

OPEN STM Tips

Pre-Equalization Analyzer

1. 背景・目的

米国 Cable Labs が 2010 年 4 月に公開した”Proactive Network Maintenance Using Pre-equalization”(以降、PNM)の派生技術・製品への関心が高まっている。

本 Tips では、米国 Cable Labs による PNM のリファレンス実装（以降、「PNM RI」）の知財構成、アーカイブの構造、Web UI デモと CLI デモの操作方法、使用上の留意事項について説明する。

2. 対象読者

OSSBN-TIPS-10-08-001 の読者

3. 参考文献・関連文書

Pre-Equalization Analyzer Reference Implementation and Demo Package (Revision: 1.0)

CM-GL-PNMP-V02-110623

OSSBN-TIPS-10-08-001

4. その他

本 Tips 中の図表番号につき、参考文献からの抜粋には原文の番号をそのまま流用し、独自に作成した図表には”Tips-*”の形式で番号を付与した。

5. 最終更新日

2014 年 12 月 26 日

OPEN STM Tips

Pre-Equalization Analyzer

6. 詳細

6.1 PNM RI の知財構成

PNM RI アプリケーションは、プログラム本体（アナライザー&FFT）と各サンプル実装（Web UI デモ、CLI デモ）から構成されている。

プログラム本体は Java、各サンプル実装は Java・HTML・shell・Windows バッチファイル形式で書かれており、米 Cable Labs の公式サイトから非商用版を無償入手できる。

PNM RI の著作権・工業所有権は、原則的に米国 Cable Labs に帰属するが、入手・使用に先立ち、以下の点に注意が必要である。

(1) FFT の実装アルゴリズム（プログラム本体）

PNM RI に含まれる `SpiralFFTfloat32.java` の実装アルゴリズムは、カーネギーメロン大学の `Spiral` プロジェクトで GNU 2.0 に基づき開発された `Spiral 6.0` を部分的に Java に移植したものである。

このため、同 Java クラスを含む PNM RI の一部又は全部を他の既存ソフトに組み込む場合、GNU2.0 のソース開示義務が既存ソフトの非開示部分に及ぶ可能性がある。

GNU との相性の悪い他のソフトに PNM を組み込みたい場合、`SpiralFFTfloat32.java` の他の非 GNU クラスへの置換等、法的な意味での GNU 回避策を検討すべきである。

(2) Web UI デモの開発フレームワーク（サンプル実装）

Web UI デモは、Google の GWT(Google Web Toolkit)により開発されている。GWT は、Java を使ってウェブ用 Ajax アプリケーションを開発できるオープンソースの開発フレームワークであり、Google より Apache License 2.0 に基づきライセンスされている。

PNM と連動させたい既存ソフトが他の Web UI フレームワークで開発されており、GWT との相性が悪い場合、機能面での支障が懸念されるため、非 GWT 環境への移植等、統合先の開発環境に準じた何らかの対策を検討すべきである。

以上の点を除き、PNM RI の利用・応用は、非商用・商用を問わず、米 Cable Labs により利用者に無償許諾されている。但し商用利用時には、`pnm-ri@cablelabs.com` への事前連絡が義務付けられているので、商用利用に際しては、米 Cable Labs への事前連絡が必要な点に十分注意されたい。

OPEN STM Tips

Pre-Equalization Analyzer

6.2 PNM RI アーカイブの構造

6.2.1 PNM RI の開発環境構築&ビルド手順

(1) 必須ツール&アプリケーションのインストール・設定

以下のツール&アプリケーションを開発環境としてインストール・設定する。

- Apache Maven (version: 2.2.x を推奨)
<http://maven.apache.org/download.html>
- Java SDK (Version 1.5 以降を推奨)
<http://java.sun.com/javase/downloads/index.jsp>
- GWT Charting Library, gflot (version 1.0.0)
<http://code.google.com/p/gflot/downloads/list>
- Apache Tomcat, for the web demo (version 6.0 以降)
<http://tomcat.apache.org/>

(2) PNM RI のダウンロード&解凍

Pre-equalizationAnalyzer-RI-DEMO-PACKAGE-v1.0.zip (非商用版 PNM RI アーカイブ) をダウンロード・解凍し、Modules フォルダをローカルディレクトリに配置する。

(3) ビルド手順

GWT 描画ライブラリ(gflot-1.0.0.jar) ファイルを Modules フォルダにコピー後、以下のコマンドを実行する。

```
# mvn install:install-file -DgroupId=ca.nanometrics -DartifactId=gflot -Dversion=1.0.0  
-Dpackaging=jar -Dfile=gflot-1.0.0.jar  
# cp gflot-1.0.0.jar ¥<RI インストール先>/Pre-equalizationAnalyzer-RI-v1.0/Modules  
# cd <RI インストール先>/Pre-equalizationAnalyzer-RI-v1.0/Modules  
# mvn install:install-file -DgroupId=ca.nanometrics -DartifactId=gflot  
-Dversion=1.0.0 -Dpackaging=jar -Dfile=gflot-1.0.0.jar
```

Modules ディレクトリから以下を実行する。

```
# mvn package javadoc:javadoc
```

各 jar ファイル、war ファイル、リソースファイル、Javadoc 等が作成される。

```
../target/Pre-equalizationAnalyzer-1.0.jar  
../target/site/apidocs/index.html  
../PreEqDemoCLI/target/PreEqDemoCLI.jar  
../PreEqDemoWeb/target/PreEqDemoWeb.war
```

文書番号: OSSBN-TIPS-14-12-002/02

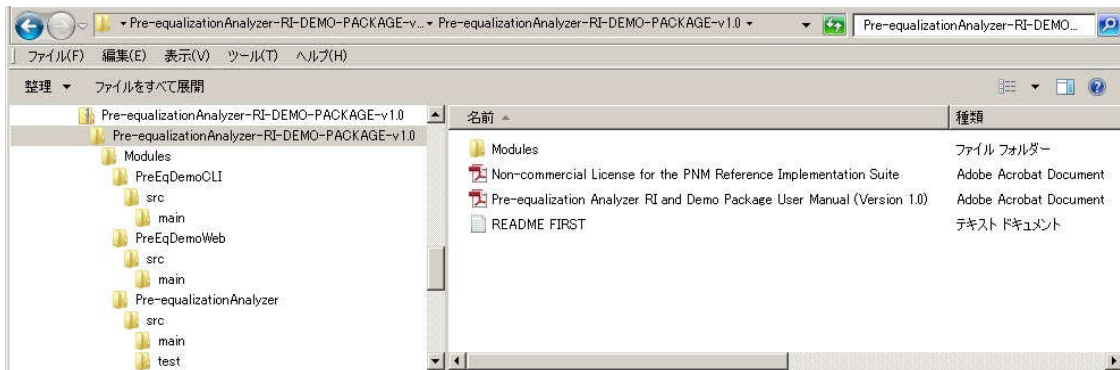
All Rights Reserved, Copyright © OSS BroadNet 2014

