PureFlow WS1

トラフィックシェーパー
NF7500 シリーズ
コンフィギュレーションガイド

第7版

・製品を適切・安全にご使用いただくために，製品をご使用になる前に，本書を必ずお読みください。
・本書に記載以外の各種注意事項は，取扱説明書（NF7500-W011J）に記載の事項に準じますので，そちらをお読みください。
・本書は製品とともに保管してください。

アンリツ株式会社

管理番号：NF7500-W013J-7.0

安全情報の表示について

当社では人身事故や財産の損害を避けるために，危険の程度に応じて下記のようなシグナルワードを用いて安全に関する情報を提供しています。記述内容を十分理解して機器を設置および操作するようにしてください。
下記の表示およびシンボルは，そのすべてが本器に使用されているとは限りません。また，外観図などが本書に含まれるとき，製品に貼り付けたラベルなどがその図に記入されていない場合があります。

本書中の表示について

⚠ 危険	回避しなければ，死亡または重傷に至る切迫した危険があることを示します。
⚠ 警告	回避しなければ，死亡または重傷に至る恐れがある潜在的な危険があることを示します。
⚠ 注意	回避しなければ，軽度または中程度の人体の傷害に至る恐れがある潜在的危険，または，物的損害の発生のみが予測されるような危険があることを示します。

機器に表示または本書に使用されるシンボルについて

機器の内部や操作箇所の近くに，または本書に，安全上および操作上の注意を喚起するための表示があります。
これらの表示に使用しているシンボルの意味についても十分理解して，注意に従ってください。

⃠	禁止行為を示します。丸の中や近くに禁止内容が描かれています。
○	守るべき義務的行為を示します。丸の中や近くに守るべき内容が描かれています。
△	警告や注意を喚起することを示します。三角の中や近くにその内容が描かれています。
□	注意すべきことを示します。四角の中にその内容が書かれています。

PureFlow WS1
トラフィックシェーパー　NF7500 シリーズ
コンフィギュレーションガイド

2017年（平成29年）５月31日（初　版）
2021年（令和３年）10月25日（第７版）

・予告なしに本書の内容を変更することがあります。
・許可なしに本書の一部または全部を転載・複製することを禁じます。
Copyright © 2017-2021, ANRITSU CORPORATION
Printed in Japan

当社へのお問い合わせ

本製品については，安全マニュアルに記載の「本製品についてのお問い合わせ窓口」へご連絡ください。

保守契約について

保守契約を結んでいただくと種々のサービスを受けることが可能です。保守契約の詳細については，ご購入いただいた販売店にお問い合わせください。

日本国外持出しに関する注意

本製品および添付マニュアル類は，輸出および日本国外持ち出しの際には，「外国為替及び外国貿易法」により，日本国政府の輸出許可や役務取引許可を必要とする場合があります。また，米国の「輸出管理規則」により，日本からの再輸出には米国政府の再輸出許可を必要とする場合があります。
本製品は日本国以外の安全規格などに準拠していない場合があります。
本製品や添付マニュアル類を輸出または日本国外持ち出しする場合は，事前に必ず弊社の営業担当までご連絡ください。
輸出規制を受ける製品やマニュアル類を廃棄処分する場合は，軍事用途等に不正使用されないように，破砕または裁断処理していただきますようお願い致します。

商標・登録商標

Windows および Windows Server，Active Directory は，米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。
OpenFlow は，Open Networking Foundation の商標または登録商標です。

本書の内容

この取扱説明書は，PureFlow WS1 トラフィックシェーパー（以下，本装置）で動作するソフトウェアの設定方法と使用方法を説明します。本装置を設置，導入，管理を行うネットワーク管理者を対象としています。インターネットワーキングに対する以下のような基礎知識を持った読者を想定しています。

- ローカルエリアネットワーク（LAN）
- Ethernet
- インターネットプロトコル（IP）

本説明書が適用できる本装置の形名を下記に示します。

- NF7501A

本装置の取扱説明書は，以下の①～④で構成されています。本書は③です。

① **取扱説明書**（NF7500-W011J）
この説明書は，本装置の設置および取り扱いについて記述してあります。

② **コマンドリファレンス**（NF7500-W012J）
この説明書は，本装置で使用するコマンドの詳細について記述してあります。

③ **コンフィギュレーションガイド**（NF7500-W013J）
この説明書は，本装置の持つ基本的な機能およびその機能を使ってネットワークを構築する際の具体的な設定例について記述してあります。

④ **WebGUI 操作説明書**（NF7500-W014J）
この説明書は，ネットワーク接続した端末の Web ブラウザを利用して，本装置の設定や表示を行うための操作方法について記述してあります。

また，本製品に関連する下記文書または機能に関する文書が発行された場合，必ずご一読ください。

リリースノート

（リリースノートの発行については，ご購入いただいた販売店にお問い合わせください）

目次

本書の内容..... I

第 1 章　ソフトウェアの概要..... 1-1

第 2 章　基本機能説明 2-1

2.1　トラフィックコントロール機能..... 2-2
2.2　リンクダウン転送機能 2-2
2.3　SSH 機能..... 2-2
2.4　Simple Network Management Protocol（SNMP）機能 ... 2-2
2.5　統計情報..... 2-2
2.6　RADIUS 機能..... 2-2
2.7　WebAPI 機能 2-3
2.8　WebGUI 機能..... 2-3
2.9　OpenFlow 機能 2-3
2.10　ネットワークバイパス機能..... 2-3
2.11　トップカウンタ機能 2-3
2.12　ドメインフィルタ機能..... 2-3
2.13　トラフィック分析機能..... 2-3

第 3 章　設定の基本..... 3-1

3.1　Command Line Interface（CLI）..... 3-2
3.2　コマンド構造の説明..... 3-3
3.3　コマンドシンタックス 3-4
3.4　ヘルプ機能 3-5
3.5　コマンドの省略形と補完 3-5
3.6　ヒストリ機能 3-6
3.7　コマンド編集機能..... 3-7
3.8　ページャ機能 3-8
3.9　起動とログイン 3-9
3.10　設定の保存方法..... 3-11
3.11　設定のリストア方法 3-11
3.12　装置の起動時間..... 3-12

第 4 章　装置本体の情報表示と設定 4-1

4.1　日付／時刻..... 4-2
4.2　Simple Network Time Protocol（SNTP） 4-4
4.3　ユーザ名とパスワード..... 4-5
4.4　SYSLOG..... 4-6
4.5　モジュール情報..... 4-9
4.6　ライセンスキー 4-11

第 5 章　Ethernet ポートの設定 5-1

第 6 章　Network ポートの設定..... 6-1

6.1　概要..... 6-2
6.2　メディアタイプの設定..... 6-4
6.3　Network ポートの属性の設定..... 6-5
6.4　最大フレーム長の設定..... 6-7
6.5　設定，状態の確認 6-9

第 7 章　システムインタフェースの設定 7-1

7.1　概要..... 7-2
7.2　システムインタフェース通信..... 7-3
7.3　システムインタフェースフィルタ..... 7-12
7.4　コンフィギュレーション例..... 7-13
7.5　設定，状態の確認 7-20

第 8 章　トラフィックコントロール機能 8-1

8.1　概要..... 8-2
8.2　トラフィックシェーピング 8-4
8.3　トラフィックアクセラレーション..... 8-5
8.4　大規模ネットワークへの適用 8-6
8.5　チャネル 8-7
8.6　シナリオ..... 8-9
8.7　階層化シナリオ 8-13
8.8　アクセラレーショントンネル 8-18
8.9　設定方法..... 8-20
8.10　ルールリストの設定方法 8-41

8.11 チャネルインタフェース通信 8-44
8.12 アプリケーション高速化機能 8-47
8.13 コンフィギュレーション例 8-51
8.14 さらに高度な設定 8-65
8.15 トラフィックアクセラレーション実行時のアドレス 8-104

第 9 章 リンクダウン転送機能 9-1

9.1 リンクダウン転送機能 9-2

第 10 章 SSH 機能 10-1

10.1 概要 10-2
10.2 仕様一覧 10-3
10.3 SSH の利用方法 10-4

第 11 章 SNMP の設定 11-1

11.1 SNMP の概要 11-2
11.2 SNMPv1/SNMPv2c の設定 11-3
11.3 SNMPv3 の設定 11-5
11.4 TRAP の設定 11-7

第 12 章 統計情報 12-1

12.1 ポート統計情報 12-2
12.2 シナリオ統計情報 12-3

第 13 章 RADIUS 機能 13-1

13.1 概要 13-2
13.2 ログイン認証の制御 13-3
13.3 ログインモードの制御 13-3
13.4 RADIUS 機能の設定 13-4
13.5 RADIUS サーバの設定 13-6

第 14 章 ダウンロードとアップロード..... 14-1

14.1 ソフトウェアのダウンロード／アップロード 14-2
14.2 ソフトウェアアップデートパッチの適用..... 14-6
14.3 コンフィギュレーションのダウンロード／アップロード..... 14-7
14.4 ソフトウェアを再起動する 14-11

第 15 章 WebAPI 機能..... 15-1

15.1 概要..... 15-2
15.2 通信プロトコル 15-3
15.3 HTTP メソッド..... 15-3
15.4 JSON 形式..... 15-4
15.5 API 一覧 15-5
15.6 共通エラーメッセージ 15-6
15.7 エラーメッセージ一覧 15-7

第 16 章 OpenFlow 機能 16-1

16.1 概要..... 16-2
16.2 OpenFlow バージョン 16-3
16.3 OpenFlow 対応メッセージ..... 16-4
16.4 CLI コマンド対応 OpenFlow メッセージ 16-6
16.5 JSON 形式..... 16-7
16.6 対応コマンド一覧..... 16-8
16.7 共通エラーメッセージ 16-9
16.8 エラーメッセージ一覧 16-10

第 17 章 ネットワークバイパス機能 17-1

17.1 概要..... 17-2
17.2 設定と確認方法 17-3
17.3 注意事項..... 17-6

第 18 章 トップカウンタ機能 18-1

18.1 概要 18-2
18.2 トップカウンタの表示単位について 18-2
18.3 トップカウンタの測定範囲について 18-3
18.4 トラフィックカウンタについて 18-4
18.5 アプリケーションポート番号の測定について 18-5
18.6 操作コマンド一覧 18-5
18.7 操作手順 18-6
18.8 操作例 18-7
18.9 注意事項 18-9

第 19 章 ドメインフィルタ機能 19-1

19.1 概要 19-2
19.2 ドメインフィルタ機能仕様について 19-3
19.3 設定手順 19-5
19.4 確認手順 19-7
19.5 DNS レスポンスパケット装置不通過の場合の対応 19-9
19.6 注意事項 19-10

第 20 章 トラフィック分析機能 20-1

20.1 概要 20-2
20.2 トラフィック分析の測定項目 20-3
20.3 トラフィック分析の集計方法 20-6
20.4 トラフィック分析の設定方法 20-9
20.5 トラフィック生成機能 20-16
20.6 注意事項 20-21

付録A デフォルト値 A-1

付録B SYSLOG 一覧 B-1

付録C SNMP Trap 一覧 C-1

付録D Enterprise MIB 一覧 D-1

付録 E JSON の記述方法 E-1

付録 F　WebAPI 詳細..... F-1

付録 G　WebAPI サンプルプログラム..... G-1

付録 H　CLI コマンド対応
OpenFlow メッセージ詳細..... H-1

付録 I　OpenFlow メッセージ詳細..... I-1

（空白ページ）

第1章　ソフトウェアの概要

ここでは，本装置ソフトウェアの概要について説明します。

基本機能を以下に列記します。

- トラフィックコントロール機能
- リンクダウン転送機能
- SSH 機能
- Simple Network Management Protocol（SNMP）機能
- 統計情報
- RADIUS 機能
- WebAPI 機能
- WebGUI 機能
- OpenFlow 機能
- ネットワークバイパス機能
- トップカウンタ機能
- ドメインフィルタ機能
- トラフィック分析機能

（空白ページ）

第2章　基本機能説明

ここでは，本装置ソフトウェアの基本機能について説明します。

2.1　トラフィックコントロール機能 2-2
2.2　リンクダウン転送機能 2-2
2.3　SSH 機能 2-2
2.4　Simple Network Management Protocol（SNMP）機能　2-2
2.5　統計情報 2-2
2.6　RADIUS 機能 2-2
2.7　WebAPI 機能 2-3
2.8　WebGUI 機能 2-3
2.9　OpenFlow 機能 2-3
2.10　ネットワークバイパス機能 2-3
2.11　トップカウンタ機能 2-3
2.12　ドメインフィルタ機能 2-3
2.13　トラフィック分析機能 2-3

2.1　トラフィックコントロール機能

音声通信や TV 会議などのミッションクリティカルな業務は，回線帯域不足によるパケット消失や通信遅延が発生すると業務効率を低下させ，重大な支障をきたすことにつながります。このようなミッションクリティカルなトラフィックを回線帯域不足や通信遅延から守るために，回線帯域を拠点やユーザ，またはアプリケーションごとに分割し，必要な帯域を割り当てたり，トラフィックの優先制御を行う必要があります。本装置は，ネットワークの通信経路上に設置され，回線帯域を分割し，割り当てた帯域に対して最低帯域を保証したり，最大帯域制限を行うなどのトラフィックシェーピングを行うことができます。

また，今日のデータセンタは，機器コストや運用コストを削減し，かつ，セキュリティを強化するため，データセンタにサーバとストレージをセンタ集中型に配置する需要が高まっています。一方で，災害時においてもサーバのデータを確実に復旧するためにバックアップデータを遠隔に転送する需要が高まっています。しかし，多くのデータ通信で使用する TCP/IP は，回線遅延により転送速度が低下します。本装置のトラフィックアクセラレーション機能は，回線遅延により影響を受ける TCP/IP の転送性能を向上し，高速なデータ通信を実現します。また，トラフィックシェーピング機能を使用することでネットワークにおけるパケット廃棄を防ぎ，TCP/IP の転送速度を向上することができます。

トラフィックコントロール機能についての詳細な説明は，「第 8 章 トラフィックコントロール機能」を参照してください。

2.2　リンクダウン転送機能

一方のリンクのダウンを検出すると他方のリンクをダウンさせ，リンク異常を通知することができます。

リンクダウン転送機能についてのさらに詳細な説明は「第 9 章 リンクダウン転送機能」を参照してください。

2.3　SSH 機能

SSH サーバ機能により，本装置と SSH クライアント間の通信が暗号化され，安全性が保証されていないネットワークを経由する場合でも，セキュアな遠隔操作が可能になります。

SSH 機能についてのさらに詳細な説明は「第 10 章 SSH 機能」を参照してください。

2.4　Simple Network Management Protocol（SNMP）機能

SNMP は，ルータやサーバなどのネットワーク機器に対してネットワークを通してリモートで管理するためのプロトコルです。

SNMP 機能についてのさらに詳細な説明は「第 11 章 SNMP の設定」を参照してください。

2.5　統計情報

各カウンタ，キューバッファ情報などの統計情報があります。

統計情報についてのさらに詳細な説明は「第 12 章 統計情報」を参照してください。

2.6　RADIUS 機能

RADIUS 機能は，TELNET，SSH，およびシリアルコンソールのログイン時に，RADIUS（RFC2865）を使用してユーザ認証する機能です。

RADIUS についてのさらに詳細な説明は「第 13 章 RADIUS 機能」を参照してください。

2.7 WebAPI 機能

WebAPI 機能は，本装置のトラフィックコントロール機能の設定を行う際に，HTTP（Hypertext Transfer Protocol：RFC2616）を使用して設定を行う機能です。

WebAPI についてのさらに詳細な説明は「第 15 章 WebAPI 機能」を参照してください。

2.8 WebGUI 機能

WebGUI 機能は，ネットワーク接続した端末の Web ブラウザを利用して，本装置の設定や表示を行う機能です。

WebGUI についてのさらに詳細な説明は「WebGUI 操作説明書（NF7500-W014J）」を参照してください。

2.9 OpenFlow 機能

OpenFlow 機能は，本装置のトラフィックコントロール機能の設定を行う際に，OpenFlow プロトコルを使用して設定を行う機能です。

OpenFlow 機能についてのさらに詳細な説明は，「第 16 章 OpenFlow 機能」を参照してください。

2.10 ネットワークバイパス機能

本装置は，Network ポートのバイパス機能を実装しています。装置異常発生時に Network ポートをバイパスして，通信経路を確保することができます。

ネットワークバイパス機能についてのさらに詳細な説明は，「第 17 章 ネットワークバイパス機能」を参照してください。

2.11 トップカウンタ機能

トップカウンタ機能は，トラフィックの利用状況を把握するための機能です。

トップカウンタ機能についてのさらに詳細な説明は，「第 18 章 トップカウンタ機能」を参照してください。

2.12 ドメインフィルタ機能

ドメインフィルタ機能は，帯域制御のパケット分類識別子としてドメイン名を使用することができる機能です。

ドメインフィルタ機能についてのさらに詳細な説明は，「第 19 章 ドメインフィルタ機能」を参照してください。

2.13 トラフィック分析機能

トラフィック分析機能は，ネットワークのパケット損失、転送遅延やサーバクライアントの負荷状況などを測定する機能です。

トラフィック分析機能についてのさらに詳細な説明は，「第 20 章 トラフィック分析機能」を参照してください。

（空白ページ）

第3章　設定の基本

ここでは，設定の基本について説明します。

3.1　Command Line Interface（CLI） 3-2
3.2　コマンド構造の説明 3-3
3.3　コマンドシンタックス 3-4
3.4　ヘルプ機能 3-5
3.5　コマンドの省略形と補完 3-5
3.6　ヒストリ機能 3-6
3.7　コマンド編集機能..... 3-7
3.8　ページャ機能 3-8
3.9　起動とログイン 3-9
3.10　設定の保存方法..... 3-11
3.11　設定のリストア方法..... 3-11
3.12　装置の起動時間..... 3-12

本装置の設定は Command Line Interface（以下 CLI）を使用できます。CLI はコンソールポートからコンソールケーブル経由で接続したターミナル（端末），またはシステムの IP ネットワークインタフェース（システムインタフェース）へのネットワーク経由で Telnet および SSH によるリモートアクセスが利用可能です。システムインタフェースへの通信は，Ethernet ポートまたは Network ポート経由のどちらかで行うことができます。

3.1 Command Line Interface（CLI）

CLI は，装置の動作パラメータの表示や設定を行うことができます。コマンドの詳細については，「PureFlow WS1 トラフィックシェーパー NF7500 シリーズ コマンドリファレンス」を参照してください。

（1）コンソールポート

コンソールポートの接続条件は次のとおりです。

通信速度：　9600 bit/s
キャラクタ長：　8 ビット
パリティ：　なし
ストップビット長：1 ビット
フロー制御：　なし

コンソールを接続するシリアルインタフェースコネクタは本体の前面にあります。オプション品のコンソールケーブル（RJ-45 用）または，市販のコンソールケーブル（miniUSB 用）を使用して接続してください。RJ-45 または miniUSB によるコンソールセッションは，最大 1 セッションまで同時利用可能です。

注：

通信速度を 115200 bit/s で使用する場合，お使いの環境（端末ハードウェア，ソフトウェア）によっては文字化けや文字抜けが発生する場合があります。文字化けや文字抜けが発生した場合は，通信速度を下げて使用してください。

本装置では，“set console baudrate”コマンドで通信速度を 9600 bit/s，19200 bit/s，38400 bit/s，115200 bit/s のいずれかに変更が可能です。

（2）Telnet

Telnet を使用するためには，本装置のシステムインタフェースの設定を行う必要があります。SSH セッションと Telnet セッションを合わせ，最大 8 セッションまで同時利用可能です。なお，セッション数には WebGUI と WebAPI のセッション数は含みません。

Ethernet ポートに接続されたネットワークを経由した端末から Telnet を使用してください。

システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。

Telnet を使用しない場合は，“set telnet”コマンドで Telnet を無効にしてください。

（3）SSH

本装置の SSH（Secure Shell）は，SSH Version2 をサポートしています。SSH を使用するためには，本装置のシステムインタフェースの設定を行う必要があります。SSH セッションと Telnet セッションを合わせ，最大 8 セッションまで同時利用可能です。なお，セッション数には WebGUI と WebAPI のセッション数は含みません。

SSH を使用しない場合は，“set ssh”コマンドで SSH を無効にしてください。

3.2　コマンド構造の説明

本装置の CLI には normal モードと administrator モードの 2 つがあります。normal モードでは，ステータスやカウンタ，および設定値の表示だけができます。administrator モードでは，すべての設定・変更・表示を行うことができます。

本装置のセキュリティを確保するため，normal モードに入るためのパスワードと administrator モードに入るためのパスワードを設定できます。パスワードが設定されている状態では，正しいパスワードを入力しないと，それぞれのモードに移行できません。

また，RADIUS 機能を使用しログイン認証を実施した場合，RADIUS サーバに設定されるユーザごとのサービスタイプに従って，normal モードまたは administrator モードに入ります。詳細は，「13 章 RADIUS 機能」を参照してください。

CLI プロンプト表示	CLI モード
PureFlow>	normal モード
PureFlow(A)>	administrator モード

図 3.2-1　コマンド構造

3.3　コマンドシンタックス

本装置 CLI のコマンドシンタックスは以下のような体系です。

アクション　設定項目　値

たとえば

アクション	設定項目	値
↓	↓	↓
設定する	時刻	数値
↓	↓	↓
set	date	20170323101010

また，機能に関する設定項目が多数あるので，「設定項目」は，「設定グループ＋設定項目」のように階層化している場合があります。

設定グループの例

ip
scenario
port

設定グループを伴うコマンドシンタックスの例を以下に示します。

アクション	設定グループ	設定項目	値
↓	↓	↓	↓
設定する	PORT グループ	1/2 の SPEED	100 M 固定
↓	↓	↓	↓
set	port	speed 1/2	100M

3.4　ヘルプ機能

システムプロンプト，またはコマンドの途中で疑問符(?)を入力すると，各コマンド入力モードで使用できるコマンドのリストを表示します。

```
PureFlow(A)> ?
Command                    Description
-------------------------------------------------------------------------------
?                          Lists the top-level commands available
add                        Adds some parameters, use 'add ?' for more
                           information
arp                        Shows address resolution table and control
clear                      Clears system statistics, use 'clear ?' for
                           more information
delete                     Deletes some parameters, use 'delete ?' for
                           more information
     ・                          ・
     ・                          ・

PureFlow(A)> set port ?
  flow_control             Sets the flow control parameters

  speed                    Sets the port speed
```

注:

<?>キーによるヘルプ機能はコマンドラインの最後尾でのみ動作します。

3.5　コマンドの省略形と補完

各コマンドは判別可能な範囲で省略可能です。たとえば，s で始まるコマンドは save，set，show などがありますが，2 文字目が異なっているので，se と入力されれば“set”コマンドであると判別可能です。以下の 2 つの入力は，同じコマンドを表します。

set port autonegotiation 1/2 disable　＝　se po au 1/2 d

判別可能な文字を入力した段階で，<TAB>キーを入力すると，キーワードを補完して表示します。

```
PureFlow(A)> set po<TAB>
                    ↓
PureFlow(A)> set port
```

注:

<TAB>キーによる補完機能はコマンドラインの最後尾でのみ動作します。また，コマンドのキーワードによっては，省略および<TAB>キーが動作しないものがあります。その場合は，ヘルプ機能でキーワードを確認し，すべてのキーワードを入力してください。

