](#footnote-ref-3)