IEEE P802.11
Wireless LANs

|  |
| --- |
| NAI-Realms Hint IE inclusion in Beacons for Fast Roaming |
| Date: 2025-03-18 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Azin Neishaboori | General Motors | 29427 Louis Chevrolet Road, Warren MI 48093 |  | Azin.neishaboori@gm.com |
|  |  |  |  |  |

Abstract

This submission proposes text updates to mitigate the latency incurred by ANQP query/response message exchanges for high-mobility non-AP STAs during the roaming process in order to identify at least one AP STA that the non-AP STA can successfully authenticate with, based on probabilistic information about the supported NAI realms by an AP obtained via a new (proposed) optional element included in the AP’s beacon/probe response as well as optionally in the neighbor report elements included in the AP’s beacon/probe response and neighbor report response frames indicating such information about the neighboring APs when known.

***TGmf editor: Please note Baseline is 11REVme D7.0***

**Revision Notes:**

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGm Draft. This introduction is not part of the adopted material.

***TGmf editor: Editing instructions preceded by “Instruction to Editor” are instructions to the TGmf editor to modify existing material in the TGm draft. As a result of adopting the changes, the TGmf editor will execute the instructions rather than copy them to the TGm Draft.***

**Discussion:**

High mobility non-AP STAs such as vehicles require very low AP handover latency for scenarios where sustanined network connectivity is needed. This requires the time needed for identification of a target AP that the STA can successfully authenticate with to be minimized.

Currently, for a STA to determine whether it can successfully authenticate with an interworking-capable AP which supports one or more SSPs requires at least one ANQP query and response pair: The STA queries the list of NAI realms from an AP, and the AP responds with a frame including this list. Subsequently, the STA searches within the list of NAI realms the AP supports and determines whether it has credentials for at least one of those realms. If not, the AP will not be a viable candidate target AP for the STA to roam to. When there are several APs within STA’s range (but all are outside the current AP’s ESS, and thus the STA needs to perform full authentication with), this would necessitate for the STA to perform an ANQP request/response exchange with each of those APs. Accordingly, a few rounds of trial and error may be needed before the STA could identify a down-selected subset of those APs which are viable candidate target APs for roaming, i.e., that the STA can successfully authenticate and establish a secure link with. This could result in an excessive roaming latency, and potentially traffic disruption if the high-mobility STA does not roam to a new (target) AP before it falls out of range of its current AP.

For APs catering to high-mobility STAs, we propose including as as an IE in beacon and probe response an optional space-efficient hash of all NAI realms supported by an AP using Bloom filter. We also propose optionally including this construct in neighbor report elements transmitted by such APs in their beacons and probe responses, as well as in neihbor report response frames to enable a receiving STA to easier identify which of the APs in its surrounding it might pontentially be able to successfully authenticate with.

Once the STA receives such a beacon or probe response from a potential target AP, it looks into this *NAI Realms Hint IE*, identifies the hash functions used by the Bloom filter utilized in the IE, applies the same set of hash functions on each of the NAI realms it has credentials for, and checks whether the respective bits are set in the Bloom filter bit array included in the respective field in the IE for each such realm.

Because a small chance of false positives remains when using Bloom filter as described above, once the STA determines a “hit” using Bloom filter, ANQP query/response is still needed to acertain that a potentially supported NAI realm is indeed supported by the potential target AP. The overall gain however is that the STA will only send an ANQP query to a nearby AP that it is almost certain it can roam to successfully.

NOTE - In case there is more than one candidate AP which potentially supports an NAI realm the STA has credentials for, the STA may concurrently query all the candidate APs with a single group-addressed GAS request frame described in 9.6.7.12 according to the procedure defined in 11.22.3.2. Alternatively, the STA may concurrently query all the APs in the candidate list according to the Query AP List procedure described in 11.22.3.3.14, using an ANQP query constructed according to 9.4.5.24. In case the STA receives indication that there is more than one NAI realm supported by one or more APs in range, the selection of a single NAI realm is outside the scope of this standard.



**Summary of changes:**

* Added a new NAI Realms Hint element to BSSDescription table in clause 6.5.3.3.2
* Added NAIrealmsHint to parameters included in MLME-START.request(…) and a row to the corresponding table in subclause 6.5.11.2.2
* Added NAIrealmsHint to parameters included in MLME-START.request(…) and a row to the corresponding table in subclause 6.5.24.2.2
* Updated table 9-62 in subclause 9.3.3.2 to include NAI Realms Hint element in Beacon frame body
* Updated table 9-69 in subclause 9.3.3.10 to include NAI Realms Hint element in Probe Response frame body
* Update table 9-130 in subclause 9.4.2.1 to include NAI Realms Hint element
* Added a new subclause 9.4.2.x to define NAI Realms Hint element
* Modified subclause 9.4.2.35 to include NAI Realms Hint in the neighbor report element as an optional subelement
* Added a new subclause 11.22.10 to describe modified operation of Interworking procedures when adverstisement of supported NAI Realms Hint element is added to beacon and probe response frames
* Updated subclause 11.23.5 to extend Bloom filter hash function operation to also encompass cases when NAI Realms Hint element is present

