Project	**IEEE 802.16 Broadband Wireless Access Working Group** <http://ieee802.org/16>
Title	**Self-Optimizing FFR**
Date Submitted	**2009-01-05**
Source(s)	Joey Chou Shilpa Talwar Clark Chen Intel E-mail: joey.chou@intel.com Shilpa.Talwar@intel.com Clark.Chen@intel.com
Re:	TGm SDD: SON
Abstract	This contribution proposes text for Self-Organizing networks.
Purpose	Adopt proposed text.
Notice	
Release	
Patent Policy	

Self-Optimizing FFR

Joey Chou, Shilpa Talwar, Clark Chen
Intel

I. Introduction

As described in section 20.1, FFR is intended to use frequency reuse factor = 1 to serve MSs located in inner cell that do not experience significant inter-cell interference, and frequency reuse factor < 1 for MSs located at the cell edge that tend to receive unacceptable level of interference. The distribution of frequency partitions and power levels parameters among BSs has to be coordinated in order to avoid collision between neighboring BSs.

This contribution proposes a self-optimizing FFR architecture to support dynamic allocation of FFR frequency partitions and load balancing.

II. Self-Optimizing FFR Architecture

Increasing complexity and dynamic environment in today's mobile networks require constant analysis, provisioning and tuning of huge amount of parameters for equipment spread across great geographical area in order to optimize network performance, coverage, and capacity. Self-optimization is the process of utilizing measurement data to optimize the network performance by automatically tuning the system attributes. Figure 1 shows an example of the Self-organizing Network Architecture.

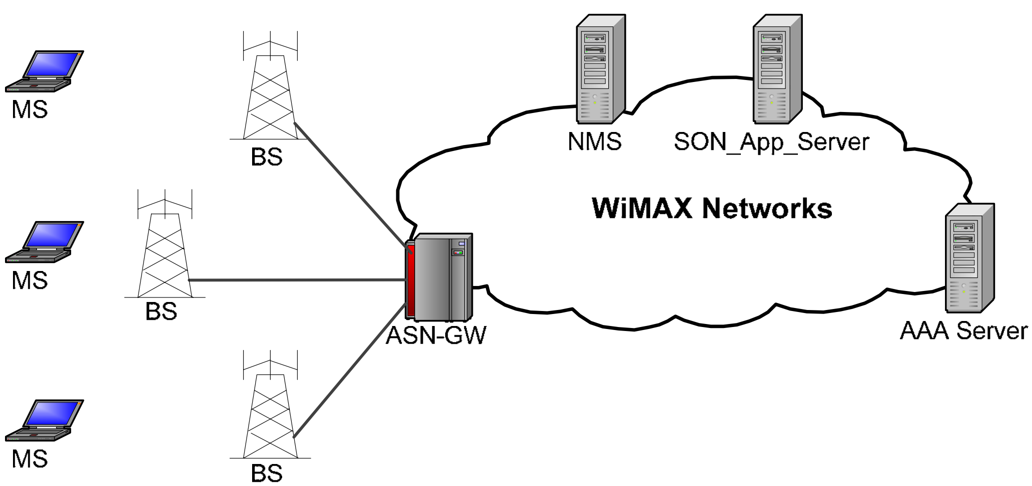

Figure 1: Self-organizing Network Architecture

Figure 2 shows the Self-optimizing FFR Algorithm to be implemented in the SON_App_Server. All BS in the serving area will report performance metrics to the SON_App_Server. In the interval defined by FFR partition refresh timer parameter, SON_App_Server will analyze performance metrics reported by BSs during the past interval to recalculate frequency partitions and power levels that should be sent to all BSs. When a BS is added to the serving area or becomes on line, it can request the frequency partition and power level from the SON_App_Server.