OPEN STM Tips

Pre-Equalization Analyzer

6.2.2 PNM RI アーカイブの内容

PNM RI アーカイブの内容を Tips-001 に示す。



Tips-001 PNM RI アーカイブの内容

PNM RI アーカイブは、ソースコードを格納する **Modules** フォルダと以下の 3 文書ファイルから構成される。

- "Non-Commerical License forpdf"
PNM RI 非商用ライセンス規約。
- "Pre-equalization Analyzer RIpdf"
ユーザーマニュアル。インストール&ビルド手順、画面例と簡単な操作説明、FAQ。
- "README FIRST.txt"
PNM RI アーカイブの説明。

Modules フォルダは、以下の 3 サブフォルダから構成される。

- **Pre-equalizationAnalyzer** フォルダ
プログラム本体。
- **PreEqDemoWeb** フォルダ
Web UI デモのサンプル実装。
- **PreEqDemoCLI** フォルダ
CLI デモのサンプル実装。

OPEN STM Tips

Pre-Equalization Analyzer

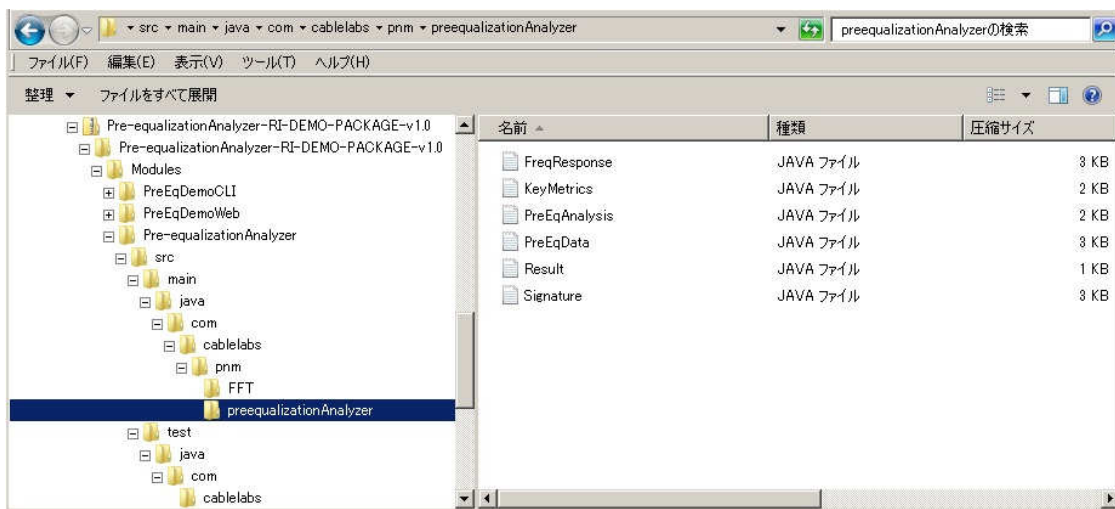
6.2.3 Pre-equalizationAnalyzer フォルダ

Pre-equalizationAnalyzer フォルダは、preequalizationAnalyzer パッケージ (以降、「Analyzer パッケージ」) と FFT パッケージ (以降、「FFT パッケージ」) の各ソースツリーにより構成される。

(1) Analyzer パッケージ

Analyzer パッケージは、PNM RI のプログラム本体である。

Analyzer パッケージの構成を Tips-002 に示す。



Tips-002 Analyzer パッケージの構成

Analyzer パッケージは、6 つの Java クラスにより構成される。

Analyzer パッケージの各 Java クラスの機能概要を Tips-003 に示す。

クラス名	機能概要	備考
FreqResponse	FFT パッケージを使用し、PreEQ データを時間座標から周波数座標に変換・保持。	複素数振幅、ベクトル振幅、群遅延、位相を計算し、呼出元に返す。
KeyMetrics	各キーメトリックス (状態指標は備考を参照) を計算・保持。	MTC, MTE, MTNA, MTNE, NMTER, PPESR, PPTSR, TTE, PostTTE, PostMTTER, PreMTE, PreMTTER
PreEqAnalysis	PI 解析ロジックの Façade クラス。タップ列、FFT タイプ&点数、シンボルレートを設定、PreEqData, FreqResponse, Signature, KeyMetrics を返す。	コンストラクタの SpiralFFT 指定を別アルゴリズムに変更したい場合、呼出元の setFitMethod の引数指定を変更要。
PreEqData	PreEQ の各タップ係数を計算・保持。	タップ形式を自動判断。
Result	メソッドの実行結果オブジェクト。	
Signature	マイクロリフレクションの最大タップ要素の Delay (遅延時間) を計算・保持。NMTER に基づき Severity (即時対応 or 継続観察の要否を判断・保持。	-25dB 以下: noActionRequired -18dB 以上: immediateActionRequired 両者の中間: highMonitoringFrequency

Tips-003 Analyzer パッケージの各 Java クラスの機能概要

OPEN STM Tips

Pre-Equalization Analyzer

1.1CM と 2.0/3.0CM ではタップ形式が異なっているため、Pre-EQ の各タップ係数をプログラムでデコードする場合、各 CM の DOCSIS バージョンに応じて場合分けし、これに基づき各指標の計算式や比較判断基準を切り替える必要がある。

DOCSIS バージョン別・Pre-EQ 対応の違いを Tips-004 に示す。

DOCS ver.	Pre-EQ	タップ形式	TLV
1.0	無	-	-
1.1	有	8 タップ(T, T/2, T/4)	04
2.0/3.0	有	24 タップ(T)	09

Tips-004 DOCSIS バージョン別・Pre-EQ 対応の違い

PreEqData クラスには、SNMP で取得した Pre-EQ データ群の 16 進数文字列に基づき、以下の基準で対象 CM の DOCSIS バージョンを自動判断するアルゴリズムが実装されている。

72 文字の場合 -> DOCSIS 1.1 (8 文字ヘッダー + 8 タップ)

64 文字の場合 -> DOCSIS 1.1 (0 文字ヘッダー + 8 タップ)

200 文字の場合 -> DOCSIS 2.0/3.0 (8 文字ヘッダー + 24 タップ)

192 文字の場合 -> DOCSIS 2.0/3.0 (0 文字ヘッダー + 24 タップ)

