Project	**IEEE 802.16 Broadband Wireless Access Working Group** <http://ieee802.org/16>	
Title	**Neighboring BS Self-configuration**	
Date Submitted	**2009-01-05**	
Source(s)	Joey Chou Intel	E-mail: joey.chou@intel.com
Re:	TGm SDD: SON	
Abstract	This contribution proposes text for Self-Organizing networks.	
Purpose	Adopt proposed text.	
Notice	*This document does not represent the agreed views of the IEEE 802.16 Working Group or any of its subgroups*. It represents only the views of the participants listed in the "Source(s)" field above. It is offered as a basis for discussion. It is not binding on the contributor(s), who reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy	The contributor is familiar with the IEEE-SA Patent Policy and Procedures: <http://standards.ieee.org/guides/bylaws/sect6-7.html#6> and <http://standards.ieee.org/guides/opman/sect6.html#6.3>. Further information is located at <http://standards.ieee.org/board/pat/pat-material.html> and <http://standards.ieee.org/board/pat>.	

Neighboring BS Self-configuration

Joey Chou

Intel

I. Introduction

Serving BS self-configuration is currently supported by network management systems via pre-provisioning. MAC/PHY or other system attributes will be downloaded automatically to a BS, as soon as such becomes online. However, existing cellular networks still require much manual configuration and optimization to ensure neighboring cell sites work properly and hand off connections successfully. As 4th generation mobile networks start to grow, such manual tasks would greatly burden the operators. This was the major concern that was voiced in the NGMN Industry Conference 2008.

This contribution proposes an architecture that is able to automatically

1) Update the neighboring BS list for the serving BS, and then download the BS attributes to all neighboring BS, as all BSs are going online / offline dynamically,
2) Propagate any attribute changes of a given BS to its neighboring BS.

As the result, neighboring BS should be self-configured to meet the dynamic environment of mobile wireless networks.

II. Neighboring BS Self-configuration Architecture

Figure 1 shows the neighboring BS self-optimizing algorithm to be implemented in the SON_App_Server. When a BS goes online, it will submit its BSID, cell site, Sector Bearing, and BS attributes to the SON_App_Server that implements the following steps:

1 Store Online_BS attributes, including cell site and Sector Bearing, into the database
2 Identify Online_BS neighbor list, based on cell site and Sector Bearing
3 Send a message to each neighboring BS in the list to report Online_BS attributes
4 Send a message to the Online_BS to report the neighboring BS attributes

When NMS detects a BS going offline, it will submit its BSID to the SON_App_Server that implements the following steps:

1 Update Offline_BS status in the database to offline
2 Retrieve Offline_BS neighbor list from the database
3 Send a message to each neighboring BS in the list to report Offline_BS = offline

When a BS changes its BS attribues, it will submit its BSID and updated BS attributes to the SON_App_Server that implements the following steps:

1 Store updated BS attributes into the database
2 Retrieve BS' neighbor list from the database
3 Send a message to each neighboring BS in the list to report the updated BS attributes