**6.5.3.3.2** **Semantics of the service primitive**

**TGmf editor: Please update the BSSDecription table with an added row for NAI Realms Hint element:**

Valid Range

Type

Description

IBSS adoption

Name

NAIRealmsHint

Do not adopt

Provides an indication of the NAI realms (as defined in 9.4.5.10) advertised in Beacon frames and Probe Response frames. The values from the NAI Realms Hint element, if such an element was present in the Beacon or Probe Response frame; else null. d Probe Response frames. The values from the NAI Realms Hint element, if such an element was present in the Beacon or Probe Response frame; else null

As defined in 9.4.2.x (NAI Relams Hint element)

NAI Realms Hint element

**6.5.11.2.2 Semantics of the service primitive**

***TGmf editor: Please update this subclause as below:***

***Plesse add NAIrealmsHint to parameters included in MLME-START.request (…)***

***Please add to the corresponding table a row for NAIRealmsHint:***

Type

Description

Valid Range

Name

NAI Realms Hint element

NAIRealmsHint

This element may be advertised in Beacon and Probe Response frames. The element is optionally present if dot11InterworkingService is true and absent otherwise.

As defined in 9.4.2.x (NAI Relams Hint element)

**6.5.24.2.2 Semantics of the service primitive**

***TGmf editor: Please update the parameter list included in MLME-Update.request(…), with adding a new parameter NAIRealmsHint, and accordingly, the table that follows with a corresponding row:***

Valid Range

Description

Name

Type

As defined in 9.4.2.x (NAI Relams Hint element)

If null, requests that NAI Realms Hint element be removed from Beacon and Probe Response frames. Otherwise, provides an indication of the NAI Realms advertised in Beacon and Probe Response frames. The parameter is optionally present if dot11InterworkingServiceActivated.

NAIRealmsHint

NAI Realms Hint element

**9.3.3.2 Beacon frame format**

***TGmf editor: Please update Table 9-62—Beacon frame body by adding a new a row for NAI Realms Hint element with the values of the columns specified as follows:***

Order

Information

Notes

NAI Realms Hint

The NAI Realms element is optionally present if dot11InterworkingServiceActivated is true.

xx

**9.3.3.10 Probe Response frame format**

***TGmf editor: Please update Table 9-69—Beacon frame body by adding a new row for NAI Realms Hint element with the values of the columns specified as follows:***

Order

Information

Notes

NAI Realms Hint

The NAI Realms element is optionally present if dot11InterworkingServiceActivated is true.

xx

**9.4.2.1 General**

***TGmf editor: Please update Table 9-130—Element IDs by adding a new row for NAI Realms Hint element with the values of the columns specified as follows:***

Fragmentable

Element

Element ID

Extensible

Element ID Extension

No

No

255

NAI Realms Hint (see 9.4.2.x (NAI Realms Hint element))

<ANA>

***TGmf editor: Please add, as below, subclause 9.4.2.x NAI Realms Hint element***

**9.4.2.x NAI Realms Hint element**

The NAI Realms Hint element provides a probabilistic representation of a set of NAI realms that are supported by the BSS. The format of the NAI Realms Hint element is shown in Figure 9-xxx (NAI Realms Hint element format).

Bloom Filter Bit Array

Bloom Filter Information

Element ID Extension

Length

Element ID

Octets:

Figure 9-xxxx----NAI Realms Hint element format

variable

1

1

1

1

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

The Bloom Filter Information field represents the stochastic characteristics of a Bloom filter (Tarkoma et al.

[B68]) that conveys the probabilistic data. The format of the Bloom Filter Information field is shown in

Figure 9-xxx (Bloom Filter Information field format).

Number Of Hash Functions

False Positive Probability Range

Bits

4

4

Figure 9-xxxx----Bloom Filter Information field format

The False Positive Probability Range subfield represents the false positive probability range of the Bloom

filter. The False Positive Probability Range subfield is shown in Table 9-xxx (False Positive Probability

Range subfield values).

***TGmf editor: Please repeat Table 9-372 in this subclause after the text above:***

Table 9-xxx--- False Positive Probability Range subfield values

|  |  |
| --- | --- |
| **Value** | **False positive probability range, p** |
| 0 | $$p>0.25$$ |
| 1 | $$20\%\leq p\leq 25\%$$ |
| 2 | $$15\%\leq p\leq 20\%$$ |
| 3 | $$10\%\leq p\leq 15\%$$ |
| 4 | $$5\%\leq p\leq 10\%$$ |
| 5 | $$1\%\leq p\leq 5\%$$ |
| 6 | $$0.5\%\leq p\leq 1\%$$ |
| 7 | $$0.1\%\leq p\leq 0.5\%$$ |
| 8 | $$0.05\%\leq p\leq 0.1\%$$ |
| 9 | $$0.01\%\leq p\leq 0.05\%$$ |
| 10 | $$p\leq 0.25$$ |
| 11-15 | Reserved |

The Number Of Hash Functions subfield is set to k-1, where k is the number of hash

functions used by the Bloom filter as described in 11.x.x (Bloom filter hash function operation).

