Data Links

User Guide – EDL Nano Product

AE301628-001

REV B

3 October 2019

For:
Part No. AE102474 & AE102882

Prepared by:

AERONIX

1775 West Hibiscus Blvd, Suite 200
Melbourne, FL 32901
(321) 984-1671

Document Authorization

Review/Approval Title	Name
Originator:	S. Baduini
Approved By: Program Manager	L. Clarkson
Approved By: Quality Assurance	C. Hertzberg
Approved By: Configuration Management	D. Franks

Document Revision History

Ver/Rev	Description of Change	Chg'd By	Chg/Rel #	Date
-001/-	Initial L1 Release to CM	LC & SB	SP-2493	7/10/2019
-001/A	Release with updates for FCC testing	LC	SP-2519	9/11/2019
-001/B	Updated with FCC Statements	LC		10/3/2019

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1.1. Purpose

User guide for Aeronix EDL Nano based family of data links. This user guide has information based on the NANO GMSK and OFDM software release.

1.2. References

The following documents are referenced in or were used in the preparation of this document:

1.2.1. Government Documents

None.

1.2.2. Non-Government Documents

[1] "IEEE STD 802 2004 Part 16d: Air Interface for Fixed Broadband Wireless Access Systems" Revision 2004. June 24,2004

[2] "IEEE STD 802 2005 Part16e: Air Interface for Fixed and Mobile Broadband Wireless Access System Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands" Revision 2005.

[3] AE301432-001 : Report RF Statistics Via Ethernet Interface Description

1.3. Overview

This user guide provides information deemed necessary to bring up the following radio products from point of delivery:

- Part No. AE102429 – EDL Nano Digital Data Link
- Part No. AE102882 – EDL Nano Digital Data Link Rugged

(WARNING: DO NOT POWER UP WITHOUT A LOAD ON THE ANTENNA PORT)

1.4. Notes for FCC ID

FCC testing was completed using GMSK low power mode (30 dBm) and frequencies within the ISM Band (2400 – 2483.4 MHz).

Exposure Statement:

RF Exposure WARNING: This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.

At 1000 mW Power (1W) and Antenna Gain of 4 (6 dB), and a minimum distance of 20cm (min separation distance for “Mobile Devices”).

S = 0.8 mW/cm2, which is less than the Uncontrolled Exposure Maximum Permissible Exposure (MPE) limit.

For radios labeled with FCC ID T2KAE102882 the following applies:

	FCC Part 15.19 Warning Statement- (Required for all Part 15 devices)
	THIS DEVICE COMPLIES WITH PART 15 OF THE FCC RULES. OPERATION IS SUBJECT TO THE FOLLOWING TWO CONDITIONS: (1) THIS DEVICE MAY NOT CAUSE HARMFUL INTERFERENCE, AND (2) THIS DEVICE MUST ACCEPT ANY INTERFERENCE RECEIVED, INCLUDING INTERFERENCE THAT MAY CAUSE UNDESIRED OPERATION.
	FCC Part 15.21 Warning Statement-
	NOTE: THE GRANTEE IS NOT RESPONSIBLE FOR ANY CHANGES OR MODIFICATIONS NOT EXPRESSLY APPROVED BY THE PARTY RESPONSIBLE FOR COMPLIANCE. SUCH MODIFICATIONS COULD VOID THE USER’S AUTHORITY TO OPERATE THE EQUIPMENT.

2. EDL Nano - Enhanced Data Link (AE102429) and Rugged Enhanced Data Link (AE102882)

This unit is a software reprogrammable data link. This unit supports both Aeronix LAW functionality, GMSK waveform and video compression capabilities. The EDL Nano operates over the frequency band of 1.8 to 2.5 GHz with the appropriate antenna systems. **(WARNING: DO NOT POWER UP WITHOUT A LOAD ON THE ANTENNA PORT).**

Note: The pictures in this section are from AE102429. But, AE102882 operates the same and is just in a different housing.

Figure 2-1 NANO

2.1. EDL Nano Features

- Software Reprogrammable Radio
- Simple yet powerful GUI for radio configuration.
- Option to use SNMP for radio configuration
- Frequency Range 1.8-2.4 GHz in 1 MHz steps
- Linux OS on Quad Core Processor
- Xilinx Artix 200 FPGA
- Diversity Receive
- Less than 4 cubic inches
- Aeronix provided OFDM and GMSK Waveforms available

2.2. EDL Nano Installation and Startup Procedures

2.2.1. Input Power Supply

The EDL unit requires a voltage be provided between 8 – 18V. Application of power to the data link turns on the system.

2.2.1.1. NANO GMSK current and power measurements

Below is a measurement of the maximum current draw from a NANO GMSK radio over a period of time.

Table 2-1. GMSK Typical Peak Current and Power 1800 Mhz

Description	Voltage (VDC)	Current (mA)	Power (W)	Tx power dBm
BS Tx 50% ,50% Rx, data, Hi Power	12	1496	18.900	35
BS Tx 50% ,50% Rx, data, Low Power	12	1425	17.100	33
BS Tx 70% , 30% Rx, data, Hi Power	12	1794	21.528	35
BS Tx 70% , 30% Rx, data, Low Power	12	1600	19.200	33
BS Tx/Rx Off	12	411	4.932	N/A

Table 2-2. GMSK Typical Peak Current and Power 2300 MHz

Description	Voltage (VDC)	Current (mA)	Power (W)	Tx power dBm
BS Tx 50% ,50% Rx, data, Hi Power	12	1275	15.300	35
BS Tx 50% ,50% Rx, data, Low Power	12	996	11.952	33
BS Tx 70% , 30% Rx, data, Hi Power	12	1419	17.028	35
BS Tx 70% , 30% Rx, data, Low Power	12	1235	14.820	33
BS Tx/Rx Off	12	408	4.896	N/A

2.2.1.2. NANO OFDM current and power measurements

Measurements were taken at room temperature (25 C).

Table 2-3. OFDM Typical Current and Power 1800 MHz

Description	Voltage (VDC)	Current (mA)	Power (W)	TX power (dBm)
BS Tx 50% data, no video, Hi data rate	12	920	11.0	31
BS Tx 50% data, no video, Hi power	12	965	11.6	33
BS Tx 70% data, no video, Hi data rate	12	945	11.3	31
BS Tx 70% data, no video, Hi power	12	995	11.9	33

Table 2-3. OFDM Typical Current and Power 1800 MHz				
Description	**Voltage (VDC)**	**Current (mA)**	**Power (W)**	**TX power (dBm)**
BS Tx/Rx Off	12	600	7.2	N/A

Table 2-4. OFDM Typical Current and Power 2300 MHz				
Description	**Voltage (VDC)**	**Current (mA)**	**Power (W)**	**TX power (dBm)**
BS Tx 50% data, no video, Hi data rate	12	845	10.1	29
BS Tx 50% data, no video, Hi power	12	870	10.4	32
BS Tx 70% data, no video, Hi data rate	12	850	10.2	29
BS Tx 70% data, no video, Hi power	12	880	10.6	32
BS Tx/Rx Off	12	600	7.2	N/A

Table 2-5. OFDM Typical Current and Power 2427 MHz				
Description	**Voltage (VDC)**	**Current (mA)**	**Power (W)**	**TX power (dBm)**
BS Tx 50% data, no video, Hi data rate	12	910	10.9	30
BS Tx 50% data, no video, Hi power	12	930	11.2	33
BS Tx 70% data, no video, Hi data rate	12	915	11.0	30
BS Tx 70% data, no video, Hi power	12	940	11.3	33
BS Tx/Rx Off	12	600	7.2	N/A

2.3. EDL Nano Physical I/O

The front panel of an EDL Nano is shown in the following figure. Indicated ports are described below.

Figure 2-2 NANO RF Input/Output

Table 2-6. Nano Hardware Interface

Connector	Description	EDL Connector	Mating Connector
P1	Tx/Rx1: Female SMA connector connects to the desired antenna	Male SMA	Female SMA
P2	Rx2: Female SMA connector connects to the desired antenna	Male SMA	Female SMA

The back panel of an EDL Nano is shown in the following figure. Indicated connectors are described below.

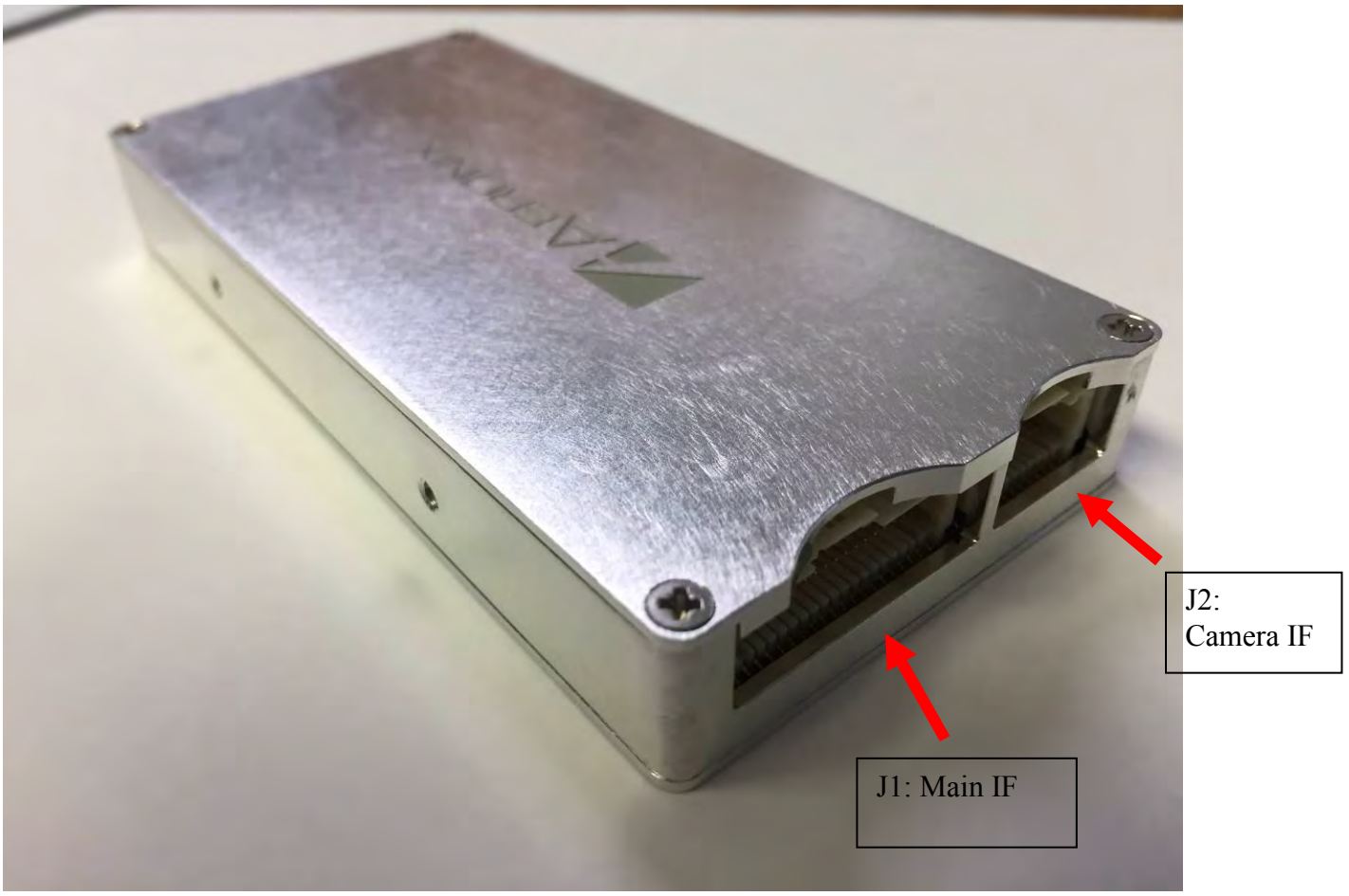


Figure 2-3 Nano Physical I/O

Table 2-7. Nano Hardware Interface			
Connector	**Description**	**EDL Connector**	**Mating Connector**
J1	Main IF: Header 2x20	Molex 501571-4007	Molex 501189-4010
J2	Camera IF: Header 2x10	Molex 501571-2007	Molex 501189-2010

Table 2-8. J1: Main IF Pin Definitions	
Pin	**Description**
1	8-18 VDC
2	Ground

3	8-18 VDC
4	Ground
5	8-18 VDC
6	Ground
7	Platform RS-232 Rx
8	USB OTG VBUS
9	Platform RS-232 Tx
10	USB OTG DP
11	Ext GPIO 0 – External Tx Enable (1.8V)
12	USB OTG DN
13	Ext GPIO 2 – External Rx Enable (1.8V)
14	Ground
15	Ext GPIO 1
16	Console RS-232 Tx
17	Ext GPIO 3
18	Console RS-232 Rx
19	Green LED
20	Ground
21	Red LED
22	Ground
23	RX TX Enable
24	Ground
25	TAMPER
26	Ground
27	VBUS 5V
28	USB Host DN
29	Ground
30	USB Host DP
31	Ether2 Tx+
32	Ether1 Tx+
33	Ether2 Tx-
34	Ether1 Tx-
35	Ground
36	Ground
37	Ether2 Rx+
38	Ether1 Rx+
39	Ether2 Rx-
40	Ether1 Rx-

Table 2-9. J2: Camera IF Pin Definitions	
Pin	**Description**
1	3 VDC
2	Ground
3	CAM SDA
4	CAM CLK
5	CAM SCL
6	CSI D3P
7	CAM EN
8	CSI D3M
9	CAM RST#
10	Ground
11	CSI D0P
12	CSI D2P
13	CSI D0M
14	CSI D2M
15	Ground
16	Ground
17	CSI D1P
18	CSI CLKDP
19	CSI D1M
20	CSI CLKDM

2.3.1. USB OTG

USB OTG port is currently only used to download initial firmware. Software to use the USG OTG to recognize other applications has not been implemented.

2.3.2. USB Host

Software has not been formally tested.

2.3.3. Ethernet 1 & 2

Ethernet ports are both IEEE802.3 compliant 10/100 Mbps connection. Ethernet 1 is the primary Ethernet for the radio. Ethernet 2 currently is not configurable and usable on the radio.

2.3.4. Console RS-232

The Serial interface allows direct communication with the radio's control processor. This interface is a RS-232 standard interface. Port is configured as 8-N-1, 115200 BAUD. This port provides the radio's login console. You can login with administrative privileges by entering "root" at the login prompt and password "1234".

2.3.5. Platform RS-232

Serial interface is for serial to serial communication. This interface is a RS-232 standard interface. The debug port is configured as 8-N-1, the BAUD rate is configurable via the webgui or SNMP. This interface is used to pass serial data from a BS serial port to one of its SS serial port.

2.3.6. P1 Antenna Interface TxRx 1

The antenna interface is an SMA connector that provides both input and output RF signalling.

2.3.7. P2 Antenna Interface Rx 2

The antenna interface is an SMA connector that provides input RF signalling.

2.3.8. Tx Rx Enable

The Tx/Rx Enable is a three state condition. It requires an external pull-up resistor of 4.7 kilo ohms to supply voltage V (customer determined with a max of 5V), see diagram below.

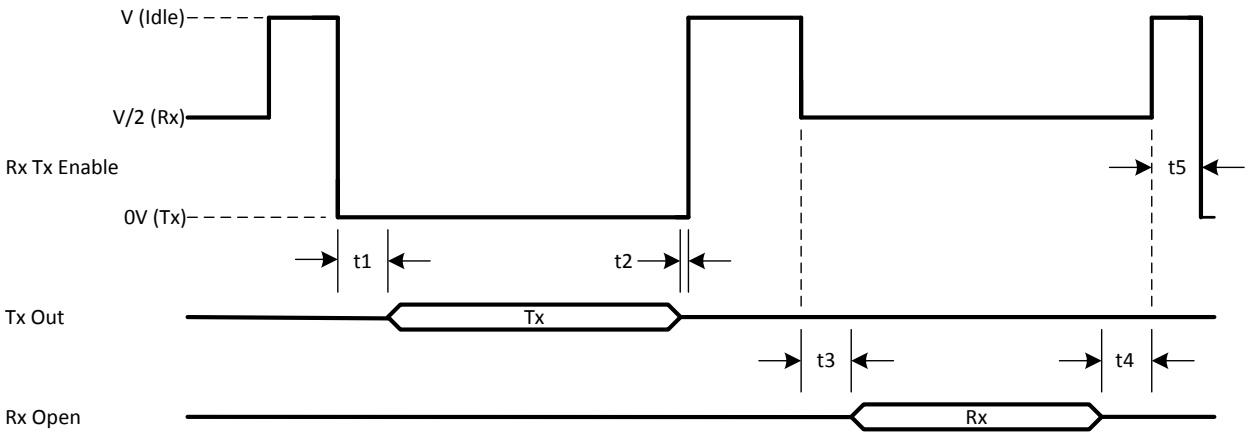


Figure 2-4 TR SW Timing

Table 2-10 GMSK 14 MHz T/R SW Timing		
Marker	**Time (us)**	**Description**
t1	TBD	Minimum time from Tx Enable Active to start of Transmit Out
t2	TBD	Minimum time from Transmit Out Complete to Tx Enable Inactive
t3	TBD	Minimum Time from Rx Enable Active to Receive energy
t4	TBD	Minimum time from Receive Energy Complete to Rx Enable Inactive
t5	TBD	Minimum switching time, ie minimum amount of Idle time before a radio will switch between Rx and Tx state or Rx and Rx state

Table 2-11 OFDM T/R SW Timing		
Marker	**Time (us)**	**Description**
t1	12 us	Minimum time from Tx Enable Active to start of Transmit Out
t2	0	Minimum time from Transmit Out Complete to Tx Enable Inactive
t3	80 us	Minimum Time from Rx Enable Active to Receive energy
t4	11 us	Minimum time from Receive Energy Complete to Rx Enable Inactive
t5	10.5 us	Minimum switching time, ie minimum amount of Idle time before a radio will switch between Rx and Tx state or Rx and Rx state

2.3.1. Tx Rx Enable 4 State (OFDM only)

External Tx and Rx control lines with 1.8V logic.

2.3.2. External GPIO 1 & 3

Not Used

2.3.3. Red and Green LED

The external Red and Green LEDs will give the user some basic status of the radios current state.

- Green only LED on -> The NANO is in sync with another radio
- Red only LED on -> The Nano's RF is turned off via the software Tx/Rx enable/disable setting.
- Green and Red LED on (Yellow) -> The NANO RF is on, but NOT in sync with another radio.

2.3.4. Tamper

Not Used

2.3.5. RF Output Power

The NANO output power will remain unchanged regardless of the RF environment. The NANO output power is based on frequency and is calibrated per Radio Card. Each radio will have its own calibration programmed and available for the user to view. Each NANO may vary from 1-2 dB of output power. See section 7.1 NANO output power for more information on determining the output power measured on a particular frequency.

3. System Configuration

3.1. Aeronix GUI interface

This simple yet powerful GUI uses uncomplicated intuitive screens to allow the average user the ability to:

- Configure the system for their network
- Make the system operational
- Perform maintenance

The GUI screens and all their respective elements are described in the following subsections.

To get to the main page, type the following URL: http://<ip address of the radio>/

The recommended browsers to use are Internet Explorer Version 7 or newer and Mozilla Firefox.

3.2. SNMP interface

The NANO can be configured to run the Aeronix SNMP agent, and process SNMP (Simple Network Management Protocol) requests using the radio's IP. This allows the user to configure and monitor the radio over an IP network.

3.3. Startup

Each unit is configured with an IP address. Using the IP address, one can access the GUI which can be used for configuring the radio to the user's specifications. The typical factory default IP address that is set from the factory is 192.168.1.1 and the units default configuration is that of subscriber 1, unless specified otherwise. To browse the GUI type in the IP address of the unit in an internet browser. More information about the GUI screens and how to configure the unit can be found in this section.

3.4. Radio Maintenance Login

3.4.1. Using Serial Port A

1) Make sure antenna port has a load on it.
2) Apply appropriate power to unit.
3) Connect to Serial A port with terminal emulator. (Baud = 115200, 8-N-1 no flow control)
4) Login by typing "root"
5) Enter password "1234".
6) $ cd /mnt/jffs2

3.5. Ethernet alias IP address

The user can always access the radio over its Ethernet alias IP address of 169.254.1.1. This IP is consistent so if the user needs to access the GUI or SNMP and has forgotten the configured IP address for the radio's Ethernet, then the user can access 169.254.1.1.

3.6. WEB Based GUI System Configuration

This simple yet powerful GUI uses uncomplicated intuitive screens to allow the average EDL user:

- Configure the system for their network
- Take the system operational
- Perform maintenance

The GUI screens and all their respective elements are described in the following subsections.

To get to the main page, types the following URL: http://<ip address of the radio>/

The recommended browsers to use are Internet Explorer Version 8 or newer and Mozilla Firefox.

The sections below show screen shots using either OFDM or GMSK mode. If there is a difference in the GMSK versus OFDM screen captures then both captures will be displayed.

3.6.1. Base Station Status Page

The base station status page is the default page displayed when the radio's IP address is entered. The status page lists the current active subscribers received by the BS. It also lists the current configuration of the base station.
There are attributes each web page share, these are highlighted by items A and B in the image below.

A	This section displays the current mode radio is in and the HTML links to the all of the pages available.
B	The radio's configured name and IP address, located on the web page tab.

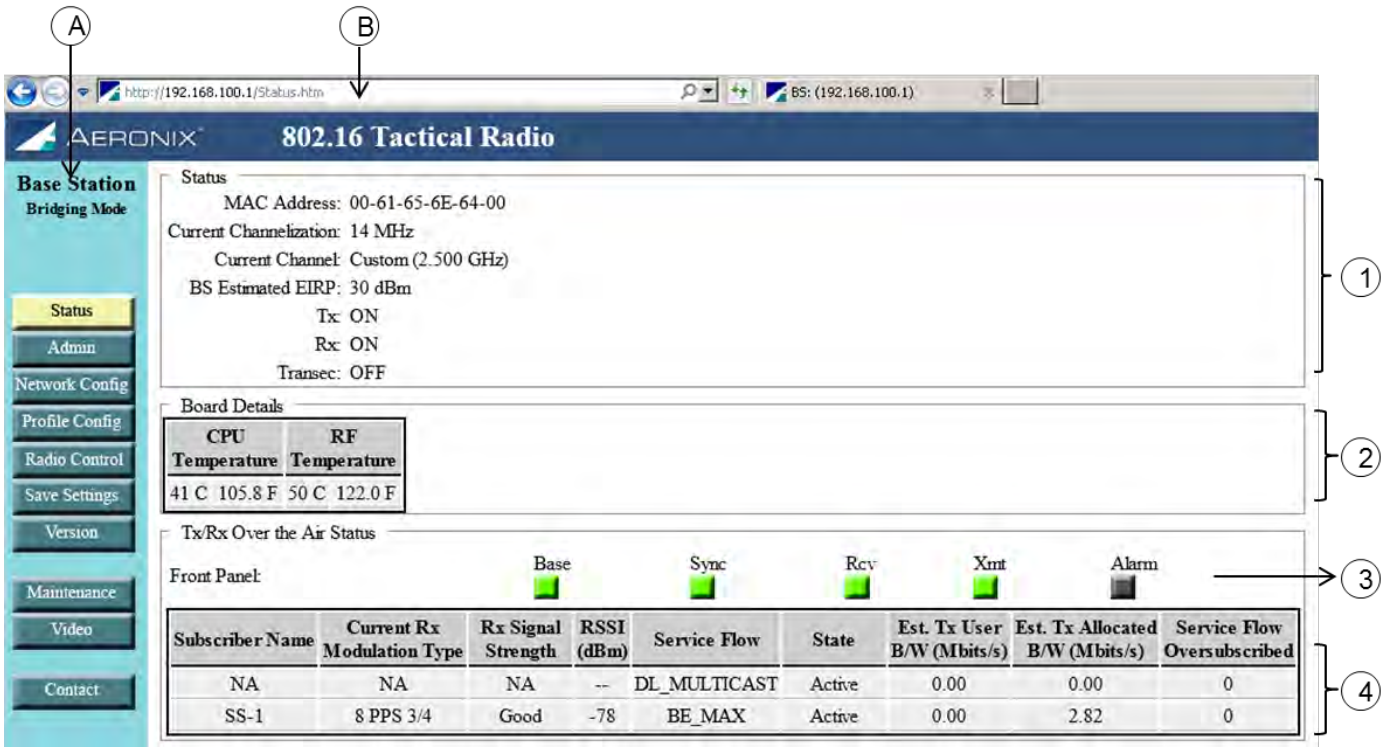


Figure 3-1 BS GMSK Status

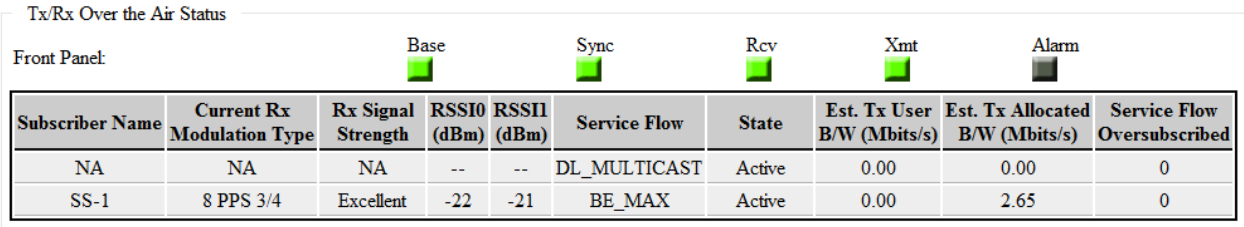
Table 3-1. Base Station Status Page

	Field/Control	Description
1	Status	Status displays a subset of current settings. • MAC Address – The IEEE 802.16 MAC address of the base station • Current Channelization – Displays the channelization of the base station. The channelization may be changed on the Radio Control Screen. • Current Channel – The current 802.16 specified channel that the base station uses for transmit and receive. The channel may be changed on the Radio Control screen. • BS Estimated EIRP – Displays the estimated EIRP of the base station based upon user configured gain a loss values. (Est Tx Pwr + Antenna Gain – External Tx Cable Loss – Antennal Cable Loss – Internal Tx Loss) • Tx - Provides the status of the transmit portion of the unit. (On/Off) The status may be changed on the Radio Control screen. • Rx - Provides the status of the receive portion of the unit. (On/Off) The status may be changed on the Radio Control screen. • Transec Status – Displays the status of Transec (on or off). The Transec status can be changed on the Admin Screen.
2	Board Details	Display Temperature • CPU Temperature – Displays the temperature reported by the processor in Celsius and Fahrenheit • RF Temperature – Displays the temperature reported by the AD9361 chip on the RF card in Celsius and Fahrenheit
3	Front Panel	Displays PHY status on an interval basis. Base is a constant green light if the radio is configured as a Base Station • Sync is off on the base station when no subscribers are registered and is illuminated continuously when subscribers are registered. • Recv is green when the radio is receiving messages • Xmit is green when the radio is transmitting data • Alarm

Table 3-1. Base Station Status Page		
	Field/Control	**Description**
4	Tx/Rx Over the Air Status	The first part of the Tx/Rx Status Table Displays the receive status of each subscriber connection. The second part of the table is the amount of data the BS is transmitting to each active SS. • Subscriber Name - Configured Subscriber's name. • Current Modulation Type – Contains the current modulation/FEC type for the subscriber connection. The display show the modulation of the receive frame at the time sampled. • Rx Signal Strength: Contains an assessment of the receive signal strength. • RSSI (dBm) – Display the RSSI value measured at the P1 antenna port.See Figure 3-2 BS Status GMSK RX Dual mode for RX Dual mode display • Service Flow – Identifies the service class of the service flow for the connection. • State – Displays the state of the connection. The valid values are provisioned and active. • Est User B/W (Mbits/sec) – Displays the estimated bandwidth in megabits per second being used by the connection in the transmit direction. • Est Allocated B/W (Mbits/sec) – Displays the estimated bandwidth in megabits per second being allocated to the connection in the transmit direction. • Service Flow Oversubscribed – Displays the number of Ethernet packets dropped because the number of packets for that specific service flow has reached its limit and is full.

3.6.1.1. Base Station Status Page in Receive Dual

When the user enables RX Dual on the Radio control page the second receive RSSI will be reported on the RSSI1 column on the Tx/Rx Over the Air Status section.


Tx/Rx Over the Air Status

Front Panel: Base Sync Rcv Xmt Alarm

Subscriber Name	Current Rx Modulation Type	Rx Signal Strength	RSSI0 (dBm)	RSSI1 (dBm)	Service Flow	State	Est. Tx User B/W (Mbits/s)	Est. Tx Allocated B/W (Mbits/s)	Service Flow Oversubscribed
NA	NA	NA	--	--	DL_MULTICAST	Active	0.00	0.00	0
SS-1	8 PPS 3/4	Excellent	-22	-21	BE_MAX	Active	0.00	2.65	0

Figure 3-2 BS Status GMSK RX Dual mode

Table 3-2. Base Station Status Page RX Dual Mode		
	Field/Control	**Description**
	Tx/Rx Over the Air Status	The first part of the Tx/Rx Status Table Displays the receive status of each subscriber connection. The second part of the table is the amount of data the BS is transmitting to each active SS. • Subscriber Name - Configured Subscriber's name. • Current Modulation Type – Contains the current modulation/FEC type for the subscriber connection. The display show the modulation of the receive frame at the time sampled. • Rx Signal Strength: Contains an assessment of the receive signal strength. • RSSI0 (dBm) – Display the RSSI value measured at the P1 antenna • RSSI1 (dBm) – Display the RSSI value measured at the P2 antenna • Service Flow – Identifies the service class of the service flow for the connection. • State – Displays the state of the connection. The valid values are provisioned and active. • Est User B/W (Mbits/sec) – Displays the estimated bandwidth in megabits per second being used by the connection in the transmit direction. • Est Allocated B/W (Mbits/sec) – Displays the estimated bandwidth in megabits per second being allocated to the connection in the transmit direction. • Service Flow Oversubscribed – Displays the number of Ethernet packets dropped because the number of packets for that specific service flow has reached its limit and is full.

If the OFDM base station is no longer receiving valid receive information for a connection, the row for that connection will be displayed in red, which could mean the data is stale. If the base

station is a GMSK base station the row will be displayed in red. For both modes, if the reception loss continues for a period of time the base station deregisters the subscriber and the row disappears from the display.

3.6.2. Base Station Admin Page

The admin page allows administration of station names for the base station and subscribers. It also provides the capability to enable and disable TRANSEC and the selection of the TRANSEC key. Start script generation and software download are also supported on the admin page.

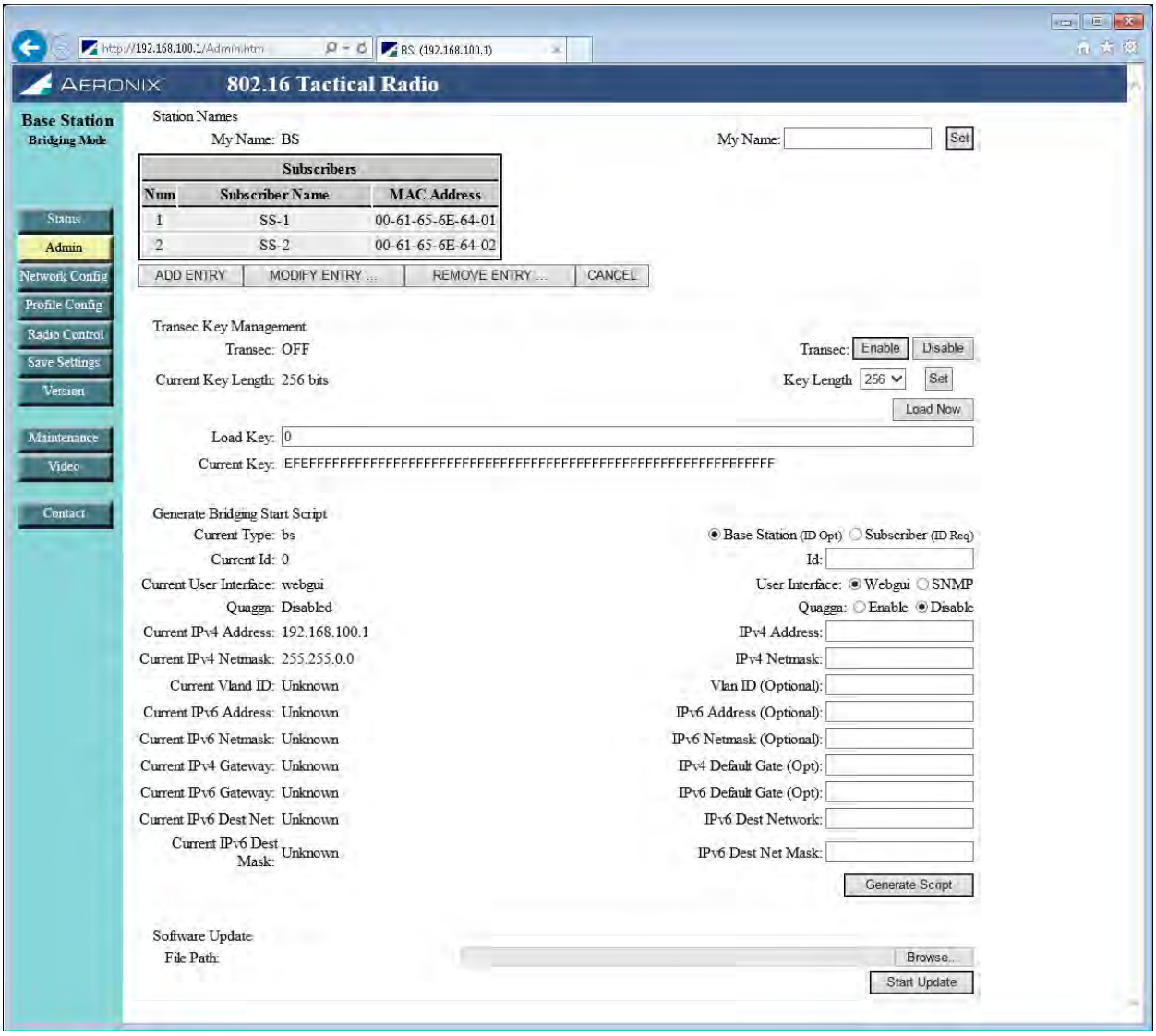


Figure 3-3 BS Admin Page

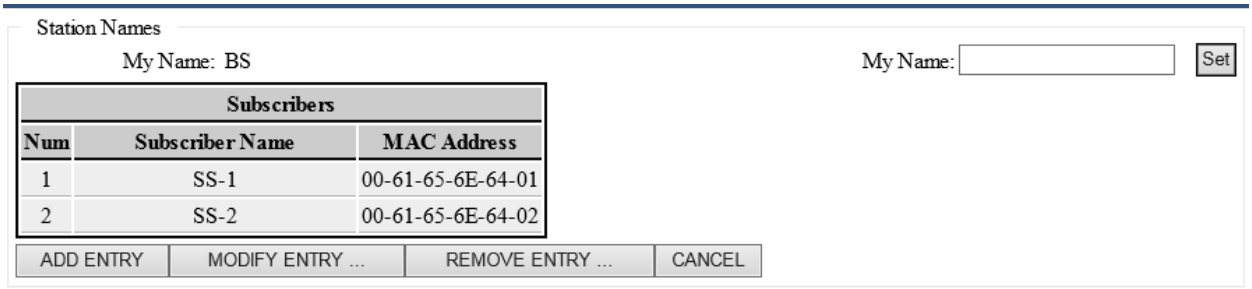


Figure 3-4 BS Admin Station Names

Table 3-3. BS Admin Page Station Names

	Field/Control	Description
1	My Name	Displays the current name of the base station.
2	My Name	Provides the capability to change or set the name of the base station. Type in the desired name and click the set button. The new name will automatically be distributed throughout the system
3	Subscribers	Relates a Subscriber's MAC Address with a user friendly name. • Number : Auto incrementing number. • Subscriber-Name: A unique name for the Subscriber, this is an alphanumeric string that will identify the SS. • MAC Address: MAC – the IEEE 802.16 MAC address of the subscriber station. The MAC address should be of the format 00-61-65-6E-64-XX where XX is the subscriber number in the system. The subscriber number is configured in the start script as part of the start script generation which is described later in this section of the document. The user can configure up to 32 subscribers for the network. • Add Entry Button – Add Subscriber to network • Modify Entry – Modify Subscriber Station entry • Remove Entry – Remove Subscriber Station Note: A subscriber's information must be added here first before moving on to adding a Provisioned Service Flow for it. Note: The BS will only allow a max of 10 Subscribers to be configured.

3.6.2.1. BS Admin Transec Key Management

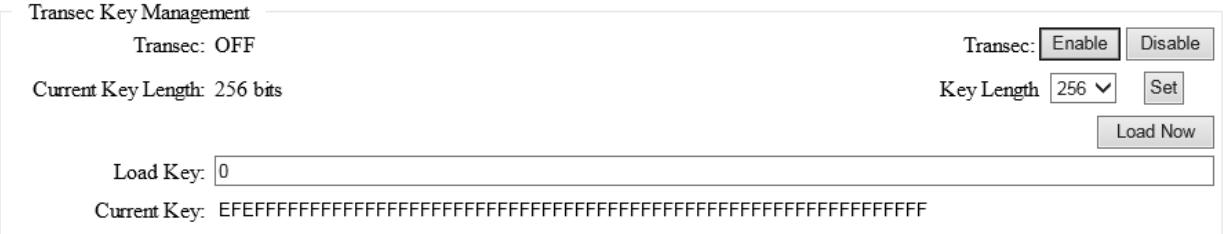


Figure 3-5 BS Admin Transec Key Mgmt

Table 3-4. BS Admin Transec Key Mgmt		
	Field/Control	**Description**
	Transec Key Management	Displays the settings for Transec capability. • Transec - Displays current Transec state (ON/OFF). • Transec Enable/Disable buttons - Ability to enable or disable. • Key Length drop down menu – Allows user to specify the number of bits for the Transec Key • Load Key – Allows the ability to change the Transec Key in hexadecimal based on Key length. • Current Key – Displays the current key.

3.6.2.2. BS Admin Bridging Start Script and Software Update

Bridging is the default mode for operating the units. The Generate Bridging Start Script section of the admin page allows for configuration of the start script. Once the configuration is set by selecting the Generate Script button the unit must be rebooted or power cycled for the configuration to take effect.

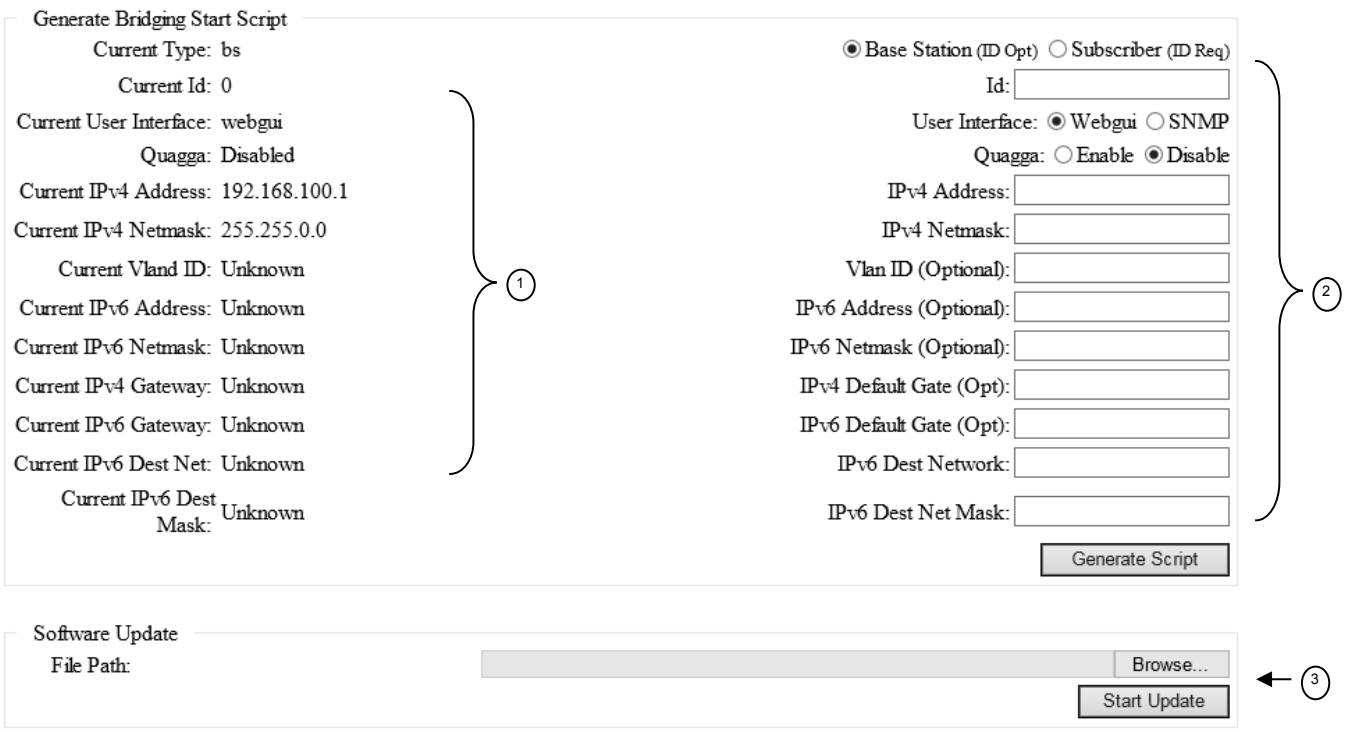


Figure 3-6 BS Bridging Script and S/W Update

Table 3-5. BS Admin Start Script and S/W Update

	Field/Control	Description
1	Current Start Script	The current start script parameters.
2	Generate Script	User can change the mode the radio boots in or the IP address,etc. The unit should not be powered cycled while script generation is in process. • Type radio buttons: Select BS or SS for type • Id: If setting up start script as SS enter its Subscriber Number. • User Interface radio buttons: NANO can be statistics and configuration can be monitored using either the webgui interface or the SNMP interface o Click on the radio button for the desired interface • Quagga radio buttons: Option to enable additional network routing software suite providing implementations of Open Shortest Path First (OSPF). Note there is user configuration necessary to implement Quagga, this information is not covered in this document.

Table 3-5. BS Admin Start Script and S/W Update		
	Field/Control	**Description**
		• IPv4 Address: Provides the capability to set a new IPv4 address for the unit. • IPv4 Netmask: Provides the capability to set a new IPv4 netmask address for the unit • IPv4 Default Gate (Opt) : Optional ability to configure an IPv4 default gateway. • Vlan ID (Optional) : Allows user to add a VLAN ID to IP packets coming from radio. The radio will pass VLAN tagged packets unmodified, so this does not need to be set to allow VLAN packets to flow through the radio. Only if communication to the radio needs to have a VLAN ID. • IPv6 Default Gate (Opt) : Optional ability to configure an IPv6 default gateway. • IPv6 Address (Optional) : Optional IPv6 address for the Ethernet (Note : current release of EDL does not support IPv6) • IPv6 Netmask (Optional) : Optional IPv6 netmask for IPv6 address (Note: current release of EDL does not support IPv6) • Generate Script Button : Generates a new start script for the unit with the values input in the "Generate Bridging Start Script" section of the Admin page. Once the script has been generated, the unit must be rebooted or power cycled for the new values to take effect. The unit should not be power cycled while script generation is in process.
3	Software Update	Allows user to select a signed package file provided by the manufacturer to update the software on the unit. • File Path – Allows users to browse to location where the package file resides. • Start Update - Selection of the start update button starts the software update using the factory signed package file specified in the File Path. Once the update process begins, any interruption can render the box unusable and require its return to the factory.

3.6.2.3. Example Adding a SS Entry

A subscriber must be added here in order to continue configuring that subscriber in network. Enter a name for the subscriber and then a IEEE 802.16 MAC address for the subscriber station. The MAC address should be of the format 00-61-65-6E-64-XX where XX is the subscriber number in the system. The subscriber number is configured in the start script as part of the subscriber start script generation which is described later in this section of the document.

The system will allow the user to configure 10 Subscribers.

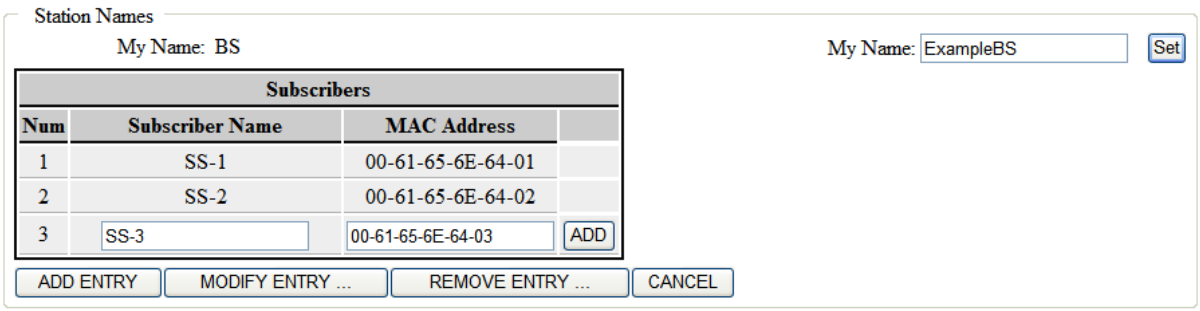


Figure 3-7 Example Adding a SS entry

3.6.2.4. Example Modify Transec Key

The following screen shows and example for updating a Transec key in the base station. The key is a 256 bit hexadecimal value. User must enter a new value in the 'Load Key' entry box. Then the user must click on the 'Load Now" Button.

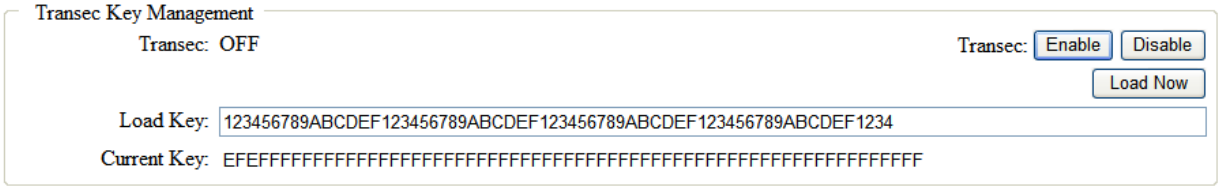


Figure 3-8 Example Modify Transec Key

Once the user click on the 'Load Now' button the Transec key will be updated.

