



SCENARGIE®

Scenargie®2.1 Dot Eleven Module

ユーザガイド

Space-Time Engineering, LLC

2016 年 9 月

目次

はじめに.....	1
1. プロダクトの内容.....	2
2. インストール方法.....	3
2.1. Linux、MacOS 環境へのインストール.....	3
2.1.1. シミュレータ実行ファイルの作成.....	3
2.1.2. Visual Lab で使用する場合の設定 Linux 編.....	5
2.1.3. Visual Lab で使用する場合の設定 MacOS 編.....	6
2.2. Windows 環境へのインストール.....	8
2.2.1. シミュレータ実行ファイルの作成.....	8
2.2.2. Visual Lab で使用する場合の設定.....	9
2.3. サンプルシナリオ.....	12
2.3.1. dot11_aggregation.....	12
2.3.2. dot11_crossroad.....	13
2.3.3. dot11_indoor.....	14
2.3.4. dot11_manet.....	15
2.3.5. dot11_handover.....	16
2.3.6. dot11_dtn.....	17
2.3.7. dot11_multibandwidth.....	18
2.4. 旧サンプルシナリオの実行.....	19
2.4.1. Scenargie1.8 r19737 版/Scenargie2.0 r19737 版サンプルシナリオの実行.....	19
2.4.2. Scenargie2.1 r20324 版サンプルシナリオの実行.....	21
3. プロパティ.....	22
3.1. プロパティ一覧.....	22
3.1.1. Dot Eleven Standard プロパティ一覧.....	22
3.1.2. Dot Eleven Professional プロパティ一覧.....	34
4. 統計値、トレース.....	38
4.1. 統計値の標準設定一覧表.....	38
4.2. トレースの標準設定一覧表.....	42
5. モデル概要.....	45
5.1. 変調方式とデータレート.....	45
5.2. Mac 関連 API.....	46
5.2.1. Dot11Mac.....	46
5.2.2. SimpleMacAddressResolver.....	50
5.2.3. Dot11ApManagementController.....	51
5.2.4. Dot11StaManagementController.....	53

5.2.5.	AdaptiveRateController	54
5.2.6.	AdaptiveTxPowerController	56
5.3.	PHY 関連 API.....	58
5.3.1.	Dot11Phy	58
5.3.2.	Dot11MacInterfaceForPhy	61
6.	参考文献.....	63

はじめに


本書は、離散事象シミュレータ Scenargie2.1 Dot Eleven Module の操作方法を示すものです。

関連ドキュメント

インストレーションガイド
プログラマーズガイド
Visual Lab ユーザガイド
Base Simulator ユーザガイド

1. プロダクトの内容

Scenargie2.1 Dot Eleven Module は Scenargie2.1 Base Simulator 用オプションモジュールで、Base Simulator 本体と組み合わせて使用することを想定しています。

(図 1-1  に示す部分)

Dot Eleven Module は、IEEE802.11a/g/n/ac 通信規格を想定したシミュレーションモデルを含む Standard 版と、Standard 版のシミュレーションモデルに加え、TGn チャネルモデル、IEEE802.11ad、IEEE802.11ah 通信規格を想定したシミュレーションモデルを含む Advanced 版が存在します。

Scenargie2.1 Dot Eleven Module には以下の内容が含まれます。

- IEEE802.11 通信規格のシミュレーションモデルソースコード [1]、[2]、[3]、[4]、[5]、[6]
- makefile
- サンプルシナリオ

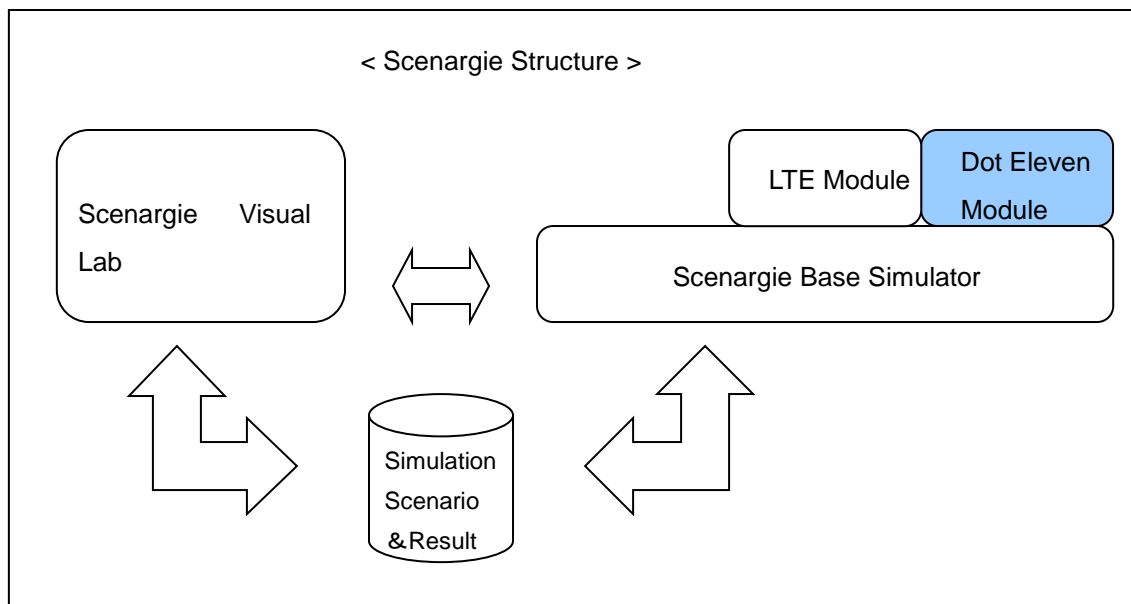


図 1-1 Scenargie システム構成

2. インストール方法

Scenargie Dot Eleven Module パッケージは、Linux 環境、MacOS 環境および Windows 環境へのインストールが可能です。

Scenargie Base Simulator インストール後、下記のようにインストールを行ってください。

Scenargie Base Simulator および Scenargie Dot Eleven Module インストール後のソースコードディレクトリ構成は下記ようになります。

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

2.1. Linux、MacOS 環境へのインストール

2.1.1.シミュレータ実行ファイルの作成

1) パッケージの展開

以下の説明では、Scenargie Base Simulator がユーザホームディレクトリ下にディレクトリ「scenargie_simulator」として展開されているものとします。Scenargie Dot Eleven Module のパッケージをユーザホームディレクトリ下にコピーまたは移動し、パッケージを展開してください。

<コマンド例>

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

2) 実行ファイルの作成

Scenargie Dot Eleven Module のインストールにより scenargie_simulator/2.1/source/dot11/ が作成され、このディレクトリにサンプルメイクファイル makefile.linux が含まれています。このサンプルメイクファイルを使用して下記のように実行ファイルを作成します。

<コマンド例>

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

上記ビルドが正常に終了した後、カレントディレクトリにシミュレータ実行ファイル「sim」が作成されます。

ビルド時にオプションを指定することで、並列ビルドや IPv6 の有効化などが行えます。また、複数のオプションモジュールを組み合わせる場合は、マルチシステム用の makefile を使用してビルドします。その際、ビルドオプションとして有効化するオプションモジュールを指定します。詳しくは、「インストレーションガイド」を参照ください。

Advanced 版をご利用で、802.11ad または 802.11ah 規格のモデルをご使用される場合は、dot11 直下の makefile ではなく、dot11/ad_version または dot11/ah_version 内の makefile を使用して実行ファイルを作成します。

<コマンド例>

```
$ 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
```

注意)

一度シミュレータ実行ファイルの作成を行った後、再作成する場合、シミュレータ実行ファイルやオブジェクトファイルを dot11 ディレクトリから事前に削除することを推奨します。

<コマンド例>

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

2.1.2. Visual Lab で使用する場合の設定 Linux 編

Linux 環境で Scenargie Visual Lab で使用する場合、以下の項目も設定します。

1) データファイルのコピー

Scenargie Dot Eleven Module パッケージには Scenargie Visual Lab で Dot Eleven Module の機能を使用するためのデータファイルが含まれます。このデータファイルが含まれるディレクトリ (data/dot11) を Scenargie Visual Lab のインストールディレクトリにコピーします。

データファイル:

```
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 (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/data/dot11/dot11_advanced.objtype (Advanced 版のみ)
```

コピー先ディレクトリ:

Scenargie Visual Lab インストールディレクトリ下の「data/」ディレクトリ下
例) visuallab/data/dot11/

2) サンプルファイルのコピー

Scenargie Dot Eleven Module パッケージには Scenargie Visual Lab で Dot Eleven の機能を使用するためのサンプルファイルが含まれます。このサンプルファイルを Scenargie Visual Lab のインストールディレクトリにコピーします。

サンプルファイル:

```
scenargie_simulator/2.1/visuallab/sample/dot11modes.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes.ber (Advanced 版のみ)
```


scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel1.ber (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel2.ber (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel3.ber (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel4.ber (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel5.ber (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel6.ber (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/sample/dot11ahmodes.ber (Advanced 版のみ)

コピー先ディレクトリ:

Scenargie Visual Lab インストールディレクトリ下の「sample」ディレクトリ下
 例) visuallab/sample/

3) シミュレータ実行ファイルの指定

[Tools]- [Object Properties] Global: Simulation Executable Name に作成したシミュレータ実行ファイルを指定します。

2.1.3. Visual Lab で使用する場合の設定 MacOS 編

MacOS 環境で Scenargie Visual Lab で使用する場合、以下の項目も設定します。

1) データファイルのコピー

Scenargie Dot Eleven Module パッケージには Scenargie Visual Lab で Dot Eleven Module の機能を使用するためのデータファイルが含まれます。このデータファイルが含まれるディレクトリ (data/dot11) を Scenargie Visual Lab のインストールディレクトリにコピーします。

データファイル:

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 (Advanced 版のみ)
 scenargie_simulator/2.1/visuallab/data/dot11/dot11_advanced.objtype (Advanced 版のみ)

コピー先ディレクトリ:

Scenargie Visual Lab インストールディレクトリ下の「data/」ディレクトリ下
 例) /Applications/Scenargie.app/Contents/data/dot11/

2) サンプルファイルのコピー

Scenargie Dot Eleven Module パッケージには Scenargie Visual Lab で Dot Eleven Module の機能を使用するためのサンプルファイルが含まれます。このサンプルファイルを Scenargie Visual Lab のインストールディレクトリにコピーします。

サンプルファイル:

```
scenargie_simulator/2.1/visuallab/sample/dot11modes.ber
scenargie_simulator/2.1/visuallab/sample/dot11admodes.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel1.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel2.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel3.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel4.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel5.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11admodes_chmodel6.ber (Advanced 版のみ)
scenargie_simulator/2.1/visuallab/sample/dot11ahmodes.ber (Advanced 版のみ)
```

コピー先ディレクトリ:

Scenargie Visual Lab インストールディレクトリ下の「sample」ディレクトリ下
例) /Applications/Scenargie.app/Contents/sample/

3) シミュレータ実行ファイルの指定

[Tools]- [Object Properties] Global: Simulation Executable Name に作成したシミュレータ実行ファイルを指定します。

2.2. Windows 環境へのインストール

Scenargie Dot Eleven Module の Windows 環境へのインストールは管理者アカウントで実施します。

<コマンド例>は、Visual C++のインストールにより作成される「Visual Studio 2013 Command Prompt」での実行を想定しています。

32bit でのビルドを行う場合は、

[スタート]-[プログラム]- [Visual Studio 2013]-[Visual Studio Tools]-[VS2013 x86 Native Tools Command Prompt]をクリックして起動される Command Prompt を使用してください。

64bit でのビルドを行う場合は、

[スタート]-[プログラム]- [Visual Studio 2013]-[Visual Studio Tools]-[VS2013 x64 Cross Tools Command Prompt]をクリックして起動される Command Prompt を使用してください。