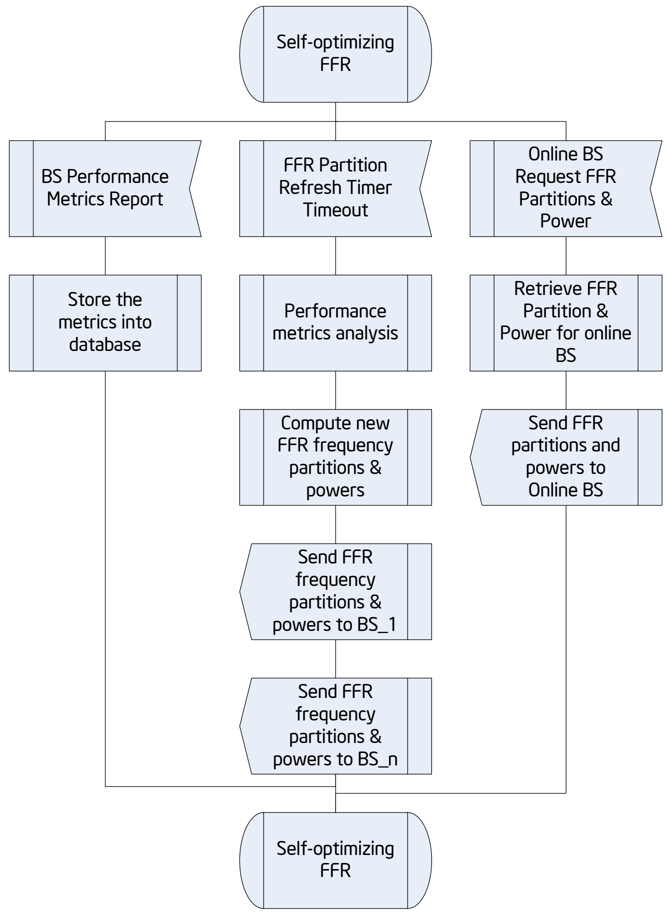

Figure 2: Self-optimizing FFR Algorithm

Figure 3 shows an example of FFR frequency partitions for frequency reuse factor 1 and 1/3. The frequency partitions are represented as $F_{i,j}$, where *i* = frequency reuse factor, *j* = frequency band.

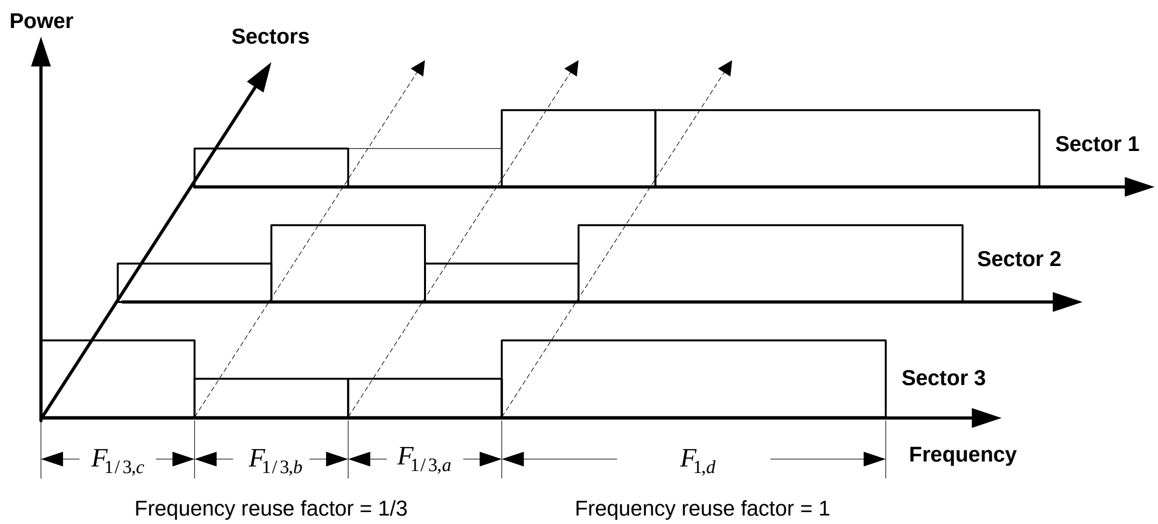

Figure 3: Example of FFR Frequency Partitions

Figure 4 shows the distribution of frequency partitions in a 3 sector cellular networks. $F_{1/3,a}$ / $F_{1/3,b}$ / $F_{1/3,c}$ and $F_{1,d}$ represent the frequency partitions for frequency reuse 1/3 and 1 respectively. A key requirement of self-optimizing FFR is to avoid collisions among neighboring sectors, when distributing frequency partitions and power levels to each BS in the serving area.