3.6　ヒストリ機能

コマンド ヒストリの使用方法

CLI は，入力されたコマンドの履歴（記録）機能を持っています。
コマンドヒストリから，次に入力しようとするコマンドに類似した履歴コマンドを呼び出し，あとで説明するコマンド編集機能で編集後，実行することができます。

コマンドヒストリは下記のキー入力で履歴を呼び出すことができます。

Ctrl-Pまたは上矢印キー

最も新しいコマンドからヒストリ バッファのコマンドを呼び出します。このキー操作を繰り返すと，続けて古いコマンドが呼び出されます。

Ctrl-Nまたは下矢印キー

Ctrl-P または上矢印キーでコマンドが呼び出されてから，ヒストリ バッファの新しいコマンドに戻ります。このキー操作を繰り返すと，続けて新しいコマンドが呼び出されます。

また，“show history”コマンドにより，コマンド履歴を表示することができます。

3.7　コマンド編集機能

コマンドラインを編集するために必要なキーストロークを示します。

Ctrl-Bまたは左矢印キー

カーソルを 1 文字分後退させます。

Ctrl-Fまたは右矢印キー

カーソルを 1 文字分前に進めます。

Ctrl-Aキー

カーソルを行の先頭に戻します。

Ctrl-Eキー

カーソルを行の最後に進めます。

Ctrl-DまたはDeleteキー

カーソルの位置にある文字を削除します。

Ctrl-HキーまたはBSキー

カーソルの位置の前の文字を削除します。

Ctrl-Kキー

カーソル以降の文字列を削除するとともに，バッファにコピーします。

Ctrl-Wキー

カーソルで選択された単語を削除するとともに，バッファにコピーします。

Ctrl-Yキー

カーソル位置にバッファの内容をペーストします。

Ctrl-Uキー

カーソルの行を削除するとともに，バッファにコピーします。

注:

コマンドライン編集機能はコマンドライン表示が 1 行に収まる場合のみ動作します。

3.8　ページャ機能

ターミナルへの表示を伴うコマンドを実行したとき表示内容が 24 行を超えるものについては，画面単位および行単位のページャ機能による表示を行います。その場合は画面の最終行に“—More—”と表示し，表示内容がその行以降も続いていることを表します。
ページャ機能を無効にする場合は，CLI より以下のコマンドで設定します。
PureFlow(A)> set pager disable
また，ページャ機能を有効にする場合は，CLI より以下のコマンドで設定します。
PureFlow(A)>set pager enable

“—More—”が表示されているときに入力可能なキーは，以下のとおりです。

スペースまたはFキー

次の画面を表示します。

Enterキー

次の行を表示します。

Qキー

表示を終了します。

3.9　起動とログイン

本装置の電源を投入すると，装置内部の内蔵フラッシュメモリ（以下，内蔵フラッシュメモリ）のソフトウェアオブジェクトを自動で読み込み起動します。また，ソフトウェアオブジェクト（ファイル名：nf7500.bin）が入ったSD カードまたは USB メモリ（以下，外部メディア）を挿入して電源を投入すると外部メディア内のソフトウェアオブジェクトを優先して読み込み起動します。外部メディアの優先順位は USB メモリ，SD カードの順です。

コンフィギュレーションについても同様に，設定ファイル（ファイル名：extcnf.txt）が入った外部メディアが挿入されている場合，外部メディア内の設定ファイルを優先して読み込みます。

外部メディアの読み込み中は，外部メディアに対しアクセスをしていますので，起動が完了する前に外部メディアを抜いたり電源をオフにすると，外部メディアが破損する恐れがあります。

本装置のコンソールポートに接続されている場合は，下記のような起動メッセージが表示されます（起動メッセージでの表示項目については，ソフトウェアバージョンによって，変更されることがあります）。

```
Anritsu PureFlow NF7500-S001A Software Version 1.5.1
Copyright 2017-2021 ANRITSU CORPORATION

Power Supply Unit 0             ... [OK]
Fan Unit 0                      ... [OK]
Serial Port                     ... [OK]
Backup Memory Checking          ... [OK]
Real Time Clock Checking        ... [OK]
File System Checking            ... [OK]
EEPROM Checking                 ... [OK]
Ethernet Controller Checking
        Management Port         ... [OK]
        Internal Port           ... [OK]

. .
Slot 1 boot up complete
      Medium type GbE/2T, GbE/4SFP 4 ports

System booting up
.........................
Loading Configuration from Master.

Restoration in Progress
100 % done

Restoration completed

Warning. Channel does not exist.
Please add the channel by "add channel" command.

PureFlow login:
```

設定に際しては，まずシステムコンソールとしてコンソールポートにコンソールケーブルを接続します。コンソールが接続され，［Enter］キーを入力すると，コンソール上に次のようなメッセージを表示し，ログイン受付状態となります。

```
PureFlow login:
```

本装置のユーザ名は“root”です。また，工場出荷時の初期状態において，ログインパスワードは未設定となっています。ログインが認証されると，プロンプトが表示され，コマンド受付状態になります。

```
PureFlow login:root
Password:([Enter]キーを入力)
PureFlow＞
```

この状態は normal モードで，設定内容を見ることはできますが，内容を変更することはできません。設定を行うためには administrator モードに移行する必要があります。この移行は“admin”コマンドで行います。

```
PureFlow＞admin
Enter the Admin Password:([Enter]キーを入力)
PureFlow(A)＞
```

この administrator モードでは，各種パラメータの表示に加えて，動作パラメータの変更，パスワードの設定が可能になります。administrator モードは，同時に複数のユーザが移行でき，同時に設定変更が可能です。administrator モードの権限は，パスワードを設定するなど，ユーザ管理を行ってください。

3.10 設定の保存方法

本装置にて設定した内容は，コマンドによる設定後すみやかに有効となりますが，そのままでは電源断時に設定内容は失われ，再起動後は無効となります。本装置は内蔵フラッシュメモリに設定内容をコンフィギュレーションファイルとして保存することが可能です。次回電源投入後に設定内容を有効にするためには，内蔵フラッシュメモリに save コマンドにて設定内容を保存する必要があります。

保存方法は次のとおりです。

```
PureFlow(A)＞ save config
Do you wish to save the system configuration into the flash memory (y/n)? y
........
Done
PureFlow(A)＞
```

⚠ 注意

コンソール画面に“Done”の表示がされる前に，本装置の電源を切断すると正しく設定値が保存されない場合があります。また，内蔵フラッシュメモリの故障の原因となります。

3.11 設定のリストア方法

本装置の電源を投入すると，内蔵フラッシュメモリに保存されたコンフィギュレーションファイルを自動で読み込みます。また，コンフィギュレーションファイル（ファイル名：extcnf.txt）が格納されている SD カードまたは USB メモリ（以下，外部メディア）を挿入して電源を投入すると，外部メディア内のコンフィギュレーションファイルを優先して読み込みます。外部メディアの優先順位は USB メモリ，SD カードの順です。

外部メディアの読み込み中は，外部メディアに対しアクセスをしていますので，起動が完了する前に外部メディアを抜いたり電源をオフにすると，外部メディアが破損する恐れがあります。

3.12 装置の起動時間

コンフィギュレーション情報量によって，save コマンド実行時間および電源投入時の起動時間が異なります。以下に，参考値を示します。

表 3.12-1　装置の起動時間（参考値）

	save コマンド実行時間	起動時間
デフォルト	—	2 分 00 秒
シナリオ 100 件 フィルタ 100 件	5 秒	2 分 10 秒

※　フィルタ／シナリオの設定の説明は「第 8 章　トラフィックコントロール機能」を参照してください。

※　save コマンド実行時間と起動時間は，設定コマンドのライン数やパラメータの数によって異なります。

第4章　装置本体の情報表示と設定

ここでは，装置本体の情報表示と設定について説明します。

4.1　日付／時刻 4-2
4.2　Simple Network Time Protocol（SNTP） 4-4
4.3　ユーザ名とパスワード 4-5
4.4　SYSLOG 4-6
4.5　モジュール情報 4-9
4.6　ライセンスキー 4-11

本装置には時刻，CLI パスワードなどの装置全体にかかわる設定や，ハードウェア，ソフトウェアのバージョン表示などの装置全体にかかわる情報があります。これらの情報表示と設定について説明します。

本装置には，下記の装置本体情報と設定項目があります。

表 4-1　基本情報と設定

日付／時刻	装置内蔵のカレンダ・クロックです。SYSLOG によるイベントの記録に使用されます。
SNTP	Simple Network Time Protocol（SNTP）クライアント機能です。
ユーザ名とパスワード	CLI による装置へのアクセス制御のためのユーザ名とパスワードです。
SYSLOG 設定	装置の状態変化イベントやエラーイベントを内蔵メモリ，バッテリバックアップメモリに保存したり，リモートホストに送信することができます。
モジュール情報	装置内の各モジュール情報（バージョンなど）です。

4.1　日付／時刻

本装置は，カレンダ機能に対応しています。日付，時刻はSYSLOGによるイベントの記録に使用されます。日付，時刻の設定は CLI コマンドで指定する方法と，SNTP クライアント機能により NTP サーバの時刻に自動同期させる方法があります。

CLIコマンドによる設定

CLI で設定する場合は以下のコマンドを使用します。

表 4.1-1　日付／時刻設定

set date <yyyymmddhhmmss>	日付，時刻の設定を行います。
set timezone 　<hours-offset> [<minutes-offset>]	協定世界時（UTC：Coordinated Universal Time）からのタイムゾーンオフセットを設定します。 デフォルト値は+9[時間]0[分]です。
set summertime 　from <week> <day> <month> <hh> 　to <week> <day> <month> <hh> 　[offset]	夏時間の適用期間を設定します。 デフォルトでは設定されていません。
unset summertime	夏時間の設定を解除します。
show date	日付，時刻の表示を行います。

コマンドの実行例を示します。

```
PureFlow(A)> set timezone +9
PureFlow(A)> set summertime from 2 Sunday March 2 to 1 Sunday November 2
PureFlow(A)> set date 20170323124530
PureFlow(A)> show date
Mar 23 2017(Thu) 12:45:32
UTC Offset      : +09:00
Summer Time     : From Second Sunday March 02:00
                  To     First Sunday November 02:00
                  Offset 60 minutes
PureFlow(A)>
```

タイムゾーンの設定は，UTC（協定世界時）からのオフセット時間を符号付きで入力します。必要ならば分単位のオフセットを入力します。

夏時間の設定は，夏時間の開始日時と終了日時を指定します。必要ならば夏時間である間時刻に加えるオフセットを分単位で入力します。オフセットを省略した場合は 60［分］が適用されます。
開始日時および終了日時は以下のフォーマットで指定します。

日付，時刻の設定は西暦年，月，日，時，分，秒を続けて 14 桁で入力します。

カレンダ・クロックに設定した時刻は，装置内部のバッテリで駆動され，装置電源がオフの状態でも止まらずに進み続けます。

4.2　Simple Network Time Protocol（SNTP）

本装置は，SNTP クライアント機能を実装しています。SNTP クライアントはシステムインタフェース経由で NTP サーバと通信し，本装置の日付および時刻を NTP サーバと同期させます。SNTP クライアントを使用するためには，本装置のシステムインタフェースの設定を行う必要があります。システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。

SNTP クライアントの設定には以下のコマンドを使用します。

表 4.2-1　SNTP コマンド

set sntp {enable | disable}	SNTP クライアント機能を有効化／無効化します。 有効化後，interval 設定時間が経過すると時刻同期を開始します。
set sntp server <IP_address>	NTP サーバの IP アドレスを設定します。NTP サーバは 1 つのみ指定できます。
unset sntp server	NTP サーバの IP アドレスを設定解除します。
set sntp interval <interval>	NTP サーバへ定期的に時刻の問い合わせを行う間隔を秒単位で設定します。設定範囲は 60～86400［秒］です。デフォルトは 3600［秒］です。設定可能な値は上記のとおりですが，実際の動作は 60 秒単位に端数切り上げで丸められます。 変更後の interval 設定時間が経過すると時刻同期を開始します。
sync sntp	NTP サーバへ時刻の問い合わせを行います。 SNTP クライアント機能が有効の場合のみ実行可能です。
show sntp	SNTP クライアント機能の状態および設定を表示します。

NTP サーバ 192.168.10.10，問い合わせ間隔 86400 秒を設定する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set sntp server 192.168.10.10
PureFlow(A)> set sntp interval 86400
PureFlow(A)> set sntp enable
PureFlow(A)> sync sntp
Transmitted to the server.
PureFlow(A)> show sntp
Status        : enable
Server        : 192.168.10.10
Interval      : 86400
Sync          : kept
PureFlow(A)>
```

“show sntp”コマンドの Sync の表示が kept になっていれば，NTP サーバとの同期が取れている状態です。

4.3　ユーザ名とパスワード

装置のセキュリティを保つために装置設定をシリアルコンソール，Telnet または SSH で行う前にはユーザ名とパスワードによる認証が行われます。このパスワードはユーザが変更することができます。

表 4.3-1　パスワード設定

set password	ログインパスワードを設定します。ログインパスワードは 16 文字以内です。
set adminpassword	administrator モードに移行するためのログインパスワードを設定します。ログインパスワードは 16 文字以内です。

コマンドの実行例を示します。

```
PureFlow(A)> set password

New Password:
Retype the new Password:
```

設定したいパスワードを入力してください。

設定したいパスワードをもう一度入力してください。

ログインパスワードに設定できる文字は，以下の ASCII 文字です。

1234567890
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
!#$%&'()=~-^|¥@`[]{}:*;+_/.,<>

ログインパスワードを設定解除する場合は，“New password”の問いに対し，パスワードを入力せず，[Enter]キーを入力してください。

4.4 SYSLOG

装置に起きたエラーイベントやリンクアップ・ダウンなどのイベント(以後，ログデータと呼ぶ)を複数の方法で記録することができます。本装置はログデータを通電状態で内蔵メモリに最大 8000 イベント保持します。内蔵メモリに保持するログデータは電源が遮断されると消失します。ログデータは内蔵バックアップメモリと，ネットワークを介した SYSLOG ホストに記録することができます。内蔵バックアップメモリへは，前回と前々回の装置稼働時におけるログデータを，それぞれ最大 1000 イベント保持します。内蔵バックアップメモリに保持するログデータは，電源を遮断しても消失しません。

表 4.4-1　シスログコマンド

show syslog	内蔵メモリに記録されたログデータを表示します。
show backup syslog [last | second_last]	内蔵バックアップメモリに記録されたログデータを表示します。
clear syslog	内蔵メモリに記録されたログデータをクリアします。
set syslog severity <severity_level>	syslog ホストに送信するシステムログの最低レベル(重大度)を設定します。設定されたレベルより低いレベルのログは syslog ホストに送信されません。なお，装置のシステムログは本設定に関わらず，全ての重大度のシステムログが記録されます。
show syslog host	システムログ出力に関する設定を表示します。
set syslog host {enable | disable}	SYSLOG ホストへの記録を有効化／無効化します。
add syslog host <IP_address> [<udp_port>]	SYSLOG ホストの IP アドレス／UDP ポートを追加します。
delete syslog host <IP_address>	SYSLOG ホストの IP アドレス／UDP ポートを削除します。
set syslog facility {ccpu | fcpu} <facility_code>	システムログの facility を設定します。 ccpu： 制御系処理部で検出，記録したログメッセージ fcpu： フォワーディング系処理部で検出，記録したログメッセージ

ログデータはテキストデータとして以下のフォーマットで装置内に記録されています。

・show syslog コマンドで表示される内蔵メモリのログデータ

プライオリティ	日時	Host	Ident	PID	メッセージ
134	Jun 30 16:51:19	PureFlow	System	[10330]	Port 1/1 changed Up from Down.

・show backup syslog コマンドで表示される内蔵バックアップメモリのログデータ

プライオリティ	日時	メッセージ
134	2012 Jun 30 16:51:19	Port 1/1 changed Up from Down.

プライオリティ

プライオリティはログメッセージの特徴を示すコードです。プライオリティのコードは RFC3164 で規定されている方式で計算し，格納されます。プライオリティコードはメッセージのカテゴリを表す Facility とメッセージの重大度を示す Severity の 2 つの数値を組み合わせたコードで表現されます。

プライオリティ＝Facility×8＋Severity

本装置の SYSLOG メッセージの Facility は設定が可能です。設定可能な Facility の範囲は 0～23 です。
デフォルト値は以下となります。
制御系処理部: 16
フォワーディング系処理部: 17

コマンドの実行例を以下に示します。

```
PureFlow(A)> set syslog facility ccpu 18
PureFlow(A)> set syslog facility fcpu 19
```

制御系処理部のFacilityを18にします。

フォワーディング系処理部のFacilityを19にします。

Severity には 0 から 6 までの数値が格納されます。プライオリティ 0 が最も重大度が高く，数値が大きくなるほど低くなります。各メッセージの重大度は RFC 3164 に規定された以下の基準に従って割り当てられています。

Numerical Code	Severity	
0	Emergency:	system is unusable
1	Alert:	action must be taken immediately
2	Critical:	critical conditions
3	Error:	error conditions
4	Warning:	warning conditions
5	Notice:	normal but significant condition
6	Informational:	informational messages

たとえばプライオリティ 129（16×8＋1）のメッセージは Facility が 16，Severity が 1 です。つまり制御系処理部で検出された Alert レベル（緊急）メッセージです。

日時

イベントが発生した日時です。

Host

Host はログメッセージを記録した装置名を示します。
装置名は“set snmp sysname”コマンドで変更できます。
装置名に下記条件が含まれる場合，装置名は“PureFlow”固定となります。
・英字(a～z,A～Z)，数字(0～9)，ハイフン(-)以外の文字が含まれる場合
・最初の文字を英字または数字以外としている場合
・最後の文字をハイフン(-)としている場合

Ident

Ident はログメッセージを記録したプログラムの識別子を示します。“System”固定です。

PID

PID はログメッセージを記録したプロセスのプロセス ID 値を示します。

メッセージ

イベントの内容を示すメッセージが格納されます。

メッセージは show syslog コマンドで表示できます。

```
PureFlow(A)> show syslog
-----------------------------------------------------------------------------------------------------
Pri Date      Time                          Host          Ident       [PID]      Message
-----------------------------------------------------------------------------------------------------
134 Jan 25 21:50:54 PureFlow    System       [10330]: Port 1/1 changed Up from Down.
```

データは装置の通電中，メモリに保持されていますが，オペレータがメッセージをクリアすることができます。

```
PureFlow(A)> clear syslog
PureFlow(A)> show syslog
-----------------------------------------------------------------------------------------------------
Pri Date      Time                          Host          Ident       [PID]      Message
-----------------------------------------------------------------------------------------------------
PureFlow(A)>
```

4.5　モジュール情報

装置内の各モジュール情報を表示します。バージョン，製造番号などを確認することができます。

表 4.5-1　モジュール情報

show module	各モジュール情報を表示します。

モジュール情報には，以下のものがあります。

System MAC Address

システムインタフェースの MAC アドレスを表します。

Channel MAC Address

チャネルインタフェースの MAC アドレスを表します。

Chassis Model Name

本体の形名を表します。

Chassis Serial Number

本体の製造番号を表します。

Module Version

内蔵プリント板のハードウェアバージョンを表します。

Software Version

インストールしたソフトウェアのバージョンを表します。

U-Boot Version

U-Boot バージョンを表します。

MCU Version

MCU バージョンを表します。

Uptime

本装置が起動してからの動作時間を表します。

Temperature

入気温度を表します。

Power Supply Unit N

電源ユニットの状態を表します。

FAN Unit N

ファンユニットの状態を表します。
本装置は背面側に排気用のファンが 2 個あります。Fan 0 が，背面からみて右側のファンの回転数を示します。Fan 1 が，背面からみて左側のファンの回転数を示します。

コマンドの実行例を示します。

```
PureFlow(A)> show module
Anritsu PureFlow NF7500-S001A Software Version 1.5.1
Copyright 2017-2021 ANRITSU CORPORATION

System MAC Address              : 00-00-91-12-34-56
Channel MAC Address             : 00-00-91-12-34-57

Chassis Model Name              : NF7501A
Chassis Serial Number           : 1234567890

Module Version                  : 01B
Software Version                : 1.5.1
U-Boot Version                  : 3.1.3
MCU Version                     : 112

Uptime                          : 0 days, 00:27:17
Temperature
  Intake   Temperature          : 32C
Power Supply Unit 0
  Operation Status              : operational
FAN Unit 0
  Operation Status              : operational
  Fan 0 Speed                   : 3840[rpm]
  Fan 1 Speed                   : 3840[rpm]
PureFlow(A)>
```

4.6　ライセンスキー

ライセンスキーを購入することにより，本装置の機能や性能を拡張することが可能です。

ライセンスキーは，キーを記載したライセンス証書で提供されます。ライセンスキーを装置購入後に購入する場合は，装置シリアル番号をご指定ください。

ライセンスキーを本装置に設定するには，“set option”コマンドを入力してください。ライセンスキー入力を促すメッセージが表示されますので，ライセンスキーを入力してください。ライセンスキー入力の際，4 文字ごとにハイフンを入力しても，ハイフンを入力しなくても同じライセンスキーとして認識します。入力されたライセンスキーと装置のシリアル番号を比較し，一致した場合にライセンスが有効となります。

ライセンスキーに関するコマンドには以下ものがあります。

表 4.6-1　ライセンスキーコマンド

set option	ライセンスキーを本装置に設定します。
show option	有効になっているライセンスを表示します。

コマンドの実行例を示します。

```
PureFlow(A)> set option
Enter the option key : XFS8wbFEFBNkfqLJ

Authentication succeed.