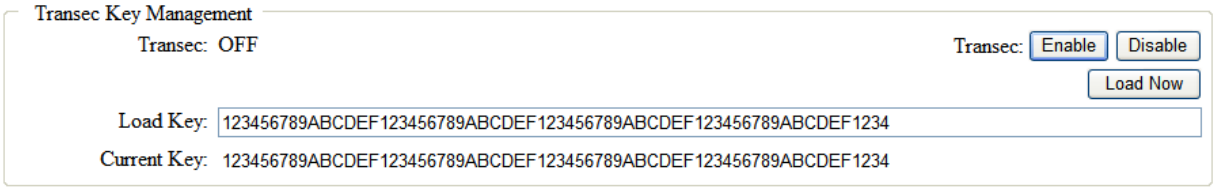


Figure 3-9 Example Modify Transec Key Result

In order for the Transec Key to be saved to the database for future use, the user must select the 'Save Settings' Button on the left menu (not shown in picture above, refer to 3.6.15.1 Save Settings Button).

3.6.2.5. Generate Bridging Start Script Example

Bridging is the default mode for operating the units. The Generate Bridging Start Script section of the admin page allows for configuration of the start script. Once the configuration is set by selecting the Generate Script button the unit must be rebooted or power cycled for the configuration to take effect. In this example below depicts new BS start script with an IPv4 address of 10.1.100.1.

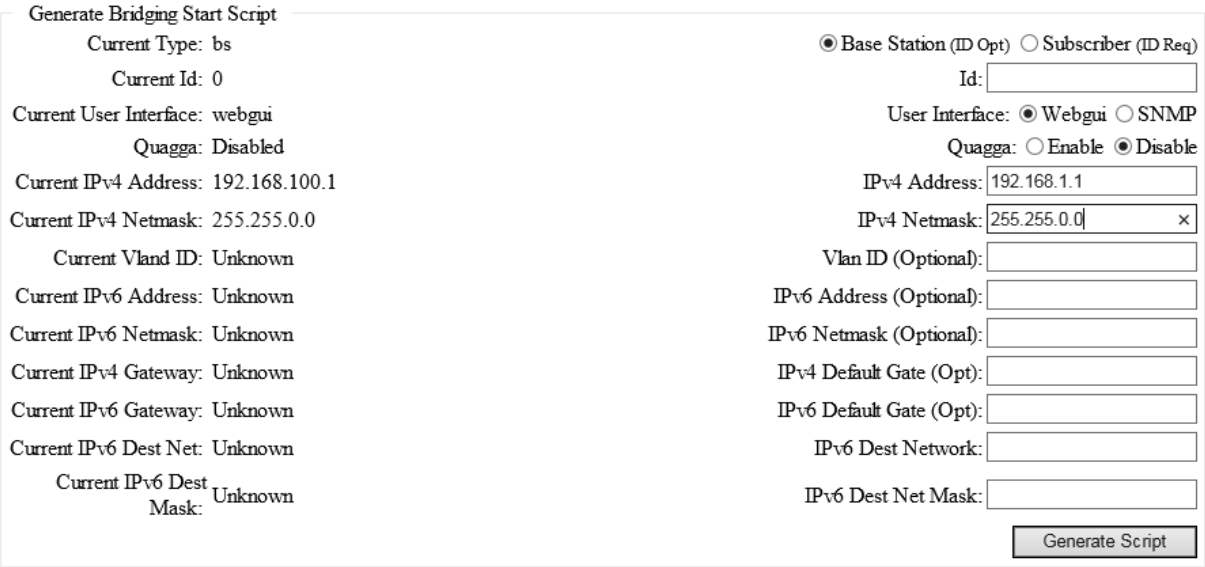


Figure 3-10 Example Generate Start Script

The left section of the Generate Bridging Start Script will display the current start script settings. The right section displays the changes the user desires to make to the start script. Once the 'Generate Script' button is selected, the old start script will be overwritten with the new start script changes. Once a reboot or power cycle occurs the new changes will take effect.

3.6.2.6. Software Update Example

The following screens show an example of the software update process. When the user selects the 'Browse' button a window will pop up allowing the user to locate the package file. A user browses to the location of the manufacturer signed package file and selects the Start Update button. After a few seconds the update process begins. This process should not be interrupted. When the process completes, the unit reboots and the new software executes. If problems occur during software update the error can be viewed via the webgui after reboot or via the serial console.

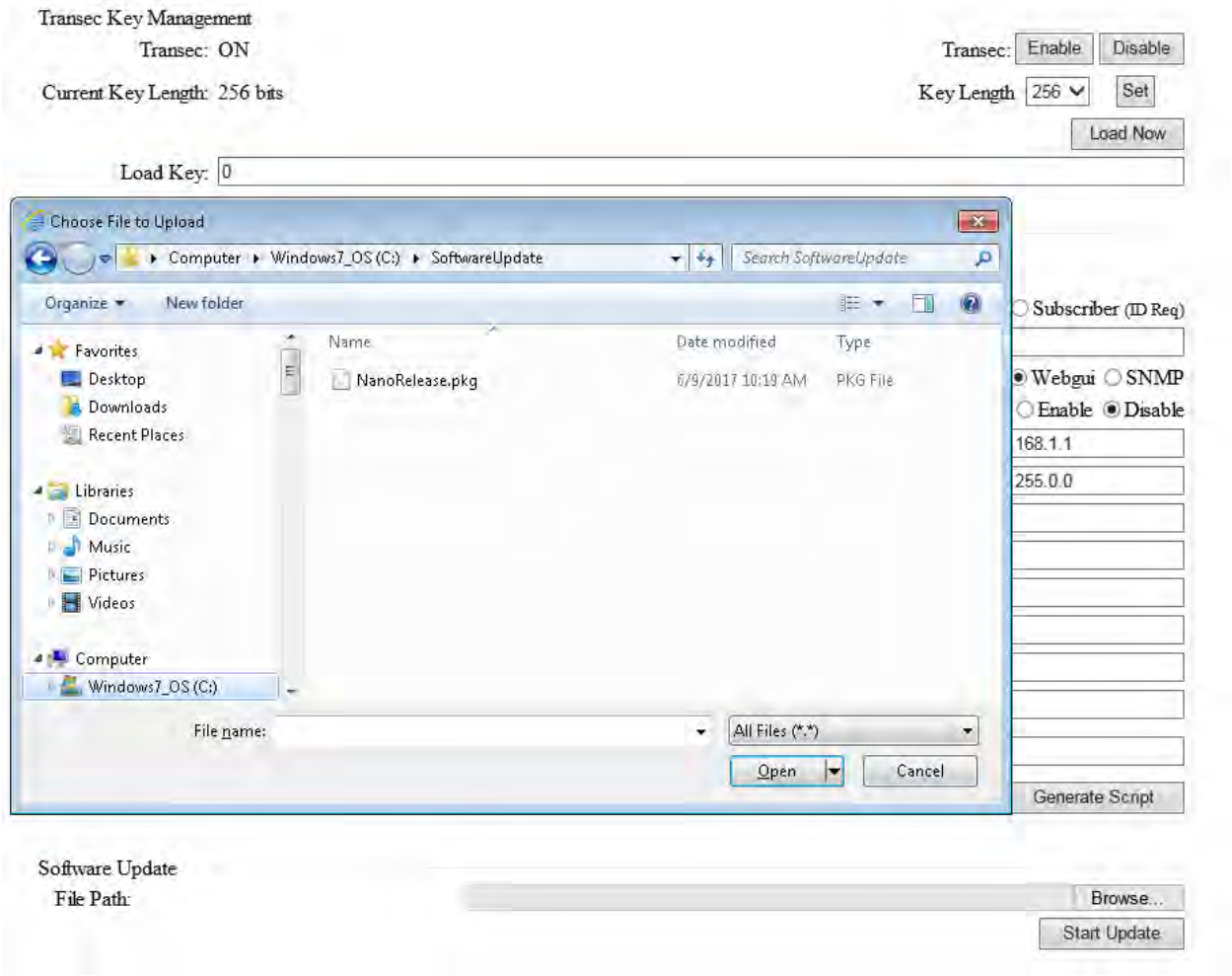


Figure 3-11 Example Software Update

The picture below displays the results when the 'Start Update' button is selected. The software update process begins and the GUI displays the software update page. This process should not be interrupted.

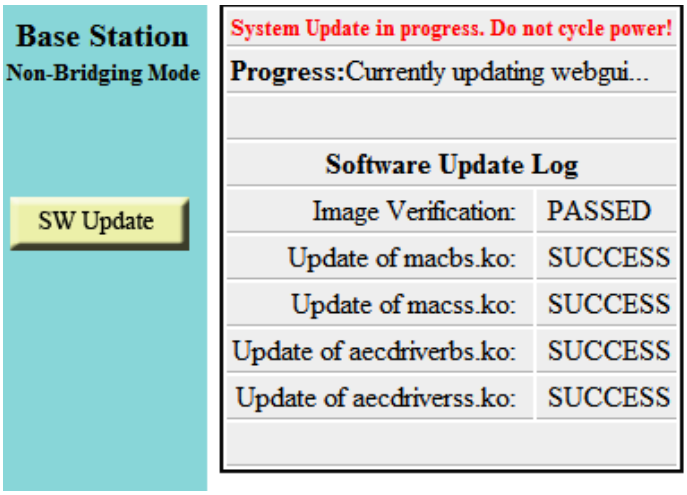


Figure 3-12 Example Software Update Progress

When the software update is complete the radio will reboots. When the radio completes startup, the Webgui will default back to the SW Update page and display the results of the software update.

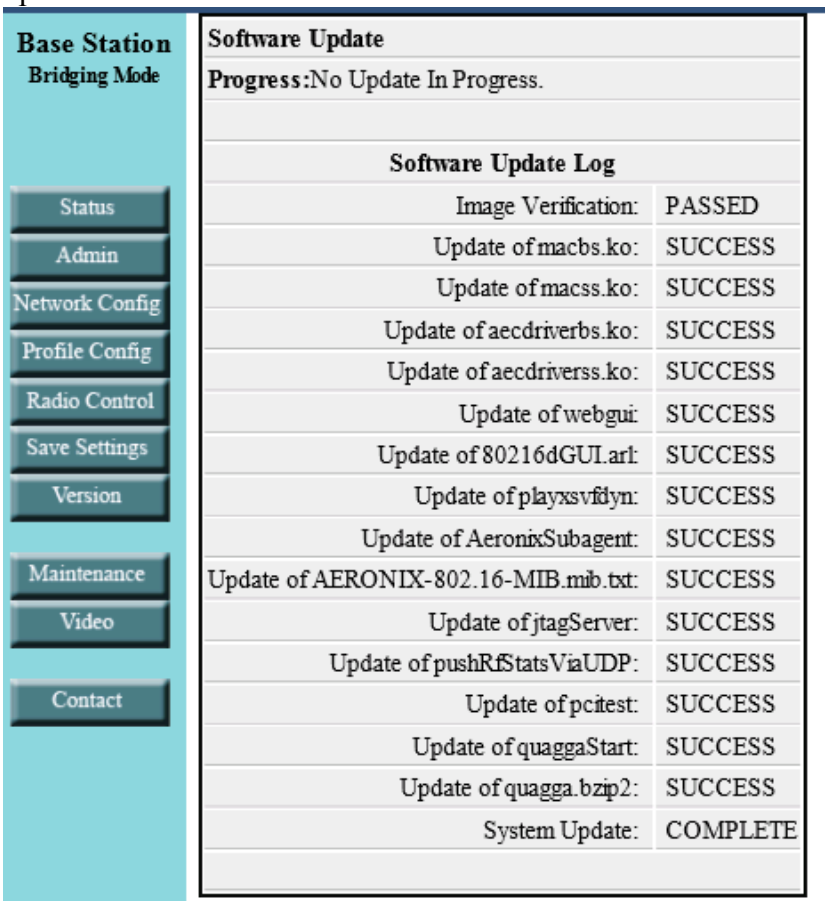


Figure 3-13 Software Update Result

3.6.3. Base Station Network Configuration Page

The base station network configuration screen allows the administrator to configure the 802.16d quality of service parameters and the traffic flow through network.

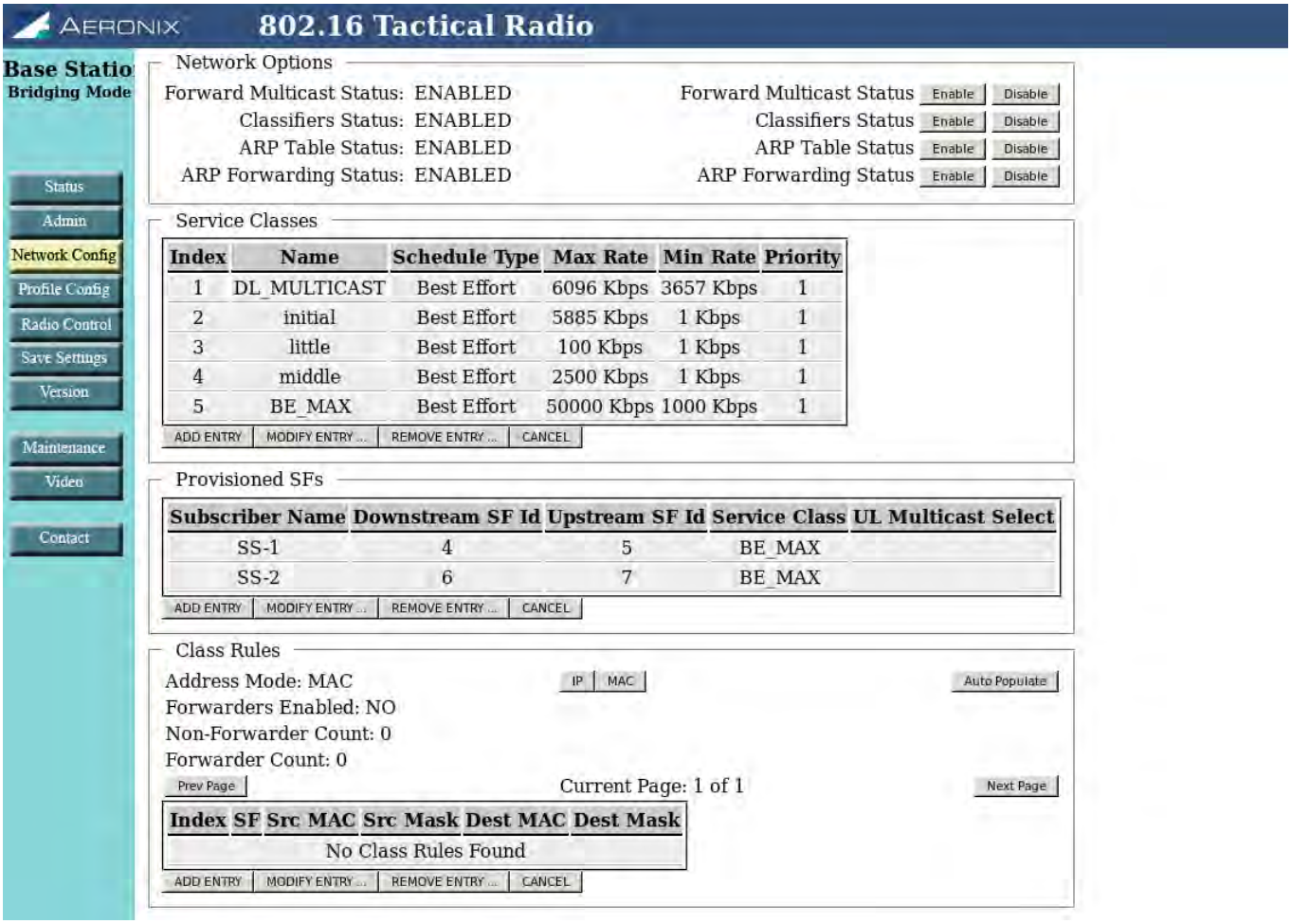


Figure 3-14 BS Network Config

3.6.3.1. Base Station Network Options

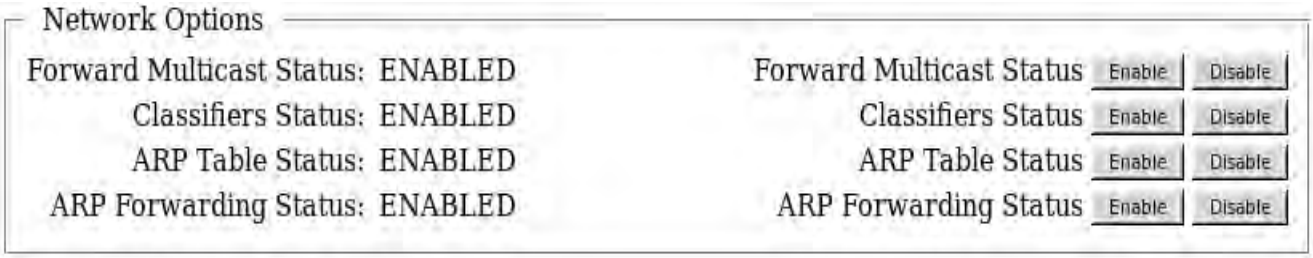


Figure 3-15 BS Network Config Network Options

Table 3-6. Base Station Network Config Network Options		
	Field/Control	**Description**
1	Network Options	• Forward Multicast Status : This allows the BS to re-direct Multicast messages from one Subscriber to remaining Subscribers through the BS RF interface. If this is disabled Multicast messages will not be forwarded from one SS to another. • Classifier Status : Disables or Enables using the Classifier Rules Table. See 3.6.3.7 Base Station Network Config Class Rules • ARP Table Status: Disables or Enables the ARP Table Feature. See 7.2 ARP Table feature • ARP Forwarding Status : Disables or Enables the ability for Subscribers to pass data between each other. This is only an option on the BS.

3.6.3.2. Base Station Network Config Service Classes

Service classes define the parameters for the 802.16d quality of services that are used to schedule the traffic throughout the network. A total of 12 service classes can be configured on the base station.

Service classes define the 802.16d scheduling type parameters used for scheduling the traffic throughout the network.

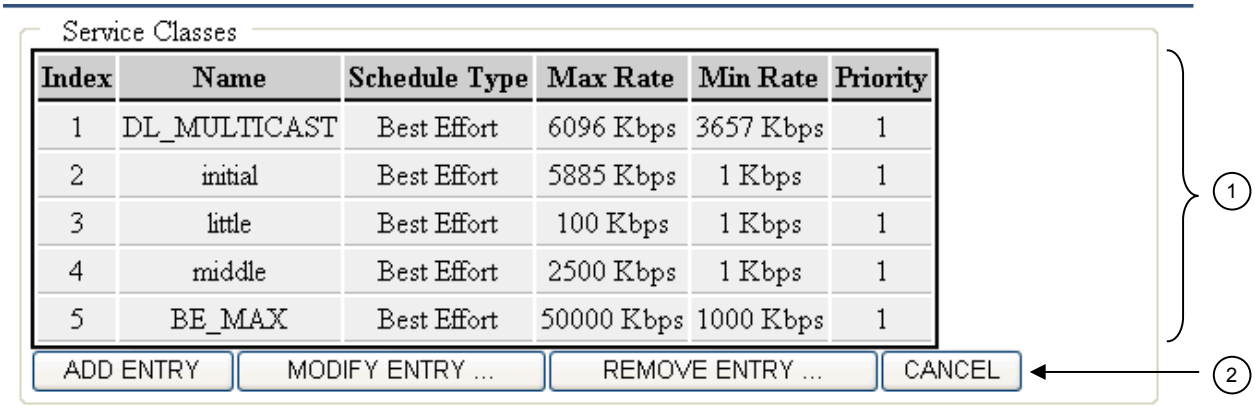


Figure 3-16 BS Network Config Service Classes

Table 3-7. Base Station Network Config Service Class		
	Field/Control	**Description**
1	Service Classes	• Index – the index into the service class table for the entry. This field is not entered by the user. • Name – the user defined name for the service class. The name can be up to 31 valid characters in length. • Schedule Type – Defines the 802.16d scheduling type for the service class. The entries are defined in a drop down by the system. • Max Rate – The maximum data rate for the service class in Kbps as defined by the 802.16d schedule type. • Min Rate – The minimum data rate for the service class in Kbps as defined by the 802.16d schedule type. • Priority - Defines the priority of the service class as defined by the 802.16d schedule type. The higher number defines a higher priority (7 is the highest priority). The priority field is valid for schedule types of non real-time and real-time.
2	Control Buttons	• Add Entry Button allows for adding a new service class entry. Click on the button and enter the fields. When the fields are complete click on the add button to the right of the entry fields. • Modify Entry Button allows for modifying a service class entry in the table. Click on the button and then select the entry to modify by clicking on the modify button to the right of the entry line. When modifications are complete click the save button to the right of the entry line. • Remove Entry Button allows for removing a service class entry from the table. Click on the button and then click on the remove button to the right of the entry line to remove. • Cancel Button allows for canceling the current add or modify request.

Multicast Note:

A service class for downlink multicast data is automatically configured that defines the possible bandwidth for multicast messages. The maximum and minimum rate can be modified according to the expected multicast traffic of your system. The multicast service class cannot be deleted as multicast support is always on in the bridging configuration. Multicast is not currently supported in the non-bridging configuration.

- **Adding an Entry for a Service Class:**

Service classes define the parameters for the 802.16d quality of services that are used to schedule the traffic through the network. A total of 12 service classes can be configured on the base station.

The following screen displays an example for adding a service class. Once the ADD ENTRY button is selected the administrator can type in a name for the new service class and can select the schedule type from a pull down menu. The max and min data rates should be entered and a priority selected if valid for the chosen scheduling type. Once all of the information is input the administrator selects the ADD button to enter the information input.

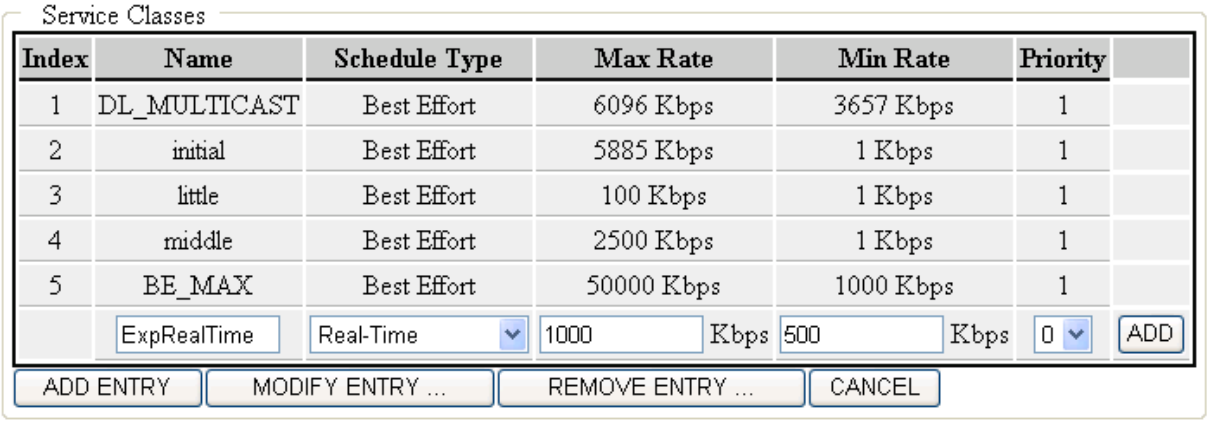


Figure 3-17 BS Example Add Service Class

The above example adds a Real-Time service class named "ExpRealTime" with a maximum rate of 1000Kbps and a minimum rate of 500 kbps, at a priory of 0.

3.6.3.3. Modifying a Service Class

The following screen displays an example for modifying an existing service class entry. Once the administrator selects the MODIFY ENTRY button the max and min data rate fields and the priority fields can be modified. Once the new values are entered the administrator selects the SAVE button to the right of the entry to save the changes. To change the schedule type and service class name, the administrator adds another service class entry with the new parameters or deletes a current entry and adds a service class entry with the new parameters.

Service Classes

Index	Name	Schedule Type	Max Rate	Min Rate	Priority	
1	BE_MULTICAST	Best Effort	6096 Kbps	3657 Kbps	1	
2	initial	Best Effort	5885 Kbps	1 Kbps	1	
3	little	Best Effort	100 Kbps	1 Kbps	1	
4	middle	Best Effort	2500 Kbps	1 Kbps	1	
5	BE_MAX	Best Effort	50000 Kbps	1000 Kbps	1	
6	ExpRealTime	Real-Time	5000 Kbps	600 Kbps	0	SAVE

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Figure 3-18 BS Example Modify Service Class

3.6.3.4. Base Station Network Config Provisioned SFs

The service flows define the subscriber connections that are allowable within the network. They also define the service class of the subscriber connection which is used to determine the data traffic and rate of data flow for that connection. A total of 32 pairs of service flows can be defined in the network. It is allowable to define multiple pairs of service flows per subscriber when there are service flow entries available.

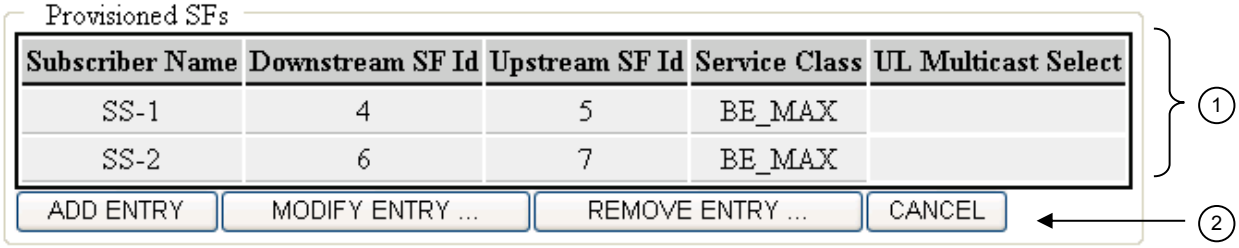


Figure 3-19 BS Provisioned SFs

Table 3-8. Base Station Network Config Service Class		
	Field/Control	**Description**
1	Provisioned Service Flows	• Subscriber Name – A drop down selection box with the alphanumeric names for all of the configured Subscribers on the Admin page. • Downstream SF Id - Defines the identification number for the downstream service flow (base station to subscriber). The values of 1,2,3 are reserved. • Upstream SF Id – Defines the identification number for the upstream service flow (subscriber to base station).

Table 3-8. Base Station Network Config Service Class		
	Field/Control	**Description**
		Not a user entry, this value will be Downstream SF Id + 1 • Service Class – A drop down selection for the service class used for scheduling traffic for the service flow pair. • UL Multicast Select – Defines the service flow from which the uplink data will be sent on. This is only critical if the user has configured multiple Service Flows for a subscriber station. Otherwise the only service flow defined for the Subscriber will be used.
2	Control Buttons	• Add Entry Button allows for adding a new provisioned service flow entry. Click on the button and enter the fields. When the fields are complete click on the add button to the right of the entry fields. • Modify Entry Button allows for modifying a provisioned service flow entry in the table. Click on the button and then select the entry to modify by clicking on the modify button to the right of the entry line. When modifications are complete click the save button to the right of the entry line. • Remove Entry Button allows for removing a provisioned service flow entry from the table. Click on the button and then click on the remove button to the right of the entry line to remove. • Cancel Button allows for canceling the current add or modify request Note: The GUI adds both downlink and uplink SFs at the same time.

3.6.3.5. Generating Service Flows

Service flow pairs can be added for a subscriber in the Network Config page. Only subscribersconfigured to the network in the station name portion of the Admin Page appear in the drop drown list for the subscriber name selection of the service flow. The subscriber that the service flow pair represents is selected. The administrator provides a service flow identification number for the downstream service flow which should be a number greater than 4 and not be used for any other service flow id. The upstream identification number is generated automatically. The service identification numbers are only used for reference.

The service class drop down list contains all service classes defined in the service class section of the Network Config page. The administrator selects the desired service class. The UL Multicast Select box, when checked, indicates to the system which service flow will have the uplink data sent on. Each subscriber can have a maximum of one UL Multicast selected to it. The UL Multicast is optional; however, if there are multiple service flows defined for a subscriber and there is no UL Multicast selected, the system will automatically pick one at random. Once all the fields are complete, the administrator selects the ADD button to the right of the new entry, adding the new service flow pair to the network.

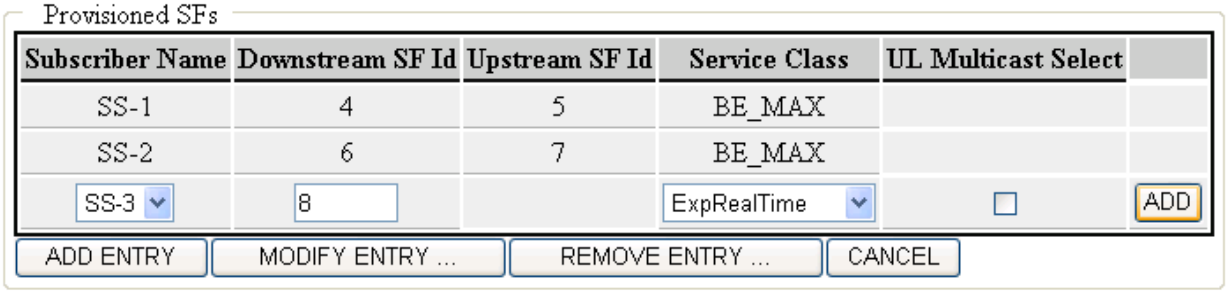


Figure 3-20 BS Example Add Provisioned SFs

3.6.3.6. Modifying a Service Flow Pair

Existing service flow pair entries can be modified using the MODIFY ENTRY button. The administrator can only modify the service class for the service flow pair to a service class of the same schedule type. Changing any of the other service flow parameters requires adding a new service flow pair with the desired parameters. If a subscriber station already has a UL Multicast checked for one of its service flows, none of its other service flows will be allowed to have their Multicast also selected. To select the status of the UL Multicast for a specific subscriber's service flow, any service flow already set to use the UL Multicast must be unchecked first. Any undesired service flow pairs should be removed.

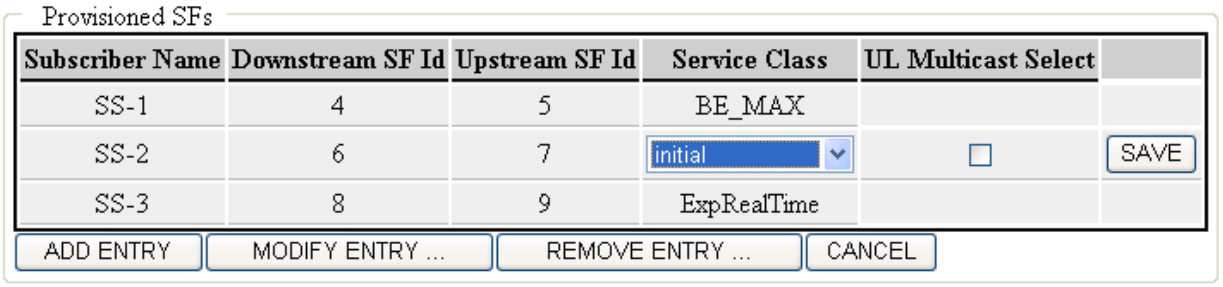


Figure 3-21 BS Example Modify Service Flow

3.6.3.7. Base Station Network Config Class Rules

If user is not using the ARP Table feature, then they will need to configure classifiers for IP data to pass between the radios. The Classifier Status must be enabled in the Network Options section of the Network Config page.

The class rules define the paths that are valid for data transmission within the network. It is valid to define class rules by using IP addresses or MAC addresses. All class rules must be defined using the same address type. Class Rules are configured on the base station and transferred to the subscribers when they connect to the base station. A valid class rule must exist for each downstream service flow and for each upstream service flow for data to pass in the define direction.

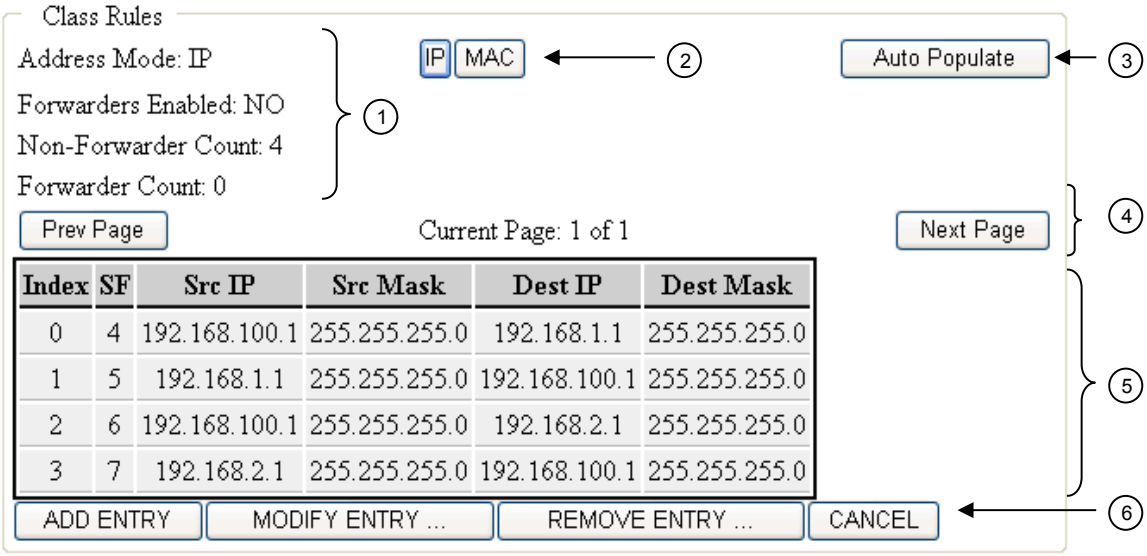


Figure 3-22 BS Network Config Class Rules

Table 3-9. Base Station Network Config Service Class		
	Field/Control	**Description**
3	Current Class Rules Mode	• Address Mode - Displays the current address mode for the class rules. The valid values are IP and MAC • Forwarders Enabled - If selected, the GUI will populate forwarders for each configured SS. Auto populate does not account for multiple service flows per subscriber. The additional class rules need to be entered manually. • Non-Forwarder Count - Indicates the number of class rules not using forwarders.

Table 3-9. Base Station Network Config Service Class		
	Field/Control	**Description**
		• Forwarder count - Indicates the number of class rules using forwarders.
2	IP/MAC Buttons	Provides the capability to change the address mode of the class rules. Changing the mode will delete the current class rules. The administrator receives a warning prior to rule deletion.
3	Auto Populate	Provides the capability to automate the class rule entry process such that each address gets entered once. Auto populate creates a class rule for the downstream service flow and a class rule for the upstream service flow.
4	Prev/Next Page	Allows for paging through the class rules if the information on the Network Config is larger than one page.
5	Classifier Table	IP addresses in this section are just examples. • Index – Internal index used for class rules. The index is automatically generated. • SF – Identifies the service flow that pertains to the class rule. • Src IP/MAC – The source address for the class rule. The address presented to the unit when initiating traffic through it. • Src Mask – The address mask for the source address. Masking portions of the IP address is only valid when the addresses of the units are configured on different subnets. Currently only the last subnet byte is allowed to be masked • Dest IP/MAC - The destination address for the class rule. The destination address for the traffic through the unit. • Dest Mask - The address mask for the destination address. Masking portions of the IP address is only valid when the addresses of the units are configured on different subnets. Currently only the last subnet byte is allowed to be masked
6	Control Buttons	The control buttons allow the administrator to add, modify and remove class rules and to cancel the current add, modify or remove transaction.

3.6.3.8. Base Station Network Configuration Auto Populate GUI

Class rules can be auto populated. The Aeronix S/W will look at either the IP Layer or the Ethernet layer of the Ethernet packet, depending if the classifiers are configured in IP mode or MAC mode. The addresses inputted should be those presented to the unit in bridging mode. They are typically the address of the peripherals generating the data for the source address and the peripheral receiving the data for the destination address

The screen below appears when the user clicks on the Auto Populate Button on the Base Station Network Configuration Screen.

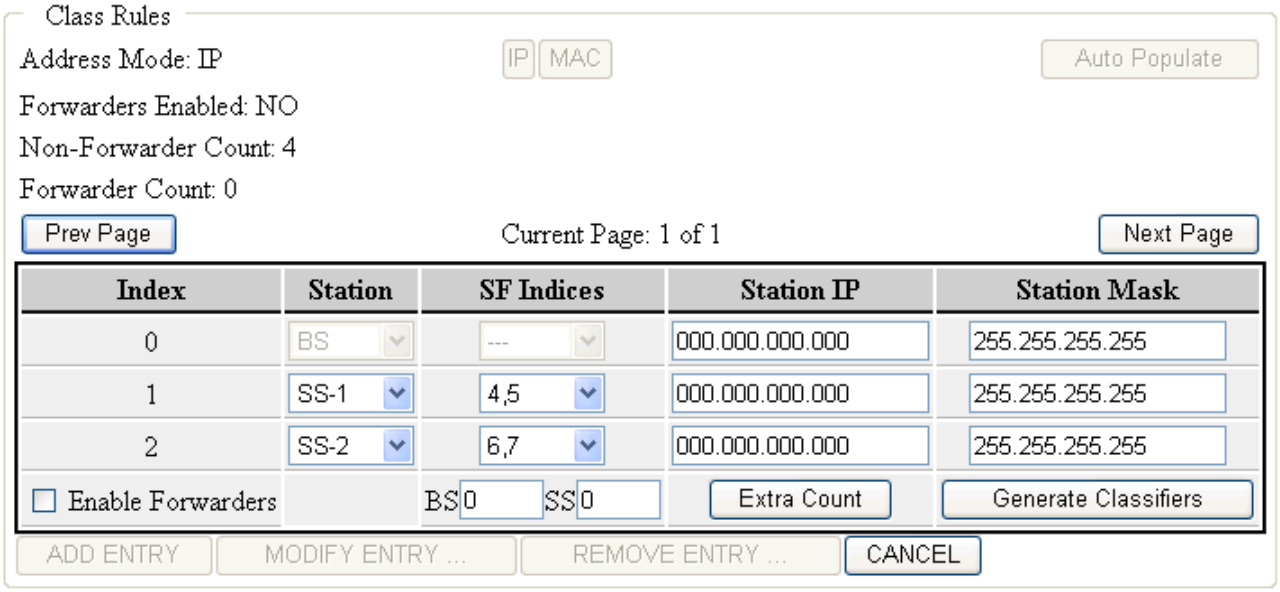


Figure 3-23 BS Network Config Auto Populate

Table 3-10. Base Station Network Config Auto Populate

	Field/Control	Description
1	Auto Populate User Interface	Automatically generates classifiers based on the list of Subscribers that are configured. User must enter the IP/MAC address of the data from each Station. • Station – A list of all the stations (including BS) that are configured, not a user entry. • SF Indices – The SF index that is associated with a configured SS, not a user entry (note: BS does not have SF Id so it is "---"). • Station IP - The source external network station's Ethernet IP address. This is the source of data packets. • Station Mask - Mask to be applied to Source IP fields, allows you to open up a range of IP addresses. Currently only the last subnet byte is allowed to be masked off. • Enable Forwarders check box - If selected, the GUI will populate forwarders for each configured SS.

Table 3-10. Base Station Network Config Auto Populate		
	Field/Control	**Description**
		• BS – Number of extra classifiers that will be added to the base station. • SS – Total number of extra classifiers that will be added to the subscriber stations. • Extra Count Button – Allows a user to add additional classifiers to a radio. • Generate Classifiers – This button triggers the GUI to generate the classifiers. It will create BS ↔ SS classifiers, and forwarders if enabled.

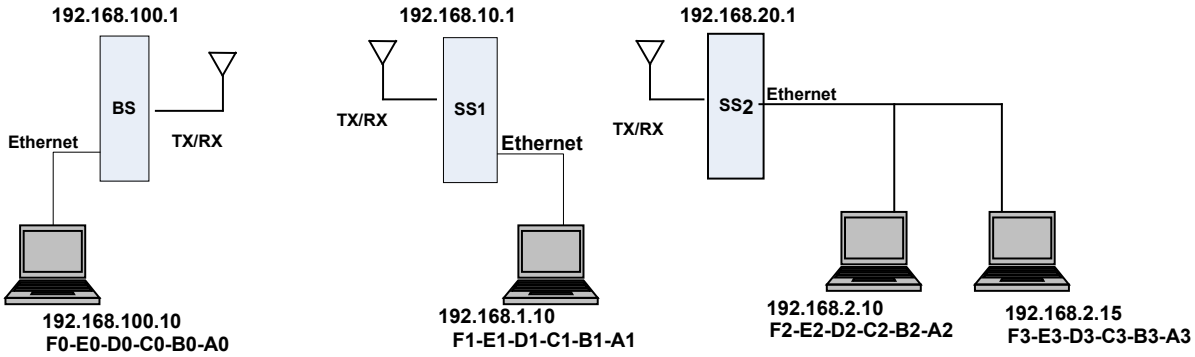


Figure 3-24 Example Auto Populate Network Layout

3.6.3.9. Base Station Network Configuration Auto Populate Example

The following screens provide an example of auto populating the class rules when the address type is set to IP. The user must select the 'Auto Populate' button on the Network Config page to get to the auto populate screen. The IP addresses input should be those presented to the unit in bridging mode. They are typically the address of the peripherals generating the data for the source address and the peripheral receiving the data for the destination address. In the case of the example network configuration above, the BS Station IP would be 192.168.100.0 with a mask of 255.255.255.0, SS-1 Station IP would be 192.168.10.0 with a mask of 255.255.255.0 and SS-2 Station IP would be 192.168.20.0 with a mask of 255.255.255.0. Since the networks are on different subnets the user can mask the last byte. When possible, masking the addresses reduce the amount of class rules generated and allow for the use of numerous peripherals on the units subnet.

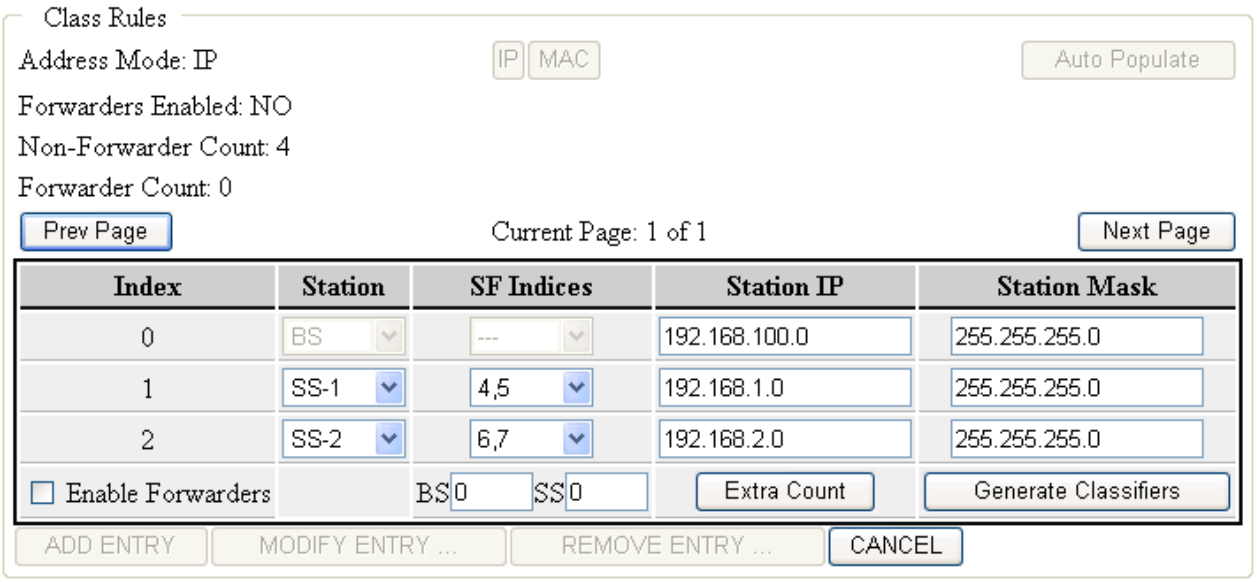


Figure 3-25 BS Example Auto Populate

The administrator selects the Generate Classifiers button after the IP addresses and masks are entered. The class rule generation process begins and should not be interrupted until complete. Figure 3-26 BS Example Auto Populate Result shows the rules after generation is complete without forwarders.

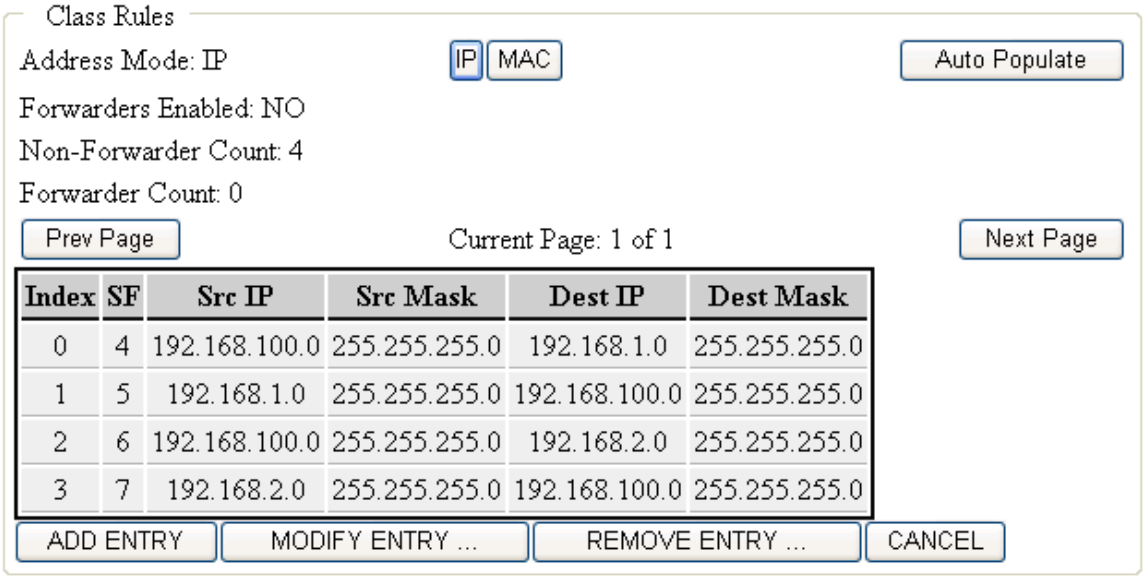


Figure 3-26 BS Example Auto Populate Result

3.6.3.10. Adding Additional Class Rules to Stations using Auto Populate

Additional classifiers can be added for each station. It is best to determine how many entries will be required before populating the table. Once the Auto Populate button is selected, the number of additional base station and subscriber station classifiers should be entered in the BS and SS text fields respectively. The Extra Count button is then selected to confirm the additional classifiers and the table will expand. Once the desired number of classifiers is added to the table, the station, SF indices, station IP and station masks can be selected/entered. The station and the service flow index per entry must match one of the entries in the Provisioned SF's table. Once the table is filled out, the Generate Classifiers button is selected to create the class rules. This process should not be interrupted until complete.

