A Method to Optimize SCN messages

IEEE 802.16 Presentation Submission Template (Rev. 9)

Document Number:

IEEE S802.16j-08/067

Date Submitted:

2008-01-14

Source:

David Comstock, Junxian Mo E-mail: dcomstock@huawei.com

Huawei Technologies

*<http://standards.ieee.org/faqs/affiliationFAQ.html>

Venue:

LB#26/IEEE P802.16Rev2/D2

Base Contribution:

None

Purpose:

For information pertaining to Comment #1 in LB26a_Comstock_David commentary file

Notice:

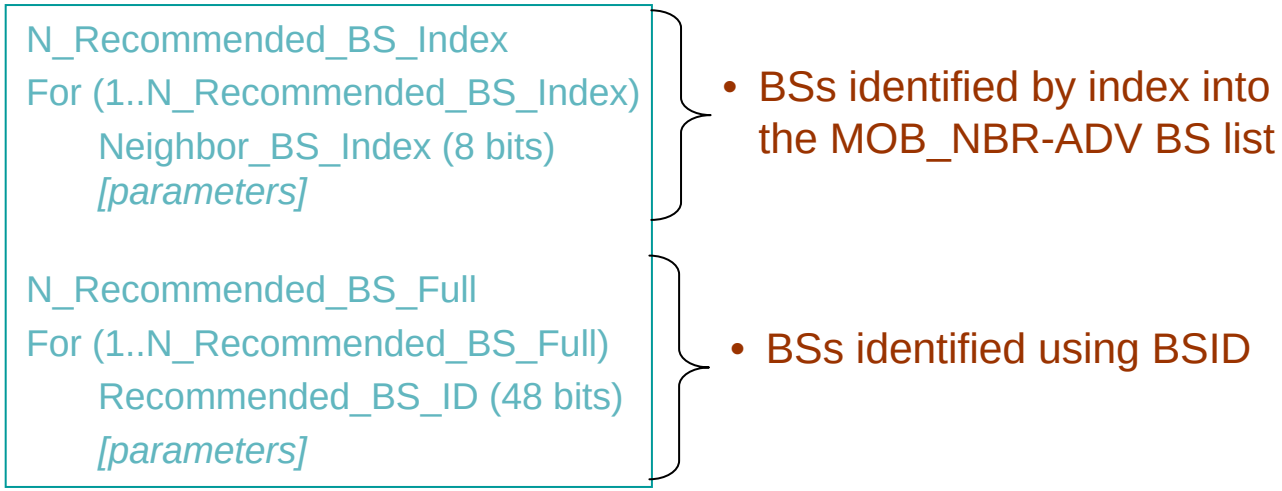
Release:

Patent Policy:

Background

In MOB_SCN messages, BSs may be identified in 2 ways

1. If the BS was included in a MOB_NBR-ADV message
   - Use an 8-bit index into the MOB_NBR-ADV's BS list
   - Avoids the use of the 48-bit BSID
2. Otherwise, the full BSID is used



MOB_NBR-ADV bitmap index

Issue

- For MOB_SCN-REQ/RSP/REP
- The 8-bit MOB_NBR-ADV index
  - Supports 256 BSs in the MOB_SCN messages
  - In most cases, this is many more than needed

Proposal

- Add ***Nbr_Bitmap_Index*** associated with MOB_NBR-ADV
  - May be used instead of the 8-bit index
  - Size (number of bits) is up to the number of BSs included in the MOB_NBR-ADV message
  - A ‘1’ in the bitmap identifies a BS’s index in the MOB_NBR-ADV’s BS list

MOB_NBR-ADV bitmap index

- Either the bitmap index or the existing index may be used

Use_Nbr_Bitmap_Index
If (Use_Nbr_Bitmap_Index)
    Nbr_Bitmap_Index
    For (each '1' in Nbr_Bitmap_Index)
        [parameters]
else
    N_Recommended_BS_Index
    For (1..N_Recommended_BS_Index)
        Neighbor_BS_Index (8 bits)
        [parameters]

- BSs identified by a bitmap index into the MOB_NBR-ADV BS list
- BSs identified by existing index into the MOB_NBR-ADV BS list

N_Recommended_BS_Full
For (1..N_Recommended_BS_Full)
    Recommended_BS_ID (48 bits)
    [parameters]