    Making be available : License Key NF7500-L114A (1G Bandwidth License)
Updation done.
Enter update scenario command to change port bandwidth.
PureFlow(A)>
PureFlow(A)> show option
    License Key NF7500-L114A available (1G Bandwidth License)
PureFlow(A)>
```

（空白ページ）

本装置は，ネットワーク経由のリモートによる設定，制御を行うために，Ethernet ポートを装置前面に実装しています。
本ポートはマネジメント用のローカルポートで，Network ポートとは切り離されています。本ポートは Auto-MDIX をサポートした 10/100/1000BASE-T ポートです。

Ethernet ポートには以下の設定が可能です。
　AutoNegotiation の有効／無効（注 1 参照）
　通信速度（10 Mbit/s，100 Mbit/s，1 Gbit/s）（注 2 参照）
　duplex モード（full，half）（注 2 参照）

図 5-1　管理ネットワーク

Ethernet ポートに接続されたネットワーク経由のリモートによる設定，制御を行うためには，本装置のシステムインタフェースの設定を行う必要があります。システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。

（注 1）
1 Gbit/s 通信を行う場合は，AutoNegotiation 有効で使用してください。AutoNegotiation 無効で通信速度を 1Gbit/s 固定設定とすることはできません。AutoNegotiation 無効で通信速度を 1Gbit/s 設定とした場合，強制的に AutoNegotiation 有効となります。この場合，通信速度は 1Gbit/s のみアドバタイズされます。

（注 2）
通信速度／duplex モードの設定は，AutoNegotiation 無効のときのみ有効です。AutoNegotiation 有効のとき，AutoNegotiation の結果が反映されるため，これらの設定内容は適用されず，AutoNegotiation 無効に設定したときに反映されます。“`show port`”コマンドでリンク状態が半二重の場合，ポートの AutoNegotiation／通信速度／duplex モードの設定が接続装置と合っているか確認してください。

（注 3）
Ethernet ポートの最大フレーム長は 1518 Byte 固定です。

（空白ページ）

第6章　Network ポートの設定

ここでは，本装置の Network ポートの設定について説明します。

6.1　概要 6-2
6.2　メディアタイプの設定 6-4
6.3　Network ポートの属性の設定 6-5
6.4　最大フレーム長の設定 6-7
6.5　設定，状態の確認 6-9

6.1　概要

Network ポートとは，ネットワーク上に流れるトラフィックをコントロール（トラフィックコントロール）するためのポートです。

本装置は，以下に示す 2 種類の Network ポートをサポートしています（注 1 参照）。

Network ポート識別番号	ポート種別
1/1（1B）～1/2（2B）	RJ-45（10/100/1000BASE-T／Auto-MDIX）
1/1～1/4	SFP

本装置の SFP ポートには，以下に示す SFP を装着することが可能です。

SFP　1000BASE-SX/1000BASE-LX（LC コネクタ）
SFP　10/100/1000BASE-T（Auto-MDIX）

Network ポートには以下の設定が可能です。
AutoNegotiation の有効／無効（注 2 参照）
フローコントロール（auto，pause フレーム受信／送信）
通信速度（10 Mbit/s，100 Mbit/s，1 Gbit/s）（注 3 参照）
duplex モード（full，half）（注 3 参照）
最大フレーム長（2048 Byte，10240 Byte）（注 4 参照）

上記設定は装着されているポート種別によって適用範囲が異なります。

表 6.1-1　ネットワークポート設定パラメータ

	1000BASE-SX/LX	10/100/1000BASE-T
AutoNegotiation	有効／無効	有効／無効
通信速度	1G のみ	10M/100M/1G
duplex モード	Full のみ	Full/Half
フローコントロール	Auto 受信 ON/OFF 送信 ON/OFF	Auto 受信 ON/OFF 送信 ON/OFF
最大フレーム長	2048／10240[Byte]	2048／10240[Byte]

CLI から Network ポートを指定するには＜スロット番号／ポート番号＞の組み合わせで指定します。
本装置のスロット番号には 1 を指定します。

スロット内のポート番号は左から順番に 1/1，1/2，1/3，1/4 と番号付けられており，これにより，Network ポートの識別番号は以下のようになります。

図 6.1-1　ネットワークポート識別番号

（注 1）
Network ポート識別番号 1/1 と 1/2 は，RJ-45 と SFP の選択ポートです。

（注 2）
10/100/1000BASE-T（RJ-45／SFP）で 1 Gbit/s 通信を行う場合は，AutoNegotiation 有効で使用してください。AutoNegotiation 無効で通信速度を 1Gbit/s 固定設定とすることはできません。ただし，10/100/1000BASE-T の SFP を使用した場合，AutoNegotiation 無効で通信速度を 1Gbit/s 設定とした場合，強制的に AutoNegotiation 有効となります。この場合，通信速度は 1Gbit/s のみアドバタイズされます。

（注 3）
1000BASE-SX/LX の場合，AutoNegotiation の設定にかかわらず，通信速度は 1Gbit/s，duplex モードは Full となります。
10/100/1000BASE-T（RJ-45／SFP）の通信速度／duplex モードの設定は，AutoNegotiation 無効のときのみ有効です。AutoNegotiation 有効のとき，これらの設定内容は適用されず，AutoNegotiation 無効に設定したときに反映されます。

（注 4）
最大フレーム長の設定は，Ethernet ポートには適用されません。Ethernet ポートの最大フレーム長は 1518 Byte 固定です。

（注 5）
“`show port`”コマンドでリンク状態が半二重の場合，ポートの AutoNegotiation／通信速度／duplex モードの設定が接続装置と合っているか確認してください。

6.2　メディアタイプの設定

Network ポートのメディアタイプを CLI から選択できます。Network ポート識別番号 1/1 と 1/2 のメディアタイプとして，RJ-45 または SFP のどちらかを選択可能です。

表 6.2-1　メディアタイプ設定

set port media-type <slot/port> {rj45 | sfp}	Network ポートで使用するメディアタイプを設定します。デフォルトは rj45 です。

Network ポート 1/2 のメディアタイプとして SFP を設定する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set port media-type 1/2 sfp
PureFlow(A)>
```

6.3　Network ポートの属性の設定

1000BASE-T SFP 使用時，AutoNegotiation 無効のときは，Network ポートの通信速度や duplex モードといったポートの動作属性を CLI から変更できます。これらの Network ポート属性は，通常 AutoNegotiation により，最も適切な動作モードに自動的に設定されます。接続先のスイッチやノードが AutoNegotiation をサポートしていない場合は，Network ポートの通信速度と duplex モードをマニュアル設定する必要があります。接続先が AutoNegotiation 設定になっている場合は，本装置も AutoNegotiation 設定にしてください。片方がマニュアル設定で，他方が AutoNegotiation 設定になっていると，正しく接続できません。

表 6.3-1　Network ポートの属性設定

set port autonegotiation <slot/port> {enable | disable}	Network ポートの AutoNegotiation の有効／無効を設定します。デフォルトは enable です。
set port speed <slot/port> {10M | 100M | 1G}	Network ポートの通信速度を設定します。本設定は，AutoNegotiation 無効のときの通信速度設定です。AutoNegotiation 有効のとき，この設定内容は無効です。デフォルトは 1G です。
set port duplex <slot/port> {full | half}	Network ポートの duplex モードを設定します。本設定は，AutoNegotiation 無効のときの duplex モード設定です。AutoNegotiation 有効のとき，この設定内容は無効です。デフォルトは full です。

Network ポート 1/2 の AutoNegotiation 無効，通信速度 100 Mbit/s，duplex モード full を設定する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set port autonegotiation 1/2 disable
PureFlow(A)> set port speed 1/2 100M
PureFlow(A)> set port duplex 1/2 full
PureFlow(A)>
```

また，Network ポートのフローコントロールを CLI から変更できます。

表 6.3-2　フローコントロール設定

set port flow_control <slot/port> auto set port flow_control <slot/port> {recv | send} {on | off}	Network ポートのフローコントロールを設定します。デフォルトは auto です。 なお，auto を指定した場合，ポートタイプにより次のように動作します。 ポートタイプ 1000BASE-T および 1000BASE-X の場合： AutoNegotiation により pause フレームの受信および送信を決定します。 AutoNegotiation 無効の場合は受信および送信ともに有効となります。

Network ポート 1/2 のフローコントロールで pause フレームを送受信しない設定にする場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set port flow_control 1/2 recv off
PureFlow(A)> set port flow_control 1/2 send off
PureFlow(A)>
```

6.4　最大フレーム長の設定

Network ポートで転送可能な最大フレーム長を CLI から変更できます。一般的に，MTU（Maximum Transmission Unit）とはヘッダや FCS を含まないペイロード長を指しますが，本コマンドでは Ethernet ヘッダおよび FCS を含むフレーム全体の長さを指定します。ただし，VLAN Tag ありフレームの場合は「本設定値+4」バイト，2 重 VLAN Tag ありフレームの場合は「本設定値+8」バイトが実際の MTU となります。最大フレーム長は，すべての Network ポートで共通の設定値です。

表 6.4-1　最大フレーム長設定

set port mtu {2048 | 10240}	Network ポートの最大フレーム長を設定します。 デフォルトは 2048 Byte です。

最大フレーム長の設定変更を適用するには，装置の再起動が必要です。最大フレーム長を 10240 バイトに設定する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set port mtu 10240
Warning
This configuration change will be take effect on next boot.
Please save the system configuration and reboot the system.
If changed to 10240, some scenario parameters will be rounded as below.
  bandwidth minimum        1k -> 5k
  bandwidth resolution   1k ->   5k
  buffer size minimum     2k -> 11k
If changed to 2048, cahnnel mtu specified larger than 2048 will be rounded.

Do you wish to save the system configuration into the flash memory (y/n)? y

Done

Rebooting the system, ok (y/n)? y
```

コマンドを実行すると，再起動が必要であること，およびシナリオパラメータの設定範囲に関する Warning メッセージとともに，コンフィギュレーションの保存を確認するプロンプトが表示されます。“y”を入力してコンフィギュレーションを保存してください。次に，装置の再起動を確認するプロンプトが表示されます。“y”を入力して装置を再起動してください。装置を再起動すると 10240 バイトへの設定変更が適用されます。

注:

1. 本設定値によって，下記シナリオパラメータで有効な設定範囲と設定単位が変化します。
最大フレーム長の変更によって，すでに登録済みのシナリオパラメータが範囲外となる場合，自動的に範囲内への丸めを行います。また，追加で登録する場合は丸めが適用される旨の Warning メッセージが表示されます。いずれにおいても，トラフィックコントロールは丸め後の値で実行されます。

表 6.4-2　最大フレーム長ごとのシナリオパラメータ

シナリオパラメータ		最大フレーム長（Network ポート）	
		2048[Byte]	10240[Byte]
最低帯域	設定範囲	1k[bit/s]～1G[bit/s] および 0	5k[bit/s]～1G[bit/s] および 0
	設定単位	1k[bit/s]	5k[bit/s]
最大帯域	設定範囲	1k[bit/s]～1G[bit/s]	5k[bit/s]～1G[bit/s]
	設定単位	1k[bit/s]	5k[bit/s]
入力バースト長	設定範囲	2k[Byte]～100M[Byte]	11k[Byte]～100M[Byte]
	設定単位	1k[Byte]	1k[Byte]

2. 本設定値によって，下記チャネルパラメータで有効な設定範囲が変化します。
最大フレーム長を 2048 バイトにした場合，すでに登録済みのチャネルパラメータが範囲外となる場合，自動的に範囲内への丸めが行われます。

表 6.4-3　最大フレーム長ごとのチャネルパラメータ

チャネルパラメータ		最大フレーム長（Network ポート）	
		2048[Byte]	10240[Byte]
MTU	設定範囲	300～10200[Byte] 2008 より大きい値に設定していた場合，デフォルト値 1488[Byte]に丸めを行います。	300～10200[Byte]

3. 本設定値によって，下記ピークバーストサイズで有効な設定範囲が変化します。
最大フレーム長を 2048 バイトにした場合，すでに登録済みのピークバーストサイズが範囲外となる場合，自動的に範囲内への丸めが行われます。

表 6.4-4　最大フレーム長ごとのピークバーストサイズ

ピークバーストサイズ		最大フレーム長（Network ポート）	
		2048[Byte]	10240[Byte]
ピークバーストサイズ	設定範囲	0～9216[Byte] ピークバーストサイズを 9216 より大きい値に設定していた場合，デフォルト値 1536[Byte]に丸めを行います。	0～46080[Byte]

6.5　設定，状態の確認

設定コマンドで設定した内容や，現在の Network ポートの動作状態を確認するには，“show port”コマンドを使用します。

```
PureFlow(A)>   show port
Port     Type            Media type  Status    Link     Autonego Speed Duplex
----     --------------  ----------  ------    ----     -------- ----- ------
1/1      1000BASE-T      RJ45        Enabled   Up       Enabled  1G    Full
1/2      1000BASE-T      RJ45        Enabled   Up       Enabled  1G    Full
1/3      1000BASE-T      SFP         Enabled   Up       Enabled  100M  Full
1/4      1000BASE-T      SFP         Enabled   Up       Enabled  100M  Full
system   1000BASE-T      RJ45        Enabled   Up       Enabled  100M  Full
PureFlow(A)>
```

“show port”コマンドにより，実装されているすべての Network ポートの状態が確認できます。さらに詳細な情報を確認するには，Network ポート識別番号をコマンド引数で指定することにより，確認できます。

```
PureFlow> show port 1/1

Slot/Port                 : 1/1
Port type                 : 1000BASE-T
Media type                : RJ45
Admin status              : Enabled
Oper status               : Up
Auto negotiation          : Enabled
Admin speed               : 1G
Oper speed                : 1G
Admin duplex              : Full
Oper duplex               : Full
Admin Tx Flow control     : Auto
Admin Rx Flow control     : Auto
Oper Tx Flow control      : On
Oper Rx Flow control      : On
Admin MTU                 : 2048
Oper MTU                  : 2048
PureFlow>
```

Network ポートの統計情報を確認するには，“show counter”コマンドを使用します。本コマンドで表示するカウンタ長は，32 ビットです。

```
PureFlow(A)> show counter
Port          Rcv Octets     Rcv Packets    Trs Octets     Trs Packets
-------       ----------     ----------     ----------     ----------
  1/1         57566366       14194297               0               0
  1/2                0              0        59383412        14195494
  1/3         57566366       14194297               0               0
  1/4                0              0        59383412        14195494
system           58368            152           85424             152

Port          Rcv Broad      Rcv Multi      Trs Broad      Trs Multi
-------       ----------     ----------     ----------     ----------
  1/1            10000       14208097               0               0
  1/2                0              0           10000        14209615
  1/3            10000       14208097               0               0
  1/4                0              0           10000        14209615
system               5              0              10               0

Port          Err Packets    Collision      Discard
-------       ----------     ----------     ----------
  1/1                0              0              0
  1/2                0              0              0
  1/3                0              0              0
  1/4                0              0              0
system             N/A            N/A            N/A
```

また，Network ポート識別番号をコマンド引数で指定することにより，詳細内容を表示できます。本コマンドで表示するカウンタ長は，64 ビットです。“show counter”コマンドの 32 ビットカウンタがラップアラウンドした場合，“show counter <slot/port>”コマンドの 64 ビットカウンタと異なる値が表示されることに注意してください。

```
PureFlow(A)> show counter 1/1
Rcv Packets                    14194297
Rcv Broad                         10000
Rcv Multi                      14208097
Rcv Octets                     57566366
Rcv Rate                             16 [kbps]
Trs Packets                           0
Trs Broad                             0
Trs Multi                             0
Trs Octets                            0
Trs Rate                              0 [kbps]
Collision                             0
Drop                                  0
Discard                               0
Error Packets                         0
        CRC Align Error                     0
        Undersize Packet                    0
        Oversize Packet                     0
        Fragments                           0
        Jabbers                                     0
```

第7章　システムインタフェースの設定

ここでは，本装置のシステムインタフェースの設定について説明します。

7.1　概要 7-2
7.2　システムインタフェース通信 7-3
7.3　システムインタフェースフィルタ 7-12
7.4　コンフィギュレーション例 7-133
7.5　設定，状態の確認 7-20

7.1　概要

システムインタフェースとは，管理者が本装置をネットワーク経由でリモートアクセスするための IP ネットワークインタフェースです。本装置へのリモートからの制御には Telnet，SNMP などの手段を用い，本装置の設定および状態監視を行うことができます。

以下に示すように，システムインタフェースは Ethernet ポート経由でアクセスするか，Network ポート経由でアクセスするかのいずれかを選択することができます。

（1）Ethernet ポート経由によるリモート管理

トラフィックコントロールを行うネットワーク（Network ポートからの入出力）とは別の管理用ネットワークに管理者端末を配置し，Ethernet ポートを経由して制御することができます。セキュリティ上，トラフィックコントロールを行うネットワーク内から分離させたい場合などに有効です。

図 7.1-1　Ethernet ポート経由によるリモート管理

（2）Network ポート経由によるリモート管理

トラフィックコントロールを行うネットワーク内に管理者端末を配置し，Network ポートを経由して制御することができます。管理専用のネットワークを用意する必要がないため，ネットワーク構成をシンプルにすることができます。

図 7.1-2　Network ポート経由によるリモート管理

7.2　システムインタフェース通信

システムインタフェースへの通信は，Ethernet ポート経由または Network ポート経由のどちらかで行うことができます。Ethernet ポート経由で行う場合は，VLAN Tag なしフレームの通信を行うことができます。また，Network ポート経由で行う場合は，通信を行う Network ポートを指定（1/1 のみ，1/2 のみ，1/3 のみ，1/4 のみ，すべて）でき，VLAN Tag なしパケット，VLAN Tag ありパケット，二重 VLAN Tag なしパケット，二重 VLAN Tag ありパケットの通信を行うことができます。
また，不特定多数の端末からシステムインタフェースへの通信を制限するためにフィルタ機能を使用することもできます。

システムインタフェース通信は IPv4 および IPv6 の同時利用が可能ですが，一部の機能は IPv4 のみのサポートとなります。

表 7.2-1　サポートシステムインタフェース通信

機能	IPv4	IPv6
Telnet	○	○
SSH	○	○
RADIUS	○	○
TFTP	○	○
FTP	○	○
SYSLOG	○	○
SNTP	○	○
SNMP	○	×
PING	○	○
Traceroute	○	○
Telnet クライアント	○	○
システムインタフェース フィルタ	○	○
WebAPI	○	○
WebGUI	○	○
OpenFlow	○	○
トラフィック分析	○	○
NF7201A NF7202A モニタリングマネージャ	○	×

ファイアーウォールなどのセキュリティ設定を行っている場合は，以下のサービスが通信できるように設定を変更してください。

表 7.2-2　サービスごとのポート番号

ポート番号	TCP/UDP	サービス名	備考
23	TCP	telnet	telnet 接続
22	TCP	ssh	SSH 接続
1812	UDP	radius	RADIUS 認証
69	UDP	tftp	TFTP 接続
21	TCP	ftp	FTP 制御
20	TCP	ftp	FTP データ転送
514	UDP	syslog	SYSLOG 送信
123	UDP	ntp	SNTP クライアント機能
161	UDP	snmp	SNMP 監視
162	UDP	snmptrap	SNMP TRAP 送信
80	TCP	http	WebAPI，WebGUI，トラフィック分析
443	TCP	https	WebAPI，WebGUI，トラフィック分析
6653	TCP	openflow	OpenFlow 接続（デフォルト値）
51967	TCP	－	モニタリングマネージャとの接続

（注 1）
Ethernet ポートと Network ポートのどちらか一方でのみ通信を行うことができます。

（注 2）
Ethernet ポート経由の場合，VLAN Tag なしパケットのみ通信を行うことができます。

（注 3）
Network ポート経由の場合，システムインタフェースへの通信中は Network ポートの帯域を使用します。ネットワーク上を流れるトラフィックをコントロールするための帯域を割り当てるときは，システムインタフェース通信の帯域も考慮して設定してください。トラフィックコントロールの設定の説明は「第 8 章 トラフィックコントロール機能」を参照してください。システムインタフェースからの出力トラフィック転送動作は，ソフトウェアバージョンおよびシナリオのツリーモード設定により異なります。

ソフトウェアバージョン Ver.1.3.1 以前は，以下の動作となります。

・シナリオのツリーモード設定：inbound（デフォルト）
システムインタフェースからの出力トラフィックは，出力ポート側とは逆側ポートのポートシナリオ（ポート 1/1 と 1/2 にチャネル設定され，ポート 1/1 への出力なら“/port2”シナリオ）の帯域を使用し，最優先のクラス 1 を割り当てています。また，当該シナリオの入出力カウンタにも加算されます。入力トラフィックについてはシナリオの帯域は使用せず，シナリオカウンタにも加算されません。

図 7.2-1　inbound 設定時の動作(Ver1.3.1 以前)

表 7.2-3　inbound 設定時の動作(Ver1.3.1 以前)

システムインタフェース	シナリオカウンタ		シナリオ動作
	受信	送信	
入力トラフィック	×	×	シナリオの帯域を使用しません。
出力トラフィック	○	○	出力ポートとは逆側ポートのポートシナリオの帯域を使用し，最優先のクラス 1 を割り当てています。

○：カウント対象，×：カウント対象外

・シナリオのツリーモード設定：outbound

システムインタフェースからの出力トラフィックは，出力ポート側のポートシナリオ（ポート 1/1 への出力なら“/port1”シナリオ）の帯域を使用し，最優先のクラス 1 を割り当てています。また，当該シナリオの入出力カウンタにも加算されます。入力トラフィックについてはシナリオの帯域は使用せず，シナリオカウンタにも加算されません。

図 7.2-2　outbound 設定時の動作(Ver1.3.1 以前)

表 7.2-4　outbound 設定時の動作(Ver1.3.1 以前)

システムインタフェース	シナリオカウンタ		シナリオ動作
	受信	送信	
入力トラフィック	×	×	シナリオの帯域を使用しません。
出力トラフィック	○	○	出力ポートとは逆側ポートのポートシナリオの帯域を使用し，最優先のクラス 1 を割り当てています。

○：カウント対象，×：カウント対象外

ソフトウェアバージョン Ver.1.5.1 以降は，set ip system network port scenario コマンドで，従来のポートシナリオから最優先の転送，またはシステムインタフェースの通信に該当するフィルタとシナリオ（第 2 階層から第 4 階層）を設定している場合は，当該シナリオで制御とするかを選択可能です。デフォルト値は”disable”で従来通りの（ソフトウェアバージョン Ver.1.3.1 以前と同じ）転送動作です。
トラフィック分析機能のトラフィック生成機能を使用する場合，同コマンドを”enable”にしてください。
”enable”時の転送動作は以下の通りです。

・シナリオのツリーモード設定：inbound
システムインタフェースからの出力トラフィックは，出力ポートとは逆側ポートのフィルタ条件に一致するシナリオの帯域を使用します。また，当該シナリオの入出力カウンタにも加算されます。システムインタフェースへの入力トラフィックは，シナリオの帯域は使用せず，シナリオカウンタは受信のみ加算されます。
なお，フィルタ条件に一致しない場合は，従来通りの転送動作です。

図 7.2-3　inbound 設定時の動作(Ver1.5.1 以降)

表 7.2-5　inbound 設定時の動作(Ver1.5.1 以降)

システムインタフェース	シナリオカウンタ		シナリオ動作
	受信	送信	
入力トラフィック	○	×	シナリオの帯域を使用しません。
出力トラフィック	○	○	出力ポートとは逆側ポートのフィルタ条件に一致するシナリオの帯域を使用します。 フィルタ条件に一致しない場合，出力ポートとは逆側ポートのポートシナリオの帯域を使用し，最優先のクラス 1 を割り当てています。

○：カウント対象，×：カウント対象外

・シナリオのツリーモード設定：outbound
システムインタフェースからの出力トラフィックは，出力ポート側のフィルタ条件に一致するシナリオの帯域を使用します。また，当該シナリオの入出力カウンタにも加算されます。システムインタフェースへの入力トラフィックは，シナリオの帯域は使用せず，シナリオカウンタは受信のみ加算されます。
なお，フィルタ条件に一致しない場合は，従来通りの転送動作です。

図 7.2-4　outbound 設定時の動作(Ver1.5.1 以降)

表 7.2-6　outbound 設定時の動作(Ver1.5.1 以降)

システムインタフェース	シナリオカウンタ		シナリオ動作
	受信	送信	
入力トラフィック	○※1	×	シナリオの帯域を使用しません。
出力トラフィック	○	○	出力ポート側のフィルタ条件に一致するシナリオの帯域を使用します。 フィルタ条件に一致しない場合，出力ポート側のポートシナリオの帯域を使用し，最優先のクラス 1 を割り当てています。

○：カウント対象，×：カウント対象外
※1 出力ポートとは逆側のポートがリンクアップしていない場合，入力トラフィックに関する受信シナリオカウンタは，カウントしません。

ソフトウェアバージョン Ver.1.4.1 のみ，システムインタフェースからの出力トラフィックについて，従来のポートシナリオから最優先の転送，またはシステムインタフェースの通信に該当するシナリオで制御とするかは選択できません。システムインタフェースの通信に該当するフィルタとシナリオ（第 2 階層から第 4 階層）を設定している場合は，当該シナリオで制御されます。
なお，フィルタ条件に一致しない場合は，従来通りの転送動作です。

システムインタフェースの設定には以下のコマンドを使用します。

表 7.2-7　システムインタフェース CLI コマンド

set ip system <IP_address> netmask <netmask> [{up | down}]	システムインタフェースの IP アドレスを設定します。 IPv4 アドレスのデフォルト値は 192.168.1.1 です。サブネットマスクのデフォルト値は 255.255.255.0 です。 IPv6 アドレスのデフォルト値は::192.168.1.1（::C0A8:101）です。プレフィックス長のデフォルト値は 64 です。
set ip system port ethernet set ip system port network in {<slot/port> | all} vid {<VID> | none} [tpid <tpid>] inner-vid {<VID> | none} [inner-tpid <tpid>]	システムインタフェースの通信ポート（Ethernet ポート／Network ポート）を設定します。 また，システムインタフェースへの通信ポートとして Network ポートを指定した場合は，以下の内容も設定します。 - Network ポート識別番号（1/1，1/2，1/3，1/4，all） - VLAN ID（0～4094／none），出力 Tag Protocol ID - Inner-VLAN ID（0～4094／none），出力 Tag Protocol ID Network ポート識別番号のデフォルト値は“all”（すべての Network ポート）です。VLAN ID および Inner-VLAN ID のデフォルト値は“none”（VLAN Tag なしパケット通信）です。出力 Tag Protocol ID は VLAN Tag ありパケットの通信または二重 VLAN Tag ありパケットの通信を行う場合で，システムインタフェースが送信するパケットの Tag Protocol ID を指定するときに設定します。デフォルトではいずれも 0x8100 を使用します。 通信ポートのデフォルト値は Ethernet ポートです。
set ip system port network scenario {enable | disable}	システムインタフェースの通信ポート設定が Network ポート経由の場合，シナリオによるトラフィックコントロールの有効／無効を設定します。
set ip system gateway <gateway>	システムインタフェースのデフォルトゲートウェイアドレスを設定します。
unset ip system gateway <gateway>	システムインタフェースのデフォルトゲートウェイアドレスを解除します。
show ip system	システムインタフェース情報を表示します。

システムインタフェースに IPv4 アドレス（192.168.10.3），サブネットマスク（255.255.255.0），デフォルトゲートウェイ（192.168.10.1）を設定する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set ip system 192.168.10.3 netmask 255.255.255.0 up
PureFlow(A)> set ip system port ethernet
PureFlow(A)> set ip system gateway 192.168.10.1
```