72 又は 200 文字の場合、主タップ位置・タップ数・taps/Sym は Pre-EQ のヘッダー文字列から取得・解釈される。

64 又は 192 文字の場合、主タップ位置・タップ数・taps/Sym の根拠となるヘッダー文字列がないため、主タップ位置は Pre-EQ のデータから自動判断、タップ数は 64 文字の場合は 8、192 文字の場合は 24、taps/Sym はいずれも 1 として固定的に処理される。

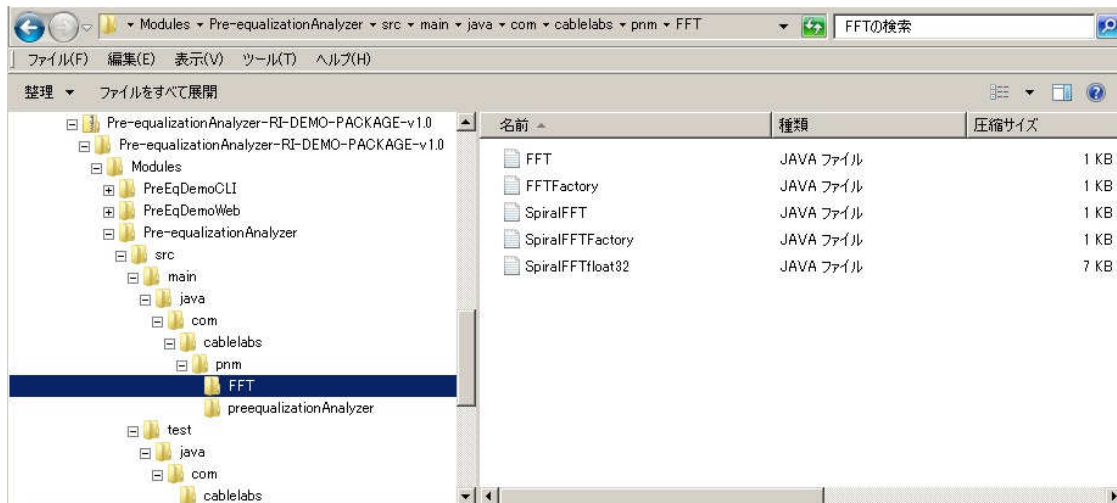
OPEN STM Tips

Pre-Equalization Analyzer

(2) FFT パッケージ

FFT パッケージは、Analyzer パッケージから呼び出される FFT 計算ライブラリである。

FFT パッケージの構成を Tips-005 に示す。



Tips-005 FFT パッケージの構成

FFT パッケージは、5つの Java クラスにより構成される。

FFT パッケージの各 Java クラスの機能概要を Tips-006 に示す。

クラス名	機能概要	備考
FFT	getFFT()メソッドシグニチャ。	インタフェース。
FFTFactory	getFFTFactory()メソッドを定義。 引数指定により SPIRAL、DTF の各アルゴリズムを切替。	抽象クラス。現バージョンでは SPIRAL 指定時のみインスタンス化、DTF 指定時は null を返す。
SpiralFFT	FFT インタフェースの実装クラス。	SpiralFFTfloat32 インスタンスの getSpiralFFTfwd()の実行結果を返す。
SpiralFFTFactory	FFTFactory の継承クラス。	getFFT()呼出時、SpiralFFT インスタンスを返す。
SpiralFFTfloat32	Spiral 6.0 の FFT アルゴリズムの Java 移植。	

Tips-006 FFT パッケージの各 Java クラスの機能概要

GNU2.0 を回避したい場合、ライセンス制約の対象は SpiralFFTfloat32.java に限定されるので、左記クラスのアロリズムを使用しない形にプログラムを改造し、ビルド対象から除外する。

但しこの場合、米 Cable Labs の FFT パッケージソースの改変に相当するため、同団体への事後連絡が必要となる点に注意されたい。

OPEN STM Tips

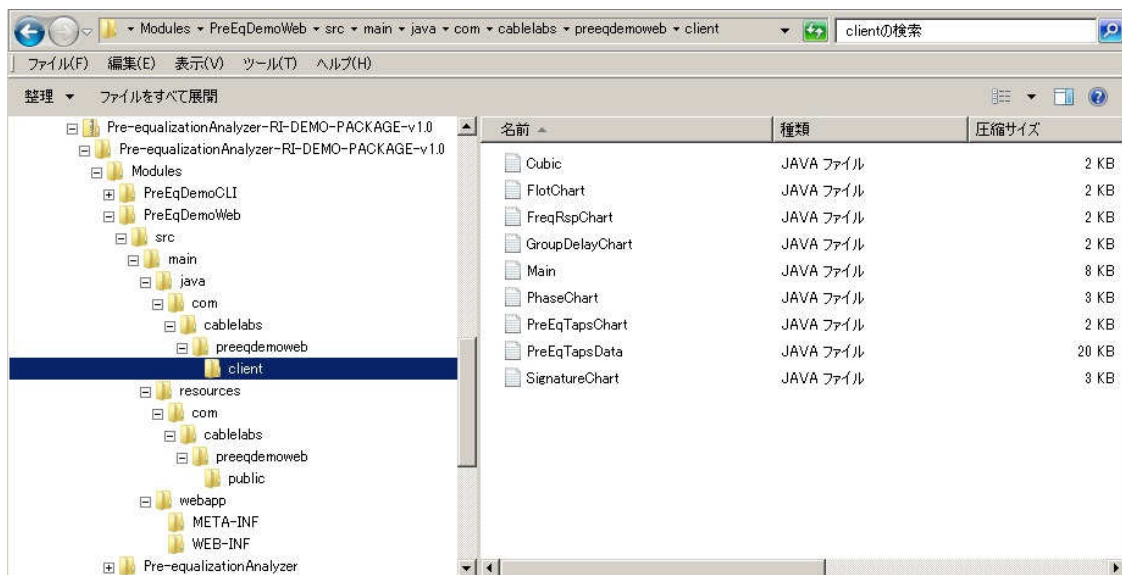
Pre-Equalization Analyzer

6.2.4 PreEqDemoWeb フォルダ

PreEqDemoWeb フォルダは、client パッケージ（以降、「Client パッケージ」）と resources フォルダ（以降、「リソースフォルダ」）により構成される。