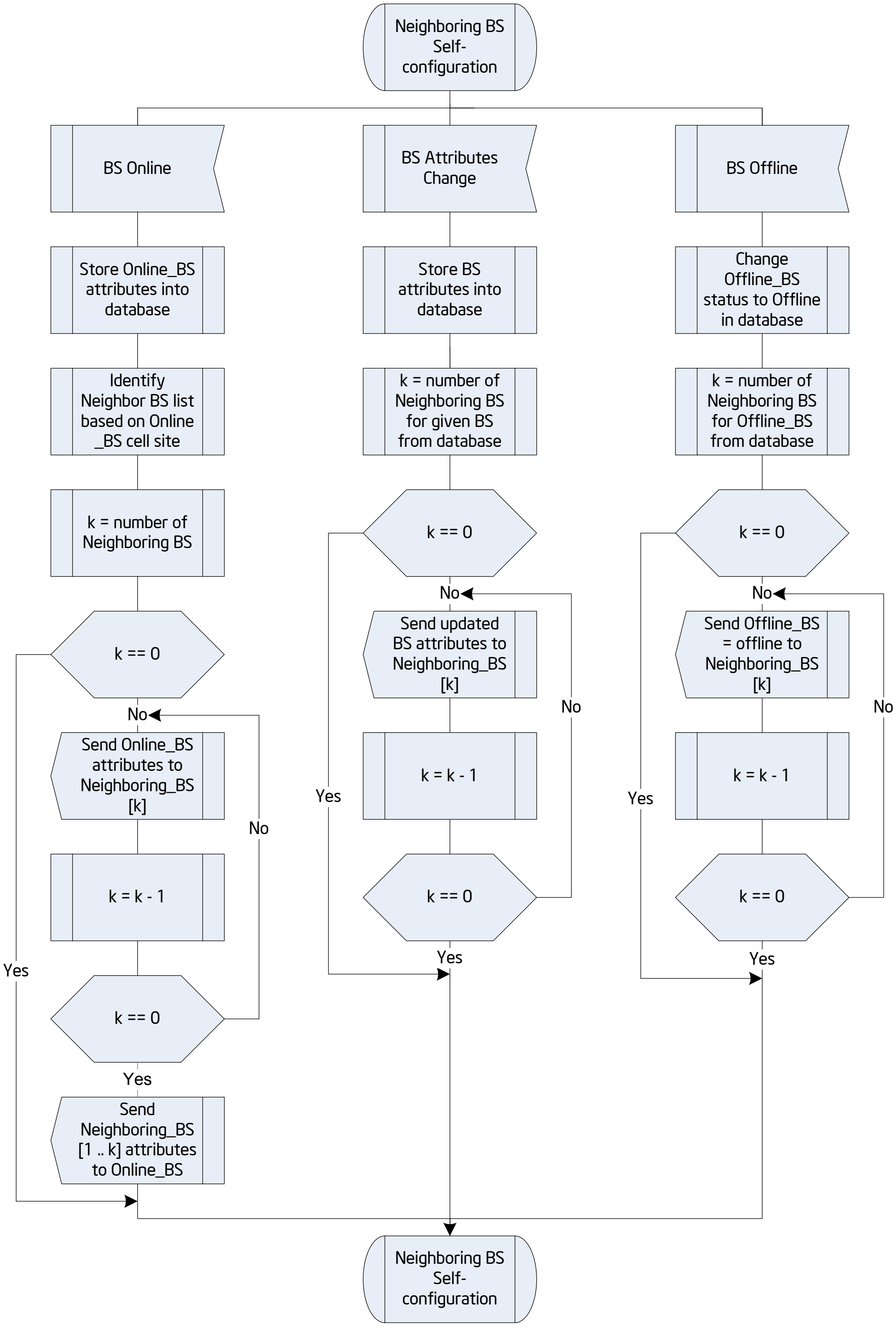
Neighboring BS Self-configuration
BS Online
BS Attributes Change
BS Offline
Store Online_BS attributes into database
Store BS attributes into database
Change Offline_BS status to Offline in database
Identify Neighbor BS list based on Online _BS cell site
k = number of Neighboring BS for given BS from database
k = number of Neighboring BS for Offline_BS from database
k = number of Neighboring BS
k == 0
No
Yes
Send Online_BS attributes to Neighboring_BS [k]
Send updated BS attributes to Neighboring_BS [k]
Send Offline_BS = offline to Neighboring_BS [k]
k = k - 1
Send Neighboring_BS [1 .. k] attributes to Online_BS
Neighboring BS Self-configuration

Figure 1: Neighboring BS Self-configuration Algorithm

Figure 2 is a cell planning example that describes how SON_APP_Server can identify its neighbors from cell site and Sector Bearing. The blue dot in each cell indicates the Cell Center in longitude and latitude. Haversine formula (1), as shown below, can be used to compute the distance *d* between two Cell Centers.

Haversine formula (1):

R = earth's radius (mean radius = 6,371km)

$\Delta lat = lat_2 - lat_1$
$\Delta long = long_2 - long_1$
$a = \sin^2(\Delta lat/2) + \cos(lat_1).\cos(lat_2).\sin^2(\Delta long/2)$
$c = 2.\mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)})$
$d = R.c$

It is assumed that Online_BS is in Cell_A with Cell Center = *A*. By computing the distances between *A* and all other cells in the serving area, SON_APP_Server can identify its neighboring BSs that have the shortest distance to Cell_A. For single sector cellular networks, Cell_B, Cell_C, Cell_D, Cell_E, Cell_F, and Cell_G will be the neighboring cells of Online_BS.