2.2.1. シミュレータ実行ファイルの作成

1) パッケージの展開

以下の説明では、Scenargie Base Simulator がCドライブのルート下に「C:¥scenargie_simulator」として展開されているものとします。Scenargie Dot Eleven Module のパッケージを C ドライブのルート下にコピーまたは移動し、パッケージを展開してください。

<操作例>

エクスプローラ等で入手したパッケージ Scenargie-2.1-DotElevenModule-Standard-rxxxx.zip または Scenargie-2.1-DotElevenModule-Advanced-rxxxx.zip を C:¥に展開する。

2) 実行ファイルの作成

Scenargie Dot Eleven Module の イ ン ス ト ー ル に よ り scenargie_simulator¥2.1¥source¥dot11¥ が作成され、このディレクトリにサンプルメイクファイル makefile.win が含まれています。このサンプルメイクファイルを使用して下記のように実行ファイルを作成します。

<コマンド例>

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

上記ビルドが正常に終了した後、カレントディレクトリにシミュレータ実行ファイル「sim.exe」が作成されます。

ビルド時にオプションを指定することで、並列ビルドや IPv6 の有効化などが行うことができます。また、複数のオプションモジュールを組み合わせて使用する場合は、マルチシステム用の makefile を使用してビルドします。その際、ビルドオプションとして有効化するオプションモジュールを指定します。詳しくは、「インストレーションガイド」を参照ください。

Advanced 版をご利用で、802.11ad または 802.11ah 規格のモデルをご使用される場合は、dot11 直下の makefile ではなく、dot11¥ad_version または dot11¥ah_version 内の makefile を使用して実行ファイルを作成します。

<コマンド例>

```
> 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
```

注意)

一度シミュレータ実行ファイルの作成を行った後、再作成する場合、シミュレータ実行ファイルやオブジェクトファイルを dot11 ディレクトリから事前に削除することを推奨します。

<コマンド例>

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

2.2.2. Visual Lab で使用する場合の設定

Scenargie Visual Lab で使用する場合、以下の項目も設定します。

1) データファイルのコピー

Scenargie Dot Eleven Module パッケージには Scenargie Visual Lab で Dot Eleven の機能を使用するためのデータファイルが含まれます。このデータファイルが含まれるディレクトリ(data¥dot11)を Scenargie Visual Lab のインストールディレクトリにコピーします。