The Bloom Filter Bit Array field provides an indication of the NAI realms offered by or through the AP with a target probability of false positive p. The length of the Bloom Filter Bit Array field in octets is $\left⌈m/\right.\left.8\right⌉$

where m is the size of the Bloom filter in bits. How the size of the Bloom filter is determined is out

of the scope of this standard. The maximum length of the Bloom Filter Bit Array field is 128 octets.

**9.4.2.35 Neighbor Report Element**

***TGmf editor: Please modify subclause 9.4.2.35 by adding an optional subelement to Table 9-212 as shown below and adding the following corresponding text:***

Subelement

Name

Extensible

NAI Realms Hint

Yes

222

Reserved

223-225

The NAI Realms Hint subelement has the same format as defined in the NAI Realms Hint element (See 9.4.2.x).

***TGmf editor: Please add, as below, subclause 11.22.10 NAI Realms Hint element:***

**11.22.10 Interworking procedures: adverstisement of supported NAI realms hint**

**11.22.10.1 General**

APs can assist non-AP STAs performing network discovery and selection through the advertisement of a

NAI Realms Hint element. The NAI Realms Hint element contains information hints to NAI realms that identify SSPs or other entities whose networks or services are accessible via the advertising AP.

The NAI Realms Hint element provides a non-AP STA with a bit array constructed based on supported NAI realms of a STA if the STA includes this element in its beacons and probe responses.

NAI Realms Hint element is defined in 9.4.2.x and includes a bit array corresponding to a Bloom filter Information field and a Bloom Filter bit array, representing respectively the stochastic characteritstics of a Bloom filter that conveys the probabilistic data, and an indication of the NAI realms supported by or through the AP with a target probability of false positive $p$.

NOTE- This probabilistic information about the supported NAI realms by an AP can be obtained by the non-AP STA using beacons and probe responses, and prior to receiving information about one or more supported NAI realms of a STA via the use of ANQP request and response as indicated in Table 11-20, and Table 9-426 (ANQP-element definitions).

If a non-AP STA receives an NAI Realms Hint element from an AP, and determines, using the Bloom filter bit array therein that at least one of the NAI realms it has security credentials for is probably supported by the AP, it concludes that with probalility $p$ as indicated in the Bloom Filter Information field as defined in 9.4.2.x, the AP has at least one potentially matching NAI realm. The STA then proceeds to use ANQP to obtain information on one or more of the potentially matching NAI realms from the AP. The STA uses the obtained information to determine the specific security credentials corresponding to one of the NAI realms supported by the AP for which the STA has credentials for, and uses those credentials to authenticate with the AP.

A STA that has dot11NAIReamsl Hint equal to true indicating its support for NAI Realms Hint, and the inclusion of an NAI Realms Hint element in its beacons and probe responses shall have dot11InterworkingServiceActivated equal to true,.

**11.22.10.2 NAI Realm Hash**

An NAI realm hash is generated from an NAI realm. An NAI realm is defined as a subfield within the NAI Realm Tuples field (shown in Figure 9-1091) of the NAI Realm ANQP-element in 9.4.5.10, and based on the NAI Realm Encoding Type, may be formatted in accordance with IETF RFC 4282 and RFC 7542 or is as a UTF-8 string that might not be formatted in accordance with IETF RFC 4282. The NAI Realm subfield may contain one or more semicolon-separated NAI realms.

If an NAI realm hash is used as input into the Bloom filter as described below, the NAI Realm Encoding Type shall be set to 1, i.e., each NAI realm included in the NAI Realm subfield shall be encoded as a UTF-8 string.

An NAI Realm hash used to map into the Bloom Filter Bit Array, utilizes 6 octets, and is generated as follows:

NAI Realm hash = (#3506)ExtractBits(SHA-256(NAI Realm name), 0, 48)

The Bloom filter hash function operation is as defined in 11.23.5.

**11.23.5 Bloom filter hash function operation**

**TGmf editor: Please update the following paragraphs in this subclause, as follows:**

Let m denote the number of bits in the Bloom filter, and let k-1 be the setting of the Number Of Hash Functions

subfield in the Bloom Filter information field (see 9.4.2.236 (Service Hint element) and 9.4.2.xxx (NAI Realms Hint element)), i.e., k is the number of Bloom filter hash functions (out of a maximum of 16) used by the Bloom filter…

…

Create the Bloom filter as follows:

a) Set all bits in the Bloom filter to 0.

b) For each service hash in the set of service hashes or NAI realm in the set of NAI realm hashes, compute the k bit positions by setting (#515)j = 0,

…, k-1, in the function H(j,X,m) shown below.

…

Let H(j,X,m) denote the Bloom filter hash function,

Where j is the Bloom filter hash function prepend parameter used in the computation; j is a single octet and

ranges from 0x00 to 0x0F

X is the service hash that is mapped into the Bloom Filter Bit Array field (see 11.23.4 (Service hash

procedure) for procedures for generating a service hash and 11.22.10.2 (NAI Realm Hash) for procedures for generating an NAI realms hash used to map into the Bloom Filter Bit Array

field), m is the size, in number of bits, of the Bloom filter; how the size of the Bloom filter is determined is

out of the scope of this standard.