システムインタフェースに IPv4 アドレス（192.168.10.3），サブネットマスク（255.255.255.0），通信ポート（Network ポート（1/1 のみ）），VLAN ID（10），二重 VLAN Tag なし，デフォルトゲートウェイ（192.168.10.1）を設定する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> set ip system 192.168.10.3 netmask 255.255.255.0 up
PureFlow(A)> set ip system port network in 1/1 vid 10 inner-vid none
PureFlow(A)> set ip system gateway 192.168.10.1
```

IPv6 を使用する場合でも IPv4 の場合と同様に設定します。以下に示すコマンドを実行して，IPv6 アドレス(2001:DB8::1)，プレフィックス長(32)，デフォルトゲートウェイ(2001:DB8::FE)を設定してください。
IPv6 プレフィックス長は set ip system コマンドの netmask 引数に指定します。

```
PureFlow(A)> set ip system 2001:db8::1 netmask 32 up
PureFlow(A)> set ip system gateway 2001:db8::fe
```

また，システムインタフェースでは以下のコマンドを使用して，ネットワークの疎通確認をすることができます。

表 7.2-8　ネットワーク疎通コマンド

ping <IP_address>	ICMP ECHO_REQUEST パケットを指定 IP アドレスに送信します。(IPv4／IPv6)
traceroute <IP_address>	指定した IP アドレスに到達するまでの経路を表示します。
arp –a arp –d <IP_address>	ARP エントリの内容を表示(-a)，または削除(-d)します。(IPv4 のみ)
delete ndp neighbor <IP_address>	NDP エントリの削除を行います。(IPv6 のみ)
show ndp neighbor	NDP エントリの内容を表示します。(IPv6 のみ)

IPv4 アドレス 192.168.10.100 との疎通確認を行う場合，以下に示すコマンドを実行します。

```
PureFlow(A)> ping 192.168.10.100
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.372 ms

--- 192.168.10.100 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.372/0.372/0.372/0.000 ms
PureFlow(A)> arp –a
IP address        MAC address          type
-----------------------------------------------------------------------------------
192.168.10.3      00-00-91-01-11-23    permanent publish
192.168.10.100    00-00-91-01-23-45
PureFlow(A)>
```

疎通確認失敗時は，以下のように表示します。システムインタフェースの設定，およびネットワーク接続を確認してください。

```
PureFlow(A)> ping 192.168.10.101
PING 192.168.10.101 (192.168.10.101) 56(84) bytes of data.

--- 192.168.10.101 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 100ms
PureFlow(A)>
```

IPv4 アドレス 192.168.10.101 の ARP エントリを削除する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> arp –d 192.168.10.100
PureFlow(A)> arp –a
IP address          MAC address                 type
-----------------------------------------------------------------------------------------
192.168.10.3        00-00-91-01-11-23           permanent publish
PureFlow(A)>
```

IPv6 アドレス 2001:DB8::1 との疎通確認を行う場合，以下に示すコマンドを実行します。

```
PureFlow(A)> ping 2001:db8::1
PING 2001:db8::1 (2001:db8::1) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.372 ms

--- 2001:db8::1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.372/0.372/0.372/0.000 ms
PureFlow(A)> show ndp neighbor
IP address                  MAC address                 type
-----------------------------------------------------------------------------------------
2001:db8::1                 00-00-91-01-23-45           reachable
PureFlow(A)>
```

疎通確認失敗時は，以下のように表示します。システムインタフェースの設定，およびネットワーク接続を確認してください。

```
PureFlow(A)> ping 2001:db8::10
PING 2001:db8::10 (2001:db8::10) 56(84) bytes of data.

--- 2001:db8::10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 100ms
PureFlow(A)>
```

IPv6 アドレス 2001:db8::10 の NDP エントリを削除する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> delete ndp neighbor 2001:db8::10
PureFlow(A)> show ndp neighbor
IP address                  MAC address                 type
-----------------------------------------------------------------------------------------
PureFlow(A)>
```

7.3　システムインタフェースフィルタ

システムインタフェースへの通信を，ホストごとなどの単位で許可するか，拒否するかを選択することができます。

システムインタフェースへの通信を識別するルールは，システムフィルタにより定義します。IP パケットの以下のフィールド，およびその組み合わせで定義します。

- 送信元 IP アドレス
- 宛先 IP アドレス
- プロトコル番号
- 送信元ポート番号（Sport）
- 宛先ポート番号（Dport）

システムインタフェースフィルタの設定には以下のコマンドを使用します。

表 7.3-1　システムインタフェースフィルタコマンド

add ip system filter	システムインタフェースのフィルタを設定します。
delete ip system filter	システムインタフェースのフィルタを削除します。
show ip system	システムインタフェース情報を表示します。

システムインタフェースに IPv4 アドレス（192.168.10.3），サブネットマスク（255.255.255.0）を設定し，IPv4 アドレス（192.168.10.100）のパソコンからのみ装置にアクセスできるようにする場合は，以下に示すコマンドを実行します。

```
PureFlow(A)> set ip system 192.168.10.3 netmask 255.255.255.0 up
PureFlow(A)> set ip system gateway 192.168.10.1
PureFlow(A)> add ip system filter 20 sip 192.168.10.100 permit
PureFlow(A)> add ip system filter 30 deny
```

システムインタフェースフィルタをすべて解除する場合は，以下に示すコマンドを実行します。

```
PureFlow(A)> delete ip system filter all
```

システムインタフェースフィルタの 30 を解除する場合は，以下に示すコマンドを実行します。

```
PureFlow(A)> delete ip system filter 30
```

（注意）
システムインタフェースフィルタは十分に気をつけて設定してください。
機能を有効にする場合は permit を初めに設定し，そのあとに deny の設定を行ってください。機能を削除する場合は，deny を初めに削除し，そのあと permit の削除を行ってください。または，delete ip system filter all コマンドですべてのフィルタを削除してください。

7.4　コンフィギュレーション例

以下のネットワーク環境において，遠隔による保守／監視を行う場合のコンフィギュレーション例を示します。

[Case 1]ローカルネットワークから Ethernet ポートを経由して保守／監視を行う

- 本社内のローカルネットワークは 192.168.10.0/255.255.255.0 です。
- システムインタフェースの IPv4 アドレス 192.168.10.100，サブネットマスク 255.255.255.0 です。
- システムインタフェースのデフォルトゲートウェイアドレス 192.168.10.1 です。
- 保守用端末（CLI，ダウンロード／アップロード）の IPv4 アドレス 192.168.10.5 です。
- 監視用端末（SNMP，Syslog）の IPv4 アドレス 192.168.10.6 です。
- SNTP サーバの IPv4 アドレス 192.168.10.7 です。

図 7.4-1　Ethernet ポート経由での保守／監視例

以下のコマンドを実行します。

＜システムインタフェース設定＞

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system gateway 192.168.10.1
```

＜SNMP ホスト設定＞

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community honsya_system_management view All
PureFlow(A)> add snmp host 192.168.10.6 version v2c
                community honsya_system_management trap
```

＜Syslog ホスト設定＞

```
PureFlow(A)> add syslog host 192.168.10.6
PureFlow(A)> set syslog host enable
```

＜SNTP サーバ設定＞

```
PureFlow(A)> set sntp server 192.168.10.7
PureFlow(A)> set sntp enable
```

［Case 2］広域イーサネット／IP-VPNのネットワークとローカルネットワークから Networkポートを経由して保守／監視を行う（VLAN Tagありパケット通信）

- 拠点 A へのネットワークは VLAN ID 10 です。
- 保守監視センタへのネットワークは VLAN ID 20 です。
- システムインタフェースの IPv4 アドレス 192.168.20.100，サブネットマスク 255.255.255.0 です。
- システムインタフェースのデフォルトゲートウェイアドレス 192.168.20.1 です。
- すべての Network ポートからシステムインタフェースへの通信を行います。
- 保守用端末（CLI，ダウンロード／アップロード）の IPv4 アドレス 192.168.20.5，192.168.20.200 です。
- 監視用端末（SNMP，Syslog）の IPv4 アドレス 192.168.20.6 です。
- SNTP サーバの IPv4 アドレス 192.168.20.7 です。

図 7.4-2　Network ポート経由での保守／監視例

以下のコマンドを実行します。

＜システムインタフェース設定＞

```
PureFlow(A)> set ip system 192.168.20.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system port network in all vid 20 inner-vid none
PureFlow(A)> set ip system gateway 192.168.20.1
```

＜SNMP ホスト設定＞

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community honsya_system_management view All
PureFlow(A)> add snmp host 192.168.20.6 version v2c
                community honsya_system_management trap
```

＜Syslog ホスト設定＞

```
PureFlow(A)> set syslog host ip 192.168.20.6
PureFlow(A)> set syslog host enable
```

＜SNTP サーバ設定＞

```
PureFlow(A)> set sntp server 192.168.20.7
PureFlow(A)> set sntp enable
```

［Case 3］広域イーサネット／IP-VPNのネットワークからNetworkポートを経由して保守／監視を行う（VLAN Tagなしパケット通信）

- 拠点 A へのネットワークは 192.168.2.0/24 です。
- 保守監視センタへのネットワークは 192.168.50.0/24 です。
- システムインタフェースの IPv4 アドレス 192.168.10.100，サブネットマスク 255.255.255.0 です。
- システムインタフェースのデフォルトゲートウェイアドレス 192.168.10.1 です。
- Network ポート 1/2 からのみシステムインタフェースへの通信を行います。
- 保守用端末（CLI，ダウンロード／アップロード）の IPv4 アドレス 192.168.50.5 です。
- 監視用端末（SNMP，Syslog）の IPv4 アドレス 192.168.50.6 です。
- SNTP サーバの IPv4 アドレス 192.168.50.7 です。

図 7.4-3　Network ポート経由での保守／監視例

以下のコマンドを実行します。

＜システムインタフェース設定＞

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system port network in 1/2 vid none inner-vid none
PureFlow(A)> set ip system gateway 192.168.10.1
```

＜SNMP ホスト設定＞

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community honsya_system_management view All
PureFlow(A)> add snmp host 192.168.50.6 version v2c
                community honsya_system_management trap
```

＜Syslog ホスト設定＞

```
PureFlow(A)> set syslog host ip 192.168.50.6
PureFlow(A)> set syslog host enable
```

＜SNTP サーバ設定＞

```
PureFlow(A)> set sntp server 192.168.50.7
PureFlow(A)> set sntp enable
```

［Case 4］広域イーサネット／IP-VPNのネットワークからNetworkポートを経由して保守／監視を行う（特定ネットワークからのアクセスのみ許可）

- 拠点 A へのネットワークは 192.168.2.0/24 です。
- 保守監視センタへのネットワークは 192.168.50.0/24 です。
- システムインタフェースの IPv4 アドレス 192.168.10.100，サブネットマスク 255.255.255.0 です。
- システムインタフェースのデフォルトゲートウェイアドレス 192.168.10.1 です。
- Network ポート 1/2 からのみシステムインタフェースへの通信を行います。
- 保守用端末（CLI，ダウンロード／アップロード）の IPv4 アドレス 192.168.50.5 です。
- 監視用端末（SNMP，Syslog）の IPv4 アドレス 192.168.50.6 です。
- SNTP サーバの IPv4 アドレス 192.168.50.7 です。
- システムインタフェースへの通信は，保守監視センタからのみ許可します。

図 7.4-4　Network ポート経由での保守／監視例

以下のコマンドを実行します。

＜システムインタフェース設定＞

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system port network in 1/2 vid none inner-vid none
PureFlow(A)> set ip system gateway 192.168.10.1
```

＜システムインタフェースフィルタ設定＞

```
PureFlow(A)> add ip system filter 10 sip 192.168.50.0/255.255.255.0 permit
PureFlow(A)> add ip system filter 20 deny
```

＜SNMP ホスト設定＞

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community honsya_system_management view All
PureFlow(A)> add snmp host 192.168.50.6 version v2c
                    community honsya_system_management trap
```

＜Syslog ホスト設定＞

PureFlow(A)> set syslog host ip 192.168.50.6

PureFlow(A)> set syslog host enable

＜SNTP サーバ設定＞

PureFlow(A)> set sntp server 192.168.50.7

PureFlow(A)> set sntp enable

［Case 5］広域イーサネット／IP-VPNのネットワーク（Networkポート経由）と ローカルネットワーク（Ethernetポート経由）から保守／監視を行う

- 本社内のローカルネットワークは 192.168.10.0/24 です。
- 拠点 A へのネットワークは 192.168.2.0/24 です。
- 保守監視センタへのネットワークは 192.168.50.0/24 です。
- システムインタフェースの IPv4 アドレス 192.168.10.100，サブネットマスク 255.255.255.0 です。
- システムインタフェースのデフォルトゲートウェイアドレス 192.168.10.1 です。
- 保守用端末（CLI，ダウンロード／アップロード）の IPv4 アドレス 192.168.50.5，192.168.10.5 です。
- 監視用端末（SNMP，Syslog）の IPv4 アドレス 192.168.50.6 です。
- SNTP サーバの IPv4 アドレス 192.168.50.7 です。

図 7.4-5　Ethernet ポート経由での保守／監視例（Network ポート経由）

以下のコマンドを実行します。

＜システムインタフェース設定＞

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system gateway 192.168.10.1
```

＜SNMP ホスト設定＞

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community honsya_system_management view All
PureFlow(A)> add snmp host 192.168.50.6 version v2c
             community honsya_system_management trap
```

＜Syslog ホスト設定＞

```
PureFlow(A)> set syslog host ip 192.168.50.6
PureFlow(A)> set syslog host enable
```

＜SNTP サーバ設定＞

```
PureFlow(A)> set sntp server 192.168.50.7
PureFlow(A)> set sntp enable
```

［Case 6］特定の端末からEthernetポートを経由して保守／監視を行う 不特定の端末からは監視を行わない

- 本社内のローカルネットワークは 192.168.10.0/255.255.255.0 です。
- システムインタフェースの IPv4 アドレス 192.168.10.100，サブネットマスク 255.255.255.0 です。
- システムインタフェースのデフォルトゲートウェイアドレス 192.168.10.1 です。
- 保守用端末（CLI，ダウンロード／アップロード）の IPv4 アドレス 192.168.10.5 です。
- 通常業務用端末の IPv4 アドレス 192.168.10.10 です。

図 7.4-6　Ethernet ポート経由での保守／監視例

以下のコマンドを実行します。

＜システムインタフェース設定＞

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system gateway 192.168.10.1
```

＜システムインタフェースフィルタ設定＞

```
PureFlow(A)> add ip system filter 10 sip 192.168.10.5 permit
PureFlow(A)> add ip system filter 20 deny
```

7.5　設定，状態の確認

システムインタフェースの設定コマンドで設定した内容を確認するには，“show ip system”コマンドを使用します。

```
PureFlow(A)> show ip system
Status            : Up
IP Address        : 192.168.10.3
Netmask           : 255.255.255.0
Broadcast         : 192.168.10.255
Default Gateway   : 192.168.10.1
IPv6 Address      : 2001:DB8::1
Prefix            : 32
Default Gateway   : 2001:DB8::FE
Port              : Network (1/2)
VID               : 20
TPID              : 0x8100
Inner-VID         : none
Inner-TPID        : ----

Number of system filter entries: 0
PureFlow(A)>
```

システムインタフェースの統計情報を確認するには，“show counter”コマンドを使用します。本コマンドで表示するカウンタ長は，32 ビットです。

```
PureFlow(A)> show counter
Port        Rcv Octets      Rcv Packets     Trs Octets      Trs Packets
-------     ----------      ----------      ----------      ----------
 1/1          57566366        14194297               0               0
 1/2                 0               0        59383412        14195494
 1/3          57566366        14194297               0               0
 1/4                 0               0        59383412        14195494
system           58368             152           85424             152

Port        Rcv Broad       Rcv Multi       Trs Broad       Trs Multi
-------     ----------      ----------      ----------      ----------
 1/1             10000        14208097               0               0
 1/2                 0               0           10000        14209615
 1/3             10000        14208097               0               0
 1/4                 0               0           10000        14209615
system             N/A             N/A             N/A             N/A

Port        Err Packets     Collision       Discard
-------     ----------      ----------      ----------
 1/1                 0               0               0
 1/2                 0               0               0
 1/3                 0               0               0
 1/4                 0               0               0
system             N/A             N/A             N/A
```

また，システムインタフェースをコマンド引数に指定することにより，詳細内容を表示できます。本コマンドで表示するカウンタ長は，64 ビットです。“show counter”コマンドの 32 ビットカウンタがラップアラウンドした場合，“show counter system”コマンドの 64 ビットカウンタと異なる値が表示されることに注意してください。