- BSs identified using BSID

MOB_SCN-REQ bitmap index for MOB_SCN-RSP

Issue

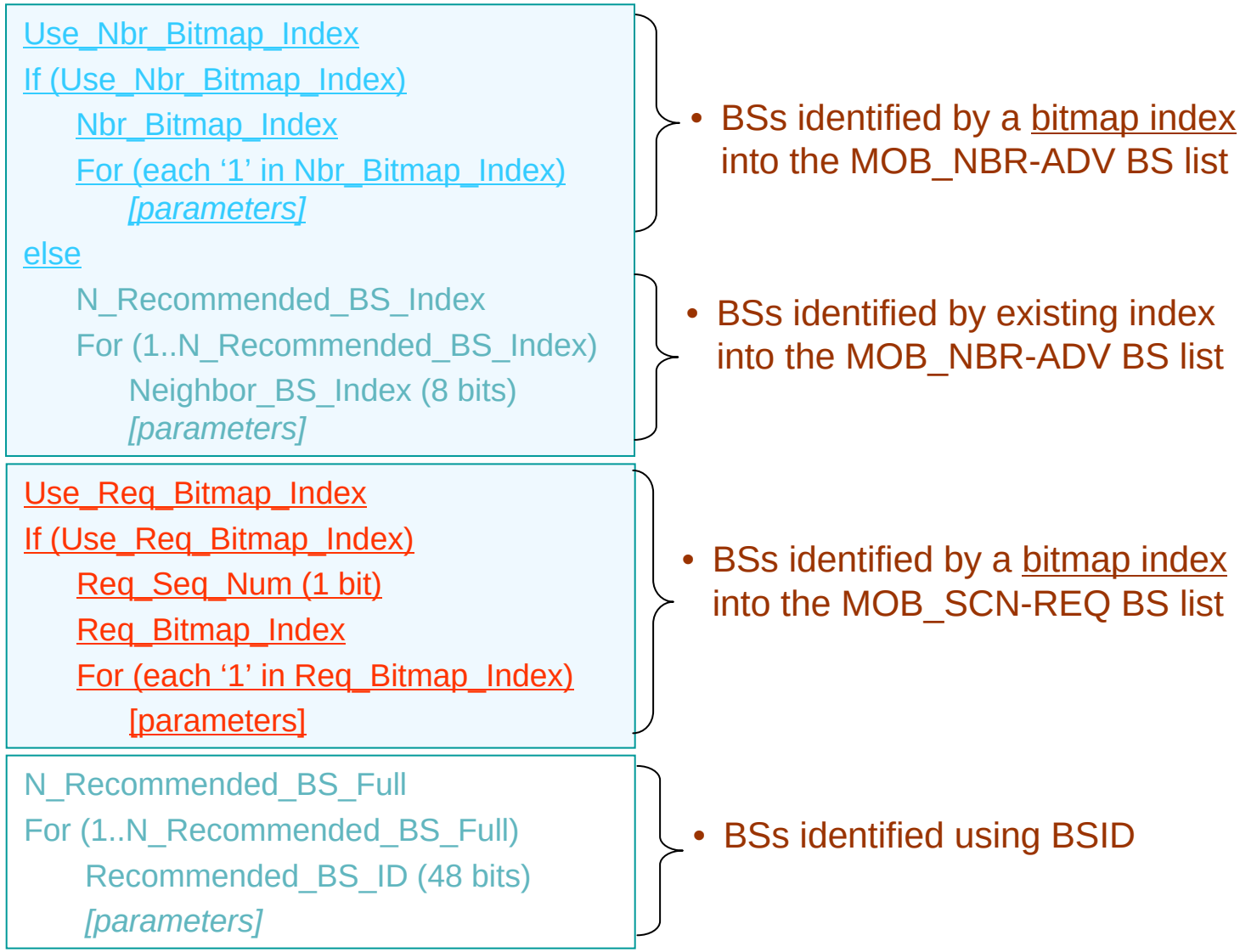
- For MOB_SCN-RSP
  - A MOB_SCN-REQ message may include BSs that are not in the associated MOB_NBR-ADV message
  - BSID for these BSs must be used in the MOB_SCN-RSP message, which requires 48 bits apiece