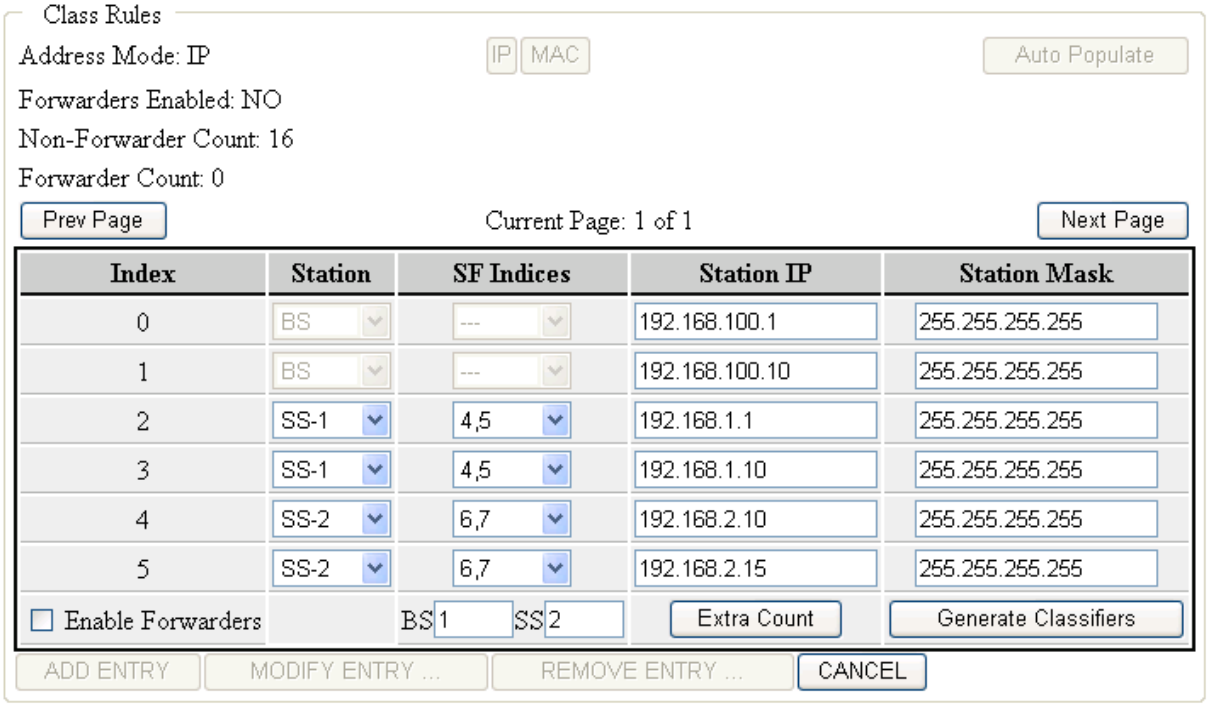


Figure 3-27 BS Example Auto Populate Additional Class Rules

.

Class Rules

Address Mode: IP IP | MAC Auto Populate

Forwarders Enabled: NO

Non-Forwarder Count: 16

Forwarder Count: 0

Prev Page Current Page: 1 of 1 Next Page

Index	SF	Src IP	Src Mask	Dest IP	Dest Mask
0	4	192.168.100.1	255.255.255.255	192.168.1.1	255.255.255.255
1	5	192.168.1.1	255.255.255.255	192.168.100.1	255.255.255.255
2	4	192.168.100.1	255.255.255.255	192.168.1.10	255.255.255.255
3	5	192.168.1.10	255.255.255.255	192.168.100.1	255.255.255.255
4	6	192.168.100.1	255.255.255.255	192.168.2.10	255.255.255.255
5	7	192.168.2.10	255.255.255.255	192.168.100.1	255.255.255.255
6	6	192.168.100.1	255.255.255.255	192.168.2.15	255.255.255.255
7	7	192.168.2.15	255.255.255.255	192.168.100.1	255.255.255.255
8	4	192.168.100.10	255.255.255.255	192.168.1.1	255.255.255.255
9	5	192.168.1.1	255.255.255.255	192.168.100.10	255.255.255.255
10	4	192.168.100.10	255.255.255.255	192.168.1.10	255.255.255.255
11	5	192.168.1.10	255.255.255.255	192.168.100.10	255.255.255.255
12	6	192.168.100.10	255.255.255.255	192.168.2.10	255.255.255.255
13	7	192.168.2.10	255.255.255.255	192.168.100.10	255.255.255.255
14	6	192.168.100.10	255.255.255.255	192.168.2.15	255.255.255.255
15	7	192.168.2.15	255.255.255.255	192.168.100.10	255.255.255.255

ADD ENTRY MODIFY ENTRY ... REMOVE ENTRY ... CANCEL

Figure 3-28 BS Example Additional Class Rules Result

3.6.3.11. Auto Populating Class rules of Type IP while Enabling Forwarders

Forwarders provide the capability for subscribers to talk to each other via the base station. They define data traffic from a subscriber that should be forwarded to another subscriber and used in the downlink direction only. It requires additional configuration of class rules and if data transmission occurs from subscriber to subscriber it will consume bandwidth allocated for those subscribers. Forwarding classifiers must be consistent with non-forwarding classifiers. For example, if using IP masks with the non-forwarders, one must also use the same mask with the

forwarders. Forwarders can be enabled or disabled when selecting the auto populate. They are generated and applied to the entire Class Rules table.

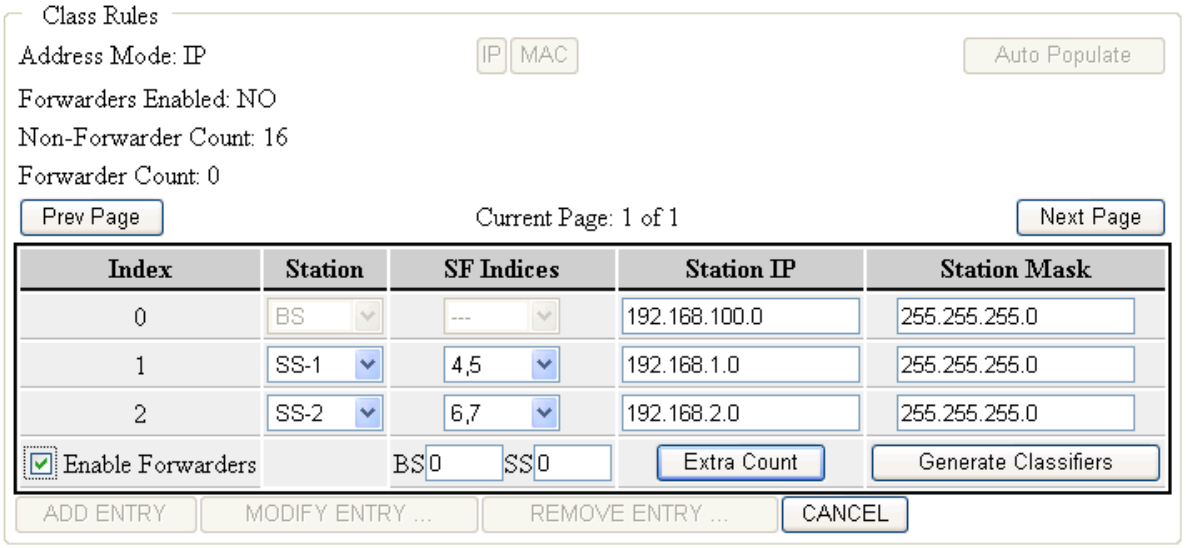


Figure 3-29 BS Example Auto Populate IP with Forwarders

Below is what is displayed after user hits the "Generate Classifiers" button. The forwarder class rules are not displayed, but the "Forwarder Count" has incremented for each forwarder created.

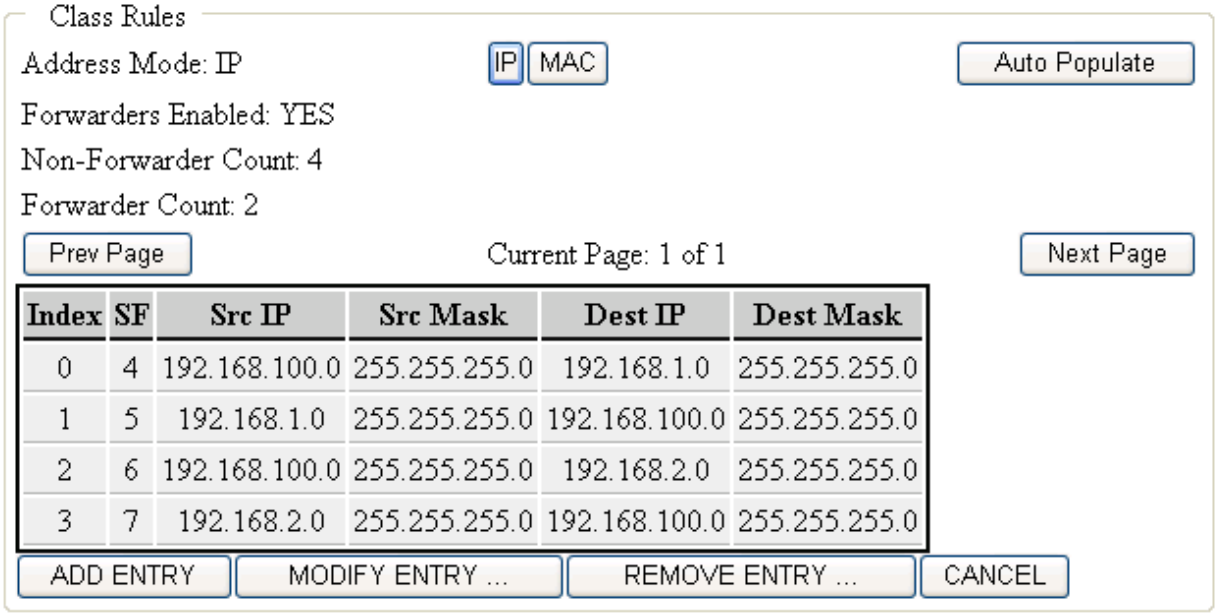


Figure 3-30 BS Example Auto Populate IP with Forwarders Result

3.6.3.12. Auto Populating Class Rules of Type MAC

Auto populating the class rules can also be done when the address type is set to MAC. The MAC addresses input should be those presented to the unit in bridging mode. They are typically the address of the peripherals generating the data for the source address and the peripheral receiving the data for the destination address. In the case of the example network configuration in **Error! Reference source not found.** the BS Station MAC would be F0-E0-D0-C0-B0-A0, S-1 Station MAC would be F1-E1-D1-C1-B1-A1. SS-2 Station has 2 laptops connected to it so the user needs to add an "Extra Count" of 1 for SS2. SS MACs would be F2-E2-D2-C2-B2-A2 and F3-E3-D3-C3-B3-A3.

The administrator selects the Generate Classifiers button after the MAC addresses and masks are entered. The class rule generation process begins and should not be interrupted until complete.

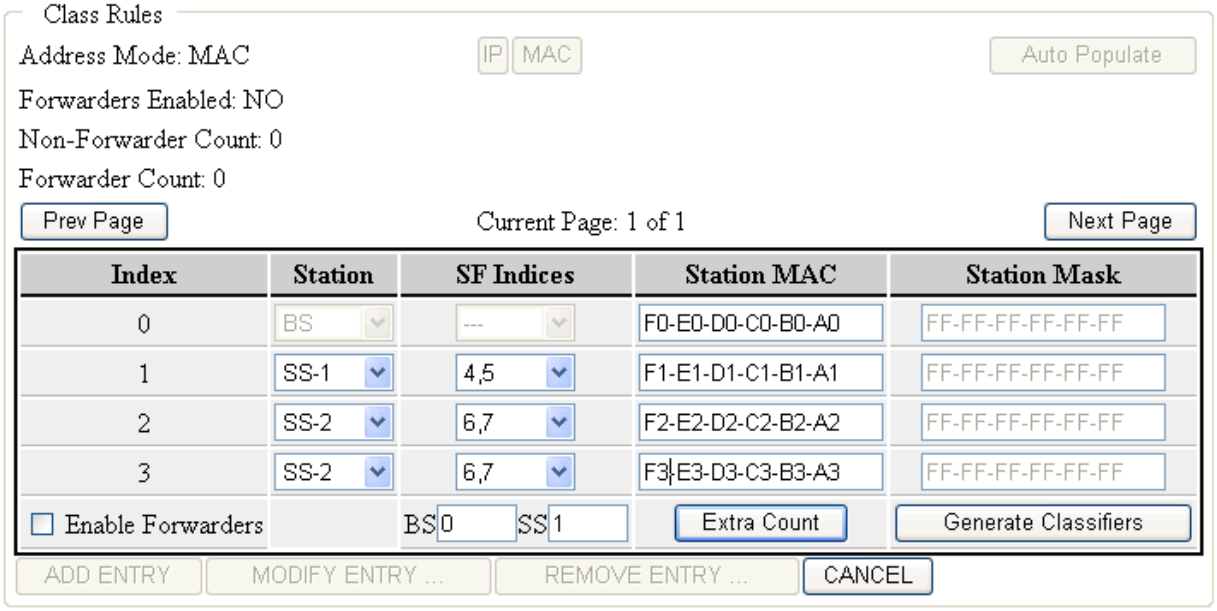


Figure 3-31 BS Example Auto Populate with MAC Address

Below is the result of the user selecting the "Generate Classifiers" button.

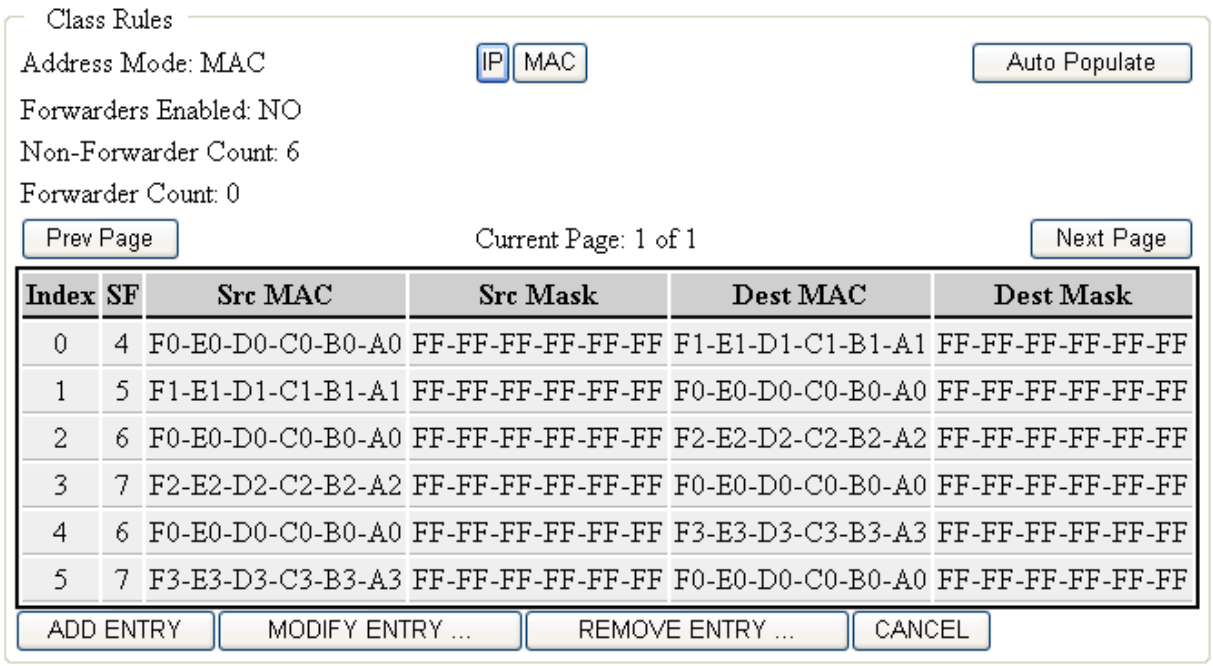


Figure 3-32 BS Example Auto Populate with MAC Address Result

3.6.3.13. Adding Class Rule using ADD ENTRY

The following screen provides an example of the ADD ENTRY functionality for a class rule. The user can add a class rule entry by selecting the ADD ENTRY button within the Class Rule section of the Network Config page (Figure 3-22 BS Network Config Class Rules). Once that occurs a similar screen to Figure 3-33 BS Example Class Rules ADD ENTRY appears. The administrator selects the service flow for which the rule pertains from the pull down menu. The administrator then inputs the valid IP/MAC source address and mask and the valid destination address and mask. Select the ADD button to the right of the entry to input the new class rule. The rule distributes through the network as needed.

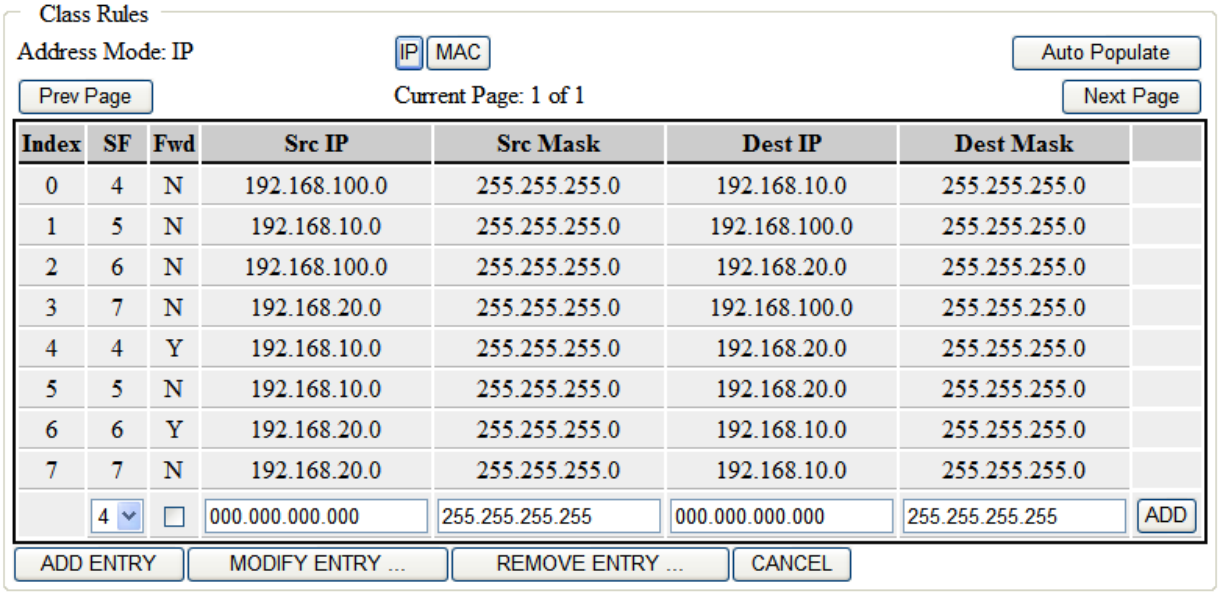


Figure 3-33 BS Example Class Rules ADD ENTRY

3.6.3.14. Modifying a Class Rule Using MODIFY Entry

The user can modify a class rule entry by selecting the MODIFY ENTRY button within the Class Rule section of the Network Config page (Figure 3-22 BS Network Config Class Rules). The MODIFY button to the right of the entry is selected to modify that entry (Figure 3-34 BS Example MODIFY ENTRY selection). The user may change the source and destination address for that class rule (**Error! Reference source not found.**). The SAVE button to the right of the ntry is selected to input the new class rule data. The Forwarder Enabled status of the table can be changed by selecting the Auto Populate button and then checking or unchecking the Enable Forwarders box. The rule distributes throughout the network as needed.

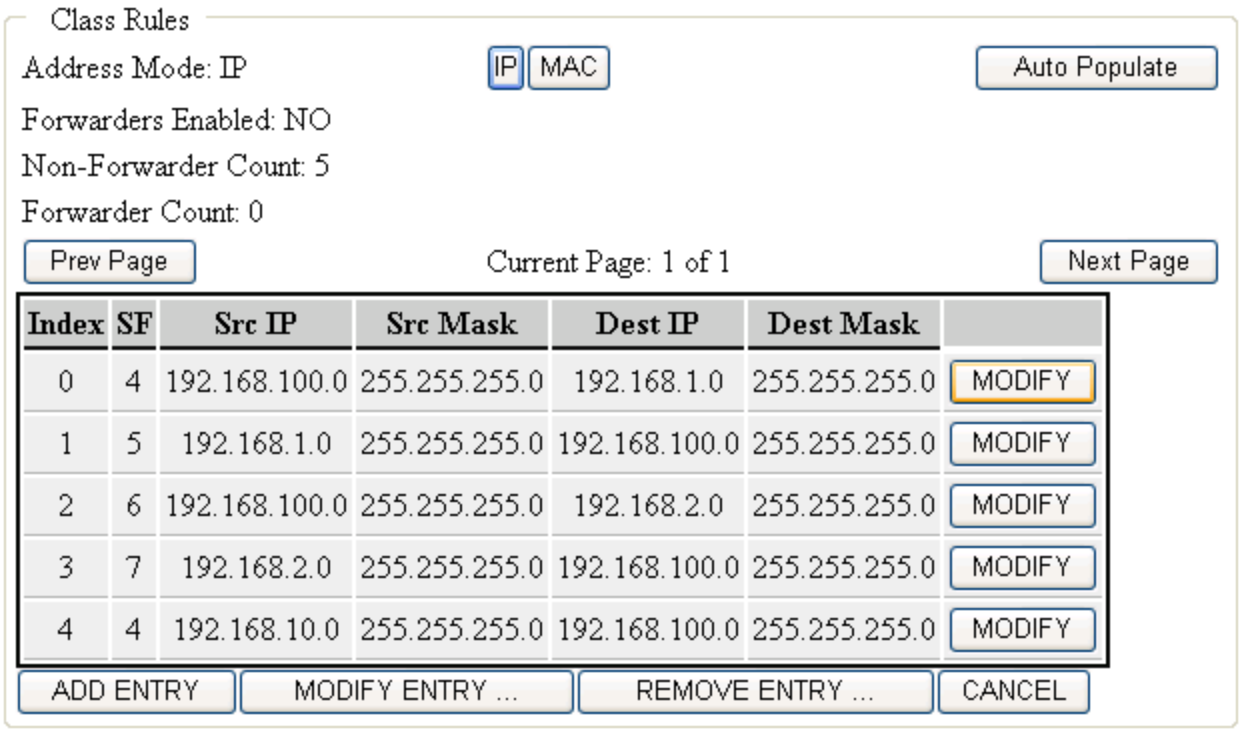


Figure 3-34 BS Example MODIFY ENTRY selection

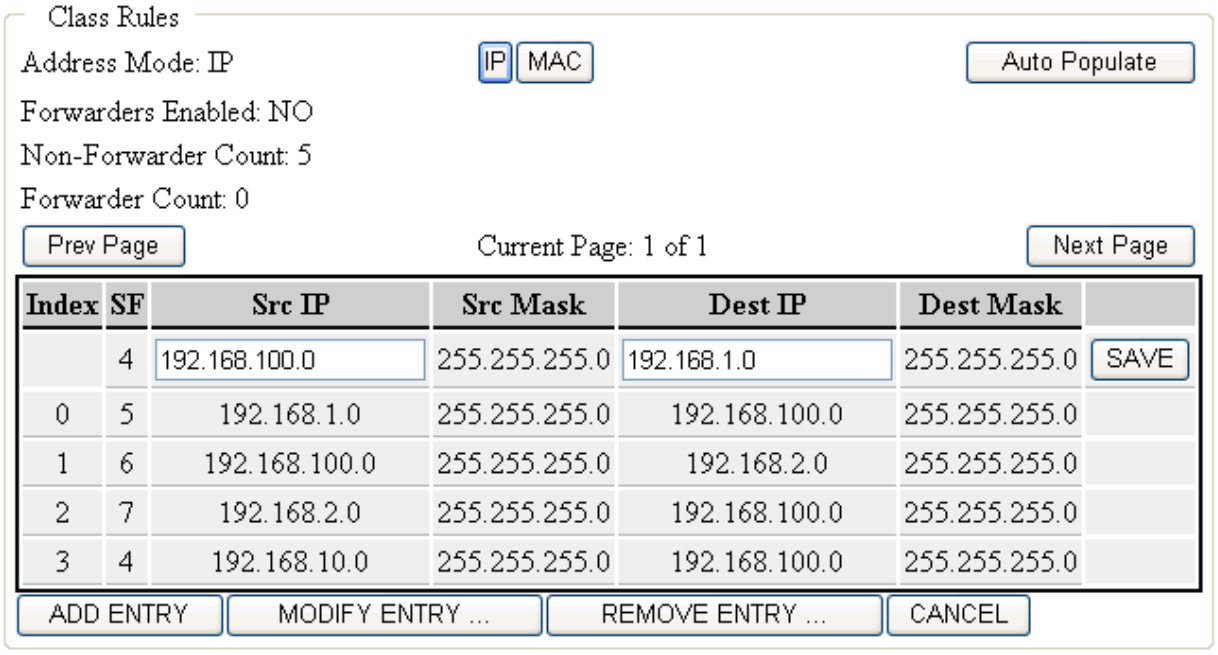


Figure 3-35 BS Example MODIFY ENTRY

3.6.4. Base Station Profile Config Page

The Profile Config page allows the administrator to configure the parameters which control the automatic change modulation (ACM) capability within the network.

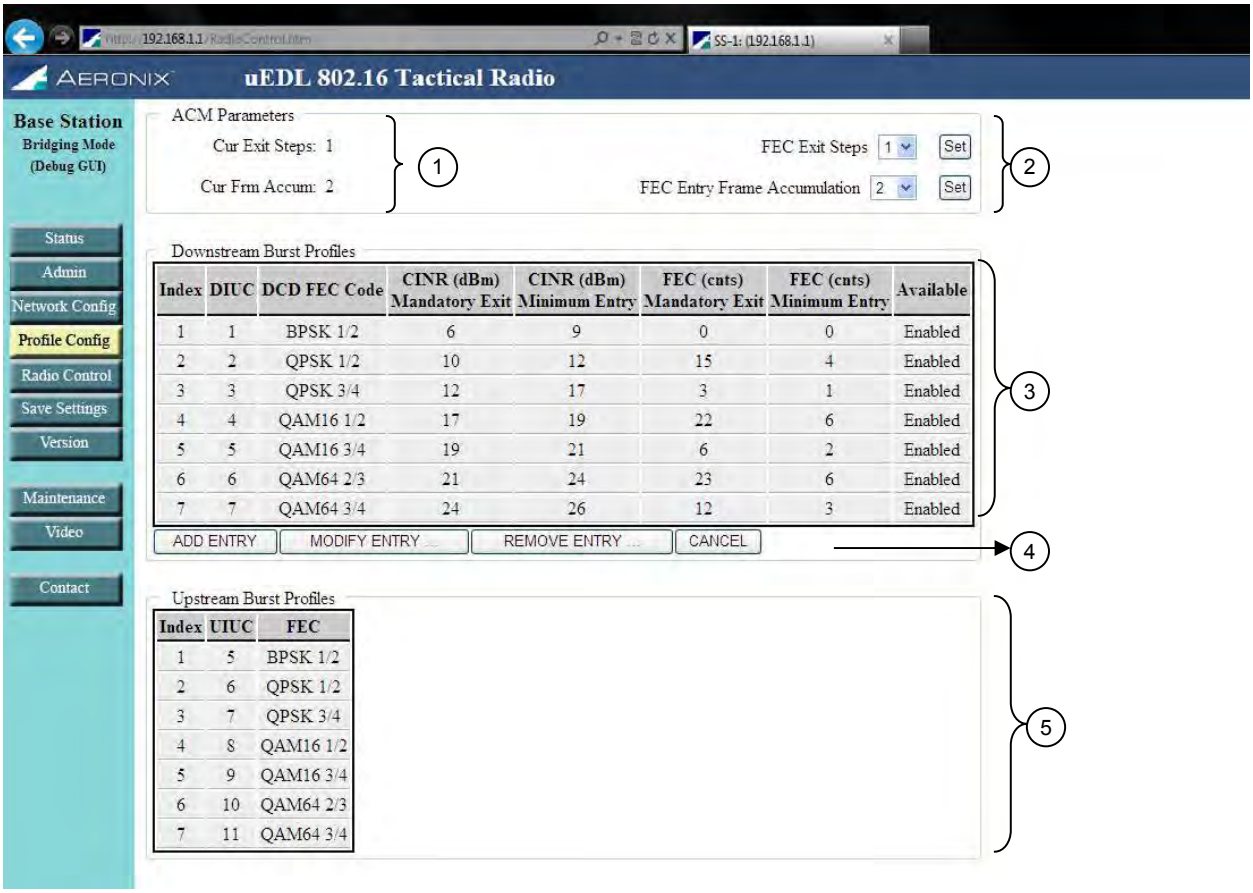


Figure 3-36 OFDM BS Profile Config Page

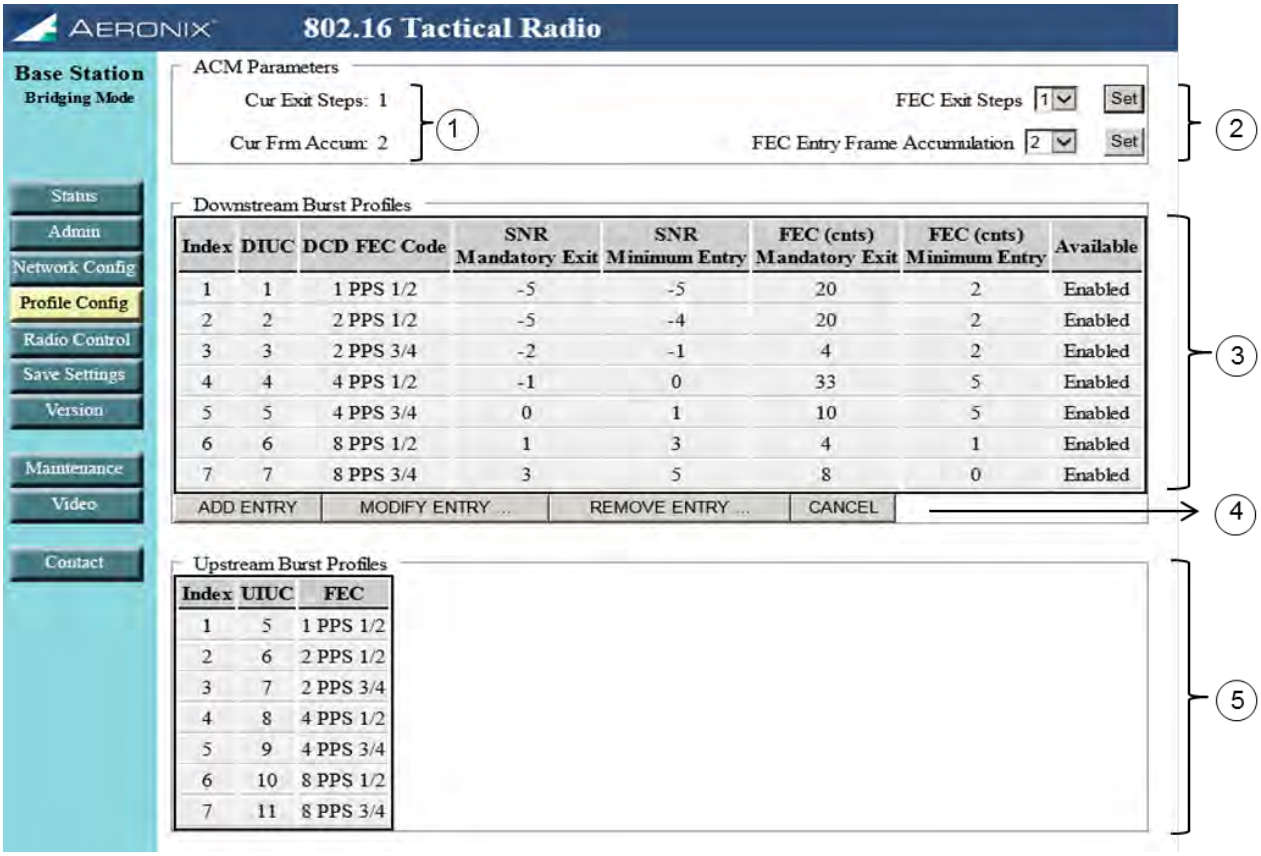


Figure 3-37 GMSK BS Profile Config Page

Table 3-11. Base Station Profile Config Page		
	Field	**Description**
1	Current ACM Parameters	OFDM ACM decisions are based on the received CINR and FEC corrected bits. GMSK ACM decisions are based on the received SNR and FEC corrected bits. • Cur Exit Steps – Current number of steps to lower the FEC Code type if need be. • Cur Frame Accum – Current number of frames to Accumulate before making the decision to move between FEC Codes.
2	Change ACM Parameters	• FEC Exit Steps – Provides the capability for the administrator to configure the number of steps in the burst profile table to take when ACM determines it is time to exit the current FEC. Valid of values are 0-2. A value of 0 means to jump to most robust FEC configured and enabled. • FEC Entry Frame Accumulation – Provides the capability for the administrator to configure the number of frames that must be accumulated in

Table 3-11. Base Station Profile Config Page		
	Field	**Description**
		succession that meet the change FEC criteria before moving to another FEC. Valid values are provided in the pull down menu.
3	Downstream Burst Profiles	Displays the downstream burst profiles which define the FEC types that the base station uses for transmissions. • Index - Value used internally to index the bust profiles. The value gets automatically generated. • DIUC – Downlink interval usage code defined by 802.16d. The values get generated automatically and are for viewing and debug purposes. • DCD FEC Code – Defines the forward error correction and coding for the burst profile entry. • CINR/SNR Mandatory Exit – Defines the CINR/SNR value in dBm used by ACM. If current average CINR/SNR is less than this value then ACM will exit current FEC code type. • CINR/SNR Minimum Entry – Defines the CINR/SNR value in dBm at which ACM allows entry into the FEC code type. The CINR/SNR average must be greater than the current FEC code types CINR/SNR minimum entry and equal to or greater than the next FEC code types CINR/SNR minimum entry. • FEC Mandatory Exit – Defines the largest number of FEC bits of one symbol in a frame averaged over the number of frames specified in FEC Entry Frame Accumulation that can occur before the decision is made to move out of the current FEC code type and into a lower more robust FEC code type. The FECs averaged over FEC Entry Frame Accumulation must be greater than the FEC Mandatory Exit before the move will occur. • FEC Minimum Entry – Defines the largest number of FEC bits of one symbol in a frame averaged over the FEC Entry Frame Accumulation can have before ACM attempts to move to a higher FEC code type. The FECs averaged over FEC Entry Frame Accumulation must be equal to or less than the FEC Mandatory Entry of the current FEC code type before the move will occur. • Available – Enable/Disable drop down menu. This option allows user to disable a certain FEC Code without deleting it from the database. Enabled profiles are used in the ACM scheme and disabled profiles are not used.
4	Control Buttons	The control buttons allow the administrator to add, modify and remove burst profiles and to cancel the current add, modify or remove transaction.

Table 3-11. Base Station Profile Config Page		
	Field	**Description**
5	Upstream Burst Profiles	The upstream burst profiles are displayed for informational and debug purposes. The upstream burst profiles use the same FEC code types and entry and exit thresholds defined for the downstream burst profiles.

ACM Note:

The OFDM and GMSK ACM methodology uses two criteria for changing modulations. It is uses the average CINR/SNR value for the connection and the number of FECs. The number of FECs represents the largest number of FECs in any symbol in the given burst. ACM requires that only one threshold is met to cause the modulation to switch to a more robust modulation and it requires that both CINR and FEC thresholds be met to move to a less robust modulation.

3.6.4.1. ADD ENTRY Downstream Burst Profile Example

The user can add a burst profile entry by selecting the ADD ENTRY button within the Downstream Burst Profile section of the Profile Config page. The user selects the FEC code from the pull down menu and selects the ADD button to the right of the entry. The new burst profile distributes through the network as needed. Each burst profile added gets populated with the default CINR and FEC thresholds. The user can change those values by modifying the burst profile.

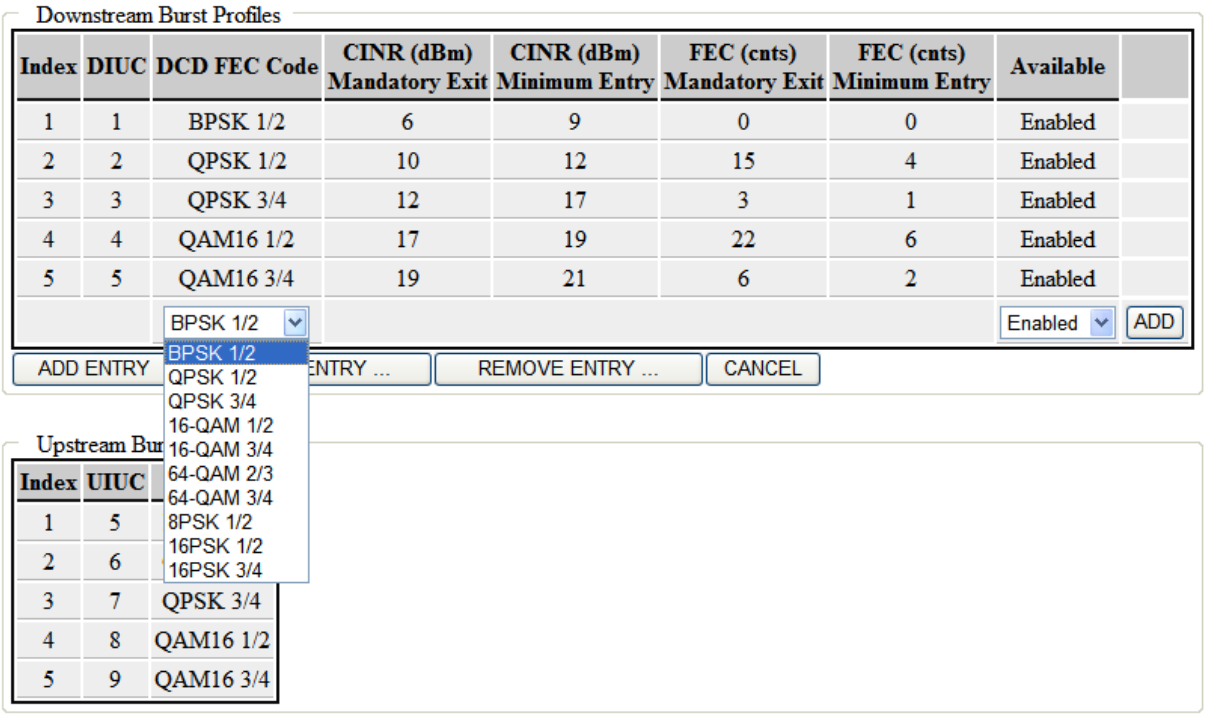


Figure 3-38 BS 802.16 Example Add Burst Profile

3.6.4.2. MODIFY ENTRY Downstream Burst Profile Example

The user can modify a burst profile entry by selecting the MODIFY ENTRY button within the Downstream Burst Profile section of the Profile Config page. The user can modify the CINR entry and exit thresholds as well as the FEC entry and exit thresholds. The user can change the available status of the profile by using the pull down menu in the Available field. Modifications are saved and distributed through the network by selecting the SAVE button to the right of the burst profile. If the user modifies the CINR entry and exit thresholds or the FEC entry and exit thresholds to values outside of the recommend range, the values appear as red on the screen to provide a warning even though they are used in the network.

Downstream Burst Profiles

Index	DIUC	DCD FEC Code	CINR (dBm) Mandatory Exit	CINR (dBm) Minimum Entry	FEC (cnts) Mandatory Exit	FEC (cnts) Minimum Entry	Available	
1	1	BPSK 1/2	6	9	0	0	Enabled	
2	2	QPSK 1/2	10	12	15	4	Enabled	
3	3	QPSK 3/4	12	17	3	1	Enabled	
4	4	QAM16 1/2	17	19	22	6	Enabled	
5	5	QAM16 3/4	19	21	6	2	Enabled	
6	6	QAM64 2/3	21	24	23	6	Enabled	
7	7	QAM64 3/4	24	26	12	3	Enabled	SAVE

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Figure 3-39 BS 802.16 Example Modify Burst Profile

3.6.5. Base Station Radio Control

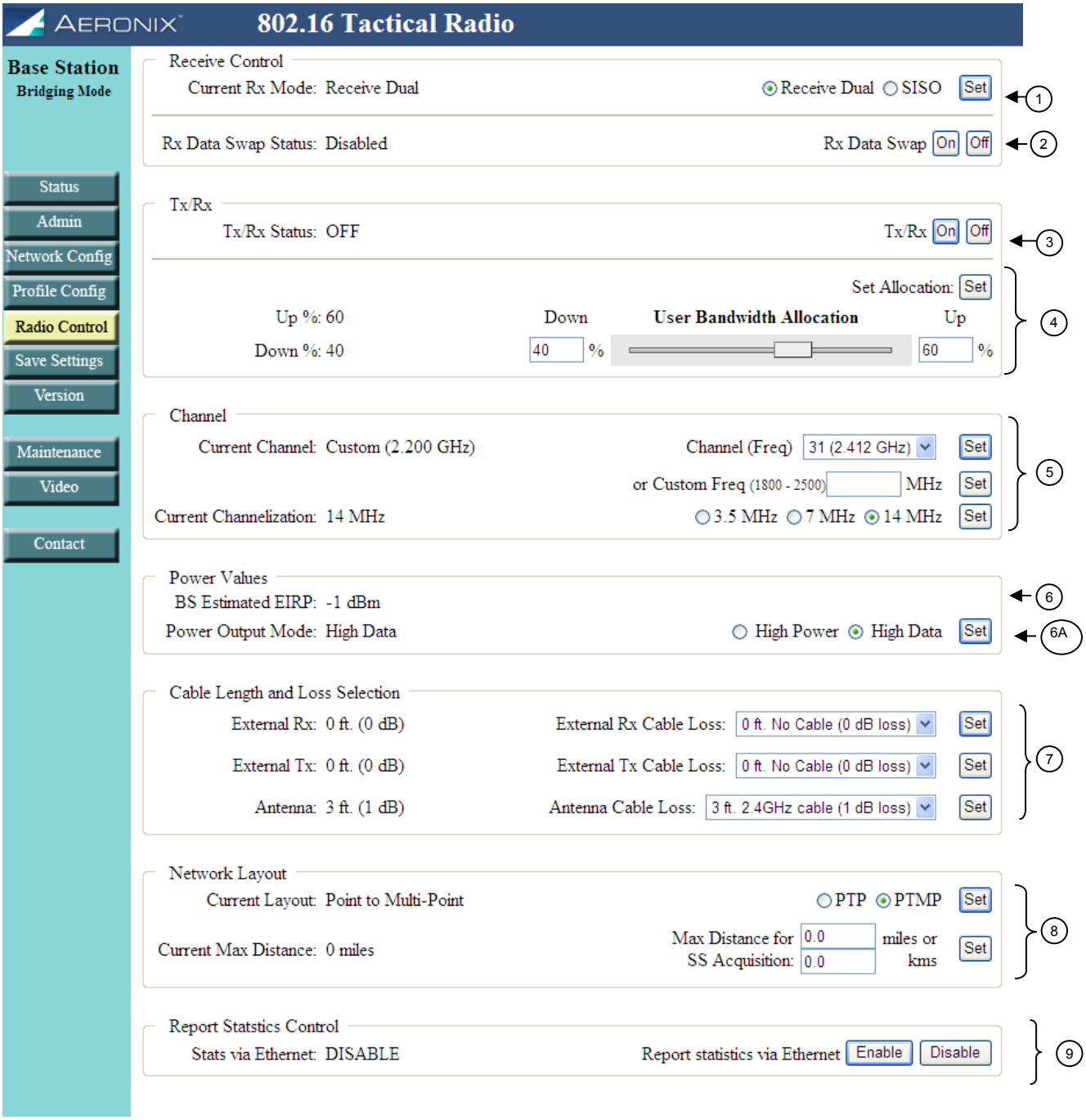


Figure 3-40 OFDM BS Radio Control Page

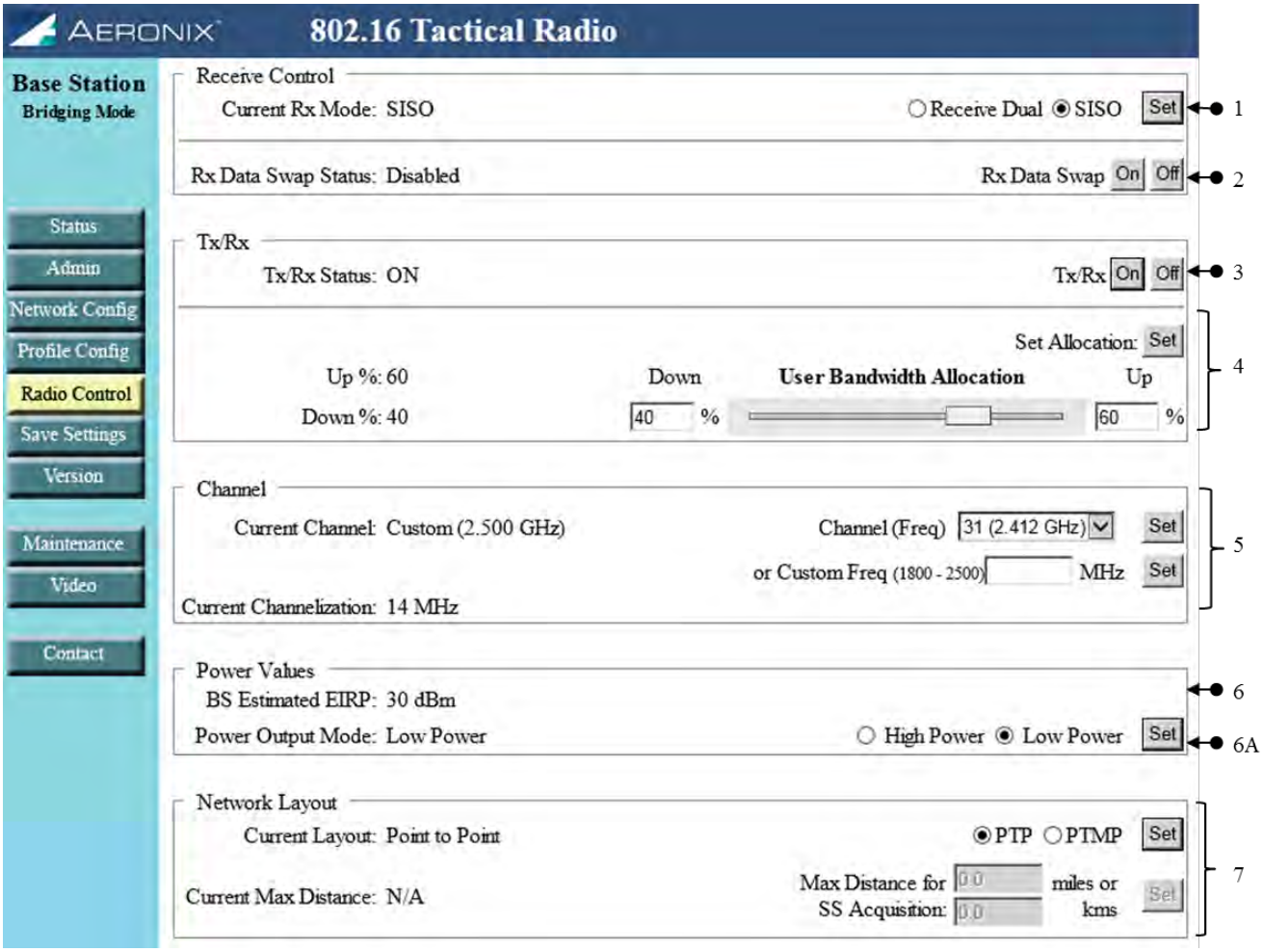


Figure 3-41 GMSK BS Radio Control

Table 3-12. Base Station Radio Control		
	Field/Control	**Description**
1	Rx Mode	• Receive Dual – Both receivers are enabled • SISO – Only one Receive path is enabled
2	Rx Data Swap	• If enabled, the receive paths will be switched in the firmware.
3	Tx/Rx	• Tx/Rx Status – Displays the current status of transmit and receive of the unit. • Tx/Rx On and Off Buttons - Provides the capability to turn on or turn off the transmit/receive capability of the unit. When the selection is off the unit does not transmit or receive.