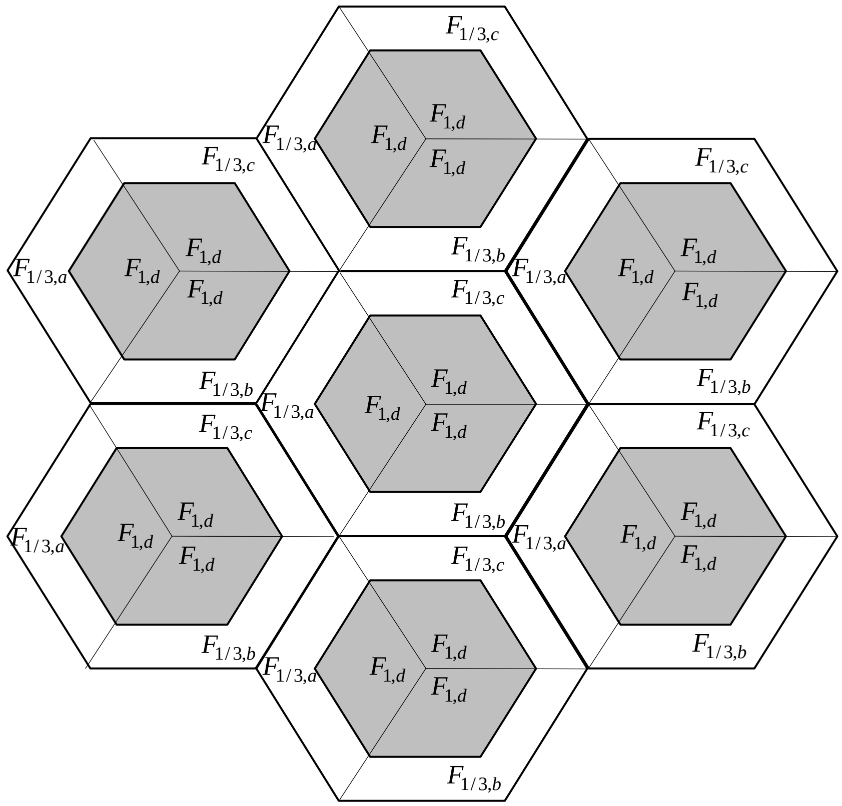

Figure 4: FFR Frequency Partition Distributions

Many factors should be considered in the selection of FFR frequency partitions parameters. As described previously, since FFR is mainly designed for the benefit of cell edge users, an obvious parameter will be the MS location distribution. However, some MSs, even though not located in the cell edge, may receive poor SINR due to fading or shadowing. Therefore, SINR distribution parameters should be considered.

In mobile WiMAX, the number of MS and the traffic load carried in a BS will fluctuate up and down continuously, as MSs roam from BS to BS. In traditional frequency planning, the bandwidth allocated to each BS is fixed that result in either traffic overload in some BSs or bandwidth waste in other BSs. FFR can support load balancing by taking into account the sector traffic loads of each sector in the FFR frequency partitions selection process.

The BS traffic load metrics can be measured by counting the aggregate user data passing through in a fixed sampling interval. The smaller the sampling interval, the better resolution the traffic load data provides at the cost of higher overhead to the BS. Typically, the sampling interval can be in the order of seconds.

The following lists the parameters that each BS should send to the SON_App_Server to be used by the Self-optimizing FFR algorithm.