---

Proposal

- Add ***Req_Bitmap_Index*** associated with MOB_SCN-REQ
  - Size (number of bits) is up to the number of BSs included in the MOB_SCN-REQ message
  - A '1' in the bitmap identifies a BS's index in the MOB_SCN-REQ's BS list
- Add ***1-bit sequence number*** to identify the associated MOB_SCN-REQ to resolve the following error scenario:
  - MS retransmits a MOB_SCN-REQ maximum number of times
  - MS transmits a new MOB_SCN-REQ but then receives MOB_SCN-RSP based on the first MOB_SCN-REQ

MOB_SCN-REQ bitmap index for MOB_SCN-RSP



MOB_SCN-RSP bitmap index for MOB_SCN-REP

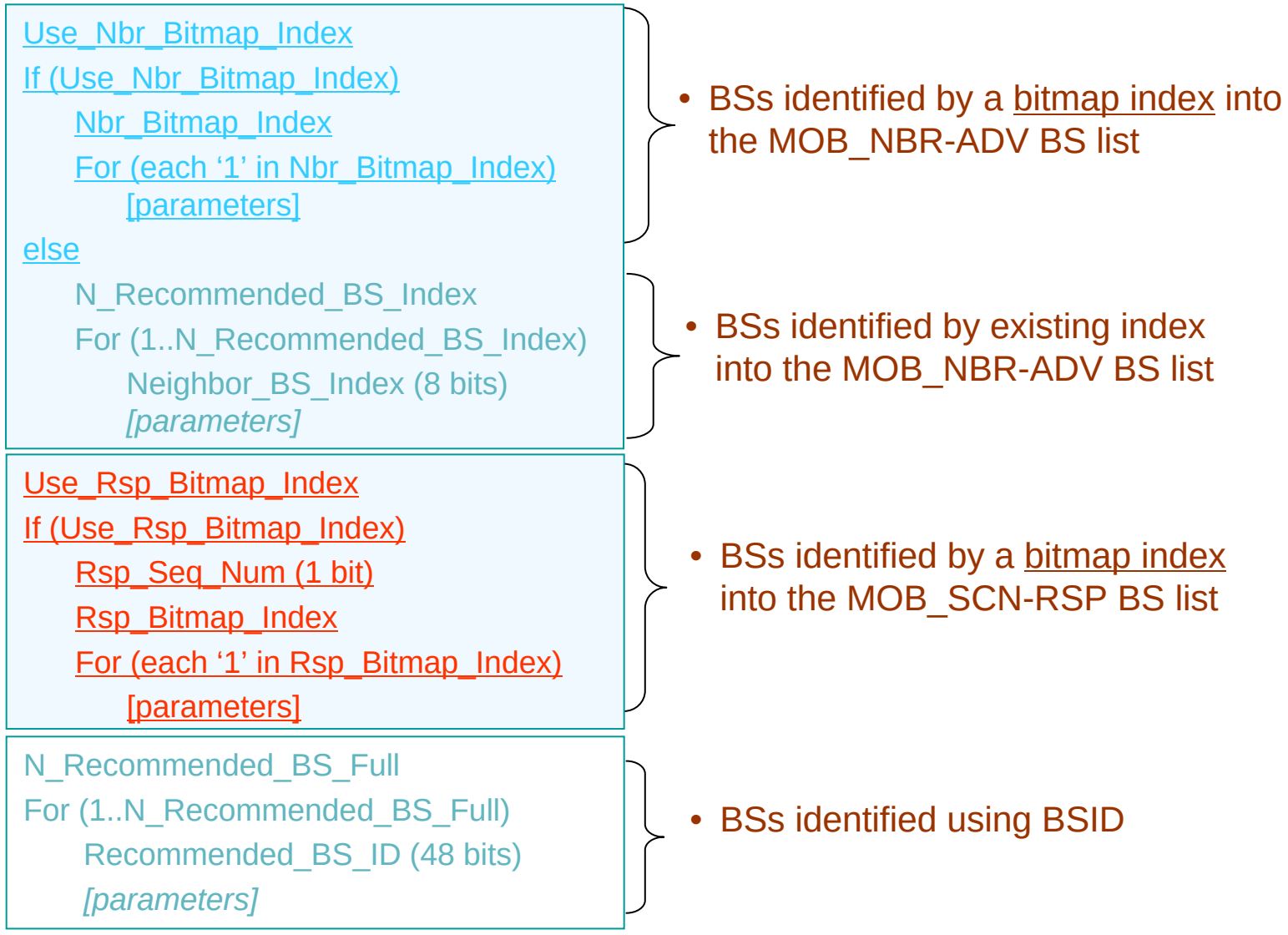
Issue

- For MOB_SCN-REP
  - A MOB_SCN-RSP message may include BSs that are not in the associated MOB_NBR-ADV or MOB_SCN-REQ
  - BSID for these BSs must be used in the MOB_SCN-REP message, which requires 48 bits apiece

Proposal

- Add ***Rsp_Bitmap_Index*** associated with MOB_SCN-RSP
  - Size (number of bits) is up to the number of BSs included in the MOB_SCN-RSP message
  - A '1' in the bitmap identifies a BS's index in the MOB_SCN-RSP's BS list
- Add ***1-bit sequence number*** to identify the associated MOB_SCN-RSP in to resolve the following error scenario:
  - BS retransmits a MOB_SCN-RSP maximum number of times
  - BS transmits a new MOB_SCN-RSP but then receives MOB_SCN-REP based on the first MOB_SCN-RSP

MOB_SCN-RSP bitmap index for MOB_SCN-REP



MOB_SCN-REQ message

Syntax	Size (bit)
MOB_SCN-REQ_Message_format() {	—
Management Message Type = 54	8
Scan duration	
Interleaving interval	
Scan Iteration	
N_Recommended_BS_Index	
If(N_Recommended_BS_Index != 0){	
Configuration change count for MOB_NBR-ADV	
}	
For(j = 0; j < N_Recommended_BS_Index; j++){	
Neighbor_BS_Index	8
Reserved	1
Scanning type	3
}	
N_Recommended_BS_Full	8
For(j = 0; j < N_Recommended_BS_Full; j++){	—
Recommended BS ID	48
Reserved	1
Scanning type	3
