Scenargie®2.1 Dot Eleven Module

User Guide

Space-Time Engineering, LLC
September 2016

Copyright© 2007-2016 Space-Time Engineering, LLC

Contents

Copyright© 2007-2016 Space-Time Engineering, LLC

Copyright© 2007-2016 Space-Time Engineering, LLC

Preface

This document describes how to use Scenargie 2.1 Dot Eleven Module, an extension module of Scenargie 2.1.

Related documents

Installation Guide
Programmer’s Guide
Visual Lab User Guide
Base Simulator User Guide

Copyright© 2007-2016 Space-Time Engineering, LLC

1. Dot Eleven Module Overview

Scenargie 2.1 Dot Eleven Module is an extension module for Scenargie 2.1 Base Simulator. This module should be executed in combination with Scenargie Base Simulator.

Fig. 1-1 depicts a whole structure of Scenargie and a blue colored block (🟦) corresponds to Scenargie 2.1 Dot Eleven Module.

There are two Dot Eleven Modules. The one is Standard version which includes simulation models based on the IEEE802.11a/g/n/ac standards. The other is Advanced version which includes simulation models based on the IEEE802.11ad and 11ah standards as well as the simulation modules of the 11a/g/n/ac standards. This Advanced version also supports MIMO channel models (named TGn channel model) specified by the IEEE802.11n standard.

Scenargie 2.1 Dot Eleven Module includes the following items.

- Source code of simulation models for IEEE802.11 specifications [1], [2], [3], [4], [5], [6]
- makefile
- Sample scenarios

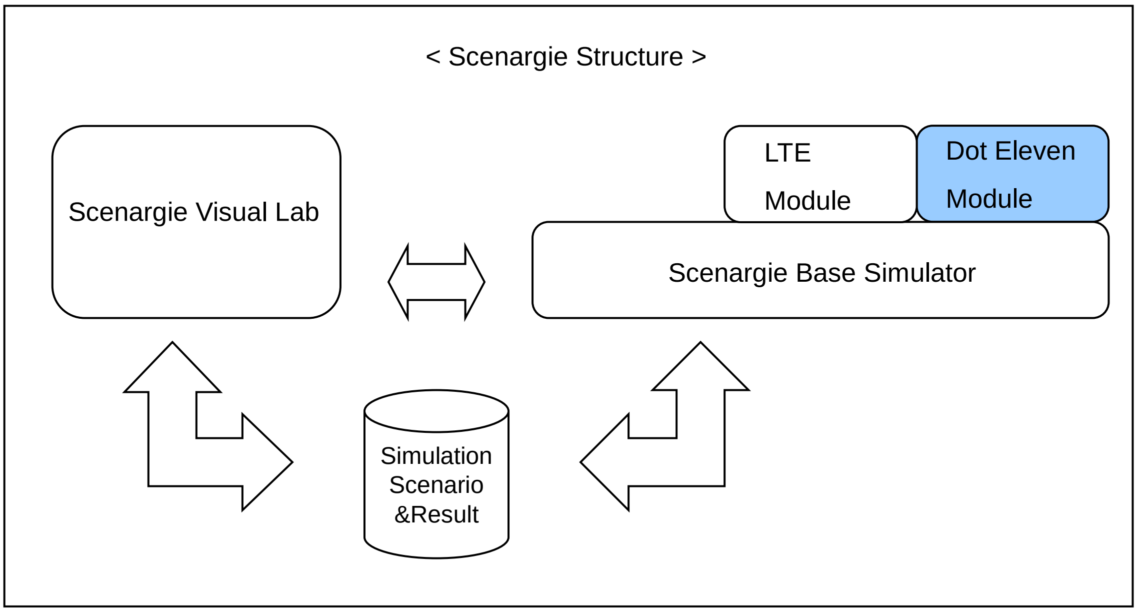


Fig. 1-1 Scenargie system structure

Copyright© 2007-2016 Space-Time Engineering, LLC

2. Installation

Scenargie 2.1 Dot Eleven Module package can be installed on Linux, Mac or Windows operating system (OS). Follow the next instructions after installing Scenargie Base Simulator on your target OS to use this module.

After extracting Scenargie Base Simulator and this Scenargie 2.1 Dot Eleven Module, a structure of directories is as follows.

```
scenargie_simulator/2.1
|-- document
|-- package_tree
|-- scenarios_linux
|-- scenarios_windows
`-- source
|     |-- base
|     |-- boost
|     |-- dot11
|     |-- include
|     |-- multisystems
|     |-- objlibs
|     `-- simulator
|-- util
`--visuallab
       |-- data
      `-- sample
```

Copyright© 2007-2016 Space-Time Engineering, LLC

2.1. Installation on Linux or Mac OS

2.1.1. Building executable file

1) Extracting the package

In this document, we assume a directory named “scenargie_simulator” has already been created in a user’s home directory after installing Scenargie Base Simulator. Copy or move a package of Scenargie Dot Eleven Module into the user’s home directory where the “scenargie_simulator” directory exists. Then, unzip the package as follows.

<Example of commands>

```
$ cp Scenargie-2.1-DotElevenModule-rxxxx.zip ~/
$ cd ~/
$ unzip Scenargie-2.1-DotElevenModule-rxxxx.zip
```

2) Building a simulation executable file

A new directory named “dot11” is created in a directory of “scenargie_simulator/2.1/source/” after extracting Scenargie 2.1 Dot Eleven Module. The directory includes a sample makefile named “makefile.linux.” Build a simulation executable file by using the makefile.linux as follows.

<Example of commands>

```
$ cd ~/scenargie_simulator/2.1/source/dot11
$ make –f makefile.linux
```

Success of the build creates a simulation executable file named “sim” in the “dot11” directory.

Build options should be set to activate IPv6 support and so on if necessary. Use a makefile for a multi-system when more than one extension modules are required to build a simulator, and specify the necessary extension module names. Refer “Installation Guide” for details.

Please use a make file included in a directory named “dot11/ad_version” or “dot11/ah_version” to create a simulation executable file instead of the make file in the directory “dot11” when your target simulation model is 11ad or 11ah. The followings are example commands to create a simulation executable file for 11ad and 11ah simulation model.

<Example of commands>

```
$ cd ~/scenargie_simulator/2.1/source/dot11/ad_version
$ make –f makefile.linux
```

```
$ cd ~/scenargie_simulator/2.1/source/dot11/ah_version
$ make –f makefile.linux
```

Note)

If you want to rebuild a simulation executable file, the old simulation executable file and object files should be removed.

<Example of commands>

```
$ cd ~/scenargie_simulator/2.1/source/dot11
$ make –f makefile.linux clean
$ make –f makefile.linux
```

2.1.2. Setting for Visual Lab on Linux

Follow the next instructions to utilize Scenargie 2.1 Dot Eleven Module through Scenargie Visual Lab on Linux.

1) Copying data files

Scenargie 2.1 Dot Eleven Module package includes data files to utilize Dot Eleven Module through Scenargie Visual Lab. Copy the directory named “data/dot11” including those files to a directory where Scenargie Visual Lab has been installed.

Data files：

scenargie_simulator/2.1/visuallab/data/dot11/dot11.component
scenargie_simulator/2.1/visuallab/data/dot11/dot11.objtype
scenargie_simulator/2.1/visuallab/data/dot11/dot11_advanced.component
(only Advanced version includes this file.)
scenargie_simulator/2.1/visuallab/data/dot11/dot11_advanced.objtype
(only Advanced version includes this file.)

Target directory to be copied：

A directory named “data” under a directory where Scenargie Visual Lab is installed

<Example>

visuallab/data/dot11/

2) Copying sample files

Scenargie 2.1 Dot Eleven Module package includes a sample file to utilize Dot Eleven Module through Scenargie Visual Lab. Copy it to a directory where Scenargie Visual Lab has been installed.

Sample file：

scenargie_simulator/2.1/visuallab/sample/dot11modes.ber
Advanced version also includes the following files.
scenargie_simulator/2.1/visuallab/sample/dot11admodes.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel1.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel2.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel3.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel4.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel5.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel6.ber
scenargie_simulator/2.1/visuallab/sample/dot11ahmodes.ber

Target directory to be copied：

A directory named “sample” under a directory where Scenargie Visual Lab has been installed.

<Example>

visuallab/sample/

3) Specifying executable file

Specify a simulation executable file name at [Tools]–[Object Properties] Global: Simulation Executable Name.

Copyright© 2007-2016 Space-Time Engineering, LLC

2.1.3. Setting for Visual Lab on Mac OS

Follow the next instructions to execute Scenargie 2.1 Dot Eleven Module through Scenargie Visual Lab on Mac OS.

1) Copy of data files

Scenargie 2.1 Dot Eleven Module package includes data files to utilize Dot Eleven Module through Scenargie Visual Lab. Copy a “data/dot11” directory including those files to a directory where Scenargie Visual Lab has been installed.

Data files：

scenargie_simulator/2.1/visuallab/data/dot11/dot11.component
scenargie_simulator/2.1/visuallab/data/dot11/dot11.objtype
scenargie_simulator/2.1/visuallab/data/dot11/dot11_advanced.component
(Only Advanced version includes this file.)
scenargie_simulator/2.1/visuallab/data/dot11/dot11_advanced.objtype
(Only Advanced version includes this file.)

Target directory to be copied：

A directory named “data” under a directory where Scenargie Visual Lab has been installed.

<Example>

/Applications/Scenargie.app/Contents/data/dot11/

2) Copying sample files

Scenargie Dot Eleven Module package includes a sample file to utilize Dot Eleven Module through Scenargie Visual Lab. Copy it to a directory where Scenargie Visual Lab has been installed.

Sample file：

scenargie_simulator/2.1/visuallab/sample/dot11modes.ber
Advanced version also includes the following files.
scenargie_simulator/2.1/visuallab/sample/dot11admodes.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel1.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel2.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel3.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel4.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel5.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel6.ber
scenargie_simulator/2.1/visuallab/sample/dot11ahmodes.ber

Target directory to be copied：

A directory named “sample” under a directory where Scenargie Visual Lab has been installed.

<Example>

/Applications/Scenargie.app/Contents/sample/

3) Specifying a simulation executable file

Specify a simulation executable file name at [Tools] – [Object Properties] Global: Simulation Executable Name.

Copyright© 2007-2016 Space-Time Engineering, LLC

2.2. Installation on Windows

Administrator account must be used to install Scenargie Dot Eleven Module to Windows. In an example of commands, it is assumed to use “Visual Studio 2013 Command Prompt”.

When your OS is 32 bits version, use a Command Prompt from [Start] – [Program] – [Visual Studio 2013] – [Visual Studio Tools] – [VS2013 x86 Native Tools Command Prompt].

When your OS is 64 bits version, use a Command Prompt from [Start] – [Program] – [Visual Studio 2013] – [Visual Studio Tools] – [VS2013 x64 Cross Tools Command Prompt].

2.2.1. Building executable file

1) Extracting the package

In this document, we assume that a directory named “scenargie_simulator” has already created in “C:¥scenargie_simulator” after installing Scenargie Base Simulator. Copy or move a package of Scenargie Dot Eleven Module into the directory where the “scenargie_simulator” directory exists. Then, unzip the package as follows.

<Example>

Extract Scenargie-2.1-DotElevenMdoule-Standard-rxxxx.zip or Scenargie-2.1-DotElevenMdoule-Advanced-rxxxx.zip under C:¥ directory

2) Building a simulation executable file

A new directory named “dot11” is created in a directory of “scenargie_simulator¥2.1¥source” after extracting Scenargie 2.1 Dot Eleven Module. The directory includes a makefile named makefile.win. Build a simulation executable file by using the makefile.win as follows.

<Example of commands>

```
> cd C:¥scenargie_simulator¥2.1¥source¥dot11
> nmake -f makefile.win
```

Success of the build creates a simulation executable file named “sim.exe” in the “dot11” directory.

Build options should be set to activate IPv6 support and so on if necessary. Use a makefile for a multi-system when more than one extension modules are required to build a simulator, and specify the extension module names which you want to enable. Refer “Installation Guide” for details.

Please use a make file included in a directory named “dot11/ad_version” or “dot11/ah_version” to create a simulation executable file instead of the make file in the directory “dot11” when your target simulation model 11ad or 11ah. The followings are example commands to create a simulation executable file for 11ad and 11ah simulation model.

