IEEE P802.11
Wireless LANs

|  |
| --- |
| Combining Service Hashes |
| Date: March 14, 2016 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Payam Torab | Broadcom Ltd. |  |  | payam.torab@broadcom.com |

 Abstract

Extend the Service Hash element (8.4.2.215) and solicited PAD procedure (10.26.3) to allow a general boolean combination of service hashes included in the Service Hash element. The current boolean function is an implicit OR (any). This submission resolves CID 2283. All changes are relative to Draft 3.1.

**Revision History**

R0: Initial revision

R1: Replace SOPP form with general SOP form

R2: Corrected dates, CID.

|  |  |  |
| --- | --- | --- |
| **CID 2283** | **Category** | **Subclause** |
| "The Service Hash element definition is not flexible enough to allow an arbitrary boolean combination of services (service hash values). The implicit relationship assumed is a Boolean OR (ANY) when seeking services, and AND (ALL) when indicating services. For example, a STA that includes two services S1 and S2 in the Service Hash element is interpreted as seeking S1 OR S2, prompting a response by any AP or PCP that provides S1 alone, S2 alone, or both S1 and S2.This is a common use case but does not cover all applications. A STA may be exclusively interested in APs that provide BOTH S1 and S2. This becomes especially important with carrier Wi-Fi where a specific combination of capabilities such as carrier, billing, and performance metrics may be of interest. There are applications in consumer electronics as well, for example, connecting to a wireless docking station that offers a set of services such as a mouse AND a keyboard AND a printer.The Boolean function implicitly assumed in the Service Hash element needs to be extended from the implicit OR function (S1 + S2+ ... + SN) (when searching) and AND function (S1 . S2 . ... . SN) (when advertising) to a a more general form such as canonical sum-of-products (SoP) or product-of-sums (PoS) to allow an arbitrary combination of services.The benefit of providing more context and more processing is fewer message exchanges (probes for example) over the air when certain combination of services is of interest." | Technical | 8.4.2.215 |

**Background**

The Service Hash element (8.4.2.215) carries an array of 6-octet Service Hash values (also referred to as Service Hash, Hash, Hash Value, with upper case or with lower case in discussion), with each value representing a service of interest. When seeking services, the implicit function combining the service values in the Service Hash element is the logical OR (ANY) function. On the advertising side, when indicating available services, the implicit function combining the subfields of the Basic Service Information Descriptors field of the Service Advertisement element (8.4.2.214) is the logical AND (ALL) function.

A searching STA that includes two services S1 and S2 in the Service Hash element is interpreted as seeking S1 OR S2, prompting a response by any service provider that provides S1 alone, S2 alone, or both S1 and S2 – even if the STA is interested in both services at the same time. The ambiguous request semantics can trigger responses by many APs that provide at least one of the listed services in probe. While the correct service provider can ultimately be determined at the service layer, there are unnecessary (and costly in crowded spaces) transmissions (mostly Probe Response frames) that could be avoided by simply better request semantics.

A similar problem exists for advertising STAs: An advertising STA may have resource or policy conflicts that prevent providing certain combinations of services, e.g., a wireless docking station with a single physical USB port may be able to provide a variety of services off of its USB port, but probably not all of them at the same time. There is currently no way to formulate such restrictions on the advertising side.

It is possible to extend the implicit functions assumed in these scenarios from the logical OR (in searching) and AND (in advertising) functions to more general boolean functions that would result in fewer transmissions over the air and fewer connection attempts.

This contribution extends the searching semantics. Extending the advertising semantics is for further study.

**Encoding**

All boolean function of N boolean variables can be expressed in the canonical sum-of-products (SOP) (or product-of-sums (POS)) form. Expressing an arbitrary boolean function of N boolean variables as a sum of up to 2N minterms with SOP form (or product of 2N maxterms with POS form) requires 2N bits or ⎡2N/8⎤ octets.

We observe that service discovery applications can be mostly addressed by a class of boolean functions known as positive boolean functions (a boolean function f of N boolean variables is positive if it is increasing, i.e., X ≤ Y ⇒ f(X) ≤ f(Y) for any X, Y belonging to {0, 1}N, where ≤ is interpreted as ordered comparison). For example, it makes little sense for a service seeker to include services that it is *not* seeking, and makes little sense for a service provider to include services that it is *not* providing[[1]](#footnote-1).

Compact representation of positive boolean functions has seen some studied (mostly in the database field); previous revision of this contribution described a sum-of-positive-products (SOPP) format, which although compact in nature (O(N2) vs. O(2N) for SOP) cannot represent all positive boolean functions. A representation using a combination of ANYM boolean functions is promising (ANYM(X1, X2, ..., XN) is 1 if at least M of its N boolean arguments are equal to 1 and 0 otherwise, e.g., ANY1(S1, S2) = S1 + S2, ANY2(S1, S2) = S1. S2, and ANY2(S1, S2, S3) = S1S2 + S1S3 + S2S3) but has received little study.

We have chosen the SOP format because of its generality and (semantics) supports for applications involving boolean functions that are not positive. SOP form size remains reasonable for up to 6-8 service hash values. Table 1 lists the SOP representation overhead relative to the service hash size.