```
PureFlow(A)> show counter system
Rcv Packets                    152
Rcv Broad                      N/A
Rcv Multi                      N/A
Rcv Octets                     58368
Rcv Rate                       N/A
Trs Packets                    152
Trs Broad                      N/A
Trs Multi                      N/A
Trs Octets                     85424
Trs Rate                       N/A
Collision                      N/A
Drop                           N/A
Discard                        N/A
Error Packets                  N/A
        CRC Align Error                N/A
        Undersize Packet               N/A
        Oversize Packet                N/A
        Fragments                      N/A
        Jabbers                        N/A
```

（空白ページ）

第8章　トラフィックコントロール機能

ここでは，トラフィックコントロール機能と設定について説明します。

8.1　概要 8-2
8.2　トラフィックシェーピング 8-4
8.3　トラフィックアクセラレーション 8-5
8.4　大規模ネットワークへの適用 8-6
8.5　チャネル 8-7
8.6　シナリオ 8-9
　8.6.1　トラフィックアトリビュート 8-10
　8.6.2　フィルタ 8-11
8.7　階層化シナリオ 8-13
　8.7.1　フィルタの階層関係 8-15
　8.7.2　フィルタとシナリオの関係 8-16
　8.7.3　ルールリスト 8-18
8.8　アクセラレーショントンネル 8-19
8.9　設定方法 8-21
8.10　ルールリストの設定方法 8-42
8.11　チャネルインタフェース通信 8-45
8.12　アプリケーション高速化機能 8-48
　8.12.1 SMB プロトコル高速化機能 8-49
　8.12.2 SMB プロトコル高速化機能の注意点 8-51
8.13　コンフィギュレーション例 8-52
8.14　さらに高度な設定 8-66
　8.14.1 フローおよびフロー識別モード 8-67
　8.14.2 キュー 8-71
　8.14.3 通信ギャップモード 8-80
　8.14.4 ピークバーストサイズ 8-82
　8.14.5 トラフィックアクセラレーションバイパス 8-84
　8.14.6 トラフィックアクセラレーションの冗長構成 8-88
　8.14.7 TCP-FEC 機能 8-95
　8.14.8 TCP 輻輳制御機能 8-98
　8.14.9 リマーキング機能 8-100
8.15　トラフィックアクセラレーション実行時のアドレス 8-105

8.1　概要

従来の専用線や ATM 回線に変わり，より高速で低コストの IP-VPN や広域イーサネットサービスにより拠点間を接続する形態が普及してきました。専用線や ATM 回線と異なり，IP-VPN や広域イーサネットは QoS が保証されないパケット交換網を使用します。IP-VPN や広域イーサネットの回線は，通信事業者から提供され，回線速度や最大帯域が規定されていますが，特定のユーザやアプリケーションが回線の帯域を多く占有してしまうと，その他のユーザやアプリケーションが利用できる回線帯域が不足したり，通信遅延が発生したりするなどの障害が起こります。

このような通信品質の劣化は，音声通信や TV 会議などのミッションクリティカルな業務効率を低下させ，重大な支障をきたすことにつながります。こうしたミッションクリティカルなトラフィックを回線帯域不足や通信遅延から守るために，回線帯域を拠点やユーザ，またはアプリケーションごとに分割し，必要な帯域を割り当てたり，トラフィックの優先制御を行ったりする必要があります。回線帯域を分割し，割り当てた帯域に対して最低帯域を保証したり，最大帯域制限を行ったりすることをトラフィックコントロールと呼びます。

大規模な企業ネットワークでは，トラフィックコントロールを拠点やユーザ，またはアプリケーションごとに複雑に組み合わせる必要があります。たとえば，特定ユーザ（拠点 A）に 2 Mbps の帯域を割り当て，さらにその帯域内で VoIP に 70 kbps の帯域を保証するといった階層的なトラフィックコントロールが必要とされます。本装置はトラフィックシェーピング機能を実装しており，回線帯域を分割し，必要な帯域を割り当て，さらにその帯域内で帯域を再分割することが可能です。

図 8.1-1　大規模な企業ネットワーク

また，本装置は，トラフィックアクセラレーション機能を実装しており，WAN など長距離回線による遅延の影響を受けない高速なデータ通信を可能にします。
今日のデータセンタは，機器コストや運用コストを削減し，かつ，セキュリティを強化するため，データセンタにサーバとストレージをセンタ集中型に配置する需要が高まっています。一方で，災害時においてもサーバのデータを確実に復旧するため，バックアップデータを遠隔地に転送する需要が高まっています。しかし，多くのデータ通信で使用する TCP/IP は，回線遅延により転送速度が低下します。

本装置のトラフィックアクセラレーション機能は，回線遅延による TCP/IP のデータ転送速度の低下を抑え，高速なデータ通信を実現します。
また，本装置は，トラフィックシェーピング機能も持ち，一度に複数のサーバやクライアントから送信されるバーストトラフィックを平滑化し，ネットワーク中に配置されたルータやスイッチなどでのパケット廃棄を防止します。

図 8.1-2　長距離データセンタ間通信

8.2　トラフィックシェーピング

本機能は，一度に複数のサーバやクライアントから送信されるバーストトラフィックを平滑化し，ネットワーク中に配置されたルータやスイッチなどでのパケット廃棄を防止します。これにより，高速でかつ安定したネットワーク通信を可能にします。トラフィックシェーピングの IP バージョンは，IPv4 および IPv6 をサポートします。

図 8.2-1　トラフィックシェーピング

8.3　トラフィックアクセラレーション

本機能は，長距離間の TCP/IP のデータ通信を高速化します。長距離間のデータ転送においては，伝送路の物理的距離が長く，かつ，ルータやスイッチなどの中継装置が多くなるため，送信側のサーバがパケットを送信してから，受信側のクライアントがパケット受信するまでの遅延時間が大きくなります。多くのサーバ，クライアントが使用する TCP/IP プロトコルは，遅延が大きい場合，データ転送速度が低下します。本装置は，数十ミリの遅延がある通信回線を経由した場合においても，データ転送速度の低下を抑え，高速なデータ転送を可能にします。

また，本装置は，TCP-FEC 機能，TCP 輻輳制御機能により，WAN 回線等のパケット廃棄率の高い回線を利用した場合においても，データ転送速度の低下を抑え，高速なデータ転送を可能にします。

TCP-FEC 機能については，「8.14.7　TCP-FEC 機能」を参照してください。

TCP 輻輳制御機能については，「8.14.8　TCP 輻輳制御機能」を参照してください。

トラフィックアクセラレーションの IP バージョンは，IPv4 および IPv6 をサポートします。

図 8.3-1　トラフィックアクセラレーション

注:

トラフィックアクセラレーション機能，TCP-FEC機能，TCP輻輳制御機能を使用する場合は，「TCP 高速化機能ライセンス(NF7500-L201A)」を購入していただく必要があります。また TCP-FEC 機能を使用する場合は，別途「FEC 機能ライセンス(NF7500-L211A)」を購入していただく必要があります。

8.4　大規模ネットワークへの適用

本装置は，多数拠点を持つ大規模な企業基幹ネットワークや，複数の企業向けにクラウドサービスを提供するような大規模なネットワークに適用可能です。トラフィックを階層的にグループ化し，グループ単位で管理できますので，運用が容易です。

たとえば，トラフィックを企業単位でグループ化し，さらに拠点単位に細分化し，さらにアプリ（サービス）やユーザ単位に細分化します。さらに，それぞれのグループに対し，トラフィックシェーピングおよびトラフィックアクセラレーションが可能です。また，会社や拠点，アプリやユーザといった任意のグループごとにトラフィックを階層的に分類し，トラフィックシェーピングやトラフィックアクセラレーションを行うことも可能です。

図 8.4-1　大規模ネットワークへの適用

8.5　チャネル

本装置は，Network ポートを 4 つ持ち，任意の 2 ポート間でブリッジ動作を行います。ブリッジ動作を行う 2 ポート間の組み合わせを「チャネル」と呼びます。チャネルにより LAN 側と WAN 側のネットワークを接続するため，チャネルに対して LAN 側ポートと WAN 側ポートを指定する必要があります。

図 8.5-1　チャネル設定

4 つの Network ポートは，2 ポートごとをチャネルとして自由に組み合わせることが可能ですが，特に理由がない場合は 1/1 と 1/2 または 1/3 と 1/4 をチャネルの組み合わせとして使用してください。

チャネルには，トラフィックアクセラレーションで使用する IP ネットワークインタフェース（チャネルインタフェース）の装置 IP アドレスを設定します。本装置は，このチャネルインタフェース経由で TCP/IP のデータ通信を高速化するためのアクセラレーショントンネルを構成します。アクセラレーショントンネルの詳細は，「8.8 アクセラレーショントンネル」を参照してください。
また，WAN 回線が VLAN を使用する広域 Ethernet 回線の場合，VLAN ごとに異なるチャネルと装置 IP アドレスが必要になります。チャネルには設定した VLAN に該当したフローを転送する通常チャネルと，通常チャネルに該当しないフローを転送するデフォルトチャネルの 2 種類があります。通常チャネルは，トラフィックアクセラレーション機能を使用するときのみ作成します。トラフィックシェーピング（帯域制御）ではデフォルトチャネルのみ作成します。

図 8.5-2　VLAN ごとのチャネル設定

8.6　シナリオ

本装置は，ネットワークを流れるトラフィックに対して実行する制御をシナリオと呼ぶ単位で指定します。シナリオには，トラフィックを分類する条件を記述したフィルタと，分類されたトラフィックに対する制御を指定するトラフィックアトリビュートを指定します。

本装置は，通過するパケットをフィルタルールにより分類し，トラフィックをグループ化します。グループ化されたトラフィックは，シナリオと呼ばれるトラフィックアトリビュートに従ってトラフィックコントロールします。

1 つのシナリオに複数のフィルタを設定することが可能です。

図 8.6-1　トラフィックシェーピングでのシナリオ設定

図 8.6-2　トラフィックアクセラレーションでのシナリオ設定

8.6.1 トラフィックアトリビュート

トラフィックアトリビュートには，ネットワークを流れるトラフィックに対して実行するトラフィックコントロールの種別（アクセラレーション，シェーピング）とトラフィックに対する制御パラメータを指定します。

トラフィックアトリビュートの動作（アクション）には，

(1)　アクセラレーションモード（Wan-accel モード）
(2)　集約キューモード（Aggregate モード）
(3)　個別キューモード（Individual モード）
(4)　パケットを廃棄する廃棄モード（Discard モード）
(5)　転送モード（Forward モード）

のモードがあります。

アクセラレーションモード（Wan-accel モード）は，フィルタに一致したトラフィックのうち TCP トラフィックに対し，トラフィックアクセラレーションを実行し，TCP 通信を高速化します。TCP 以外のトラフィックは，受信パケットをそのまま転送します。
※アクセラレーションモード（Wan-accel モード）を使用する場合は，「TCP 高速化機能ライセンス(NF7500-L201A)」を購入していただく必要があります。

集約キューモード（Aggregate モード）は，フィルタに一致したトラフィックをひとつの固まりとして通信帯域をコントロールします。

個別キューモード（Individual モード）は，フィルタに一致したトラフィックをさらに個々のフロー（装置内で識別できるトラフィックの最小単位）ごとに識別し，各フローの通信帯域をコントロールします。

廃棄モード（Discard モード）は，フィルタに一致したトラフィックを廃棄します。

転送モード（Forward モード）は，フィルタに一致したトラフィックを上位階層のシナリオに転送します。

8.6.2 フィルタ

シナリオごとにパケットを分類する条件をフィルタに設定します。フィルタには，Bridge-ctrl フレームのみを分類する“Bridge-ctrl フィルタ”，Ethernet ヘッダの length/type フィールド／VLAN Tag フィールドを分類する“Ethernet フィルタ”，VLAN Tag フィールド／IP ヘッダ／プロトコルヘッダを分類する“IP フィルタ”の 3 種類があります。

1. Bridge-ctrl フィルタとは，スパニングツリープロトコルの BPDU やリンクアグリゲーションの LACP など，スイッチのコントロール用として予約されている MAC アドレスを対象としたフィルタです。
   たとえば，スパニングツリー構築環境下において BPDU を優先，もしくは帯域確保したい場合に使用します。
   対象となる宛先 MAC アドレスは以下です。

   - 宛先 MAC アドレス　01-80-C2-00-00-00～01-80-C2-00-00-FF

2. Ethernet フィルタとは，Ethernet フレーム全般を対象としたフィルタです。
   VLAN ごとに分類したい場合，パケット種別ごとに分類したい場合に使用します。
   たとえば，VLAN のみを指定することにより VLAN ごとの帯域制御が実現できます。
   また，ARP パケットを優先，もしくは帯域確保したい場合には，Ethernet Type「0806」を指定することにより実現できます。

3. IP フィルタとは，IP パケットを対象にしたフィルタです。
   IP パケットフィールドにより IP パケットを分類したい場合に使用します。
   IP を IP フィルタにより分類する場合は，さらに以下の IP パケットフィールドの値を用いて細分化することができます。

   - VLAN ID
   - CoS
   - 送信元 IP アドレス（SIP）
   - 宛先 IP アドレス（DIP）
   - ToS またはトラフィッククラス
   - プロトコル番号
   - 送信元ポート番号（Sport）
   - 宛先ポート番号（Dport）

トラフィックをフィルタにより分類する際，適用するフィルタ種別はパケットの内容によって固定的です。宛先 MAC アドレスが 01-80-C2-00-00-XX であるフレームは，それ以外のフィールドの内容にかかわらず Bridge-ctrl フィルタのみが適用されます。宛先 MAC アドレスが 01-80-C2-00-00-XX ではなく，Ethernet Type 値が 0x0800 であるパケットは IPv4 フィルタおよび Ethernet フィルタのみが，また，0x86DD であるパケットは IPv6 フィルタおよび Ethernet フィルタのみが適用されます。上記のいずれにも該当しないパケットは Ethernet フィルタのみが適用されます。

図 8.6.2-1　フィルタ設定

8.7　階層化シナリオ

本装置は，シナリオを階層的に指定することができます。

第 1 階層（レベル 1）では，物理回線帯域を任意の帯域で制御（トラフィックシェーピング）します。第 2 階層（レベル 2）では，会社や拠点，ユーザなどのトラフィックを分類し，トラフィックアクセラレーションやトラフィックシェーピングすることができます。レベル 2 の仮想回線にトラフィックを流すことで，仮想回線ごとに回線帯域を分割し，それぞれに個別の帯域を割り当てることができます。第 3 階層（レベル 3）以降でも同様に上位レベルに割り当てた帯域を分割し，制御できます。
以下に，階層化シナリオの概念図を示します。

図 8.7-1　階層化シナリオ

レベル 1（第 1 階層）：
　レベル 1 を通過する総帯域をトラフィックシェーピングすることができます。
　レベル 1 は 1 つ，または複数のレベル 2 を集約できます。
レベル 2（第 2 階層）：
　レベル 1 のトラフィックを分類，制御します。
　トラフィックに対し，トラフィックアクセラレーションやトラフィックシェーピングすることができます。
　レベル 2 は 1 つ，または複数のレベル 3 を集約できます。
レベル 3（第 3 階層）：
　レベル 2 内の帯域を分割，制御します。
　トラフィックに対し，トラフィックアクセラレーションやトラフィックシェーピングすることができます。
　レベル 3 は 1 つ，または複数のレベル 4 を集約できます。

同様に，レベル 4（第 4 階層）まで帯域を分割，制御することができます。

注：

アクセラレーション（Wan-accel）モードと集約キュー（Aggregate）モード，個別キュー（Individual）モードのシナリオを同一階層に設定することは可能ですが，Wan-accel モードシナリオは最優先(クラス 1)で転送されるため，他のシナリオの優先度がデフォルト(クラス 2)の場合，Wan-accel モードシナリオが帯域を占有する可能性があります。このため，Wan-accel モードシナリオには，最大帯域を設定することを推奨します。また，Wan-accel モードシナリオに最大帯域や最低帯域の設定も可能で

すが，最低帯域を設定される場合，同一階層のその他シナリオの最低帯域を保証するため，その他シナリオの優先度を最優先(クラス 1)に設定することを推奨します。

8.7.1　フィルタの階層関係

各シナリオのフィルタは，上位レベルシナリオのフィルタ条件を継承し，階層的にパケットを分類します。

上位レベルシナリオのフィルタ条件に一致し，下位レベルシナリオのフィルタ条件にも一致するトラフィックは，下位レベルシナリオのトラフィックとして分類します。上位レベルシナリオのフィルタ条件に一致し，下位レベルシナリオのフィルタ条件には一致しないトラフィックは，上位レベルシナリオのトラフィックとして分類され，上位レベルシナリオの空き帯域を使ってトラフィックを送出します。

以下の図は，パケットを階層的に分類した例です。レベル 2 シナリオのフィルタに，IPv4 を指定することにより IPv4 パケットと IPv4 以外のパケットを分類します。さらに，レベル 3 シナリオのフィルタに Subnet アドレスを指定することにより，SubnetA のパケットと SubnetB のパケットとそのほかの Subnet の IPv4 パケットに分類します。続いて，レベル 4 シナリオのフィルタに ProtocolTCP を指定することにより，SubnetB の TCP パケットと TCP 以外のパケットに分類します。

図 8.7.1-1　フィルタの階層関係

8.7.2 フィルタとシナリオの関係

本装置は，物理回線内に流れるパケットをフィルタで分類し，トラフィックを抽出します。抽出したトラフィックを帯域，バッファサイズなどのトラフィックアトリビュートに従ってトラフィックコントロール転送します。

図 8.7.2-1　フィルタとシナリオの関係

上図は，フィルタとシナリオの設定と，実際のトラフィックコントロール動作の関係を示した概念図です。

レベル 1 からレベル 4 での帯域制御，およびレベル 2 からレベル 4 でのフィルタ設定による廃棄，転送が制御できます。
また，レベル 2 からレベル 4 でのトラフィックアクセラレーションが可能です。
フィルタの動作は，“aggregate”，“individual”，“discard”，“wan-accel”，“forward”の指定が可能です。
フィルタルールに一致したパケットは，フィルタに設定した動作に従います。
装置はパケットを受信すると，レベル 2 のフィルタにてフィルタ優先度の高い順からフィルタルールに一致するかどうか調べます。

レベル 2 フィルタルールに一致すると，フィルタに関連付けされているレベル 2 シナリオの動作が“aggregate”ならばそのシナリオで指定されたトラフィックアトリビュートに従ってパケットを転送します。また，そのシナリオに関連付けされているレベル 3 シナリオのレベル 3 フィルタにてフィルタ優先度の高い順からフィルタルールに一致するかどうか調べます。
“individual”ならばそのシナリオで指定されたトラフィックアトリビュートに従ってパケットを転送します。
“individual”シナリオの下位レベルにシナリオおよびフィルタを登録できますが，“individual”シナリオより下位レベルのフィルタ検索は行いません。“individual”シナリオより下位レベルのシナリオでパケットが転送されることはなく，無効となります。
“discard”の場合，パケットを廃棄します。“discard”シナリオの下位レベルにシナリオおよびフィルタを登録できますが，“discard”シナリオより下位レベルのフィルタ検索は行いません。“discard”シナリオより下位レベルのシナリオでパケットが転送されることはなく，無効となります。
“wan-accel”の場合，トラフィックアクセラレーションを実行します。“wan-accel”シナリオの下位レベルにシナリオおよびフィルタを登録できますが，“wan-accel”シナリオより下位レベルのフィルタ検索は行いません。
“wan-accel”シナリオより下位レベルのシナリオでパケットが転送されることはなく，無効となります。
“forward”の場合，上位階層のシナリオに転送します。
レベル 3 から 4 までのフィルタに関しても同じです。

フィルタには，Bridge-ctrl フィルタ，Ethernet フィルタ，IP フィルタを指定できます。各フィルタとも任意の文字列でフィルタ名を指定します。全フィルタ合計で 10000 個のフィルタルールを設定できます。

また，各フィルタルールには優先度を設定できます。
シナリオに関連付けされた同一レベルのフィルタ群のうち，一致するフィルタルールが複数ある場合，どのフィルタが適用されるかをフィルタ優先度に従って決定します。フィルタ優先度は値が小さいほど優先度が高くなります。

図 8.7.2-2　優先度によるフィルタ検索順序

なお，一致するフィルタルールの優先度が同じである場合，どのフィルタが適用されるかは任意となります。複数のフィルタルールに一致するようなフィルタ構成を行う場合，フィルタ優先度を調整して適用されるフィルタを明確にすることを推奨します。フィルタ優先度を指定しない場合，優先度 20000 が自動的に設定されます。

8.7.3　ルールリスト

ルールリストは，複数のトラフィック分類条件（IP アドレス，ポート番号，ドメイン名[注 1]）をグループ化する機能です。これにより，複数のトラフィック分類条件を単一のルールリスト名で指定できます。

ルールリスト名をフィルタ追加コマンドの引数に指定することで，トラフィック分類条件として設定できます。

ルールリストに指定可能なトラフィック分類条件は，以下のとおりです。

① IPv4 アドレス　　：IP アドレス／アドレスマスク
② IPv6 アドレス　　：IP アドレス／アドレスマスク
③ L4 ポート番号　　：ポート番号範囲
④ ドメイン　　：ドメイン名

ルールリストは複数のフィルタで繰り返し指定できます。ルールリストを使用することでフィルタ数や設定行数を削減できます。

図 8.7.3-1　ルールリストとフィルタの関係

上図は，ルールリストの設定と，実際のトラフィックコントロール動作の関係を示した概念図です。この概念図では，ルールリスト 1 に，複数の TCP/UDP ポート番号を登録しておき，拠点 1 アプリ群仮想回線と拠点 2 アプリ群仮想回線のフィルタ設定コマンドにおいて，sport（送信元ポート番号）のパラメータとして利用しています。

（注 1）
ルールリストにドメインを指定する場合は，「ドメインフィルタ機能ライセンス(NF7500-L141A)」を購入していただく必要があります。

8.8　アクセラレーショントンネル

本装置は，シナリオのトラフィックアトリビュートでアクセラレーションモードを選択することにより，トラフィックアクセラレーションを行います。

トラフィックアクセラレーションを行うためには，WAN 回線を経由した対向装置間でアクセラレーショントンネルを構成する必要があります。アクセラレーショントンネルを構成するためには，対向装置の IP アドレス／TCP 接続ポート番号／VLAN をシナリオに指定します。この対向装置と，自装置に設定したチャネルインタフェースの装置 IP アドレス／VLAN の間でアクセラレーショントンネルを構成します。

図 8.8-1　アクセラレーショントンネル

また，WAN 回線が VLAN を使用する広域 Ethernet 回線の場合，VLAN ごとのネットワークでトラフィックアクセラレーションが可能です。この場合，VLAN ごとに自装置および対向装置の IP アドレスが必要です。

図 8.8-2　VLAN ごとのアクセラレーショントンネル

また，本装置を複数拠点に設置し，複数拠点間でトラフィックアクセラレーションを行うことが可能です。この場合，宛先拠点ごとにシナリオを作成し，シナリオで対向装置の IP アドレスを指定します。

図 8.8-3　複数拠点間のアクセラレーショントンネル

アクセラレーショントンネルは，自装置の IP アドレス／VLAN（チャネルインタフェース）と対向装置の IP アドレス／VLAN（シナリオ）の組み合わせで構成されます。自装置と対向装置のチャネルインタフェースおよびシナリオの VLAN が異なる場合，アクセラレーショントンネルは構成されません。また，対向装置のシステムインタフェース通信がネットワークポート経由の場合，システムインタフェース宛ての通信はトラフィックアクセラレーションバイパスに切り替わります。

図 8.8-4　アクセラレーショントンネルとシナリオの関係

複数の VLAN を用いてトンネルを構成する場合は，シナリオの VLAN によりチャネルが決定されます。

図 8.8-5　複数 VLAN でのアクセラレーショントンネルとシナリオの関係

8.9　設定方法

設定方法の流れをまとめると下図のようになります。

図 8.9-1　設定方法の流れ

次に，流れに沿って設定方法を説明します。

STEP 1：チャネルの設定

本装置では，トラフィックコントロールを行うため，チャネル登録により LAN 側の Network ポートと WAN 側の Network ポートおよび VLAN を指定します。本設定は，トラフィックコントロールを動作させる場合に必要です。

また，WAN 回線が VLAN を使用する広域 Ethernet 回線の場合，VLAN ごとに異なるチャネルが必要になります。チャネルには VLAN に該当したフローを転送する通常チャネルと，該当しないフローを転送するデフォルトチャネルの 2 種類があります。

チャネル登録のパラメータを以下に示します。

表 8.9-1　チャネルのパラメータ

パラメータ	設定範囲	省略可能／不可	説明
チャネル名 (channel_name)	“xxxxxxxx”	不可	チャネル名を指定します。 設定範囲は 1～32 文字です。
LAN 側ポート (slot/port)	1/1，1/2，1/3，1/4	不可	LAN 側の Network ポートを指定します。スロット位置は 1 固定です。
WAN 側ポート (slot/port)	1/1，1/2，1/3，1/4	不可	WAN 側の Network ポートを指定します。スロット位置は 1 固定です。
VLAN ID (VID， none)	1～4094 none	不可	チャネルの VLAN ID を指定します。 VLAN Tag なしフレームをトラフィックアクセラレーションする場合は，“none”を指定します。
TPID (tpid)	0x8100，0x88a8，0x9100，0x9200，0x9300	可能	チャネルの TPID（Tag Protocol Identifier）を指定します。
Inner-VLAN ID (VID， none)	1～4094 none	可能	チャネルの Inner-VLAN ID を指定します。 Inner-VLAN Tag なしフレームをトラフィックアクセラレーションする場合は，“none”を指定します。
Inner-TPID (tpid)	0x8100，0x88a8，0x9100，0x9200，0x9300	可能	チャネルの Inner-TPID（Tag Protocol Identifier）を指定します。
MTU (mtu)	300～10200[Byte]	可能	チャネルの MTU（Maximum Transmission Unit）を指定します。 本パラメータは，LAN 側と WAN 側の Network ポートの両方に適用します。
チャネルタイプ (default)	default	可能	デフォルトチャネルを登録する場合は“default”を指定します。

以下に，チャネル設定に関する CLI コマンドを示します。

表 8.9-2　チャネル設定に関する CLI コマンド

add channel <channel_name> lan {<slot/port>} wan {<slot/port>} vid {<VID> | none>} [tpid <tpid>] [inner-vid {<VID> | none}] [inner-tpid <tpid>] [mtu <mtu>]	通常チャネルを登録します。 VLAN ごとに登録します。
add channel <channel_name> lan {<slot/port>} wan {<slot/port>} default	デフォルトチャネルを登録します。 通常チャネルに該当しないフローを転送するために登録します。
delete channel all	すべてのチャネルを削除します。
delete channel　<channel_name>	指定したチャネルを削除します。
show channel all	すべてのチャネル情報を表示します。
show channel name <channel_name> [next]	指定したチャネルのチャネル情報を表示します。

以下に，チャネルの設定例を示します。

Sample 1）Network ポート 1/1 を LAN 側に接続，Network ポート 1/2 を WAN 側に接続し，VLAN Tag なしフレームをチャネル名“ch1”として，通常チャネルを登録する場合