<Example of commands>

```
> C:¥scenargie_simulator¥2.1¥source¥dot11¥ad_version
> nmake –f makefile.win
```

```
> C:¥scenargie_simulator¥2.1¥source¥dot11¥ah_version
> nmake –f makefile.win
```

Note)

If you want to rebuild a simulation executable file, the old simulation executable file and object files should be removed.

<Example of commands>

```
> cd C:¥scenargie_simulator¥2.1¥source¥dot11
> namke –f makefile.win clean
> nmake -f makefile.win
```

2.2.2.Setting for Visual Lab on Windows

Follow the next instructions to utilize Scenargie 2.1 Dot Eleven Module through Scenargie Visual Lab on Windows.

1) Copying data files

Scenargie 2.1 Dot Eleven Module package includes data files to utilize Dot Eleven Module through Scenargie Visual Lab. Copy the directory named “data¥dot11” including those files to a directory where Scenargie Visual Lab has been installed.

Copyright© 2007-2016 Space-Time Engineering, LLC

Data files：

scenargie_simulator¥2.1¥visuallab¥data¥dot11¥dot11.component

scenargie_simulator¥2.1¥visuallab¥data¥dot11¥dot11.objtype

scenargie_simulator¥2.1¥visuallab¥data¥dot11¥dot11_advanced.component

(Only Advanced version includes this file.)

scenargie_simulator¥2.1¥visuallab¥data¥dot11¥dot11_advanced.objtype

(Only Advanced version includes this file.)

Target directory to be copied：

A directory named “data” under a directory where Scenargie Visual Lab has been installed.

<Example>

C:¥Scenargie¥data¥dot11¥

2) Copying sample files

Scenargie Dot Eleven Module package includes a sample file to utilize Dot Eleven Module through Scenargie Visual Lab. Copy it to a directory Scenargie Visual Lab is installed.

Sample file：

scenargie_simulator¥2.1¥visuallab¥sample¥dot11modes.ber

Advanced version also includes the following files.

scenargie_simulator/2.1/visuallab/sample/dot11admodes.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel1.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel2.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel3.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel4.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel5.ber

scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel6.ber

scenargie_simulator/2.1/visuallab/sample/dot11ahmodes.ber

Target directory to be copied：

A directory named “sample” under a directory where Scenargie Visual Lab has been installed.

<Example>

C:¥Scenargie¥sample¥

3) Specifying a simulator executable file

Specify a simulation executable file name at [Tools] – [Object Properties] Global: Simulation Executable Name.

Note:

The above step1) and 2) can be done by executing the following script file.

Installation script file:

scenargie_simulator¥2.1¥visuallab¥install-dot11-win.vbs

Copyright© 2007-2016 Space-Time Engineering, LLC

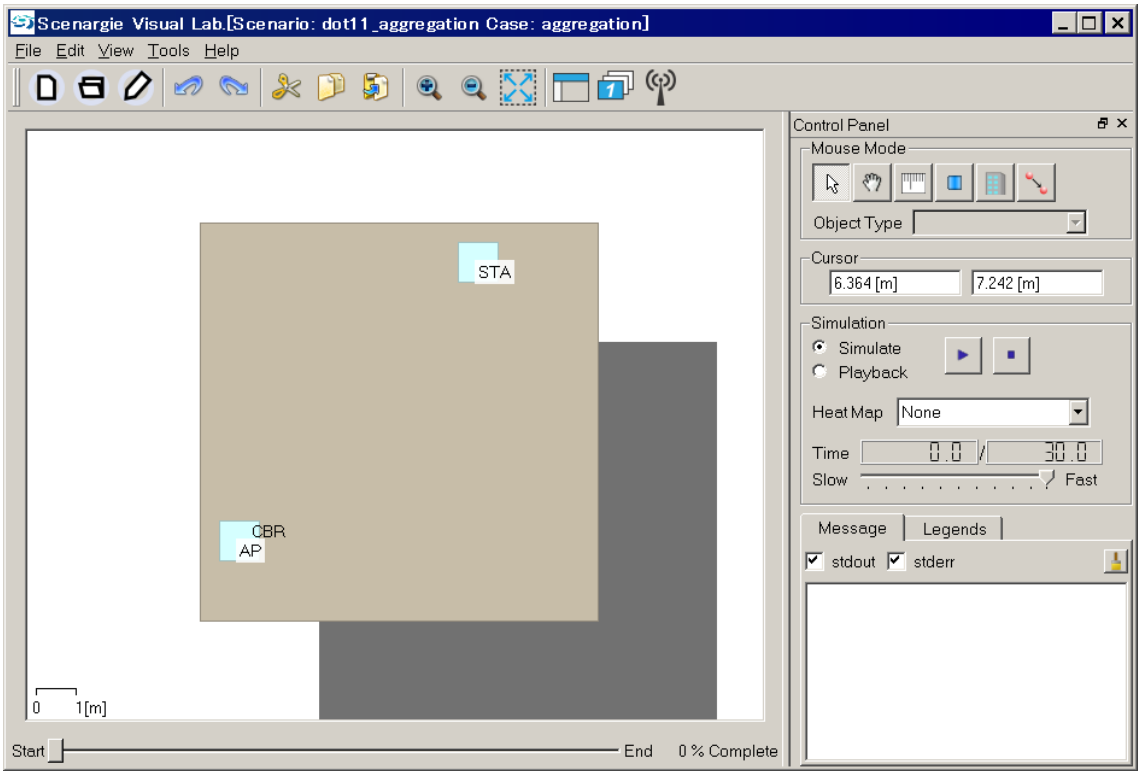
2.3. Sample scenarios

Sample scenarios are crated in the following directories after extracting the package mentioned in the previous section (see 2.1 for Linux and Mac OS and 2.2 for Windows).

scenargie_simulator/2.1/scenarios_linux/
scenargie_simulator/2.1/scenarios_windows/

Sample scenarios of Dot Eleven Module are located in the directories that the name starts with “dot11_”. Each scenario directory includes a file with a suffix “.case” which is for Scenargie Visual Lab. A scenario for command line execution is stored in a directory named “commandline” in each sample scenario directory.

2.3.1.dot11_ aggregation



Scenargie Visual Lab.[Scenario: dot11_aggregation Case: aggregation]

Cursor: 6.364 [m] 7.242 [m]

Simulation: ● Simulate ○ Playback

Heat Map: None

Time 0.0 / 30.0

☑ stdout ☑ stderr

- Frame of scenario

Communication objects (IEEE802.11n)：

- AP（stationary）× 1
- STA（mobile）× 1

Application:

Copyright© 2007-2016 Space-Time Engineering, LLC

CBR ： AP → STA

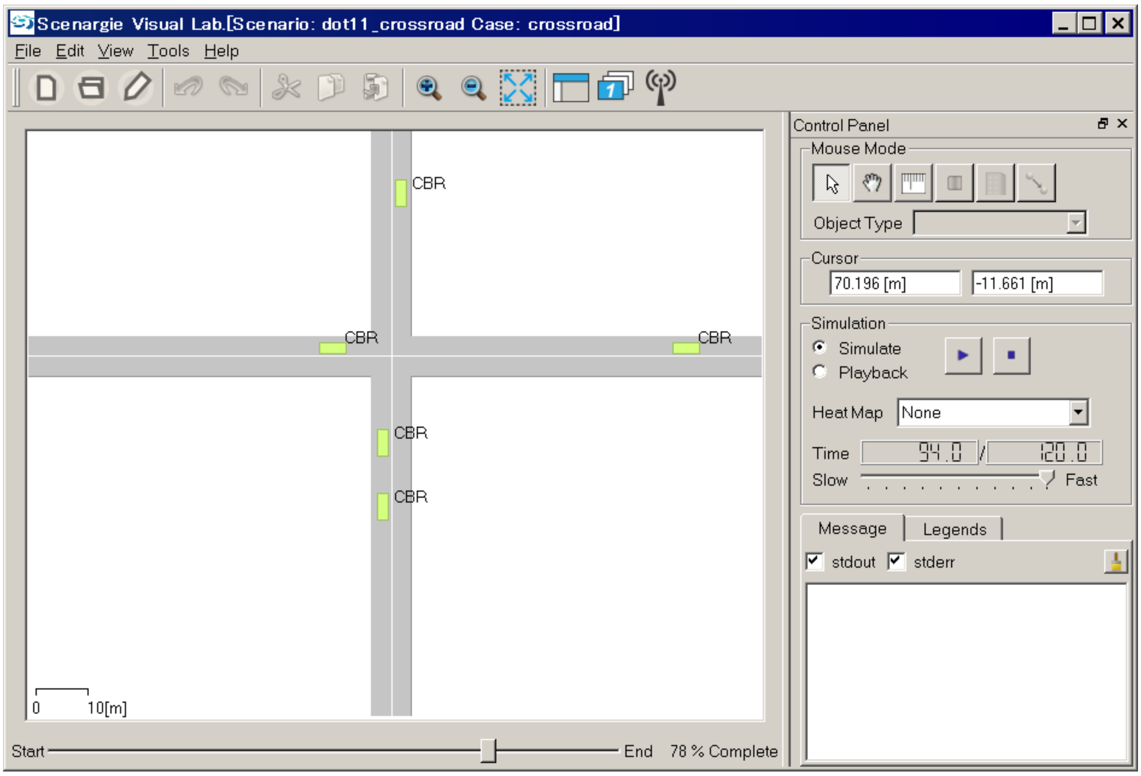
Path loss model:

WallCount

- Description

This scenario simulates a performance of IEEE802.11n in an indoor environment. An AP is located on the first floor and a STA is located on the second floor and moves around randomly on the floor. CBR application between the AP and the STA are transmitted with A-MPDU mechanism.

2.3.2.dot11_crossroad



Scenargie Visual Lab.[Scenario: dot11_crossroad Case: crossroad]

- Frame of scenario

Communication objects (IEEE802.11p):

－ Dot11p（mobile）× 27

Application：

CBR ： Dot11p → * (Broadcast)

Copyright© 2007-2016 Space-Time Engineering, LLC

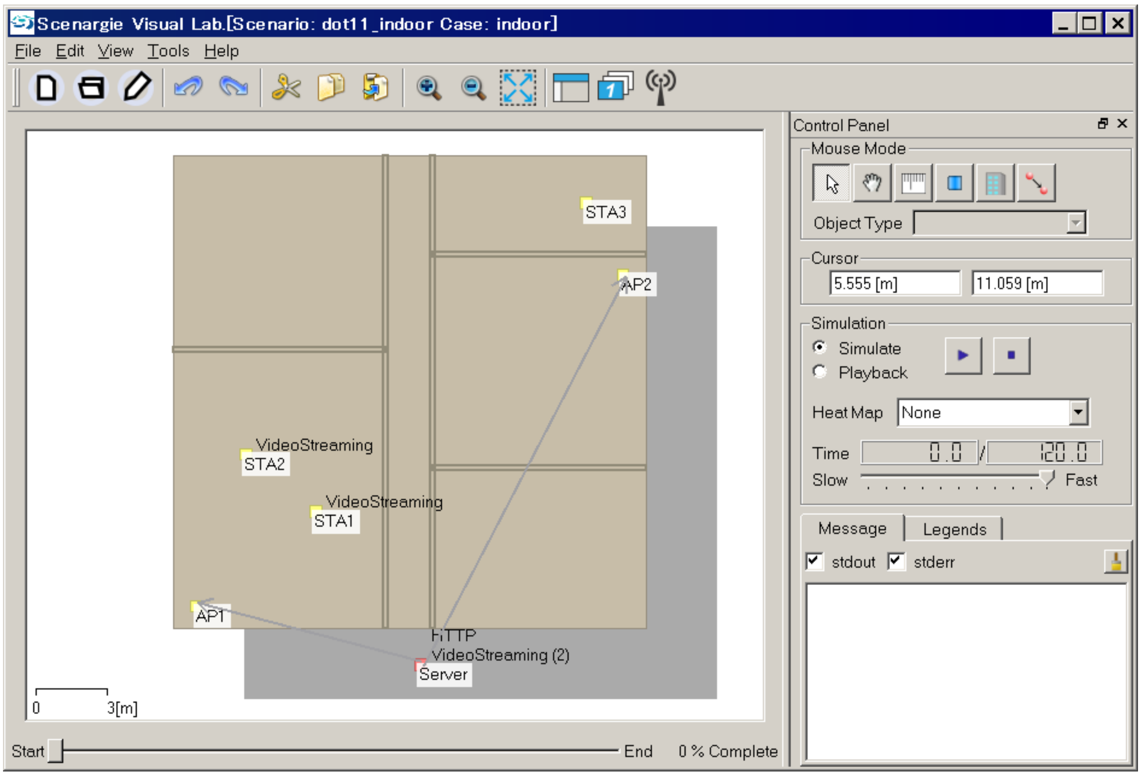
- Description

This scenario uses a mobility file which lists time and node positions. Communication objects start at edges of a road, move along with the roads, and get out of a designated area shown in the window.