Table 3-12. Base Station Radio Control		
	Field/Control	**Description**
4	User Bandwidth Allocation	The current link bandwidth allocation is shown in percentage of upstream and downstream allocation. This value only gets set on the base station and gets distributed to the subscribers. Currently the limits for the allocation are 80% and 20% in either direction • Up % and Down % - Displays the current allocation of the network. • Link Bandwidth Allocation Slide Bar – Provides a method to adjust the allocation. The Down and Up percentages on each side of the slide bar displays the adjusted value. The user can also type in the percentages in the Down and Up boxes instead of sliding the bar. • Set Allocation Button – Changes the current allocation to the new settings displayed in the Down and Up boxes adjacent to the slide bar. Once selected the Up % and Down % on the left updates and the new values distribute through the network.
5	Channel / Channelization	Displays channel information for the base station. • Current Channel – Displays the current channel and frequency of the base station. • Channel (Freq) – Drop down list to choose a different channel/frequency for the base station. The base station and subscriber stations need to be configured for the same channelization for proper operation. • Set Button – Changes the current channel/frequency of the base station to the channel selected by the channel (freq) pull down or the custom frequency. • Custom Freq (1800 – 2500) — Allows the administrator to type in a frequency instead of selecting one from the channel (freq) drop down list. The frequency must be a valid operating frequency for the unit. The base station and subscriber stations need to be configured for the same channelization for proper operation. • Set Button – Changes the current channel of the base station to the custom frequency • Current Channelization – Displays the current channelization of the radio. The base station and subscriber stations need

Table 3-12. Base Station Radio Control		
	Field/Control	**Description**
		to be configured for the same channelization. • Channelization Selection – Allows the selection of the channelization. Current choices are 3.5 MHz, 7 MHz, and 14 MHz for OFDM s/w and 14Mhz for GMSK s/w • Set Button – Changes the current channelization of the base station to the new value selected. **Note: Selecting a frequency outside of the ISM band (2400 – 2483.5MHz) voids the FCC Part 15.247 Certification.**
6	Power Values	Allows the user to adjust the transmit power of the radio. • BS Estimated EIRP - Displays the estimated EIRP of the base station based upon user configured gains and loss values. PA Gain + Antenna Gain – External Tx Cable Loss – Antennal Cable Loss – Internal Tx Loss.
6A	Power output mode	OFDM • High Power – radios TX power is calibrated to a higher power that still has a good enough EVM to maintain the QAM16 modulations. The QAM64 modulations will be disabled, if user re-enables them there is no guarantee the signal quality level will sustain QAM64 • High Data – radios TX power will be set to a power level who's EVM will sustain the QAM64 modulations. This power may be lower to maintain the higher EVMs required by QAM64 modulations. GMSK • High Power – radios TX power is calibrated to an output power of 35 dBm while still remaining within our GMSK mask. • High Data – radios TX power will be set to a power level of 30 dBm while remaining within our GMSK mask. **Note: Selecting the high power mode voids the FCC Part 15.247 Certification.**

Table 3-12. Base Station Radio Control		
	Field/Control	**Description**
		See 7.1 NANO output power for details on how to view transmit power values.
7	Network Layout	Displays and allows configuration of the network layout as a point to point or point to multi-point network. These items appear only on the base station and are distributed through the network. • Current Layout – Displays the current setting of the network layout. • PTP/PTMP radio buttons – o PTP button – Point to Point. Allows user to switch to a point-to-point solution (only one SS allowed in system). This option maximizes the distance one SS can be from the BS. o PTMP radio button – Point to MultiPoint. Allows multiple Subscribers to enter the system. • Set button – Changes to the selected layout option • Current Max Distance – Display the current Max Distance value • Max Distance for SS Acquisition – Allows the administrator to set the maximum distance that any subscriber in the network can be and still acquire connectivity with the base station. The value can be entered in miles or kilometers as defined by the two entry boxes. One only needs to enter a valid value in one of the boxes. This entry is only valid for the PTMP network layout and can be set to a value of up to 250 miles for OFDM radios and 210 miles for GMSK radios. Distances set for 25 miles or less default to 25 miles internally in PTMP • Set button – Sets the max distance

3.6.6. Base Station Maintenance Page

The Maintenance page provides more detailed information about the operational status of the unit than the Status page. It provides debug information as well as test capabilities.

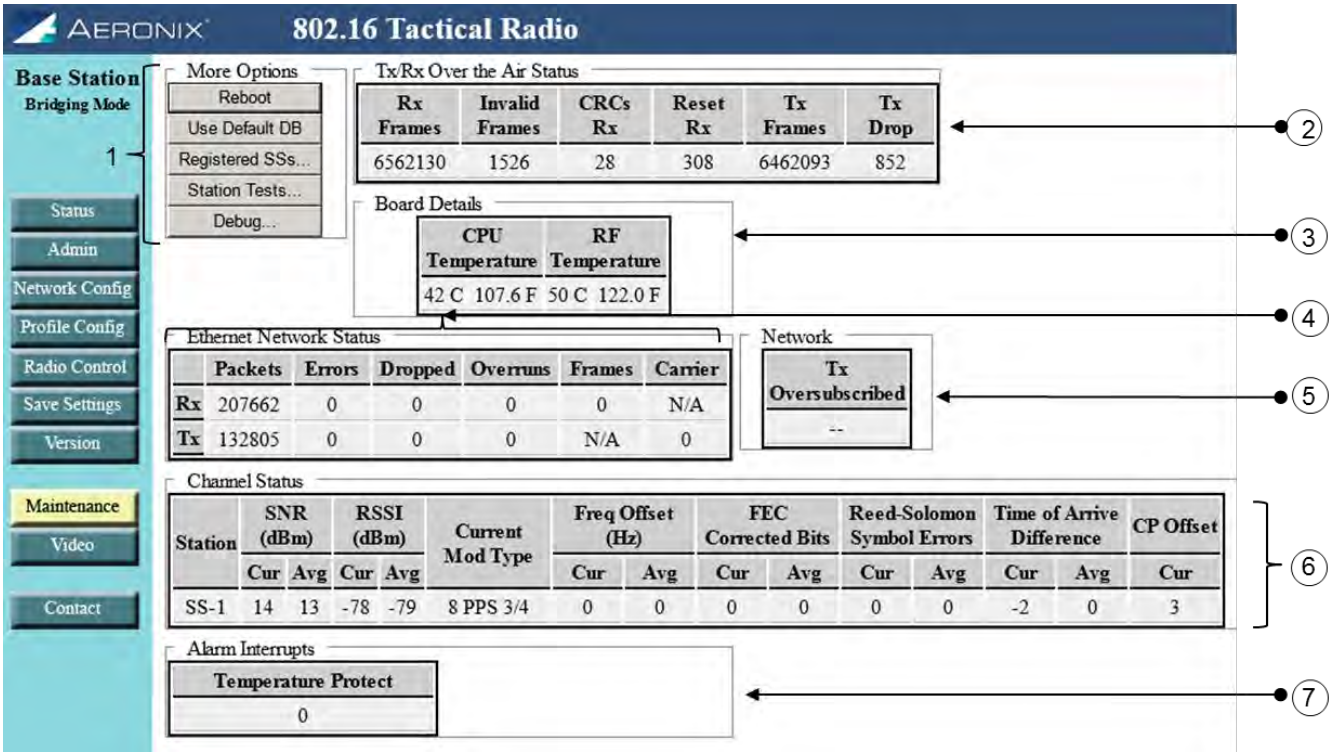


Figure 3-42 BS GMSK Maintenance Page

Table 3-13. Base Station Maintenance Page		
	Field/Control	**Description**
1	More Options	• Reboot - Provides the capability to reboot the unit remotely. Confirmation for reboot is required via a pop up window. • Use Default DB - Provides the capability to revert to the factory default database. All changes made to configuration are lost by confirming the default database option via the pop up window. • Registered SSs - Provides a table of subscribers that have successfully registered with the base station. • Station Tests - Provides some test mechanisms for debug purposes. • Debug - Provides some additional debug capabilities for trouble shooting purposes.
2	Tx/Rx Over the Air Status	Provides the transmit and receive frame counts as well as error counts for debugging. • Rx Frames – Provides the number of frames received by the unit. • Invalid Frames – Provides the number of frames received that were invalid.

Table 3-13. Base Station Maintenance Page		
	Field/Control	**Description**
		• CRCs Rx – Provides the number of CRCs received. The packet within the frame where the CRC error was received is dropped. • Reset Rx – Provides the number of times the receive path is reset. • Tx Frames – Provides the number of frames transmitted by the unit. • Tx Drop – Provides the number of transmit packets that were dropped because there is no valid class rule for that packet.
3	Board Details	Display Temperature • CPU Temperature – Displays the temperature reported by the processor in Celsius and Fahrenheit • RF Temperature – Displays the temperature reported by the AD9361 chip on the RF card in Celsius and Fahrenheit
4	Ethernet Network status	Displays the number of transmitted and received frames on the network side of the unit. This reflects the eth0 interface in a non-bridging configuration and the mybridge interface in a bridging configuration.
5	Network	Tx Oversubscribed – displays the number of network Ethernet packets that were dropped due to oversubscribing the unit.
6	Channel Status	Provides some additional values on a connection basis for debugging and antenna pointing purposes. • Station – Displays the name of the subscriber the entry represents. -------- OFDM only • Current CINR – Displays the instantaneous CINR for the connection. • Average CINR – Displays the average CINR of the connection over a period of received frames. GMSK Only • Current SNR – Displays the instantaneous estimated SNR for the connection. • Average SNR – Displays the average estimated SNR of the connection over a period of received frames. -------

Table 3-13. Base Station Maintenance Page		
	Field/Control	**Description**
		• Current RSSI – Displays the instantaneous RSSI at the preamble for the connection. • Average RSSI – Displays the average RSSI at the preamble of the connection over a period of received frames. • Current Mod Type – Displays the current modulation type received by the BS. This represents the modulation of the SS transmissions. • Curr Freq Offset (Hz) – Displays the frequency offset at the time of the display update. • Avg Frequency Offset (Hz) – Displays the average frequency offset of the connection over a period of received frames. • FEC Corrected Bits – Displays the instantaneous forward error correction corrected bits. • Average FEC Corrected Bits – Displays the average forward error correction corrected bits over a period of received frames. • Reed-Solomon Symbol Errors Cur – Displays the instantaneous number of Reed-Solomon errors. • Reed-Solomon Symbol Errors Avg – Displays the average Reed-Solomon errors over a period of received frames. • Time of Arrive Difference – Displays the offset in time of when the BS expected to receive a burst and when it actually received a burst. • CP Offset Cur – Displays the instantaneous cyclic prefix offset. This offset refers to where the FFT is positioned in the symbol. • CP Offset Avg – Displays the average CP offset over a period of received frames.
7	Alarm Interrupts	• Temperature Protect –Provides a count of how many times the RF Temperature reached 83° C. When the he RF temperature reaches 83° C, the software will disable the RF Transmit and Receive. When RF temperature lowers to 65°C, the

Table 3-13. Base Station Maintenance Page		
	Field/Control	**Description**
		software will re-enable the RF Transmit and Receive.

3.6.6.1. Base Station Maintenance Registered Subscriber Page

The Registered SSs page shows all subscribers that have completed the 802.16d registration process with the base station.

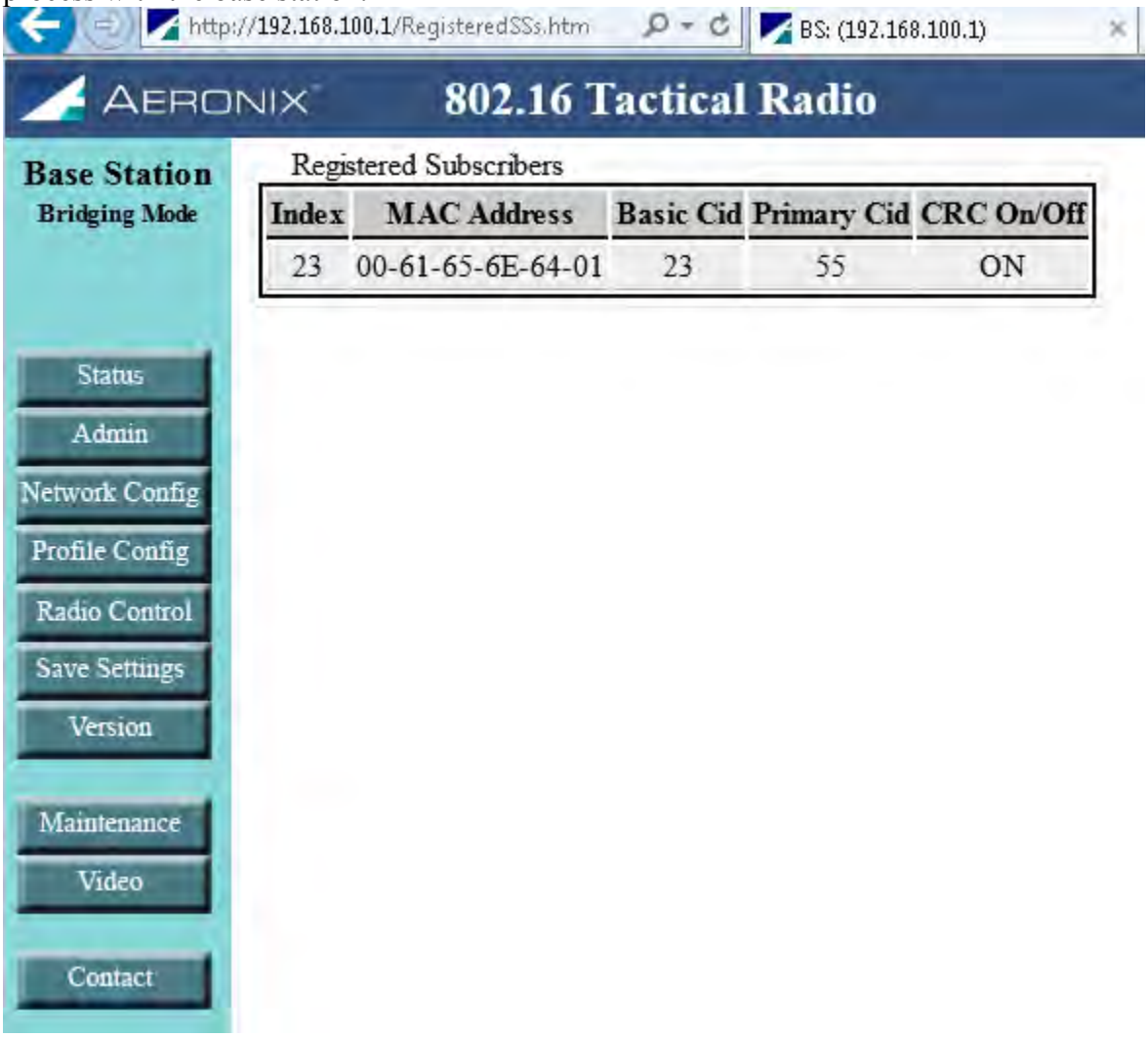


Figure 3-43 BS Maintenance Registered SS Page

3.6.7. Base Station Video Page

Figure 3-44 BS Video Page

Table 3-14. Base Station Video Page		
	Field/Control	**Description**
1	Video Control Current Settings	• Allow Bad Frames – Enabling Bad Frames will allow UDP packets received over the RF link that have bit errors in them to continue to be pushed up to the radio's Ethernet stack. This check is only made on UDP packets with destination port addresses between 1232-1247 • Currently the camera interface on the NANO radio has not been implemented.
2	Serial Flow Control Type	The serial flow feature wraps the serial data in a UDP IP Packet. • Serial B to Serial B – This allows bidirectional serial data to flow from Serial B of one radio to Serial B of another radio. • Serial B To Ethernet – Transfer data between serial B and Ethernet port

Table 3-14. Base Station Video Page		
	Field/Control	**Description**
		o Rx Only – Receive data from designated receive port and push the data out to serial B o Tx Only – Transmit data from serial B to designated destination IP address and IP port o Both (Rx and Tx) – Radio will push data from its Serial B to an external node with the destination IP address and IP Port configured at GUI. The radio will also receive data from receive port and send that data to its Serial B.

3	Serial Flow Control Configuration	• Serial Baud Rate – A drop down menu with a selection of baud rates available.
		• Serial Raw Mode –
		  • Enable – Enables Raw mode and serial B will process serial input in non-canonical mode.
		  • Disable – Serial B will be set in canonical mode. This will enable line processing mode and each line designated by a new line. The entire line will be wrapped into a UDP packet and then sent across the RF link.
		• Destination Port – Ability to set the port that the BS serial data is being transmitted on.
		• Receive Port – Ability to set the port that the BS is receiving serial data on.
		• Destination IP Addr – Ability to set the Radio's IP address that the serial data is being transmitted to and received from.
		• Raw Serial mode processing configurations
		  • Raw Buffer – in bytes, the maximum is 255 bytes.
		  • Raw Timeout (deci seconds) - in tenths of seconds, maximum is 255 deci seconds.
		When in Raw mode the user can configure the serial to buffer bytes and to be more efficient than sending just one character, below are the valid configurations.
		  • Raw Buffer > 0 Raw and Timeout > 0 – The serial will buffer the bytes until the first condition is met and then the data will be packaged into a UDP packet and sent. The timer will be started when the first byte is received.

Table 3-14. Base Station Video Page		
	Field/Control	**Description**
		• Raw Buffer > 0, and Raw Timeout = 0 – The serial will buffer bytes until the number of bytes on the serial match the Raw Buffer bytes and then the data will be packaged into a UDP packet and sent. • Raw Buffer = 0 and Raw Timeout > 0 – The serial will buffer the amount of bytes in the amount of time specified by the Raw Timeout. The timer is started at each interval, ie not when the first byte is received. • Raw Buffer = 0, and Raw Timeout = 0 – This will package up a byte as it is received from the serial. • Disable/Enable Flow Radio Buttons – Allows user to disable or enable the serial flow control • Submit Button – Once submitted all of the above changes will take effect

3.6.8. Subscriber Station Status

When the GUI is accessed using an approved browser, the unauthenticated status page is displayed.

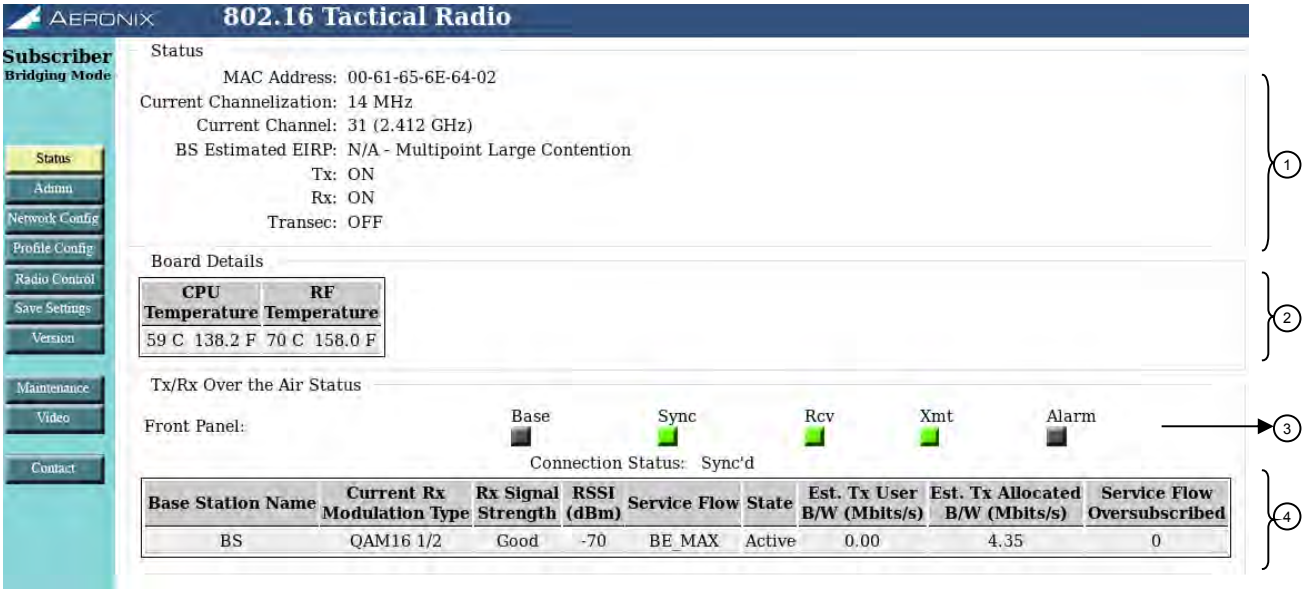


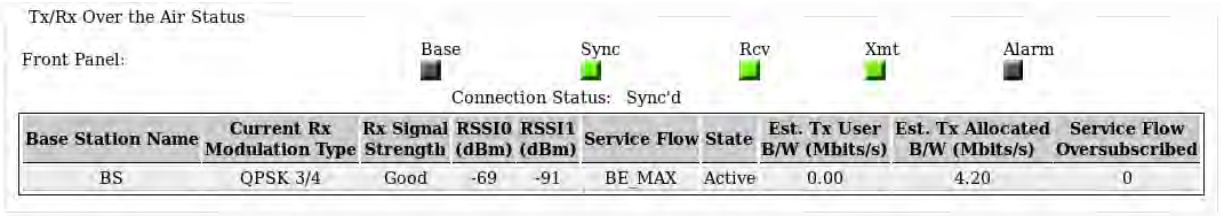
Figure 3-45 SS OFDM Status Page

Table 3-15. Subscriber Station Status Page		
	Field/Control	**Description**
1	Status	Status displays a subset of current settings. • MAC Address – The IEEE 802.16 MAC address of the base station • Current Channelization – Displays the channelization of the subscriber station. • Current Channel – The current 802.16 specified channel that the subscriber station uses for transmit and receive. • BS Estimated EIRP – Displays the estimated EIRP of the base station that the subscriber received from the base station. A value of N/A – No Sync displays when the subscriber is not connected to the base station. • Tx - Provides the status of the transmit portion of the unit. (On/Off) The status may be changed on the Radio Control screen. • Rx - Provides the status of the receive portion of the unit. (On/Off). • Transec Status – Displays the status of Transec (on or off).
2	Board Details	Display Temperature • CPU Temperature – Displays the temperature reported by the processor in Celsius and Fahrenheit
3	Front Panel	Displays PHY status on an interval basis. • Base is a constant green light if the radio is configured as a Base Station • Sync is off on the base station when no subscribers are registered and is illuminated continuously when subscribers are registered. • Recv is green when the radio is receiving messages • Xmit is green when the radio is transmitting data
4	Tx/Rx Over the Air Status	The first part of the Tx/Rx Status Table Displays the receive status of each subscriber connection. The second part of the table is the amount of data the BS is transmitting to each active SS. • Base Station Name - Contains the name of the base station for which the information pertains to. • Current Modulation Type – Contains the current modulation/FEC type for the subscriber connection. The display show the modulation of the receive frame at the time sampled. • Rx Signal Strength: Contains an assessment of the receive signal strength.

Table 3-15. Subscriber Station Status Page		
	Field/Control	**Description**
		• RSSI at LNA – Displays the average RSSI value for the subscriber connection at the LNA. The value is determined using the RSSI value seen at the PHY card at the preamble and the values entered in the GUI for the receive gain and loss from the PHY card to the LNA. Values included: Avg Rssi, Internal Rx Loss, Rx Cable Loss, LNA • Service Flow – Identifies the service class of the service flow for the connection. • State – Displays the state of the connection. The valid values are provisioned and active. • Est User B/W (Mbits/sec) – Displays the estimated bandwidth in megabits per second being used by the connection in the transmit direction. • Est Allocated B/W (Mbits/sec) – Displays the estimated bandwidth in megabits per second being allocated to the connection in the transmit direction. • Service Flow Oversubscribed – Displays the number of Ethernet packets dropped for that specific service flow.

3.6.8.1. Subscriber Station Status Page in Receive Dual

When the user enables RX Dual on the Radio control page the second receive RSSI will be reported on the RSSI1 column on the Tx/Rx Over the Air Status section.



Base Station Name	Current Rx Modulation Type	Rx Signal Strength	RSSI0 (dBm)	RSSI1 (dBm)	Service Flow	State	Est. Tx User B/W (Mbits/s)	Est. Tx Allocated B/W (Mbits/s)	Service Flow Oversubscribed
BS	QPSK 3/4	Good	-69	-91	BE_MAX	Active	0.00	4.20	0

Figure 3-46 SS OFDM Status Rx Dual

3.6.9. Subscriber Station Admin Page

The admin page for the subscriber station provides the capability to enable and disable Transec and the selection of the Transec key. Account management, start script generation and software download are also supported on the admin page.

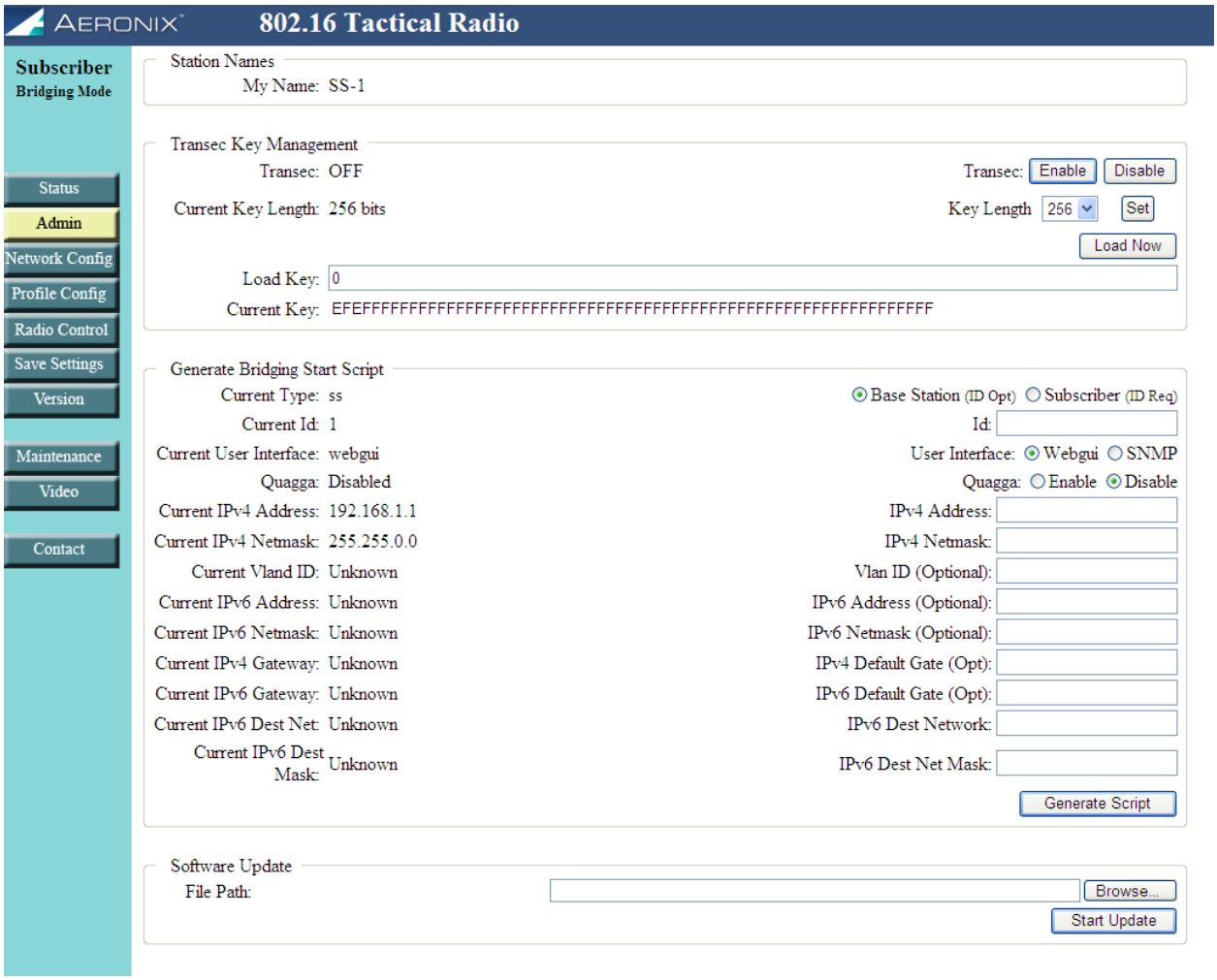


Figure 3-47 SS Admin Page

3.6.9.1. SS Admin Station Name and Transec Key Management

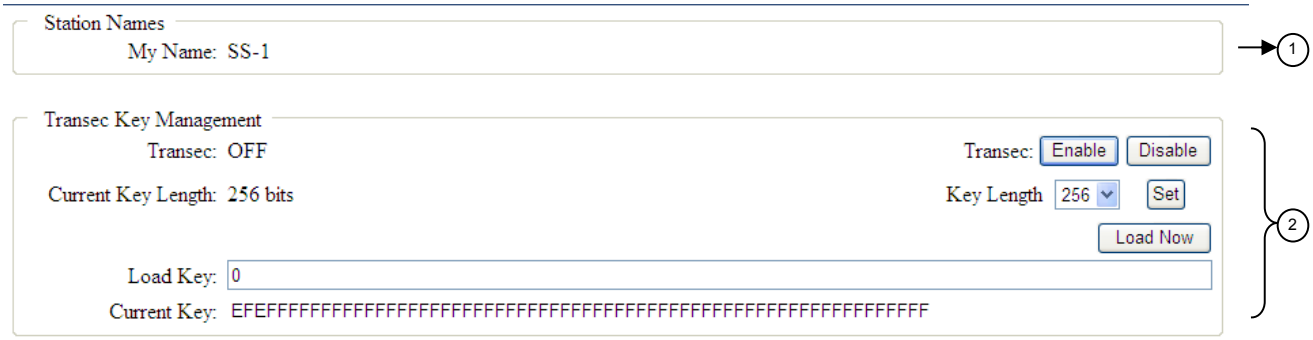


Figure 3-48 SS Admin Station Name and Transec Key

Table 3-16. Subscriber Station Admin Page Transec Key		
1	Station Names	• My Name - Displays the current name of the subscriber. The base station provides the subscriber name over the network. Configuration of subscriber names occurs on the base station Admin page.
2	Transec Key Management	Displays the settings for Transec capability. • Transec - Displays whether Transec is on or off on the base station. • Transec Enable/Disable buttons - Turn Transec on or off. • Key Length drop down menu – Allows user to specify the number of bits for the Transec Key • Load Key – Allows the ability to change the Transec Key in hexadecimal based on Key length. • Current Key – Displays the current key.

3.6.9.2. SS Admin Bridging Start Script and Software Update

Bridging is the default mode for operating the units. The Generate Bridging Start Script section of the admin page allows for configuration of the start script. Once the configuration is set by selecting the Generate Script button, the unit must be rebooted or power cycled for the configuration to take effect.

For account management and software update examples, refer to the 3.6.2.2 Base Station BS Admin Bridging Start Script and Software Update.

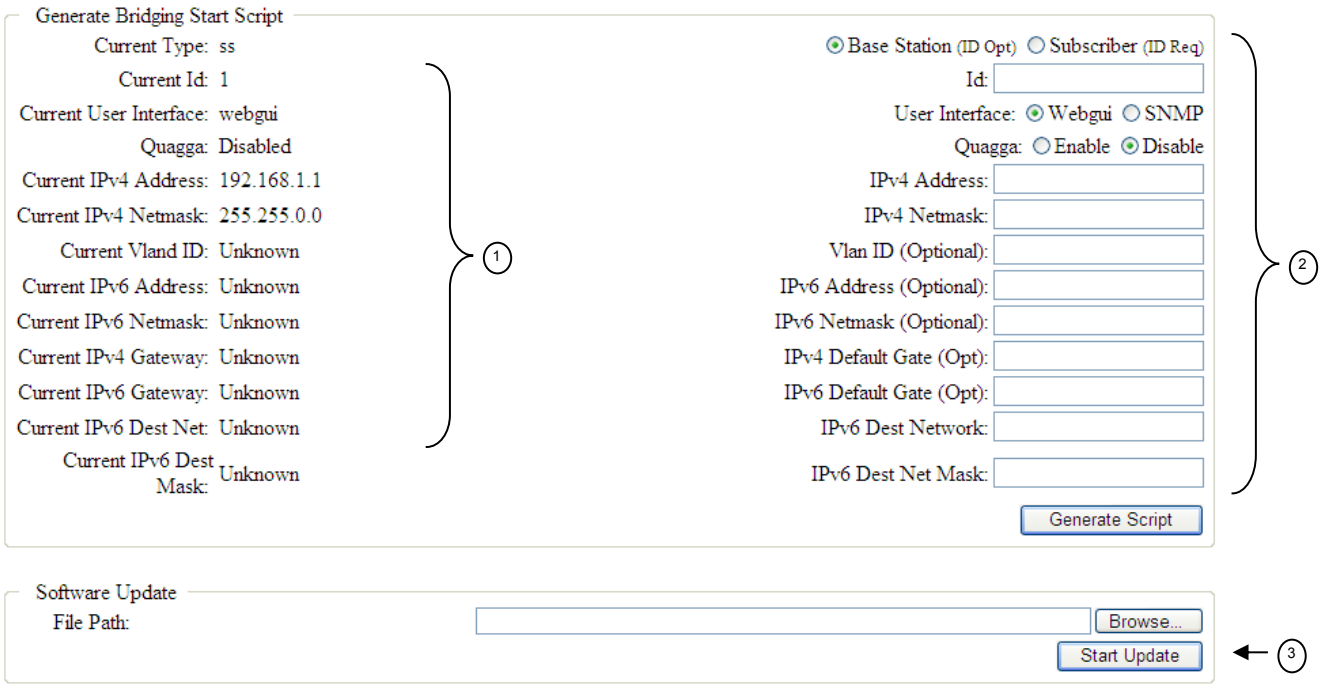


Figure 3-49 SS Admin Start Script and SW Update

Table 3-17. Subscriber Station Admin Page Transec Key		
1	Current Start Script	The current start script parameters.
2	Generate Script	User can change the mode the radio boots in or the IP address,etc. The unit should not be powered cycled while script generation is in process. • Type radio buttons: Select BS or SS for type • Id: If setting up start script as SS enter its Subscriber Number. • User Interface: Only valid interface for EDL is the Webgui • Quagga: Option to enable additional network routing software suite providing implementations of Open Shortest Path First (OSPF). Note there is user configuration necessary to implement Quagga. • IPv4 Address: Provides the capability to set a new IPv4 address for the unit. • IPv4 Netmask: Provides the capability to set a new IPv4 netmask address for the unit

Table 3-17. Subscriber Station Admin Page Transec Key		
		• Vlan ID: Allows user to add a VLAN ID to IP packets coming from radio. The radio will pass VLAN tagged packets unmodified, so this does not need to be set to allow VLAN packets to flow through the radio. Only if communication to the radio needs to have a VLAN ID. • IPv6 Address (Optional) : Optional IPv6 address for the Ethernet (Note : current release of EDL does not support IPv6) • IPv4 Default Gate (Opt) : Optional ability to configure an IPv4 default gateway. • IPv6 Default Gate (Opt) : Optional ability to configure an IPv6 default gateway • IPv6 Netmask (Optional) : Optional IPv6 netmask for IPv6 address (Note: current release of EDL does not support IPv6) • Enable/Disable Video Support Radio Buttons: If the start script is being configured for a SS then user has the option of enabling the video software to run at start up. Currently not a valid selection if the start script is being configured as a BS • Generate Script Button : Generates a new start script for the unit with the values input in the "Generate Bridging Start Script" section of the Admin page. Once the script has been generated, the unit must be rebooted or power cycled for the new values to take effect. The unit should not be power cycled while script generation is in process.
3	Software Update	Allows user to select a signed package file provided by the manufacturer to update the software on the unit. • File Path – Allows users to browse to location where the package file resides. • Start Update - Selection of the start update button starts the software update using the factory signed package file specified in the File Path. Once the update process begins, any interruption can render the box unusable and require its return to the factory.

3.6.10. Subscriber Network Config Page

The Network Config page on the subscriber contains the class rule information received from the base station.

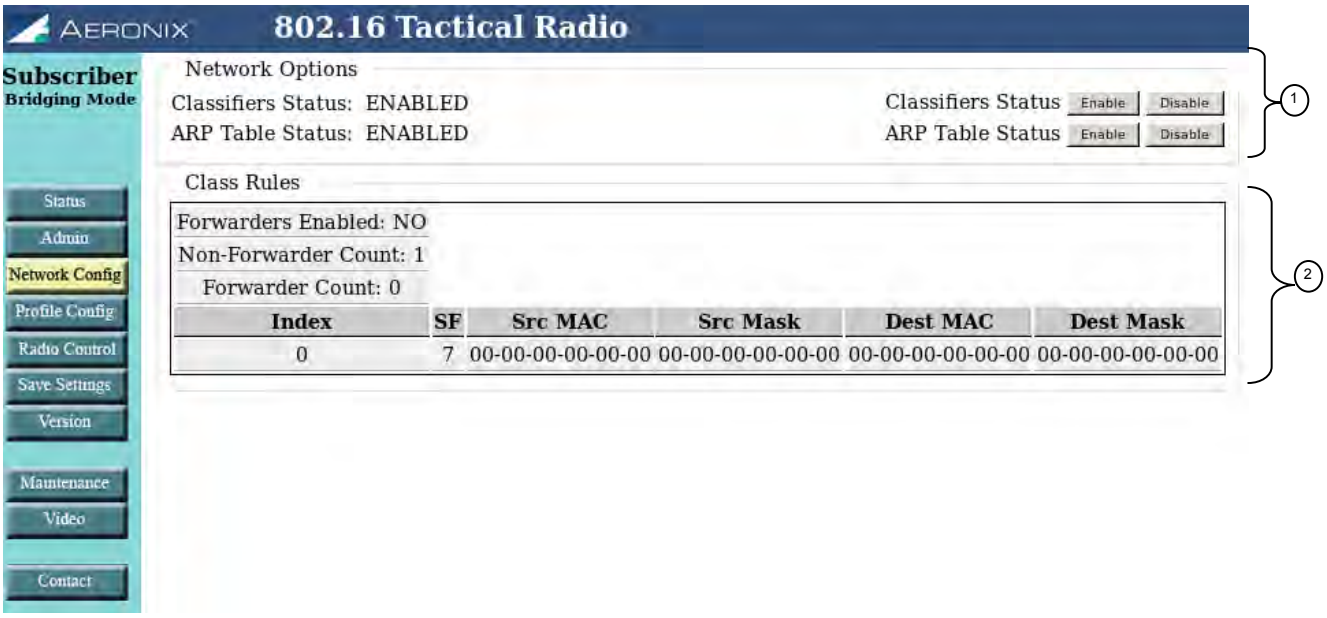


Figure 3-50 SS Network Config Page

Table 3-18. Subscriber Station Network Config

	Field/Control	Description
1	Network Options	• Classifier Status : Disables or Enables using the Classifier Rules Table • ARP Table Status: Disables or Enables the ARP Table Feature. See 7.1
2	Class Rules	Contains the following: • Index - Internal index used for class rules. • SF - Identifies the service flow that pertains to the class rule. • SRC IP/MAC - The source address for the class rule. The address presented to the unit when initiating traffic through it. • Src Mask - The address mask for the source address. Masking portions of the IP address is only valid when the addresses of the units are configured on different subnets. Currently only the last subnet byte is allowed to be masked. • Dest IP/MAC – The destination address for the class rule. The destination address for the traffic through the unit. • Dest Mask - The address mask for the destination address. Masking portions of the IP address is only valid when the addresses of the units are configured on different subnets. Currently only the last subnet byte is allowed to be masked.

3.6.11. Subscriber Profile Config Page

The subscriber station Profile Config page contains the DCD and UCD information for the unit received from the base station. This page is not modifiable from the subscriber. The information can only be modified on the base station.

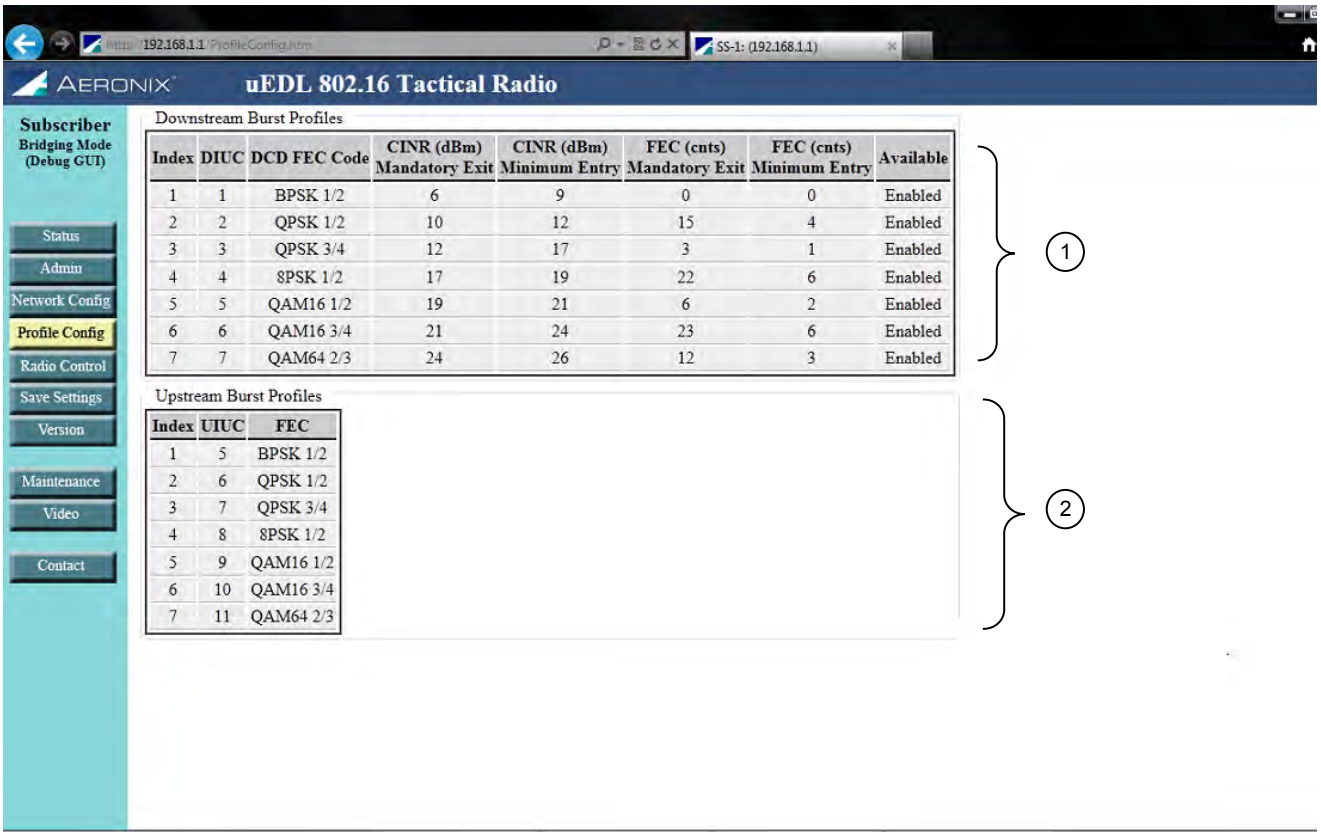


Figure 3-51 SS OFDM Profile Config Page

Table 3-19. Subscriber Profile Config Page		
	Field	**Description**
1	Downstream Burst Profiles	• Index – Value used internally to index the bust profiles. The value gets automatically generated. • DIUC - Downlink interval usage code defined by 802.16d. The values get generated automatically and are for viewing and debug purposes. • DCD FEC Code - Defines the forward error correction and coding for the burst profile entry. • CINR/SNR Mandatory Exit - Defines the CINR/SNR value in dBm at which ACM exits the FEC code type.

Table 3-19. Subscriber Profile Config Page		
	Field	**Description**
		• CINR/SNR Minimum Entry - Defines the CINR/SNR value in dBm at which ACM allows entry into to the FEC code type. • FEC Mandatory Exit - Defines the largest number of FEC bits one symbol can have before the decision is made to move out of the current FEC code type. • FEC Minimum Entry - Defines the largest number of FEC bits one symbol can have before ACM attempts to move to a higher data per symbol FEC Code type. • Available - symbol FEC Code type. • Available – Displays the availability of the burst profile. Enabled profiles are used in the ACM scheme and disabled profiles are not used.
2	Upstream Burst Profiles	The upstream burst profiles are displayed for informational and debug purposes. The upstream burst profiles use the same FEC code types and entry and exit thresholds defined for the downstream burst profiles.

3.6.12. Subscriber Station Radio Control Page

The subscriber station radio control page allows for configuring the 802.16d radio settings.