```
PureFlow(A)> add channel “ch1” lan 1/1 wan 1/2 vid none
```

Sample 2）Network ポート 1/1 を LAN 側に接続，Network ポート 1/2 を WAN 側に接続し，VLAN ID が 100 のトラフィックをチャネル名“ch2”として，通常チャネルを登録する場合

```
PureFlow(A)> add channel “ch2” lan 1/1 wan 1/2 vid 100
```

Sample 3）Network ポート 1/1 を LAN 側に接続，Network ポート 1/2 を WAN 側に接続するデフォルトチャネルを登録する場合

```
PureFlow(A)> add channel “default” lan 1/1 wan 1/2 default
```

STEP 2：チャネルの IP アドレス設定

本装置は，チャネルの IP アドレス設定により，チャネルインタフェースを作成します。自装置のチャネルインタフェースの IP アドレスと対向装置のチャネルインタフェースの IP アドレスとの間でアクセラレーショントンネル（TCP 接続ポート：10000）を構成します。本設定は，トラフィックアクセラレーションを動作させる場合に必要です。トラフィックシェーピングのみを動作させる場合，設定は不要です。

チャネルインタフェース設定のパラメータを以下に示します。

表 8.9-3　チャネルインタフェースのパラメータ

パラメータ	設定範囲	省略可能／不可	説明
チャネル名 （channel_name）	“xxxxxxxx”	不可	チャネル名を指定します。 デフォルトチャネル名は指定できません。
IP アドレス （IP_address）	IPv4 および IPv6 アドレス	不可	チャネルインタフェースの IPv4／IPv6 アドレスを指定します。 1 つのチャネルに対して IPv4 アドレスと IPv6 アドレスを同時に設定できます。
サブネットマスク／プレフィックス長 （netmask）	xxx.xxx.xxx.xxx／ 0～128	不可	チャネルインタフェースに IPv4 アドレスを指定する場合は，サブネットマスクを指定します。 チャネルインタフェースに IPv6 アドレスを指定する場合は，プレフィックス長を指定します。

以下に，チャネルインタフェースに関する CLI コマンドを示します。

表 8.9-4　チャネルインタフェースに関する CLI コマンド

set ip channel <channel_name> <IP_address> netmask <netmask>	チャネルの IP ネットワークインタフェース（チャネルインタフェース）を設定します。
unset ip channel all	すべてのチャネルインタフェースを設定解除します。
unset ip channel <channel_name> [{ipv4 | ipv6}]	指定したチャネル／IP バージョンのチャネルインタフェースを設定解除します。
show ip channel all	すべてのチャネルインタフェース情報を表示します。
show ip channel name <channel_name> [next]	指定したチャネルのチャネルインタフェース情報を表示します。

以下に，チャネルインタフェースの設定例を示します。

Sample 1）チャネル名“ch1”に対して，IPv4 アドレス 20.1.5.9，サブネットマスク 255.255.255.0 のチャネルインタフェースを設定する場合

```
PureFlow(A)> set ip channel “ch1” 20.1.5.9 netmask 255.255.255.0
```

Sample 2）チャネル名“ch1”のチャネルインタフェースを設定解除する場合

```
PureFlow(A)> unset ip channel “ch1”
```

STEP 3：IP ルートの設定

本装置は IP ルートの登録により，チャネルインタフェースのスタティック経路（デフォルト経路およびターゲット経路）を登録し，トラフィックの転送先を決定します。本設定は，トラフィックアクセラレーションを動作させる場合に必要です。

スタティック経路登録のパラメータを以下に示します。

表 8.9-5　スタティック経路のパラメータ

パラメータ	設定範囲	省略可能／不可	説明
IP アドレス（IP_address）	xxx.xxx.xxx.xxx	不可	宛先ネットワークの IPv4／IPv6 アドレスを指定します。
サブネットマスク／プレフィックス長（netmask）	xxx.xxx.xxx.xxx／0～128	不可	宛先ネットワークに IPv4 アドレスを指定する場合は，サブネットマスクを指定します。 宛先ネットワークに IPv6 アドレスを指定する場合は，プレフィックス長を指定します。
ゲートウェイアドレス（gateway）	xxx.xxx.xxx.xxx	不可	ゲートウェイの IPv4／IPv6 アドレスを指定します。
チャネル名（channel_name）	“xxxxxxxx”	不可	チャネル名を指定します。 デフォルトチャネル名は指定できません。
LAN 側経路／WAN 側経路（lan，wan）	lan， wan	不可	LAN 側のスタティック経路を登録する場合は“lan”を，WAN 側のスタティック経路を登録する場合は“wan”を指定します。

以下に，スタティック経路設定に関する CLI コマンドを示します。

表 8.9-6　スタティック経路設定に関する CLI コマンド

add route default gateway <IP_address> channel <channel_name> {lan | wan}	チャネルインタフェースのスタティック経路（デフォルト経路）を登録します。
add route target <IP_address> netmask <netmask> gateway <gateway> channel <channel_name> {lan | wan}	チャネルインタフェースのスタティック経路（ターゲット経路）を登録します。
delete route all	すべてのスタティック経路を削除します。
delete route target <IP_address> netmask <netmask> gateway <gateway> channel <channel_name> {lan | wan}	指定した宛先ネットワークのスタティック経路を削除します。
show route all	すべてのスタティック経路情報を表示します。
show route channel <channel_name>	指定したチャネルのスタティック経路情報を表示します。
show route target <IP_address> netmask <netmask> gateway <gateway> channel <channel_name> {lan | wan} [next]	指定した宛先ネットワークのスタティック経路情報を表示します。

以下に，スタティック経路の設定例を示します。

Sample 1）チャネル名“ch1”の WAN 側のスタティック経路として，宛先 IPv4 ネットワークアドレス 30.2.1.0/24，ゲートウェイアドレス 20.1.5.1 のスタティック経路を登録する場合

```
PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1
             channel “ch1” wan
```

Sample 2）チャネル名“ch1”の WAN 側のスタティック経路で，宛先 IPv4 ネットワークアドレス 30.2.1.0/24，ゲートウェイアドレス 20.1.5.1 のスタティック経路を削除する場合

```
PureFlow(A)> delete route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1
             cannel “ch1” wan
```

Sample 3）すべてのスタティック経路を削除する場合

```
PureFlow(A)> delete route all
```

STEP 4：シナリオの設定

本装置は，シナリオの登録により，各仮想回線のトラフィックアトリビュートを割り当てます。本設定は，トラフィックコントロールを動作させる場合に必要です。

レベル 2 以降のシナリオに設定できるパラメータを以下に示します。

表 8.9-7　シナリオのパラメータ

パラメータ	設定範囲	省略可能／不可	説明
シナリオ名 (scenario_name)	“/port1/xxxx”(2 階層) “/port2/xxxx”(2 階層) “/port3/xxxx”(2 階層) “/port4/xxxx”(2 階層) “/port1/xxxx/xxxx”(3 階層) “/port2/xxxx/xxxx”(3 階層) “/port3/xxxx/xxxx”(3 階層) “/port4/xxxx/xxxx”(3 階層) “/port1/xxxx/xxxx/xxxx”(4 階層) “/port2/xxxx/xxxx/xxxx”(4 階層) “/port3/xxxx/xxxx/xxxx”(4 階層) “/port4/xxxx/xxxx/xxxx”(4 階層)	不可	update コマンドでの省略，変更不可。 第1階層目には，Network ポートのポート番号を“/port1”のように指定し，第 2 階層以降に登録するシナリオ名を指定してください。 設定範囲は全階層(/port1，/port2，/port3，/port4)を含めて 1～128 文字です。
アクションモード	aggregate：集約キューモード フィルタに一致したすべてのトラフィックを 1 つのキューでトラフィックコントロールします。 individual：個別キューモード フィルタに一致したトラフィックを個別のキューでトラフィックコントロールします。 wan-accel：アクセラレーションモード フィルタに一致した TCP トラフィックをトラフィックアクセラレーションします。 discard　：廃棄モード フィルタに一致したトラフィックを廃棄します。 forward　：転送モード フィルタに一致したトラフィックを上位階層のシナリオに転送します。	不可	update コマンドでの省略，変更不可。
クラス (class)	1～8	可能	省略時：2 1(高)⇔(低)8 aggregate, individual モードで有効
最低帯域 (min_bandwidth)	0，1 k[bit/s]～1 G[bit/s] (設定単位：1 k[bit/s])	可能	省略時および 0：最低帯域保証なし(最小値の確保) aggregate，individual，wan-accel モードで有効
最大帯域 (peak_bandwidth)	1 k[bit/s]～1 G[bit/s] (設定単位：1 k[bit/s])	可能	省略時：最大帯域制限なし aggregate，individual，wan-accel モードで有効

パラメータ	設定範囲	省略可能／不可	説明
バッファサイズ (bufsize)	2 k[Byte]～100 M[Byte] (設定単位:1 k[Byte])	可能	省略時:1 M[Byte] (aggregate, individual モード時), 15 M[Byte] (wan-accel モード時) aggregate, individual, wan-accel モードで有効
シナリオインデックス (scenario_id)	1～4096	可能	update コマンドでの変更不可。 省略時:自動付与 すべてのアクションモードで有効
最大キュー数 (maxquenum)	1～4096(シナリオ拡張ライセンス有効時) 1～2048(シナリオ拡張ライセンス無効時)	可能	省略時: 4096(シナリオ拡張ライセンス有効時) 2048(シナリオ拡張ライセンス無効時) individual モードで有効
キュー分割対象 (quedivision)	default : "vid, inner-vid, sip, dip, proto, sport, dport"の組み合わせでキューを分割します。 vid : VLAN ID でキューを分割します。 cos : CoS でキューを分割します。 inner-vid : インナーVLAN ID でキューを分割します。 inner-cos : インナーCoS でキューを分割します。 ethertype : EthernetType/Length でキューを分割します。 sip : 送信元 IP アドレスでキューを分割します。 dip : 宛先 IP アドレスでキューを分割します。 proto : プロトコル番号でキューを分割します。 sport : 送信元ポート番号でキューを分割します。 dport : 宛先ポート番号でキューを分割します。	可能	省略時:default individual モードで有効

パラメータ	設定範囲	省略可能／不可	説明
キュー最大数超過アクション (failaction)	discard　：廃棄します。 forwardbesteffort ：ベストエフォート(クラス 8)転送します。 forwardattribute ：トラフィックアトリビュートを指定して転送します。	可能	省略時: forwardbesteffort 個別キューの最大数を超過した場合の動作, または，キュー分割対象で5tuple（sip，dip，proto，sport，dport）のいずれかが指定された場合の IP 以外のフロー（ARP など）に適用される動作を指定します。 individual モードで有効
キュー最大数超過時の最低帯域 (fail_min_bw)	0，1 k[bit/s]～1 G[bit/s]	可能	省略時:最低帯域保証なし individual モードで“forwardattribute”指定時のみ有効
キュー最大数超過時の最大帯域 (fail_peak_bw)	1 k[bit/s]～1 G[bit/s]	可能	省略時:最大帯域制限なし individual モードで“forwardattribute”指定時のみ有効
キュー最大数超過時のクラス (fail_class)	1～8	可能	省略時:8 1(高)⇔(低)8 個別キューモードで“forwardattribute”指定時のみ有効
対向装置の Primary IP アドレス (peer)	IPv4 または IPv6 アドレス	不可	update コマンドでの変更不可。 アクセラレーショントンネルを構成する対向装置の Primary IP アドレスを設定します。 wan-accel モードのみ有効
対向装置の Secondary IP アドレス (second-peer)	IPv4 または IPv6 アドレス	可能	update コマンドでの変更不可。 アクセラレーショントンネルを構成する対向装置の Secondary IP アドレスを指定します。 最大 100 個までのアクセラレーションモードシナリオに指定できます。 wan-accel モードのみ有効 （詳細は「8.14.6 トラフィックアクセラレーションの冗長構成」を参照してください。）

パラメータ	設定範囲	省略可能／不可	説明
TCP 接続ポート番号 (dport)	10001～20000	可能	update コマンドでの変更不可。 省略時:10000 アクセラレーショントンネルを構成する対向装置の TCP 接続ポート番号を指定します。 指定する場合は自装置と対向装置に設定するシナリオの dport は同じ値を指定してください。 wan-accel モードのみ有効
VLAN ID (VID)	1～4094	可能	省略時:none トラフィックアクセラレーションするチャネルの VLAN ID を設定します。 wan-accel モードのみ有効
Inner-VLAN ID (VID)	1～4094	可能	省略時:none トラフィックアクセラレーションするチャネルの Inner-VLAN ID を設定します。 wan-accel モードのみ有効
CoS (through, user_priority)	through 0～7	可能	省略時:through CoS の書き換え値を設定します。 aggregate , individual , wan-accel モードで有効 wan-accel モードのみ update コマンドでの変更不可。
Inner-CoS (through, user_priority)	through 0～7	可能	省略時:through Inner-CoS の書き換え値を設定します。 aggregate , individual , wan-accel モードで有効 wan-accel モードのみ update コマンドでの変更不可。
DSCP (through, user_priority)	through 0～63	可能	省略時:through DSCP の書き換え値を設定します。 aggregate , individual , wan-accel モードで有効 wan-accel モードのみ update コマンドでの変更不可。
圧縮機能の有効／無効 (compression)	enable disable	可能	省略時:有効 トラフィックアクセラレーションする TCP データについて, 圧縮の有効／無効を指定します。 wan-accel モードのみ有効

パラメータ	設定範囲	省略可能／不可	説明
TCP バッファサイズ (tcp-mem)	auto 64 k[Byte]～200 M[Byte] (設定単位:1 k[Byte])	可能	省略時:auto TCP のバッファサイズを指定します。 wan-accel モードのみ有効
輻輳制御モード (cc-mode)	normal semi-fast fast	可能	省略時:normal トラフィックアクセラレーションの輻輳制御モードを指定します。 wan-accel モードのみ有効
トラフィックアクセラレーションの自動バイパスの RTT しきい値 (bypass-thresh)	0～10000 ミリ秒	可能	省略時:0 トラフィックアクセラレーションの自動バイパス機能の RTT(Round Trip Time:往復遅延時間)しきい値をミリ秒単位で指定します。 wan-accel モードのみ有効 (詳細は「8.14.5 トラフィックアクセラレーションバイパス」を参照してください。)
トラフィックアクセラレーションの自動バイパスの Keep Alive 監視の有効／無効 (bypass-keepalive)	enable disable	可能	省略時:disable トラフィックアクセラレーションの自動バイパス機能の Keep Alive 監視の有効／無効を指定します。 最大 100 個までのアクセラレーションモードシナリオに指定できます。 wan-accel モードのみ有効 (詳細は「8.14.5 トラフィックアクセラレーションバイパス」を参照してください。)
TCP-FEC 機能の有効／無効 (fec)	enable disable	可能	省略時:無効 TCP-FEC 機能の有効／無効を指定します。 wan-accel モードのみ有効 (詳細は「8.14.7 TCP-FEC 機能」を参照してください。)
FEC ブロックサイズ (block-size)	2 k[Byte]～50 k[Byte] (設定単位:1 k[Byte])	可能	省略時:2k TCP-FEC 機能の FEC ブロックサイズを指定します。 wan-accel モードのみ有効 (詳細は「8.14.7 TCP-FEC 機能」を参照してください。)
データブロックサイズ (data-block-size)	2 k[Byte]～200 k[Byte] (設定単位:1 k[Byte])	可能	省略時:20k TCP-FEC 機能のデータブロックサイズを指定します。 wan-accel モードのみ有効 (詳細は「8.14.7 TCP-FEC 機能」を参照してください。)

パラメータ	設定範囲	省略可能／不可	説明
FEC セッション数 （fec-session）	0～400 セッション （装置全体で最大 400 セッション）	可能	省略時：40 TCP-FEC 機能を使用する TCP セッション（FEC セッション）数を指定します。本パラメータにより，各シナリオで使用する FEC セッション数を制限します。 wan-accel モードのみ有効 （詳細は「8.14.7 TCP-FEC 機能」を参照してください。）

以下に，レベル 2 以降のシナリオ設定に関する CLI コマンドを示します。

表 8.9-8　シナリオ設定に関する CLI コマンド

add scenario <scenario_name> action discard [scenario <scenario_id>]	廃棄モードのシナリオを登録します。 シナリオインデックスは自動付与されるので，通常は設定する必要はありません。
add scenario <scenario_name> action aggregate [cos {through | <user_priority>] [inner-cos {through | <user_priority>] [dscp {through | <user_priority>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>] [scenario <scenario_id>]	集約キューモードのシナリオを登録します。 帯域，バッファサイズなどのトラフィックアトリビュートを指定します。 シナリオインデックスは自動付与されるので，通常は設定する必要はありません。
add scenario <scenario_name> action individual [cos {through | <user_priority>] [inner-cos {through | <user_priority>] [dscp {through | <user_priority>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>] [scenario <scenario_id>] [maxquenum <quenum>] [quedivision <field>] [failaction {discard | forwardbesteffort | forwardattribute}] [fail_min_bw <min_bandwidth>] [fail_peak_bw <peak_bandwidth>] [fail_class <class>]	個別キューモードのシナリオを登録します。 帯域，バッファサイズなどのトラフィックアトリビュートを設定します。また，個別キューの最大数，キュー分割対象，個別キュー数超過時の動作を指定します。 シナリオインデックスは自動付与されるので，通常は設定する必要はありません。
add scenario <scenario_name> action wan-accel peer <IP_address> second-peer <IP_address> [dport <port>] [vid <vid>] [inner-vid <VID>] [cos {through | <user_priority>] [inner-cos {through | <user_priority>] [dscp {through | <user_priority>] [compression {enable | disable}] [tcp-mem {auto | <size>] [cc-mode {normal | semi-fast | fast}] [bypass-thresh <rtt>] [bypass-keepalive {enable | disable}] [fec {enable | disable}] [block-size <size>] [data-block-size <size>] [fec-session <session>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>]	アクセラレーションモードのシナリオを登録します。 対向装置の IP アドレス，TCP 接続ポート番号，VLAN などトラフィックアクセラレーションのトラフィックアトリビュートを設定します。 シナリオインデックスは自動付与されるので，通常は設定する必要はありません。 “Secondary”の対向装置の IP アドレスを指定すると，同時に機能が有効になります。

[bufsize <bufsize>] [scenario <scenario_id>]	
add scenario <scenario_name> action forward [scenario <scenario_id>]	転送モードのシナリオを登録します。 シナリオインデックスは自動付与されるので，通常は設定する必要はありません。
update scenario <scenario_name> action aggregate [cos {through | <user_priority>] [inner-cos {through | <user_priority>] [dscp {through | <user_priority>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>]	集約キューモードのシナリオを変更します。 本コマンドにより，トラフィックコントロールされている状態でトラフィックアトリビュートを変更できます。各パラメータは省略可能ですが，すべてを省略することはできません。 変更したいパラメータを 1 つ以上指定してください。 シナリオ名，アクションモード，シナリオインデックスは変更できません。
update scenario <scenario_name> action individual [cos {through | <user_priority>] [inner-cos {through | <user_priority>] [dscp {through | <user_priority>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>] [maxquenum <quenum>] [quedivision <field>] [failaction {discard | forwardbesteffort | forwardattribute}] [fail_min_bw <min_bandwidth>] [fail_peak_bw <peak_bandwidth>] [fail_class <class>]	個別キューモードのシナリオを変更します。 本コマンドにより，トラフィックコントロールされている状態でトラフィックアトリビュートを変更できます。各パラメータは省略可能ですが，すべてを省略することはできません。 変更したいパラメータを 1 つ以上指定してください。 シナリオ名，アクションモード，シナリオインデックスは変更できません。
update scenario <scenario_name> action wan-accel [vid <vid>] [inner-vid <VID>] [compression {enable | disable}] [tcp-mem {auto | <size>] [cc-mode {normal | semi-fast | fast}] [bypass-thresh <rtt>] [bypass-keepalive {enable |disable}] [fec {enable |disable}] [block-size <size>] [data-block-size <size>] [fec-session <session>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [bufsize <bufsize>]	アクセラレーションモードのシナリオを変更します。 本コマンドにより，トラフィックコントロールされている状態でトラフィックアトリビュートを変更できます。各パラメータは省略可能ですが，すべてを省略することはできません。 変更したいパラメータを 1 つ以上指定してください。 シナリオ名，アクションモード，対向装置の IP アドレス，TCP 接続ポート番号，VLAN，シナリオインデックスは変更できません。
delete scenario all	すべてのシナリオを削除します。
delete scenario <scenario_name> [recursive]	指定したシナリオ名のシナリオを削除します。 recursive を指定した場合には，指定シナ

	リオ以下のシナリオを削除します。 recursive を指定しない場合には，下位層のシナリオを持つシナリオは，削除できません。
show scenario all	すべてのシナリオ情報を表示します。
show scenario name <scenario_name> [summary] [next]	指定したシナリオ名のシナリオ情報を表示します。 summary を指定した場合には，フィルタ情報は表示しません。 next を指定した場合には，次のシナリオ情報を表示します。
set scenario tree mode {inbound | outbound}	シナリオのツリーモード（入力側／出力側）を設定します。シナリオおよびフィルタ分類を，Network ポートへの入力トラフィックに対して適用するか，Network ポートからの出力トラフィックに対して適用するかを指定します。
show scenario tree	シナリオの階層関連を示すツリーを表示します。

以下に，レベル 2 シナリオの設定例を示します。

Sample 1）Network ポート 1/1 から受信した“Tokyo”拠点への集約キューモードシナリオについて，最大帯域を 300 Mbit/s のシナリオを登録する場合

```
PureFlow (A)> add scenario "/port1/Tokyo" action aggregate peak_bw 300M
```

Sample 2）Network ポート 1/1 から受信した“Osaka”拠点への個別キューモードシナリオについて，最大帯域 500 kbit/s，最大キュー数 20 個のシナリオを登録する場合

```
PureFlow (A)> add scenario "/port1/Osaka" action individual peak_bw 500k maxquenum 20
```

Sample 3）Network ポート 1/1 から受信した“Nagoya”拠点へのアクセラレーションモードシナリオについて，対向装置の IP アドレス 20.1.2.9 のシナリオを登録する場合

```
PureFlow (A)> add scenario "/port1/Nagoya" action wan-accel peer 20.1.2.9
```

Sample 4）ポートグループ設定した“group1”から受信した“Nagoya”拠点へのアクセラレーションモードシナリオについて，対向装置の IP アドレス 20.1.2.9 のシナリオを登録する場合

```
PureFlow (A)> add scenario "/group1/Nagoya" action wan-accel peer 20.1.2.9
```

レベル 3 以降のシナリオについても同様に，シナリオ名で上位シナリオと階層を指定します。

Sample 5）“Tokyo”拠点配下の“Shinjuku”エリアを集約キューモードシナリオで登録し，最大帯域を 100 Mbit/s のシナリオを登録する場合

```
PureFlow (A)> add scenario "/port1/Tokyo/Shinjuku" action aggregate peak_bw 100M
```

シナリオを削除する例を以下に示します。

Sample 6）“Tokyo”拠点配下のシナリオを削除する場合