Note: Mobility of communication objects used in the scenario utilizes trace information generated by MATES*.

* Multi-Agent based Traffic and Environment Simulator: Intelligent Multi-Agent traffic flow simulator developed by Dr. Yoshimura's laboratory of the University of Tokyo.

2.3.3.dot11_indoor


Scenargie Visual Lab.[Scenario: dot11_indoor Case: indoor]

Cursor: 5.555 [m] 11.059 [m]

Simulation: ● Simulate ○ Playback

Heat Map: None

Time: 0.0 / 120.0

☑ stdout ☑ stderr

- Frame of scenario

Communication objects：

- Server (Wired) × 1
- AP (Dot11gAndWired) × 27
- STA (Dot11g) × 3

Application:

Copyright© 2007-2016 Space-Time Engineering, LLC

VideoStreaming : Server → STA1、STA2

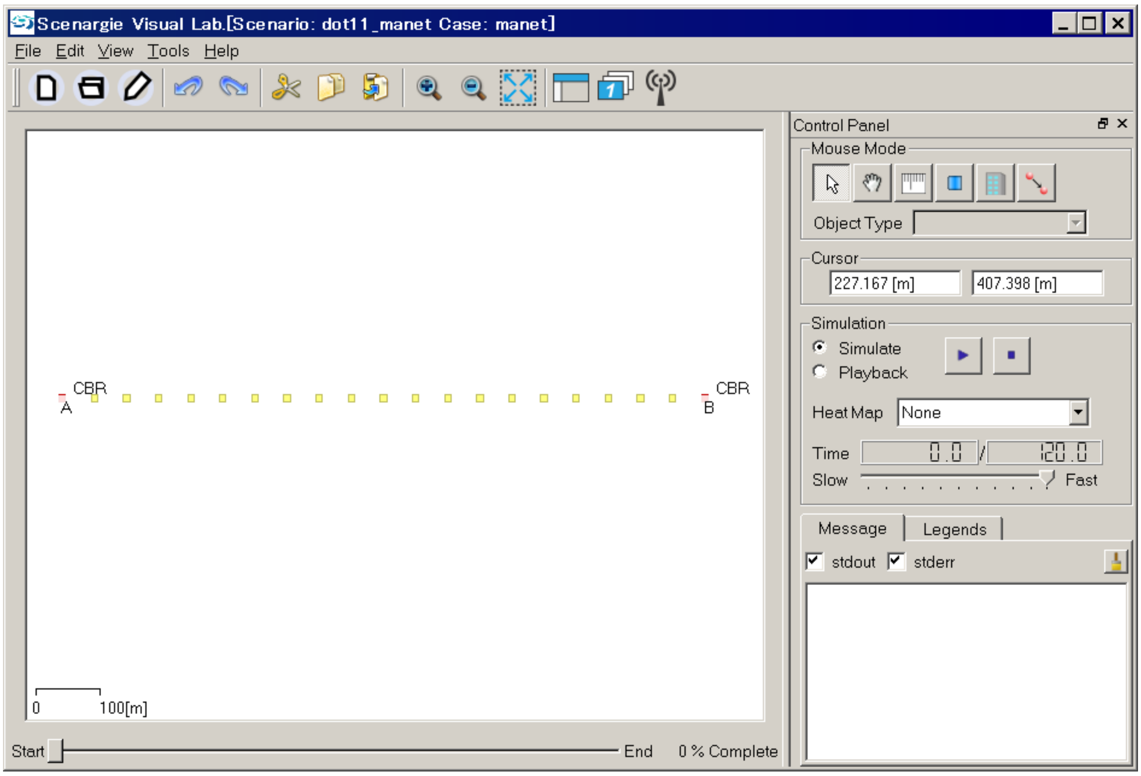
VideoStreaming : STA1、STA2 → Server

HTTP : Server → STA3

- Description

This scenario simulates a performance of IEEE802.11g in an indoor environment. COST231Indoor model which can consider losses of walls is used as a propagation model.

2.3.4.dot11_manet



Scenargie Visual Lab.[Scenario: dot11_manet Case: manet]

Cursor: 227.167 [m] 407.398 [m]

Simulation: ● Simulate ○ Playback

Heat Map: None

Time: 0.0 / 120.0

☑ stdout ☑ stderr

- Frame of scenario

Communication objects：

− Dot11g × 21

Application：

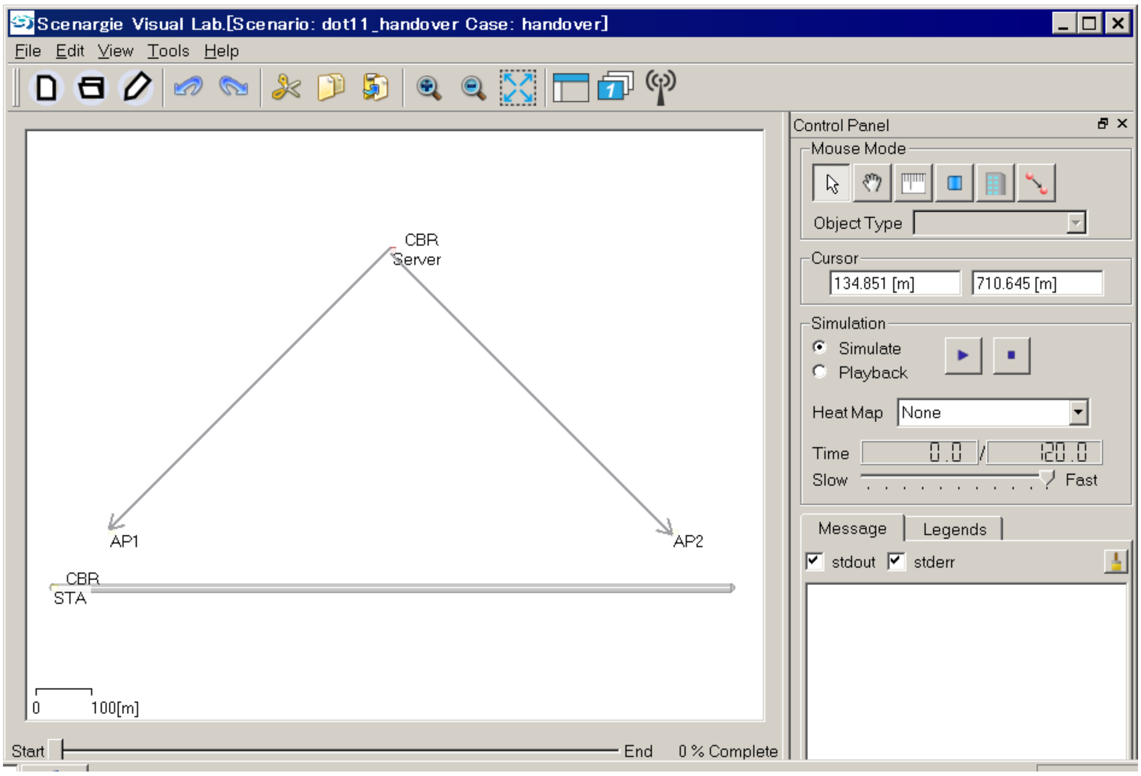
CBR : A → B

CBR : B → A

- Description

This scenario simulates an ad-hoc network performance when OLSRv2 (Optimized Link State Routing) is applied. Packets of CBR application from STA A are forwarded by multiple nodes between STA A and STA B and are delivered to STA B.

2.3.5.dot11_handover



- Frame of scenario

Communication objects

- Server×1
- AP（IEEE802.11g)×2
- STA（IEEE802.11g)×1

Application：

CBR ： Server → STA

CBR ： STA → Server

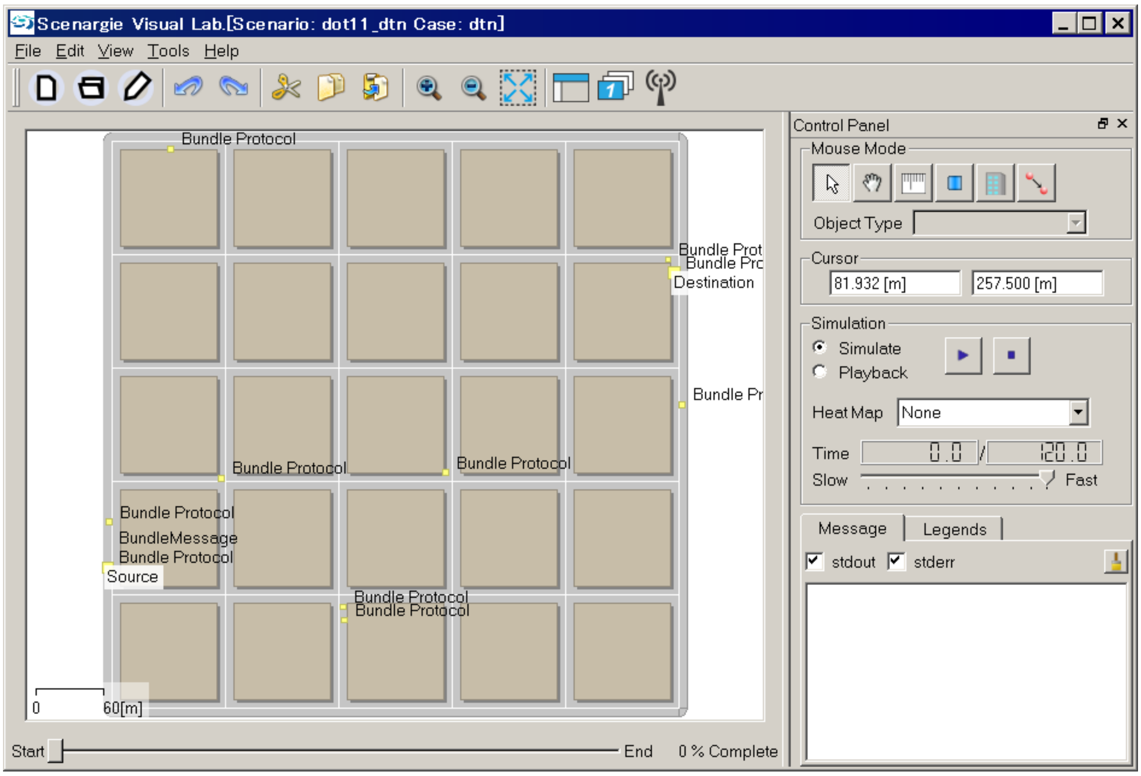
- Description

This scenario simulates a handover performance of a STA traveling from one AP’s communication range to another AP’s communication range. At the beginning of this simulation, a STA communicates with a Server via AP1. After occurring handover, the STA communicates

Copyright© 2007-2016 Space-Time Engineering, LLC

with the server via AP2. In this scenario, IP address of the STA is assigned dynamically by DHCP mechanism.