Figure 3-52 SS OFDM Radio Control Page

Figure 3-53 SS GMSK Radio Control

Table 3-20. Subscriber Station Radio Control

	Field/Control	Description
1	Rx Mode	• Receive Dual – Both receivers are enabled • SISO – Only one Receive path is enabled
2	Rx Data Swap	• If enabled, the receive paths will be switched in the firmware.
3	Tx/Rx	• Tx/Rx Status - Shows the current status of the transmit and receive of the unit. • Tx/Rx On and Off Buttons - Provides the capability to turn on or turn off the transmit/receive capability of the unit. When the selection is off the unit does not transmit or receive.
4	Channel / Channelization	Displays channel information for the base station. • Scan Channels – Displays the current setting for scan channels (OFF/ON). • Scan Channels button – Allows the user to turn scan channels on or off. If on, the subscriber scans through the available channels per the channelization selected and tries to connect to a base station at each frequency. The subscriber remains

Table 3-20. Subscriber Station Radio Control		
	Field/Control	**Description**
		connected to the first base station a successful connection is made • Current Channel – Displays the current channel and frequency of the base station. • Channel (Freq) – – Pull down menu to choose a different channel/frequency for the base station. The base station and subscriber stations need to be configured for the same channelization for proper operation. • Set Button – Changes the current channel of the base station to the channel selected by the channel (freq) drop down. • Custom Freq (2412 – 2462) – Allows user to create a frequency outside of the drop down frequency list. The frequency must be a valid operating frequency for the unit. • Set Button – Changes the current channel of the base station to the custom frequency • Current Channelization – Displays the current channelization of the radio. The base station and subscriber stations need to be configured for the same channelization. • Channelization Selection – Allows the selection of the channelization. Current choices are 3.5 MHz, 7 MHz, and 14 MHz • Set Button – Changes the current channelization selected by the user. **Note: Selecting a frequency outside of the ISM band (2400 – 2483.5MHz) voids the FCC Part 15.247 Certification.**
5	Power Values	• BS Estimated EIRP - Displays the estimated EIRP of the base station based upon user configured gain and loss values.
5A	Power output mode	OFDM only • High Power – radios TX power is calibrated to a higher power that still has a good enough EVM to maintain the QAM16 modulations. The QAM64 modulations will be disabled, if user re-enables them there is no guarantee the signal quality level will sustain QAM64 • High Data – radios will TX power will be set to a power level whose EVM will

Table 3-20. Subscriber Station Radio Control		
	Field/Control	**Description**
		sustain the QAM64 modulations. this power may be lower to maintain the higher EVMs required by QAM64 modulations. GMSK • High Power – radios TX power is calibrated to an output power of 35-36 dB while still remaining within our GMSK mask. • High Data – radios TX power will be set to a power level of 30 dBm while remaining within our GMSK mask. **Note: Selecting the high power mode voids the FCC Part 15.247 Certification.** See 7.1 NANO output power for details on how to view transmit power values.

3.6.13. Subscriber Station Maintenance Page

The Maintenance page provides more detailed information about the operational status of the unit than does the Status page. It provides debug information and capabilities as well as test capabilities.

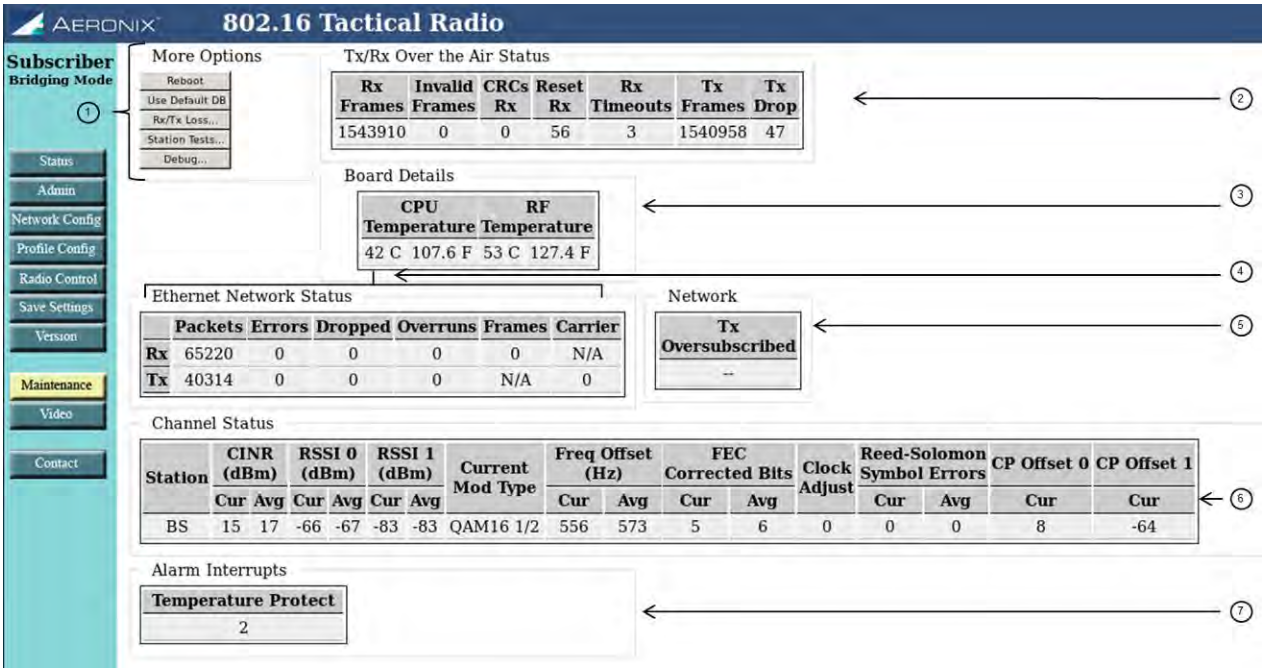


Figure 3-54 SS OFDM Maintenance Page

Table 3-21. Subscriber Station Maintenance Page		
	Field/Control	**Description**
1	More Options	• Reboot - Provides the capability to reboot the unit remotely. Confirmation for reboot is required via a pop up window. • Use Default DB - Provides the capability to revert to the factory default database. All changes made to configuration are lost by confirming the default database option via the pop up window. • Registered SSs - Provides a table of subscribers that have successfully registered with the base station. • Rx/Tx Loss - Provides a method to configure receive and transmit loss values that are internal to the unit. • Station Tests - Provides some test mechanisms for debug purposes. • Debug - Provides some additional debug capabilities for trouble shooting purposes.
2	Tx/Rx Over the Air Status	Provides the transmit and receive frame counts as well as error counts for debugging. • Rx Frames – Provides the number of frames received by the unit. • Invalid Frames – Provides the number of frames received that were invalid. • CRCs Rx – Provides the number of CRCs received. The packet within the frame where the CRC error was received is dropped. • Reset Rx – Provides the number of times the receive path is reset. • Rx Timeouts – Number of bursts the Subscriber has missed. • Tx Frames – Provides the number of frames transmitted by the unit. • Tx Drop – Provides the number of transmit packets that were dropped because there is no valid class rule for that packet.
3	Board Details	Display Temperature • CPU Temperature – Displays the temperature reported by the processor in Celsius and Fahrenheit • RF Temperature – Displays the temperature reported by the AD9361 chip on the RF card in Celsius and Fahrenheit
4	Ethernet Network status	Displays the number of transmitted and received frames on the network side of the unit. This

Table 3-21. Subscriber Station Maintenance Page

	Field/Control	Description
		reflects the eth0 interface in a non-bridging configuration and the mybridge interface in a bridging configuration.
5	Network	Tx Oversubscribed – displays the number of network Ethernet packets that were dropped due to oversubscribing the unit.
6	Channel Status	Provides some additional values on a connection basis for debugging and antenna pointing purposes. • Station – Displays the name of the subscriber the entry represents. --- OFDM only • Current CINR – Displays the instantaneous CINR for the connection. • Average CINR – Displays the average CINR of the connection over a period of received frames. GMSK Only • Quality Cur – Displays a number that quantifies the link reliability, with 26 being the best link value • Quality Avg – Average quality value --- • Current RSSI – Displays the instantaneous RSSI at the preamble for the connection. • Average RSSI – Displays the average RSSI at the preamble of the connection over a period of received frames. • Current Mod Type – Displays the current modulation type received by the BS. This represents the modulation of the SS transmissions. • Curr Freq Offset (Hz) – Displays the frequency offset at the time of the display update. • Avg Frequency Offset (Hz) – Displays the average frequency offset of the connection over a period of received frames. • FEC Corrected Bits – Displays the instantaneous forward error correction corrected bits. • Average FEC Corrected Bits – Displays the average forward error correction corrected bits over a period of received frames.

Table 3-21. Subscriber Station Maintenance Page		
	Field/Control	**Description**
		• Reed-Solomon Symbol Errors Cur – Displays the instantaneous number of Reed-Solomon errors. • Reed-Solomon Symbol Errors Avg – Displays the average Reed-Solomon errors over a period of received frames. • Time of Arrive Difference – Displays the offset in time of when the BS expected to receive a burst and when it actually received a burst. • CP Offset Cur – Displays the instantaneous cyclic prefix offset. This offset refers to where the FFT is positioned in the symbol. • CP Offset Avg – Displays the average CP offset over a period of received frames.
7	Alarm Interrupts	• Temperature Protect –Provides a count of how many times the RF Temperature reached 85°C. When the he RF temperature reaches 85°C, the software will disable the RF Transmit and Receive. When RF temperature lowers to 65°C, the software will re-enable the RF Transmit and Receive.

3.6.14. Subscriber Station Video Page

Figure 3-55 SS Video Page

Table 3-22. Subscriber Station Video Page		
	Field/Control	**Description**
1	Video Control Current Settings	• Allow Bad Frames – Enabling Bad Frames will allow UDP packets received over the RF link that have bit errors in them to continue to be pushed up to the radio's Ethernet stack. This check is only made on UDP packets with destination port addresses between 1232-1247. • Currently the camera interface on the NANO radio has not been implemented.
2	Serial Flow Control Type	The serial flow feature wraps the serial data in a UDP IP Packet. • Serial B to Serial B – This allows bidirectional serial data to flow from Serial B of one node to Serial B of another node. • Serial B To Ethernet – Transfer data between serial B and Ethernet port o Rx Only – Receive data from designated receive port and push the data out to serial B

Table 3-22. Subscriber Station Video Page		
	Field/Control	**Description**
		o Tx Only – Transmit data from serial B to designated destination IP address and IP port o Both (Rx and Tx) – Radio will push data from its Serial B to an external node with the destination IP address and IP Port configured at GUI. The radio will also receive data from receive port and send that data to its Serial B.

3	Serial Flow Control Configuration	• Serial Baud Rate – A drop down menu with a selection of baud rates available.
		• Serial Raw Mode –
		  • Enable – Enables Raw mode and serial B will process serial input in non-canonical mode.
		  • Disable – Serial B will be set in canonical mode. This will enable line processing mode and each line designated by a new line. The entire line will be wrapped into a UDP packet and then sent across the RF link.
		• Destination Port – Ability to set the port that the BS serial data is being transmitted on.
		• Receive Port – Ability to set the port that the BS is receiving serial data on.
		• Destination IP Addr – Ability to set the Radio's IP address that the serial data is being transmitted to and received from.
		• Raw Serial mode processing configurations
		  • Raw Buffer – in bytes, the maximum is 255 bytes.
		  • Raw Timeout (deciseconds) - in tenths of seconds, maximum is 255 deciseconds.
		When in Raw mode the user can configure the serial to buffer bytes and to be more efficient than sending just one character, below are the valid configurations.
		  • Raw Buffer > 0 Raw and Timeout > 0 – The serial will buffer the bytes until the first condition is met and then the data will be packaged into a UDP

Table 3-22. Subscriber Station Video Page		
	Field/Control	**Description**
		packet and sent. The timer will be started when the first byte is received. • Raw Buffer > 0, and Raw Timeout = 0 – The serial will buffer bytes until the number of bytes on the serial match the Raw Buffer bytes and then the data will be packaged into a UDP packet and sent. • Raw Buffer = 0 and Raw Timeout > 0 – The serial will buffer the amount of bytes in the amount of time specified by the Raw Timeout. The timer is started at each interval, ie not when the first byte is received. • Raw Buffer = 0, and Raw Timeout = 0 – This will package up a byte as it is received from the serial. • Disable/Enable Flow Radio Buttons – Allows user to disable or enable the serial flow control • Submit Button – Once submitted all of the above changes will take effect

3.6.15. BS and SS Common Pages

3.6.15.1. Save Settings Button

The Save Settings button works the same for both the base station and subscriber. The Save Settings button saves any changes that have been made on the GUI to the database with the exception of the items mentioned in default database section that do not go back to default. All modifications made and set take effect automatically at the time they are set with the exception of generating start scripts. But, the changes do not get saved to the database unless the Save Settings button is selected and confirmed. Changes made and not saved to the database will be lost on a reboot or a power cycle. The save settings occurs for all screens simultaneously and not just for the screen currently being browsed. A confirmation popup appears when the Save Settings button gets selected.

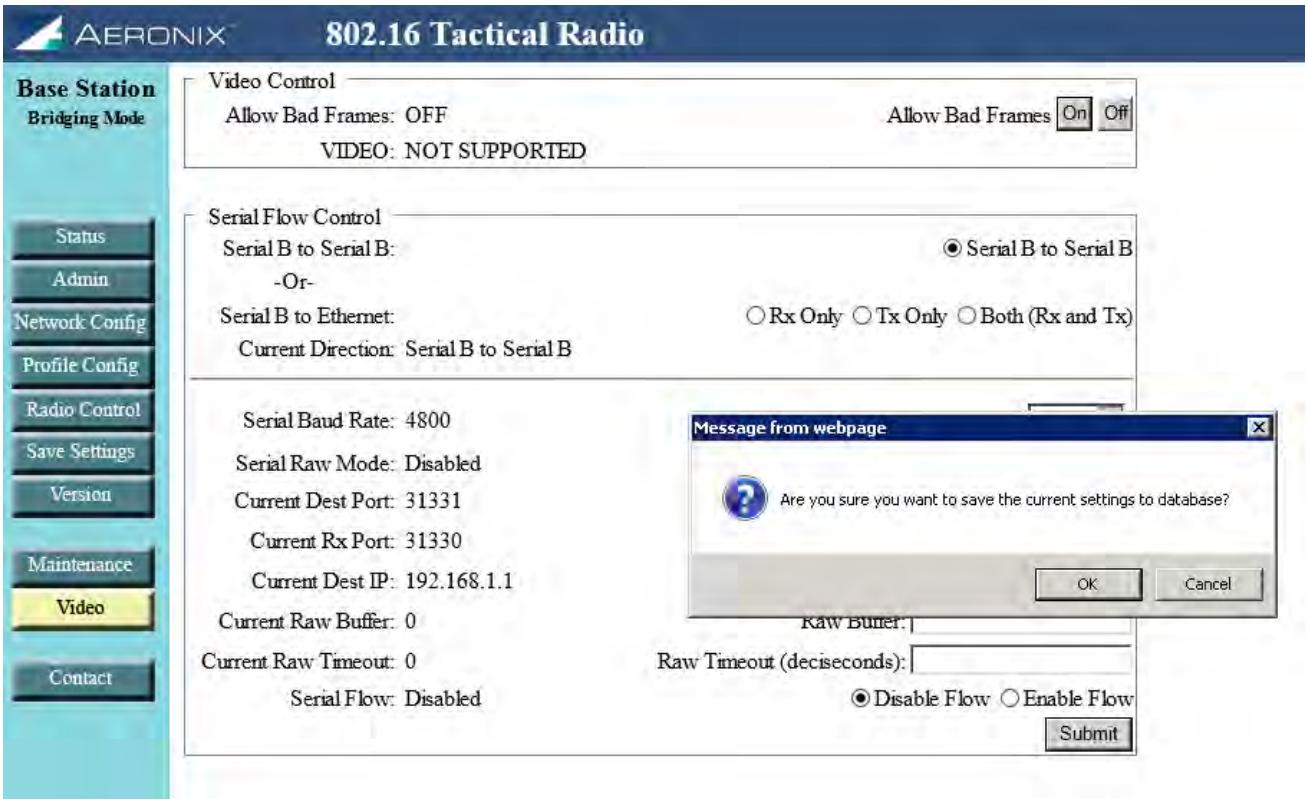


Figure 3-56 Save Settings Selection

3.6.15.2. Version Page

The Version page may be viewed by an authenticated or unauthenticated viewer by selecting the Version button. The base station and subscriber station Version page contains the same content.

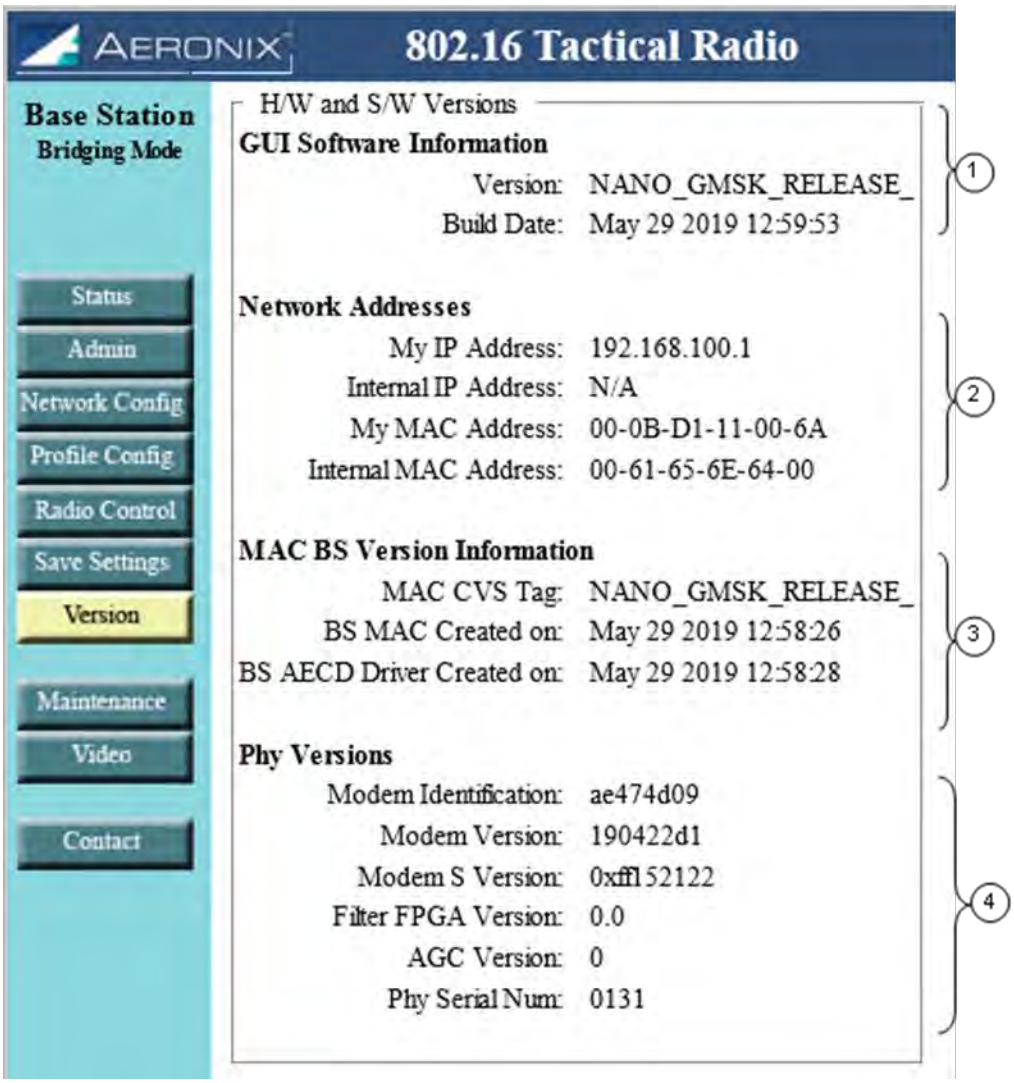


Figure 3-57 Version Page

Table 3-23. Version Page

	Field/Control	Description
1	GUI Software Information	Version, date and time of the GUI build on the unit. OFDM software releases will have 80216 string in the version. The GMSK software releases will have the GMSK string in the version.
2	Network Addresses	• My IP Address - Ethernet IP Address of the unit. • Internal IP Address - MAC S/W network interface IP Address. There is no internal IP address in the bridging mode

Table 3-23. Version Page		
	Field/Control	**Description**
		• My MAC Address - MAC H/W Address of unit. • Internal MAC Address - Internal MAC address of the unit used by the MAC S/W.
3	MAC BS/SS Version Information	Version, date and time of the MAC S/W build on the unit. OFDM software releases will have 80216 string in the MAC CVS tag. The GMSK software releases will have the GMSK string in the MAC CVS tag.
4	Phy Versions	Firmware versions for PHY and Serial Number of PHY hardware

3.6.15.3. Maintenance Page Reboot Button

The reboot button in the More Options section allows the administrator to reboot the unit remotely. The administrator must confirm the reboot.

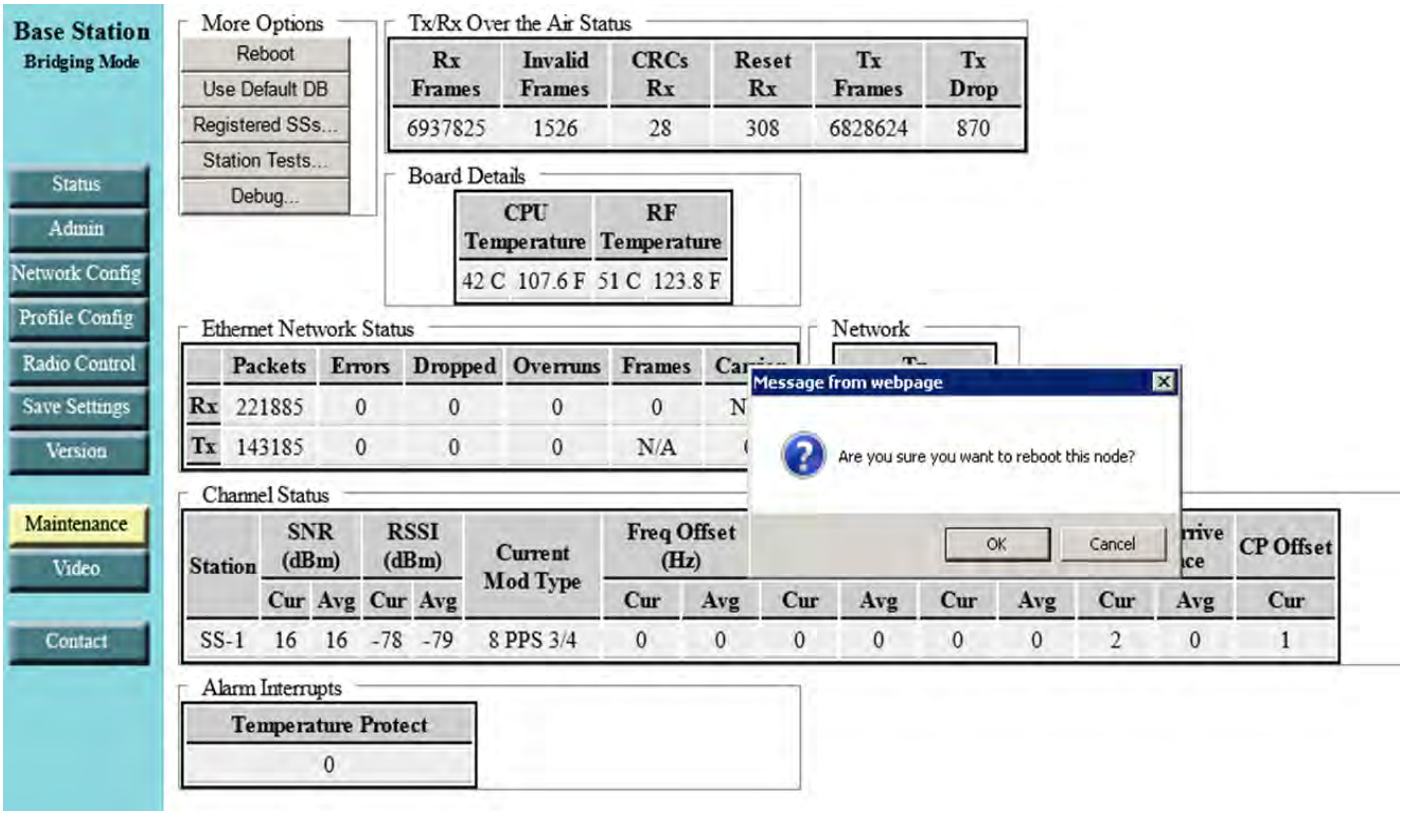


Figure 3-58 Maintenance Reboot Selection

3.6.15.4. Maintenance Page Use Default DB

The Use Default DB button allows the user to revert to the factory default settings. Most saved configuration settings are lost once the user confirms the selection. A warning message pops up after selecting the Use Default DB button (Figure 3-59 Maintenance Use Default DB Selection). The node will reboot once the user confirms their selection.

Note: Settings that do not go back to default:
Start script information
Transec keys

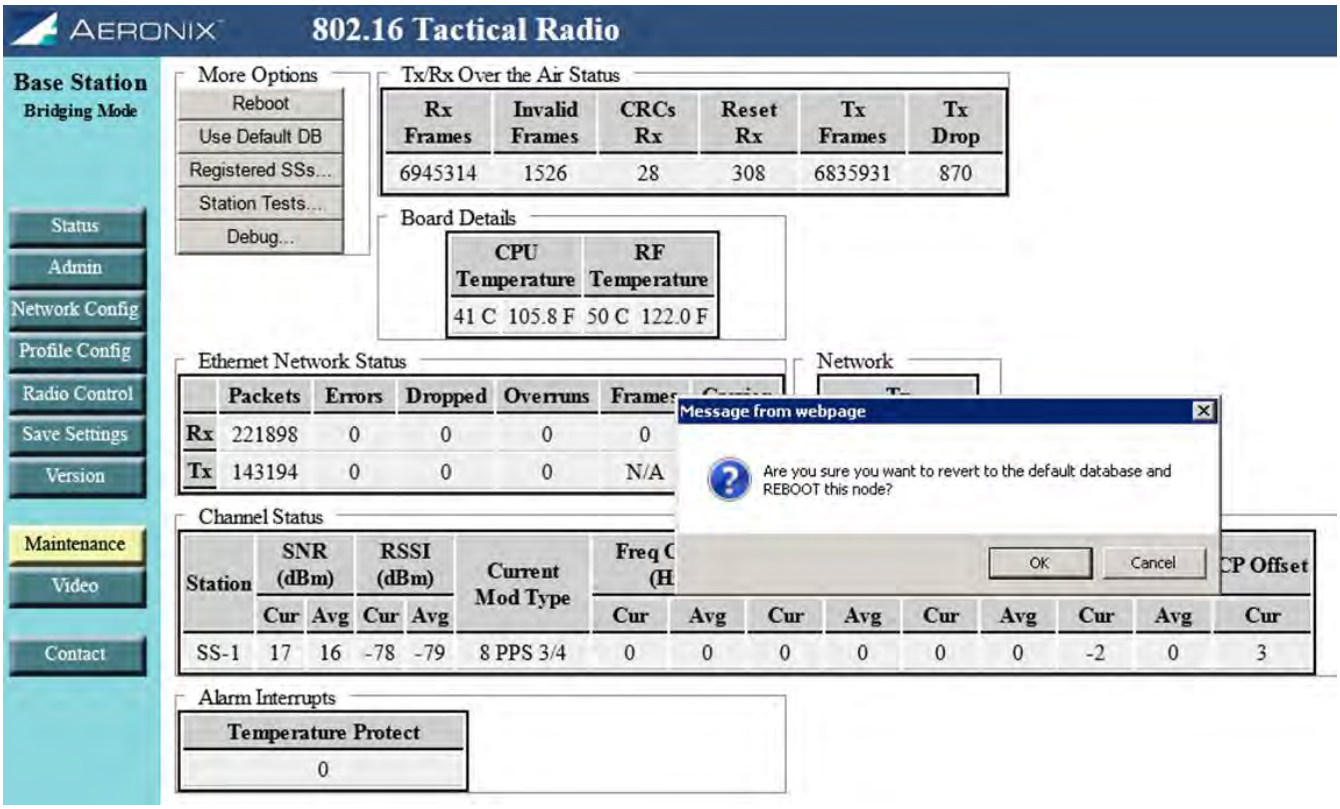


Figure 3-59 Maintenance Use Default DB Selection

3.6.15.5. Maintenance System Tests Page

Station tests can be configured by selecting the Station Tests button on the More Options section of the Maintenance page. The Station Tests page provides the capability to execute a continuous sine wave test or a bit error rate test. When either of the tests are executing, the normal operation of the unit is not functional. Upon test completion the unit should return to normal operation.

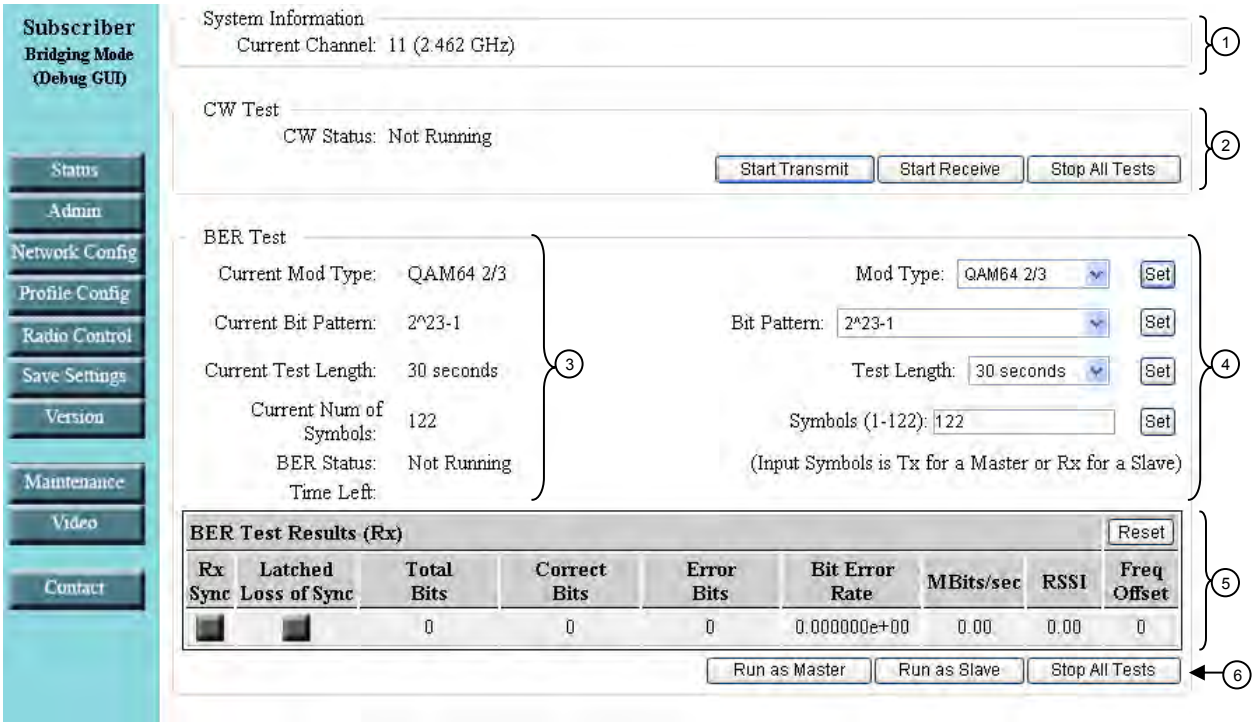


Figure 3-60 OFDM Maintenance System Tests Page

Table 3-24. Maintenance System Tests Page		
	Field/Control	**Description**
1	System Information	• Current Channel – Displays the current channel and frequency of the unit. This defines the channel where test execution occurs.
2	CW Test	Continuous Sine Wave Test • CW Status – Displays the state of the test o Not Running o Transmitting… o Receiving…(RSSI x) if actually receiving a CW then the RSSI level is displayed • Start Transmit button – Starts the CW transmission • Start Receive button – Puts the unit in receive mode to capture the CW transmission from another unit. • Stop All Tests – Stop the test from running and returns the unit to normal operation.
3	BER Test Current settings	• Current Mod Type - Displays the current modulation setting for test execution.

Table 3-24. Maintenance System Tests Page		
	Field/Control	**Description**
		• Current Bit Pattern – Bit pattern to use in BER test • Current Test Length – The current length of time the test executes. • Current Num Of Symbols – Displays the current number of symbols in a frame to use for test execution. • BER Status – Status of test o Not Running o Running as Master o Running as Slave • Time left – Displays a countdown for the length of the test, if the length is "Until Stopped" this field displays a "∞" symbol
4	BER Test	Bit Error Rate Test requires a unit configured as the master and a unit configured as the slave for proper execution and results. • Mod Type – Allows the administrator to change the modulation for test execution. The drop down list provides the selections. • Set button – Changes the modulation to the value chosen from the Mod Type drop down list. • Bit Pattern – Allows the administrator to select a new bit pattern for test execution. The drop down list provides the selections. • Set button – Changes the bit pattern to the value chosen from the Bit Pattern drop down list. • Test Length – Allows the administrator to select a new test length. The drop down list provides the selections. • Set button - Changes the test execution length to the value chosen from the Test Length drop down list. • Symbols (0 –122) – Allows the administrator to provide a new number of symbols for test execution. The configured numbers of symbols much match on the master and slave. • BER Status – Displays the current state of the test. o Not Running o Running as Master o Running as Slave

Table 3-24. Maintenance System Tests Page		
	Field/Control	**Description**
		Time left – Displays a countdown for the test execution time left. If the Test Length selected is Until Stopped this field displays a ∞ symbol.
4	BER Test Results (Rx)	If the number of symbols to receive by the unit is not 0, the BER Test Results table contains the results after test execution completes. Test execution comprises of the master unit transmitting for the number of symbols selected and the slave listening for those symbols and the slave transmitting for the same number of symbols and the master listening for those symbols. This occurs for the value selected in the Test Length field. • Reset Button – Resets the test result fields in the table. • Rx Sync light – Appears green if the unit receives sync from the transmitter. • Latched Loss of Sync – Appears green if unit loses sync. • Total Bits – Displays the total bits received by the unit during test execution. • Correct Bits – Displays the number of correct bits during test execution. • Error Bits – Displays the number of bits with errors during test execution. • Bit Error Rate – Displays ratio of non-correct bits versus correct bits • Mbits/sec – Displays the number of bits received divided by the number of seconds the test executed. • RSSI – Provides the received signal strength. • Freq Offset – Provides the frequency offset.
5	BER Mode Buttons	• Run as Master button – Starts the test in master mode. • Run as Slave button – start test in Slave mode • Stop All Tests button – Stops the BER test execution and returns the unit to normal operation.

3.6.15.6. Maintenance Debug Page

The Debug page can be accessed by selecting the Debug button in the More Options section of the Maintenance page. It provides additional debug capabilities that can be used for lab testing and in the field

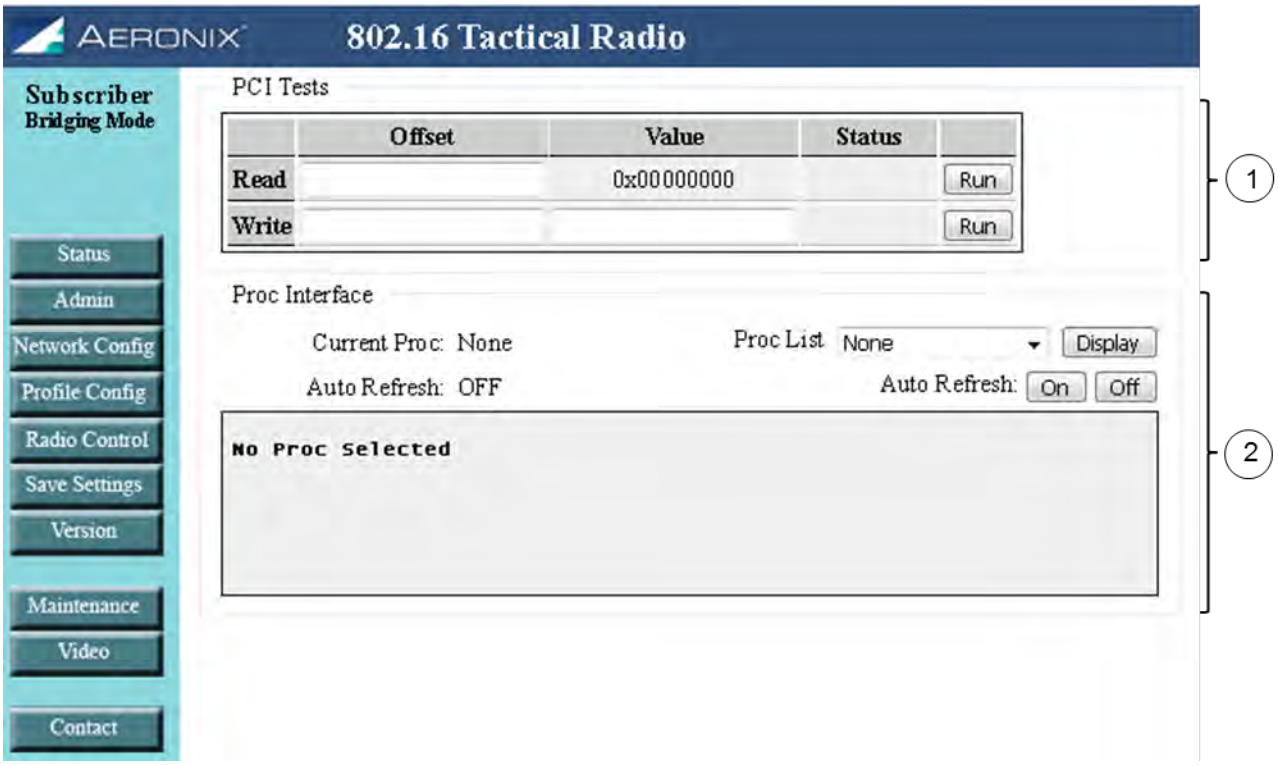


Figure 3-61 Maintenance Debug Page

Table 3-25. Maintenance Debug Page

	Field/Control	Description
1	PCI Tests	Writing values to FPGA registers can render the unit nonoperational and could leave the unit in a condition where a reboot or power cycle is requires. Values must be preceded by a 0x. • Read - Provides the capability to input an offset to an FPGA register and to read the value by selecting the Run button to the right of the row. • Write - Provides the capability to change the value of an FPGA register by supplying the offset of the register and the value in the appropriate locations. Selecting the Run button to the right of the row updates the value.
2	Proc Interface	The proc interface provides supplies debug information for the different interfaces in the unit.

Table 3-25. Maintenance Debug Page		
	Field/Control	**Description**
		The proc interface is for detailed debugging usually handled by the unit developers. Therefore each parameter will not be described in detail in this document. • Proc List – Provides a pull down list of available proc interfaces for viewing. • Display Button – Displays the debug information for the interface chosen in the Proc List drop down and displays it in display box below • Auto Refresh – o On – Enable auto refresh which will update the debug interface every second o Off – Disable auto refresh, the display will only be updated when user clicks on the Display button

3.6.15.7. Contact Page

The contact page provides contact information for the 802.16 product.

3.7. SNMP Interface configuration

The user can use SNMP to set most the radios configuration. Aeronix SNMP agent is compatible with SNMPv3.

Aeronix has used the following third party applications to interact with our SNMP agent.

- net-snmp – suite of applications to implement SNMP
- SNMPc – GUI application for SNMP
- iReasoning MIB browser

Aeronix is listed under the enterprise OID,

The Aeronix OID is 1.3.6.1.4.17701.10.1.x.x, where x.x is the object identifier that is defined in the Aeronix MIB, see

Appendix A: SNMP MIB.
Click the advanced button next to the *Address* field. Enter "user" for USM User, "SHA" for Authentication and "DES" for privacy. Then enter "AERONIX1%" for the SHA and DES passwords.

3.7.1. SNMP User Authentication

The Aeronix User-based Security name is "user". The Aeronix authentication protocol is "SHA". The privacy type is "DES". The privacy key or pass phrase to use is "AERONIX1%".

3.7.2. SNMP examples

Below are some examples on how to query and modify Aeronix tables. The examples are using the net-snmp library of commands.

3.7.2.1. Basic SNMP get request

```
snmpget -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> <item>

Example:
snmpget -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 txrxEnable.1
```

3.7.2.2. Basic SNMP set request

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> <item> <item type> <value>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 txrxEnable.1 i 0
```

3.7.2.3. Basic SNMP Table request

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsSsNameTable
```

3.7.2.4. SNMP Save Database

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m $PWD/AERONIX-802.16-MIB.mib.txt 192.168.8.1
savedatabase.1 u 1
```

3.7.2.5. SNMP start script

If a data object is not referenced in the snmpset command then it will default to what it was previously set to. This could be what the system default is, or if the value has been changed by the user then the value will remain what the user has set since boot.

```
Configure Start Script:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m AERONIX-802.16-MIB.mib.txt 192.168.2.1 basestation.1
unsigned 0 ssId.1 unsigned 2 webgui.1 unsigned 0 udhcpc.1
unsigned 0 bridging.1 unsigned 1 sysIP4Address.1 string
192.168.2.1 sysIP4Netmask.1 string 255.255.0.0
```

Create start script

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m AERONIX-802.16-MIB.mib.txt 192.168.2.1
makeStartScript.1 u 1
```

3.7.2.6. SNMP frequency

Frequency is in Hertz units.

```
Display Frequency
snmpget -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m AERONIX-802.16-MIB.mib.txt 192.168.100.1 customFreq.1

Configure Frequency
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m AERONIX-802.16-MIB.mib.txt 192.168.100.1 customFreq.1
s 1800000000
```

Note Frequency is entered in Hertz

3.7.2.7. SNMP Bandwidth

```
Display Bandwidth
snmpget -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m AERONIX-802.16-MIB.mib.txt 192.168.100.1
freqBandwidth.1

Configure Bandwidth
```

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m AERONIX-802.16-MIB.mib.txt 192.168.100.1
freqBandwidth.1 u 14000000
```

3.7.2.8. SNMP add service class

This is only done on the BS.

- Poll service class table (wmanIfBsServiceClassTable):

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsServiceClassTable

Expected Output:
wmanIfBsQoSProfileIndex      wmanIfBsQosServiceClassName bitArray
wmanIfBsQoSTrafficPriority wmanIfBsQoSMaxSustainedRate wmanIfBsQoSMinReservedRate
wmanIfBsQosScSchedulingType wmanIfBsQoSServiceClassRowStatus

1 DL_MULTICAST.....................  ?   1   6096000 b/s  3657600 b/s bestEffort active
2 initial..........................  ?   1   5885696 b/s  1000 b/s    bestEffort active
3 little...........................  ?   1   100000  b/s  1000 b/s    bestEffort active
4 middle...........................  ?   1   2500000 b/s  1000 b/s    bestEffort active
5 BE_MAX...........................  ?   1   50000000 b/s 1000000 b/s bestEffort active
```

- Add row to basestation service class:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfBsQoSServiceClassRowStatus.<row
index> i <row status (4 = create, 6 = destroy>
```

Since there are five entries in the service class table, to create a new table we reference the sixth.

```
Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfBsQoSServiceClassRowStatus.6 i 4

Expected output:
AERONIX-80216-MIB::wmanIfBsQoSServiceClassRowStatus.6 = INTEGER:
createAndGo(4)
```

- Fill in row:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfBsQosServiceClassName.<row index> s
"<custom service name>" wmanIfBsQoSTrafficPriority.<row index> i
<priority [0-7]> wmanIfBsQoSMaxSustainedRate.<row index> u
```

```
<value in bits> wmanIfBsQoSMinReservedRate.<row index> u <value
in bits> wmanIfBsQosScSchedulingType.<row index> i <scheduling
type [2-5]>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfBsQosServiceClassName.6 s "data"
wmanIfBsQoSTrafficPriority.6 i 1 wmanIfBsQoSMaxSustainedRate.6 u
2500005 wmanIfBsQoSMinReservedRate.6 u 1000005
wmanIfBsQosScSchedulingType.6 i 2

Expected output:
AERONIX-80216-MIB::wmanIfBsQosServiceClassName.6 = STRING: data
AERONIX-80216-MIB::wmanIfBsQoSTrafficPriority.6 = INTEGER: 1
AERONIX-80216-MIB::wmanIfBsQoSMaxSustainedRate.6 = Gauge32:
2500005 b/s
AERONIX-80216-MIB::wmanIfBsQoSMinReservedRate.6 = Gauge32:
1000005 b/s
AERONIX-80216-MIB::wmanIfBsQosScSchedulingType.6 = INTEGER:
bestEffort(2)
```

- Verify new row added and active:

```
If new row is added but NOT active, verify all values were set
correctly.
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> wmanIfBsServiceClassTable

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user ¬-m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsServiceClassTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsServiceClassTable

 wmanIfBsQoSProfileIndex      wmanIfBsQosServiceClassName bitArray
wmanIfBsQoSTrafficPriority wmanIfBsQoSMaxSustainedRate wmanIfBsQoSMinReservedRate
wmanIfBsQosScSchedulingType wmanIfBsQoSServiceClassRowStatus

1 DL_MULTICAST.....................  ?   1   6096000 b/s  3657600 b/s bestEffort active
2 initial..........................  ?   1   5885696 b/s  1000 b/s    bestEffort active
3 little...........................  ?   1   100000  b/s  1000 b/s    bestEffort active
4 middle...........................  ?   1   2500000 b/s  1000 b/s    bestEffort active
5 BE_MAX...........................  ?   1   50000000 b/s 1000000 b/s bestEffort active
6 data.............................  ?   1   2500005 b/s  1000005 b/s bestEffort active
```

3.7.2.9. Remove service class:

- Poll service class table (wmanIfBsServiceClassTable):

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsServiceClassTable

Expected Output:
SNMP table: AERONIX-80216-MIB::wmanIfBsServiceClassTable

 wmanIfBsQoSProfileIndex      wmanIfBsQosServiceClassName bitArray
wmanIfBsQoSTrafficPriority wmanIfBsQoSMaxSustainedRate wmanIfBsQoSMinReservedRate
wmanIfBsQosScSchedulingType wmanIfBsQoSServiceClassRowStatus
1 DL_MULTICAST.................  ?   1   6096000 b/s  3657600 b/s bestEffort active
2 initial......................  ?   1   5885696 b/s  1000 b/s    bestEffort active
3 little.......................  ?   1   100000  b/s  1000 b/s    bestEffort active
4 middle.......................  ?   1   2500000 b/s  1000 b/s    bestEffort active
5 BE_MAX.......................  ?   1   50000000 b/s 1000000 b/s bestEffort active
6 data.........................  ?   1   2500005 b/s  1000005 b/s bestEffort active
```

- Remove row from basestation service class:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfBsQoSServiceClassRowStatus.<row
index> i <row status (4 = create, 6 = destroy>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfBsQoSServiceClassRowStatus.6 i 6

Expected output:
AERONIX-80216-MIB::wmanIfBsQoSServiceClassRowStatus.6 = INTEGER:
destroy(6)
```

- Verify row deletion:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsServiceClassTable
```

```
Expected Output:
wmanIfBsQoSProfileIndex      wmanIfBsQosServiceClassName bitArray
wmanIfBsQoSTrafficPriority wmanIfBsQoSMaxSustainedRate wmanIfBsQoSMinReservedRate
wmanIfBsQosScSchedulingType wmanIfBsQoSServiceClassRowStatus
wmanIfBsQoSProfileIndex      wmanIfBsQosServiceClassName bitArray
wmanIfBsQoSTrafficPriority wmanIfBsQoSMaxSustainedRate wmanIfBsQoSMinReservedRate
wmanIfBsQosScSchedulingType wmanIfBsQoSServiceClassRowStatus
1 DL_MULTICAST.................  ?   1   6096000 b/s  3657600 b/s bestEffort active
2 initial......................  ?   1   5885696 b/s  1000 b/s    bestEffort active
3 little.......................  ?   1   100000  b/s  1000 b/s    bestEffort active
4 middle.......................  ?   1   2500000 b/s  1000 b/s    bestEffort active
5 BE_MAX.......................  ?   1   50000000 b/s 1000000 b/s bestEffort active
```

3.7.2.10. Viewing/Adding Subscribers

This is only done on the BS.

- **Viewing Subscriber Table**

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSsNameTable
```

- **Create new Subscriber Entry**

Assuming we already have two entries we must first create a new row. After running the below command it will create a placeholder entry with Default as the name with a mac address of 00:00:00:00:00:00

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSsNameRowStatus.3 i 4
```

Then we fill in the MAC address and name separately.

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSsNameSubscriberName.3 s SS-3
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSsNameMACAddress.3 s 112233445566
```

- **Removing a Subscriber Entry**

Remove first entry in table, .1 after wmanIfBsSsNameRowStatus is the row index.