}	
Padding	variable
TLV encoded information	variable
}	—

Recommended BSs included in MOB_NBR-ADV message can be referenced according to their position in the MOB_NBR-ADV message

Number of recommended BSs that were included in MOB_NBR-ADV message

Position of this BS in MOB_NBR-ADV based on index

Recommended BSs not included in MOB_NBR-ADV message

Full BS ID used to identify BS

Yellow: MOB_NBR-ADV part
Blue: Full BS ID part

MOB_SCN-RSP message

Syntax	
MOB_SCN-RSP_Message_format() {	
Management Message Type = 55	
Scan duration	
Report mode	
Reserved	
Report period	
Report metric	
if (Scan Duration != 0) {	
Start frame	
Interleaving interval	
Scan iteration	
N_Recommended_BS_Index	N_Recommended_BS_Full
If(N_Recommended_BS_Index != 0){	
Configuration change count for MOB_NBR-ADV	
}	
For(j = 0; j < N_Recommended_BS_Index; j++){	For(j = 0; j < N_Recommended_BS_Full; j++){
Neighbor_BS_Index	Recommended BS ID
Reserved	Reserved
Scanning type	Scanning type
If (Scanning type == 0b010) OR (Scanning type == 0b011) {	If (Scanning type == 0b010) OR (Scanning type == 0b011) {
Rendezvous time	Rendezvous time
CDMA_code	CDMA code
Transmission_opportunity offset	Transmission_opportunity offset
}	}
}	}
Padding	
}	
TLV encoded information	
}	

Yellow: MOB_NBR-ADV part

Blue: Full BS ID part

Syntax	Size (bit)
MOB_SCN-REQ_Message_format() {	
Management Message Type = 54	
Scan duration	8
Interleaving interval	8
Scan Iteration	
N_Recommended_BS_Index	
If(N_Recommended_BS_Index != 0){	
Configuration change count for MOB_NBR-ADV	
}	
If(N_Recommended_BS_Index == 0xFF){	
Req_Seq_Num	1
Nbr_Bitmap_Index	Up to the Number BSs in MOB_NBR-ADV
For(each '1' in Nbr_Bitmap_Index)	—
Scanning type	3
}	—
} else {	—
For(j = 0; j < N_Recommended_BS_Index; j++){	—
Neighbor_BS_Index	8
Req_Seq_Num [delete Reserved]	1
Scanning type	3
}	—
}	—
N_Recommended_BS_Full	8
[…]	