2.3.6.dot11_dtn


Scenargie Visual Lab.[Scenario: dot11_dtn Case: dtn]

Cursor 81.932 [m] 257.500 [m]

Simulation ● Simulate ○ Playback

Heat Map None

Time 0.0 / 120.0

☑ stdout ☑ stderr

Start End 0 % Complete

- Frame of scenario

Communication objects (IEEE802.11g)

− Source × 1

− Destination × 1

− Relay node × 8

Application：

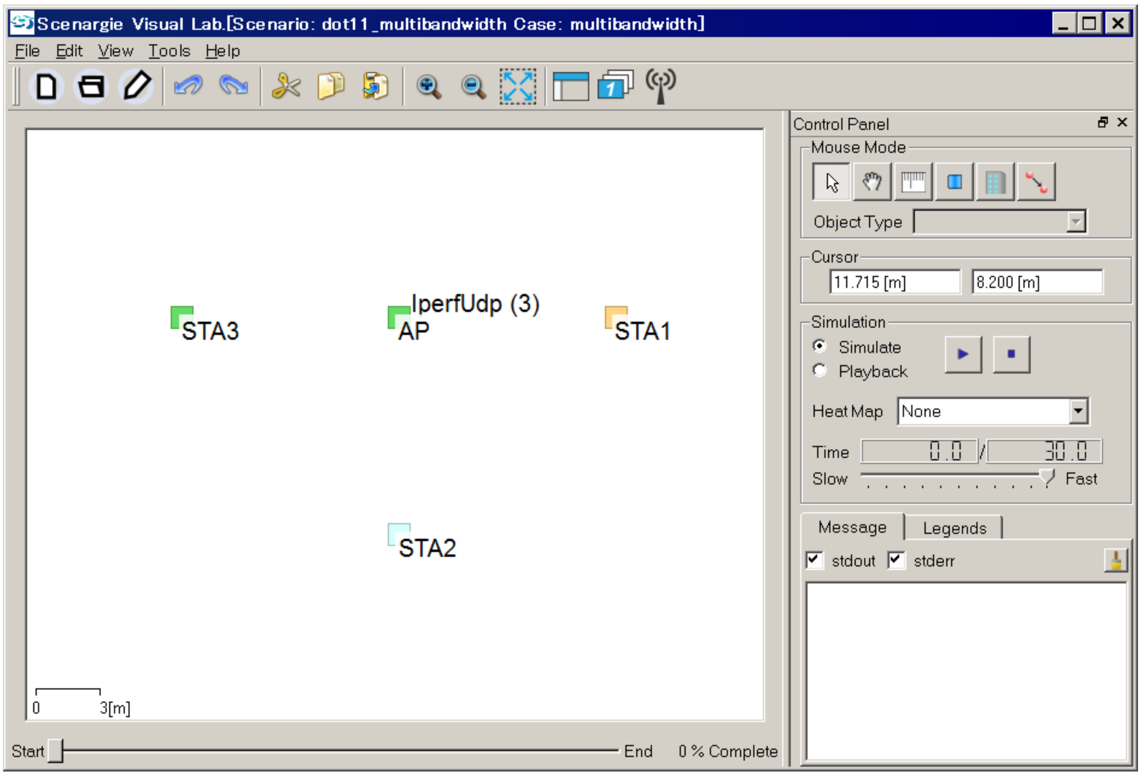
BundleMessage ： Source → Destination

- Description

This scenario simulates a performance of DTN (Delay / Disruption Tolerant Network). When there is no direct link between a Source node and a Destination node, data is conveyed via a relay node between a Source node and a Destination node by bundle protocol (Epidemic routing algorithm). Connections between two nodes which send and receive a Bundle message can be displayed on a screen by Trace Visualization Settings.

Copyright© 2007-2016 Space-Time Engineering, LLC

2.3.7.dot11_multibandwidth



- Frame of scenario

Communication objects

－ Dot11a × 1 (STA)

－ Dot11n × 1 (STA)

－ Dot11ac × 2 (AP, STA3)

Application :

IperfUdp : AP → STA1

IperfUdp : AP → STA2

IperfUdp : AP → STA3

- Description

This scenario simulates a performance of communication between AP and STAs with different channel bandwidths by using the IEEE802.11ac standard. STA1, STA2 and STA3 occupy 20MHz, 40MHz and 160MHz channel bandwidths respectively. The Iperf application is applied on the communication between AP and STAs.

Copyright© 2007-2016 Space-Time Engineering, LLC

2.4. Running old sample scenarios

This section describes how to run old sample scenarios with the latest Scenargie 2.1 Visual Lab and an executable file newly built.

Note: Make sure that data files and sample files are installed correctly before doing the next things.

2.4.1. How to run sample scenarios shipped with Scenargie1.8 r19737 / Scenargie 2.0 r19737

dot11_indoor

After opening indoor.case with Visual Lab, change the following property values through Object Properties.

Property	Before	After
Dot11g->STA1->interface/dot11g->Network (Interface)->Gateway Address	192.168.1.254	192.168.1.2
Dot11g->STA3->interface/dot11g->Network (Interface)->Gateway Address	192.168.1.254	192.168.1.2
Dot11g->STA3->interface/dot11g->Network (Interface)->Gateway Address	192.168.2.254	192.168.2.3
Dot11g->AP1->interface/dot11g->Network (Interface)->Interface Network Address	192.168.1.254	192.168.1.0 + $n
Dot11g->AP1->interface/dot11g->Network (Interface)->Interface Network Address	192.168.2.254	192.168.2.0 + $n

Regarding command line execution, change the following parameters in the configuration file.

Before	After
[4;dot11g] network-gateway-address = 192.168.1.254	[4;dot11g] network-gateway-address = 192.168.1.2
[5;dot11g] network-gateway-address = 192.168.1.254	[5;dot11g] network-gateway-address = 192.168.1.2
[6;dot11g] network-gateway-address = 192.168.2.254	[6;dot11g] network-gateway-address = 192.168.2.3
[2;dot11g] network-address =	[2;dot11g] network-address =

Copyright© 2007-2016 Space-Time Engineering, LLC

192.168.1.254	192.168.1.0 + $n
[3;dot11g] network-address = 192.168.2.254	[3;dot11g] network-address = 192.168.2.0 + $n

dot11_ad

Copy dot11admodes.ber from the scenario directory (dot11_ad) in the Scenargie 2.1 Dot Eleven Module Advanced to the target scenario directry.

Regarding command line execution, add the following parameters in the configuration file.

```
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs1 = 385.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs2 = 770.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs3 = 962.500000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs5 = 1251.250000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs6 = 1540.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs7 = 1925.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs8 = 2310.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs10 = 3080.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs11 = 3850.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs13 = 693.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs14 = 866.250000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs15 = 1386.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs16 = 1732.500000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs17 = 2079.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs18 = 2772.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs19 = 3465.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs20 = 4158.000000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs21 = 4504.500000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs22 = 5197.500000000
[1-2;dot11ad] dot11ad-mbps-datarate-for-dot11ad-mcs24 = 6756.750000000
```

Other scenarios

No special operation is needed for the scenarios exclusing dot11_indoor and dot11_ad. Execute the scenario as-is.

Copyright© 2007-2016 Space-Time Engineering, LLC

2.4.2. How to run sample scenarios shipped with Scenargie2.0 r20324

Running sample scenarios for Visual Lab

No special operation is needed. Execute the scenario as-is.

Running samples scenarios for command line execution

No special operation is needed. Execute the scenario as-is.

Copyright© 2007-2016 Space-Time Engineering, LLC

3. Properties

3.1. List of properties

The following tables show properties for Scenargie Dot Eleven Module and can be specified with a configuration file. Refer to "Scenargie Base Simulator User Guide" to learn how to write a configuration file and common properties for Scenargie Base Simulator.

Set a certain value for a parameter which does not have a default value (marked as "Not Assigned") because any model does not specify an initial value for the parameter.

Time type parameters must be written in the following format. Note that values without unit are treated as second.

[Time value] [Unit]

Examples

100 seconds:	100s or 100
100 milliseconds:	100ms or 0.1
100 microseconds:	100us or 0.0001
100 nanoseconds:	100ns or 0.0000001

Note: Default values shown in the following tables are different from default values set by Visual Lab.

3.1.1. Properties for Dot Eleven Standard

Properties for system settings

Parameter name	Scope	Type	Default value	Description
dot11-bit-error-rate-curve-file	Global	String	(Not Assigned)	File name used for BER (Bit Error Rate) curve
dot11-channel-model	Global	String	(Not Assigned)	Selection of channel model Either of the following must be set when MIMO channel model is applied. Available model name - MIMO - TGnMIMO

Copyright© 2007-2016 Space-Time Engineering, LLC

Properties for MAC

Parameter name	Scope	Type	Default value	Description
dot11-node-type	Interface	String	Ad-hoc	Following three values are available. - Access-Point - Mobile-STA - Ad-hoc
dot11-map-ip-multicast-addresses	Interface	Bool	false	Activation of multicast mode function. This mode is available only when CBR or VBR application is used for Ad-hoc node or AP node. "true" must be set for sender and receiver nodes using this function.
dot11-multicast-group-number-list	Interface	String	Null	List of group number. This group number is used only when IP multicast is active. Example: Set "0 1" when a STA belongs to group number 0 and 1.
dot11-enable-high-throughput-mode	Interface	Bool	false	Activation of HT (High Throughput) mode.
dot11-force-use-of-high-throughput-frames	Interface	Bool	false	Activation of HT mode all the time whenever HT mode operation is set.
dot11-bonded-channel-number-list	Interface	String	(Not Assigned)	List of channel numbers used for channel bonding in HT mode. When channel 0 and 1 is used for bonding, write a value as follows. "0 1"
dot11-max-channel-bandwidth-mhz	Interface	Real	(Not Assigned)	The maximum channel bandwidth for channel bonding.
dot11-rts-threshold-size	Interface	Integer	2346	Threshold length in bytes of

-bytes				PSDU (Physical layer Service Data Unit). RTS frame is transmitted before sending PSDU when PSDU length is larger than this value.
dot11-short-frame-retry-limit	Interface	Integer	7	Retry limit for short frames, which corresponds to “Short Retry Counter” specified in IEEE802.11 standard. This value determines the maximum number of retransmission of PSDU with shorter than the value of dot11-rts-threshold-size-bytes.
dot11-long-frame-retry-limit	Interface	Integer	4	Retry limit for long frames, which corresponds to “Long Retry Counter” specified in IEEE802.11 standard. This value determines the maximum number of retransmission of PSDU with larger than the value of dot11-rts-threshold-size-bytes.
dot11-disallow-adding-new-packets-to-retries	Interface	Bool	true	Indication to disallow to add new packet into A-MPDU in retries (true: new packet is not added into retried A-MPDU).
dot11-contention-window-min-slots	Interface	Integer	15	The number of minimum slots for calculation of DCF and EDCA contention window sizes.
dot11-contention-window-max-slots	Interface	Integer	1023	The number of maximum slots for calculation of DCF and EDCA contention window sizes.
dot11-disabled-to-jump-on-medium-without-backoff	Interface	Bool	false	Inactivation of backoff procedure before a frame transmission when wireless medium is idle

				during DIFS. true: Execution of backoff procedure false: No backoff procedure
dot11-max-packet-priority	Interface	Integer	3	Maximum number of packets priority
dot11-qos-type	Interface	String	EDCA	QoS type. The following types are available. DCF EDCA
dot11-dcf-num-difs-slots	Interface	Integer	2	The number of slots for DIFS in DCF access procedure.
dot11-dcf-contention -window-min-slots	Interface	Integer	15	The minimum number of slots of contention window (CW) in DCF.
dot11-dcf-contention -window-max-slots	Interface	Integer	23	The maximum number of slots of contention window (CW) in DCF.
dot11-dcf-frame-lifetime	Interface	Integer	infinite	Lifetime of a frame in a transmission queue in DCF.
dot11-num-edca-access -categories	Interface	Integer	4	The number of access categories in EDCA. Allowable maximum number is 4.
dot11-edca-category-<n> -priority-list <n> indicates a number of access category	Interface	String	0 for AC0, 1 for AC1, 2 for AC2, 3 for AC3	List of priority of access category (AC) <n> in EDCA. When priorities 0 and 1 belong to AC0, write as follows. dot11-edca-category -0-priority-list = “0 1”
dot11-edca-category-<n> -num-aifs-slots <n> indicates a number of access category	Interface	Integer	9 for AC0, 6 for AC1, 3 for AC2, 2 for AC3	AIFSN value for access category <n> in EDCA. When AIFSN is 9 for AC0, write as follows. dot11-edca-category-0-num -aifs-slots = 9
dot11-edca-category-<n> -contention-window-min-slots	Interface	Integer	15 for AC0, 7 for AC1,	The minimum number of slots of contention window (CW) for access category <n> in EDCA

Copyright© 2007-2016 Space-Time Engineering, LLC

<n> indicates a number of access category			3 for AC2, 3 for AC3	access category. When the minimum number of slots for AC0is 15, write as follows. dot11-edca-category-0-contention-window-min-slots = 15
dot11-edca-category-<n> -contention-window-max-slots <n> indicates a number of access category	Interface	Integer	1023 for AC0, 1023 for AC1, 7 for AC2, 7 for AC3	The maximum number of slots of contention window (CW) for access category <n> in EDCA access category. When the maximum number of slots for AC0is 1023, write as follows. dot11-edca-category-0-contention -window-max-slots = 1023
dot11-edca-category-<n> -frame-lifetime <n> indicates a number of access category	Interface	Integer	Infinite	Lifetime of a frame in a transmission queue of access category <n> in EDCA.
dot11-edca-category-<n> -downlink-txop-duration <n> indicates a number of access category	Interface	Integer	0ms	TXOP limitation in time of access category <n> in EDCA.
dot11-edca-category-<n> -max-non-fifo -aggregate-size-bytes <n> indicates a number of access category	Interface	Integer	Not Assigned	The maximum bytes of an aggregated frame for access category <n> in EDCA.
dot11-ap-scheduling-algorit	Interface	String	FIFO	Scheduling algorithm.

hm				(Available only if dot11-node-type = Access-Point) The following types are available. FIFO RoundRobin
dot11-max-aggregate-mpdu-size-bytes	Interface	Integer	0	Maximum A-MPDU size in bytes.
dot11-max-num-aggregate-subframes	Interface	Integer	64	Maximum number of subframes in an A-MPDU
dot11-protect-aggregate-frames-with-single-acked-frame	Interface	Bool	true	Activation of capability that an aggregated frame is transmitted after receiving an acknowledge frame solicited by an initial frame in a TXOP.
dot11-allow-frame-aggregation-with-txop-zero	Interface	Bool	false	Use of an aggregated frame when TXOP is zero.