(1) Client パッケージ

Client パッケージの構成を Tips-007 に示す。



Tips-007 Client パッケージの構成

クライアントパッケージは、9つの Java クラスにより構成される。

クライアントパッケージの各 Java クラスの機能概要を Tips-008 に示す。

クラス名	機能概要	備考
Cubic	ベジエ曲線、B スプライン関数、エルミート関数等、曲線グラフの描画に使用する各変換行列定義を実装。	
FlotChart	GWT の Composite の継承クラス。 setData()抽象メソッドを定義。	抽象クラス。
FreqRspChart	周波数-レベル特性。	FlotChart を継承。線グラフ。
GroupDelayChart	周波数-群遅延特性。	FlotChart を継承。線グラフ。
Main	GWT の EntryPoint の継承クラス。 クライアントアプリの本体。	SNMP で取得した実データを分析したい場合、本プログラムを修正要。
PhaseChart	周波数-位相特性。	FlotChart を継承。線グラフ。既存の Main クラスでは不使用。
PreEqTapsChart	タップ-エネルギー特性。	FlotChart を継承。棒グラフ。
PreEqTapsData	Web UI デモ用のサンプルデータ。 CM*285 台分。	
SignatureChart	マイクロリフレクション特性。	FlotChart を継承。点グラフ。

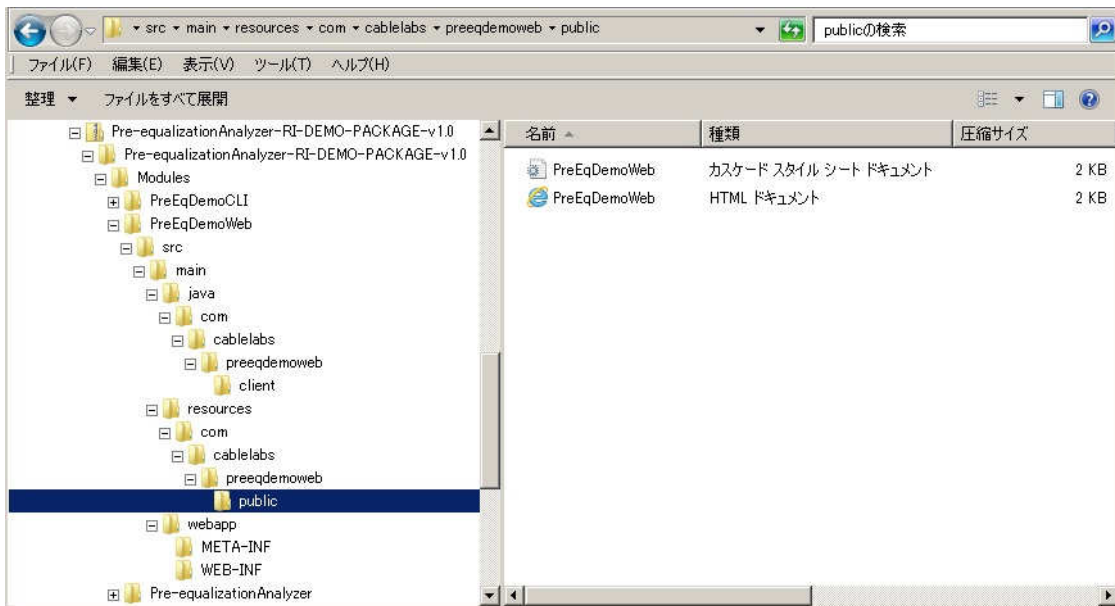
Tips-008 Client パッケージの各 Java クラスの機能概要

OPEN STM Tips

Pre-Equalization Analyzer

(2) リソースフォルダー

リソースフォルダーの構成を Tips-009 に示す。



Tips-009 リソースフォルダーの構成

リソースフォルダーは、2つのファイルにより構成される。

リソースフォルダーの各ファイルの機能概要を Tips-010 に示す。

ファイル名	機能概要	備考
PreEqDemoWeb	Web UI デモ画面の CSS 定義。	CSS ドキュメント
PreEqDemoWeb	Web UI デモ画面の HTML テンプレート。	GWT コンパイル時に置換。

Tips-010 リソースフォルダーの各ファイルの機能概要

OPEN STM Tips

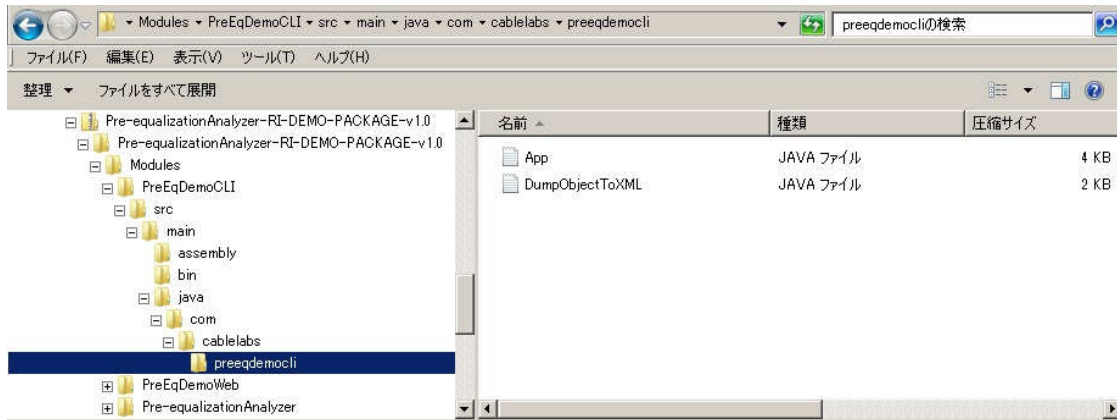
Pre-Equalization Analyzer

6.2.5 PreEqDemoCLI フォルダ

PreEqDemoCLI フォルダのソースは、preeqdemocli パッケージ (CLI デモパッケージ) と bin フォルダ (実行スクリプトフォルダ) により構成される。

(1) CLI デモパッケージ

CLI デモパッケージの構成を Tips-011 に示す。



Tips-011 CLI デモパッケージの構成

CLI デモパッケージは、2つの Java クラスにより構成される。

CLI デモパッケージの各 Java クラスの機能概要を Tips-012 に示す。

クラス名	機能概要	備考
App	クライアントアプリの本体。 CLI デモ用のサンプルデータを含む。 CM*6 台分。	
DumpObjectToXML	PNM 解析結果の XML 出力。	