データファイル:

```
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 (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥data¥dot11¥dot11_advanced.objtype (Advanced 版のみ)

コピー先ディレクトリ:

Scenargie Visual Lab インストールディレクトリ下の「data¥」ディレクトリ下

例) C:¥Scenargie¥data¥dot11¥

2) サンプルファイルのコピー

Scenargie Dot Eleven Module パッケージには Scenargie Visual Lab で Dot Eleven Module の機能を使用するためのサンプルファイルが含まれます。このサンプルファイルを Scenargie Visual Lab のインストールディレクトリにコピーします。

サンプルファイル:

scenargie_simulator¥2.1¥visuallab¥sample¥dot11modes.ber

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes_chmodel1.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes_chmodel2.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes_chmodel3.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes_chmodel4.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes_chmodel5.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11admodes_chmodel6.ber (Advanced 版のみ)

scenargie_simulator¥2.1¥visuallab¥sample¥dot11ahmodes.ber (Advanced 版のみ)

コピー先ディレクトリ:

Scenargie Visual Lab インストールディレクトリ下の「sample」ディレクトリ下

例) C:¥Scenargie¥sample¥

3) シミュレータ実行ファイルの指定

[Tools]- [Object Properties] Global: Simulation Executable Name に作成したシミュレータ実行ファイルを指定します。

上記 1)および 2)の操作は、インストールスクリプトにより実行可能です。

インストールスクリプト:

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

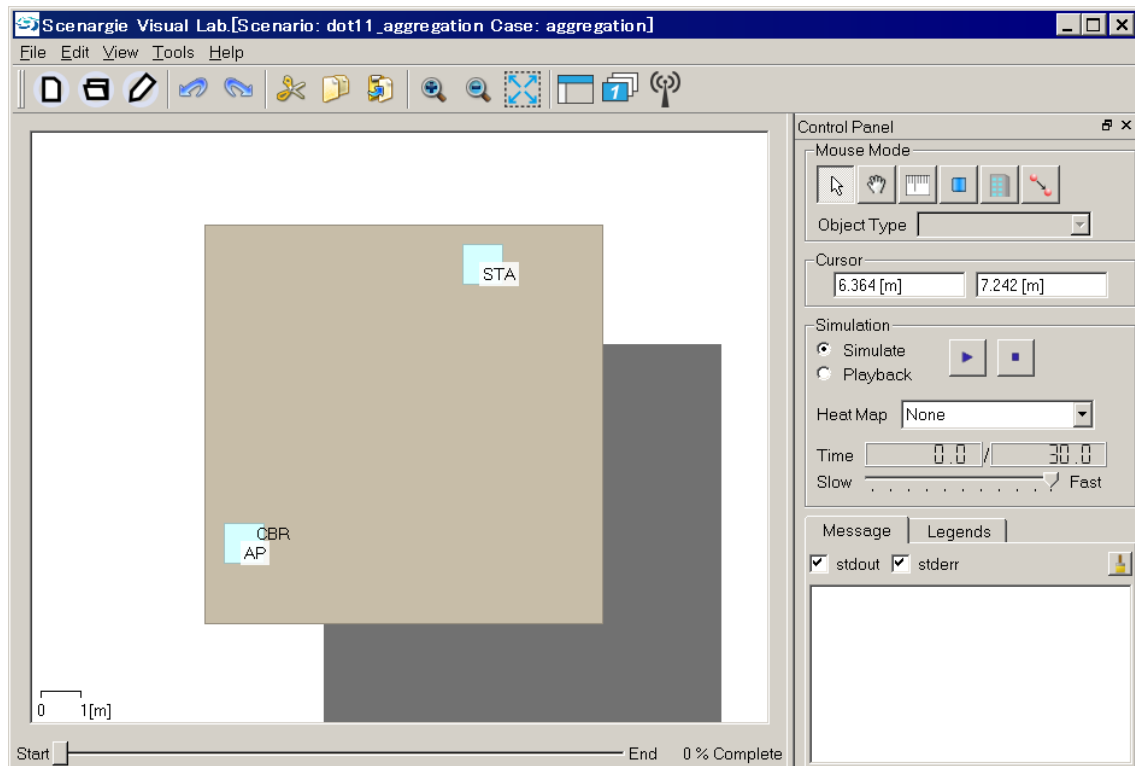
2.3. サンプルシナリオ

サンプルシナリオは、前節に記載の「パッケージの展開」により以下のディレクトリに展開されます。

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

Dot Eleven Module のサンプルシナリオは dot11_ をプリフィックスとするディレクトリになります。Scenargie Visual Lab 用のシナリオは当該ディレクトリの .case ファイルを読み込んで使用します。コマンドライン実行用のシナリオは当該ディレクトリ内の commandline ディレクトリに展開されます。

2.3.1.dot11_aggregation



- シナリオ構成

通信オブジェクト(IEEE802.11n):

- AP×1
- STA×1

アプリケーション:

CBR : AP → STA

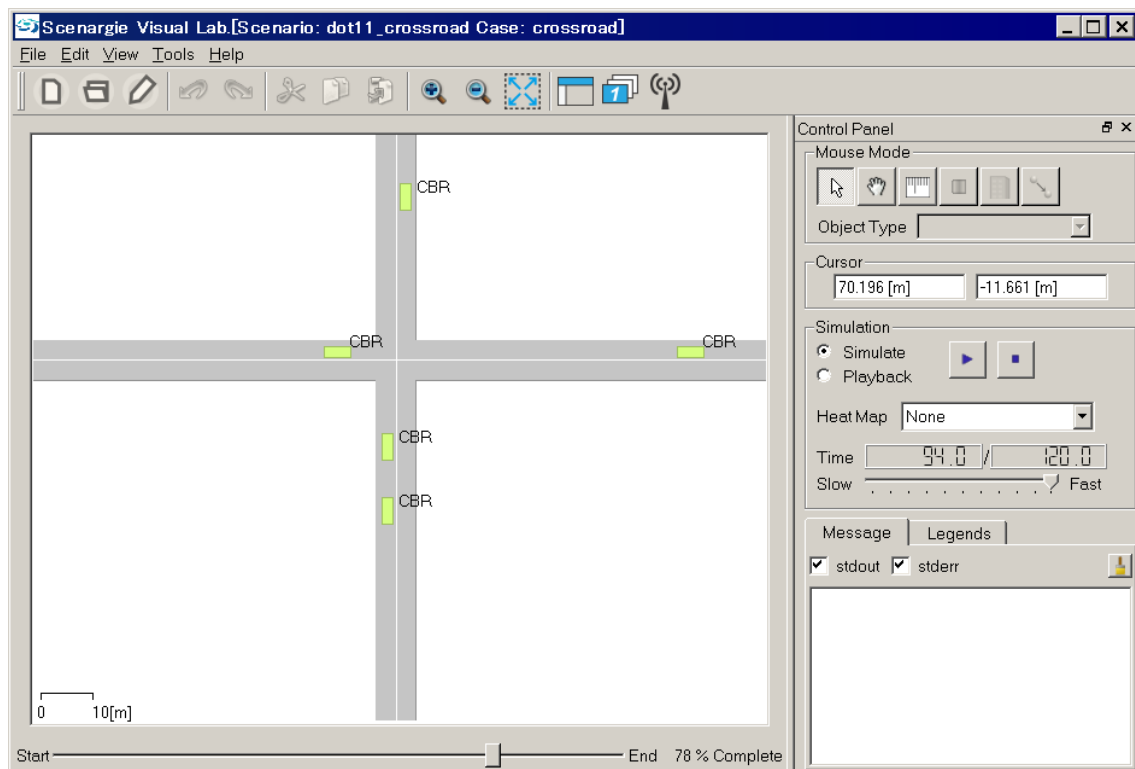
パスロスモデル:

WallCount

- シナリオ概要

本シナリオは、屋内環境において IEEE802.11n の使用を想定したシミュレーションを行っています。屋内の 1 階に配置された AP から、屋内の 2 階を動き回る STA に対して CBR アプリケーションのパケットを A-MPDU によるフレームアグリゲーションの機能を利用して送信します。

2.3.2.dot11_crossroad



- シナリオ構成

通信オブジェクト(IEEE802.11p):

- Dot11p(mobile) × 27

アプリケーション:

CBR : Dot11p → * (Broadcast)

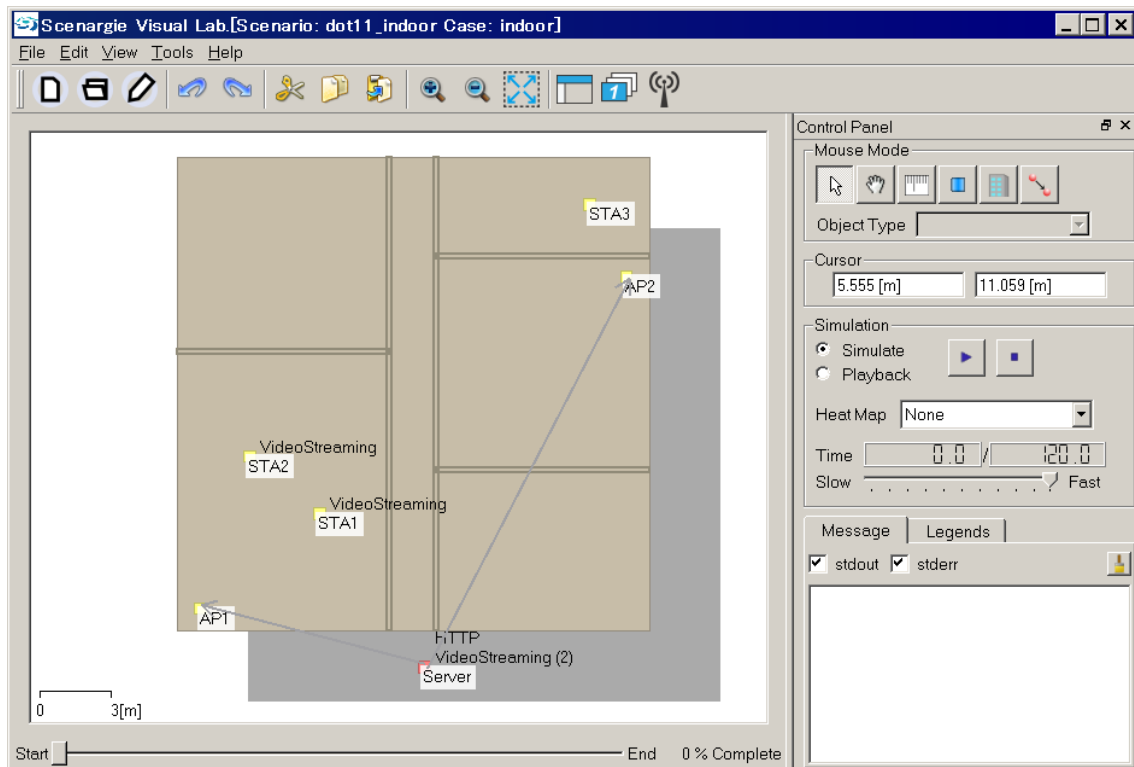
- シナリオ概要

本シナリオは、モビリティ設定ファイルを使用しています。通信オブジェクトは、十字路交差点の端点から生成され、道路上を移動して端点で消滅します。

※本シナリオで使用している通信オブジェクトのモビリティは、MATES※¹を利用して作成したトレース情報を利用しています。

※¹Multi-Agent based Traffic and Environment Simulator=東京大学吉村研究室で開発された知的マルチエージェント交通流シミュレータ

2.3.3.dot11_indoor



- シナリオ構成

通信オブジェクト:

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

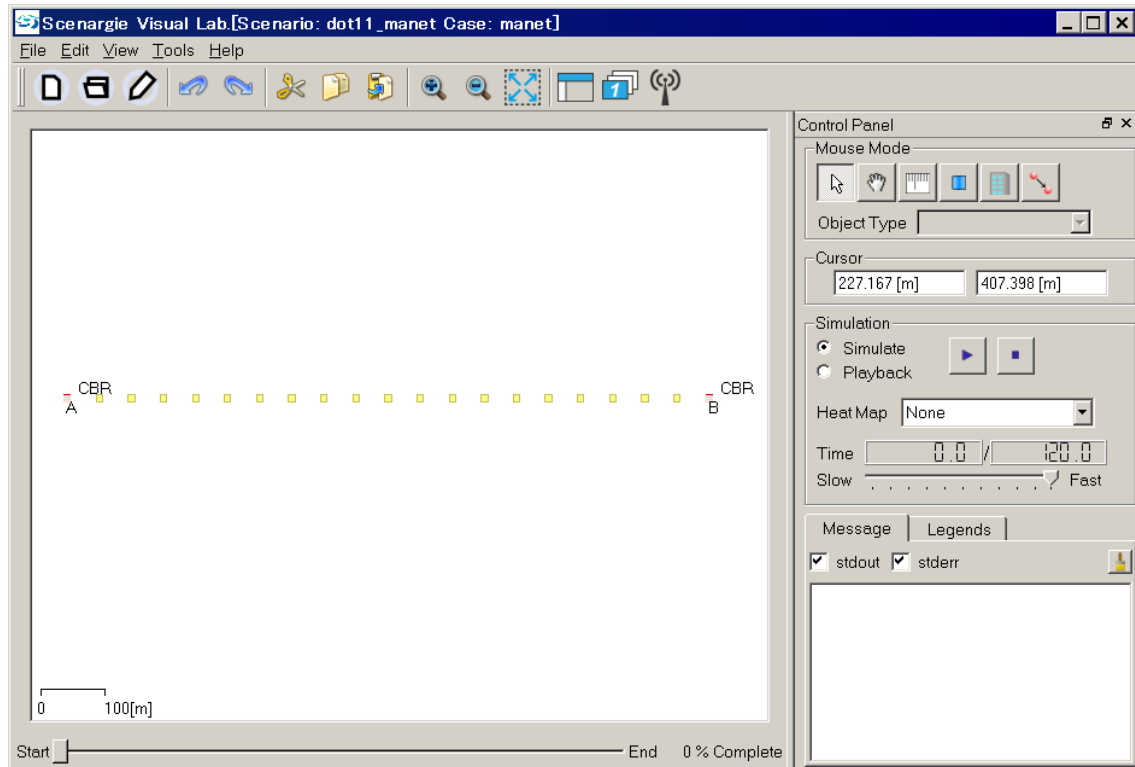
アプリケーション:

VideoStreaming : Server → STA1、STA2