Properties for PHY

Parameter name	Scope	Type	Default value	Description
dot11-phy-use-short-guard-interval-and-shrink-ofdm-symbol-duration	Global	Bool	false	Utilization of short guard interval.
dot11-phy-protocol	Interface	String	(Not Assigned)	Type of PHY protocol (IEEE802.11).
tx-power-specified-by	Interface	String	PhyLayer	Layer which orders transmission power level. Options are PhyLayer or UpperLayer.
dot11-tx-power-dbm	Interface	Real	(Not Assigned)	Transmission power level in dBm when PhyLayer is a commander.
dot11-default-tx-power-dbm-when-not-specified	Interface	Real	(Not Assigned)	Transmission power level in dBm when UpperLayer is a commander.
dot11-radio-noise-figure-db	Interface	Real	(Not	Noise figure in dB.

			Assigned)	
dot11-preamble-detection-power-threshold-dbm	Interface	Real	(Not Assigned)	Minimum power level in dBm to detect a preamble.
dot11-preamble-detection-probability-for-sinr-db-table	Interface	String	(Not Assigned)	Table of probabilities of preamble detection for given SINR. [SINR:probability] is one value and values are separated by a space. Example: -0.5:0.0 -3.0:0.5 -1.0:0.9 1.0:0.99
dot11-energy-detection-power-threshold-dbm	Interface	Real	(Not Assigned)	Energy detection threshold value in dBm when a preamble is not detected in carrier sense process (CCA-ED value) Default value may be 20 dB higher than the value of dot11-preamble-detection-power-threshold-dbm
dot11-signal-capture-ratio-threshold-db	Interface	Real	1000.0	Power difference in dB to receive a newer frame. A newer frame could be received when receive power level of the frame is stronger than power level of a frame already starting reception by this threshold value.
dot11-ofdm-symbol-duration	Interface	Time	(Not Assigned)	OFDM symbol length.
dot11-slot-time	Interface	Time	(Not Assigned)	Time duration of a slot. This value corresponds to aSlotTime.
dot11-sifs-time	Interface	Time	(Not Assigned)	Time duration of SIFS. This value corresponds to aSIFSTime.
dot11-rx-tx-turnaround-time	Interface	Time	(Not Assigned)	Transition time from reception mode to transmission mode.

Copyright© 2007-2016 Space-Time Engineering, LLC

				This value corresponds to aRxTxturnaroundTime.
dot11-phy-rx-start-delay	Interface	Time	(Not Assigned)	Time difference between timing of transmission request and transmission from an antenna. This value corresponds to aTxPHYDelay.
dot11-preamble-length-duration	Interface	Time	(Not Assigned)	Preamble length.
dot11-short-training-field-duration	Interface	Time	Same as dot11-preamble-length-duration	Short training field (STF) duration
dot11-plcp-header-length-duration	Interface	Time	(Not Assigned)	PLCP header length.
dot11-phy-high-throughput-header-additional-duration	Interface	Time	(Not Assigned)	Additional time duration of PHY header in HT mode
dot11-phy-high-throughput-header-additional-per-stream-duration	Interface	Time	(Not Assigned)	Additional time duration of PHY header per stream.
dot11-phy-artificial-frame-drop-probability-for-test	Interface	Real	0.0	Probability of artificial frame drop for test.
dot11-phy-artificial-subframe-drop-probability-for-test	Interface	Real	0.0	Probability of artificial subframe drop for test.

Properties for data rate / MCS (Modulation and Coding Scheme)

Parameter name	Scope	Type	Default value	Description
dot11-adaptive-rate-control-type	Interface	String	Static	Adaptive rate control type Static Arf
dot11-modulation-and-coding	Interface	String	(Not Assigned)	Default modulation and coding scheme (MCS).
dot11-modulation-and-coding-table	Interface	String	(Not Assigned)	MCS settings per link. Example: When a node uses BPSK with

Copyright© 2007-2016 Space-Time Engineering, LLC

				code rate 3/4, QPSK with code rate 3/4 and 16QAM with code rate 3/4 for receiver node #2, #3 to #5, and #7 and #9 respectively, write as follows. [Node number]-([Node number]):[Modulation name]_[Code rate in float number] 2:BPSK_0.75 3-5:QPSK_0.75 7,9:16QAM_0.75
dot11-modulation-and-coding-for-management-frames	Interface	String	(Not Assigned)	Modulation and coding scheme name used for transmission of management frames.
dot11-modulation-and-coding-for-broadcast	Interface	String	(Not Assigned)	Modulation and coding scheme name used for transmission of broadcast frames.
dot11-ack-datarate-selection-type	Interface	String	Lowest	Data rate of acknowledgement frame. Options are SameAsData, or Lowest.
dot11-ack-datarate -selection-table	Interface	String	(Not Assigned)	Modulation and coding scheme used for Ack. Usage of this parameter is [modulation and coding scheme for reception frame]:[modulation and coding scheme for Ack frame] Example: Delimiter is a space character. dot11-ack-datarate-selection -table = BPSK_0.5:BPSK_0.5 BPSK_0.75:BPSK_0.5 QPSK_0.5:BPSK_0.5 QPSK_0.75:BPSK_0.5 16QAM_0.5:BPSK_0.5

Copyright© 2007-2016 Space-Time Engineering, LLC

				16QAM_0.75:BPSK_0.5 64QAM_0.67:BPSK_0.5 64QAM_0.75:BPSK_0.5 64QAM_0.83:BPSK_0.5 256QAM_0.75:BPSK_0.5 256QAM_0.83:BPSK_0.5
dot11-ack-datarate-match-num-spatial-streams	Interface	String	(Not Assigned)	Determination to match the number of spatial streams of Ack frame with the one of reception frame.
dot11-modulation-and-coding-list	Interface	String	(Not Assigned)	Set of modulation and coding scheme name used for ARF and Minstrel-HT mode. Delimiter is a space character. Example: dot11-modulation-and-coding-list = BPSK_0.5 BPSK_0.75 QPSK_0.5 QPSK_0.75 16QAM_0.5 16QAM_0.75 64QAM_0.67 64QAM_0.75

Properties for ARF

Parameter name	Scope	Type	Default value	Description
dot11-arf-timer-duration	Interface	Time	(Not Assigned)	Timer to upgrade to a higher data rate. When no ACK reception occurs within this time duration, a higher data rate is used for next transmission.
dot11-arf-ack-in-success-count	Interface	Integer	(Not Assigned)	Threshold value of the number of success of ACK receptions to upgrade to a higher data rate. When the number of successful ACK reception in a row exceeds

Copyright© 2007-2016 Space-Time Engineering, LLC

				this threshold, a higher data rate is used for next transmission.
dot11-arf-ack-in-failure-count	Interface	Integer	(Not Assigned)	Threshold value of the number of failed ACK reception to downgrade to a lower data rate. When the number of failed ACK reception in a row exceeds this threshold, a lower data rate is used for next transmission.
dot11-arf-ack-in-failure-count-of-new-rate-state	Interface	Integer	(Not Assigned)	Threshold value of the number of failed ACK receptions to downgrade to a lower data rate just after upgrading to a higher data rate. When the number of failed ACK reception in a row exceeds this threshold just after upgrading to a higher data rate, a lower data rate is used for next transmission.

Properties for Minstrel HT

Parameter name	Scope	Type	Default value	Description
dot11-minstrel-ht-max-number-of-spatial-streams	Interface	Integer	4	Maximum number of streams
dot11-minstrel-ht-typical-transmission-unit-length-bytes	Interface	Integer	1200	Typical transmission frame length (Reference frame length to determine the maximum number of retransmissions for any data rate. Unit: byte)
dot11-minstrel-ht-max-retry-count	Interface	Integer	7	Maximum number of retransmissions for any transmission rate

dot11-minstrel-ht-min-retry-count	Interface	Integer	2	Minimum number of retransmissions for any transmission rate
dot11-minstrel-ht-sampling-retry-count	Interface	Integer	1	Number of sampling retransmissions
dot11-minstrel-ht-moving-average-exponentially-weight	Interface	Real	0.75	Exponentially weighted average of success rate of transmission
dot11-minstrel-ht-good-success-rate	Interface	Real	0.95	Threshold value to determine good success of transmission
dot11-minstrel-ht-bad-success-rate	Interface	Real	0.2	Threshold value to determine bad success of transmission
dot11-minstrel-ht-worst-success-rate	Interface	Real	0.1	Threshold value to determine worst success of transmission
dot11-minstrel-ht-max-success-rate-to-estimate-throughput	Interface	Real	0.9	Maximum success rate of transmission when throughput estimation is applied.
dot11-minstrel-ht-sampling-result-update-interval	Interface	Time	100ms	Minimum sampling result update interval
dot11-minstrel-ht-max-transmission-duration-for-a-multirate-retry-stage	Interface	Time	6ms	Maximum transmission duration for a data rate (one stage on multi-rate retry)
dot11-minstrel-ht-low-rate-sampling-trying-threshold	Interface	Integer	20	Sampling threshold for low rate
dot11-minstrel-ht-max-low-rate-sampling-count	Interface	Integer	2	Maximum sampling count for low rate
dot11-minstrel-ht-enough-number-of-transmitted-mpdus-to-sample	Interface	Integer	30	Required number of MPDU to judge for downgrade
dot11-minstrel-ht-sampling-transmission-interval-count	Interface	Integer	16	Sampling interval (Sample per N transmissions) (Unit: count)
dot11-minstrel-ht-initial-sampling-transmission-count	Interface	Integer	4	Sampling interval for initial state

Properties for association

Parameter name	Scope	Type	Default value	Description
dot11-access-point-ssid	Interface	String	(Not assigned)	SSID of access point. （available for AP） When no value is assigned, any STA can communicate with this AP.
dot11-beacon-transmit-interval	Interface	Time	100ms	Beacon interval. (available for AP)
dot11-access-point-auth-processing-delay	Interface	Time	100ms	Process delay of authentication. (available for AP)
dot11-mobile-sta-ssid	Interface	String	(Not assigned)	Target SSID to communicate with. （available for STA） （Any AP can be targeted when no value is assigned.）
dot11-channel-scan-interval	Interface	Time	500ms	Scan time assigned for each channel. (available for STA)
dot11-channel-scan-start-time-max-jitter	Interface	Time	Designated value by dot11-channel-scan-interval	The maximum time jitter of starting scan. （available for STA）
dot11-association-threshold-rssi-dbm	Interface	Real	Designated value by dot11-preamble-detection-power-threshold-dbm	The minimum reception signal power level for association process with a target AP. (available for STA)
dot11-associate-failure-timeout-interval	Interface	Time	1s	Timeout value to determine that an association process fails. (available for STA)
dot11-disassociation-threshold-rssi-dbm	Interface	Real	Designated value by	Threshold of reception power level to start disassociation.