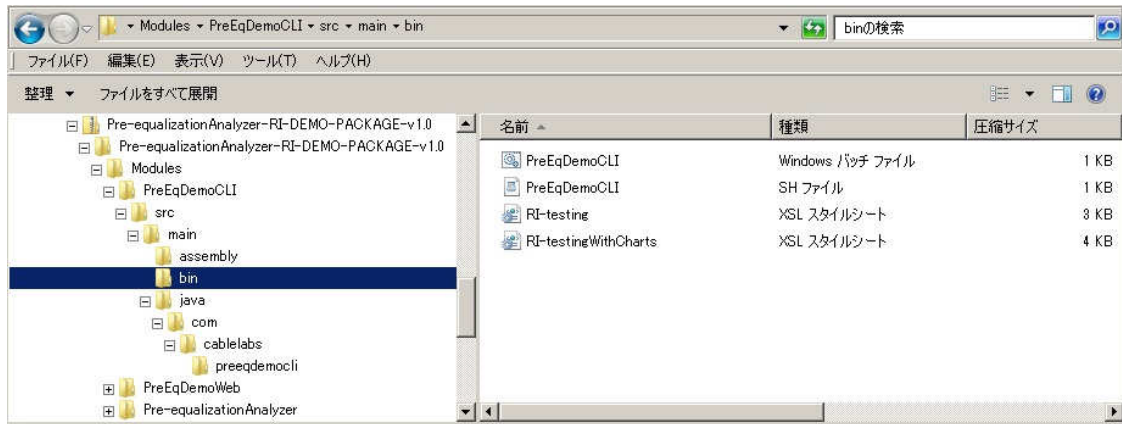
Tips-012 CLI デモパッケージの各 Java クラスの機能概要

OPEN STM Tips

Pre-Equalization Analyzer

(2) 実行スクリプトフォルダー

実行スクリプトフォルダーの構成を Tips-013 に示す。



Tips-013 実行スクリプトフォルダーの構成

実行スクリプトフォルダーは、4つのファイルにより構成される。

実行スクリプトフォルダーの各ファイルの機能概要を Tips-014 に示す。

ファイル名	機能概要	備考
PreEqDemoCLI	Windows 環境で CLI を実行。	Windows バッチファイル
PreEqDemoCLI	Unix 環境で CLI を実行。	Shell ファイル
RI-testing	テスト目的のため説明省略。	XSL スタイルシート
RI-testingWithCharts	同上。	XSL スタイルシート

Tips-014 実行スクリプトフォルダーの各ファイルの機能概要

OPEN STM Tips

Pre-Equalization Analyzer

6.3 Web UI デモの操作方法

6.3.1 Web UI デモアプリケーションのデプロイ

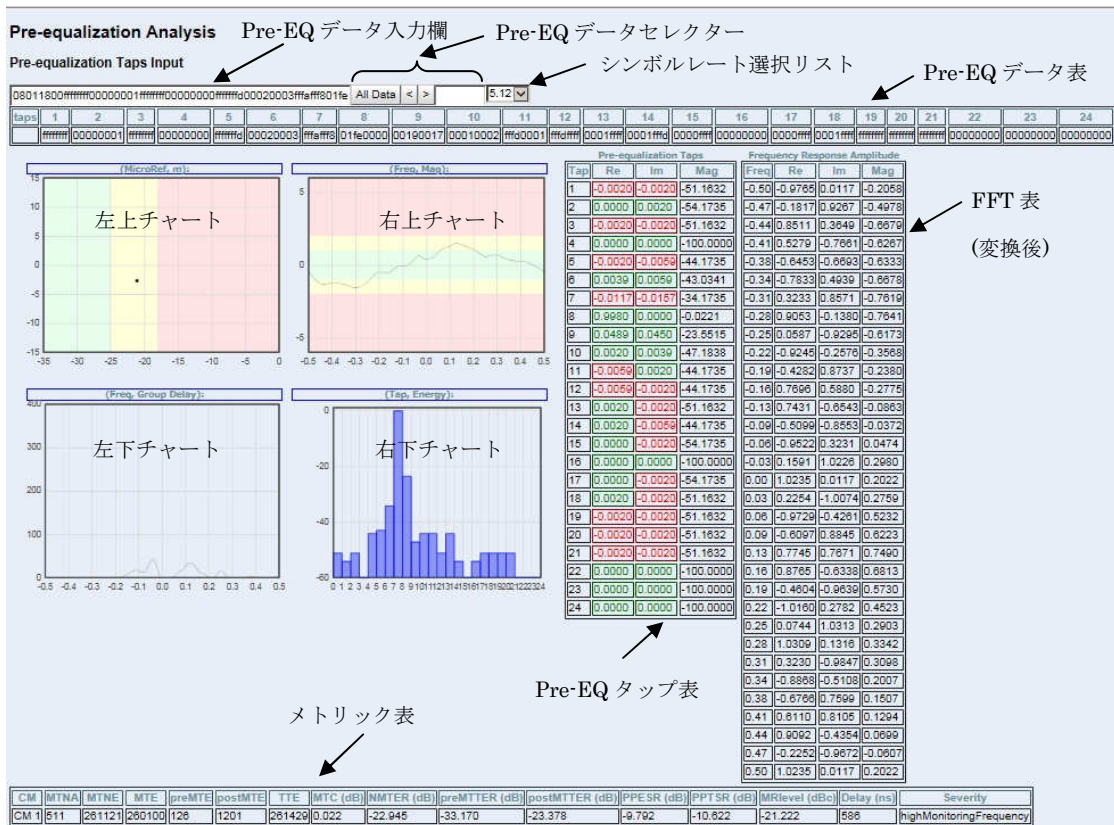
< RI インストール先 >/Modules/PreEqDemoWeb/target の PreEqDemoWeb.war を Apache/Tomcat サーバーの tomcat/webapps ディレクトリに配置する。

6.3.2 ブラウザの起動と URL の指定

ブラウザを起動し、http://<Apache/Tomcat サーバーの IP アドレス>:8080/PreEqDemoWeb を指定する。

6.3.3 Web UI デモのスクリーンショット

単一 CM・単一上り CH の Pre-EQ データに基づく Web UI デモのスクリーンショットを Tips-015 に示す。



Tips-015 Web UI デモのスクリーンショット (単一 CM・単一上り CH)

Web UI デモの画面構成・内容を変更したい場合、6.2.4 に示した PreEqDemoWeb フォルダの内の各クラス・ファイルを修正し、6.2.1(3) に示したビルド手順に従い再ビルド、作成された war ファイルを Apache/Tomcat にデプロイ・再起動する。

OPEN STM Tips

Pre-Equalization Analyzer

6.3.4 操作方法

Web UI デモの各画面部品の操作方法・表示内容の説明・留意事項を以下に示す。

(1) Pre-EQ データ入力欄

画面を開いた初期状態では、PreEqTapsData クラスの ArrayList の第一要素の Pre-EQ データを表示・描画する。(2)の Pre-EQ データセレクト操作により、対象要素の Pre-EQ データに更新する。

Pre-EQ データの直接入力指定により、各画面部品の表示内容を更新できる。これを応用すれば、他のツール (SNMP マネージャー等) で取得した Pre-EQ データ 16 進文字列のコピー&ペーストにより、任意の Pre-EQ データを PNM の視点・基準で解析できる。