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSsNameRowStatus.1 i 6
```

3.7.2.11. Add new provisioned SFs

This is only done on the BS.

- Poll provisioned SFs table (wmanIfBsProvisionedSfTable):

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsProvisionedSfTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsProvisionedSfTable

wmanIfBsSfId wmanIfBsSsProvmacAddress wmanIfBsSfDirection
wmanIfBsServiceClassName wmanIfBsSfState wmanIfBsSfUlMulticast
wmanIfBsProvisionedSfRowStatus
4    00:61:65:6E:64:01.     downstream  BE_MAX……   authorized       off       active
5    00:61:65:6E:64:01.     upstream    BE_MAX…     authorized       off       active
6    00:61:65:6E:64:02.     downstream  BE_MAX…     authorized       off       active
7    00:61:65:6E:64:02.     upstream    BE_MAX…     authorized       off       active
```

- Add row to provisioned SFs:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfBsProvisionedSfRowStatus.<row index>
i <row status (4 = create, 6 = destroy>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfBsProvisionedSfRowStatus.5 i 4

Expect output:
AERONIX-80216-MIB::wmanIfBsProvisionedSfRowStatus.5 = INTEGER:
createAndGo(4)
```

- Verify new row created in provisioned SFs table:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsProvisionedSfTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsProvisionedSfTable

 wmanIfBsSfId wmanIfBsSsProvmacAddress wmanIfBsSfDirection
wmanIfBsServiceClassName wmanIfBsSfState wmanIfBsSfUlMulticast
wmanIfBsProvisionedSfRowStatus
4       00:61:65:6E:64:01.  downstream BE_MAX…      authorized     off        active
```

```
5       00:61:65:6E:64:01.  upstream   BE_MAX…      authorized     off        active
6       00:61:65:6E:64:02.  downstream BE_MAX…      authorized     off        active
7       00:61:65:6E:64:02.  upstream   BE_MAX…      authorized     off        active
8       FF:FF:FF:FF:FF:FF.  downstream …             255           off        notReady
```

- Fill in row:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfBsSsProvmacAddress.<row index> s <ss
mac address> wmanIfBsServiceClassName.<row index> s <name of
service class>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfBsSsProvmacAddress.5 s 0061656E6401
wmanIfBsServiceClassName.5 s middle

Expected output:
AERONIX-80216-MIB::wmanIfBsSsProvmacAddress.5 = STRING:
0061656E6401
AERONIX-80216-MIB::wmanIfBsServiceClassName.5 = STRING: middle
```

- Verify row has been added to provisioned SFs:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsProvisionedSfTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsProvisionedSfTable

 wmanIfBsSfId wmanIfBsSsProvmacAddress wmanIfBsSfDirection
wmanIfBsServiceClassName wmanIfBsSfState wmanIfBsSfUlMulticast
wmanIfBsProvisionedSfRowStatus
4       00:61:65:6E:64:01.  downstream BE_MAX…      authorized     off        active
5       00:61:65:6E:64:01.  upstream   BE_MAX…      authorized     off        active
6       00:61:65:6E:64:02.  downstream BE_MAX…      authorized     off        active
7       00:61:65:6E:64:02.  upstream   BE_MAX…      authorized     off        active
8       00:61:65:6E:64:01.  downstream middle…      authorized     off        active
9       00:61:65:6E:64:01.  upstream   middle…      authorized     off        active
```

3.7.2.12. Remove provisioned SF:

- Poll provisioned SF table (wmanIfBsProvisionedSfTable):

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

When removing a provisioned SF row, you need only remove the
downstream row. The corresponding upstream row will
automatically be deleted as well.

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsProvisionedSfTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsProvisionedSfTable

 wmanIfBsSfId wmanIfBsSsProvmacAddress wmanIfBsSfDirection
wmanIfBsServiceClassName wmanIfBsSfState wmanIfBsSfUlMulticast
wmanIfBsProvisionedSfRowStatus
4       00:61:65:6E:64:01.  downstream BE_MAX…      authorized     off        active
5       00:61:65:6E:64:01.  upstream   BE_MAX…      authorized     off        active
6       00:61:65:6E:64:02.  downstream BE_MAX…      authorized     off        active
7       00:61:65:6E:64:02.  upstream   BE_MAX…      authorized     off        active
8       00:61:65:6E:64:01.  downstream middle…      authorized     off        active
9       00:61:65:6E:64:01.  upstream   middle…      authorized     off        active
```

- Remove provisioned SF row:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfBsProvisionedSfRowStatus.<row index>
i <row status (4 = create, 6 = destroy>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/mbersing/projects/nano/nano/mac/SNMP/AERONIX-
802.16-MIB.mib.txt 192.168.100.1
wmanIfBsProvisionedSfRowStatus.5 i 6

Expected output:
AERONIX-80216-MIB::wmanIfBsProvisionedSfRowStatus.5 = INTEGER:
destroy(6)
```

- Verify provisioned SF row removed:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsProvisionedSfTable
```

```
Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsProvisionedSfTable

 wmanIfBsSfId wmanIfBsSsProvmacAddress wmanIfBsSfDirection
wmanIfBsServiceClassName wmanIfBsSfState wmanIfBsSfUlMulticast
wmanIfBsProvisionedSfRowStatus
4       00:61:65:6E:64:01.  downstream BE_MAX…      authorized     off        active
5       00:61:65:6E:64:01.  upstream   BE_MAX…      authorized     off        active
6       00:61:65:6E:64:02.  downstream BE_MAX…      authorized     off        active
7       00:61:65:6E:64:02.  upstream   BE_MAX…      authorized     off        active
```

3.7.2.13. Add new downstream burst profile:

- Poll downstream burst profile table (wmanIfBsOfdmDcdBurstProfileTable):

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsOfdmDcdBurstProfileTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsOfdmDcdBurstProfileTable

 wmanIfCmnOfdmDcdBurstProfIndex wmanIfCmnOfdmDiucValue wmanIfCmnOfdmDcdFecCodeType
wmanIfCmnOfdmDiucCinrMandatoryExitThresh wmanIfCmnOfdmDiucCinrMinEntryThresh
wmanIfCmnOfdmDiucFecMandatoryExitThresh wmanIfCmnOfdmDiucFecMinEntryThresh
wmanIfCmnOfdmAvailableType wmanIfCmnOfdmDcdBurstProfilerowStatus
1       1         fecBPSK-12        0       7     0     3       2          active
2       2         fecQPSK-12        8      10    15     4       2          active
3       3         fecQPSK-34        9      12     4     1       2          active
4       4         fecQAM-1612      12      16    28     7       2          active
5       5         fecQAM-1634      16      18     8     2       2          active
6       6         fecQAM-6423      20      23    25     6       2          active
```

- Add row to downstream burst profile:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfCmnOfdmDcdBurstProfilerowStatus.<row
index> i <row status (4 = create, 6 = destroy>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfCmnOfdmDcdBurstProfilerowStatus.7 i 4

Expect output:
AERONIX-80216-MIB::wmanIfCmnOfdmDcdBurstProfilerowStatus.7 =
INTEGER: createAndGo(4)
```

- Verify new row created in provisioned SFs table:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsOfdmDcdBurstProfileTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsOfdmDcdBurstProfileTable

 wmanIfCmnOfdmDcdBurstProfIndex wmanIfCmnOfdmDiucValue wmanIfCmnOfdmDcdFecCodeType
wmanIfCmnOfdmDiucCinrMandatoryExitThresh wmanIfCmnOfdmDiucCinrMinEntryThresh
wmanIfCmnOfdmDiucFecMandatoryExitThresh wmanIfCmnOfdmDiucFecMinEntryThresh
wmanIfCmnOfdmAvailableType wmanIfCmnOfdmDcdBurstProfilerowStatus
1       1         fecBPSK-12        0       7     0     3       2           active
2       2         fecQPSK-12        8      10    15     4       2           active
3       3         fecQPSK-34        9      12     4     1       2           active
4       4         fecQAM-1612      12      16    28     7       2           active
5       5         fecQAM-1634      16      18     8     2       2           active
6       6         fecQAM-6423      20      23    25     6       2           active
invalid 0         0                 0       0     0     notReady
```

- Fill in row:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfCmnOfdmDcdFecCodeType.<row index> i
<fec code [0-9]> wmanIfCmnOfdmDiucCinrMandatoryExitThresh.<row
index> i <cinr exit threshold>
wmanIfCmnOfdmDiucFecMandatoryExitThresh.<row index> i <fec exit
threshold> wmanIfCmnOf
dmAvailableType.<row index> i <enable=2, disable=3>
wmanIfCmnOfdmDiucCinrMinEntryThresh.<row index> i <cinr entry
threshold> wmanIfCmnOfdmDiucFecMinEntryThresh.<row index> i <fec
entry threshold>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfCmnOfdmDcdFecCodeType.7 i 6
wmanIfCmnOfdmDiucCinrMandatoryExitThresh.7 i 8
wmanIfCmnOfdmDiucFecMandatoryExitThresh.7 i 10 wmanIfCmnOf
dmAvailableType.7 i 2 wmanIfCmnOfdmDiucCinrMinEntryThresh.7 i 0
wmanIfCmnOfdmDiucFecMinEntryThresh.7 i 9

Expected output:
AERONIX-80216-MIB::wmanIfCmnOfdmDcdFecCodeType.7 = INTEGER:
fecQAM-6434(6)
```

```
AERONIX-80216-MIB::wmanIfCmnOfdmDiucCinrMandatoryExitThresh.7 =
INTEGER: 8
AERONIX-80216-MIB::wmanIfCmnOfdmDiucFecMandatoryExitThresh.7 =
INTEGER: 10
AERONIX-80216-MIB::wmanIfCmnOfdmAvailableType.7 = INTEGER: 2
AERONIX-80216-MIB::wmanIfCmnOfdmDiucCinrMinEntryThresh.7 =
INTEGER: 0
AERONIX-80216-MIB::wmanIfCmnOfdmDiucFecMinEntryThresh.7 =
INTEGER: 9
```

- Verify row has been added to downstream burst profile:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsOfdmDcdBurstProfileTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsOfdmDcdBurstProfileTable

 wmanIfCmnOfdmDcdBurstProfIndex wmanIfCmnOfdmDiucValue wmanIfCmnOfdmDcdFecCodeType
wmanIfCmnOfdmDiucCinrMandatoryExitThresh wmanIfCmnOfdmDiucCinrMinEntryThresh
wmanIfCmnOfdmDiucFecMandatoryExitThresh wmanIfCmnOfdmDiucFecMinEntryThresh
wmanIfCmnOfdmAvailableType wmanIfCmnOfdmDcdBurstProfilerowStatus
1       1         fecBPSK-12        0       7     0     3       2           active
2       2         fecQPSK-12        8      10    15     4       2           active
3       3         fecQPSK-34        9      12     4     1       2           active
4       4         fecQAM-1612      12      16    28     7       2           active
5       5         fecQAM-1634      16      18     8     2       2           active
6       6         fecQAM-6423      20      23    25     6       2           active
7       7         fecQAM-6434      21      24    15     0       2           active
```

3.7.2.14. Remove downstream burst profile:

- Poll downstream burst profile table (wmanIfBsOfdmDcdBurstProfileTable):

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Example:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsOfdmDcdBurstProfileTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsOfdmDcdBurstProfileTable
```

```
 wmanIfCmnOfdmDcdBurstProfIndex wmanIfCmnOfdmDiucValue wmanIfCmnOfdmDcdFecCodeType
wmanIfCmnOfdmDiucCinrMandatoryExitThresh wmanIfCmnOfdmDiucCinrMinEntryThresh
wmanIfCmnOfdmDiucFecMandatoryExitThresh wmanIfCmnOfdmDiucFecMinEntryThresh
wmanIfCmnOfdmAvailableType wmanIfCmnOfdmDcdBurstProfilerowStatus
1       1         fecBPSK-12        0       7     0     3       2           active
2       2         fecQPSK-12        8      10    15     4       2           active
3       3         fecQPSK-34        9      12     4     1       2           active
4       4         fecQAM-1612      12      16    28     7       2           active
5       5         fecQAM-1634      16      18     8     2       2           active
6       6         fecQAM-6423      20      23    25     6       2           active
7       7         fecQAM-6434      21      24    15     0       2           active
```

- Remove downstream burst profile row:

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m <mib> <ip> wmanIfCmnOfdmDcdBurstProfilerowStatus.<row
index> i <row status (4 = create, 6 = destroy>

Example:
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m /home/general/SNMP/AERONIX-802.16-MIB.mib.txt
192.168.100.1 wmanIfCmnOfdmDcdBurstProfilerowStatus.7 i 6

Expected output:
AERONIX-80216-MIB::wmanIfCmnOfdmDcdBurstProfilerowStatus.7 =
INTEGER: destroy(6)
```

- Verify downstream burst profile row removed:

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m <mib> <ip> <table name>

Exampled:
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m /home/general/SNMP/AERONIX-802.16-
MIB.mib.txt 192.168.100.1 wmanIfBsOfdmDcdBurstProfileTable

Expected output:
SNMP table: AERONIX-80216-MIB::wmanIfBsOfdmDcdBurstProfileTable

 wmanIfCmnOfdmDcdBurstProfIndex wmanIfCmnOfdmDiucValue wmanIfCmnOfdmDcdFecCodeType
wmanIfCmnOfdmDiucCinrMandatoryExitThresh wmanIfCmnOfdmDiucCinrMinEntryThresh
wmanIfCmnOfdmDiucFecMandatoryExitThresh wmanIfCmnOfdmDiucFecMinEntryThresh
wmanIfCmnOfdmAvailableType wmanIfCmnOfdmDcdBurstProfilerowStatus
1       1         fecBPSK-12        0       7     0     3       2           active
2       2         fecQPSK-12        8      10    15     4       2           active
3       3         fecQPSK-34        9      12     4     1       2           active
4       4         fecQAM-1612      12      16    28     7       2           active
5       5         fecQAM-1634      16      18     8     2       2           active
6       6         fecQAM-6423      20      23    25     6       2           active
```

3.7.2.15. Viewing/Creating/Deleting Classifiers

This is only done on the BS.

- **Viewing classifier rule table**

```
snmptable -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X
Aeronix1% -u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsClassifierRuleTable
```

- **Creating classifier MAC rule entry**

1) Verify SNMP is classifier mod is set to mac, if not set classifierMacAddrMode.1 to 1.

```
classifierMacAddrMode :
```

0 = IP Mode
1 = MAC Mode

```
snmpget -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
classifierMacAddrMode.1
```

2) Create entry row

Assuming an empty classifier table first index will start at 1.

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsClassifierRuleRowStatus.1 i 4
```

3) Fill in new entry row

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSfIndex.1 i 4
```

wmanIfBsClassifierRuleDestMacAddr.1 s <xxxxxxxxxxxx>
wmanIfBsClassifierRuleSourceMacMask.1 s <xxxxxxxxxxxx>
wmanIfBsClassifierRuleSourceMacAddr.1 s <xxxxxxxxxxxx>
wmanIfBsClassifierRuleDestMacMask.1 s <xxxxxxxxxxxx>

4) Create entry row

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsClassifierRuleRowStatus.2 i 4
```

5) Fill in new entry row

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSfIndex.2 i 5
```

wmanIfBsClassifierRuleDestMacAddr.2 s <xxxxxxxxxxxx>
wmanIfBsClassifierRuleSourceMacMask.2 s <xxxxxxxxxxxx>
wmanIfBsClassifierRuleSourceMacAddr.2 s <xxxxxxxxxxxx>
wmanIfBsClassifierRuleDestMacMask.2 s <xxxxxxxxxxxx>

- **Creating classifier IP rule entry**

1) Verify SNMP is classifier mod is set to IP, if not set classifierMacAddrMode.1 to 0

```
snmpget -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
classifierMacAddrMode.1
classifierMacAddrMode:
```

0 = IP Mode
1 = MAC Mode

2) Create entry row

Assuming an empty classifier table first index will start at 1.

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsClassifierRuleRowStatus.1 i 4
```

3) Fill in new entry row

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSfIndex.1 i 4 wmanIfBsClassifierRuleIpSourceAddr.1 s
192.168.100.2 wmanIfBsClassifierRuleIpSourceMask.1 s 255.255.0.0
wmanIfBsClassifierRuleIpDestAddr.1 s 192.168.1.2
wmanIfBsClassifierRuleIpDestMask.1 s 255.255.0.0
```

4) Create entry row

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsClassifierRuleRowStatus.2 i 4
```

5) Fill in new entry row

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsSfIndex.2 i 5 wmanIfBsClassifierRuleIpSourceAddr.2 s
192.168.1.2 wmanIfBsClassifierRuleIpSourceMask.2 s 255.255.0.0
wmanIfBsClassifierRuleIpDestAddr.2 s 192.168.100.2
wmanIfBsClassifierRuleIpDestMask.2 s 255.255.0.0
```

- **Removing classifier rule entry**

Remove first entry in table, .1 after wmanIfBsClassifierRuleRowStatus is the row index

```
snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m ./AERONIX-802.16-MIB.mib.txt 192.168.100.1
wmanIfBsClassifierRuleRowStatus.1 i 6
```

3.7.2.16. SNMP configure Serial Flow

```
Serial Flow Types enumerations (unsigned) :
Serial B to Serial B = 0
```

```
Serial B To Ethernet Rx Only = 1
Serial B To Ethernet Tx Only = 2
Serial B To Ethernet Both Rx and Tx = 3

Serial Baud Rates (unsigned) : 4800, 9600, 19200, 38400, 57600,
115200

Raw Mode enumeration (integer):
Raw Mode  = 1
Normal = 0

snmpset -v 3 -l AUTHPRIV -a SHA -A Aeronix1% -x DES -X Aeronix1%
-u user -m $PWD/AERONIX-802.16-MIB.mib.txt 192.168.8.1
serialDirection.1 u 0 serialBaudRate.1 u 4800
enaSerialTransfer.1 i 1 serialTxPort.1 u 3000 serialRxPort.1 u
3001 serialTxIP.1 s "192.168.2.10" serialRawMode.1 i 0