 VideoStreaming : STA1、STA2 → Server
 HTTP : Server → STA3

- シナリオ概要

本シナリオは、屋内環境において IEEE802.11g の使用を想定したシミュレーションを行っています。電波伝搬モデルには、壁などの影響を考慮可能な COST231Indoor モデルを使用しています。

2.3.4.dot11_manet



- シナリオ構成

通信オブジェクト:

- Dot11g × 21

アプリケーション:

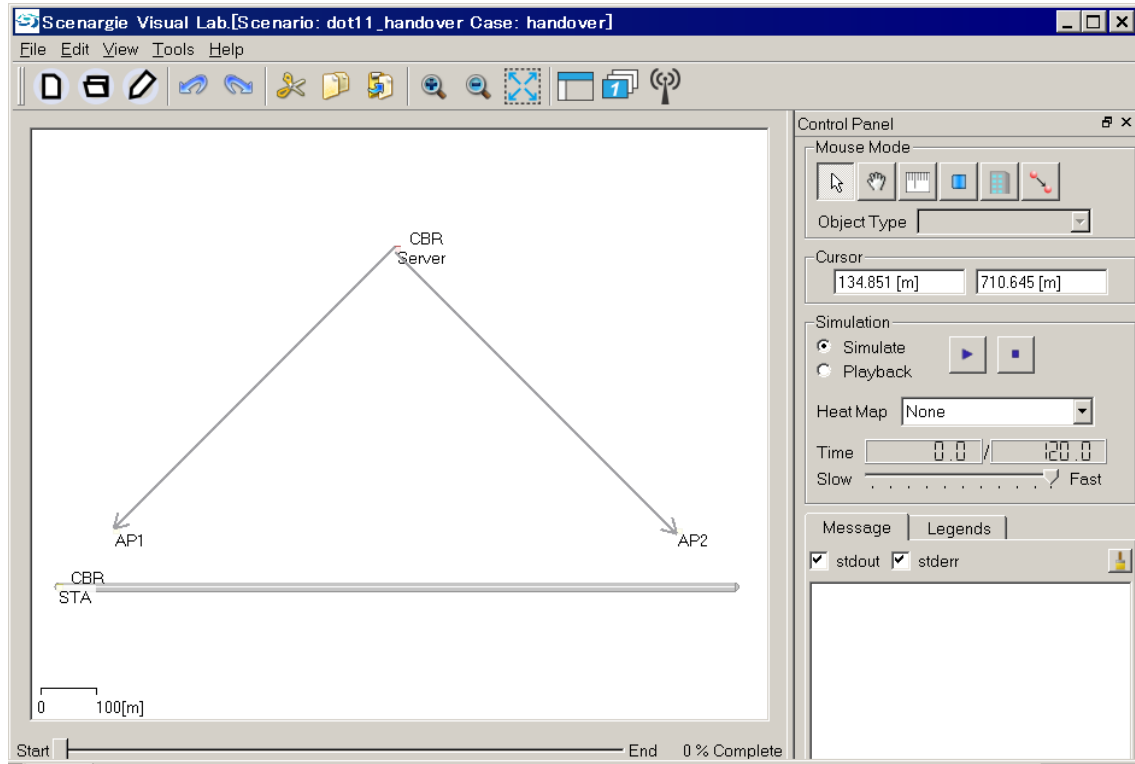
CBR : A → B

CBR : B → A

- シナリオ概要

本シナリオは、OLSRv2 を使用したアドホックネットワークのシミュレーションを行っています。両端のノード間から送信された CBR アプリケーションのパケットは複数のノードを経由し宛先ノードへ転送されます。

2.3.5.dot11_handover



- シナリオ構成

通信オブジェクト:

- Serverx1
- AP(IEEE802.11g) × 2
- STA(IEEE802.11g) × 1

アプリケーション:

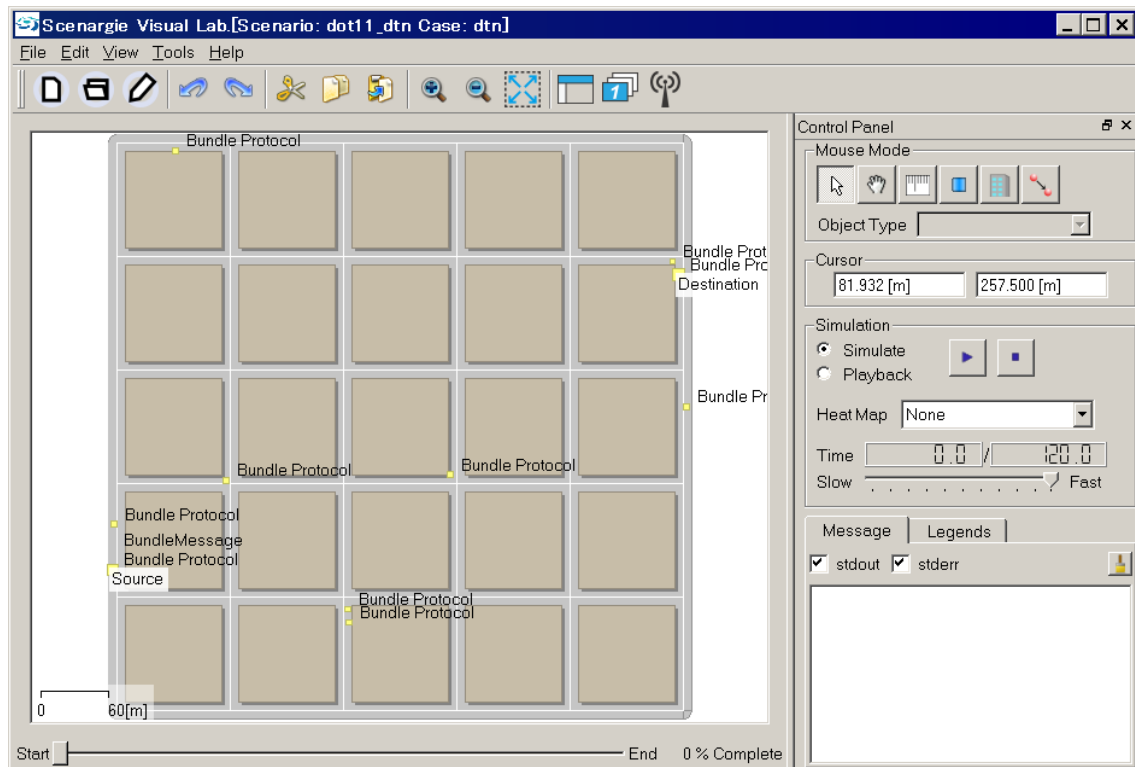
CBR : Server → STA

CBR : STA → Server

- シナリオ概要

本シナリオは、STA が移動することで、STA が 2 つの AP 間をハンドオーバーするシナリオです。Server と STA 間で最初は AP1 を経由し、ハンドオーバー後は AP2 を経由してアプリケーションの送受信を行います。STA には、DHCP により動的に IP アドレスが割り与えられます。

2.3.6.dot11_dtn



- シナリオ構成

通信オブジェクト(IEEE802.11g):

- Source×1
- Destination × 1
- 中継ノード × 8

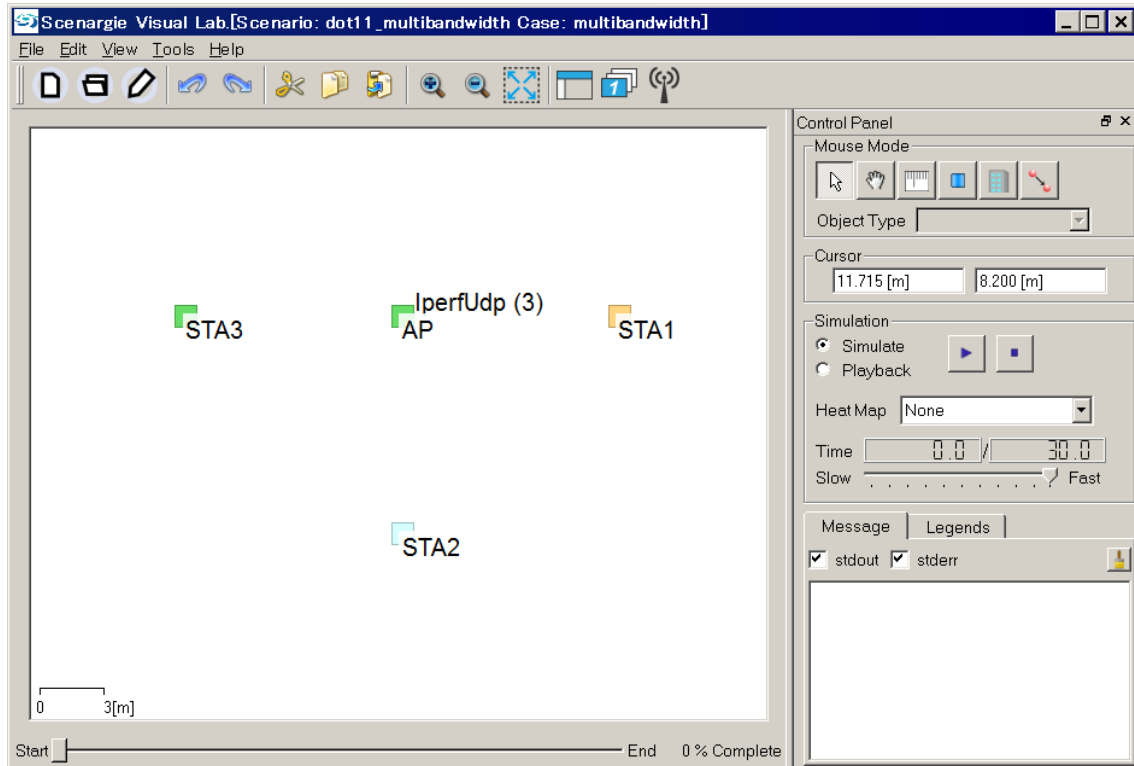
アプリケーション:

BundleMessage : Source → Destination

- シナリオ概要

本シナリオは、DTN (Delay/Disruption Tolerant Network)を想定したシナリオです。Source と Destination にエンドツーエンドの通信リンクが存在しない場合に、BundleProtocol(Epidemic ルーティングアルゴリズム)を使用することで、中継ノードを経由して Source と Destination 間のメッセージ送受信を行います。Trace Visualization Settings の設定により、Bundle 転送時に送受信ノード間のリンクを表示しています。

2.3.7.dot11_multibandwidth



- シナリオ構成

通信オブジェクト

- Dot11a × 1 (STA1)
- Dot11n × 1 (STA2)
- Dot11ac × 2 (AP、STA3)

アプリケーション:

IperfUdp : AP → STA1
 IperfUdp : AP → STA2
 IperfUdp : AP → STA3

- シナリオ概要

本シナリオは、IEEE 802.11ac 規格のインターフェースを持つ AP が、異なるチャネル帯域幅を使用する 3 台の STA と通信を行うことを想定したシナリオです。STA1 は 20MHz、STA2 は 40MHz、STA3 は 160MHz のチャネル帯域幅を使用します。AP は、Iperf アプリケーションを利用して、各 STA と通信を行います。

2.4. 旧サンプルシナリオの実行

新たにインストールされた Scenargie2.1 Visual Lab、または、新たにビルドされたシミュレータ実行ファイルを用いて旧サンプルシナリオを実行する方法を説明します。

注意) データファイル、サンプルファイルも正しくインストールされていることを確認してください。

2.4.1.Scenargie1.8 r19737 版/Scenargie2.0 r19737 版サンプルシナリオの実行

dot11_indoor

Visual Lab 用サンプルシナリオを実行するためには、Visual Lab で indoor.case を開き、Object Properties で以下のように設定を変更してください。

プロパティ	変更前の値	変更後の値
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

コマンドライン用サンプルシナリオを実行するためには、simulation.config を以下のように編集してください。

変更前	変更後
[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 = 192.168.1.254	[2;dot11g] network-address = 192.168.1.0 + \$n
[3;dot11g] network-address = 192.168.2.254	[3;dot11g] network-address = 192.168.2.0 + \$n

dot11_ad

Visual Lab 用サンプルシナリオを実行するためには、ad.case ファイルがあるのと同じフォルダに 2.1 の同シナリオから dot11admodes.ber をコピーしてください。

コマンドライン用サンプルシナリオを実行するためには、simulation.config ファイルがあるのと同じフォルダに 2.1 の同シナリオから dot11admodes.ber をコピーしてください。

また、simulation.config に以下の行を追加してください。