- BSID
- MS Number in the BS

- MS location distribution – is indicated by the mean and standard deviation of MS timing advances that are measured in the periodic ranging process.
- MS UL/DL SINR distributions per FFR partition – are indicated by the mean and standard deviation of MS UL/DL SINR that are measured on per FFR partition basis
- UL / DL traffic distribution per FFR partition – are indicated by the mean and standard deviation of UL/DL traffic load samples, as shown in Figure 5, on per FFR partition basis. The traffic load samples count the number of octets of MAC PDUs (i.e. user data in MAC SDU, MAC headers, and MAC management messages) transmitted or received at the BS in a sampling interval. UL / DL traffic distribution can be use to validate the performance of self-optimizing FFR algorithm.
- Converged resource metrics per FFR partition – see SDD section 20.1.1.1 for resource metrics

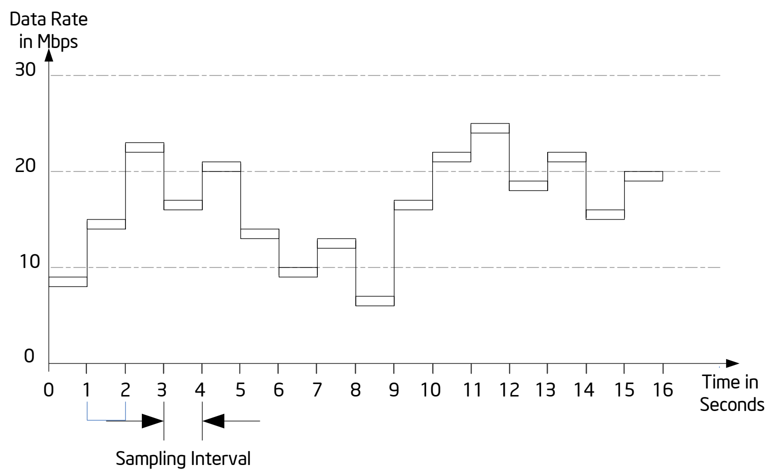

Figure 5: Example of Traffic Load Metrics

The following lists the parameters that SON_App_Server should distribute to each BS in the serving area.

- FFR partitions – $F_{1/3,a}$, $F_{1/3,b}$, $F_{1/3,c}$, and $F_{1,d}$, as shown in Figure 3
- Power levels – the power level should be used in each partition
- Relative Load indicator – the average traffic of a the given BS in comparison with other BS
- Time stamp change – indicates when the change will take effective in all BS in the serving area

Figure 6 shows a control flow example of the self-optimizing FFR algorithm.