- Backward compatible
- 1.x MS must work with 16e BS

- Assume won't have 256 BSs in MOB_ NBR-ADV,
- 1.x BS will use 0xFF value to signal to 1.x MS that bitmap is used.
- 16e BS will use old format

Proposed MOB_SCN-REQ

MOB_SCN-REQ message solution

FULL BS ID Part：

N_Recommended_BS_Full	8
For(j = 0; j < N_Recommended_BS_Full; j++){	—
Recommended BS ID	48
Req_Seq_Num [delete Reserved]	1
Scanning type	3
}	
Padding	variable
TLV encoded information	variable
}	—

Proposed MOB_SCN-REQ

Proposed

MOB_SCN-RSP message solution

```
MOB_SCN-RSP_Message_format() {
    Management Message Type = 55
    Scan duration
    Report mode
    Reserved 3 (delete 6)
    Rsp_Seq_Num 1
    Use_Nbr_Bitmap_Index 1
    Use_Req_Bitmap_Index 1
    Report period
    Report metric
    if (Scan Duration != 0) {
        Start frame
        Interleaving interval
        Scan iteration
        If(Use_Nbr_Bitmap_Index == 1){
            Configuration change count for MOB_NBR-ADV
            Nbr_Bitmap_Index
            For(each '1' in Nbr_Bitmap_Index){
                Scanning type
                If (Scanning type == 0b010) OR (Scanning type == 0b011) {
                    Rendezvous time
                    CDMA_code
                    Transmission_opportunity offset
                }
            }
        } else {
            N_Recommended_BS_Index
            If(N_Recommended_BS_Index != 0){
                Configuration change count for MOB_NBR-ADV
            }
            For(j = 0; j < N_Recommended_BS_Index; j++){
                Neighbor_BS_Index
                Reserved
                Scanning type
                If (Scanning type == 0b010) OR (Scanning type == 0b011) {
                    Rendezvous time
                    CDMA_code
                    Transmission_opportunity offset
                }
            }
        }
```

Proposed MOB_SCN-RSP message solution

If(Use_Req_Bitmap_Index == 1){
Req_Seq_Num
Req_Bitmap_Index
For(each '1' in Req_Bitmap_Index){
Scanning type
If (Scanning type == 0b010) OR (Scanning type == 0b011) {
Rendezvous time
CDMA_code
Transmission_opportunity offset
}
}
}
N_Recommended_BS_Full
[...] No change [...]
Padding
}
TLV encoded information
}

Proposed MOB_SCN-REP

```
MOB_SCN-REP_Message_format() {
    Management Message Type = 60
    Report Mode
    N_current_BSs
    Use_Nbr_Bitmap_Index
    Use_Rsp_Bitmap_Index
    Reserved
    Report metric
    For (j = 0; j < N_current_BSs; j++) {
        Temp BSID
        Reserved
        If (Report metric Bit 0] == 1)
        [...]
    If(Use_Nbr_Bitmap_Index == 1){
        Configuration change count for MOB_NBR-ADV
        Nbr_Bitmap_Index
        For(each '1' in Nbr_Bitmap_Index){
            If(Report metric[Bit 0] == 1)
        [...]
    }else {
        N_Neighbor_BS_Index
        If (N_ Neighbor_BS_Index != 0){
            Configuration change count for MOB_NBR-ADV
        For(j = 0; j < N_Neighbor_BS_Index; j++) {
            Neighbor_BS_Index
            If(Report metric[Bit 0] == 1)
        [...]
    If(Use_Rsp_Bitmap_Index == 1){
        Rsp_Seq_Num
        Rsp_Bitmap_Index
        For(each '1' in Rsp_Bitmap_Index){
            If(Report metric[Bit 0] == 1)
        [...]
    N_ Neighbor_BS_Full
    For(j = 0; j < N_Neighbor_BS_Full;
    j++) {
        Neighbor BSID
        If(Report metric[Bit 0] == 1)
    [...]
}
```