(2) Pre-EQ データセレクト

画面を開いた初期状態では、PreEqTapsData クラスの ArrayList の第一要素の Pre-EQ データを表示・描画する。左右移動ボタン押下時、Array List のインデックス順で要素を移動、対象 Pre-EQ データを取得、各画面部品の表示内容を更新する。

All Data ボタン押下時、ArrayList 内の全対象要素の Pre-EQ データを集計する。この時、(7)~(9)は重ね合わせ表示、(4)(5)(6)(10)(11)は集計表示となる。

ブラウザと GWT の相性により、特に All Data ボタン押下時の再描画に時間が掛かる場合がある。ユーザーマニュアル FAQ によると、Apple® Safari や Google® Chrome 等、WebKit に基づくブラウザの方が、描画性能が良い。

インデックスの直接入力指定も一応可能だが、As is では動作が不安定なので推奨しない。

(3) シンボルレート選択リスト

画面を開いた初期状態では 5.12 だが、リスト選択により 2.56 に変更できる。

変更時、(11)に後述する Delay(ns)値が更新される。

計算式上、その他の値・グラフ・メトリックは、本シンボルレート変更の影響を受けない。

なお As is では 1.28 がリストにないため、1.28 を選択する必要がある場合、ソース修正&再ビルドが必要である。

OPEN STM Tips

Pre-Equalization Analyzer

(4) Pre-EQ データ表

ヘッダーを除いた Pre-EQ データをタップ番号毎に区切って表示する。

1.1CM の場合、9 タップ以降の表示が連結され空欄となる。

(5) Pre-EQ タップ表

Re と Im は、Pre-EQ の各タップ値を 16 進 10 進変換後、(11)に後述する MTNA で除した値である。

Mag は、の Re と Im の各自乗和に基づき、1 を 0、0 を-100 として対数変換した主タップ基準の相対 dB 表現であり、PreEqTapsChart クラスの棒グラフ描画に使用される。

1.1CM の場合、9 タップ以降は表示されない。

(6) FFT 表

Pre-EQ タップ表の FFT 変換値であり、FreqResChart, GroupDelayChart クラスの各線グラフ描画に使用される。Freq は、対象上り CH の帯域幅を±0.5 とした相対指標であり、FFT 計算で指定した要素数分の行データが生成・表示される。

なお RI の FFT 表は 32 要素の筈なのに 33 要素あり、参考文献の計算式定義と矛盾するが、本表の各値を運用で直接使用する事はないため、実際の運用には特に支障ない。

(7) 左上チャート

RI では SignatureChart クラスの点グラフを配置している。

X 軸: マイクロリフレクション(dB)。

Y 軸: 不明(単位不明)。

各点が単一の Pre-EQ データを表す。緑色は安全領域、黄色は要観察領域、ピンクは要対処領域を表現し、(11)に後述する Severity に対応付けられている。

米 Cable Labs のガイドライン Figure 25 と表現形式が似ているが、Y 軸のスケールが異なる。また、(11)に後述する Delay との相関も見られず、Y 軸の意味が不明である。

このため、PNM 指針としては X 軸のみに注目し、単一 CM ではなく、対象エリア全体のマイクロリフレクションの影響度合いを視覚的に把握する目的で、(2)の Pre-EQ データセレクターで All Data を指定した状態で使用すると良い。

OPEN STM Tips

Pre-Equalization Analyzer

(8) 右上チャート

RI では FreqResChart クラスの線グラフを配置している。

X 軸: 相対周波数。中心が中心周波数、左端から右端までが帯域幅。

Y 軸: レベル差(dB)。FFT の Mag 値に基づきスムージング描画。

緑色は安全領域、黄色は要観察領域、ピンクは要対処領域を表現している。

本グラフは上を目安に、著しい規定範囲割れがない事を視覚的に把握する目的で使用する。

なお **As is** では、マーカー表示値は **FFT 表** に一致しているが、**Y 軸のスケール表示が狂っている (± 6 ではなく ± 3)** ため、実運用で使う場合、ソース修正&再ビルドが必要である。

(2)の Pre-EQ データセレクトで All Data を指定した状態で使用すれば、複数 CM の共通傾向を視覚的に把握できるため、伝送路の障害源推定に有効である。

(9) 左下チャート

RI では GroupDelayChart クラスの線グラフを配置している。

X 軸: 相対周波数。中心が中心周波数、左端から右端までが帯域幅。

Y 軸: 群遅延(ns)。

DOCSIS3.0PHY 仕様では上り帯域の群遅延特性を、「1MHz 内で特性差が 200nsec 以内」と規定している。

本グラフは上を目安に、著しい規定範囲割れがない事を視覚的に把握する目的で使用する。

(2)の Pre-EQ データセレクトで All Data を指定した状態で使用すれば、複数 CM の共通傾向を視覚的に把握できるため、伝送路の障害源推定に有効である。

なお、(11)に後述する Delay(ns)はマイクロリフレクションの最大要素の遅延時間であり、本チャートが表現する群遅延とは別概念であり、両者間には相関関係がない点に注意されたい。