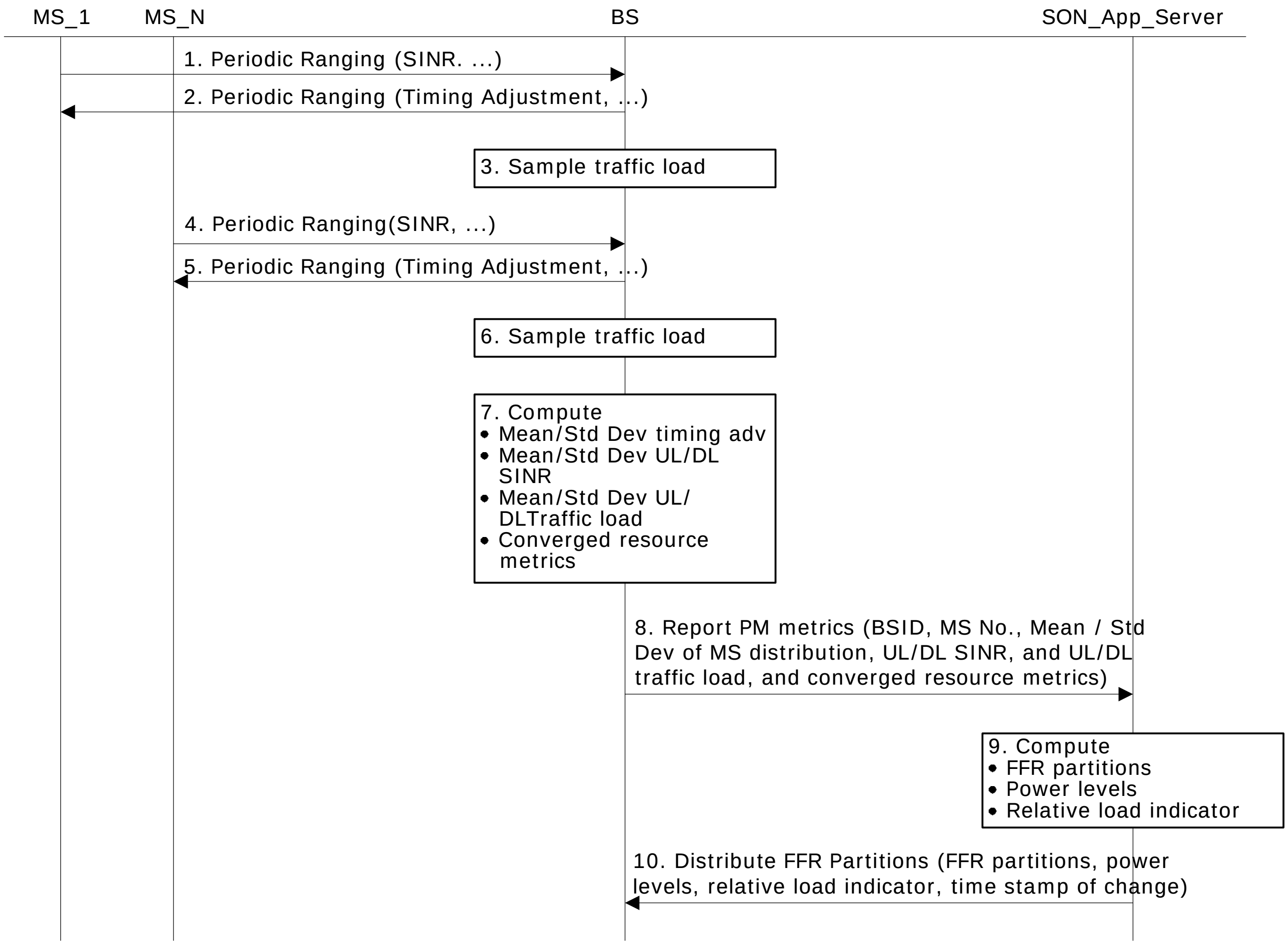

Figure 6: Control Flow Example of Self-optimizing FFR Algorithm

III. Proposed text

18. Support for Self-organization

18.2 Self-Optimization

18.2.x Self-optimizing FFR

FFR utilizes multiple partitions of different frequency reuse factors 1 and < 1 to increase sector throughput while addressing interference issues for cell edge users. Self-optimizing FFR is designed to automatically adjust FFR parameters, frequency partitions and power levels, among BS sectors in order to optimize system throughput and user experience.

The following lists the parameters that each BS should send to the SON_App_Server to be used by the Self-optimizing FFR algorithm. This algorithm optimizes FFR parameters and supports load balancing among BS by taking into account factors such as MS distribution, SINR distribution, resource utilization (metrics), and traffic load requirements for each partition.

- BSID
- MS Number in the BS
- MS location distribution – is indicated by the mean and standard deviation of MS timing advances that are measured in the periodic ranging process.
- MS UL/DL SINR distributions per FFR partition – are indicated by the mean and standard deviation of MS UL/DL SINR that are measured on per FFR partition basis
- UL / DL traffic distribution per FFR partition – are indicated by the mean and standard deviation of UL/DL traffic load samples, on per FFR partition basis. The traffic load samples count the number of octets of MAC PDUs (i.e. user data in MAC SDU, MAC headers, and MAC management messages) transmitted or received at the BS in a sampling interval. UL / DL traffic distribution can be use to validate the performance of self-optimizing FFR algorithm.
- Converged resource metrics per FFR partition – see section 20.1.1.1 for resource metrics description

The following lists the parameters that SON_App_Server should distribute to each BS in the serving area.

- FFR partitions – frequency partitions for frequency reuse factors 1 and 1/3
- Power levels – the power level should be used in each partition
- Relative Load indicator – the average traffic of a the given BS in comparison with other BS
- Time stamp change – indicates when the change will take effective in all BS in the serving area

Figure x shows a control flow example of the self-optimizing FFR algorithm.