Table 1: SOP form overhead relative to service has values

|  |  |  |
| --- | --- | --- |
| Number of ServicesN | SOP form(⎡2N/8⎤ octets) | Relative size compared to service hash values (6N octets) |
| 1 | 0 | 0 |
| 2 | 1 | 8% |
| 3 | 1 | 6% |
| 4 | 2 | 8% |
| 5 | 4 | 13% |
| 6 | 8 | 22% |
| 7 | 16 | 38% |
| 8 | 32 | 67% |

Finally, special case of ANY M OUT of N (which covers both ANY and ALL cases) is represented independently for efficiency.

**8.4.2.215 Service Hash element**

The Service Hash element contains one or more service hashes and a logical function to interpret the combination of service hashes. The format of the Service Hash element is shown in Figure 8-577co (Service Hash element format).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Element ID | Length | Element ID Extension | Flags | Service Hashes | Service Combination |
| Octets: | 1 | 1 | 1 | 2 | variable | variable |

**Figure 8-577co—Service Hash element format**

The Element ID, Length and Element ID extension fields are defined in 9.4.2.1 (General).

The Flags field is defined in Figure 8-xxx (Flags field format).

|  |  |  |  |
| --- | --- | --- | --- |
|  | B0 B5 | B6 B11 | B12 B15 |
|  | Number of Included Services | Number ofRequested Services | Reserved |
| Bits: | 6 | 6 | 4 |

**Figure 8-xxx—Flags field format**

The Number of Included Services field indicates the number of service hashes that are included in the element. This field is set to a nonzero value.

A value of *r* for the Number of Requested Services field indicates search for STAs that provide at least *r* services among those specified by the service hashes included in the element. Any value of the Number of Requested Services field that is equal to or greater than the value of the Number of Included Services field indicates search for STAs that provide all the services included in the element. The field is set to 0 to indicate search for STAs that provide a generally more complex combination of services included in the element, where the combination is specified by the Service Combination field.

The Service Hashes field contains one or more 6-octet service hash values. See 10.26.6 (Service hash procedures) for procedures for generating a service hash used in the Service Hash element.

The Service Combination field is present only if the Number of Requested Services field is set to 0. If present, denoting the number of service hashes in the element by *n*, the Service Combination field carries a service combination bitmap that is 2*n* bits in length and is organized into ⎡2*n*/8⎤ octets such that bit number *b* (0 ≤ *b* < 2*n*) in the bitmap corresponds to bit number (*b* mod 8) in octet number ⎣*b*/8⎦, where the low order bit of each octet is bit number 0, and the high order bit is bit number 7. The service combination bitmap is the sum-of-products representation of a boolean function of *n* boolean variables x1,....,*xn* where *xi* (*i* = 1,...,*n*) indicates search for the service corresponding to the *i*-th service hash included in the element. Specifically, bit *b* (0 ≤ *b* < 2*n*) in the bitmap corresponds to minterm *mb* in a sum-of-products representation.

NOTE—To illustrate the Service Combination field format consider a search for STAs that provide service S1 or service S2 or both services S3 and S4, where services S1, S2, S3 and S4 appear in the Service Hash element in that order. The service combination of interest can be represented by the boolean function x1 + x2 + x3.x4, or the sum of minterms m1, m2, m3, m5, m6, m7, m9, m10, m11, m12, m13, m14, and m15 using the sum-of-product representation. The resulting bitmap is 1111111011101110 binary, and the value of the Service Combination field is 0xFEEE.

*[Editor Note: Separate comments are probably needed -- All instances of “non-AP STA” in the draft need to be replaced with “non-AP and non-PCP STA”. Also, solicited PAD procedure should be equally applicable to DMG active scan where a DMG Beacon frame (with Discovery Mode field set to 1) is typically used instead of the Probe Request frame.]*

**10.26.3 Solicited PAD procedure**

When dot11SolicitedPADActivated is true, a non-AP and non-PCP STA transmits to an AP or PCP a Probe Request frame with a Service Hash element. This element includes one or more service hashes generated from the service name(s) of the service(s) that the non-AP and non-PCP STA is requesting, as well as valid combinations of services being requested.

When dot11SolicitedPADActivated is true, an AP or PCP shall determine if it can provide the requested combination of services in the received Probe Request frame. Determination is based on the service hash values in the Service Hash field of the Service Hash element, and valid service combinations specified through the Flags and Service Combination fields of the Service Hash element. If the AP or PCP determines that it can provide the requested combination of services, it shall respond with a Probe Response frame with a Service Advertisement element that contains a Basic Service Information Descriptor field for a set of services that satisfy the request.

NOTE—For example, an AP or PCP that receives a Probe Request frame with a Service Hash element that includes hash values for 4 services S1, S2, S3 and S4 (in that order) and a value of 0xFEEE in its Service Combination field responds to the request if and only if it can provide service S1 or service S2 or both services S3 and S4. The Service Advertisement element returned in the Probe Response frame can contain a Basic Service Information Descriptor field for any set of available services that satisfy the request, e.g., S1, S3 and S4.

The requesting non-AP and non-PCP STA shall process the Service Advertisement element in the received Probe Response frame to select a service combination that satisfies the non-AP and non-PCP STA request, and also to obtain the corresponding instance names. If there is a matching service name, the non-AP and non-PCP STA may decide to proceed with the ANQP-SD procedure (10.26.4 (ANQP-SD procedure) or authentication and association procedure (10.3 (STA authentication and association)) based on the nature of the service (see examples illustrated in Annex AA.1 (Pre-association discovery usage scenarios)), the details of which are out of the scope of this standard.

1. A service seeker (advertiser) can be envisioned to include a service to indicate that it will never use (provide) that service in order to assist the client or server selection. [↑](#footnote-ref-1)