(10)右下チャート

RI では PreEqTapsChart クラスの棒グラフを配置している。

X 軸: タップ番号。8 タップの場合、0~8 スケールに自動変換される。

Y 軸: 主タップに対する各タップの相対エネルギー(dB)。

本グラフは、主タップに対する各タップへのエネルギー分布度合いを視覚的に把握する目的で使用する。

OPEN STM Tips

Pre-Equalization Analyzer

(11)メトリック表

各列要素の説明と使用方法を Tips-016 に示す。

列要素 (指標)	単位	説明	使用方法
CM	無	CM 番号。RI の As is では、(2)のセレクターで対象 Pre-EQ データを変更しても"CM1"のままとなる (バグ)。	Pre-EQ データの一意特定、他アプリへの画面遷移キーに使用。 運用的には CM-MAC 等に修正すべき。
MTNA	無	Main Tap Nominal Amplitude: 2047, 1023, 511 のいずれか (CM 依存)。	PNM では本値に基づき、各メトリックを相対 dB 値に換算する。 監視業務では使用しない。
MTNE	無	Main Tap Nominal Energy: MTNA の二乗値。主タップの理論上の最大 Mag 値に相当。	例: MTNA=511 時、511*511=261121。 監視業務では使用しない。
MTE	無	Main Tap Energy: MTNA による重み付け後の主タップ Mag 計算値。	監視業務では使用しない。
PreMTE	無	0~主タップ以前の各タップ Mag 値の総和。群遅延の程度を表す。	監視業務では使用しない。
PostMTE	無	主タップ以降の各タップ Mag 値の総和。マイクロリフレクションの程度を表す。	監視業務では使用しない。
TTE	無	Total Tap Energy: 全タップの Mag 値の総和。	監視業務では使用しない。
MTC	dB	Main Tap Compression: CM 取得値は Pre-EQ 補正の度合い、CMTS 取得値は補正不能 (障害) の度合い。	CM 取得値が 2dB を超える場合、Pre-EQ 補正能力超過のため対応要。
NMTER	dB	Non-Main Tap to Total Energy Ratio: 歪成分の程度。Signature クラスによる Severity の判断基準に使用される。	-25dB 以下: 対処不要 -18dB 以上: 即時の対処要 両者の中間: 高頻度監視を推奨
PreMTTER	dB	Pre-Main Tap to Total Energy Ratio: 群遅延対全タップ電力総和比。	同程度値の集中を CAD/Mapping データから抽出、発生箇所を推定。
PostMTTER	dB	Post-Main Tap to Total Energy Ratio: マイクロリフレクション対全タップ電力総和比。	マイクロリフレクションの程度。 品質低下障害の発生状況を把握、即時対処の要否を判断。
PPESR	dB	Pre-Post Energy Symmetry Ratio: 群遅延対マイクロリフレクション比。	群遅延・マイクロリフレクションのいずれが支配因子かの判断指標。
PPTSR	dB	Pre-Post Tap Symmetry Ratio: PPESR の簡易計算表現。	監視業務では使用しない。
MRlevel	dBc	Micro Reflection level: マイクロリフレクション(dB)。	伝送路の劣化度合いの判断指標。
Delay	ns	マイクロリフレクションの最大要素の遅延時間。電磁波が 2 箇所の反射点の往復に掛かる時間を表すため、2 で除した値を電磁波の伝播速度に乗じれば、各反射点間の距離を計算できる。	各反射点間の距離を CAD/Mapping データと照合し、距離が類似の 2 箇所を抽出・推定。 セレクター右横のシンボルレートを変えると値が変わる。
Severity	無	Signature クラスによる判断結果の文字列表現。例: no action required.(対処不要) 等	アラームイベントとして使用。 (画面操作の開始契機 等)

Tips-016 メトリック表の各項目の説明と使用方法

MTC~MRlevel の各メトリック値は、時系列分析による経年劣化判断の根拠として有効である。

文書番号: OSSBN-TIPS-14-12-002/02

All Rights Reserved, Copyright © OSS BroadNet 2014

OPEN STM Tips

Pre-Equalization Analyzer

6.4 CLI デモの操作方法

6.4.1 CLI デモアプリケーションの起動と各引数の指定

../PreEqDemoCLI/target/PreEqDemoCLI.jar を実行する。

```
# java -jar PreEqDemoCLI-1.0-jar-with-dependencies.jar -?
```

“-?” には、以下の各引数を指定する。

- 1 ~ -6 : CLI デモパッケージの App クラスに含まれる 6CM のサンプルデータを使用。
- d : 全 PNM 計算結果・メトリック・シグニチャを XML 形式でダンプ出力。
- e <arg> : Pre-EQ データ(16 進表現文字列)を指定、メトリック・シグニチャの一部を出力。
- s <arg> : シンボルレートを指定 (MHz。デフォルト: 5.12)。
- x <arg> : “d”出力に使用するスタイルシートファイルを指定。

OPEN STM Tips

Pre-Equalization Analyzer

6.4.2 CLI デモの実行例

CLI デモの実行例を以下に示す。

<<-1 オプション指定時 >>

```
# java -jar PreEqDemoCLI-1.0-jar-with-dependencies.jar -1
```

Analysis for:

```
08011800ffffffff00000001ffffffff00000000ffffffffd00020003ffaaff801fe00000019001700010002fffd  
0001ffdffff0001ffff0001ffd0000ffff000000000000ffff0001ffffffffffffffffffffffff00000000000000  
0000000000
```

Computation took: 7 milliseconds for 1 iterations.

microreflection=-21.221525 delay=585.9375 severity=highMonitoringFrequency

<<-e オプション指定時 >>

```
# java -jar PreEqDemoCLI-1.0-jar-with-dependencies.jar -e
```

```
08011800ffffffff00000001ffffffff00000000ffffffffd00020003ffaaff801fe00000019001700010002fffd  
0001ffdffff0001ffff0001ffd0000ffff000000000000ffff0001ffffffffffffffffffffffff00000000000000  
0000000000
```

Analysis for:

```
08011800ffffffff00000001ffffffff00000000ffffffffd00020003ffaaff801fe00000019001700010002fffd  
0001ffdffff0001ffff0001ffd0000ffff000000000000ffff0001ffffffffffffffffffffffff00000000000000  
0000000000
```

Computation took: 7 milliseconds for 1 iterations.

microreflection=-21.221525 delay=585.9375 severity=highMonitoringFrequency

6.4.3 参考情報

CLI デモアプリケーションは、Windows 環境でも実行可能である。

2014 年 12 月現在、Java version “1.8.0_25”での動作が確認されている。

OPEN STM Tips

Pre-Equalization Analyzer

6.5 使用上の留意事項

6.5.1 PNM RI の使い方

6.2.4 に示した PreEqDemoWeb フォルダのクライアントパッケージ中、Main.java のソースコードを修正し、PreEqTapsData.java のサンプルデータでなく、SNMP マネージャーの取得値を PNM への画面遷移時にオブジェクト渡しする形に変更すれば、動作検証目的に必要な十分な水準の機能統合と、GWT ベースでの改良が可能である。

本格的に独自の Web UI を設計・開発したい場合、6.2.3(1)に示した Analyzer パッケージの API を使用する開発環境から呼び出す形式で、PreEqDemoWeb フォルダのクライアントパッケージの各ソースコードを参考にしながらアプリケーションを実装すれば良い。

逆に、一切のプログラミング作業を避け、とにかく PNM の各キーマトリック値を計算したいだけの場合（例: CMTS/CM 機種の変換性検証 等）、既存 SNMP マネージャーの取得値をコピー&ペースト等して引数に指定しながら、6.4 に示した CLI デモを Windows または Linux O/S のコマンドコンソールから実行すれば良い。

6.5.2 履歴データの保存指針

PNM で収集した各 Pre-EQ データと各メトリック指標の時系列分析により、線形歪成分を基準とした伝送路の劣化度合いの進行の定量的な把握が可能となる。

但し、経年劣化の把握を目的とした時系列分析には、Tips-015 に示した Web UI デモ画面の各チャート・表部品は不要であり、Tips-016 に示す各メトリックの一部指標だけで事足りる。