```
[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
```

それ以外のシナリオ

dot11_indoor、dot11_ad 以外の旧サンプルシナリオはそのまま実行可能です。

2.4.2.Scenargie2.1 r20324 版サンプルシナリオの実行

Visual Lab 用サンプルシナリオの実行

旧サンプルシナリオは、そのまま実行可能です。

コマンドライン用サンプルシナリオの実行

旧サンプルシナリオは、そのまま実行可能です。

3. プロパティ

3.1. プロパティ一覧

以下はコンフィギュレーションファイルで定義可能な Scenargie Dot11 Module 固有プロパティです。コンフィギュレーションファイルの記述方法および Scenargie Base Simulator 共通プロパティについては Scenargie Base Simulator ユーザガイドを参照してください。

デフォルト値が「なし」と記述されているパラメータは、モデル内での初期値が存在せず、何らかの値の設定が必要な項目になります。

型が時間の場合、下記のように時間＋単位:s(秒)、ms(ミリ秒)、us(マイクロ秒)、ns(ナノ秒)で表すことができます。単位を省略した場合、秒として扱われます。

<例>

100 秒: 100s、または、100

100 ミリ秒: 100ms、または、0.1

100 マイクロ秒: 100us、または、0.0001

100 ナノ秒: 100ns、または、0.0000001

尚、デフォルト値は、コンフィギュレーションファイルにパラメータが記載されていない場合に使用される初期値であり、VisualLab で初期設定される値とは異なります。

3.1.1.Dot Eleven Standard プロパティ一覧

システム設定

パラメータ名	スコープ	型	デフォルト値	説明(範囲:単位)
dot11-bit-error-rate-curve-file	グローバル	文 字 列	なし	ビットエラーレートカーブのファイル名
dot11-channel-model	グローバル	文 字 列	空文字	チャンネルモデルの設定 (MIMO を使う場合のみ、設定) モデル種別: MIMO、TGnMIMO

MAC 設定

パラメータ名	スコープ	型	デフォルト値	説明(範囲:単位)
--------	------	---	--------	-----------

dot11-node-type	インターフェース	文 字 列	Ad-hoc	ノード種別 : Access-Point、 Mobile-STA、Ad-hoc
dot11-map-ip-multicast-addresses	インターフェース	Bool	false	IP マルチキャストを行うか否か(AdhocかAPモードのノードが、CBR、VBR を送信する場合にのみマルチキャストに対応、送信および受信を行うノードは、必ず true に設定
dot11-multicast-group-number-list	インターフェース	文 字 列	なし	IP マルチキャスト使用時における所属グループ番号のリスト 例: グループ番号 0、1 に所属する場合: "0 1"
dot11-enable-high-throughput-mode	インターフェース	Bool	false	HT (High Throughput) モード使用の ON/OFF
dot11-force-use-of-high-throughput-frames	インターフェース	Bool	false	HT モードが使用可能な時に常に HT モードを使用するか否か
dot11-bonded-channel-number-list	インターフェース	文 字 列	なし	HT モード使用時におけるチャンネルボンディング用チャンネル番号リスト 例: チャンネル番号 0、1 を使用する場合: "0 1"
dot11-max-channel-bandwidth-mhz	インターフェース	実数	なし	チャンネルボンディング時の最大チャンネル幅
dot11-rts-threshold-size-bytes	インターフェース	整数	2346	RTS を送信するパケットサイズの閾値 本 閾 値 より も 長 い PSDU 長を送信する場合、最初に RTS フレームが送信されます。
dot11-short-frame-retry-limit	インターフェース	整数	7	ショートフレーム

	ス			(dot11-rts-threshold-size -byte よりも PSDU 長が 短いフレーム)の最大 再送回数
dot11-long-frame-retry-limit	インターフェース	整数	4	ロングフレーム (dot11-rts-threshold-size -byte よりも PSDU 長が 長いフレーム)の最大 再送回数
dot11-disallow-adding-new-packets-to-retries	インターフェース	Bool	true	再送時に A-MPDU に 新しいパケットの追加を 拒否するか否かの識別 (true の場合、パケット の追加は行わない)
dot11-contention-window-min-slots	インターフェース	整数	15	DCF および EDCA のコ ンテンション・ウィンドウ (CW) サイズの計算に 用いられる最小スロット 数
dot11-contention-window-max-slots	インターフェース	整数	1023	DCF および EDCA のコ ンテンション・ウィンドウ (CW) サイズの計算に 用いられる最大スロット 数
dot11-disabled-to-jump-on-medium-without-backoff	インターフェース	Bool	false	アイドル時にフレーム送 信する際、強制的にバ ックオフをさせる機能の ON/OFF (true: バック オフあり)
dot11-max-packet-priority	インターフェース	整数	3	パケットの最大優先度 番号 (mac-protocol が dot11ad か dot11ah の 場合、または、ITS Extension Module 利

				用時に使用)
dot11-qos-type	インターフェース	文 字 列	EDCA	QoS タイプ: DCF、EDCA
dot11-dcf-num-difs-slots	インターフェース	整数	2	DCF における DIFS 用 スロット数
dot11-dcf-contention-window-min-slots	インターフェース	整数	15	DCF におけるコンテン ション・ウィンドウ(CW) サイズの計算に用いら れる最小スロット数
dot11-dcf-contention-window-max-slots	インターフェース	整数	23	DCF におけるコンテン ション・ウィンドウ(CW) サイズの計算に用いら れる最大スロット数
dot11-dcf-frame-lifetime	インターフェース	時間	無限大	DCF におけるフレーム 生存時間
dot11-num-edca-access-categories	インターフェース	整数	4	EDCA のアクセスカテ ゴリ数
dot11-edca-category-<n>-priority-list <n>は、アクセスカテゴリ番号	インターフェース	文 字 列	0/1/2/3 (それぞれ、 AC=0/1/2/3 に対応)	EDCA におけるアクセ スカテゴリ<n>番に対応 する優先度番号のリス ト 記述例: アクセスカテゴリ(AC)0 に対する優先度が 0、1 の場合 “0 1”と記載のこと
dot11-edca-category-<n>-num-aifs-slots <n>は、アクセスカテゴリ番号	インターフェース	整数	9/6/3/2 (それぞれ、 AC=0/1/2/3 に対応)	EDCA におけるアクセ スカテゴリ<n> 番 の AIFSN
dot11-edca-category-<n>-contention-window-min-slots <n>は、アクセスカテゴリ番号	インターフェース	整数	15/7/3/3 (それぞれ、 AC=0/1/2/3 に対応)	EDCA におけるアクセ スカテゴリ<n>番のコン テンション・ウィンドウ (CW) サイズの計算に 用いられる最小スロット 数

dot11-edca-category-<n>-contention-window-max-slots <n>は、アクセスカテゴリ番号	インターフェース	整数	1023/1023/7/7 (それぞれ、AC=0/1/2/3に対応)	EDCA におけるアクセスカテゴリ<n>番のコンテンション・ウィンドウ (CW) サイズの計算に用いられる最大スロット数
dot11-edca-category-<n>-frame-lifetime <n>は、アクセスカテゴリ番号	インターフェース	時間	無限大	EDCA におけるアクセスカテゴリ<n>番に対応するフレーム生存時間
dot11-edca-category-<n>-downlink-txop-duration <n>は、アクセスカテゴリ番号	インターフェース	時間	0ms	EDCA におけるアクセスカテゴリ<n>番に対応する TXOP リミット (チャネル占有時間)
dot11-edca-category-<n>-max-non-fifo-aggregate-size-bytes <n>は、アクセスカテゴリ番号	インターフェース	整数	なし	EDCA におけるアクセスカテゴリ<n>番の最大連結バイト数
dot11-ap-scheduling-algorithm	インターフェース	文 字 列	FIFO	スケジューリングアルゴリズム (dot11-node-type が Access-Point の場合のみ設定) 種 別 : FIFO 、 RoundRobin
dot11-max-aggregate-mpdu-size-bytes	インターフェース	整数	0	最大 A-MPDU サイズ (A-MSDU 使用時には 0 に設定する必要があります) 単位: バイト
dot11-max-num-aggregate-subframes	インターフェース	整数	64	A-MPDU に含めることが可能な最大の subframe 数
dot11-protect-aggregate-frames-with-single-acked-frame	インターフェース	Bool	true	一つめのフレームの ACK が戻ってきてからアグリゲーションを行う

				か否か
dot11-allow-frame-aggregation-with-txop-zero	インターフェース	Bool	false	TXOP が 0 の場合でも、アグリゲーションを行うか否か

PHY 設定

パラメータ名	スコープ	型	デフォルト値	説明(範囲:単位)
dot11-phy-use-short-guard-interval-and-shrink-ofdm-symbol-duration	グローバル	Bool	false	ショートガードインターバルを使うか否か
dot11-phy-protocol	インターフェース	文字列	なし	PHY プロトコルの種別 (IEEE802.11)
dot11-tx-power-specified-by	インターフェース	文字列	PhyLayer	送信電力指定レイヤ (PhyLayer、UpperLayer)
dot11-tx-power-dbm	インターフェース	実数	なし	送信電力 (送信電力 PhyLayer 指定の場合) 単位: dBm
dot11-default-tx-power-dbm-when-not-specified	インターフェース	実数	なし	送信電力が上位レイヤで指定されていない場合の送信電力 (送信電力 UpperLayer 指定の場合) 単位: dBm
dot11-radio-noise-figure-db	インターフェース	実数	なし	雑音指数 単位: dB
dot11-preamble-detection-power-threshold-dbm	インターフェース	実数	なし	プリアンプルを検知するのに必要な最低電力 単位: dBm
dot11-preamble-detection-probability-for-sinr-db-table	インターフェース	文字列	空文字	SINR の値に対するプリアンプルの検知確率のテーブル [SINR:確率] 例) -0.5:0.0 -3.0:0.5 -1.0:0.9 1.0:0.99

dot11-energy-detection-power-threshold-dbm	インターフェース	実数	なし	プリアンプルを検知できなかった場合にチャンネルをビジーと判定するための電力 単位: dBm (dot11-preamble-detection-power-threshold-dbm の値より、通常 20dB 高い値を持つ)
dot11-signal-capture-ratio-threshold-db	インターフェース	実数	1000.0	パケットキャプチャー (パケットの乗り換え)を行うかどうかを判定する電力差閾値 単位: dB
dot11-ofdm-symbol-duration	インターフェース	時間	なし	OFDM シンボル長
dot11-slot-time	インターフェース	時間	なし	1 スロットあたりの時間
dot11-sifs-time	インターフェース	時間	なし	SIFS の長さ
dot11-rx-tx-turnaround-time	インターフェース	時間	なし	受信状態から送信状態に切り替わるまでの遅延時間
dot11-phy-rx-start-delay	インターフェース	時間	なし	受信開始までの遅延時間
dot11-preamble-length-duration	インターフェース	時間	なし	プリアンプルの長さ
dot11-short-training-field-duration	インターフェース	時間	dot11-preamble-length-duration で指定された値の半分	ショートトレーニングフィールド (STF) の長さ
dot11-plcp-header-length-duration	インターフェース	時間	なし	PLCP ヘッダの長さ
dot11-phy-high-throughput-header-additional-duration	インターフェース	時間	なし	HT モード使用時の追加 PHY ヘッダ長

dot11-phy-high-throughput-header-additional-per-stream-duration	インターフェース	時間	なし	HT モード時のストリームあたりの追加 PHY ヘッダ長
dot11-phy-artificial-frame-drop-probability-for-test	インターフェース	実数	0.0	意図的に受信フレームを破棄する確率
dot11-phy-artificial-subframe-drop-probability-for-test	インターフェース	実数	0.0	意図的に A-MPDU の受信サブフレームを破棄する確率

データレート/ MCS (変調方式・符号化率) 設定

パラメータ名	スコープ	型	デフォルト値	説明(範囲:単位)
dot11-adaptive-rate-control-type	インターフェース	文字列	Static	適応レート制御タイプ (Static 、 Arf 、 Minstrel-HT)
dot11-modulation-and-coding	インターフェース	文字列	なし	デフォルトの MCS
dot11-modulation-and-coding-table	インターフェース	文字列	なし	リンク毎の MCS 例) ノード 2: BPSK 3/4、ノード 3~5: QPSK 3/4、ノード 7、および、9: 16QAM 3/4 の場合 2: BPSK_0.75 3-5: QPSK_0.75 7,9: 16QAM_0.75
dot11-modulation-and-coding-for-management-frames	インターフェース	文字列	なし	マネジメントフレーム送信時の MCS
dot11-modulation-and-coding-for-broadcast	インターフェース	文字列	なし	ブロードキャストフレーム送信時の MCS
dot11-ack-datarate-selection-type	インターフェース	文字列	SameAsData	ACK フレームのデータレートタイプ (SameAsData 、 Table)
dot11-ack-datarate-selection-table	インターフェース	文字列	なし	受信フレームに対する ACK フレームのデータ

				<p>レートの設定(受信フレームのデータレート: ACK フレームのデータレート)</p> <p>例) (区切り文字はスペース)</p> <p>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 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</p>
dot11-ack-datarate-match-num-spatial-streams	インターフェース	文 字 列	なし	ACK フレームの空間ストリーム数を受信フレームと一致させるか否か
dot11-modulation-and-coding-list	インターフェース	文 字 列	なし	<p>ARF お よ び Minstrel-HT 使用時に利用可能な MCS のリスト</p> <p>例) (区切り文字はスペース)</p>

				ース) 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
--	--	--	--	--

ARF 設定

パラメータ名	スコープ	型	デフォルト 値	説明(範囲:単位)
dot11-arf-timer-duration	インターフェース	時間	なし	レート上昇タイマ(設定時間 ACK の受信がなかった場合、レートを上げる)
dot11-arf-ack-in-success-count	インターフェース	整数	なし	レート上昇閾値(ACK の連続受信成功回数がこの値に達した場合、次の送信からレートを上げる)
dot11-arf-ack-in-failure-count	インターフェース	整数	なし	レート下降閾値(ACK の連続受信失敗回数がこの値に達した場合、次の送信からレートを下げる)
dot11-arf-ack-in-failure-count-of-new-rate-state	インターフェース	整数	なし	レート上昇直後におけるレート下降閾値(レート上昇時に、ACK の連続受信失敗回数がこの値に達した場合、次の送信からレートを下げ

				る)
--	--	--	--	----

Minstrel HT 設定

パラメータ名	スコープ	型	デフォルト 値	説明(範囲:単位)
dot11-minstrel-ht-max-number-of-spatial-streams	インターフェース	整数	4	最大ストリーム数