Response:
AERONIX-80216-MIB::serialDirection.1 = Gauge32: 0
AERONIX-80216-MIB::serialBaudRate.1 = Gauge32: 4800
AERONIX-80216-MIB::enaSerialTransfer.1 = INTEGER: enable(1)
AERONIX-80216-MIB::serialTxPort.1 = Gauge32: 3000
AERONIX-80216-MIB::serialRxPort.1 = Gauge32: 3001
AERONIX-80216-MIB::serialTxIP.1 = STRING: 192.168.2.10
AERONIX-80216-MIB::serialRawMode.1 = INTEGER: 0
```

3.7.3. SNMP Unsupported configurations

3.7.3.1. BERT and CW test

SNMP does not allow the user to run BERT and CW test.

4. Determining NANO Radio configuration

If the user is not aware of the Radio IP or the radio mode, below are some methods by which they can learn the current configuration the NANO radio.

4.1. Two ways to determine radio IP address

4.1.1. Use Console

1) Attach USB to Serial to a Laptop and use Console to get IP address
2) Connect to Serial A port with terminal emulator. (Baud = 115200, 8-N-1 no flow control)
3) Login by typing “root”
4) Enter password “1234”.

5) Enter “ifconfig”
6) Take note of the “mybridge” interface IP address and netmask

4.1.2. Use radio’s default IP address

The user can always access the radio over its Ethernet alias IP address of 169.254.1.1. This IP is consistent so if the user needs to access the GUI or SNMP.

1) Configure laptop/computer with an IP address on the same subnet as radio default IP address for example:
    a. laptop IP : 169.254.1.10
    b. laptop mask : 255.255.255.0
2) Connect radio’s Ethernet to laptop/computer
3) Bring up a web browser (IE or firefox)
4) Type in radio’s default IP address at web browser
    a. 169.254.1.1
5) The Status page should be displayed of the radio.
6) Multiple ways to view IP address
    a. Version page
        i. Go to the Version page of the webgui
        ii. The IP address will be displayed under network Addresses
    b. Maintenance page
        i. Go to Maintenance page
        ii. Click on Debug button
        iii. Go to Proc Interface section
        iv. From scroll down menu select “ifconfig”
        v. Hit Display Button
        vi. The IP address and mask will be listed under the “mybridge” interface
    c. Admin Page
        i. Go to the Admin Page
        ii. Look at the left side of the Generate Bridging Start script section
        iii. This is the current radio IP address, netmask, mode and ID

5. Change Radio IP address via webgui

The start script controls the IP address, IP masks, user interface (webgui or SNMP), and which mode the radio will boot in (BS or SS).

1) Bring up radios webgui by typing in current radio’s IP address on a Web Browser
2) Go to Admin page
3) Scroll down to the Generate Bridging Script
4) Select Base Station or Subscriber radio button
5) Enter Radio ID if desired
6) Enter new IPv4 IP address
7) Enter new IPv4 netmask
8) Click Generate Script button

9) Once the 'Generate Script' button is selected, the old start script will be overwritten with the new start script changes. Once a reboot or power cycle occurs the new changes will take effect.

6. Adding Subscriber example using webgui

This example shows the method for adding a third subscriber to the base station configuration. This example assumes that the radios are configured in bridging mode with IP classifiers and no forwarding classifiers configured. It also assumes the use of preexisting service classes. The picture below shows the physical configuration.

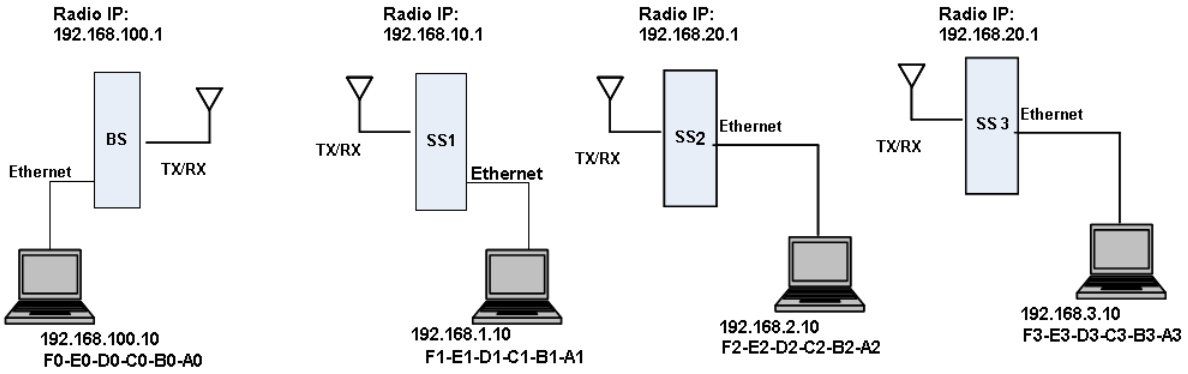


Figure 6-1 Example Adding Subscriber Layout

Adding a subscriber to the Station Names table:

1. Browse to the Admin page of the base station.
2. Click on the ADD ENTRY button in the Station Names section.
3. Add a name for the subscriber and the appropriate MAC address for subscriber number 3 which is 00-61-65-6E-64-03.
    - Note: The MAC Address needed on the Admin for a Subscriber is the Aeronix MAC address for that subscriber (00-61-65-64-XX where XX is the Subscriber number).
4. Click on the ADD button to the right of the subscriber entry.
5. The subscriber appears in the station names table as show below.

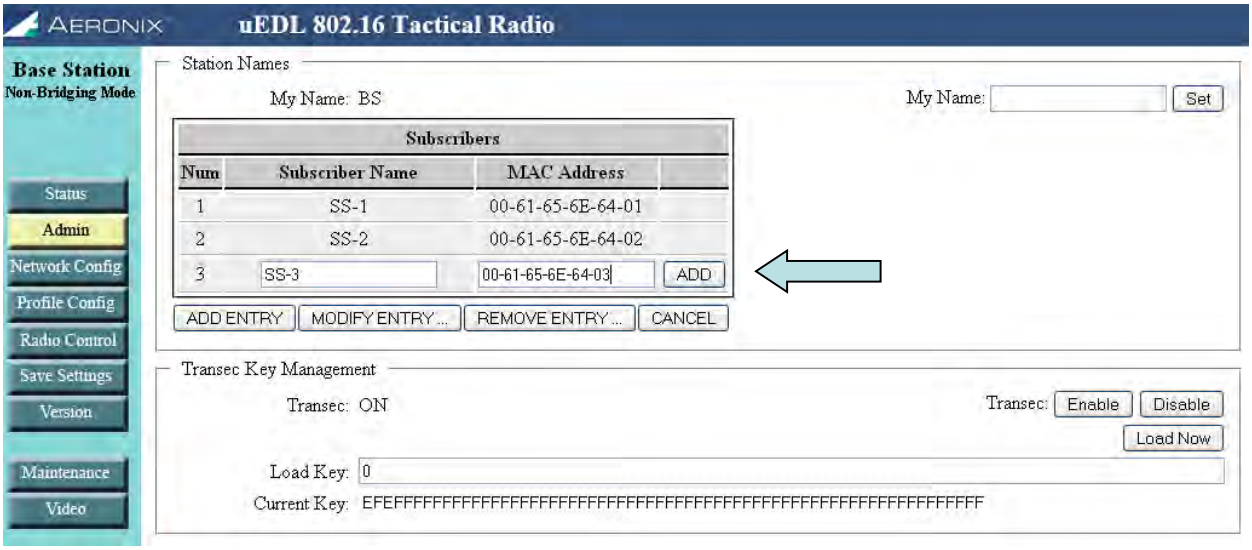


Figure 6-2 Example Adding a Subscriber station name

Add the service flows for the subscriber:

1. Browse to the Network Config page of the base station.
2. Click on the ADD ENTRY button in the Provisioned SFs section.
3. Select the subscriber name for subscriber 3 from the pull down menu.
4. Type the downstream SF Id to be used in the box.
5. Select the service class for the service flow from the pull down menu.
6. Click the ADD button to the right of the service flow entry.
7. The service flow entries appear in the provisioned service flow table as shown below.

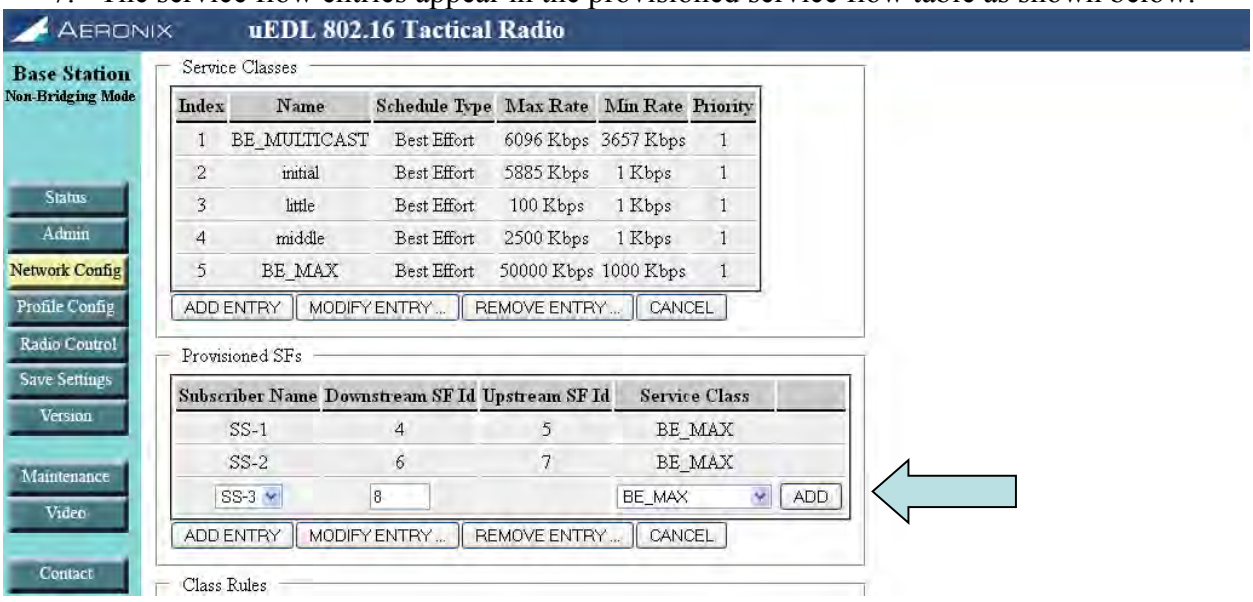


Figure 6-3 Example Adding a Subscriber Provisioned SF

At this point the subscriber should register with the base station if it is powered on, on the same channel (frequency), has the same TRANSEC state and key, Tx/Rx is on and has the same channelization.

Add the classifiers for the new service flows so that data can pass to and from the subscriber:

1. Browse to the Network Config page of the base station.
2. Click on the Auto Populate button in the Class Rules section.
3. Provide the proper Station IP and Station Mask for each entry in the table.
    a. Station IP BS – 192.168.100.1 Station Mask – 255.255.255.0
    b. Station IP SS1 – 192.168.1.1 Station Mask – 255.255.255.0
    c. Station IP SS2 – 192.168.2.1 Station Mask – 255.255.255.0
    d. Station IP SS3 – 192.168.3.1 Station Mask – 255.255.255.0
4. Click the Generate Classifiers button at the bottom of the class rules table.

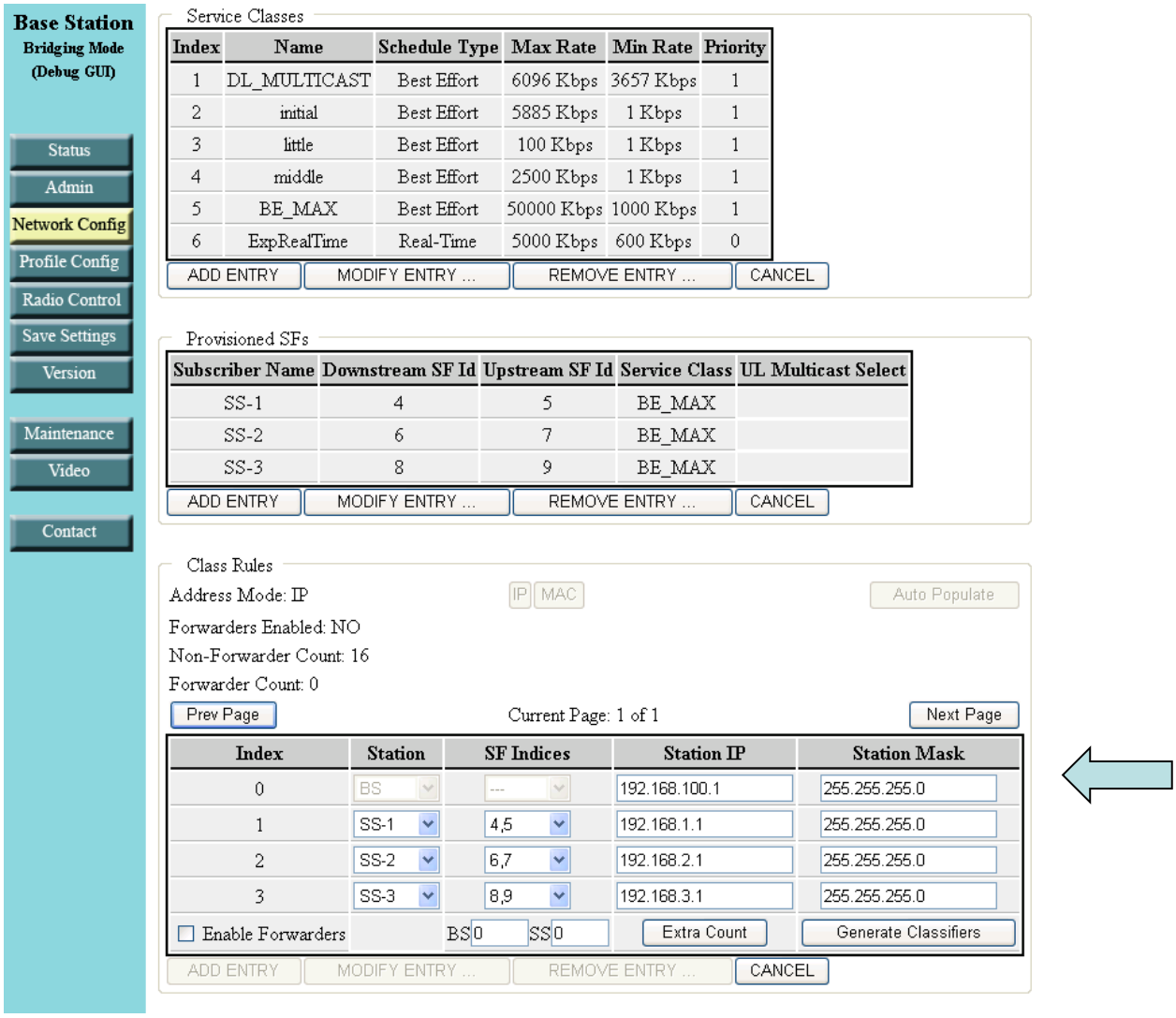


Figure 6-4 Example Adding a Subscriber Class Rules

5. Acknowledge the pop-up that the classifier regeneration takes some time.
6. The classifiers appear in the class rule table upon completion of the auto populate as shown below. These class rules and masks allow for multiple peripherals to connect in a unit's network and be able to pass data without modifying the class rule configuration.
7. Click on the Save Settings button and confirm the selection to save the subscriber 3 information to the database.

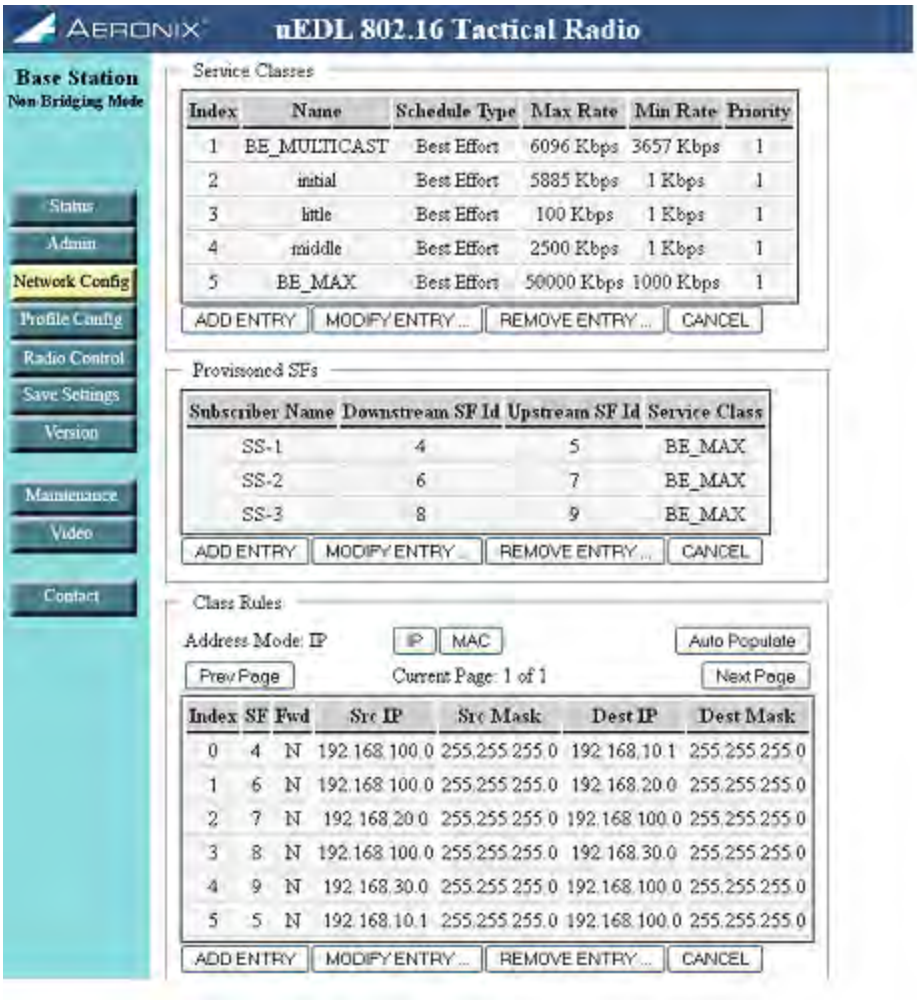


Figure 6-5 Example Adding a Subscriber Result

6.1. Configuring Multiple Service Flows

Multiple Service Flows can be assigned to a single Subscriber. This allows data from different IP address or MAC Address flowing from one radio to be assigned different Service Classes, which will give them different priorities. To configure a Subscriber to have Multiple Service flows, user must assign two different sets of Service Flows for the Subscriber. Then configure the classifiers using the different Service Flows. The user must disable the ARP Table feature on the Network Config screen.

If the user wanted to differentiate the data coming from different IP address from SS1, then multiple service flows can be used to prioritize the data. For example, the data coming from IP 192.168.1.10 to be treated as Real Time and data from IP 192.168.1.15 to be treated as Best Effort, then the configuration would look similar to below.

The user must setup and enable classifiers. The user cannot rely on the ARP Table feature to discover the correct Service flow to use for incoming IP traffic with different priorities.

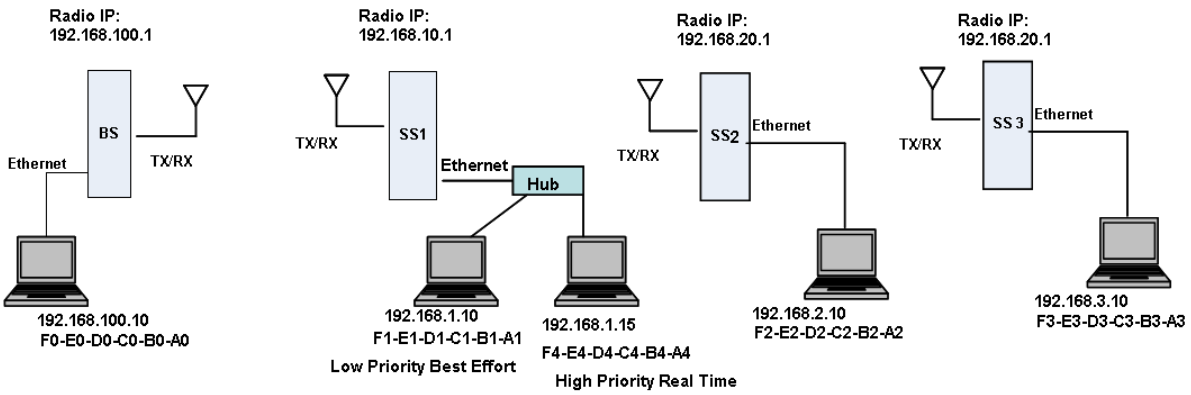


Figure 6-6 Example Multiple SF Layout

1. On the Base Station configure the Provisioned SF for the second service flow for SS1 (192.168.1.x network), using a Real Time Service Class (ExpRealTime).
2. We also selected the Real Time Service Flow (10,11) to be our UL Multicast Select, which means any multicast data will use this service flow and priority to transfer data.

Service Classes

Index	Name	Schedule Type	Max Rate	Min Rate	Priority
1	DL_MULTICAST	Best Effort	6096 Kbps	3657 Kbps	1
2	initial	Best Effort	5885 Kbps	1 Kbps	1
3	little	Best Effort	100 Kbps	1 Kbps	1
4	middle	Best Effort	2500 Kbps	1 Kbps	1
5	BE_MAX	Best Effort	50000 Kbps	1000 Kbps	1
6	ExpRealTime	Real-Time	5000 Kbps	600 Kbps	0

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

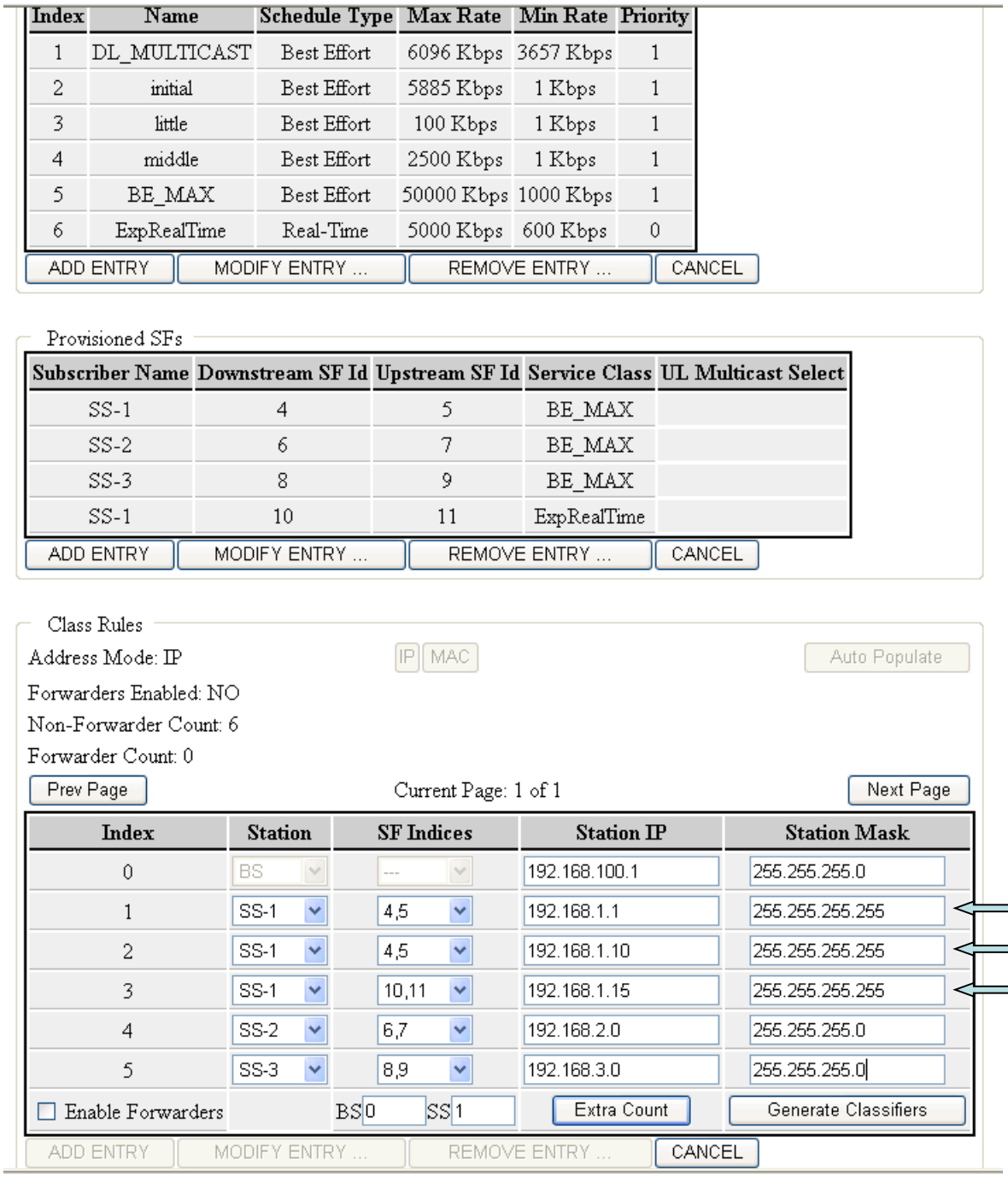
Provisioned SFs

Subscriber Name	Downstream SF Id	Upstream SF Id	Service Class	UL Multicast Select	
SS-1	4	5	BE_MAX		
SS-2	6	7	BE_MAX		
SS-3	8	9	BE_MAX		
SS-1	10	11	ExpRealTime	☑	SAVE

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Figure 6-7 Example Multiple SF Add SF

3. On the BS configure the classifiers for the new Service Flows, from our previous example our class rules for SS1 were masked off, the user will have to change this since the IPs are on the same subnet that we want to differentiate.
4. Use Auto Populate to redo the class rules table, SS1 IP address can no longer use the subnet masks.



Index	Name	Schedule Type	Max Rate	Min Rate	Priority
1	DL_MULTICAST	Best Effort	6096 Kbps	3657 Kbps	1
2	initial	Best Effort	5885 Kbps	1 Kbps	1
3	little	Best Effort	100 Kbps	1 Kbps	1
4	middle	Best Effort	2500 Kbps	1 Kbps	1
5	BE_MAX	Best Effort	50000 Kbps	1000 Kbps	1
6	ExpRealTime	Real-Time	5000 Kbps	600 Kbps	0

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Provisioned SFs

Subscriber Name	Downstream SF Id	Upstream SF Id	Service Class	UL Multicast Select
SS-1	4	5	BE_MAX	
SS-2	6	7	BE_MAX	
SS-3	8	9	BE_MAX	
SS-1	10	11	ExpRealTime	

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Class Rules

Address Mode: IP IP | MAC Auto Populate

Forwarders Enabled: NO

Non-Forwarder Count: 6

Forwarder Count: 0

Prev Page Current Page: 1 of 1 Next Page

Index	Station	SF Indices	Station IP	Station Mask
0	BS	---	192.168.100.1	255.255.255.0
1	SS-1	4,5	192.168.1.1	255.255.255.255
2	SS-1	4,5	192.168.1.10	255.255.255.255
3	SS-1	10,11	192.168.1.15	255.255.255.255
4	SS-2	6,7	192.168.2.0	255.255.255.0
5	SS-3	8,9	192.168.3.0	255.255.255.0

Enable Forwarders BS 0 SS 1 Extra Count Generate Classifiers

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

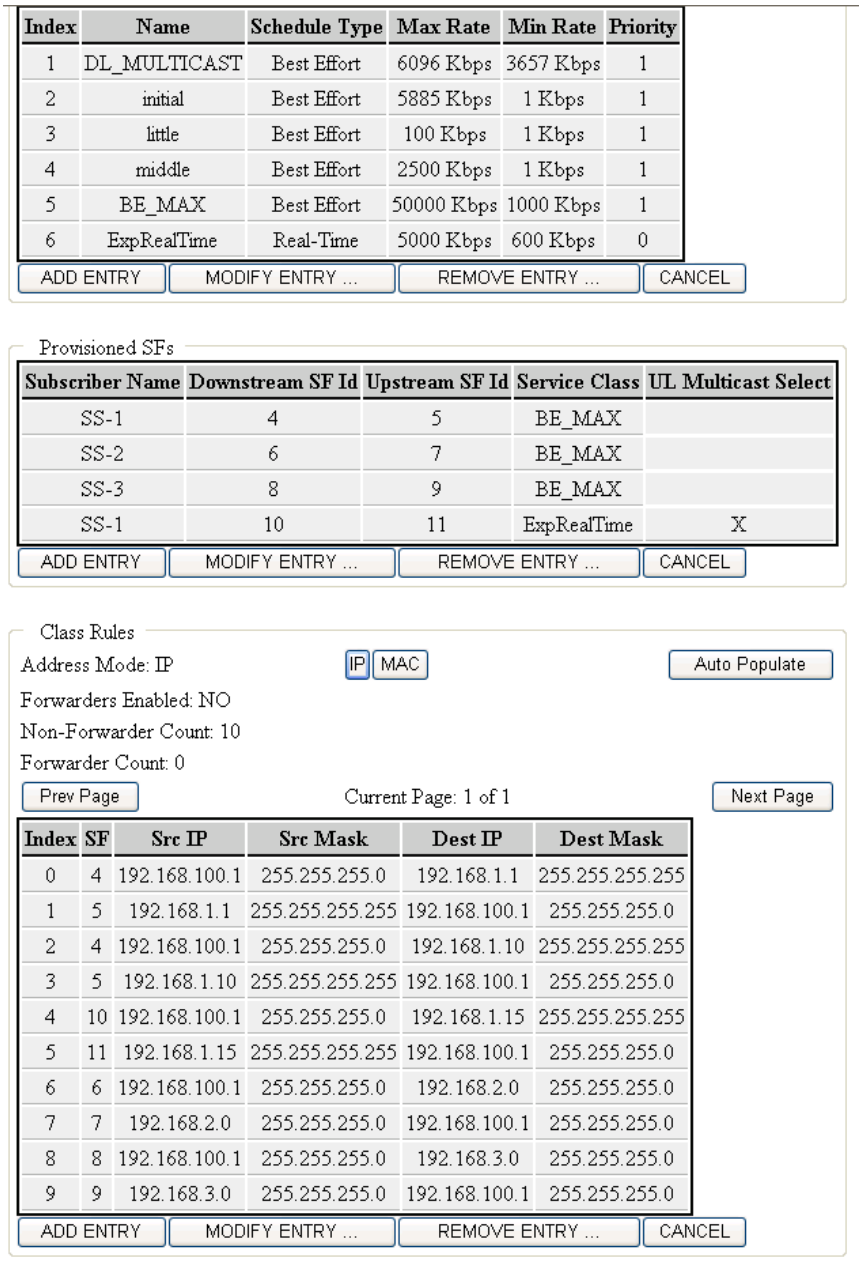
1 2 3

Figure 6-8 Example Multiple SF Add Class Rules

Table 6-1. Adding Multiple Service Flow Class Rule		
	Field/Control	**Description**
1	Radio Class Rule	This Class Rule is needed if the user wants to log into the SS1 Aeronix Webgui from any of the BS

Table 6-1. Adding Multiple Service Flow Class Rule		
	Field/Control	**Description**
		peripherals. Need a rule for IP packets flowing between 192.168.100.x <-> 192.168.1.1
2	Best Effort Class Rule	This Class Rule specifies that data with IP address 192.168.1.10 will be classified as Best Effort and transferred based on that priority.
3	Real Time Class Rule	This Class Rule specifies that data with IP address 192.168.1.15 will be classified as Best Effort and transferred based on that priority.

5. Below is the Class Rules after Clicking on the "Generate Classifiers" button.



Index	Name	Schedule Type	Max Rate	Min Rate	Priority
1	DL_MULTICAST	Best Effort	6096 Kbps	3657 Kbps	1
2	initial	Best Effort	5885 Kbps	1 Kbps	1
3	little	Best Effort	100 Kbps	1 Kbps	1
4	middle	Best Effort	2500 Kbps	1 Kbps	1
5	BE_MAX	Best Effort	50000 Kbps	1000 Kbps	1
6	ExpRealTime	Real-Time	5000 Kbps	600 Kbps	0

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Provisioned SFs

Subscriber Name	Downstream SF Id	Upstream SF Id	Service Class	UL Multicast Select
SS-1	4	5	BE_MAX	
SS-2	6	7	BE_MAX	
SS-3	8	9	BE_MAX	
SS-1	10	11	ExpRealTime	X

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Class Rules

Address Mode: IP IP MAC Auto Populate

Forwarders Enabled: NO

Non-Forwarder Count: 10

Forwarder Count: 0

Prev Page Current Page: 1 of 1 Next Page

Index	SF	Src IP	Src Mask	Dest IP	Dest Mask
0	4	192.168.100.1	255.255.255.0	192.168.1.1	255.255.255.255
1	5	192.168.1.1	255.255.255.255	192.168.100.1	255.255.255.0
2	4	192.168.100.1	255.255.255.0	192.168.1.10	255.255.255.255
3	5	192.168.1.10	255.255.255.255	192.168.100.1	255.255.255.0
4	10	192.168.100.1	255.255.255.0	192.168.1.15	255.255.255.255
5	11	192.168.1.15	255.255.255.255	192.168.100.1	255.255.255.0
6	6	192.168.100.1	255.255.255.0	192.168.2.0	255.255.255.0
7	7	192.168.2.0	255.255.255.0	192.168.100.1	255.255.255.0
8	8	192.168.100.1	255.255.255.0	192.168.3.0	255.255.255.0
9	9	192.168.3.0	255.255.255.0	192.168.100.1	255.255.255.0

ADD ENTRY | MODIFY ENTRY ... | REMOVE ENTRY ... | CANCEL

Figure 6-9 Example Multiple SF Result

6. Below are screen shots of how the multiple Service Flows are displayed on the BS (Figure 6-10 Example Multiple SF BS Status Page) Status page and the SS (**Error! Reference source not found.**) Status page.

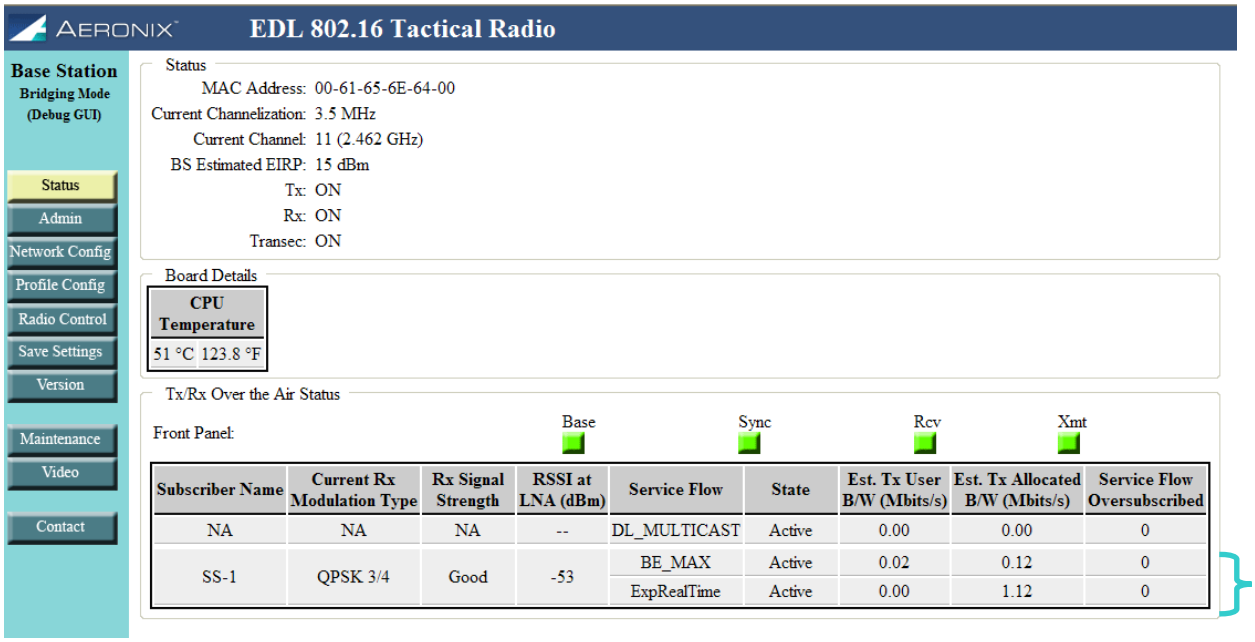


Figure 6-10 Example Multiple SF BS Status Page

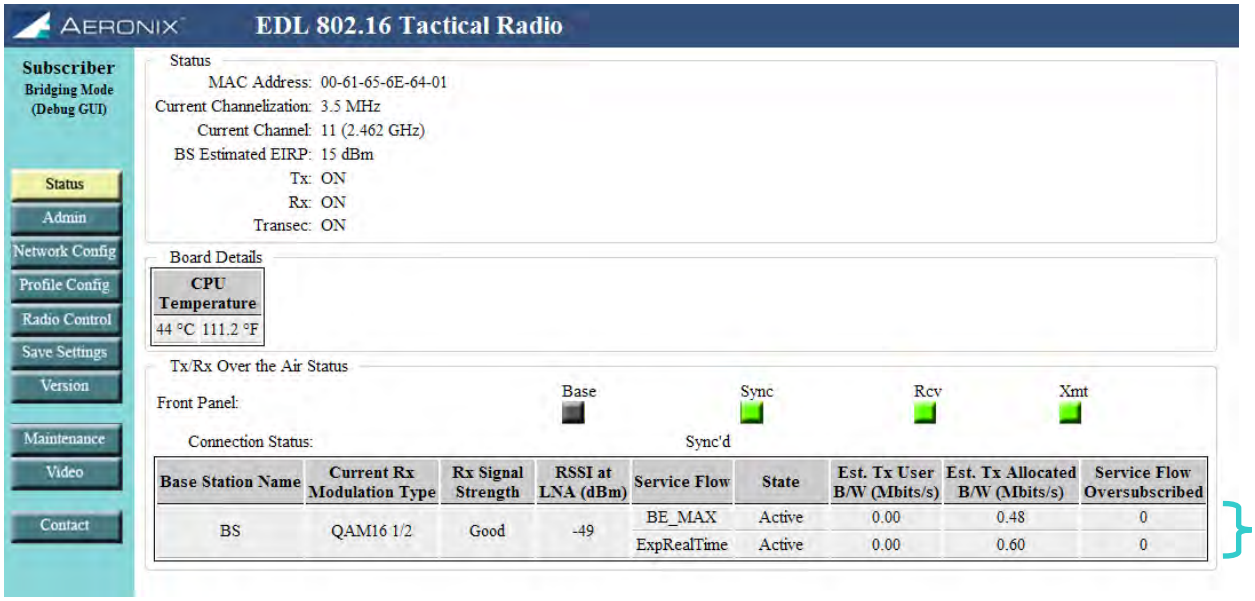


Figure 6-11 SS Example Multiple SF SS Status Page

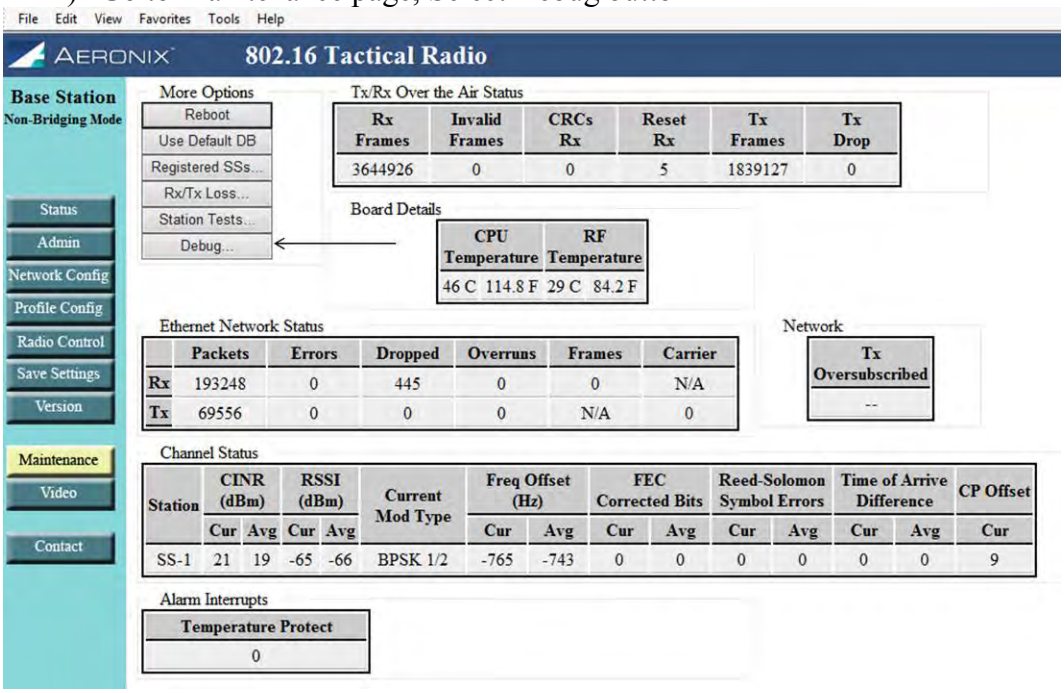
7. Aeronix Software Features

7.1. NANO output power

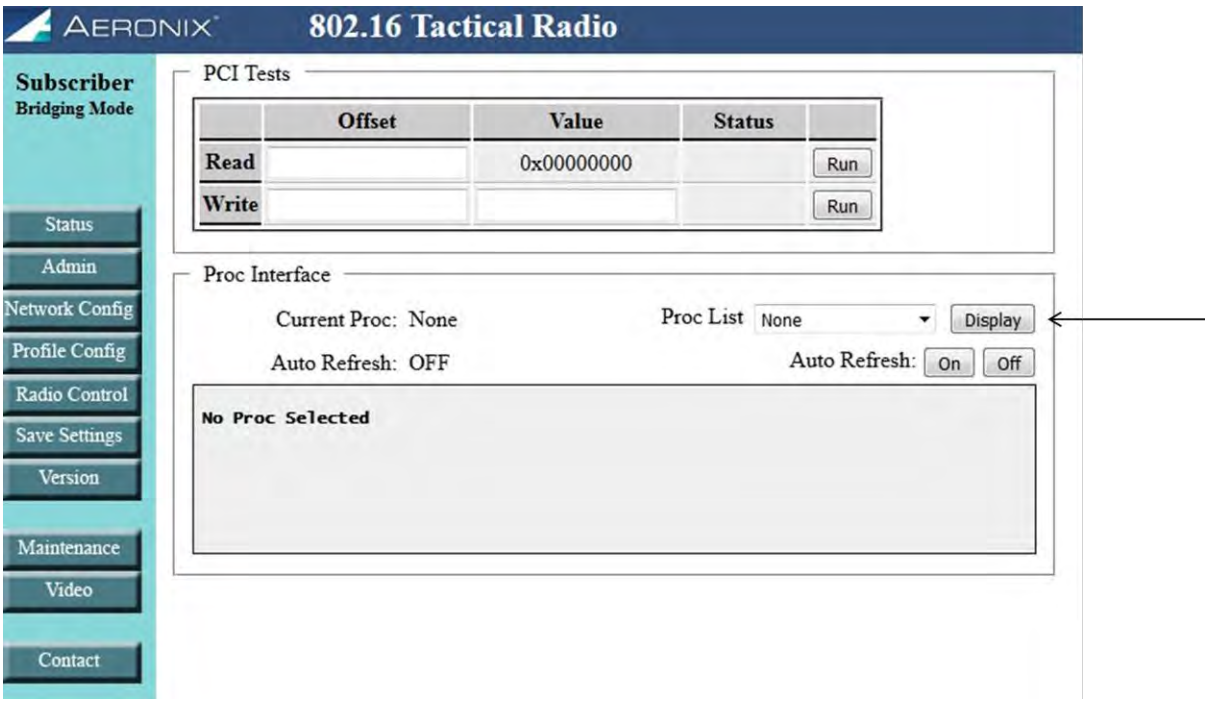
7.1.1. View NANO output power calibration

To view NANO transmit output power use the webgui to navigate to the Maintenance Page, then click on the “Debug” Button on the Maintenance Page.

1) Go to Maintenance page, Select Debug button



2) In the Proc Interface Section, use drop down menu to select “powerSettings” for GMSK radios. For OFDM radios select “phyEeprom”.



3) Click Display and the calibration table will be displayed.

7.1.1.1. OFDM Calibration tables

The OFDM NANO is calibrated to have two power schemes, High Power and High Data. The High power is calibrated to the highest power the radio achieve while maintaining an EVM that will satisfy QAM16 ¾ modulation. High Data is calibrated to a power that will maintain a modulation of QAM64 ¾. Each power schemes calibrates the power based on the output power of the OFDM preambles.

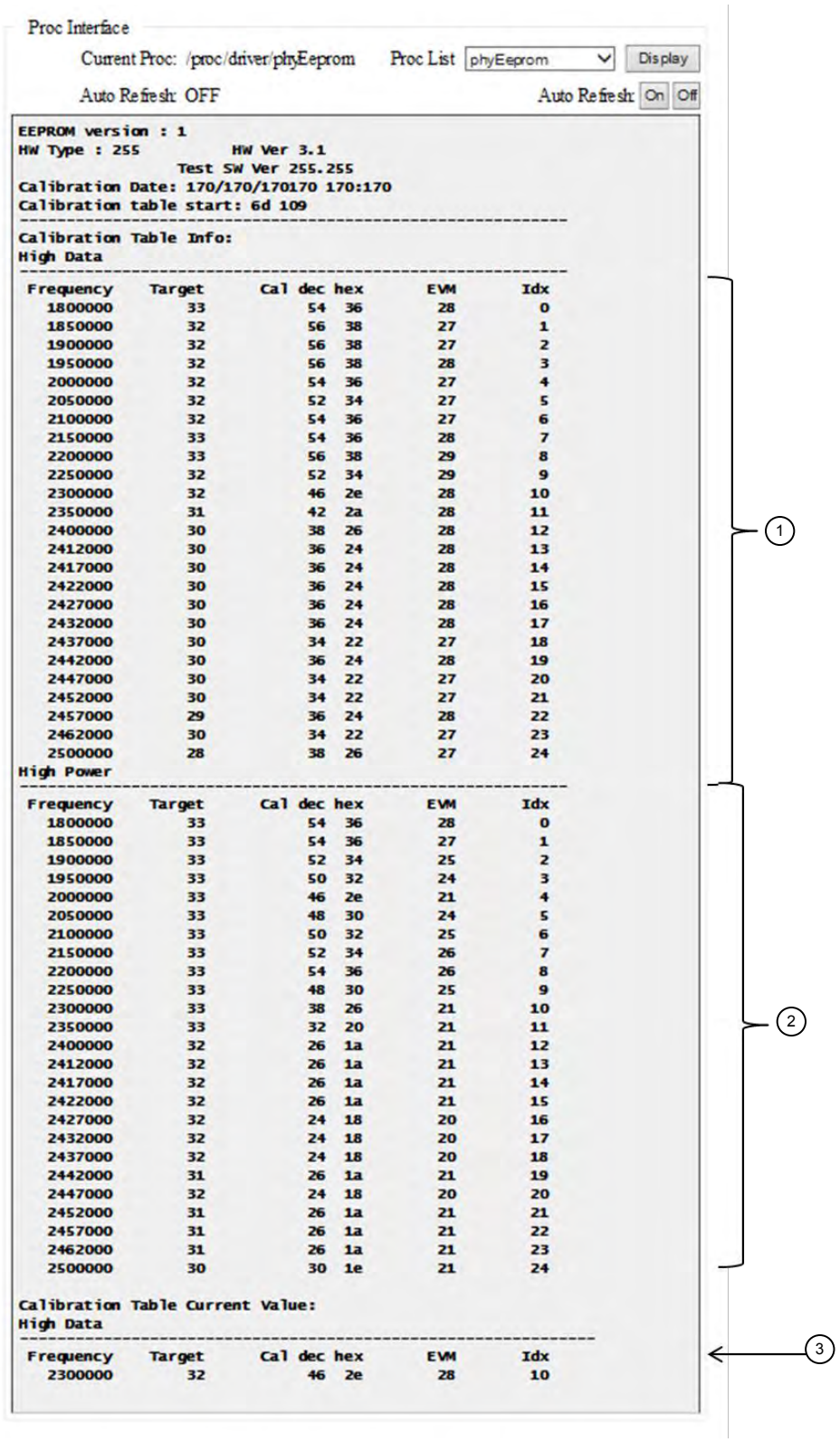


Figure 7-1 802.16 phyEeprom interface

7.1.1.2. GMSK Calibration tables

GMSK transmit power is calibrated to a High power and a Low power . The High power calibration is a target preamble output power 35 dBM. The GMSK Low power has a target preamble output power of 33 dBM.

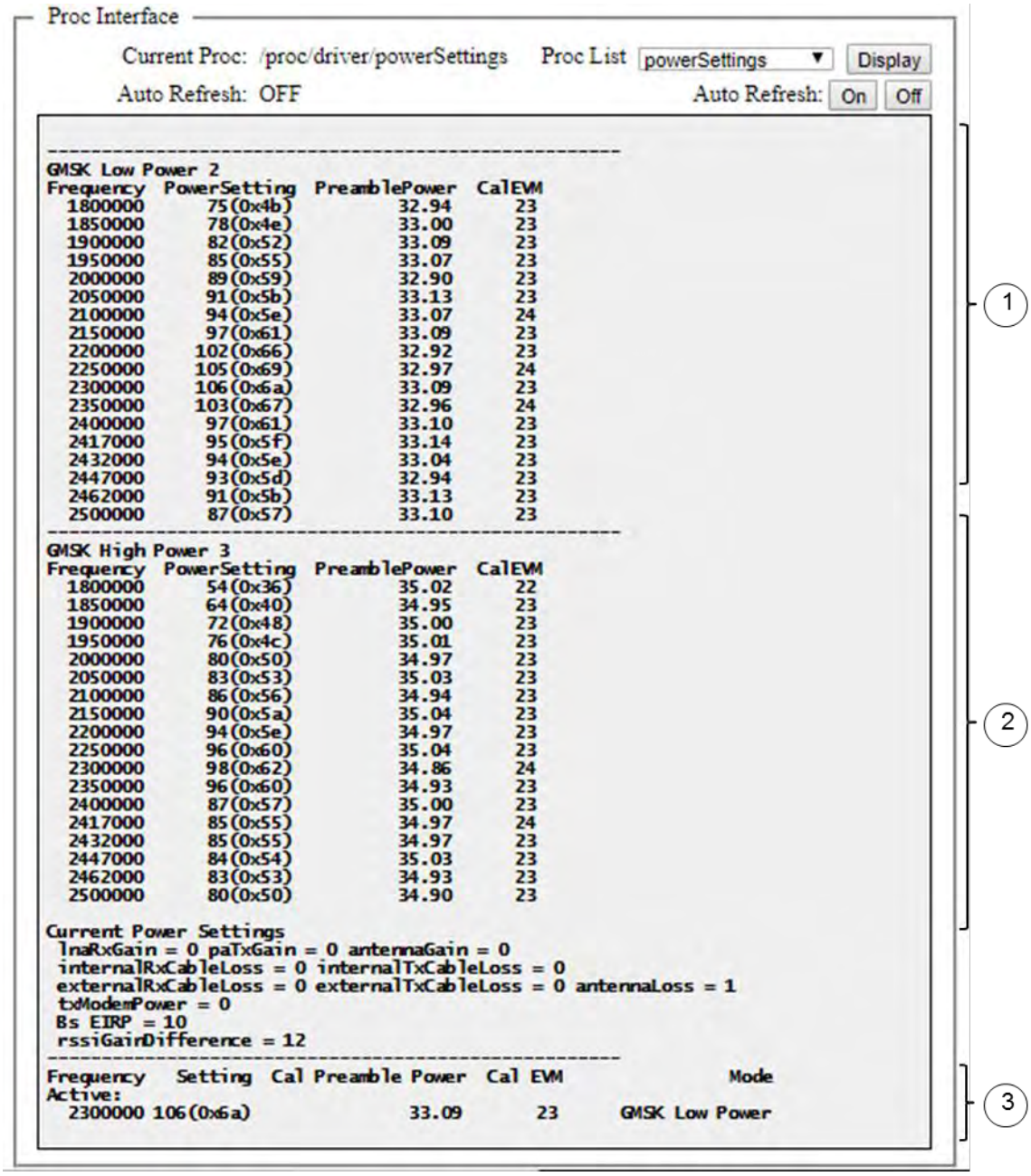


Figure 7-3 GMSK powerSettings interface

Table 7-1. Transmit Calibration interface		
	Field/Control	**Description**
1	High Data – OFDM Low Power - GMSK	The transmit calibration results for the listed frequencies for High Data if radio is OFDM and Low Power if radio is GMSK
2	High Power – OFDM High Power - GMSK	The transmit calibration results for the listed frequencies for High Power.
3	Current Transmitter setting	The radio will select the table entry whose frequency is closest to the user configured frequency.

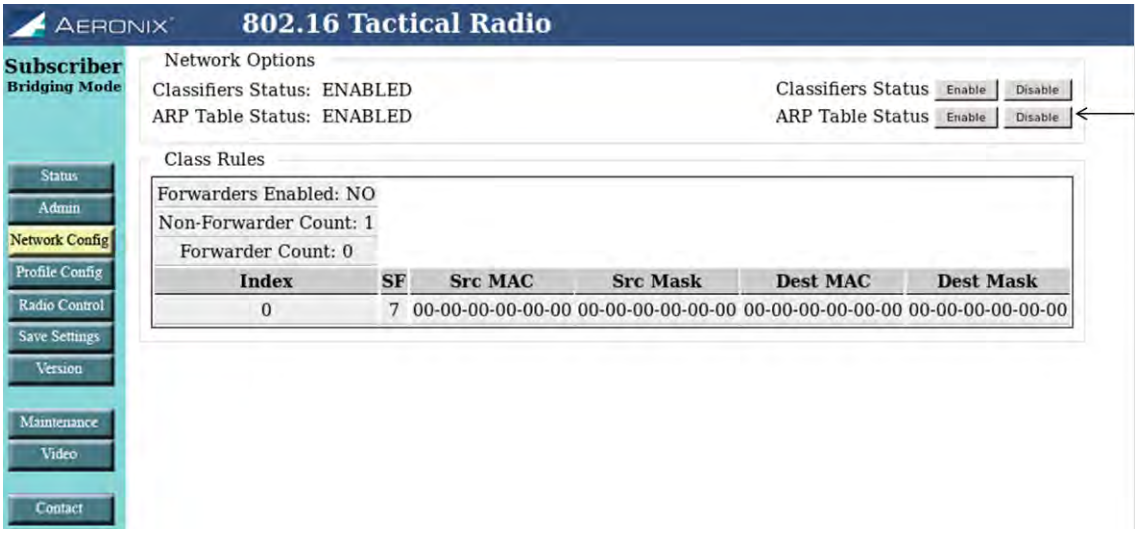
7.2. ARP Table feature

The ARP Table feature does not require the user to enter classifiers for IP traffic to flow through the Aeronix radio network. The Aeronix ARP software relies on the radio and peripherals connected to the radios, to send out IP address resolution protocol (ARP) requests when the IP traffic starts. From the ARP request and replies the Aeronix software deduces the path for each IP source and destination. If for some reason the network surrounding the radios do not send out ARPs, for instance they may have static ARPs configured, then the user will have to create Classifier Rules for their IP traffic to flow through the Aeronix network.
If the ARP Forwarding feature is enabled, then the ARP table allows Subscribers to pass data between each other. This feature is similar to the Classifier Forwarders, where all Subscriber data flows to the BS and the BS will determine if the data is destined for the BS or to be forwarded to another Subscriber.
Since the ARP feature relies on ARPs, IPv6 is not compatible with this feature, so user will have to use the Classifier Rules for IPv6 traffic.

7.2.1. Enable Aeronix ARP feature

The ARP feature has to be enabled on both the Base station and the Subscribers.

1. Browse to the Network Config page
2. Click on the ARP Table Status Enable button



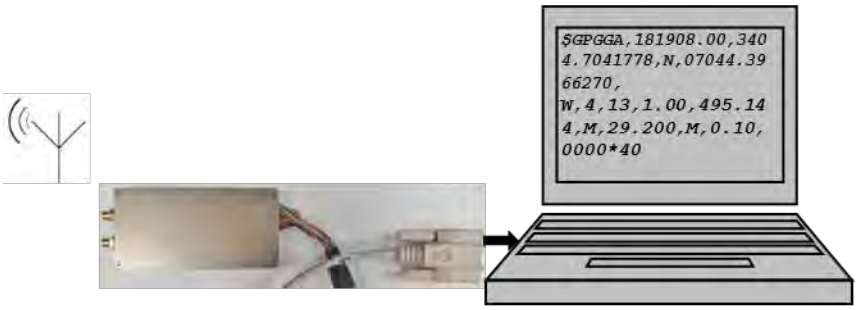
Note: If the Classifier Status and ARP Table Status are both enabled, the software will search the ARP table first for a match, if no match is found then the classifier table will be searched.

7.3. Aeronix Serial Flow Control Feature

The Aeronix serial flow control feature allows serial data to pass through the NANO's Platform RS-232 interface (See Table 2-8. J1: Main IF Pin Definitions). This section will go into further detail on how to setup the Serial B to Serial B Flow Control.

The transmitted serial data is wrapped in a UDP packet and configured with the Destination IP address for the radio the serial data is intended for and a user configured destination port ID. The example below show the BS connected to a GPS serial receiver via the radios platform serial interface. SS1 has a laptop connected to its platform serial interface with a console display. The BS will packetize all data from its RS-232 serial interface and send it over the RF destined for SS1. SS1 will push any data it receives containing destination IP address of its radio and a destination port address matching the "Receive port" of the serial flow control, to its serial device.

BS
Radio IP 192.168.100.1



SS1
Radio IP 192.168.1.1

Figure 7-4 Serial B to Serial B Exp

7.3.1.1. BS Example Radio setting

Below is the BS radio configuration for example in Figure 7-4 Serial B to Serial B Exp.

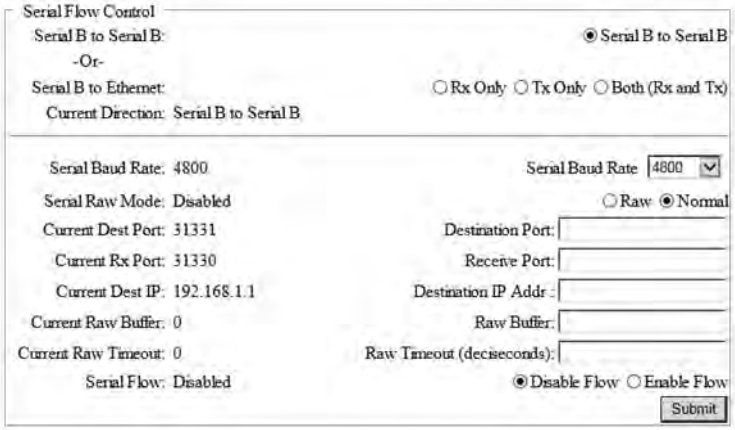


Figure 7-5 BS serial config exp

7.3.1.2. SS Example Radio setting

Below is the SS radio configuration for example in Figure 7-4 Serial B to Serial B Exp.

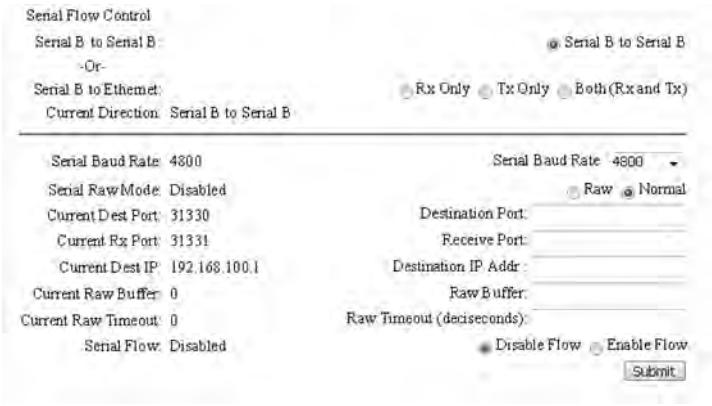


Figure 7-6 SS serial config exp

8. Radio Performance

8.1. GMSK NANO RF Interference Performance

Insert GMSK Interference info TBD

8.2. NANO GMSK Receiver Sensitivity

The sensitivity of the NANO receiver conforms to Section 8.3.11.1 of the IEEE 802.16D5-2004 specification. The GMSK sensitivity evaluations were performed at 2395 MHz. The sensitivity limit is a bit error rate of 1E-06.

Table 8-1. GMSK Receiver SNR Assumptions

Modulation	Coding Rate	Receiver Sensitivity (4dB NF)	Sensitivity Bit Error Rate
1PPS	1/2	-94	1.6E-07
2PPS	1/2	-93	0
	3/4	-92	6.0E-08
4PPS	1/2	-92	0
	3/4	-90	7.0E-07
8PPS	1/2	-89	0
	3/4	-88	0

8.3. OFDM NANO RF Interference Performance

TBD

8.4. NANO OFDM Receiver Sensitivity

The sensitivity of the NANO receiver conforms to Section 8.3.11.1 of the IEEE 802.16D5-2004 specification. The OFDM sensitivity evaluations were performed at 2395 MHz The sensitivity limit is a bit error rate of 1E-06.

Table 8-2 OFDM. 14MHz Receiver SNR Assumptions				
Modulation	**Coding Rate**	**Receiver SNR (dB)**	**Receiver Sensitivity (4dB NF)**	**Sensitivity Bit Error Rate**
BPSK	1/2	6	-88	1.1E-06
QPSK	1/2	8	-86	1.0E-06
	3/4	10	-84	8.0E-07
16 QAM	1/2	14	-81	9.0E-07
	3/4	16	-78	5.0E-07
64 QAM	2/3	21	-74	6.0E-07
	3/4	22	-72	7.0E-07

Table 8-3 OFDM. 7MHz Receiver SNR Assumptions				
Modulation	**Coding Rate**	**Receiver SNR (dB)**	**Receiver Sensitivity (4dB NF)**	**Sensitivity Bit Error Rate**
BPSK	1/2	6	-91	8.0E-07
QPSK	1/2	8	-89	9.0E-07
	3/4	10	-87	1.6E-06
16 QAM	1/2	14	-84	1.3E-06
	3/4	16	-81	1.2E-06
64 QAM	2/3	21	-77	8.0E-07
	3/4	22	-75	1.1E-06

Table 8-4 OFDM. 3.5 MHz Receiver SNR Assumptions				
Modulation	**Coding Rate**	**Receiver SNR (dB)**	**Receiver Sensitivity (4dB NF)**	**Sensitivity Bit Error Rate**
BPSK	1/2	6	-93	1.1E-06
QPSK	1/2	8	-91	7.0E-07
	3/4	10	-89	7.0E-07
16 QAM	1/2	14	-86	5.0E-07
	3/4	16	-84	1.2E-06
64 QAM	2/3	20	-80	1.1E-06
	3/4	22	-77	1.7E-06

8.5. Data Rates

8.5.1. OFDM Max Data Rates

The NANO PHY modulates and de-modulates the data packets as specified in Table 8-5.

Table 8-5. 3.5 MHz BW Max Data Rates								
Modulation	**Cyclic prefix**	**Uncoded block size (bytes/ symbol)**	**Coded block size (bytes)**	**Overall coding rate**	**RS code**	**CC code rate**	**Data rate (Mbs)**	**SNR Eb/No**
BPSK	1/4	12	24	1/2	None	1/2	1.2	6.4
QPSK	1/4	24	48	1/2	(32,24,4)	2/3	2.4	9.4
QPSK	1/4	36	48	3/4	(40,36,3)	5/6	3.6	11.2
16-QAM	1/4	48	96	1/2	(64,48,8)	2/3	4.8	16.4
16-QAM	1/4	72	96	3/4	(80,72,4)	5/6	7.2	18.2
64-QAM	1/4	96	144	2/3	(108,96,6)	3/4	9.5	22.7
64-QAM	1/8	108	144	3/4	(120,108,6)	5/6	11.9	24.4

Table 8-6. 7 MHz BW Max Data Rates								
Modulation	**Cyclic prefix**	**Uncoded block size (bytes/ symbol)**	**Coded block size (bytes)**	**Overall coding rate**	**RS code**	**CC code rate**	**Data rate (Mbs)**	**SNR Eb/No**
BPSK	1/4	12	24	1/2	None	1/2	2.4	6.4
QPSK	1/4	24	48	1/2	(32,24,4)	2/3	4.8	9.4
QPSK	1/4	36	48	3/4	(40,36,3)	5/6	7.2	11.2
16-QAM	1/4	48	96	1/2	(64,48,8)	2/3	9.6	16.4
16-QAM	1/4	72	96	3/4	(80,72,4)	5/6	14.4	18.2
64-QAM	1/4	96	144	2/3	(108,96,6)	3/4	19.0	22.7
64-QAM	1/8	108	144	3/4	(120,108,6)	5/6	23.9	24.4

Table 8-7. 14 MHz BW Max Data Rates								
Modulation	**Cyclic prefix**	**Uncoded block size (bytes/ symbol)**	**Coded block size (bytes)**	**Overall coding rate**	**RS code**	**CC code rate**	**Data rate (Mbs)**	**SNR Eb/No**
BPSK	1/4	12	24	1/2	None	1/2	4.8	6.4
QPSK	1/4	24	48	1/2	(32,24,4)	2/3	9.6	9.4
QPSK	1/4	36	48	3/4	(40,36,3)	5/6	14.4	11.2
16-QAM	1/4	48	96	1/2	(64,48,8)	2/3	19.2	16.4
16-QAM	1/4	72	96	3/4	(80,72,4)	5/6	28.8	18.2
64-QAM	1/4	96	144	2/3	(108,96,6)	3/4	38.0	22.7
64-QAM	1/8	108	144	3/4	(120,108,6)	5/6	47.8	24.4

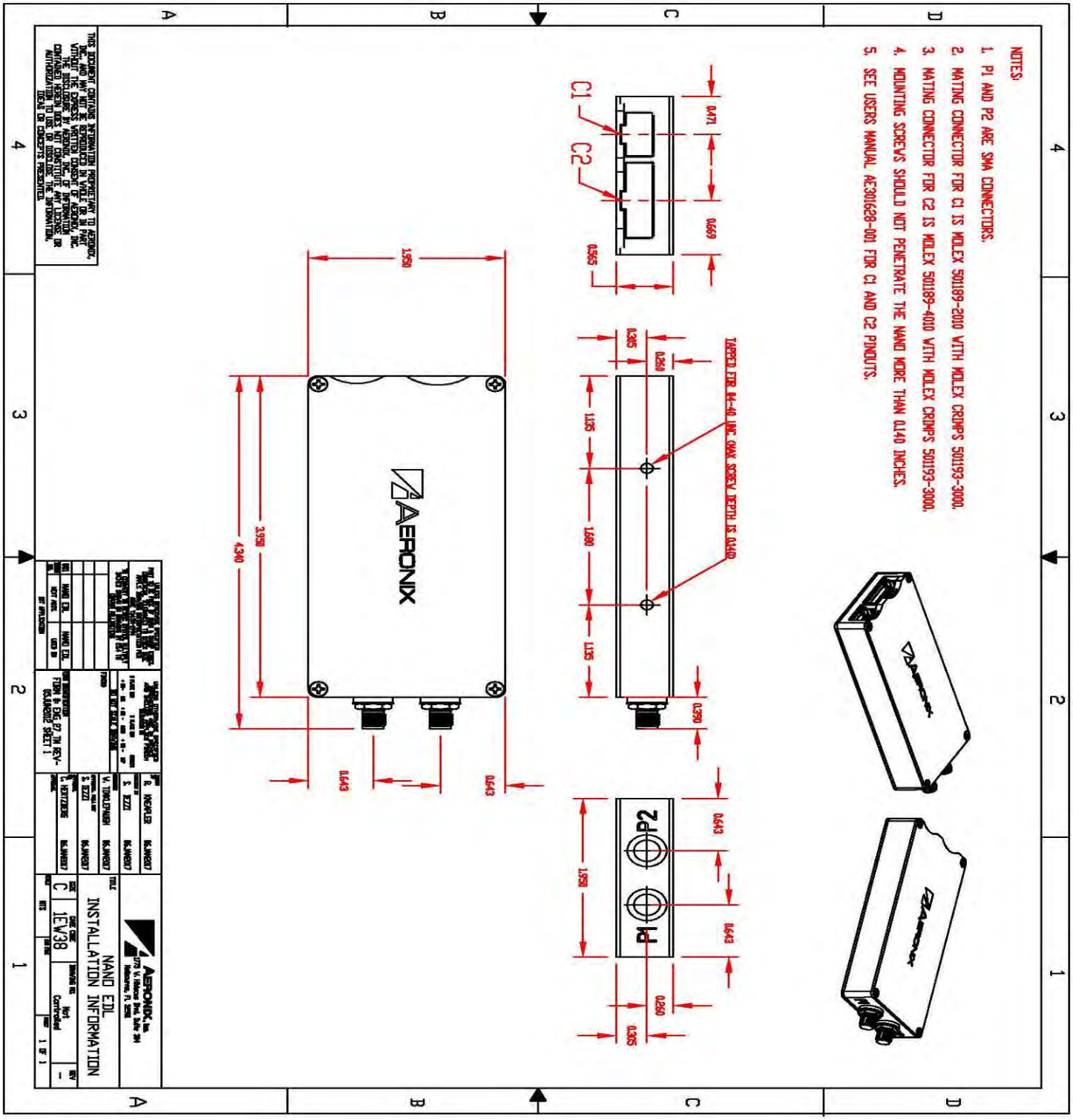
8.5.2. GMSK Data Rates

The User Data rate below is an estimation of data rate available to the user in point to point mode with minimum distance between the radios.

Table 8-8. 14 MHz BW GMSK Max Data Rates								
Modulation	**Cyclic prefix**	**Uncoded block size (bytes/ symbol)**	**Coded block size (bytes)**	**Overall coding rate**	**RS code**	**CC code rate**	**Data rate (Kbs)**	**User Data Rate (Kbs)**
1PPS	1/2	**12**	24	1/2	None		796.8	62.40
2PPS	1/2	**24**	48	1/2	(32,24,4)		1593.6	124.80
	3/4	**36**	48	3/4	(40,36,3)		2390.4	187.20
4PPS	1/2	**48**	96	1/2			3187.2	249.60
	3/4	**72**	96	3/4			4780.8	**374.40**
8PPS	1/2	**96**	192	1/2			6374.4	**499.20**
	3/4	**144**	192	3/4			9561.6	**748.80**

9. Mechanicals

9.1. Part Number AE102429



Appendix A: SNMP MIB

```
AERONIX-80216-MIB DEFINITIONS ::= BEGIN

  IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    NOTIFICATION-TYPE,
    Unsigned32, Integer32, Counter32,
    Counter64, transmission
      FROM SNMPv2-SMI
    SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
    TEXTUAL-CONVENTION,
    MacAddress, RowStatus, TruthValue,
    TimeStamp, DateAndTime, DisplayString
      FROM SNMPv2-TC
    InetAddressType, InetAddress, InetAddressIPv4
      FROM INET-ADDRESS-MIB
    OBJECT-GROUP,
    MODULE-COMPLIANCE,
    NOTIFICATION-GROUP
      FROM SNMPv2-CONF
    ifIndex
      FROM IF-MIB;

wmanIfMib MODULE-IDENTITY
  LAST-UPDATED "200508020000Z" -- August 02, 2005
  ORGANIZATION "IEEE 802.16"
  CONTACT-INFO
    "WG E-mail:   stds-802-16@ieee.org
    WG Chair:   Roger B. Marks
    Postal:      (U.S.) National Institute
    of Standards and Technology
    E-mail:      r.b.marks@ieee.org

    TGf Chair: Phillip Barber
    Postal:    Huawei Technologies Co., Ltd
    E-mail:    pbarber@futurewei.com      provi

    Editor:   Joey Chou
    Postal:      Intel Corporation
    5000 W. Chandler Blvd,
    Chandler, AZ 85227, USA
    E-mail:      joey.chou@intel.com"
  DESCRIPTION
    "This material is from IEEE Std 802.16f
    Copyright (c) 2005 IEEE.
    This MIB Module defines managed objects for
    IEEE 802.16-2004 based Subscriber Station
    and Base Station."
  REVISION
    "200508020000Z"
  DESCRIPTION
    "The first approved version of WMAN-IF-MIB module."
```

```
  ::= { aeronix 1 }

--
  private   OBJECT IDENTIFIER ::= { iso org(3) dod(6) internet(1) 4 }
  enterprises   OBJECT IDENTIFIER ::= { private 1 }
  bae   OBJECT IDENTIFIER ::= { enterprises 17701 }
  aeronix   OBJECT IDENTIFIER ::= { bae 10 }
--

wmanIfBsObjects   OBJECT IDENTIFIER ::= { wmanIfMib 1 }
wmanIfCommonObjects   OBJECT IDENTIFIER ::= { wmanIfMib 2 }

-- Textual Conventions
WmanIfSfSchedulingType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "The scheduling service provided by a SC for an
    upstream service flow. If the parameter is omitted
    from an upstream QOS Parameter Set, this object takes
    the value of bestEffort (2). This parameter must be
    reported as undefined (1) for downstream QOS Parameter
    Sets."
  SYNTAX INTEGER
  {
    undefined(1),
    bestEffort(2),
    nonRealTimePollingService(3),
    realTimePollingService(4),
    reserved(5),
    unsolicitedGrantService(6)
  }

WmanIfSfState ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "WmanIfSfState defines the state of a service flow."
  SYNTAX INTEGER
  {
    authorized(1),
    admitted(2),
    active(3)
  }

WmanIfCidType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "Type of CID."
  SYNTAX INTEGER (0 .. 65535)

WmanIfOfdmFecCodeType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "FEC code type and modulation type"
  REFERENCE
```

```
    "Table 356 and Table 362
    in IEEE Std 802.16-2004"
  SYNTAX INTEGER
  {
    fecBPSK-12(0),
    fecQPSK-12(1),
    fecQPSK-34(2),
    fecQAM-1612(3),
    fecQAM-1634(4),
    fecQAM-6423(5),
    fecQAM-6434(6),
    fec8PSK-12(7),
    fec16PSK-12(8),
    fec16PSK-34(9),
    invalid(255)
  }

WmanIfOnOffToggleType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "On Off Toggle Type"
  SYNTAX INTEGER
  {
    off (0),
    on (1)
  }

WmanIfDownStreamUpStreamType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "Downstream Upstream Type"
  SYNTAX INTEGER
  {
    downstream (1),
    upstream (2)
  }

WmanIfMisoModeType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "Miso Type"
  SYNTAX INTEGER
  {
    miso (0),
    siso (1)
  }


WmanIfDiversityType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "Diversity Type"
  SYNTAX INTEGER
  {
```

```
    unknown (0),
    auto (1),
    a (2),
    b (3)
  }

WmanIfOperatingBwType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "Operating BW Type"
  SYNTAX INTEGER
  {
    bw2400MHz (24),
    bw4600or5800MHz (58)
  }

WmanIfBandwidthType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    ""
  SYNTAX INTEGER
  {
    tenMhz (10000000),
    twentyMHz(17500000)
  }

WmanIfChannelType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    ""
  SYNTAX INTEGER
  {
    rfc01 (1),
    rfc02 (2),
    rfc03 (3),
    rfc04 (4),
    rfc05 (5),
    rfc06 (6),
    rfc07 (7),
    rfc08 (8),
    rfc09 (9),
    rfc10 (10),
    rfc11 (11),
    rfc21 (21),
    rfc22 (22),
    rfc23 (23),
    rfc24 (24),
    rfc25 (25),
    rfc26 (26),
    rfc27 (27),
    rfc28 (28),
    rfc29 (29),
    rfc30 (30),
    rfc31 (31),
```

```
    rfc32 (32),
    rfc33 (33),
    rfc34 (34),
    rfc35 (35),
    rfc36 (36),
    rfc37 (37),
    rfc38 (38),
    rfc39 (39),
    rfc40 (40),
    rfc41 (41),
    rfc42 (42),
    rfc43 (43),
    rfc44 (44),
    rfc45 (45),
    rfc46 (46),
    rfc47 (47),
    rfc48 (48),
    rfc49 (49),
    rfc50 (50),
    rfc51 (51),
    rfc52 (52),
    rfc53 (53),
    rfc54 (54),
    rfc55 (55),
    rfc56 (56),
    rfc57 (57),
    rfc58 (58),
    rfc59 (59)
  }

WmanIfEnableType ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "Enable Type"
  SYNTAX INTEGER
  {
    disable (0),
    enable (1)
  }


--
-- BS object group containing tables and objects to be implemented in
-- the Base station
--
--

wmanIfBsProvisionedSfTable OBJECT-TYPE
  SYNTAX SEQUENCE OF WmanIfBsProvisionedSfEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains service flow profiles provisioned by
    NMS. The service flow should be created with SS(s)
```

```
      following instruction given by wmanIfBsSfState object.
      1. The QoS parameters of the service flow are provisioned
      in wmanIfBsServiceClassTable and referenced by
      wmanIfBsServiceClassIndex.
      2. The classifier rules of the service flow are provisioned
      in wmanIfBsClassifierRuleTable, where they refer to SF
      via wmanIfBsSfId.