```
PureFlow (A)> delete scenario "/port1/Tokyo" recursive
```

STEP 5：フィルタの設定

本装置は，Bridge-ctrl フレーム，Ethernet フレーム，IPv4 パケット，IPv6 パケットのトラフィックをフィルタにより識別します。本設定は，トラフィックコントロールを動作させる場合に必要です。
レベル 2 以降のフィルタに設定できるパラメータを，以下に示します。

表 8.9-9　フィルタのパラメータ

パラメータ		設定範囲	省略可能／不可
フィルタ名 （filter_name）		1～48 文字	不可
シナリオ名 （scenario_name）		全階層を含めて 1～128 文字 （“add scenario”コマンドで登録したもの）	不可
フィルタ種類		bridge-ctrl，ethernet，ipv4，ipv6	不可
イーサタイプ （ethertype）		Ethernet ヘッダ内 Type フィールド値指定 0x0000～0xFFFF	可能 Ethernet フィルタのみ有効
VLAN ID （VID）		IEEE802.1Q VLAN ID を指定 0～4094（範囲指定可能）， none（VLAN Tag なし）	可能
CoS		IEEE802.1Q VLAN 内 CoS を指定 0～7	可能
Inner-VLAN ID （VID）		QinQ におけるインナーVLAN ID を指定 0～4094（範囲指定可能）， none（VLAN Tag なし）	可能
Inner-CoS		QinQ におけるインナーVLAN 内 CoS を指定 0～7	可能
送信元 IP アドレス （sip）	IPv4	0.0.0.0～255.255.255.255 （範囲指定「start-end」または「address/bitmask」で指定可能） ルールリスト名	可能 IP フィルタのみ有効
	IPv6	0::0～FFFF:…:FFFF（小文字入力，範囲指定「start-end」または「address/bitmask」で指定可能） ルールリスト名	可能 IP フィルタのみ有効
宛先 IP アドレス （dip）	IPv4	0.0.0.0～255.255.255.255 （範囲指定「start-end」または「address/bitmask」で指定可能） ルールリスト名	可能 IP フィルタのみ有効
	IPv6	0::0～FFFF:…:FFFF （小文字入力，範囲指定「start-end」または「address/bitmask」で可能） ルールリスト名	可能 IP フィルタのみ有効
ToS，または Traffic Class	IPv4	0～255 （範囲指定「start-end」可能）	可能 IP フィルタのみ有効
	IPv6	0～255 （範囲指定「start-end」可能）	可能 IP フィルタのみ有効

パラメータ	設定範囲	省略可能／不可
プロトコル番号 (proto)	0～255 (範囲指定「start-end」可能) (tcp, udp, icmp は文字入力可能)	可能 IP フィルタのみ有効
送信元ポート番号 (sport)	0～65535 (範囲指定「start-end」可能) ルールリスト名	可能 IP フィルタのみ有効
宛先ポート番号 (dport)	0～65535 (範囲指定「start-end」可能) ルールリスト名	可能 IP フィルタのみ有効
フィルタ優先度 (priority)	1～40000	可能 省略時は 20000

以下に，レベル 2 以降のフィルタに関する CLI コマンドを示します。

表 8.9-10　フィルタに関する CLI コマンド

add filter scenario <scenario_name> filter <filter_name> bridge-ctrl [priority <filter_pri>]	宛先 MAC アドレスが 01-80-C2-00-00-00～01-80-C2-00-00-FF（スパニングツリープロトコル，リンクアグリゲーション，EAPoL（認証プロトコル）などを含む）であるフレームを識別します。
add filter scenario <scenario_name> filter <filter_name> ethernet [vid {<VID> | none}] [cos <user_priority>] [inner-vid {<VID> | none}] [inner-cos <user_priority>] [ethertype <type>] [priority <filter_pri>]	Ethernet ヘッダの length/type フィールドを対象としてフレームを識別します。また，VLAN Tag 内の VLAN ID，CoS を指定できます。 各パラメータは省略可能ですが，すべてを省略することはできません。“priority”以外のパラメータを 1 つ以上指定してください。
add filter scenario <scenario_name> filter <filter_name> ipv4 [vid {<VID> | none}] [cos <user_priority>] [inner-vid {<VID> | none}] [inner-cos <user_priority>] [sip [list] {<src_IP_address> | <list_name>}] [dip [list] {<dst_IP_address> | <list_name>}] [cos <user_priority>] [proto <protocol>] [sport [list] {<sport> | <list_name>}] [dport [list] {<dport> | <list_name>}] [priority <filter_pri>]	IPv4 パケットの IP アドレス，プロトコル番号，ポート番号などを対象としてパケットを識別します。また，VLAN ID，CoS を指定できます。 各パラメータは省略可能です。すべてを省略した場合は，IPv4 パケットすべてとなります。
add filter scenario <scenario_name> filter <filter_name> ipv6 [vid {<VID> | none}] [cos <user_priority>] [inner-vid {<VID> | none}]	IPv6 パケットの IP アドレス，プロトコル番号，ポート番号などを対象としてパケットを識別します。また，VLAN ID，CoS を指定できます。 各パラメータは省略可能です。すべてを省

[inner-cos <user_priority>] [sip [list] {<src_IP_address> | <list_name>}] [dip [list] {<dst_IP_address> | <list_name>}] [cos <user_priority>] [proto <protocol>] [sport [list] {<sport> | <list_name>}] [dport [list] {<dport> | <list_name>}] [priority <filter_pri>]	略した場合は，IPv6パケットすべてとなります。
delete filter scenario <scenario_name> filter <filter_name>	指定シナリオの指定フィルタを削除します。
delete filter scenario <scenario_name>	指定シナリオ内のすべてのフィルタを削除します。
delete filter all	すべてのフィルタを削除します。
show filter scenario <scenario_name> [filter <filter_name>] [summary] [next]	指定したシナリオのフィルタ情報を表示します。 summary を指定した場合には，フィルタ名のみ表示します。 next を指定した場合には，次のシナリオ情報を表示します。
show filter all	すべてのシナリオの全フィルタ設定内容を表示します。

以下に，レベル 2 フィルタの設定例を示します。

Sample 1）レベル 2 シナリオ“/port1/bpdu”に流すフィルタ条件として，BPDU のフィルタを登録する場合

```
PureFlow (A)> add filter scenario “/port1/bpdu” filter “bpdu” bridge-ctrl priority 1
```

Sample 2）レベル 2 シナリオ“/port1/arp”に流すフィルタ条件として，ARP のフィルタを登録する場合

```
PureFlow (A)> add filter scenario “/port1/arp” filter “arp” ethernet ethertype 0x0806
```

Sample 3）レベル 2 シナリオ“/port1/Tokyo”に流すフィルタ条件として，IPv4 における VLAN ID を“10”のフィルタを登録する場合。

```
PureFlow (A)> add filter scenario “/port1/Tokyo” filter “Tokyo” ipv4 vid 10
```

Sample 4）レベル 2 シナリオ“/port1/Osaka”に流すフィルタ条件として，IPv6 における VLAN ID を“20”のフィルタを登録する場合。

```
PureFlow (A)> add filter scenario “/port1/Osaka” filter “Osaka” ipv6 vid 20
```

レベル 3 以降のフィルタについても同様に，対象シナリオを指定してフィルタを設定します。

Sample 5）レベル 3 シナリオ“/port1/Tokyo/Shinjuku”に流すフィルタ条件として，IPv4 における送信元 IP アドレス“192.168.10.0 ～ 192.168.10.255”のフィルタを登録する場合。

```
PureFlow (A)> add filter scenario “/port1/Tokyo/Shinjuku” filter “Shinjuku” ipv4
                sip 192.168.10.0-192.168.10.255
```

8.10 ルールリストの設定方法

本章ではルールリストの設定方法について説明します。

ルールリストを利用するには，以下の手順で設定します。
手順 1）ルールリストを登録する。
手順 2）ルールリストに対してルールリストエントリを登録する。
手順 3）フィルタ登録コマンドにルールリストを指定する。

ルールリストおよびルールリストエントリのパラメータを，以下に示します。

表 8.10-1　ルールリストのパラメータ

パラメータ	設定範囲
ルールリスト名	1 文字-32 文字
ルールリストタイプ	ipv4，ipv6，l4port，domain

表 8.10-2　ルールリストエントリのパラメータ

パラメータ		設定範囲
ルールリスト名		登録済みのルールリスト名を指定する
ルールリストタイプ		ipv4，ipv6，l4port，domain
トラフィック分類条件	IPv4 アドレス	0.0.0.0～255.255.255.255 （範囲指定「start-end」または「address/bitmask」で指定可能）
	IPv6 アドレス	0::0～FFFF:…:FFFF （小文字入力，範囲指定「start-end」または「address/bitmask」で可能）
	TCP/UDP ポート番号	0～65535（範囲指定「start-end」可能）
	ドメイン名	・使用可能文字： 1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ -.*　（「*」は，後方一致のワイルドカード） ・ドメインの長さ： 「.」「*」を含め，253 文字以内 ・ラベルの長さ： 63 文字以内 ・大文字小文字の区別なし

以下に，ルールリスト設定に関する CLI コマンドを示します。

表 8.10-3　ルールリスト設定に関する CLI コマンド

add rulelist group <list_name> {ipv4 | ipv6 | l4port | domain}	ルールリストを登録します。 ipv4，ipv6，l4port，domain のいずれかを対象にします。
add rulelist entry <list_name> ipv4 <IP_address>	IPv4 アドレスのルールリストエントリを登録します。
add rulelist entry <list_name> ipv6 <IP_address>	IPv6 アドレスのルールリストエントリを登録します。
add rulelist entry <list_name> l4port <port>	TCP/UDP ポート番号のルールリストエントリを登録します。
add rulelist entry <list_name> domain <domain_ name>	ドメイン名のルールリストエントリを登録します。
delete rulelist group <list_name>	指定したルールリストグループを削除します。
delete rulelist group all	ルールリストグループをすべて削除します。
delete rulelist entry <list_name> ipv4 <IP_address>	IPv4 アドレスのルールリストエントリを削除します。
delete rulelist entry <list_name> ipv6 <IP_address>	IPv6 アドレスのルールリストエントリを削除します。
delete rulelist entry <list_name> l4port <port>	TCP/UDP ポート番号のルールリストエントリを削除します。
delete rulelist entry <list_name> domain <domain_ name>	ドメイン名のルールリストエントリを削除します。
delete rulelist entry <list_name> all	指定したルールリストグループのルールリストエントリをすべて削除します。
show rulelist all	すべてのルールリスト情報を表示します。
show rulelist [<list_name>]	指定したルールリスト名のルールリスト情報を表示します。

ルールリストは，以下のルールに従って設定してください。

(1) 装置内で重複しないユニークなルールリスト名を設定してください。

(2) “delete rulelist group”コマンドは，フィルタに登録されていないルールリストに対してのみ行うことができます。

(3) ルールリスト名には，“all”は指定できません。

以下に，ルールリストの設定例を示します。

手順 1)　ルールリスト“TVCservers”を登録する。

```
PureFlow(A) >  add rulelist group “TVCservers” ipv4
```

手順 2)　ルールリスト“TVCservers”に，ルールリストエントリを登録する。

```
PureFlow(A) >  add rulelist entry “TVCservers” ipv4 172.16.111.11
PureFlow(A) >  add rulelist entry “TVCservers” ipv4 172.16.112.11
                                                                    ・（リスト化するホスト
IP の追加）
                                                                    ・
```

手順 3)　フィルタ登録コマンドの sip にルールリスト名“TVCservers”を登録する。

```
PureFlow(A)> add filter scenario “/port1/Tokyo/TVC” filter “TVC” ipv4 sip list
                    “TVCservers”
```

8.11 チャネルインタフェース通信

チャネルインタフェース通信は IPv4 および IPv6 の同時利用が可能です。

表 8.11-1　チャネルインタフェース通信

機能	IPv4	IPv6
PING	○	○
TRACEROUTE	○	○

チャネルインタフェースでは以下の CLI コマンドを使用して，ネットワークの疎通確認／経路確認をすることができます。

表 8.11-2　チャネルインタフェースに関する CLI コマンド

ping <IP_address> channel <channel_name> {lan ｜ wan} [<send_count>]	ICMP ECHO_REQUEST パケットを指定 IP アドレスに送信します。 （IPv4／IPv6）
traceroute <IP_address> channel <channel_name> {lan ｜wan}	指定 IP アドレスに到達するまでの経路を表示します。
arp -a channel <channel_name> <IP_Address>	ARP エントリを表示します。（IPv4 のみ）
arp -d <IP_address> channel <channel_name>	ARP エントリを削除します。（IPv4 のみ）
delete ndp neighbor <IP_address> [channel <channel_name>]	NDP エントリを削除します。（IPv6 のみ）
show ndp neighbor [channel {<channel_name>｜all}] [<IP_address>]	NDP エントリを表示します。（IPv6 のみ）

チャネルで設定した WAN 側ポートから IPv4 アドレス 192.168.10.100 との疎通確認を行う場合，以下に示すコマンドを実行します。

```
PureFlow(A)> ping 192.168.10.100 channel “channel1” wan
PING 192.168.10.100 0(28) bytes of data.
8 byte from 192.168.10.100: icmp_req=1 time=200.208 ms
8 byte from 192.168.10.100: icmp_req=2 time=200.206 ms
8 byte from 192.168.10.100: icmp_req=3 time=200.184 ms
--- 192.168.10.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss
rtt min/avg/max = 200.184/200.199/200.208 ms
PureFlow(A)>
```

疎通確認失敗時は，以下のように表示します。チャネルインタフェースの設定，およびネットワーク接続を確認してください。

```
PureFlow(A)> ping 192.168.10.101 channel channel1 wan
PING 192.168.10.101 0(28) bytes of data.
from 192.168.10.101: icmp_req=1 Destination Host Unreachable
from 192.168.10.101: icmp_req=2 Destination Host Unreachable
from 192.168.10.101: icmp_req=3 Destination Host Unreachable
--- 192.168.10.101 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss
PureFlow(A)>
```

IPv4 アドレス 192.168.10.101 の ARP エントリを削除する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> arp –d 192.168.10.100 channel "channel1"
PureFlow(A)> arp –a channel "channel1" 192.168.10.100
IP address          MAC address                 type
-----------------------------------------------------------------------------------------
PureFlow(A)>
```

チャネルで設定した WAN 側ポートから IPv6 アドレス 2001:DB8::1 との疎通確認を行う場合，以下に示すコマンドを実行します。

```
PureFlow(A)> ping 2001:db8::1 channel "channel1" wan
PING 2001:db8::1 (2001:db8::1) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.372 ms

--- 2001:db8::1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.372/0.372/0.372/0.000 ms
PureFlow(A)> show ndp neighbor channel "channel1" 2001:db8::1
IP address                  MAC address                 type
-----------------------------------------------------------------------------------------
2001:db8::1                 00-00-91-01-23-45           reachable
PureFlow(A)>
```

疎通確認失敗時は，以下のように表示します。チャネルフェースの設定，およびネットワーク接続を確認してください。

```
PureFlow(A)> ping 2001:db8::10 channel "channel1" wan
PING 2001:db8::10 (2001:db8::10) 56(84) bytes of data.

--- 2001:db8::10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 100ms
PureFlow(A)>
```

IPv6 アドレス 2001:db8::10 の NDP エントリを削除する場合，以下に示すコマンドを実行します。

```
PureFlow(A)> delete ndp neighbor 2001:db8::10 channel "channel1"
PureFlow(A)> show ndp neighbor channel "channel1" 2001:db8::10
IP address                      MAC address                     type
-----------------------------------------------------------------------------------------------
PureFlow(A)>
```

8.12 アプリケーション高速化機能

本装置は，アプリケーション高速化機能を備えています。本機能は，トラフィックアクセラレーションのアプリケーションプロトコルにおけるコマンドのやり取りを効率化することで，アプリケーションのデータ転送を高速化します。たとえば，SMBプロトコル高速化機能を有効にした場合，ファイル共有プロトコルで接続された遠隔のファイルサーバからファイルをダウンロードする場合，ファイル転送時間を削減することができます。

図 8.12-1　アプリケーション高速化機能

注:

トラフィックアクセラレーション機能を使用する場合は，「TCP 高速化機能ライセンス(NF7500-L201A)」を購入していただく必要があります。

8.12.1 SMB プロトコル高速化機能

SMB(Server Message Block)プロトコルは，Windows®サーバネットワークにおいて，共有フォルダ設定やネットワークドライブを設定することでファイルサーバのファイルを共有するときに使用するプロトコルです。

SMB プロトコル高速化機能は，SMB プロトコルでの通信を効率化し，ファイルリードやファイルライトの操作を高速化します。
ファイルリードの操作において，ファイル属性をリードする SMB コマンド(SMB2_QUERY_INFO コマンド)とファイルデータをリードするコマンド(SMB2_READ コマンド)のコマンド通信を最適化し，全体のリード時間を短縮します。また，ファイルライトの操作において，ファイルデータを書き込む前のファイル属性をリードするコマンド(SMB2_QUERY_INFO コマンド)を最適化し，全体のライト時間を短縮します。

本機能は，SMB クライアントと SMB サーバが使用する SMB プロトコルのバージョンが SMB2.0 以上 SMB3.0 未満の場合に有効です。また，SMB クライアントが SMB3.0 以上をサポートしている場合，高速化対象外となります。SMB サーバと SMB クライアントのバージョンの組み合わせは，以下の表をご覧ください。

表 8.12.1-1　SMB サーバと SMB クライアントのバージョンの組み合わせ

SMB サーバ / SMB クライアント	SMB 2.0.2	SMB 2.1	SMB 3.0 SMB 3.0.2 SMB 3.1.1
SMB 2.0.2	TCP 高速化 SMB 高速化	TCP 高速化 SMB 高速化	TCP 高速化 SMB 高速化
SMB 2.1	TCP 高速化 SMB 高速化	TCP 高速化 SMB 高速化	TCP 高速化 SMB 高速化
SMB 3.0 SMB 3.0.2 SMB 3.1.1	TCP 高速化	TCP 高速化	TCP 高速化

本機能は，実行例①のように，SMB プロトコル高速化を指定するだけで使用できます。実行例②は，SMB 高速化対象を TCP ポート番号の 445 だけに限定する場合の設定です。

実行例①:TCP ポート番号 139 と 445 を SMB プロトコルとして高速化する場合

```
PureFlow(A)> add apl-accel scenario /port1/woc1 protocol smb
```

実行例②:SMB プロトコルの TCP ポート番号を 445 だけに制限する場合

```
PureFlow(A)> update apl-accel scenario /port1/woc1 protocol smb tcp 445
```

実行例③:SMB 高速化設定を削除する場合

```
PureFlow(A)> delete apl-accel scenario /port1/woc1 protocol smb
```

デフォルト値以外のパラメータを使用する場合は，以下のパラメータを指定します。

表 8.12.1-2　SMB プロトコル高速化のパラメータ

コマンド	パラメータ	説明
add apl-accel scenario update apl-accel scenario	[tcp <port>]	SMB プロトコルの TCP ポート番号を指定します。 SMB プロトコルは標準で TCP 139 および 445 を使用します。SMB プロトコルの TCP ポート番号を変更している場合に，本パラメータで変更後の TCP ポート番号を設定してください。 TCP ポート番号はカンマ“，”で区切って最大 16 個まで指定できます。
	[smb-session <session>]	Windows®ファイル共有高速化を使用する TCP セッション（SMB セッション）数を指定します。 本パラメータにより，各シナリオで使用する SMB セッション数を制限します。 SMB セッション数は，装置全体で最大 1,000 セッションです。 なお，本パラメータで SMB セッション数が保証されるわけではありません。
	[read-attr {enable | disable}]	通常はデフォルト値でご利用ください。SMB プロトコルの read 操作における SMB2 QUERY_INFO コマンドの代理応答を有効にする場合は“enable”を，無効にする場合は“disable”を指定します。
	read-operation {enable | disable}	通常はデフォルト値でご利用ください。SMB プロトコルの read 操作における SMB2 READ コマンドの代理応答を有効にする場合は“enable”を，無効にする場合は“disable”を指定します。
	[read-cache-size <size>]	通常はデフォルト値でご利用ください。SMB プロトコルの read 操作における SMB2 READ コマンドの代理応答のキャッシュサイズを指定します。 設定範囲は 64 k[Byte]～60 M[Byte]です。有効な設定単位は 1 k[Byte]です。 設定単位（k，M）を指定してください。
	[write-attr {enable | disable}]	通常はデフォルト値でご利用ください。SMB プロトコルの write 操作における SMB2 QUERY_INFO コマンドの代理応答を有効にする場合は“enable”を，無効にする場合は“disable”を指定します。

コマンド	パラメータ	説明
	[write-attr-1st {enable | disable}]	通常はデフォルト値でご利用ください。SMB プロトコルの write 操作前の SMB2 SET_INFO コマンドの代理応答を有効にする場合は“enable”を，無効にする場合は“disable”を指定します。
	[write-attr-2nd {enable | disable}]	SMB プロトコルの write 操作後の SMB2 SET_INFO コマンドの代理応答を有効にする場合は“enable”を，無効にする場合は“disable”を指定します。
	[write-operation {enable | disable}]	通常はデフォルト値でご利用ください。SMB プロトコルの write 操作における SMB2 WRITE コマンドの代理応答を有効にする場合は“enable”を，無効にする場合は“disable”を指定します。
show scenario	name <scenario_name>	指定したシナリオ名のシナリオ情報（アプリケーション高速化機能に関するパラメータ）を表示します。

8.12.2 SMB プロトコル高速化機能の注意点

1）SMB パケットのデジタル署名が常に有効だった場合，本機能は SMB プロトコルの高速化を実行せず，TCP 通信の高速化のみを実行します。たとえば，SMB サーバ（ファイル共有サーバ）が Active Directory®のドメインコントローラであるとき，SMB サーバと SMB クライアント（ユーザ PC）との通信は，常にデジタル署名が行われます。この場合，本機能は SMB プロトコルの高速化を実行せず，TCP 通信の高速化のみを実行します。ファイル共有サーバが Active Directory®のメンバサーバであるとき，本機能は SMB プロトコルを高速化します。

2）大容量ファイルの SMB 転送において，クライアント PC が WindowsSerer® 2008 などのサーバ OS を使用している場合，SMB 無効設定でトラフィックアクセルしたほうがファイル転送時間が短くなる場合があります。

3）SMB プロトコル高速化機能のリソースが不足したとき，Appli-Accel Sessions および Appli-Accel Buffer に関するシステムログを表示します。これらシステムログが表示されたとき，smb-session パラメータおよび read-cache-size パラメータを指定し，セッション数およびバッファ量を制限してください。