dot11-minstrel-ht-typical-transmission-unit-length-bytes	インターフェース	整数	1200	一般的な送信フレーム長(各レートでの再送上限を決定する際に利用する、フレーム長の参考値) 単位:バイト
dot11-minstrel-ht-max-retry-count	インターフェース	整数	7	任意の送信レートでの最大の再送回数
dot11-minstrel-ht-min-retry-count	インターフェース	整数	2	任意の送信レートでの最小の再送回数
dot11-minstrel-ht-sampling-retry-count	インターフェース	整数	1	サンプリング送信の再送回数
dot11-minstrel-ht-moving-average-exponentially-weight	インターフェース	実数	0.75	送信成功率の指数加重移動平均係数
dot11-minstrel-ht-good-success-rate	インターフェース	実数	0.95	送信成功率の閾値(十分な送信成功率)
dot11-minstrel-ht-bad-success-rate	インターフェース	実数	0.2	送信成功率の閾値(十分ではない送信成功率)
dot11-minstrel-ht-worst-success-rate	インターフェース	実数	0.1	送信成功率の閾値(非常に悪い送信成功率)
dot11-minstrel-ht-max-success-rate-to-estimate-throughput	インターフェース	実数	0.9	スループット推定時の最大の送信成功率
dot11-minstrel-ht-sampling-result-update-interval	インターフェース	時間	100ms	最小のサンプリング結果更新間隔
dot11-minstrel-ht-max-transmission-duration-for-a-multirate-retry-stage	インターフェース	時間	6ms	任意のレート(マルチレートリトライの 1stage 分)で占有可能な最大の送信時間

dot11-minstrel-ht-low-rate-sampling-trying-threshold	インターフェース	整数	20	低送信レートのサンプリング閾値
dot11-minstrel-ht-max-low-rate-sampling-count	インターフェース	整数	2	低送信レートの最大サンプリング数
dot11-minstrel-ht-enough-number-of-transmitted-mpdus-to-sample	インターフェース	整数	30	レートのダウングレード判定に必要な最大のMPDU送信数
dot11-minstrel-ht-sampling-transmission-interval-count	インターフェース	整数	16	サンプリング送信の間隔(N 回ごとに 1 回送信) 単位: 回数
dot11-minstrel-ht-initial-sampling-transmission-count	インターフェース	整数	4	初期状態でのサンプリング送信回数

アソシエーション (Association) 用パラメータ

パラメータ名	スコープ	型	デフォルト値	説明(範囲: 単位)
dot11-access-point-ssid	インターフェース	文字列	空文字	AP の SSID (AP) (空文字の場合は、全ての STA が接続対象)
dot11-beacon-transmit-interval	インターフェース	時間	100ms	ビーコン送信間隔(AP)
dot11-access-point-auth-processing-delay	インターフェース	時間	100ms	オーセンティケーション処理遅延時間((AP)
dot11-mobile-sta-ssid	インターフェース	文字列	空文字	接続先 SSID (STA) (空文字の場合は、全ての AP が接続対象)
dot11-channel-scan-interval	インターフェース	時間	500ms	1 チャンネル当たりのスキャン時間(STA)
dot11-channel-scan-start-time-max-jitter	インターフェース	時間	dot11-channel-scan-interval で指定された値	スキャン開始時刻の最大ジッタ (STA)
dot11-association-threshold-rs	インターフェース	実数	dot11-prea	アソシエーションに必

si-dbm	ス		mble-detection-power-threshold-dbm で指定された値	要な最低受信電力(STA)
dot11-associate-failure-timeout-interval	インターフェース	時間	1s	アソシエーション応答のタイムアウト時間(STA)
dot11-disassociation-threshold-rssi-dbm	インターフェース	実数	dot11-preamble-detection-power-threshold-dbm で指定された値 - 3dBm	ディスアソシエーション処理に移る受信電力閾値(STA)
dot11-link-status-check-interval	インターフェース	時間	dot11-channel-scan-interval ×チャンネル数	アソシエーションの接続性確認間隔(STA)
dot11-beacon-rssi-moving-average-coefficient	インターフェース	実数	0.5	ビーコンの受信電力を平均化するための平均化係数(STA)
dot11-authentication-timeout-interval	インターフェース	時間	1s	オーセンティケーション応答のタイムアウト時間(STA)
dot11-initial-channel-number	インターフェース	整数	initial-channel-number と同じ値	インターフェースのチャンネル番号

3.1.2.Dot Eleven Professional プロパティ一覧

TGn チャンネルモデル設定

パラメータ名	スコープ	型	デフォルト値	説明(範囲:単位)
--------	------	---	--------	-----------

tgn-mimo-channel-model-letter	グローバル	文字列	なし	MIMO で使用するチャネルモデル モデル種別： B、C、D、E
tgn-mimo-channel-number-antennas	グローバル	整数	なし	1 インターフェースあたりの MIMO に使用するアンテナ数
tgn-mimo-channel-normalized-antenna-spacing	グローバル	実数	なし	波長に対するアンテナ間隔
tgn-mimo-channel-scatterer-movement-meters-sec	グローバル	実数	0.33333333	散乱体移動速度 単位：m/s
tgn-mimo-channel-sampling-interval-time	グローバル	時間	チャンネル周波数と散乱体移動速度より自動算出	サンプリング間隔

Dot11ad 用パラメータ

パラメータ名	スコープ	型	デフォルト値	説明(範囲:単位)
dot11ad-bit-error-rate-curve-file	グローバル	文字列	なし	Dot11ad 用ビットエラーレートカーブのファイル名
dot11ad-forced-ap-pcp-nodeid	インターフェース	整数	なし	接続先 PCP のノード ID
dot11ad-number-directional-sectors	インターフェース	整数	なし	ビームフォーミングのセクタ総数
dot11ad-forced-beamforming-sector-list	インターフェース	文字列	なし	セクタを固定する場合のセクタ番号 ノード番号:セクタ番号
dot11ad-custom-sectored-antenna-model-sector-azimuths-degs	インターフェース	文字列	なし	セクタ毎のアンテナの水平方向の向き(ノード正面より時計回り)
dot11ad-custom-sectored-antenna-model-sector-pattern-names	インターフェース	文字列	なし	セクタ毎のアンテナモデル
dot11ad-use-custom-quasi-omni-	インターフェース	文字列	false	QuasiOmni 用のアン

antenna-model	一ス			テナモデルを使用するか否か
dot11ad-custom-quasi-omni-antenna-model-name	インターフェース	文字列	なし	QuasiOmni 用のアンテナモデル
dot11ad-beamforming-sector-selector-scheme-name	インターフェース	文字列	RSSI	ビームフォーミングのセクタ選択方式 : RSSI, SINR
dot11ad-abft-max-num-responder-txss-frames	インターフェース	整数	なし	セクタスイープ送信ノードの最大セクタ数
dot11ad-ap-receive-sector-sweep-interval-to-beacon	インターフェース	整数	0	AP の RXSS をビーコン何フレーム毎に行うか(0 の時は RXSS を行わない)
dot11ad-sta-receive-sector-sweep-interval-to-beacon	インターフェース	整数	0	STA の RXSS をビーコン何フレーム毎に行うか(0 の時は RXSS を行わない)
dot11ad-beacon-superframe-interval-duration	インターフェース	時間	なし	ビーコン間隔
dot11ad-beacon-transmission-interval-duration	インターフェース	時間	なし	BTI の長さ
dot11ad-association-beamforming-training-aka-abft-duration	インターフェース	時間	なし	ABFT の長さ
dot11ad-data-transfer-interval-<No>-relative-start-time	インターフェース	時間	なし	DTI-<No>の開始時間
dot11ad-data-transfer-interval-<No>-duration	インターフェース	時間	なし	DTI-<No>の長さ
dot11ad-data-transfer-interval-<No>-is-contention-based-access-period	インターフェース	Bool	なし	DTI-<No> がコンテンションベースか否かの指定
dot11ad-data-transfer-interval-<No>-source-nodeid	インターフェース	整数	なし	DTI-<No> がコンテンションベースでない時の送信ノード ID
dot11ad-data-transfer-interval-<No>-destination-nodeid	インターフェース	整数	なし	DTI-<No> がコンテンションベースでない時の受信ノード ID

dot11ad-short-beamforming-inter frame-space-duration	インターフェ ース	時間	なし	SBIFS の長さ
dot11ad-mbps-datarate-for-dot11 ad-mcs<n>	インターフェ ース	実数	なし	MCS<n>のデータレー ト (単位: Mbps) ただし、<n>は MCS イ ンデックス番号 (0 ~ 24)

Dot11ah 用パラメータ

パラメータ名	スコープ	型	デフォルト 値	説明(範囲:単位)
dot11ah-bit-error-rate-curve-file	グローバル	文字列	なし	Dot11ah 用ビットエラ ーレートカーブのファイ ル名
dot11ah-association-id-table-file	グローバル	文字列	なし	ノード ID と AID のマッ ピング情報を指定した ファイル (*.aid)
dot11ah-canned-restricted-acces s-windows-file	グローバル	文字列	なし	RAW のスケジューリン グ情報を指定したファ イル (*.raw)
dot11ah-use-optimized-ndp-contr ol-frames	グローバル	Bool	なし	レギュラーACKの代わ りに NDP ACK を使用 するか否か
dot11ah-is-a-restricted-access-wi ndow-sta	インターフェ ース	Bool	なし	STA が RAW 期間の みチャンネルアクセスを 行うかの指定

4. 統計値、トレース

4.1. 統計値の標準設定一覧表

レイヤ	モデル名	統計値	説明
MAC	Dot11Mac	BytesSent	送信フレームの総バイト数
		BytesReceived	受信フレームの総バイト数
		Data_UnicastFramesSent	ユニキャストデータフレーム送信数
		Data_UnicastFramesResent	ユニキャストデータフレーム再送数
		Data_BroadcastFramesSent	ブロードキャストデータフレーム送信数
		Data_AggregateFramesSent	アグリゲートデータの送信フレーム数
		Data_AggregateFramesResent	アグリゲートデータの受信フレーム数
		Data_FramesReceived	データフレーム受信数
		Data_AggregatedSubframesReceived	アグリゲートサブフレームの受信数
		Data_DuplicatedFramesReceived	データフレーム重複受信数
		ACK_FramesSent	ACK フレーム送信数
		ACK_FramesReceived	ACK フレーム受信数
		BlockACK_FramesSent	BlockACK の送信フレーム数
		BlockACK_FramesReceived	BlockACK の受信フレーム数
		BAR_FramesSent	BlockAckRequest の送信フレーム数
		BAR_FramesReceived	BlockAckRequest の受信フレーム数
		RTS_FramesSent	RTS フレーム送信数

		RTS_FramesReceived	RTS フレーム受信数
		CTS_FramesSent	CTS フレーム送信
		CTS_FramesReceived	CTS フレーム受信数
		Beacon_FramesSent	ビーコンフレーム送信数
		Beacon_FramesReceived	ビーコンフレーム受信数
		AssociationRequest_FramesSent	アソシエーションリクエストフレーム送信数
		AssociationRequest_FramesReceived	アソシエーションリクエストフレーム受信数
		AssociationResponse_FramesSent	アソシエーションレスポンスフレーム送信数
		AssociationResponse_FramesReceived	アソシエーションレスポンスフレーム受信数
		ReassociationRequest_FramesSent	リアソシエーションリクエストフレーム送信数
		ReassociationRequest_FramesReceived	リアソシエーションリクエストフレーム受信数
		ReassociationResponse_FramesSent	リアソシエーションレスポンスフレーム送信数
		ReassociationResponse_FramesReceived	リアソシエーションレスポンスフレーム受信数
		Disassociation_FramesSent	ディスアソシエーションフレーム送信数
		Disassociation_FramesReceived	ディスアソシエーションフレーム受信数

			受信数
		Authentication_FramesSent	オーセンティケーションフレーム送信数
		Authentication_FramesReceived	オーセンティケーションフレーム受信数
		FramesDropped	最大再送回数に達したことによるフレーム廃棄数
MAC	Dot11adMac	DmgBeacon_FramesSent	DMG Beacon フレーム送信数
		DmgBeacon_FramesReceived	DMG Beacon フレーム受信数
		SSW_FramesSent	SSW フレーム送信数
		SSW_FramesReceived	SSW フレーム受信数
		SswFeedback_FramesSent	SSW-Feedback フレーム送信数
		SswFeedback_FramesReceived	SSW-Feedback フレーム受信数
		SswAck_FramesSent	SSW-Ack フレーム送信数
		SswAck_FramesReceived	SSW-Ack フレーム受信数
PHY	Dot11Phy	FramesTransmitted	フレーム送信数
		FramesReceived	フレーム受信数
		FramesWithErrors	フレーム受信エラー数
		SignalsCaptured	シグナルキャプチャ(乗り換え)による受信エラー数
		InterferingSignals	干渉波到来数
		SignalsDuringTransmission	送信中による受信エラー数
		TooWeakToReceiveSignals	弱電力による受信エ

			ラー数
		ReceivedFrameRssiDbm	受信フレームの RSSI
		ReceivedFrameSinrDb	フレーム受信終了時 の SINR

注意事項

MACレイヤの統計値である ACK_FramesSent/ACK_FramesReceived には、DATA フレーム以外の
マネジメントフレームに対する ACK も含まれます。

PHY レイヤの統計値は、以下のような関係があります。

シグナル受信開始時:

受信失敗: InterferingSignals

そのうち弱電力だった場合: TooWeakToReceiveSignals

そのうち送信中(かつロックきる信号)だった場合: SignalsDuringTransmission

シグナル受信終了時(シグナル受信開始が成功したフレームのみ)

受信成功: FramesReceived

受信失敗(キャプチャされた場合): SignalsCaptured

受信失敗(キャプチャ以外。ビットエラー): FramesWithErrors

FramesTransmitted としてカウントされた送信されたフレームは、受信側で、InterferingSignals、
FramesReceived、FramesWithErrors のいずれかにカウントされ、以下の関係式が成り立ちます。

$$\text{FramesTransmitted} = \text{InterferingSignals} + \text{FramesReceived} + \text{FramesWithErrors}$$