      The MAC addresses of SSs the service flow is created with
      are provisioned in wmanIfBsSsProvisionedForSfTable, where
      they refer to SF via wmanIfBsSfId."
    REFERENCE
      "Subclause 6.3.13 and 6.3.14 in IEEE Std 802.16-2004"
    ::= { wmanIfBsObjects 1 }

wmanIfBsProvisionedSfEntry OBJECT-TYPE
    SYNTAX WmanIfBsProvisionedSfEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
      "This table provides one row for each service flow
      provisioned by NMS. The table is indexed by ifIndex and
      wmanIfBsSfId. ifIndex is associated with the BS sector."
    INDEX { snmpRowIndex }
    ::= { wmanIfBsProvisionedSfTable 1 }

WmanIfBsProvisionedSfEntry ::= SEQUENCE
{
    snmpRowIndex                    INTEGER,
    wmanIfBsSfId                    Unsigned32,
    wmanIfBsSsProvmacAddress        DisplayString,
    wmanIfBsSfDirection             WmanIfDownStreamUpStreamType,
    wmanIfBsServiceClassName        DisplayString,
    wmanIfBsSfState                 WmanIfSfState,
    wmanIfBsSfUlMulticast           WmanIfOnOffToggleType,
    wmanIfBsProvisionedSfRowStatus  RowStatus
}
    wmanIfBsSfId OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "A 32 bit quantity that uniquely identifies a service flow
        to both the subscriber station and base station (BS)."
      ::= { wmanIfBsProvisionedSfEntry 1 }

    wmanIfBsSsProvmacAddress OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "The MAC address of the SS, the service flow is created
        with."
      ::= { wmanIfBsProvisionedSfEntry 3 }
```

```
wmanIfBsSfDirection OBJECT-TYPE
  SYNTAX WmanIfDownStreamUpStreamType
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "An attribute indicating the service flow is downstream or
    upstream."
  ::= { wmanIfBsProvisionedSfEntry 4 }

wmanIfBsServiceClassName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Refers to the Service Class Name"
  REFERENCE
    "Subclause 11.13.3 in IEEE Std 802.16-2004"
  ::= { wmanIfBsProvisionedSfEntry 5 }

wmanIfBsSfState OBJECT-TYPE
  SYNTAX WmanIfSfState
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "wmanIfBsSfState determines the requested state of a service
    flow.
    - authorized state: A service flow is provisioned but
    not resource is reserved yet
    - admitted state: service flow has resources reserved.
    - active state: has resources committed by the BS (e.g., is
    actively sending maps containing unsolicited grants for a
    UGS-based service flow),"
  REFERENCE
    "Subclause 6.3.14.6, in IEEE Std 802.16-2004"
  ::= { wmanIfBsProvisionedSfEntry 6 }

wmanIfBsSfUlMulticast OBJECT-TYPE
  SYNTAX WmanIfOnOffToggleType
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { wmanIfBsProvisionedSfEntry 7 }

wmanIfBsProvisionedSfRowStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "This object is used to create a new row or modify or
    delete an existing row in this table.

    If the implementator of this MIB has choosen not
```

```
        to implement 'dynamic assignment' of profiles, this
        object is not useful and should return noSuchName
        upon SNMP request."
      ::= { wmanIfBsProvisionedSfEntry 8 }

    snmpRowIndex OBJECT-TYPE
      SYNTAX INTEGER (1..1000)
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "SNMP Row Index"
      ::= { wmanIfBsProvisionedSfEntry 999 }

--
--
--
--

wmanIfBsSsNameTable OBJECT-TYPE
  SYNTAX SEQUENCE OF WmanIfBsSsNameEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  REFERENCE
    "Subclause 6.3.13 and 6.3.14 in IEEE Std 802.16-2004"
  ::= { wmanIfBsObjects 4 }

wmanIfBsSsNameEntry OBJECT-TYPE
  SYNTAX WmanIfBsSsNameEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  INDEX { snmpRowIndex }
  ::= { wmanIfBsSsNameTable 1 }

WmanIfBsSsNameEntry ::= SEQUENCE
{
  wmanIfBsSsNameNum                    Unsigned32,
  wmanIfBsSsNameSubscriberName            DisplayString,
  wmanIfBsSsNameMACAddress              DisplayString
}

  wmanIfBsSsNameNum OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { wmanIfBsSsNameEntry 1 }

  wmanIfBsSsNameSubscriberName OBJECT-TYPE
    SYNTAX DisplayString
```

```
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        ""
      ::= { wmanIfBsSsNameEntry 2 }

    wmanIfBsSsNameMACAddress OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        ""
      ::= { wmanIfBsSsNameEntry 3 }

--
--
--
--

wmanIfBsServiceClassTable OBJECT-TYPE
  SYNTAX SEQUENCE OF WmanIfBsServiceClassEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table is provisioned and is indexed by
    wmanIfBsQoSProfileIndex. Each entry of the table contains
    corresponding service flow characteristic attributes
    (e.g. QoS parameter set). The value of
    wmanIfBsQoSProfileIndex is obtained from
    wmanIfBsServiceClassIndex in wmanIfBsProvisionedSfTable"
  REFERENCE
    "Subclause 6.3.14.4 in IEEE Std 802.16-2004"
  ::= { wmanIfBsObjects 2 }

wmanIfBsServiceClassEntry OBJECT-TYPE
  SYNTAX WmanIfBsServiceClassEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table provides one row for each service class"
  INDEX { snmpRowIndex }
  ::= { wmanIfBsServiceClassTable 1 }

WmanIfBsServiceClassEntry ::= SEQUENCE
{
  wmanIfBsQoSProfileIndex              INTEGER,
  wmanIfBsQosServiceClassName          DisplayString,
  bitArray                             DisplayString,
  wmanIfBsQoSTrafficPriority           INTEGER,
  wmanIfBsQoSMaxSustainedRate          Unsigned32,
  wmanIfBsQoSMinReservedRate           Unsigned32,
  wmanIfBsQosScSchedulingType          WmanIfSfSchedulingType,
  wmanIfBsQoSServiceClassRowStatus     RowStatus
}
```

```
wmanIfBsQoSProfileIndex OBJECT-TYPE
  SYNTAX INTEGER
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "The index value which uniquely identifies an entry
    in the wmanIfBsServiceClassTable"
  ::= { wmanIfBsServiceClassEntry 1 }

wmanIfBsQosServiceClassName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Refers to the Service Class Name"
  REFERENCE
    "Subclause 11.13.3 in IEEE Std 802.16-2004"
  ::= { wmanIfBsServiceClassEntry 2 }

bitArray OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Refers to the Service Class Name"
  REFERENCE
    "Subclause 11.13.3 in IEEE Std 802.16-2004"
  ::= { wmanIfBsServiceClassEntry 3 }

wmanIfBsQoSTrafficPriority OBJECT-TYPE
  SYNTAX INTEGER
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "The value of this parameter specifies the priority
    assigned to a service flow. For uplink service flows,
    the BS should use this parameter when determining
    precedence in request service and grant generation,
    and the SS shall preferentially select contention
    Request opportunities for Priority Request CIDs
    based on this priority. Higher numbers indicate higher
    priority"
  REFERENCE
    "Subclause 11.13.5 in IEEE Std 802.16-2004"
  ::= { wmanIfBsServiceClassEntry 4 }

wmanIfBsQoSMaxSustainedRate OBJECT-TYPE
  SYNTAX Unsigned32
  UNITS "b/s"
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This parameter defines the peak information rate
```

```
        of the service. The rate is expressed in bits per
        second and pertains to the SDUs at the input to
        the system."
      REFERENCE
        "Subclause 11.13.6 in IEEE Std 802.16-2004"
      ::= { wmanIfBsServiceClassEntry 5 }

    wmanIfBsQoSMinReservedRate OBJECT-TYPE
      SYNTAX Unsigned32
      UNITS "b/s"
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "This parameter specifies the minimum rate reserved
        for this service flow."
      REFERENCE
        "Subclause 11.13.8 in IEEE Std 802.16-2004"
      ::= { wmanIfBsServiceClassEntry 6 }

    wmanIfBsQosScSchedulingType OBJECT-TYPE
      SYNTAX WmanIfSfSchedulingType
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "Specifies the upstream scheduling service used for
        upstream service flow. If the referenced parameter
        is not present in the corresponding 802.16 QOS
        Parameter Set of an upstream service flow, the
        default value of this object is bestEffort(2)."
      REFERENCE
        "Subclause 11.13.11 in IEEE Std 802.16-2004"
      DEFVAL { bestEffort }
      ::= { wmanIfBsServiceClassEntry 7 }

    wmanIfBsQoSServiceClassRowStatus OBJECT-TYPE
      SYNTAX RowStatus
      MAX-ACCESS read-create
      STATUS current
      DESCRIPTION
        "This object is used to create a new row or modify or
        delete an existing row in this table.

        If the implementator of this MIB has choosen not
        to implement 'dynamic assignment' of profiles, this
        object is not useful and should return noSuchName
        upon SNMP request."
      ::= { wmanIfBsServiceClassEntry 8 }


--
--
--
--
```

```
wmanIfBsRegisteredSsTable OBJECT-TYPE
  SYNTAX SEQUENCE OF WmanIfBsRegisteredSsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains the basic capability information
    of SSs that have been negotiated and agreed between
    BS and SS via REG-REQ and REG-RSP messages. An entry
    in this table indicates the SS has entered and registered
    into the BS."
  REFERENCE
    "Subclause 6.3.2.3.7 in IEEE Std 802.16-2004"
  ::= { wmanIfBsObjects 3 }

wmanIfBsRegisteredSsEntry OBJECT-TYPE
  SYNTAX WmanIfBsRegisteredSsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table provides one row for each SS that has been
    registered in the BS, and is indexed by
    wmanIfBsSsMacAddress. The primary index is the ifIndex
    with an ifType of propBWAp2Mp, indicating the BS sector
    with which the SS is associated. wmanIfBsSsMacAddress
    identifies the SS being registered."
  INDEX { snmpRowIndex }
  ::= { wmanIfBsRegisteredSsTable 1 }

WmanIfBsRegisteredSsEntry ::= SEQUENCE
{
  wmanIfBsSsIdIndex                Unsigned32,
  wmanIfBsSsMacAddress                 DisplayString,
  wmanIfBsSsBasicCid               WmanIfCidType,
  wmanIfBsSsPrimaryCid              WmanIfCidType,
  wmanIfBsSsMacCrcSupport              WmanIfOnOffToggleType,
  snmpRowIndex                     Unsigned32
}

  wmanIfBsSsIdIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    REFERENCE
      "Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
    ::= { wmanIfBsRegisteredSsEntry 1 }

  wmanIfBsSsMacAddress OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "The MAC address of SS is received from the RNG-REQ
```

```
        message. When SS registers, this MAC address is entered
        into the table, and used as the identifier to the SS."
      REFERENCE
        "Subclause 6.3.2.3.5 in IEEE Std 802.16-2004"
      ::= { wmanIfBsRegisteredSsEntry 2 }

    wmanIfBsSsBasicCid OBJECT-TYPE
      SYNTAX WmanIfCidType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The value of this object indicates the SS's basic CID
        that was sent in the RNG-RSP message."
      REFERENCE
        "Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
      ::= { wmanIfBsRegisteredSsEntry 3 }

    wmanIfBsSsPrimaryCid OBJECT-TYPE
      SYNTAX WmanIfCidType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      REFERENCE
        "Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
      ::= { wmanIfBsRegisteredSsEntry 4 }

    wmanIfBsSsMacCrcSupport OBJECT-TYPE
      SYNTAX WmanIfOnOffToggleType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      REFERENCE
        "Subclause 11.13.18.3 in IEEE Std 802.16-2004"
      ::= { wmanIfBsRegisteredSsEntry 5 }

    snmpRowIndex OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      REFERENCE
        ""
      ::= { wmanIfBsRegisteredSsEntry 6 }
--
-- Common object group containing tables and objects to be implemented in
-- both the Base station and the Subscriber station
--
--

aeronixSystemStatusTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixSystemStatusEntry
```

```
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains system status values "
  ::= { wmanIfCommonObjects 1 }

aeronixSystemStatusEntry OBJECT-TYPE
  SYNTAX AeronixSystemStatusEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This Table is a unique aeronix strucutre, Used to view the
    status of the radio. It was ported from the wmanIfMib.h header file"
  INDEX { snmpRowIndex }
  ::= { aeronixSystemStatusTable 1 }

AeronixSystemStatusEntry ::= SEQUENCE
{
  statusBaseLight                Unsigned32,
  statusSyncState                Unsigned32,
  statusSyncLight                INTEGER,
  statusTx                       Unsigned32,
  statusRx                       Unsigned32,
  isBsorSs                       INTEGER
}

  statusBaseLight OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemStatusEntry 1 }

  statusSyncState OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemStatusEntry 2 }

  statusSyncLight OBJECT-TYPE
    SYNTAX INTEGER
    {
      nosync(0),
      sync(1)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemStatusEntry 3 }
```

```
  statusTx OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemStatusEntry 4 }

  statusRx OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemStatusEntry 5 }

  isBsorSs OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemStatusEntry 6 }
--
--
--
--

aeronixSystemControlTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixSystemControlEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains packet classifier rules associated
    with service flows."
  ::= { wmanIfCommonObjects 2 }

aeronixSystemControlEntry OBJECT-TYPE
  SYNTAX AeronixSystemControlEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This Table is a unique aeronix strucutre, Used to conotrol the radio.
    It was ported from the wmanIfMib.h header file"
  INDEX { snmpRowIndex }
  ::= { aeronixSystemControlTable 1 }

AeronixSystemControlEntry ::= SEQUENCE
{
  txrxEnable                    WmanIfEnableType,
  transecEnable                   WmanIfEnableType,
  freqBandwidth                   Unsigned32,
  freqChannel                   INTEGER,
  downlinkPercentage                INTEGER,
```

```
  uplinkPercentage                INTEGER,
  frameTimeMSec                   INTEGER,
  ssScanChannel                   WmanIfEnableType,
  acmUpHysteresis                 INTEGER,
  acmDownJump                     INTEGER,
  effectiveKey                    DisplayString,
  misoMode                        WmanIfMisoModeType,
  rxDataSwap                      WmanIfEnableType,
  radioTypeConfig                 INTEGER,
  enaSerialTransfer               WmanIfEnableType,
  serialTxPort                    Unsigned32,
  serialRxPort                    Unsigned32,
  serialTxIP                      DisplayString,
  usingClassifiers                WmanIfEnableType,
  usingARPTable                   WmanIfEnableType,
  customFreq                      INTEGER
}

  txrxEnable OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 1 }

  transecEnable OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 2 }

  freqBandwidth OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 3 }

  freqChannel OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 4 }

  downlinkPercentage OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 5 }

  uplinkPercentage OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 6 }

  frameTimeMSec OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 7 }

  ssScanChannel OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 8 }

  acmUpHysteresis OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 9 }

  acmDownJump OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 10 }

  effectiveKey OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 11 }

  misoMode OBJECT-TYPE
    SYNTAX WmanIfMisoModeType
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 12 }

  rxDataSwap OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 13 }

  radioTypeConfig OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 14 }

  enaSerialTransfer OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 15 }

  serialTxPort OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 16 }

  serialRxPort OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 17 }

  serialTxIP OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 18 }
```

```
  usingClassifiers OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 19 }

  usingARPTable OBJECT-TYPE
    SYNTAX WmanIfEnableType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 20 }

  customFreq OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSystemControlEntry 21 }
--
--
--
--

aeronixSnmpControlTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixSnmpControlEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains packet classifier rules associated
    with service flows."
  ::= { wmanIfCommonObjects 4 }

aeronixSnmpControlEntry OBJECT-TYPE
  SYNTAX AeronixSnmpControlEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table controls the snmp server and aeronix interface"
  INDEX { snmpRowIndex }
  ::= { aeronixSnmpControlTable 1 }

AeronixSnmpControlEntry ::= SEQUENCE
{
  repopulate                    Unsigned32,
  savedatabase                    Unsigned32,
  usedefaultdb                   Unsigned32,
  rebootSys                     Unsigned32,
  debugmode                       Unsigned32,
  classifierMacAddrMode                Unsigned32,
```

```
  basestation                 Unsigned32,
  dbStationId                 Unsigned32,
  ssId                        Unsigned32,
  webgui                      Unsigned32,
  udhcpc                      Unsigned32,
  bridging                    Unsigned32,
  sysIP4Address               DisplayString,
  sysIP4Netmask               DisplayString,
  sysIP4Gateway               DisplayString,
  sysIP4VLan                  Unsigned32,
  sysIP6Address               DisplayString,
  sysIP6Netmask               DisplayString,
  sysIP6Gateway               DisplayString,
  sysIP6DestNetwork           DisplayString,
  sysIP6DestNetmask           DisplayString,
  makeStartScript             Unsigned32
}

  repopulate OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 1 }

  savedatabase OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 2 }

  usedefaultdb OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 3 }

  rebootSys OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 4 }

  debugmode OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 5 }

  classifierMacAddrMode OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 6 }

  basestation OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 7 }

  dbStationId OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to BS Id that can only be set while running as a BS.
      Reboot needed after setting this value to cause it to take effect."
    ::= { aeronixSnmpControlEntry 8 }

  ssId OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 9 }

  webgui OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 10 }

  udhcpc OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 11 }

  bridging OBJECT-TYPE
```

```
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 12 }

  sysIP4Address OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv4 Address"
    ::= { aeronixSnmpControlEntry 13 }

  sysIP4Netmask OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv4 netmask"
    ::= { aeronixSnmpControlEntry 14 }

  sysIP4Gateway OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv4 gateway"
    ::= { aeronixSnmpControlEntry 15 }

  sysIP4VLan OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv4 gateway"
    ::= { aeronixSnmpControlEntry 16 }

  sysIP6Address OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv6 Address"
    ::= { aeronixSnmpControlEntry 17 }

  sysIP6Netmask OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv6 netmask"
    ::= { aeronixSnmpControlEntry 18 }
```

```
  sysIP6Gateway OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv6 gateway"
    ::= { aeronixSnmpControlEntry 19 }

  sysIP6DestNetwork OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv6 Dest Network"
    ::= { aeronixSnmpControlEntry 20 }

  sysIP6DestNetmask OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Refers to the Radios Eth IPv6 Dest Netmask"
    ::= { aeronixSnmpControlEntry 21 }

  makeStartScript OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixSnmpControlEntry 22 }

--
--
--

aeronixRangingParamsTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixRangingParamsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains ranging parameters."
  ::= { wmanIfCommonObjects 5 }

aeronixRangingParamsEntry OBJECT-TYPE
  SYNTAX AeronixRangingParamsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This Table is a unique aeronix structure, used to store ranging
    parameters. It was ported from the wmanIfMib.h header file"
  INDEX { snmpRowIndex }
  ::= { aeronixRangingParamsTable 1 }
```

```
AeronixRangingParamsEntry ::= SEQUENCE
{
  lnaRxGain                    Unsigned32,
  paTxGain                     Unsigned32,
  antennaGain                  Unsigned32,
  internalRxCableLoss          Unsigned32,
  internalTxCableLoss          Unsigned32,
  externalRxCableLoss          Unsigned32,
  externalTxCableLoss          Unsigned32,
  antennaLoss                  Unsigned32,
  txPwrControl                 INTEGER,
  powerOutputMode              INTEGER,
  externalRxCableList1Length   Unsigned32,
  externalRxCableList1Value    Unsigned32,
  externalRxCableList1Comment  DisplayString,
  externalRxCableList2Length   Unsigned32,
  externalRxCableList2Value    Unsigned32,
  externalRxCableList2Comment  DisplayString,
  externalRxCableList3Length   Unsigned32,
  externalRxCableList3Value    Unsigned32,
  externalRxCableList3Comment  DisplayString,
  externalRxCableListIndex     Unsigned32,
  externalTxCableList1Length   Unsigned32,
  externalTxCableList1Value    Unsigned32,
  externalTxCableList1Comment  DisplayString,
  externalTxCableList2Length   Unsigned32,
  externalTxCableList2Value    Unsigned32,
  externalTxCableList2Comment  DisplayString,
  externalTxCableList3Length   Unsigned32,
  externalTxCableList3Value    Unsigned32,
  externalTxCableList3Comment  DisplayString,
  externalTxCableListIndex     Unsigned32,
  antennaCableList1Length      Unsigned32,
  antennaCableList1Value       Unsigned32,
  antennaCableList1Comment     DisplayString,
  antennaCableList2Length      Unsigned32,
  antennaCableList2Value       Unsigned32,
  antennaCableList2Comment     DisplayString,
  antennaCableList3Length      Unsigned32,
  antennaCableList3Value       Unsigned32,
  antennaCableList3Comment     DisplayString,
  antennaCableListIndex        Unsigned32
}

  lnaRxGain OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 1 }

  paTxGain OBJECT-TYPE
```

```
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 2 }

  antennaGain OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 3 }

  internalRxCableLoss OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 4 }

  internalTxCableLoss OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 5 }

  externalRxCableLoss OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 6 }

  externalTxCableLoss OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 7 }

  antennaLoss OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 8 }
```

```
txPwrControl OBJECT-TYPE
  SYNTAX INTEGER
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { aeronixRangingParamsEntry 9 }

powerOutputMode OBJECT-TYPE
  SYNTAX INTEGER
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { aeronixRangingParamsEntry 10 }

externalRxCableList1Length OBJECT-TYPE
  SYNTAX Unsigned32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { aeronixRangingParamsEntry 11 }

externalRxCableList1Value OBJECT-TYPE
  SYNTAX Unsigned32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { aeronixRangingParamsEntry 12 }

externalRxCableList1Comment OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { aeronixRangingParamsEntry 13 }

externalRxCableList2Length OBJECT-TYPE
  SYNTAX Unsigned32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    ""
  ::= { aeronixRangingParamsEntry 14 }

externalRxCableList2Value OBJECT-TYPE
  SYNTAX Unsigned32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
```

```
      ""
    ::= { aeronixRangingParamsEntry 15 }

  externalRxCableList2Comment OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 16 }

  externalRxCableList3Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 17 }

  externalRxCableList3Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 18 }

  externalRxCableList3Comment OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 19 }

  externalRxCableListIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 20 }

  externalTxCableList1Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 21 }

  externalTxCableList1Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 22 }

  externalTxCableList1Comment OBJECT-TYPE
    SYNTAX   DisplayString
    MAX-ACCESS  read-write
    STATUS   current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 23 }

  externalTxCableList2Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 24 }

  externalTxCableList2Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 25 }

  externalTxCableList2Comment OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 26 }

  externalTxCableList3Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 27 }

  externalTxCableList3Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 28 }

  externalTxCableList3Comment OBJECT-TYPE
```

```
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 29 }

  externalTxCableListIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 30 }

  antennaCableList1Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 31 }

  antennaCableList1Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 32 }

  antennaCableList1Comment OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 33 }

  antennaCableList2Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 34 }

  antennaCableList2Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 35 }
```

```
  antennaCableList2Comment OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 36 }

  antennaCableList3Length OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 37 }

  antennaCableList3Value OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 38 }

  antennaCableList3Comment OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 39 }

  antennaCableListIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRangingParamsEntry 40 }

--
--
--
--

aeronixMaintStatusTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixMaintStatusEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains maintanence parameters."
  ::= { wmanIfCommonObjects 6 }
```

```
aeronixMaintStatusEntry OBJECT-TYPE
  SYNTAX AeronixMaintStatusEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This Table is a unique aeronix structure, used to store maintenance
    parameters. It was ported from the wmanIfMib.h header file"
  INDEX { snmpRowIndex }
  ::= { aeronixMaintStatusTable 1 }

AeronixMaintStatusEntry ::= SEQUENCE
{
  curChannel                    INTEGER,
  curChannelCenterFreq              INTEGER,
  failedResetPhyClock           Unsigned32,
  macAddress                    DisplayString,
  phyTemp                       INTEGER,
  cpuTemp                       INTEGER
}

  curChannel OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixMaintStatusEntry 1 }

  curChannelCenterFreq OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixMaintStatusEntry 2 }

  failedResetPhyClock OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixMaintStatusEntry 3 }

  macAddress OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixMaintStatusEntry 4 }

  phyTemp OBJECT-TYPE
    SYNTAX INTEGER
```

```
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixMaintStatusEntry 5 }

    cpuTemp OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixMaintStatusEntry 6 }

--
--
--
--

wmanIfBsOfdmDcdBurstProfileTable OBJECT-TYPE
   SYNTAX SEQUENCE OF WmanIfBsOfdmDcdBurstProfileEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table provides one row for each DCD burst profile.
      This table is double indexed. The primary index is an
      ifIndex with an ifType of propBWAp2Mp. The secondary
      index is wmanIfCmnOfdmDcdBurstProfIndex."
   REFERENCE
      "Table 362, in IEEE Std 802.16-2004"
   ::= { wmanIfCommonObjects 8 }

wmanIfBsOfdmDcdBurstProfileEntry OBJECT-TYPE
   SYNTAX WmanIfBsOfdmDcdBurstProfileEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table provides one row for each DCD burst profile.
      This table is double indexed. The secondary index
      is wmanIfCmnOfdmDcdFecCodeType."
   INDEX { snmpRowIndex }
   ::= { wmanIfBsOfdmDcdBurstProfileTable 1 }

WmanIfBsOfdmDcdBurstProfileEntry ::= SEQUENCE
{
   wmanIfCmnOfdmDcdBurstProfIndex           INTEGER,
   wmanIfCmnOfdmDiucValue                   INTEGER,
   wmanIfCmnOfdmDcdFecCodeType              WmanIfOfdmFecCodeType,
   wmanIfCmnOfdmDiucCinrMandatoryExitThresh  INTEGER,
   wmanIfCmnOfdmDiucCinrMinEntryThresh       INTEGER,
   wmanIfCmnOfdmDiucFecMandatoryExitThresh   INTEGER,
   wmanIfCmnOfdmDiucFecMinEntryThresh        INTEGER,
   wmanIfCmnOfdmAvailableType               INTEGER,
   wmanIfCmnOfdmDcdBurstProfilerowStatus     RowStatus
```

```
}

  wmanIfCmnOfdmDcdBurstProfIndex OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "The Downlink Interval Usage Code indicates the downlink
      burst profile in the DCD message, and is used along with
      ifIndex to identify an entry in the
      wmanIfCmnOfdmDcdBurstProfileTable."
    REFERENCE
      "Subclause 8.3.6.3.1, in IEEE Std 802.16-2004"
    ::= { wmanIfBsOfdmDcdBurstProfileEntry 1 }

  wmanIfCmnOfdmDiucValue OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { wmanIfBsOfdmDcdBurstProfileEntry 2 }

  wmanIfCmnOfdmDcdFecCodeType OBJECT-TYPE
    SYNTAX WmanIfOfdmFecCodeType
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "Downlink FEC code type and modulation type"
    REFERENCE
      "Table 362, in IEEE Std 802.16-2004"
    ::= { wmanIfBsOfdmDcdBurstProfileEntry 3 }

  wmanIfCmnOfdmDiucCinrMandatoryExitThresh OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "DIUC mandatory exit threshold: 0 - 63.75 dB CINR at or
      below where this DIUC can no longer be used and where this
      change to a more robust DIUC is required in 0.25 dB units."
    REFERENCE
      "Table 362, in IEEE Std 802.16-2004"
    ::= { wmanIfBsOfdmDcdBurstProfileEntry 4 }

  wmanIfCmnOfdmDiucCinrMinEntryThresh OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      "DIUC minimum entry threshold: 0 - 63.75 dB The minimum CINR
      required to start using this DIUC when changing from a more
      robust DIUC is required, in 0.25 dB units."
    REFERENCE
```

```
        "Table 362, in IEEE Std 802.16-2004"
      ::= { wmanIfBsOfdmDcdBurstProfileEntry 5 }

    wmanIfCmnOfdmDiucFecMandatoryExitThresh OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "DIUC mandatory exit threshold: 0 - 63.75 dB CINR at or
        below where this DIUC can no longer be used and where this
        change to a more robust DIUC is required in 0.25 dB units."
      REFERENCE
        "Table 362, in IEEE Std 802.16-2004"
      ::= { wmanIfBsOfdmDcdBurstProfileEntry 6 }

    wmanIfCmnOfdmDiucFecMinEntryThresh OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "DIUC minimum entry threshold: 0 - 63.75 dB The minimum CINR
        required to start using this DIUC when changing from a more
        robust DIUC is required, in 0.25 dB units."
      REFERENCE
        "Table 362, in IEEE Std 802.16-2004"
      ::= { wmanIfBsOfdmDcdBurstProfileEntry 7 }

    wmanIfCmnOfdmAvailableType OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        ""
      ::= { wmanIfBsOfdmDcdBurstProfileEntry 8 }

    wmanIfCmnOfdmDcdBurstProfilerowStatus OBJECT-TYPE
      SYNTAX RowStatus
      MAX-ACCESS read-create
      STATUS current
      DESCRIPTION
        "This object is used to create a new row or modify or
        delete an existing row in this table.

        If the implementator of this MIB has choosen not
        to implement 'dynamic assignment' of profiles, this
        object is not useful and should return noSuchName
        upon SNMP request."
      ::= { wmanIfBsOfdmDcdBurstProfileEntry 9 }

--
--
--
--
```

```
wmanIfBsOfdmUcdBurstProfileTable OBJECT-TYPE
  SYNTAX SEQUENCE OF WmanIfBsOfdmUcdBurstProfileEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains UCD burst profiles for each uplink
    channel"
  REFERENCE
    "Table 356, in IEEE Std 802.16-2004"
  ::= { wmanIfCommonObjects 9 }

wmanIfBsOfdmUcdBurstProfileEntry OBJECT-TYPE
  SYNTAX WmanIfBsOfdmUcdBurstProfileEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table provides one row for each UCD burst profile.
    This table is double indexed. The primary index is an
    ifIndex with an ifType of propBWAp2Mp. The secondary index
    is wmanIfCmnOfdmUcdBurstProfIndex."
  INDEX { snmpRowIndex }
  ::= { wmanIfBsOfdmUcdBurstProfileTable 1 }

WmanIfBsOfdmUcdBurstProfileEntry ::= SEQUENCE
{
  wmanIfCmnOfdmUcdBurstProfIndex          INTEGER,
  wmanIfCmnOfdmUiucValue                  INTEGER,
  wmanIfCmnOfdmUcdFecCodeType             WmanIfOfdmFecCodeType
}

  wmanIfCmnOfdmUcdBurstProfIndex OBJECT-TYPE
    SYNTAX INTEGER (1..11)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "The Uplink Interval Usage Code indicates the uplink burst
      profile in the UCD message, and is used along with ifIndex
      to identify an entry in the
      wmanIfBsOfdmUcdBurstProfileTable."
    REFERENCE
      "Subclause 8.3.6.3.1, in IEEE Std 802.16-2004"
    ::= { wmanIfBsOfdmUcdBurstProfileEntry 1 }

  wmanIfCmnOfdmUiucValue OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { wmanIfBsOfdmUcdBurstProfileEntry 2 }

  wmanIfCmnOfdmUcdFecCodeType OBJECT-TYPE
    SYNTAX WmanIfOfdmFecCodeType
    MAX-ACCESS read-only
```

```
      STATUS current
      DESCRIPTION
        "Uplink FEC code type and modulation type"
      REFERENCE
        "Table 356, in IEEE Std 802.16-2004"
      ::= { wmanIfBsOfdmUcdBurstProfileEntry 3 }

--
--
--
--

aeronixNetworkStatusTable OBJECT-TYPE
   SYNTAX SEQUENCE OF AeronixNetworkStatusEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     ""
   ::= { wmanIfCommonObjects 11 }

aeronixNetworkStatusEntry OBJECT-TYPE
   SYNTAX AeronixNetworkStatusEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     ""
   INDEX { snmpRowIndex }
   ::= { aeronixNetworkStatusTable 1 }

AeronixNetworkStatusEntry ::= SEQUENCE
{
   rxpackets                    Unsigned32,
   rxerrors                    Unsigned32,
   rxdropped                     Unsigned32,
   rxoverruns                    Unsigned32,
   rxframes                    Unsigned32,
   txpackets                    Unsigned32,
   txerrors                    Unsigned32,
   txdropped                     INTEGER,
   txoverruns                    INTEGER,
   txcarrier                   INTEGER,
   ipaddress                     DisplayString
}

   rxpackets OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixNetworkStatusEntry 1 }

   rxerrors OBJECT-TYPE
      SYNTAX Unsigned32
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 2 }

  rxdropped OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 3 }

  rxoverruns OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 4 }

  rxframes OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 5 }

  txpackets OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 6 }

  txerrors OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 7 }

  txdropped OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 8 }
```

```
  txoverruns OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 9 }

  txcarrier OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 10 }

  ipaddress OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixNetworkStatusEntry 11 }

--
--
--
--

aeronixTxRxMaintTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixTxRxMaintEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  ::= { wmanIfCommonObjects 12 }

aeronixTxRxMaintEntry OBJECT-TYPE
  SYNTAX AeronixTxRxMaintEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  INDEX { snmpRowIndex }
  ::= { aeronixTxRxMaintTable 1 }

AeronixTxRxMaintEntry ::= SEQUENCE
{
  rxFrames                    Unsigned32,
  invalidFrames                 Unsigned32,
  rxCRCs                      Unsigned32,
  resetRxCnt                   Unsigned32,
  rxTimeouts                   Unsigned32,
  txFrames                    Unsigned32,
```

```
  txDrops                       Unsigned32
}

  rxFrames OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 1 }

  invalidFrames OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 2 }

  rxCRCs OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 3 }

  resetRxCnt OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 4 }

  rxTimeouts OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 5 }

  txFrames OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 6 }

  txDrops OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixTxRxMaintEntry 7 }

--
--
--
--

aeronixVersionsTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixVersionsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  ::= { wmanIfCommonObjects 13 }

aeronixVersionsEntry OBJECT-TYPE
  SYNTAX AeronixVersionsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  INDEX { snmpRowIndex }
  ::= { aeronixVersionsTable 1 }

AeronixVersionsEntry ::= SEQUENCE
{
  serverIp                    DisplayString,
  internalIp                  DisplayString,
  macAddr                       DisplayString,
  macCVSTag                       DisplayString,
  macCreatedOn                     DisplayString,
  aecdDriverCreatedOn                DisplayString,
  modemId                        DisplayString,
  modemVersion                     DisplayString,
  modemSVersion                     DisplayString,
  filterFpgaVersion                DisplayString,
  agcVersion                     DisplayString,
  phySerialNum                     DisplayString,
  filterFpgaVersionRF1               DisplayString,
  agcVersionRF1                    DisplayString,
  phySerialNumRF1                    DisplayString
}

  serverIp OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 1 }
```

```
  internalIp OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 2 }

  macAddr OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 3 }

  macCVSTag OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 4 }

  macCreatedOn OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 5 }

  aecdDriverCreatedOn OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 6 }

  modemId OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 7 }

  modemVersion OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
```

```
    ::= { aeronixVersionsEntry 8 }

  modemSVersion OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 9 }

  filterFpgaVersion OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 10 }

  agcVersion OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 11 }

  phySerialNum OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 12 }

  filterFpgaVersionRF1 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 13 }

  agcVersionRF1 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 14 }

  phySerialNumRF1 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION
      ""
    ::= { aeronixVersionsEntry 15 }

--
--
--
--

aeronixStationTestsTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixStationTestsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  ::= { wmanIfCommonObjects 17 }

aeronixStationTestsEntry OBJECT-TYPE
  SYNTAX AeronixStationTestsEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    ""
  INDEX { snmpRowIndex }
  ::= { aeronixStationTestsTable 1 }

AeronixStationTestsEntry ::= SEQUENCE
{
  transmitOutputPower          Unsigned32,
  startTransmit                Unsigned32,
  startReceive                 Unsigned32,
  rSSI                         DisplayString,
  modType                      Unsigned32,
  bitPattern                   Unsigned32,
  symbols                      Unsigned32,
  runMaster                    Unsigned32,
  runSlave                     Unsigned32,
  shutdownAllTests             Unsigned32,
  resetBerResults              Unsigned32,
  testLengthInSec              DisplayString,
  totalBits                    Unsigned32,
  correctBits                  Unsigned32,
  errorBits                    Unsigned32,
  bitErrorRate                 DisplayString,
  mbitsPerSec                  DisplayString
}

  transmitOutputPower OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 1 }
```

```
startTransmit OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     ""
   ::= { aeronixStationTestsEntry 2 }

startReceive OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     ""
   ::= { aeronixStationTestsEntry 3 }

rSSI OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     ""
   ::= { aeronixStationTestsEntry 4 }

modType OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     ""
   ::= { aeronixStationTestsEntry 5 }

bitPattern OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     ""
   ::= { aeronixStationTestsEntry 6 }

symbols OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     ""
   ::= { aeronixStationTestsEntry 7 }

runMaster OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
```

```
      ""
    ::= { aeronixStationTestsEntry 8 }

  runSlave OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 9 }

  shutdownAllTests OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 10 }

  resetBerResults OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 11 }

  testLengthInSec OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 12 }

  totalBits OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 13 }

  correctBits OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixStationTestsEntry 14 }

  errorBits OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
```

```
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixStationTestsEntry 15 }

    bitErrorRate OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixStationTestsEntry 16 }

    mbitsPerSec OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixStationTestsEntry 17 }

--
--
--
--

aeronixBandwidthTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AeronixBandwidthEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
      "This table contains bandwidth data."
    ::= { wmanIfCommonObjects 19 }

aeronixBandwidthEntry OBJECT-TYPE
    SYNTAX AeronixBandwidthEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
      ""
    INDEX { snmpRowIndex }
    ::= { aeronixBandwidthTable 1 }

AeronixBandwidthEntry ::= SEQUENCE
{
    bandwidthIndex                  INTEGER,
    transportCid                   INTEGER,
    txUserBandwidthKbps             Unsigned32,
    txAllocBandwidthKbps            Unsigned32,
    serviceClass                   DisplayString,
    txUserBandwidthMbps             DisplayString,
    txAllocBandwidthMbps            DisplayString,
    serviceFlowId                  INTEGER,
    numSFPacketsDropped             Unsigned32,
```

```
  bwPrimaryCid                    INTEGER
}

  bandwidthIndex OBJECT-TYPE
    SYNTAX INTEGER (1..1000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 1 }

  transportCid OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 2 }

  txUserBandwidthKbps OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 3 }

  txAllocBandwidthKbps OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 4 }

  serviceClass OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 5 }

  txUserBandwidthMbps OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 6 }

  txAllocBandwidthMbps OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 7 }

  serviceFlowId OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 8 }

  numSFPacketsDropped OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 9 }

  bwPrimaryCid OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixBandwidthEntry 10 }

--
--
--
--

aeronixRFStatusTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AeronixRFStatusEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains RF status reports"
  ::= { wmanIfCommonObjects 20 }

aeronixRFStatusEntry OBJECT-TYPE
  SYNTAX AeronixRFStatusEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This Table is a unique aeronix structure, Used to view the RF Status"
  INDEX { snmpRowIndex }
  ::= { aeronixRFStatusTable 1 }

AeronixRFStatusEntry ::= SEQUENCE
{
  primaryCid                    INTEGER,
```

```
    cinr                        INTEGER,
    cinrMu                      INTEGER,
    rssi                        INTEGER,
    rssiMu                      INTEGER,
    rssiCurLeveled              INTEGER,
    rssiAvgLeveled              INTEGER,
    fecCorrected                INTEGER,
    fecCorrectedMu              INTEGER,
    freqOffsetMu                INTEGER,
    startTime                   INTEGER,
    currentModType              INTEGER,
    rssiRF1                     INTEGER,
    rssiMuRF1                   INTEGER,
    cpOffset                    INTEGER,
    cpOffsetRF1                 INTEGER
}

    primaryCid OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixRFStatusEntry 1 }

    cinr OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixRFStatusEntry 2 }

    cinrMu OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixRFStatusEntry 3 }

    rssi OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        ""
      ::= { aeronixRFStatusEntry 4 }

    rssiMu OBJECT-TYPE
      SYNTAX INTEGER
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
```

```
      ""
    ::= { aeronixRFStatusEntry 5 }

  rssiCurLeveled OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 6 }

  rssiAvgLeveled OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 7 }

  fecCorrected OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 8 }

  fecCorrectedMu OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 9 }

  freqOffsetMu OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 10 }

  startTime OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 11 }

  currentModType OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 12 }

  rssiRF1 OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 13 }

  rssiMuRF1 OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 14 }

  cpOffset OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 15 }

  cpOffsetRF1 OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      ""
    ::= { aeronixRFStatusEntry 16 }
--
--
--
--

END
```

Appendix C: Definitions, Acronyms, and Abbreviations

The section provides the necessary definitions, acronyms, and abbreviations relevant to this document and necessary in order to understand this document.

ACID – ARQ Channel ID

ARP – Address Resolution Protocol

ARQ – Automatic Repeat Request

ATM – Asynchronous Transfer Mode

Authentication – Verification of the source of information

BS – Base Station

CCD – Charge-Coupled Device

CCV – Clock Comparison Value

CDMA – Code Division Multiple Access

CID – Connection Identifier

CINR – Carrier to Interference and Noise Ratio

CS – Convergence Sub-layer

DCD – Downlink Channel Descriptor

DFS – Dynamic Frequency Selection

DIUC – Downlink Interval Usage Code

DL – Downlink

EDL – micro Enhanced Data Link

EIRP – Effective Isotropic Radiated Power

GMSK - Gaussian Minimum Shift Keying

GPS – Global Positioning System

H-ARQ – Hybrid Automatic Repeat Request

HMAC – Hashed Message Authentication Code

HW – Hardware

Instantiate – Create an entity from an abstraction.

Integrity – Verification that the information has not been altered.

I/O – Input/Output

IP – Internet Protocol

JTRS – Joint Tactical Radio System

MAC – Media Access Control

MSB – Most Significant Bit

OFDM – Orthogonal Frequency Division Multiplexing

OFDMA – Orthogonal Frequency Division Multiple Access

OID – Object Identifiers

OS – Operating System

PDU – Payload Data Unit

PHS – Payload Header Suppression

PHY – Physical Layer

PKM – Privacy Key Management

PTP – Point To Point

PTMP – Point To Multipoint

QoS – Quality of Service

RSSI – Received Signal Strength Indicator

RTG – Receive/Transmit Transition Gap

SDU – Service Data Unit

SFID – Service Flow ID

SNMP – Simple Network Management Protocol

SPB – Samples Per Bit

SS – Subscriber Station

SW – Software

TCP – Transmission Control Protocol

TLV – Type Length Value

TRANSEC – Transmission Security

TTG – Transmit/Receive Transition Gap

UCD – Uplink Channel Descriptor

UDP – User Datagram Protocol

UIUC – Uplink Interval Usage Code

UL – Uplink

EDL - micro EDL

USM – User-based Security

VLAN – Virtual Local Area Network