Copyright© 2007-2016 Space-Time Engineering, LLC

			dot11-preamble-detection-power-threshold-dbm —3dBm	(available for STA)
dot11-link-status-check-interval	Interface	Time	dot11-channel-scan-interva ×the number of channels to search	Time interval to check association status. (available for STA)
dot11-beacon-rssi-moving-average-coefficient	Interface	Real	0.5	Coefficient to average received beacon signal power. (available for STA)
dot11-authentication-timeout-interval	Interface	Time	1s	Timeout value to determine that an authentication process fails. (available for STA)
dot11-initial-channel-number	Interface	Integer	0	Initial channel number.

3.1.2. Properties for Dot Eleven Advanced

Properties for TGn channel model

Parameter name	Scope	Type	Default value	Description
tgn-mimo-channel-model-letter	Global	String	(Not assigned)	Channel model defined in IEEE TGn: Channel model B/C/D/E
tgn-mimo-channel-number-antennas	Global	Integer	(Not assigned)	Number of antennas used for MIMO per one interface
tgn-mimo-channel-normalized-antenna-spacing	Global	Real	(Not assigned)	Normalized distance between two antennas by wave length
tgn-mimo-channel-scatterer-movement	Global	Real	0.333333333	Speed of scatterer Unit : m/s

Copyright© 2007-2016 Space-Time Engineering, LLC

-meters-sec				
tgn-mimo-channel -sampling-interval-time	Global	Time	Determined by channel frequency and speed of scatterer	Sampling period

Properties for 11ad

Parameter name	Scope	Type	Default value	Description
dot11ad-bit-error-rate -curve-file	Global	String	(Not assigned)	File name of bit error rate curve for 11ad's performance
dot11ad-forced -ap-pcp-nodeid	Interface	Integer	(Not assigned)	Node ID of connected PCP
dot11ad-number -directional-sectors	Interface	Integer	(Not assigned)	Total number of sectors used for beamforming
dot11ad-forced -beamforming-sector-list	Interface	String	(Not assigned)	Sector number when it is necessary to fix used sector number [Node ID]:[target sector number]
dot11ad-custom-sectored -antenna-model-sector -azimuths-degs	Interface	String	(Not assigned)	Azimuths angle of an antenna for a sector Angle zero is face of a node and definition of angle direction is clockwise.
dot11ad-custom-sectored -antenna-model-sector -pattern-names	Interface	String	(Not assigned)	Antenna model for a sector
dot11ad-use-custom-quasi-omni-antenna-model	Interface	String	false	Indication to utilize custom antenna model for quasi omni mode
dot11ad-custom-quasi-omni-antenna-model-name	Interface	String	(Not assigned)	Custom antenna model name for quansi omni mode
dot11ad-beamforming	Interface	String	RSSI	Selection procedure of a sector

-sector-selector-scheme-name				in case of beamforming RSSI or SINR
dot11ad-abft-max-num-responder-txss-frames	Interface	Inge	(Not assigned)	Maximum number of sectors of transmission nodes with sector sweeping
dot11ad-ap-receive-sector-sweep-interval-to-beacon	Interface	Integer	0	Beacon interval with RXSS of AP Zero indicates no execution of RXSS
dot11ad-sta-receive-sector-sweep-interval-to-beacon	Interface	Integer	0	Beacon interval with RXSS of STA Zero indicates no execution of RXSS
dot11ad-beacon-superframe-interval-duration	Interface	Time	(Not assigned)	Beacon interval
dot11ad-beacon-transmission-interval-duration	Interface	Time	(Not assigned)	Length of BTI
dot11ad-association-beamforming-training-aka-abft-duration	Interface	Time	(Not assigned)	Length of ABFT
dot11ad-data-transfer-interval-<No>-relative-start-time	Interface	Time	(Not assigned)	Starting time of DTI-<No>
dot11ad-data-transfer-interval-<No>-duration	Interface	Time	(Not assigned)	Length of DTI-<No>
dot11ad-data-transfer-interval-<No>-is-contention-based-access-period	Interface	Bool	(Not assigned)	Indication of contention based access for DTI-<No>
dot11ad-data-transfer-interval-<No>-source-nodeid	Interface	Integer	(Not assigned)	Transmission node ID when DTI-<No> does not use contention based access
Dot11ad-data-transfer-interval-<No>-destination-nodeid	Interface	Integer	(Not assigned)	Reception node ID when DTI-<No> does not use contention based access

Copyright© 2007-2016 Space-Time Engineering, LLC

dot11ad-short-beamforming-interframe-space-duration	Interface	Time	(Not assigned)	Length of SBIFS
dot11ad-mbps-datarate-for-dot11ad-mcs<n>	Interface	Real	(Not assigned)	MCS<n> data rate in Mbps <n> indicates a MCS index [0,…,24]

Properties for 11ah

Parameter name	Scope	Type	Default value	Description
dot11ah-bit-error-rate-curve-file	Global	String	(Not assigned)	File name of bit error rate curve for 11ah's performance
dot11ah-association-id-table-file	Global	String	(Not assigned)	File name which includes mapping information between node ID and AID (*.aid)
dot11ah-canned-restricted-access-windows-file	Global	String	(Not assigned)	File name which includes information of schedule of RAW (*.raw)
dot11ah-use-optimized-ndp-control-frames	Global	Bool	(Not assigned)	Indication to use NDP Ack frame instead of regular Ack frame
dot11ah-is-a-restricted-access-window-sta	Interface	Bool	(Not assigned)	Indication to restrict that STA accesses channel only in RAW period

Copyright© 2007-2016 Space-Time Engineering, LLC

4. Statistics and trace

4.1. List of predefined statistics

Layer	Model name	Statistic name	Description
MAC	Dot11Mac	BytesSent	Total bytes of transmitted frames
		BytesReceived	Total bytes of received frames
		Data_UnicastFramesSent	Number of unicast data frames transmitted
		Data_UnicastFramesResent	Number of unicast data frames retransmitted
		Data_BroadcastFramesSent	Number of broadcast data frames transmitted
		Data_AggregateFramesSent	Number of aggregated data frames transmitted
		Data_AggregateFramesResent	Number of aggregated data frames retransmitted
		Data_FramesReceived	Number of data frames received
		Data_AggregatedSubframesReceived	Number of aggregated sub-frames received
		Data_DuplicatedFramesReceived	Number of duplicated data frames received
		ACK_FramesSent	Number of ACK frames transmitted
		ACK_FramesReceived	Number of ACK frames received
		BlockACK_FramesSent	Number of block ACK frames transmitted
		BlockACK_FramesReceived	Number of block ACK frames received
		BAR_FramesSent	Number of BlockAckRequest

Copyright© 2007-2016 Space-Time Engineering, LLC

			transmitted
		BAR_FramesReceived	Number of BlockAckRequest received
		RTS_FramesSent	Number of RTS frames transmitted
		RTS_FramesReceived	Number of RTS frames received
		CTS_FramesSent	Number of CTS frames transmitted
		CTS_FramesReceived	Number of CTS frames received
		Beacon_FramesSent	Number of beacon frames transmitted
		Beacon_FramesReceived	Number of beacon frames received
		AssociationRequest_FramesSent	Number of association request frames transmitted
		AssociationRequest_FramesReceived	Number of association request frames received
		AssociationResponse_FramesSent	Number of association response frames transmitted
		AssociationResponse_FramesReceived	Number of association response frames received
		ReassociationRequest_FramesSent	Number of reassociation request frames transmitted
		ReassociationRequest_FramesReceived	Number of reassociation request frames received
		ReassociationResponse_FramesSent	Number of reassociation response

			frames transmitted
		ReassociationResponse_FramesReceived	Number of reassociation response frames received
		Disassociation_FramesSent	Number of disassociation frames transmitted
		Disassociation_FramesReceived	Number of disassociation frames received
		Authentication_FramesSent	Number of authentication frames transmitted
		Authentication_FramesReceived	Number of authentication frames received
		FramesDropped	Number of frames discarded due to retransmission limit
MAC	Dot11adMac	DmgBeacon_FramesSent	Number of DMG beacons transmitted
		DmgBeacon_FramesReceived	Number of DMG beacons received
		SSW_FramesSent	Number of SSW (Sector Sweep) frames transmitted
		SSW_FramesReceived	Number of SSW (Sector Sweep) frames received
		SswFeedback_FramesSent	Number of SSW (Sector Sweep) feedback frames transmitted
		SswFeedback_FramesReceived	Number of SSW (Sector Sweep) feedback frames

			received
		SswAck_FramesSent	Number of SSW (Sector Sweep) acks transmitted
		SswAck_FramesReceived	Number of SSW (Sector Sweep) acks received
PHY	Dot11Phy	FramesTransmitted	Number of frames transmitted
		FramesReceived	Number of frames received
		FramesWithErrors	Number of frames with errors
		SignalsCaptured	Number of detected errors due to capture effect (switching to another signal with stronger power)
		InterferingSignals	Number of interferences
		SignalsDuringTransmission	Number of detected errors because frames are arrived during transmission process.
		TooWeakToReceiveSignals	Number of detected errors due to lower power level
		ReceivedFrameRssiDbm	RSSI (Received Signal Strength Indication) of received frames
		ReceivedFrameSinrDb	SINR (Signal to Interference plus Noise Ratio) of received frames at the end of receiving

Note:
ACK_FramesSent and ACK_FramesReceived include all ACK frames responded to frames not only data frames but also management frames.

Statistics of PHY layer have the following relation.

At the start of signal reception process:

If a signal fails to start reception, InterferingSignal is incremented.

In addition, if the signal is too weak to receive, TooWeakToReceiveSignals is incremented.

In addition, if the signal arrives during transimittion, SignalsDuringTransmission is incremented.

At end of signal reception process:

If a signal is successfully received, FrameReceived is incremented.

If a signal fails to be received due to capture effect, SignalsCaptured is incremented.

If a signal fails to be received due to other reasons, FramesWithErrors is incremented.

Relation among FramesTransmitted for transmitter, and InterferingSignals, FramesReceived, FramesWithErrors for receiver can be written in the below equation

FramesTransmitted = InterferingSignals + FramesReceived + FramesWithErrors

A captured signal is firstly treated as a desired signal, and SignalsCaptured is incremented by 1 when capture effect happens. At the same time, InterferingSignals is incremented by 1 as interference.

4.2. List of predefined trace events

Layer	Model name	Trace event	Additional information	Description
MAC	Dot11Mac Tag name： Mac	RxFrame	Packet ID, frame type, the number of received bytes	Reception of a frame
		ClearCh	-	Channel clear
		BusyCh	-	Channel busy
		NAV-Start	Protection duration of wireless medium	NAV timer start
		NAV-End	-	NAV timer end
		IFSAndBackoff-Start	Access category, backoff duration, indication of corruption of the latest received frame	Backoff procedure start
		IFSAndBackoff-Pause	Access category, remaining backoff duration	Backoff timer pause
		IFSAndBackoff-End	-	Backoff timer end
		Dequeue	Access category, packet ID	Dequeue
		Tx-RTS	Access category, the number of retransmissions	RTS transmission
		Tx-CTS	-	CTS transmission
		Tx-ACK	-	ACK transmission
		Tx-BlockACK	-	Block ACKtransmission
		Tx-BlockACK-Request	-	Block ACK Request transmission
		Tx-DATA-B	Packet ID, access category	Transmission of (broadcasted)

Copyright© 2007-2016 Space-Time Engineering, LLC

				data frame
		Tx-DATA-U	Packet ID, access category, the number of retransmissions	Transmission of (unicasted) data frame
		Tx-DATA-A	Packet ID, access category, the number of retransmissions, destination node ID, the number of subframes	Transmission of (aggregated) data frame
		Timeout	access category, the number of slots of a window, the number of retransmissions	Timeout
		Drop	Packet ID	Discard of a frame
		Tx-Management	Packet ID, frame type	Transmission of management frame
		TxRateUpdate	data rate (bps), destination node ID	Transmission data rate update
		Traffic(packets/sec)	-	Traffic (packet per second) (Available in Scenargie Visual Lab Trace Visualization Settings)
		Traffic(bits/sec)	-	Traffic (bit per second) (Available in Scenargie Visual

Copyright© 2007-2016 Space-Time Engineering, LLC

				Lab Trace Visualization Settings)
MAC	Dot11adMac	Tx-DMG	Packet ID, destination node ID, frame type	Transmission of a DMG frame
PHY	Dot11Phy Tag name：Phy	TxStart	Packet ID, transmitted signal power, data rate, transmission duration	Transmission of a signal
		RxStart	Packet ID, power level of received signal	Detection of a preamble
		RxEnd	Packet ID, indicator of packet error	Completion of a packet reception
	Dot11Phy Tag name：PhyInterference	NoiseStart	Transmitter node ID, power level of received signal, power level of interference, packet ID	Detection of interference
		NoiseEnd	Power level of received signal, power level of interference signal, packet ID	Disappearance of interference

5. Model description

5.1. Modulation and coding scheme and data rate

Data rate for each modulation and coding scheme (non-HT mode)

	10MHz channel bandwidth	20MHz channel bandwidth
BPSK 1/2	3Mbps	6Mbps
BPSK 3/4	4.5Mbps	9Mbps
QPSK 1/2	6Mbps	12Mbps
QPSK 3/4	9Mbps	18Mbps
16QAM 1/2	12Mbps	24Mbps
16QAM 3/4	18Mbps	36Mpbs
64QAM 2/3	24Mbps	48Mpbs
64QAM 3/4	27Mbps	54Mbps

Note) Dot11p which is defined in Visual Lab uses 10 MHz channel bandwidth, and Dot11a and Dot11g use 20 MHz channel bandwidth.