一方で Pre-EQ データはレコードサイズが比較的大きく、監視対象とする CM 台数が多い場合、データベース構造や HDD 容量によっては長期のレコード保持が困難となり、保存対象のデータを必要最小限に絞るための検討・工夫が求められる場合があり得る。

更に、CM に加えて CMTS の Pre-EQ データの双方を取得・長期保存すると、CMTS への SNMP 処理負荷上昇とレコードサイズの更なる増加を招くため、システム設計に際しては、CMTS の Pre-EQ データの履歴保存の要否についても十分に検討すべきである。

コストパフォーマンス上の観点からは、CM の MTC, NMTER, PreMTTER, PostMTTER, MRlevel メトリック指標のみ履歴として保持、かつ CM の Pre-EQ データを描画用にバイナリ形式で保存、一方で CMTS の Pre-EQ データは元来が即時の対応が必要な顕著な品質劣化指標であり時系列分析の必然性に乏しいため保存対象から除外する方針を推奨する。

OPEN STM Tips

Pre-Equalization Analyzer

6.5.3 インピーダンス特性劣化の発生箇所推定

同軸ケーブルの中では、電磁波は光速の約 87%相当で伝播する。

光速を299,792,458 m/sとすると、電磁波伝播速度は約260,819,438 m/sとなる。

本定数と PNM から得られる Tips-016 の Delay 値を以下の計算式に代入すれば、支配的なマイクロリフレクションの原因となっている 2 箇所の反射点間の線路距離を推定できる。

各反射点間の距離 (m) = Delay (ns) / 10exp9 * 同軸中の電磁波伝播速度 (m/s) / 2

実際の伝送路では、幹線・分配線系統の伝送機器が複数個所で同時に大きく劣化するケースは稀であり、インピーダンス特性劣化の原因の過半は加入者宅内配線に起因するが多い。

反射現象の一端が加入者宅内の場合、上述の計算式は「加入者宅からCMTS方向に遡る形での線路距離」を表すため、PNMを有効活用したい場合、ノードアンプやタップオフ等の伝送機器間の線路距離データに加え、各加入者宅の引込線長データが比較的有效である。

OPEN STM Tips

Pre-Equalization Analyzer

6.5.4 複数 DOCSIS バージョンの混在環境

(1) タップ数差の影響

DOCSIS2.0/3.0 では、タップ間隔は対象変調方式のシンボルレートに等しく、主信号には一般的に 24 タップ中の 8 タップ目が使われるので、Pre-EQ 回路が生成できる最大遅延時間は $16T$ (T はシンボルレート) となる。

一方で DOCSIS1.1 では、タップ間隔はシンボルレート T の約数であり、 $T, T/2, T/4$ のいずれかとなり、主信号には 8 タップ中の 4 タップ目が使われる場合が多いので、最大遅延時間は最大タップ間隔 $T \times$ 残 4 タップ = $4T$ となる。

Table 1 に、DOCSIS1.1($4 \times T$ 列)と 2.0 以降($16T$ 列)における、Pre-EQ フィルター回路が対応できる最大遅延時間を示す。

Table 1 - Maximum Delays Generated by Pre-equalization Filter Structures in DOCSIS 1.1 and 2.0

Symbol Rate MHz	Symbol Period (T) μ sec	$4 \times T$ μ sec	$16 T$ μ sec
5.12	0.195		3.125
2.56	0.391	1.563	6.250
1.28	0.781	3.125	12.500

PNM RI の原理上、タップ間隔の差は分解能に、8 と 24 のタップ数差は測定レンジ (検知能力) に影響する。

1.1CM の場合、時定数の長いマイクロリフレクションが、補正対象外=タップ化範囲外となり、Pre-EQ メトリックの計算対象から外れる。この結果、同一エリアに 1.1CM と 2.0/3.0CM が混在し、共に時定数の長いマイクロリフレクションの影響を受けている系では、1.1CM では PNM メトリック値が正常で SNR が異常、一方で 2.0/3.0CM では PNM メトリック値が異常で SNR が正常なケースが稀に発生し得る。

このため PNM は、オフライン判断のような正常 CM による消し込み処理は行わず、1 台でも疑わしい CM があれば NG と見なすルールで運用すべきである。

なお、上述のようなケースが生じた場合、RI の Web UI デモで右下グラフとして配置されている PreEqTapsChart クラスの棒グラフによる 2.0/3.0CM のタップ値確認が有効である。本ケースの場合、PreEqTapsChart クラスの棒グラフでは、マイクロリフレクションの支配因子タップが、1.1CM の補正範囲外の右側箇所に現れる。

OPEN STM Tips

Pre-Equalization Analyzer

(2) 3.0CM の上り複数チャンネル構成

3.0CM の上り複数チャンネル構成時、各 CM は上りチャンネル単位に Pre-EQ 情報を持つため、PNM 解析は CM の各上りチャンネル単位に行う必要がある。

Table 2 に、DOCSIS2.0 と 3.0 の Pre-EQ データの MIB 定義を示す。

Table 2 - DOCSIS 2.0 and 3.0 Transmit Pre-equalization MIBs

	DOCSIS 2.0 DOCS-IF-MIB [RFC4546]		DOCSIS 3.0 DOCS-IF3-MIB [DOCSISv3.0]	
	MIB Table	MIB Object	MIB Table	MIB Object
CM	docsIfCmStatusTable Single upstream channel	docsIfCmStatusEqualizationData	docsIf3CmStatusUsTable Per configured upstream channel	docsIf3CmStatusUsEqData
CMTS	docsIfCmtsCmStatusTable Single instance per CM	docsIfCmtsCmStatusEqualizationData	docsIf3CmtsCmUsStatusTable Per CM, per configured upstream channel	docsIf3CmtsCmUsStatusEqData

DOCSIS 3.0 では、同一ノード配下での CH セット混在が可能であり、各 CM がそれぞれ異なる CH セットで動作できるため、ifIndex による SNMP 取得&単純照合では、各 Pre-EQ データの周波数設定が異なる場合が発生し得る。このためアプリケーションの処理構造設計に際しては、DOCSIS3.0 の上りチャンネル構成の取り扱いに注意が必要である。

なお DOCSIS バージョンによらず、特に群遅延は上りチャンネル周波数に強い相関があるため、上りチャンネル周波数が異なる複数 CM の Pre-EQ 値を比較すると、グループ判定の合理性が失われ、正しい判定結果を期待できなくなる。

このため、CM から Pre-EQ データを SNMP 取得する場合、中心周波数とチャンネル帯域幅を Pre-EQ データと合わせて取得すると共に、中心周波数&チャンネル帯域幅を同一条件で比較・照合できるよう、レコードの構造定義や、表・グラフ部品に使用するデータの抽出・絞り込み条件等に注意すべきである。

以上