尚、キャプチャされたシグナルは、キャプチャ時、所望波として SignalsCaptured にカウントされた後、
干渉波として InterferingSignals にカウントされます。

4.2. トレースの標準設定一覧表

レイヤ	モデル名	トレースイベント	追加情報	イベント説明
MAC	Dot11Mac タグ名: Mac	RxFrame	パケット ID、フレームタイプ、受信バイト数	フレーム受信
		ClearCh	-	チャネルクリア
		BusyCh	-	チャネルビジー
		NAV-Start	メディア保護時間	NAV 開始
		NAV-End	-	NAV 終了
		IFSAndBackoff-Start	アクセスカテゴリ、バックオフ時間、直前の受信フレームが破損していたか否か	バックオフ開始
		IFSAndBackoff-Pause	アクセスカテゴリ、バックオフ残り時間	バックオフ一時停止
		IFSAndBackoff-End	-	バックオフ終了
		Dequeue	アクセスカテゴリ、パケット ID	デキュー
		Tx-RTS	アクセスカテゴリ、再送数、宛先ノード ID	RTS 送信
		Tx-CTS	-	CTS 送信
		Tx-ACK	-	ACK 送信
		Tx-BlockACK	-	Block ACK 送信
		Tx-BlockACK-Request	-	Block ACK Request 送信
		Tx-DATA-B	パケット ID、アクセスカテゴリ	データフレーム（ブロードキャスト）送信
		Tx-DATA-U	パケット ID、アクセスカテゴリ、再送数、宛先ノード ID、サブフレーム数	データフレーム（ユニキャスト）送信
		Tx-DATA-A	パケット ID、アクセスカテゴリ、再送	データフレーム（アグリゲーション）送信

			数、宛先ノード ID、サブフレーム数	ン)送信
		Timeout	アクセスカテゴリ、ウィンドウスロット、再送数	タイムアウト
		Drop	パケット ID	フレーム廃棄
		Tx-Management	パケット ID、宛先ノード ID、フレームタイプ	マネジメントフレーム送信
		TxRateUpdate	データレート (Bit/Second) 、宛先ノード ID	データフレーム送信レート更新
		Traffic(packets/sec)	-	通信トラフィック (パケット/秒) (Scenargie Visual Lab の Trace Visualization Settings でのみ使用可能)
		Traffic(bits/sec)	-	通信トラフィック (ビット/秒) (Scenargie Visual Lab の Trace Visualization Settings でのみ使用可能)
MAC	Dot11adMac	Tx-DMG	パケット ID、宛先ノード ID、フレームタイプ	DMG 制御フレーム送信
PHY	Dot11Phy タグ名: Phy	TxStart	パケット ID、送信電力、伝送レート、送信時間	シグナル送信開始
		RxStart	パケット ID、受信電力	シグナル受信開始

		RxEnd	パケット ID、パケットエラー有無	シグナル受信終了
	Dot11Phy タグ名: PhyInterference	NoiseStart	シグナル送信元 ID、受信電力、干渉雑音電力、パケット ID	ノイズ受信開始
		NoiseEnd	受信電力、干渉雑音電力、パケット ID	ノイズ受信終了

5. モデル概要

5.1. 変調方式とデータレート

変調方式とデータレートの関係(non-HT モード)

	10MHz 帯域幅	20MHz 帯域幅
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	36Mbps
64QAM 2/3	24Mbps	48Mbps
64QAM 3/4	27Mbps	54Mbps

補足) Visual Lab におけるオブジェクトタイプ Dot11p では 10MHz 帯域幅を、Dot11a、Dot11g では 20MHz 帯域幅を使用します。

変調方式とデータレートの関係(HT モード、空間ストリーム数 1、800 ns ガードインターバルの場合)

	20MHz 帯域幅	40MHz 帯域幅	80MHz 帯域幅	160MHz 帯域幅
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. Mac 関連 API

ソースファイル: [dot11_mac.h/cpp](#)

5.2.1.Dot11Mac

Dot11 用 MAC レイヤモデル

戻り値	関数(引数)	説明
void	SetCustomAdaptiveRateController (const shared_ptr< Dot11::AdaptiveRateController > &rateControllerPtr)	適応レートコントローラの設定
shared_ptr < Dot11::AdaptiveRateController >	GetAdaptiveRateControllerPtr () const	適応レートコントローラのポインタの取得
shared_ptr < MacAndPhyInfoInterface >	GetMacAndPhyInfoInterface () const	MAC/PHY 情報インタフェースポインタの取得
virtual void	NetworkLayerQueueChangeNotification ()	送信キューに変化があった場合の処理
virtual void	DisconnectFromOtherLayers ()	保持しているスマートポインタの解放
virtual GenericMacAddressType	GetGenericMacAddress () const	MAC アドレスの取得 (GenericMacAddressType)
MacAddressType	GetMacAddress () const	MAC アドレスの取得 (MacAddressType)
Dot11MacOperationMode	GetOperationMode () const	オペレーションモード(アドホック、AP、STA)の取得
bool	GetIcmpMulticastAddressToMacAddressMappingIsEnabled () const	マルチキャストアドレスと MAC アドレスのマッピングが有効化否かの識別
void	SendManagementFrame (unique_ptr< Packet > &framePtr)	マネジメントフレームの送信

bool	IsAHighThroughputStation () const	HT モードの通信ノードか否かの識別
unsigned int	GetNumberOfChannels () const	チャンネル総数の取得
unsigned int	GetCurrentChannelId () const	現在のチャンネル ID の取得
unsigned int	GetMaxBandwidthNumChannels () const	最大のボンディングチャンネル数の取得
const vector< unsigned int > &	GetCurrentBondedChannelList ()	ボンディングチャンネルリストの取得
double	GetRssiOfLastFrameDbm () const	最後に受信したフレームの RSSI (dBm) の取得
void	SendAssociationRequest (const MacAddressType &apAddress)	アソシエーション要求の送信
void	SendReassociationRequest (const MacAddressType &apAddress, const MacAddressType ¤tApAddress)	リアソシエーション要求の送信
void	SendAssociationResponse (const MacAddressType &staAddress)	アソシエーション応答の送信
void	SendReassociationResponse (const MacAddressType &staAddress)	リアソシエーション応答の送信
void	SendDisassociation (const MacAddressType &receiverAddress)	ディアソシエーションの送信
void	StopReceivingFrames ()	フレームの受信停止
void	StartReceivingFrames ()	フレームの受信開始
bool	IsNotReceivingFrames () const	フレームの受信停止中か否かの識別
void	SendAuthentication (const MacAddressType &receiverAddress)	オーエンティケーションの送信
void	SendPowerSaveNullFrame (const MacAddressType	パワーセーブ用 Null フレームの送信

	&receiverAddress, const bool goingToPowerManagementMode)	
void	SwitchToChannel (const unsigned int &channel)	チャンネルの切り替え (シングルチャネル用)
void	SwitchToChannels (const vector< unsigned int > &channels)	チャンネルの切り替え
void	ResetOutgoingLinksTo (const MacAddressType &macAddress)	指定する MAC アドレス宛のリンク情報をリセット
void	RequeueBufferedPackets ()	パケットの送信キューへの再挿入
void	RequeueBufferedPacket (unique_ptr< Packet > &packetPtr, const NetworkAddress &nextHopAddress, const PacketPriorityType priority, const EtherTypeFieldType etherType, const TimeType ×tamp, const unsigned int retryTxCount)	指定するパケットの送信キューへの再挿入
void	RequeueManagementFrame (unique_ptr< Packet > &framePtr)	マネジメントフレームのキューへの再挿入
void	SendLinkIsUpNotificationToNetworkLayer ()	ネットワークレイヤへのリンク接続完了の通知 (STA)
void	SendLinkIsDownNotificationToNetworkLayer ()	ネットワークレイヤへのリンク切断の通知 (STA)
void	SendNewLinkToANodeNotificationToNetworkLayer (const MacAddressType &macAddress)	ネットワークレイヤへのリンク接続完了の通知 (AP)
void	LookupMacAddressForNeighbor (const NodeIdType nodeId, bool &wasFound, MacAddressType &macAddress)	隣接ノードの MAC アドレスの検索
bool	MpduFrameAggregationIsEnabled () const	A-MPDU が有効化されているか否かの識別

bool	MsdUFrameAggregationIsEnabled () const	A-MSDU が有効化されているか否かの識別
void	SetMpdUFrameAggregationIsEnabledFor (const MacAddressType &destinationAddress)	指定宛先への A-MPDU の有効化
void	SetMsdUFrameAggregationIsEnabledFor (const MacAddressType &destinationAddress)	指定宛先への A-MSDU の有効化
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)	Dot11Mac クラスのコンストラクタ (MIMO チャンネル用)
static shared_ptr< Dot11Mac >	Create (const ParameterDatabaseReader &theParameterDatabaseReader,	Dot11Mac クラスのコンストラクタ

	<pre> const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const shared_ptr< SimplePropagationModelForNode< PropFrameType > > &propModelInterfacePtr, const shared_ptr< BitOrBlockErrorRateCurveDatabase > &berCurveDatabasePtr, const NodeIdType &nodeId, const InterfaceIdType &interfaceId, const unsigned int interfaceIndex, const shared_ptr< NetworkLayer > &networkLayerPtr, const RandomNumberGeneratorSeedType &nodeSeed) </pre>	
--	--	--

5.2.2. SimpleMacAddressResolver

MAC アドレスリゾルバモデル

戻り値	関数(引数)	説明
	SimpleMacAddressResolver (Dot11Mac *initMacPtr)	SimpleMacAddressResolver クラス のコンストラクタ
void	GetMacAddress (const NetworkAddress &aNetworkAddress, const NetworkAddress &networkAddressMask, bool &wasFound, MacAddressType &resolvedMacAddress))	指定するネットワークアドレスから MAC アドレスを取得
void	GetNetworkAddressIfAvailable (const MacAddressType &macAddress, const	指定する MAC アドレスからネットワー クアドレスを取得

	NetworkAddress &subnetNetworkAddress, bool &wasFound, NetworkAddress &resolvedNetworkAddress)	
--	--	--

ソースファイル: [dot11_mac_ap.h/cpp](#)

5.2.3.Dot11ApManagementController

AP モード用管理モデル

戻り値	関数(引数)	説明
	Dot11ApManagementController (Dot11Mac *initMacLayerPtr, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &initNodeId, const InterfaceOrInstanceIdType &initInterfaceId, const RandomNumberGeneratorSeedType &interfaceSeed)	Dot11ApManagementController クラスのコンストラクタ
void	ProcessManagementFrame (const Packet &managementFrame)	マネジメントフレームの処理
void	ReceiveFramePowerManagement Bit (const MacAddressType &sourceAddress, const bool framePowerManagementBitIsOn)	パワーマネジメントビットの受信
bool	IsAnAssociatedStaAddress (const MacAddressType &theMacAddress) const	アソシエーションしている STA のアドレスか否かの識別

void	LookupAssociatedNodeMacAddress (const NodeIdType &nodeId, bool &wasFound, MacAddressType &macAddress) const	指定するノードIDを持つ STA の MAC アドレスの取得
void	GetAssociatedStaAddressList (vector< MacAddressType > &associatedStaAddressList) const	アソシエーションしている STA の MAC アドレスリストの取得
bool	StationIsAsleep (const MacAddressType &staAddress) const	指定する STA がスリープモードか否 かの識別
void	BufferPacketForSleepingStation (const MacAddressType &staAddress, unique_ptr< Packet > &packetPtr, const NetworkAddress &destinationNetworkAddress, const PacketPriorityType &priority, const EtherTypeFieldType etherType, const TimeType ×tamp)	スリープ中の STA 宛パケットのバッフ ァリング
void	BufferManagementFrameForSleep ingStation (const MacAddressType &staAddress, unique_ptr< Packet > &framePtr, const TimeType ×tamp)	スリープ中の STA 宛管理フレームのバッフ ァリング
void	GetPowerSaveBufferedPacket (const MacAddressType &staAddress, bool &wasRetrieved, unique_ptr< Packet > &packetToSendPtr, unsigned int &retryTxCount, PacketPriorityType &priority, EtherTypeFieldType ðerType)	バッファしていたパケットの取得

ソースファイル: [dot11_mac_sta.h/cpp](#)

5.2.4.Dot11StaManagementController

STA モード用管理モデル

戻り値	関数(引数)	説明