Data rate for each modulation and coding scheme (HT mode, number of streams: 1, GI 800ns)

	10MHz	20MHz	80MHz	16MHz
BPSK 1/2	6.5Mbps	13.5Mbps	29.3Mbps	58.5Mbps
QPSK 1/2	13Mbps	27Mbps	58.5Mbps	117Mbps
QPSK 3/4	19.5Mbps	40.5Mbps	87.8Mbps	175.5Mbps
16QAM 1/2	26Mbps	54Mbps	117Mbps	234Mbps
16QAM 3/4	39Mbps	81Mbps	175.5Mbps	351Mbps
64QAM 2/3	52Mbps	108Mbps	234Mbps	468Mbps
64QAM 3/4	58.5Mbps	121.5Mbps	263.3Mbps	526.5Mbps
64QAM 5/6	65Mbps	135Mbps	292.5Mbps	585Mbps
256QAM 3/4	78Mbps	162Mbps	351Mbps	702Mbps
256QAM 5/6	-	180Mbps	390Mbps	780Mbps

5.2. APIs in Mac layer

Source file: dot11_mac.h/cpp

5.2.1. Dot11Mac

MAC layer model used for Dot11

Return value	Function (arguments)	Description
void	**SetCustomAdaptiveRateController** (const shared_ptr< Dot11::AdaptiveRateController > &rateControllerPtr)	Sets a customized adaptive rate controller.
shared_ptr < Dot11::AdaptiveRateController >	**GetAdaptiveRateControllerPtr** () const	Gets a pointer of adaptive rate controller.
shared_ptr < MacAndPhy InfoInterface >	**GetMacAndPhyInfoInterface** () const	Gets a pointer of MAC / PHY information interface.
virtual void	**NetworkLayerQueueChange Notification** ()	Notifies transmission queue change.
virtual void	**DisconnectFromOtherLayers** ()	Disconnects holding pointers from other layers.
virtual GenericMac AddressType	**GetGenericMacAddress** () const	Gets own MAC address. (GenericMacAddressType)
MacAddressType	**GetMacAddress** () const	Gets own of MAC address .(MacAddressType)
Dot11Mac OperationMode	**GetOperationMode** () const	Gets operation mode. (ad-hoc, AP, STA)
bool	**GetIpMulticastAddressToMacAddressMappingIsEnabled** () const	Checks that mapping between multicast address and MAC address is enabled.
void	**SendManagementFrame** (unique_ptr< Packet > &framePtr)	Sends a management frame.

Copyright© 2007-2016 Space-Time Engineering, LLC

bool	**IsAHighThroughputStation** () const	Checks that this node is HT mode.
unsigned int	**GetNumberOfChannels** () const	Gets total number of channels.
unsigned int	**GetCurrentChannelId** () const	Gets current channel ID.
unsigned int	**GetMaxBandwidthNumChannels** () const	Gets maximum bandwidth of bonded channels.
const vector< unsigned int > &	**GetCurrentBondedChannelList** ()	Gets current bonding .channel list
double	**GetRssiOfLastFrameDbm** () const	Gets RSSI (dBm) of the last received frame.
void	**SendAssociationRequest** (const MacAddressType &apAddress)	Sends an association request frame.
void	**SendReassociationRequest** (const MacAddressType &apAddress, const MacAddressType &currentApAddress)	Sends a reassociation request frame.
void	**SendAssociationResponse** (const MacAddressType &staAddress)	Sends an association response frame.
void	**SendReassociationResponse** (const MacAddressType &staAddress)	Sends a reassociation response frame.
void	**SendDisassociation** (const MacAddressType &receiverAddress)	Sends a disassociation frame.
void	**StopReceivingFrames** ()	Stops frame reception.
void	**StartReceivingFrames** ()	Starts frame reception.
bool	**IsNotReceivingFrames** () const	Checks that this network interface is not receiving any frame.
void	**SendAuthentication** (const MacAddressType &receiverAddress)	Sends an authentication frame.
void	**SendPowerSaveNullFrame** (const MacAddressType &receiverAddress, const bool goingToPowerManagementMode)	Sends a Null frame.
void	**SwitchToChannel** (const unsigned int	Switches to a designated channel.

	&channel)	(for single channel operation)
void	**SwitchToChannels** (const vector< unsigned int > &channels)	Switches to designated channels.
void	**ResetOutgoingLinksTo** (const MacAddressType &macAddress)	Resets outgoing link information for the specified MAC address.
void	**RequeueBufferedPackets** ()	Re-queues buffered packets.
void	**RequeueBufferedPacket** (unique_ptr< Packet > &packetPtr, const NetworkAddress &nextHopAddress, const PacketPriorityType priority, const EtherTypeFieldType etherType, const TimeType &timestamp, const unsigned int retryTxCount)	Re-queues a designated buffered packet.
void	**RequeueManagementFrame** (unique_ptr< Packet > &framePtr)	Re-queues a designated management frame.
void	**SendLinkIsUpNotificationToNetworkLayer** ()	Notifies accomplishment of a link to network layer. (STA)
void	**SendLinkIsDownNotificationToNetworkLayer** ()	Notifies disconnection of a link to network layer. (STA)
void	**SendNewLinkToANodeNotificationToNetworkLayer** (const MacAddressType &macAddress)	Notifies accomplishment of a link to network layer（AP）
void	**LookupMacAddressForNeighbor** (const NodeIdType nodeId, bool &wasFound, MacAddressType &macAddress)	Looks up a MAC address of a designated neighbor node.
bool	**MpduFrameAggregationIsEnabled** () const	Checks that A-MPDU is enabled.
bool	**MsduFrameAggregationIsEnabled** () const	Checks that A-MSDU is enabled.
void	**SetMpduFrameAggregationIsEnabledFor** (const MacAddressType &destinationAddress)	Sets that A-MPDU is enabled for a designated address.

void	**SetMsduFrameAggregationIsEnabledFor** (const MacAddressType &destinationAddress)	Enables A-MSDU to a designated address.
static shared_ptr < Dot11Mac >	**Create** (const ParameterDatabaseReader &theParameterDatabaseReader, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const shared_ptr< SimplePropagationModelForNode< PropFrameType > > &propModelInterfacePtr, const shared_ptr< MimoChannelModelInterface > &mimoChannelModelInterfacePtr, const shared_ptr< BitOrBlockErrorRateCurveDatabase > &berCurveDatabasePtr, const NodeIdType &nodeId, const InterfaceIdType &interfaceId, const unsigned int interfaceIndex, const shared_ptr< NetworkLayer > &networkLayerPtr, const RandomNumberGeneratorSeedType &nodeSeed)	Instantiates Dot11Mac class. (used for MIMO channel)
static shared_ptr < Dot11Mac >	**Create** (const ParameterDatabaseReader &theParameterDatabaseReader, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const shared_ptr< SimplePropagationModelForNode< PropFrameType > > &propModelInterfacePtr, const shared_ptr<	Instantiates Dot11Mac class.

	BitOrBlockErrorRateCurveDatabase > &berCurveDatabasePtr, const NodeIdType &nodeId, const InterfaceIdType &interfaceId, const unsigned int interfaceIndex, const shared_ptr< NetworkLayer > &networkLayerPtr, const RandomNumberGeneratorSeedType &nodeSeed)	

5.2.2. SimpleMacAddressResolver

MAC address resolver model

Return value	Function (arguments)	Description
	SimpleMacAddressResolver (Dot11Mac *initMacPtr)	Instantiates SimpleMacAddressResolver class.
void	**GetMacAddress** (const NetworkAddress &aNetworkAddress, const NetworkAddress &networkAddressMask, bool &wasFound, MacAddressType &resolvedMacAddress))	Gets MAC address from a designated network address.
void	**GetNetworkAddressIfAvailable** (const MacAddressType &macAddress, const NetworkAddress &subnetNetworkAddress, bool &wasFound, NetworkAddress &resolvedNetworkAddress)	Gets network address from a designated MAC address.

Source file: dot11_mac_ap.h/cpp

5.2.3. Dot11ApManagementController

Management model used for AP mode

Return value	Function (arguments)	Description
	Dot11ApManagementController (Dot11Mac *initMacLayerPtr, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &initNodeId, const InterfaceOrInstanceIdType &initInterfaceId, const RandomNumberGeneratorSeedType &interfaceSeed)	Instantiates Dot11ApManagementController class.
void	**ProcessManagementFrame** (const Packet &managementFrame)	Processes a management frame.
void	**ReceiveFramePowerManagementBit** (const MacAddressType &sourceAddress, const bool framePowerManagementBitIsOn)	Receives power management bit.
bool	**IsAnAssociatedStaAddress** (const MacAddressType &theMacAddress) const	Checks that a designated STA address is associated.
void	**LookupAssociatedNodeMacAddress** (const NodeIdType &nodeId, bool &wasFound, MacAddressType &macAddress) const	Looks up the MAC address for an associated node.
void	**GetAssociatedStaAddressList** (vector< MacAddressType > &associatedStaAddressList) const	Gets associated MAC address list.

Copyright© 2007-2016 Space-Time Engineering, LLC

bool	**StationIsAsleep** (const MacAddressType &staAddress) const	Checks that a designated STA is in sleep mode.
void	**BufferPacketForSleepingStation** (const MacAddressType &staAddress, unique_ptr< Packet > &packetPtr, const NetworkAddress &destinationNetworkAddress, const PacketPriorityType &priority, const EtherTypeFieldType etherType, const TimeType &timestamp)	Buffers a packet for a designated STA in sleep mode
void	**BufferManagementFrameForSleepingStation** (const MacAddressType &staAddress, unique_ptr< Packet > &framePtr, const TimeType &timestamp)	Buffers a management frame for a STA in sleep mode.
void	**GetPowerSaveBufferedPacket** (const MacAddressType &staAddress, bool &wasRetrieved, unique_ptr< Packet > &packetToSendPtr, unsigned int &retryTxCount, PacketPriorityType &priority, EtherTypeFieldType &etherType)	Gets packets buffered in sleep mode.

Source file: dot11_mac_sta.h/cpp

5.2.4. Dot11StaManagementController

Management controller model used for STA mode

Return value	Function (arguments)	Description

	Dot11StaManagementController (Dot11Mac *initMacLayerPtr, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &nodeId, const InterfaceOrInstanceIdType &interfaceId, const RandomNumberGeneratorSeedType &interfaceSeed))	Instantiates Dot11StaManagementController class.
void	**SetChannelScanningController** (const shared_ptr< AbstractChannelScanningController > &scanningControllerPtr)	Sets a channel scan controller.
void	**ProcessManagementFrame** (const Packet &managementFrame)	Processes a management frame.
void	**GetCurrentAccessPointAddress** (bool &hasAnAccessPoint, MacAddressType &currentAccessPointAddress) const	Gets the Mac address of associated AP.
void	**SwitchToAccessPoint** (const MacAddressType &accessPointAddress)	Switches a connection to a designated AP.