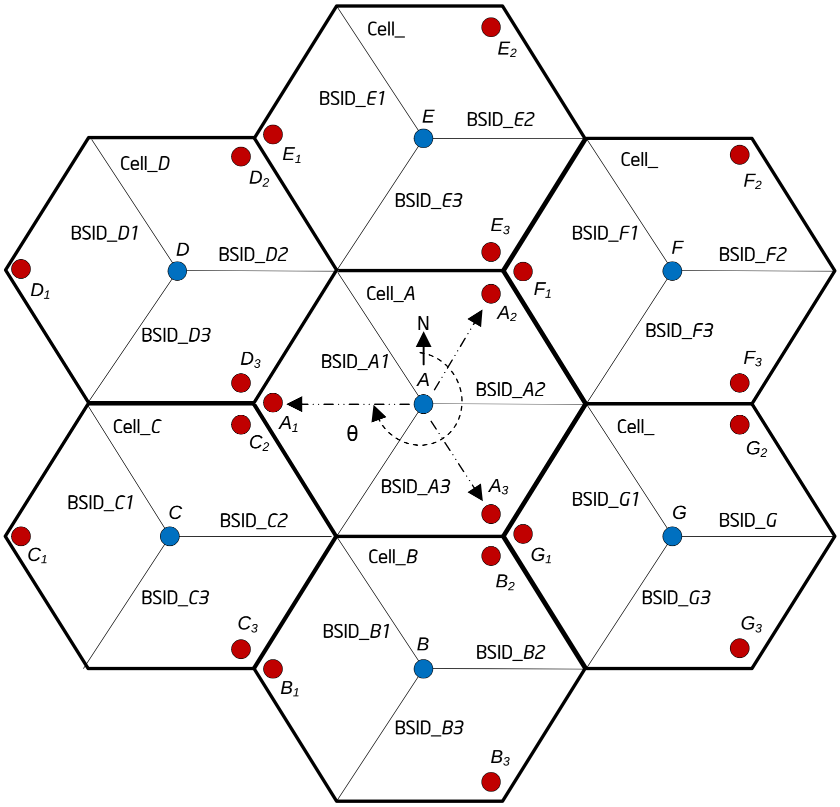

Figure 2: Example of Cell Planning

For multi-sector cellular networks, the Sector Bearing parameter, indicating the direction where the sector is pointing, will be needed to locate its neighboring sectors. By providing the Cell Center (lat_1 and lon_1), Sector Distance (d), and Sector Bearing (θ) to Haversine formula (2) below, the Sector Edge, as shown in the red dot in Figure 2, can be computed. Sector Distance should be chosen in a way that makes Sector Edge close to the edge of the sector.

Haversine formula (2):

$lat_2 = \mathrm{asin}(\sin(lat_1)*\cos(d/R) + \cos(lat_1)*\sin(d/R)*\cos(\theta))$
$lon_2 = lon_1 + \mathrm{atan2}(\sin(\theta)*\sin(d/R)*\cos(lat_1), \cos(d/R)-\sin(lat_1)*\sin(lat_2))$

It is assumed that Online_BS is Sector Edge A_1. SON_APP_Server will compute the Sector Edge for all sectors in the neighboring cells of Online_BS, based on Haversine formula (2). Then, it will compute the distance between A_1 and all other Sector Edge. SON_APP_Server can identify its neighboring sectors that have the shortest distance to A_1.

Table 1 shows an example of SON_App_Server database.

BSID	Cell ID	Cell center locations	Sector bearing	Sector edge locations	Neighboring BS List	Status	Channel Bandwidth, FFT Size, Cyclic Prefix, ….
BSID_A1	Cell_A	*A*	15°	A_1	BSID C2, C3, D3, D2, E1, E3, A2, A3, B2, B1	Online	
BSID_A2	Cell_A	*A*	135°	A_2	BSID E1, E3, F1, F3, G1, G2, A1, A3, B2	Online	
BSID_A3	Cell_A	*A*	255°	A_3	BSID F1, F3, G2, G1, B2, B1, C3, C2, A1, A3	Online	
BSID_B1	Cell_B	*B*	15°	B_1	BSID C3, C2, A1, A3, B2	Online	
BSID_B2	Cell_B	*B*	135°	B_2	BSID C3, C2, A1, A3, G1, G3, B1	Online	
BSID_B3	Cell_B	*B*	255°	B_3	BSID B1, B2, G1, G3	Offline	
BSID_C1	Cell_C	*C*	15°	C_1	BSID D1, D3, C2, C3	Offline	
BSID_C2	Cell_C	*C*	135°	C_2	BSID B1, A1, A3, B2, B1, C3	Online	
BSID_C3	Cell_C	*C*	255°	C_3	BSID C2, A1, A3, B2, B1	Online	

Table 1: Example of SON_App_Server Database

III. Proposed text

18. Support for Self-organization

18.1 Self-Configuration

18.1.x Neighboring BS Self-Configuration

Existing cellular networks still require much manual configuration of neighboring BS that will greatly burden the operators in the network deployment. Therefore, SON shall support neighboring BS self-configuration that will automatically:

1 Update the neighboring BS list for the serving BS, and then download the BS attributes to all neighboring BS, as all BSs are going online / offline dynamically,

2 Propagate any attribute changes of a given BS to its neighboring BS.

To support neighboring BS self-configuration, BS shall send the following parameters to the SON_App_server:

1 BSID

2 Cell site in longitude, latitude

3 Sector Bearing, indicating the direction where the sector is pointing

4 BS attributes (e.g. Channel Bandwidth, FFT Size, Cyclic Prefix, ….)

SON_App_Server shall send the following parameters to the neighboring BS:

1 BSID

2 BS attributes (e.g. Channel Bandwidth, FFT Size, Cyclic Prefix, ….)