	Dot11StaManagementController (Dot11Mac *initMacLayerPtr, const shared_ptr< SimulationEngineInterface > &simulationEngineInterfacePtr, const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &nodeId, const InterfaceOrInstanceIdType &interfaceId, const RandomNumberGeneratorSeedType &interfaceSeed))	Dot11StaManagementController ク ラスのコンストラクタ
void	SetChannelScanningController (const shared_ptr< AbstractChannelScanningController > &scanningControllerPtr)	チャンネルスキャンコントローラの設定
void	ProcessManagementFrame (const Packet &managementFrame)	マネジメントフレームの処理
void	GetCurrentAccessPointAddress (bool &hasAnAccessPoint, MacAddressType ¤tAccessPointAddress) const	アソシエーションしている AP のアドレ スの取得
void	SwitchToAccessPoint (const MacAddressType &accessPointAddress)	接続先 AP の切り替え

ソースファイル: dot11_ratecontrol.h

5.2.5.AdaptiveRateController

適応レート制御モデルの抽象クラス

戻り値	関数(引数)	説明
virtual bool	GetHighThroughputModelsOn () const =0	HighThroughput モードが有効か否かの識別 (純粋仮想関数)
virtual unsigned int	GetBaseChannelBandwidthMhz () const =0	基準帯域幅の取得 (純粋仮想関数)
virtual void	SetMaxChannelBandwidthMhz (const unsigned int newMaxChannelBandwidthMhz)=0	最大帯域幅の設定 (純粋仮想関数)
virtual unsigned int	GetMaxChannelBandwidthMhz () const =0	最大帯域幅の取得 (純粋仮想関数)
virtual ModulationAndCodingSchemesType	GetLowestModulationAndCoding () const =0	最低 MCS (Modulation and Coding Scheme) の取得 (純粋仮想関数)
virtual void	AddNewStation (const MacAddressType &macAddress, const unsigned int stationBandwidthNumChannels, const bool isHighThroughputStation)=0	宛先通信ノードの追加 (純粋仮想関数)
virtual void	GetDataRateInfoForDataFrameTo Station (const MacAddressType &macAddress, const shared_ptr< RetryCountManager > &retryCountManagerPtr,	データフレーム送信用 MCS の取得 (純粋仮想関数)

	TransmissionParametersType &txParameters)=0	
virtual void	GetDataRateInfoForAckFrameToStation (const MacAddressType &macAddress, const TransmissionParametersType &receivedFrameTxParameters, TransmissionParametersType &ackTxParameters) const =0	ACK フレーム送信用 MCS の取得 (純粋仮想関数)
virtual void	GetDataRateInfoForAckFrameFromStation (const MacAddressType &macAddress, const TransmissionParametersType &sentFrameTxParameters, TransmissionParametersType &ackTxParameters) const =0	ACK フレーム受信時の MCS の取得 (純粋仮想関数)
virtual void	GetDataRateInfoForManagementFrameToStation (const MacAddressType &macAddress, TransmissionParametersType &txParameters) const =0	マネジメントフレーム送信用 MCS の取得 (純粋仮想関数)
virtual void	GetDataRateInfoForBeaconFrame (TransmissionParametersType &txParameters) const =0	ビーコン用の MCS の取得 (純粋仮想関数)
virtual void	NotifyAckReceived (const MacAddressType &macAddress)	ACK 受信成功の通知
virtual void	NotifyAckFailed (const MacAddressType &macAddress)	ACK 受信失敗の通知
virtual void	ReceiveIncomingFrameSinrValue (受信フレームの SINR の取得

	const MacAddressType &sourceMacAddress, const double &measuredSinrValue)	
virtual void	NotifyStartingAFrameTransmissio nSequence (const MacAddressType &macAddress, const bool isTransmittingFrameLongFrame, const size_t numberOfTransmittingMpdus, const shared_ptr< RetryCountManager > &retryCountManagerPtr, const unsigned int retryTxCount=0)	フレーム送信処理開始の通知
virtual void	NotifyFinishingAFrameTransmissi onSequence (const MacAddressType &macAddress, const size_t numberOfTransmissionSucceededM pdus, const shared_ptr< RetryCountManager > &retryCountManagerPtr)	フレーム送信処理終了の通知
virtual void	IsMultirateRetryModel () const	マルチリトライモデルを使用している か否かの識別

ソースファイル: dot11_txpowercontrol.h

5.2.6.AdaptiveTxPowerController

適応送信電力制御モデルの基底クラス

戻り値	関数(引数)	説明
-----	--------	----

	AdaptiveTxPowerController (const ParameterDatabaseReader &theParameterDatabaseReader, const NodeIdType &nodeId, const InterfaceIdType &interfaceId)	AdaptiveTxPowerController クラスの コンストラクタ
virtual double	CurrentTransmitPowerDbm (const MacAddressType &macAddress) const	指定する MAC アドレス用の送信電力 の取得
bool	TxPowerIsSpecifiedByPhyLayer () const	送信電力が PHY レイヤで指定されて いるか否かの識別

5.3. PHY 関連 API

ソースファイル: [dot11_phy.h/cpp](#)

5.3.1.Dot11Phy

Dot11 用 PHY レイヤモデル

戻り値	関数(引数)	説明
	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)	Dot11Phy クラスのコンストラクタ
shared_ptr< Dot11InfoInterface >	GetDot11InfoInterface () const	MAC/PHY 情報インターフェースの作成
bool	IsReceivingAFrame () const	フレームを受信中か否かの識別
bool	IsTransmittingAFrame () const	フレームを送信中か否かの識別
bool	ChannelIsClear () const	チャンネルがクリアか否かの識別

void	TransmitFrame (unique_ptr< Packet > &packetPtr, const TransmissionParametersType &txParameters, const double &transmitPowerDbm, const TimeType &delayUntilAirborne)	フレームの送信
void	TransmitAggregateFrame (unique_ptr< vector< unique_ptr< Packet > > > &aggregatedFramePtr, const TransmissionParametersType &txParameters, const double &transmitPowerDbm, const TimeType &delayUntilAirborne)	アグリゲートフレームの送信
void	TakeOwnershipOfLastTransmitted Frame (unique_ptr< Packet > &framePtr)	最終送信フレームの所有権の取得 (再送用)
bool	LastSentFrameWasAggregate () const	最終送信フレームがアグリゲーション フレームか否かの識別
void	TakeOwnershipOfLastTransmitted AggregateFrame (unique_ptr< vector< unique_ptr< Packet > > > &aggregateFramePtr)	最終送信アグリゲーションフレームの 所有権の取得(再送用)
TimeType	CalculatePhysicalLayerHeaderDur ation (const TransmissionParametersType &txParameters) const	PHY レイヤヘッダ送信時間の算出
TimeType	CalculateFrameDataDuration (const unsigned int frameLengthBytes, const TransmissionParametersType &txParameters) const	データフレームの送信時間の算出
TimeType	CalculateFrameTransmitDuration (const unsigned int frameLengthBytes, const TransmissionParametersType &txParameters) const	フレーム全体の送信時間の算出

TimeType	CalculateAggregateFrameTransmi tDuration (const vector< unique_ptr< ScenSim::Packet > > &aggregateFrame, const TransmissionParametersType &txParameters) const	アグリゲートフレーム全体の送信時間 の算出
TimeType	GetSlotDuration () const	スロット時間の取得
TimeType	GetShortInterframeSpaceDuration () const	SIFS 時間の取得
TimeType	GetRxTxTurnaroundTime () const	Rx-Tx 切り替え時間の取得
TimeType	GetPhyRxStartDelay () const	受信開始までの遅延時間の取得
unsigned int	GetBaseChannelBandwidth () const	基準チャンネルの帯域幅の取得
unsigned int	GetChannelCount () const	チャンネル総数の取得
unsigned int	GetCurrentChannelNumber () const	現在のチャンネル番号の取得
unsigned int	GetCurrentBandwidthNumChanne ls () const	現在のボンディングチャンネル数の取得
unsigned int	GetMaxChannelBandwidthMhz () const	最大チャンネル帯域幅の取得
unsigned int	GetMaxBandwidthNumChannels () const	最大のボンディングチャンネル数の取得
bool	GetIsAHighThroughputStation () const	HT モード通信ノードか否かの識別
const vector< unsigned int > &	GetCurrentBondedChannelList () const	現在のボンディングチャンネルリストの 取得
void	SwitchToChannels (const vector< unsigned int > &bondedChannelList)	チャンネルの切り替え(チャンネルボンディ ング用)
void	SwitchToChannelNumber (const unsigned int channelNumber)	チャンネルの切り替え
double	GetRssiOfLastFrameDbm () const	直近の受信フレームの受信電力 (dBm)の取得
double	GetSinrOfLastFrameDb () const	直近の受信フレームの SINR (dB) の 取得

void	StopReceivingFrames ()	フレームの受信停止
void	StartReceivingFrames ()	フレームの受信開始
bool	IsNotReceivingFrames () const	フレームの受信停止中か否かの識別
const ScenSim::Object MobilityPosition	GetPosition ()	位置の取得
unsigned int	GetNumberOfReceivedFrames () const	受信フレーム数の取得
unsigned int	GetNumberOfFramesWithErrors () const	受信エラー（フレームキャプチャを除く）数の取得
unsigned int	GetNumberOfSignalCaptures () const	フレームキャプチャによる受信エラー数の取得
TimeType	GetTotalIdleChannelTime () const	総チャンネルアイドル時間
TimeType	GetTotalBusyChannelTime () const	総チャンネルビジー時間
TimeType	GetTotalTransmissionTime () const	総送信時間

5.3.2.Dot11MacInterfaceForPhy

Dot11PHY 用 MAC インターフェースの基底クラス

戻り値	関数(引数)	説明
virtual void	BusyChannelAtPhysicalLayerNotification ()	チャンネルビジーの通知 (純粋仮想関数)
virtual void	ClearChannelAtPhysicalLayerNotification ()	チャンネルクリアの通知 (純粋仮想関数)
virtual void	TransmissionIsCompleteNotification ()	送信完了の通知 (純粋仮想関数)
virtual void	DoSuccessfulTransmissionPostProcessing (const bool wasJustTransmitting)	送信の正常終了後の処理 (純粋仮想関数)
virtual void	ReceiveFrameFromPhy (const Packet &aFrame, const TransmissionParametersType &receivedFrameTxParameters)=0	フレームの受信 (純粋仮想関数)
virtual void	ReceiveAggregatedSubframeFromPhy (A-MPDU サブフレームの受信 (純粋仮想関数)

	unique_ptr< Packet > &subframePtr, const TransmissionParametersType &receivedFrameTxParameters, const unsigned int aggregateFrameSubframeIndex, const unsigned int numberSubframes)=0	
virtual void	ReceiveMsdAggregateFrameFromPhy (const vector< unique_ptr< ScenSim::Packet > > &msduAggregateFrame, const TransmissionParametersType &receivedFrameTxParameters)=0	A-MSDU フレームの受信 (純粋仮想関数)
virtual void	NotifyThatPhyReceivedCorruptedFrame ()=0	破損フレームの受信通知 (純粋仮想関数)
virtual void	NotifyThatPhyReceivedCorruptedAggregatedSubframe (const TransmissionParametersType &receivedFrameTxParameters, const unsigned int aggregateFrameSubframeIndex, const unsigned int numberSubframes)=0	破損 A-MPDU サブフレームの受信通知 (純粋仮想関数)
virtual bool	AggregatedSubframesForThisNode (const Packet &frame) const =0	A-MPDU サブフレームが本ノード宛 か否かの識別

6. 参考文献

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.