Source file: dot11_ratecontrol.h

5.2.5. AdaptiveRateController

Class of adaptive rate controller

Return value	Function (arguments)	Description

virtual bool	**GetHighThroughputModeIsOn** () const =0	Checks that High Throughput mode is activated. (Pure virtual function)
virtual unsigned int	**GetBaseChannelBandwidthMhz** () const =0	Gets base channel bandwidth in MHz. （Pure virtual function）
virtual void	**SetMaxChannelBandwidthMhz** (const unsigned int newMaxChannelBandwidthMhz)=0	Sets maximum channel bandwidth in MHz. (Pure virtual function)
virtual unsigned int	**GetMaxChannelBandwidthMhz** () const =0	Gets the maximum channel bandwidth in MHz. （Pure virtual function）
virtual ModulationAndCodingSchemesType	**GetLowestModulationAndCoding** () const =0	Gets the lowest MCS name. （Pure virtual function）
virtual void	**AddNewStation** (const MacAddressType &macAddress, const unsigned int stationBandwidthNumChannels, const bool isHighThroughputStation)=0	Adds destination node. (Pure virtual function)
virtual void	**GetDataRateInfoForDataFrameToStation** (const MacAddressType &macAddress, TransmissionParametersType &txParameters) const =0	Gets MCS information used for data frame to a designated Mac address. （Pure virtual function）
virtual void	**GetDataRateInfoForAckFrameToStation** (const MacAddressType &macAddress, const TransmissionParametersType &receivedFrameTxParameters, TransmissionParametersType &ackTxParameters) const =0	Gets MCS information used for ACK frame to a designated Mac address. (Pure virtual function)

Copyright© 2007-2016 Space-Time Engineering, LLC

virtual void	**GetDataRateInfoForAckFrameFromStation** (const MacAddressType &macAddress, const TransmissionParametersType &sentFrameTxParameters, TransmissionParametersType &ackTxParameters) const =0	Gets MCS information used for ACK frame from a designated Mac address. (Pure virtual function)
virtual void	**GetDataRateInfoForManagementFrameToStation** (const MacAddressType &macAddress, TransmissionParametersType &txParameters) const =0	Gets MCS information used for management frame to a designated Mac Address. （Pure virtual function）
virtual void	**GetDataRateInfoForBeaconFrame** (TransmissionParametersType &txParameters) const =0	Gets MCS information used for beacon. （Pure virtual function）
virtual void	**NotifyAckReceived** (const MacAddressType &macAddress)	Notifies that ACK reception has succeeded.
virtual void	**NotifyAckFailed** (const MacAddressType &macAddress)	Notifies that ACK reception has failed.
virtual void	**ReceiveIncomingFrameSinrValue (const MacAddressType &sourceMacAddress, const double &measuredSinrValue)**	受信フレームの SINR の取得
virtual void	**NotifyStartingAFrameTransmissionSequence** (const MacAddressType &macAddress, const bool isTransmittingFrameLongFrame, const size_t numberOfTransmittingMpdus, const shared_ptr< RetryCountManager >	Notifies that a frame transmission process starts.

	&retryCountManagerPtr, const unsigned int retryTxCount=0)	
virtual void	**NotifyFinishingAFrameTransmissionSequence** (const MacAddressType &macAddress, const size_t numberOfTransmissionSucceededMpdus, const shared_ptr< RetryCountManager > &retryCountManagerPtr)	Notifies that a frame transmission process finishes.
virtual void	**IsMultirateRetryModel** () const	Checks that multi-rate retry model is activated.

Source file: dot11_txpowercontrol.h

5.2.6. AdaptiveTxPowerController

Base class of a model used for adaptive transmission power level controller

Return value	Function (arguments)	Description
	AdaptiveTxPowerController (const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &nodeId, const InterfaceIdType &interfaceId)	Instantiates AdaptiveTxPowerController class.
virtual double	**CurrentTransmitPowerDbm** (const MacAddressType &macAddress) const	Gets transmission power level used for a designated MAC address.
bool	**TxPowerIsSpecifiedByPhyLayer** () const	Checks that PHY layer controls transmission power level.

5.3. API for PHY

Source file: dot11_phy.h/cpp

5.3.1. Dot11Phy

PHY layer model for Dot11

Return value	Function (arguments)	Description
	Dot11Phy (const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &nodeId, const InterfaceIdType &interfaceId, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const shared_ptr< SimplePropagationModelForNode< PropFrameType > > &propModelInterfacePtr, const shared_ptr< BitOrBlockErrorRateCurveDatabase > &berCurveDatabasePtr, const shared_ptr< Dot11MacInterfaceForPhy > macLayerPtr, const RandomNumberGeneratorSeedType &nodeSeed)	Instantiates Dot11Phy class.
shared_ptr < Dot11InfoInterface >	**GetDot11InfoInterface** () const	Gets a MAC and PHY information interface.
bool	**IsReceivingAFrame** () const	Checks that this network interface is receiving a frame.
bool	**IsTransmittingAFrame** () const	Checks that this network interface is

		transmitting a frame.
bool	**ChannelIsClear** () const	Checks that the channel is clear.
void	**TransmitFrame** (unique_ptr< Packet > &packetPtr, const TransmissionParametersType &txParameters, const double &transmitPowerDbm, const TimeType &delayUntilAirborne)	Transmits a frame.
void	**TransmitAggregateFrame** (unique_ptr< vector< unique_ptr< Packet > > > &aggregatedFramePtr, const TransmissionParametersType &txParameters, const double &transmitPowerDbm, const TimeType &delayUntilAirborne)	Transmits an aggregated frame.
void	**TakeOwnershipOfLastTransmittedFrame** (unique_ptr< Packet > &framePtr)	Gets ownership of the last frame. (for retransmission)
bool	**LastSentFrameWasAggregate** () const	Checks that the last frame has been aggregated.
void	**TakeOwnershipOfLastTransmittedAggregateFrame** (unique_ptr< vector< unique_ptr< Packet > > > &aggregateFramePtr)	Gets ownership of the last aggregated frame. (for retransmission)
TimeType	**CalculatePhysicalLayerHeaderDuration** (const TransmissionParametersType &txParameters) const	Calculates transmission time used for PHY header
TimeType	**CalculateFrameDataDuration** (const unsigned int frameLengthBytes, const TransmissionParametersType &txParameters) const	Calculates transmission time used for data frame.
TimeType	**CalculateFrameTransmitDuration** (const unsigned int	Calculates total transmission time of a frame.

Copyright© 2007-2016 Space-Time Engineering, LLC

	frameLengthBytes, const TransmissionParametersType &txParameters) const	
TimeType	**CalculateAggregateFrameTransmitDuration** (const vector< unique_ptr< ScenSim::Packet > > &aggregateFrame, const TransmissionParametersType &txParameters) const	Calculates total transmission time of an aggregated frame.
TimeType	**GetSlotDuration** () const	Gets slot duration.
TimeType	**GetShortInterframeSpaceDuration** () const	Gets SIFS duration.
TimeType	**GetRxTxTurnaroundTime** () const	Gets switching time between Rx and Tx modes.
TimeType	**GetPhyRxStartDelay** () const	Gets delay to start reception process.
unsigned int	**GetBaseChannelBandwidth** () const	Gets the base channel bandwidth.
unsigned int	**GetChannelCount** () const	Gets total number of channels.
unsigned int	**GetCurrentChannelNumber** () const	Gets current channel number.
unsigned int	**GetCurrentBandwidthNumChannels** () const	Gets current bandwidth of bonded channels.
unsigned int	**GetMaxChannelBandwidthMhz** () const	Gets maximum channel bandwidth in MHz.
unsigned int	**GetMaxBandwidthNumChannels** () const	Gets maximum number of bonded channels.
bool	**GetIsAHighThroughputStation** () const	Checks that high throughput mode is activated.
const vector < unsigned int > &	**GetCurrentBondedChannelList** () const	Gets current bonded channel list.
Void	**SwitchToChannels** (const vector< unsigned int > &bondedChannelList)	Switches to designated channels. (for channel bonding)
Void	**SwitchToChannelNumber** (	Switches to a designated channel.

Copyright© 2007-2016 Space-Time Engineering, LLC

	const unsigned int channelNumber)	
double	**GetRssiOfLastFrameDbm** () const	Gets power level in dBm for the last received frame.
double	**GetSinrOfLastFrameDb** () const	Gets SINR in dB for the latest received frame.
void	**StopReceivingFrames** ()	Stops frame reception process.
void	**StartReceivingFrames** ()	Starts frame reception process.
Bool	**IsNotReceivingFrames** () const	Checks that this network interface is not receiving any frame.
const ScenSim::ObjectMobilityPosition	**GetPosition** ()	Gets an antenna position.
unsigned int	**GetNumberOfReceivedFrames** () const	Gets the number of received frames.
unsigned int	**GetNumberOfFramesWithErrors** () const	Gets the number of frames with errors. (Cases under capture effects are not included.)
unsigned int	**GetNumberOfSignalCaptures** () const	Gets the number of frame with error due to frame capture effect.
TimeType	**GetTotalIdleChannelTime** () const	Gets total time duration in channel idle status.
TimeType	**GetTotalBusyChannelTime** () const	Gets total time duration in channel busy status.
TimeType	**GetTotalTransmissionTime** () const	Gets total time duration used for transmission.

5.3.2. Dot11MacInterfaceForPhy

Base class of MAC Interface for Dot11 PHY

Return value	Function (arguments)	Description
virtual void	**BusyChannelAtPhysicalLayerNotification** ()	Notifies that channel is busy. (pure virtual function)
virtual void	**ClearChannelAtPhysicalLayer**	Notifies that channel is clear.

	Notification ()	(pure virtual function)
virtual void	**TransmissionIsCompleteNotification** ()	Notifies that a frame transmission has completed. (pure virtual function)
virtual void	**DoSuccessfulTransmissionPostProcessing** (const bool wasJustTransmitting)	Processes after completing a frame transmission.(pure virtual function)
virtual void	**ReceiveFrameFromPhy** (const Packet &aFrame, const TransmissionParametersType &receivedFrameTxParameters)	Receives a frame from PHY layer. (pure virtual function)
virtual void	**ReceiveAggregatedSubframeFromPhy** ( ) unique_ptr< Packet > &subframePtr, const TransmissionParametersType &receivedFrameTxParameters, const unsigned int aggregateFrameSubframeIndex, const unsigned int numberSubframes)=0	Receives A-MPDU sub-frame from PHY layer. (pure virtual function)
virtual void	**ReceiveMsduAggregateFrameFromPhy** (const vector< unique_ptr< ScenSim::Packet > > &msduAggregateFrame, const TransmissionParametersType &receivedFrameTxParameters)=0	Receives A-MSDU frame from PHY layer. (pure virtual function)
virtual void	**NotifyThatPhyReceivedCorrupted**Frame ()=0	Notifies that the received frame has been corrupted. (pure virtual function)
virtual void	**NotifyThatPhyReceivedCorrupted**AggregatedSubframe (const TransmissionParametersType &receivedFrameTxParameters, const unsigned int aggregateFrameSubframeIndex,	Notifies that the received A-MPDU sub-frame has been corrupted. (pure virtual function)

	const unsigned int numberSubframes)=0	
virtual Bool	**AggregatedSubframeIsForThisNode** (const Packet &frame) const =0	Checks that a received A-MPDU sub-frame is sent to this node.

6. References

1. *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks – Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11, 2007.

2. *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE 802.11p, 2010.

3. A. Kamerman and L. Monteban, “WaveLAN-II: A high performance wireless LAN for the unlicensed band,” *Bell Labs technical J.*, vol. 2(3), 118–133, Summer 1997.

4. *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*, IEEE 802.11ac, 2013.

5. *IEEE Standard for Information Technology - Telecommunications and information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band*, IEEE 802.11ad, 2012.

6. *IEEE Draft Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment - Sub 1 GHz License-Exempt Operation*, IEEE 802.11ah Draft 0.2, 2014.

Copyright© 2007-2016 Space-Time Engineering, LLC

SPACE-TIME®
ENGINEERING

Copyright© 2007-2016 Space-Time Engineering, LLC