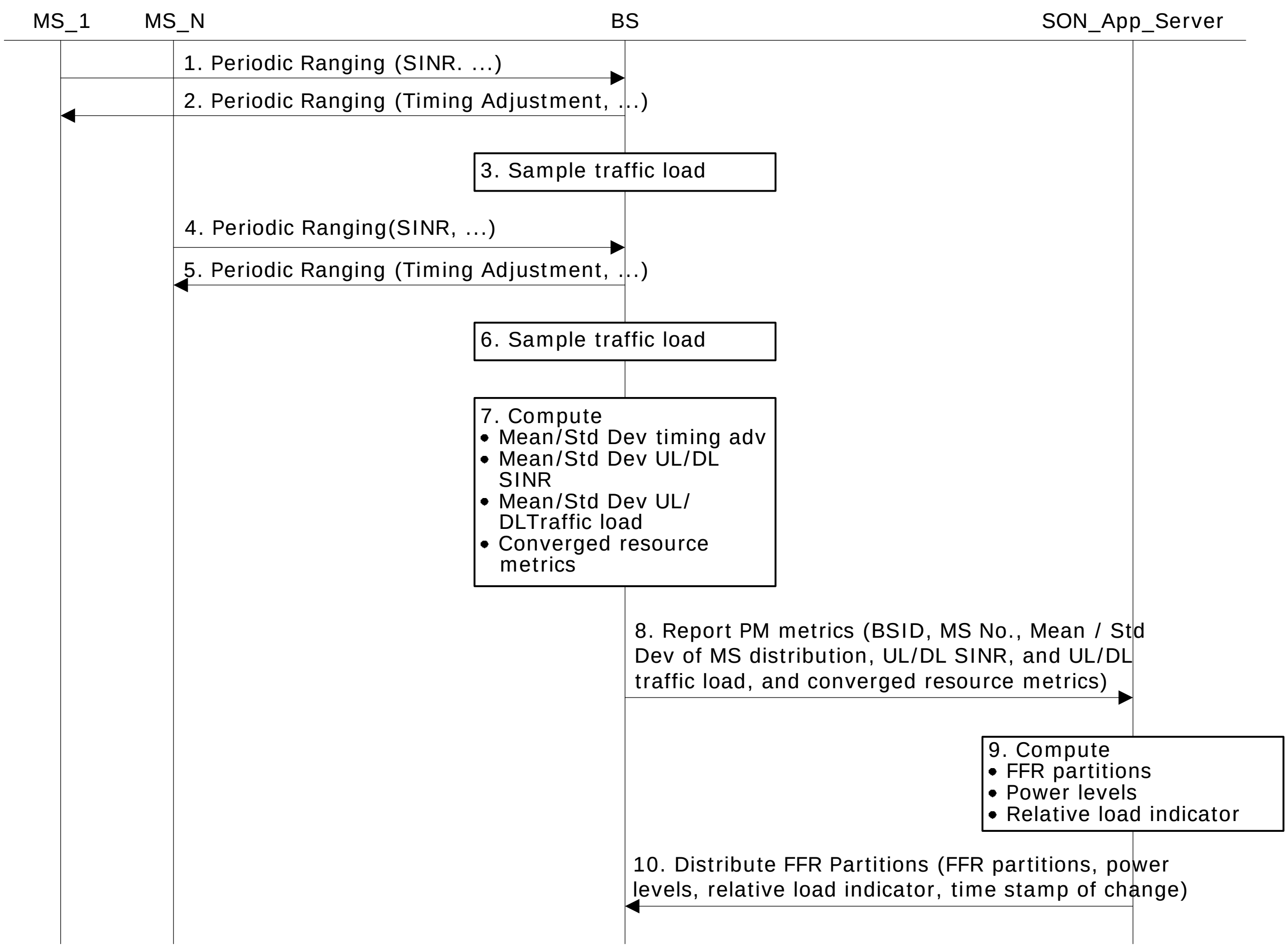

Figure x: Control Flow Example of Self-optimizing FFR Algorithm