8.13 コンフィギュレーション例

以下のネットワーク環境の設定を行う場合のコンフィギュレーション例を示します。

[Case 1]エリアおよびアプリの帯域確保を行う

- 東京のネットワークはアウターVLAN ID 10 です(レベル 2 フィルタの設定)。
- 大阪のネットワークはアウターVLAN ID 20 です(レベル 2 フィルタの設定)。
- 東京内新宿のネットワークはインナーVLAN ID 100 です(レベル 3 フィルタの設定)。
- 大阪内梅田のネットワークはインナーVLAN ID 200 です(レベル 3 フィルタの設定)。
- 新宿内制御対象アプリの送信元 IP アドレスは“192.168.10.1”です(レベル 4 フィルタの設定)。
- 梅田内制御対象アプリの送信元ポート番号は“2000”です(レベル 4 フィルタの設定)。
- 装置から送出する最大帯域を 500 Mbit/s とします(レベル 1 回線の設定)。
- センタから東京エリアへの最大帯域を 300 Mbit/s, 大阪エリアへの最大帯域を 100 Mbit/s とします(レベル 2 シナリオの設定)。
- 東京エリア確保帯域 300 Mbit/s のうち, 新宿エリアに対し最低帯域を 100 Mbit/s, 最大帯域を 300 Mbit/s とします(レベル 3 シナリオの設定)。
- 大阪エリア確保帯域 100 Mbit/s のうち, 梅田エリアに対し最低帯域を 50 Mbit/s, 最大帯域を 100 Mbit/s とします(レベル 3 シナリオの設定)。
- 新宿エリア内対象アプリに対し, 最低帯域を 1 Mbit/s, 最大帯域を 5 Mbit/s とします(レベル 4 シナリオの設定)。
- 梅田エリア内対象アプリに対し, 最大帯域を 5 Mbit/s とします(レベル 4 シナリオの設定)。

図 8.13-1　Case 1 の構成

図 8.13-2　Case 1 のコンフィギュレーション例

以下のコマンドを実行します。

＜レベル 1 シナリオ設定＞

PureFlow (A)> update scenario “/port1” action aggregate peak_bw 500M

＜レベル 2 シナリオ設定＞

PureFlow (A)> add scenario “/port1/Tokyo” action aggregate peak_bw 300M
PureFlow (A)> add scenario “/port1/Osaka” action aggregate peak_bw 100M

＜レベル 2 フィルタ設定＞

PureFlow (A)> add filter scenario “/port1/Tokyo” filter “Tokyo” ipv4 vid 10
PureFlow (A)> add filter scenario “/port1/Osaka” filter “Osaka” ipv4 vid 20

＜レベル 3 シナリオ設定＞

PureFlow (A)> add scenario “/port1/Tokyo/shinjuku” action aggregate min_bw 100M peak_bw 300M
PureFlow (A)> add scenario “/port1/Osaka/Umeda” action aggregate min_bw 50M peak_bw 100M

＜レベル 3 フィルタ設定＞

PureFlow(A)> add filter scenario “/port1/Tokyo/Shinjuku” filter “Shinjuku” ipv4 inner-vid 100

PureFlow(A)> add filter scenario “/port1/Osaka/Umeda” filter “Umeda” ipv4 inner-vid 200

＜レベル 4 シナリオ設定＞

PureFlow(A)> add scenario “/port1/Tokyo/Shinjuku/appli1” action aggregate min_bw 1M peak_bw 5M

PureFlow(A)> add scenario “/port1/Osaka/Umeda/appli1” action aggregate peak_bw 5M

＜レベル 4 フィルタ設定＞

PureFlow(A)> add filter scenario “/port1/Tokyo/Shinjuku/appli1” filter “Shin_appli1” ipv4 sip 192.168.10.1

PureFlow(A)> add filter scenario “/port1/Osaka/Umeda/appli1” filter “Ume_appli1” ipv4 sport 2000

［Case 2］ルールリストを使って，フィルタ設定を簡素化する。

- 各拠点は，本社に設置されたサーバを利用している（TV 会議，ファイルサーバ，VoIP）。
- PureFlowWS1 は，各拠点に，通信帯域を割り当て，さらにサービスごとに通信帯域を割り当てる。

図 8.13-3　Case 2 の構成とコンフィギュレーション例

＜サービスごとにルールリストに登録する＞

・ TV 会議サーバの IP アドレスをルールリストに登録する。

PureFlow(A)> add rulelist group “TVCservers” ipv4
PureFlow(A)> add rulelist entry “TVCservers” ipv4 172.16.170.11
PureFlow(A)> add rulelist entry “TVCservers” ipv4 172.16.170.12

・ ファイルサーバの IP アドレスをルールリストに登録する。

PureFlow(A)> add rulelist group “FILEservers” ipv4
PureFlow(A)> add rulelist entry “FILEservers” ipv4 172.16.170.21
PureFlow(A)> add rulelist entry “FILEservers” ipv4 172.16.170.22
PureFlow(A)> add rulelist entry “FILEservers” ipv4 172.16.170.23

・ IP 電話サーバの IP アドレスをルールリストに登録する。

PureFlow(A)> add rulelist group “VoIPservers” ipv4
PureFlow(A)> add rulelist entry “VoIPservers” ipv4 172.16.170.31
PureFlow(A)> add rulelist entry “VoIPservers” ipv4 172.16.170.32

＜東京拠点への仮想回線を登録する＞

・ 東京拠点へのトラフィック総量を設定する。

PureFlow(A)> add scenario “/port1/Tokyo” action aggregate peak_bw 10M
PureFlow(A)> add filter scenario “/port1/Tokyo” filter “Tokyo” ipv4 dip 192.168.10.0-192.168.10.255

・ TV 会議サーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Tokyo/TVC” action aggregate min_bw 5M
PureFlow(A)> add filter scenario “/port1/Tokyo/TVC” filter “Tokyo_TVC” ipv4 sip list “TVCservers”

・ ファイルサーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Tokyo/FILE” action aggregate min_bw 4M
PureFlow(A)> add filter scenario “/port1/Tokyo/FILE” filter “Tokyo_FILE” ipv4 sip list “FILEservers”

・ IP 電話サーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Tokyo/VoIP” action aggregate min_bw 1M
PureFlow(A)> add filter scenario “/port1/Tokyo/VoIP” filter “Tokyo_VoIP” ipv4 sip list “VoIPservers”

同じルールリストを使って，名古屋拠点と大阪拠点のトラフィックを登録します。

＜名古屋拠点への仮想回線を登録する＞

・ 名古屋拠点へのトラフィック総量を設定する。

PureFlow(A)> add scenario “/port1/Nagoya” action aggregate peak_bw 10M
PureFlow(A)> add filter scenario “/port1/Nagoya” filter “Nagoya” ipv4 dip 192.168.20.0-192.168.20.255

・ TV 会議サーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Nagoya/TVC” action aggregate min_bw 5M
PureFlow(A)> add filter scenario “/port1/Nagoya/TVC” filter “Nagoya_TVC” ipv4 sip list “TVCservers”

・ ファイルサーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Nagoya/FILE” action aggregate min_bw 4M
PureFlow(A)> add filter scenario “/port1/Nagoya/FILE” filter “Nagoya_FILE” ipv4 sip list “FILEservers”

・ IP 電話サーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Nagoya/VoIP” action aggregate min_bw 1M
PureFlow(A)> add filter scenario “/port1/Nagoya/VoIP” filter “Nagoya_VoIP” ipv4 sip list “VoIPservers”

＜大阪拠点への仮想回線を登録する＞

・ 大阪拠点へのトラフィック総量を設定する。

PureFlow(A)> add scenario “/port1/Osaka” action aggregate peak_bw 10M
PureFlow(A)> add filter scenario “/port1/Osaka” filter “Osaka” ipv4 dip 192.168.30.0-192.168.30.255

・ TV 会議サーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Osaka/TVC” action aggregate min_bw 5M
PureFlow(A)> add filter scenario “/port1/Osaka/TVC” filter “Osaka_TVC” ipv4 sip list “TVCservers”

・ ファイルサーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Osaka/FILE” action aggregate min_bw 4M
PureFlow(A)> add filter scenario “/port1/Osaka/FILE” filter “Osaka_FILE” ipv4 sip list “FILEservers”

・ IP 電話サーバのルールリストを使ってトラフィックを登録する。

PureFlow(A)> add scenario “/port1/Osaka/VoIP” action aggregate min_bw 1M
PureFlow(A)> add filter scenario “/port1/Osaka/VoIP” filter “Osaka_VoIP” ipv4 sip list “VoIPservers”

[Case 3]VLAN で束ねられていない通常ネットワーク間のアクセラレーションを行う

図 8.13-4　Case 3 の構成とコンフィギュレーション例

PureFlow1 の設定

以下のコマンドを実行します。

＜チャネル設定＞

PureFlow(A)> add channel “ch1” lan 1/1 wan 1/2 vid none

＜デフォルトチャネル設定＞

PureFlow(A)> add channel “ch10000” lan 1/1 wan 1/2 default

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow(A)> set ip channel “ch1” 20.1.5.9 netmask 255.255.255.0

＜WAN 側のルート設定＞

PureFlow(A)> add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

PureFlow(A)> add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

＜LAN 側のルート設定＞

PureFlow(A)> add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.5.2　channel “ch1” lan

PureFlow(A)> add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.5.2　channel “ch1” lan

＜シナリオ設定＞

PureFlow(A)> add scenario “/port1/woc1” action wan-accel peer 20.1.2.9

＜LAN 側アクセル対象のフィルタ設定＞

PureFlow(A)> add filter scenario “/port1/woc1” filter “F1” ipv4 sip 40.8.2.3

PureFlow(A)> add filter scenario “/port1/woc1” filter “F2” ipv4 sip 49.1.8.11

PureFlow2 の設定

以下のコマンドを実行します。

＜チャネル設定＞

PureFlow(A)> add channel “ch1” lan 1/1 wan 1/2 vid none

＜デフォルトチャネル設定＞

PureFlow(A)> add channel “ch10000” lan 1/1 wan 1/2 default

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow(A)> set ip channel “ch1” 20.1.2.9 netmask 255.255.255.0

＜WAN 側のルート設定＞

PureFlow(A)> add route target 20.1.5.0 netmask 255.255.255.0 gateway 20.1.2.1 channel “ch1” wan
PureFlow(A)> add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.2.1 channel “ch1” wan
PureFlow(A)> add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.2.1 channel “ch1” wan

＜LAN 側のルート設定＞

PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.2.2 channel “ch1” lan
PureFlow(A)> add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.2.2 channel “ch1” lan

＜シナリオ設定＞

PureFlow(A)> add scenario “/port1/woc1” action wan-accel peer 20.1.5.9

＜LAN 側アクセル対象のフィルタ設定＞

PureFlow(A)> add filter scenario “/port1/woc1” filter “F1” ipv4 sip 30.2.1.7
PureFlow(A)> add filter scenario “/port1/woc1” filter “F2” ipv4 sip 35.9.8.21

［Case 4］シングルネットワーク（QoS あり）

図 8.13-5　Case 4 の構成とコンフィギュレーション例

PureFlow1 の設定

以下のコマンドを実行します。

＜チャネル設定＞

PureFlow（A）> add channel “ch1” lan 1/1 wan 1/2 vid none

＜デフォルトチャネル設定＞

PureFlow（A）> add channel “ch10000” lan 1/1 wan 1/2 default

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow（A）> set ip channel “ch1” 20.1.5.9 netmask 255.255.255.0

＜WAN 側のルート設定＞

PureFlow（A）> add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

PureFlow（A）> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

PureFlow（A）> add route target 32.3.2.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

PureFlow（A）> add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.5.1　channel “ch1” wan

＜LAN 側のルート設定＞

PureFlow(A)> add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.5.2　channel "ch1" lan

PureFlow(A)> add route target 43.12.3.0 netmask 255.255.255.0 gateway 20.1.5.2　channel "ch1" lan

PureFlow(A)> add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.5.2　channel "ch1" lan

＜シナリオ設定＞

PureFlow(A)> update scenario "/port1" action aggregate peak_bw 400M

PureFlow(A)> add scenario "/port1/dc-ny" action aggregate min_bw 100M peak_bw 300M

PureFlow(A)> add scenario "/port1/dc-ny/server1" action aggregate min_bw 100M peak_bw 100M class 2

PureFlow(A)> add scenario "/port1/dc-ny/server2" action aggregate min_bw 100M peak_bw 100M class 2

PureFlow(A)> add scenario "/port1/dc-ny/server1/woc1" action wan-accel peer 20.1.2.9

PureFlow(A)> add scenario "/port1/dc-ny/server2/woc2" action wan-accel peer 20.1.2.9

PureFlow(A)> add scenario "/port1/dc-ny/woc3" action wan-accel peer 20.1.2.9

＜LAN 側アクセル対象, QoS のフィルタ設定＞

PureFlow(A)> add filter scenario "/port1/dc-ny" filter "F0" ipv4

PureFlow(A)> add filter scenario "/port1/dc-ny/server1" filter "F1-2" ipv4 sip 40.8.2.3

PureFlow(A)> add filter scenario "/port1/dc-ny/server2" filter "F2-2" ipv4 sip 43.12.3.65

PureFlow(A)> add filter scenario "/port1/dc-ny/server1/woc1" filter "F1-3" ipv4 sip 40.8.2.3

PureFlow(A)> add filter scenario "/port1/dc-ny/server2/woc2" filter "F2-3" ipv4 sip 43.12.3.65

PureFlow(A)> add filter scenario "/port1/dc-ny/woc3" filter "F3-2" ipv4 sip 49.1.8.11

＜アプリケーション高速化設定　対象アプリケーション:SMB＞

PureFlow(A)> add apl-accel scenario "/port1/dc-ny/server2/woc2" protocol smb

PureFlow2 の設定

以下のコマンドを実行します。

＜チャネル設定＞

PureFlow(A)> add channel "ch1" lan 1/1 wan 1/2 vid none

＜デフォルトチャネル設定＞

PureFlow(A)> add channel "ch10000" lan 1/1 wan 1/2 default

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow(A)> set ip channel "ch1" 20.1.2.9 netmask 255.255.255.0

＜WAN 側のルート設定＞

PureFlow(A)> add route target 20.1.5.0 netmask 255.255.255.0 gateway 20.1.2.1　channel "ch1" wan
PureFlow(A)> add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.2.1　channel "ch1" wan
PureFlow(A)> add route target 43.12.3.0 netmask 255.255.255.0 gateway 20.1.2.1　channel "ch1" wan
PureFlow(A)> add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.2.1　channel "ch1" wan

＜LAN 側のルート設定＞

PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.2.2　channel "ch1" lan
PureFlow(A)> add route target 32.3.2.0 netmask 255.255.255.0 gateway 20.1.2.2　channel "ch1" lan
PureFlow(A)> add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.2.2　channel "ch1" lan

＜シナリオ設定＞

PureFlow(A)> update scenario "/port1" action aggregate peak_bw 400M
PureFlow(A)> add scenario "/port1/dc-ny" action aggregate min_bw 1G peak_bw 300M
PureFlow(A)> add scenario "/port1/dc-ny/server1" action aggregate min_bw 100M peak_bw 100M class 2
PureFlow(A)> add scenario "/port1/dc-ny/server2" action aggregate min_bw 100M peak_bw 100M class 2
PureFlow(A)> add scenario "/port1/dc-ny/server1/woc1" action wan-accel peer 20.1.5.9
PureFlow(A)> add scenario "/port1/dc-ny/server2/woc2" action wan-accel peer 20.1.5.9
PureFlow(A)> add scenario "/port1/dc-ny/woc3" action wan-accel peer 20.1.5.9

＜LAN 側アクセル対象，QoS のフィルタ設定＞

PureFlow(A)> add filter scenario "/port1/dc-ny" filter "F0" ipv4
PureFlow(A)> add filter scenario "/port1/dc-ny/server1" filter "F1-2" ipv4 sip 30.2.1.7
PureFlow(A)> add filter scenario "/port1/dc-ny/server2" filter "F2-2" ipv4 sip 32.3.2.3
PureFlow(A)> add filter scenario "/port1/dc-ny/server1/woc1" filter "F1-3" ipv4 sip 30.2.1.7
PureFlow(A)> add filter scenario "/port1/dc-ny/server2/woc2" filter "F2-3" ipv4 sip 32.3.2.3
PureFlow(A)> add filter scenario "/port1/dc-ny/woc3" filter "F3-2" ipv4 sip 35.9.8.21

＜アプリケーション高速化設定　対象アプリケーション：SMB＞

PureFlow(A)> add apl-accel scenario "/port1/dc-ny/server2/woc2" protocol smb

[Case 5]Out of Path 接続

図 8.13-6　Case 5 の構成とコンフィギュレーション例

PureFlow1 の設定

以下のコマンドを実行します。

＜シナリオツリーモード設定＞

PureFlow(A)> set scenario tree mode outbound

＜チャネル設定＞

PureFlow(A)> add channel “ch1” lan 1/1 wan 1/1 vid none

＜デフォルトチャネル設定＞

PureFlow(A)> add channel “ch10000” lan 1/1 wan 1/1 default

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow(A)> set ip channel “ch1” 20.1.5.9 netmask 255.255.255.0

＜WAN 側のルート設定＞

PureFlow(A)> add route target 24.1.2.0 netmask 255.255.255.0 gateway 20.1.5.2　channel “ch1” wan

PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.2　channel “ch1” wan

PureFlow(A)> add route target 32.3.2.0 netmask 255.255.255.0 gateway 20.1.5.2　channel “ch1” wan

PureFlow(A)> add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.5.2　channel “ch1” wan

＜LAN 側のルート設定＞

PureFlow(A)> add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.5.2　channel "ch1" lan

PureFlow(A)> add route target 43.12.3.0 netmask 255.255.255.0 gateway 20.1.5.2　channel "ch1" lan

PureFlow(A)> add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.5.2　channel "ch1" lan

＜シナリオ設定＞

PureFlow(A)> update scenario "/port1" action aggregate peak_bw 400M

PureFlow(A)> add scenario "/port1/dc-ny" action aggregate min_bw 100M peak_bw 100M class 2

PureFlow(A)> add scenario "/port1/dc-la" action aggregate min_bw 100M peak_bw 100M class 2

PureFlow(A)> add scenario "/port1/dc-la/woc" action wan_accel peer 24.1.2.9

＜ルールリストの設定＞

PureFlow(A)> add rulelist group "ny-serv" ipv4

PureFlow(A)> add rulelist entry "ny-serv" ipv4 40.8.2.3

PureFlow(A)> add rulelist entry "ny-serv" ipv4 43.12.3.65

PureFlow(A)> add rulelist entry "ny-serv" ipv4 49.1.8.11

＜LAN 側アクセル対象，QoS のフィルタ設定＞

PureFlow(A)> add filter scenario "/port1/dc-ny" filter "F1-lan" ipv4 dip list "ny-serv"

PureFlow(A)> add filter scenario "/port1/dc-la" filter "F2-wan" ipv4 sip list "ny-serv"

PureFlow(A)> add filter scenario "/port1/dc-la/woc" filter "F2-wan-1" ipv4 sip list "ny-serv"

PureFlow2 の設定

以下のコマンドを実行します。

＜シナリオツリーモード設定＞

PureFlow(A)> set scenario tree mode outbound

＜チャネル設定＞

PureFlow(A)> add channel "ch1" lan 1/1 wan 1/1 vid none

＜デフォルトチャネル設定＞

PureFlow(A)> add channel "ch10000" lan 1/1 wan 1/1 default

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow(A)> set ip channel "ch1" 24.1.2.9 netmask 255.255.255.0

＜WAN 側のルート設定＞

PureFlow(A)> add route target 20.1.5.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" wan
PureFlow(A)> add route target 40.8.2.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" wan
PureFlow(A)> add route target 43.12.3.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" wan
PureFlow(A)> add route target 49.1.8.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" wan

＜LAN 側のルート設定＞

PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" lan
PureFlow(A)> add route target 32.3.2.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" lan
PureFlow(A)> add route target 35.9.8.0 netmask 255.255.255.0 gateway 24.1.2.2　channel "ch1" lan

＜シナリオ設定＞

PureFlow(A)> update scenario "/port1" action aggregate peak_bw 400M
PureFlow(A)> add scenario "/port1/dc-la" action aggregate min_bw 100M peak_bw 100M class 2
PureFlow(A)> add scenario "/port1/dc-ny" action aggregate min_bw 100M peak_bw 100M class 2
PureFlow(A)> add scenario "/port1/dc-ny/woc" action wan_accel peer 20.1.5.9

＜ルールリストの設定＞

PureFlow(A)> add rulelist group "la-serv" ipv4
PureFlow(A)> add rulelist entry "la-serv" ipv4 30.2.1.7
PureFlow(A)> add rulelist entry "la-serv" ipv4 32.3.2.3
PureFlow(A)> add rulelist entry "la-serv" ipv4 35.9.8.21

＜LAN 側アクセル対象，QoS のフィルタ設定＞

PureFlow(A)> add filter scenario "/port1/dc-la" filter "F1-lan" ipv4 dip list "la-serv"
PureFlow(A)> add filter scenario "/port1/dc-ny" filter "F2-wan" ipv4 sip list "la-serv"
PureFlow(A)> add filter scenario "/port1/dc-ny/woc" filter "F2-wan-1" ipv4 sip list "la-serv"

8.14 さらに高度な設定

本装置には，さらに高度な設定として，以下の設定があります。

- フローおよびフロー識別モード
- キュー
- 通信ギャップモード
- ピークバーストサイズ
- トラフィックアクセラレーションバイパス
- トラフィックアクセラレーションの冗長構成
- TCP-FEC 機能
- TCP 輻輳制御機能
- リマーキング機能

8.14.1 フローおよびフロー識別モード

フローとは，装置内で識別できるトラフィックの最小単位です。トラフィックは，複数のフローからなるグループと考えることができます。

本装置は，パケットを受信すると，そのパケットを転送するためのフローを登録します。登録したフローは，フィルタに設定した動作に従ってキューにパケットを格納し，トラフィックコントロールします。

フローには，BridgeControl フロー，EthernetType フロー，IPv4 フロー，および IPv6 フローの 4 種類があります。

（1）BridgeControl フロー

BridgeControl フローは，Bridge-ctrl フィルタによって識別するフローです。宛先 MAC アドレスが 01-80-C2-00-00-00～01-80-C2-00-00-FF であるフレームを，入力ポートごとにひとつのフローに集約します。

（2）EthernetType フロー

EthernetType フローは，Ethernet フィルタによって識別するフローです。以下の Ethernet フィールドの組み合わせでフロー識別します。

- VLAN ID（VLAN Tag あり／なしも識別）
- CoS
- Ethernet Type

（3）IPv4/IPv6 フロー

IPv4/IPv6 フローは，IPv4/IPv6 フィルタによって識別するフローです。以下の IP パケットフィールドの組み合わせでフロー識別します。

- VLAN ID（VLAN Tag あり／なしも識別）
- CoS
- 送信元 IP アドレス（SIP）
- 宛先 IP アドレス（DIP）
- ToS またはトラフィッククラス
- プロトコル番号
- 送信元ポート番号（Sport）
- 宛先ポート番号（Dport）

注：

1. 装置内部には最大 512,000 フロー（BridgeControl フロー，EthernetType フロー，および IPv4/IPv6 フローの合計）を同時に作成，帯域制御に用いることができます。
2. BridgeControl フローは，ポートに対して 1 つのみです。

フロー識別モードを設定することにより，EthernetType フローおよび IPv4/IPv6 フローでフロー識別するフィールドを選択することができます。

たとえば，通常 IPv4 フローは，送信元 IP アドレス（SIP），宛先 IP アドレス（DIP），プロトコル番号（Protocol），送信元ポート番号（SPort），宛先ポート番号（DPort）がすべて一致するトラフィックです。

SIP	DIP	Protocol	SPort	DPort
192.168.3.10	192.168.8.7	TCP	7662	23
192.168.3.10	192.168.8.7	TCP	80	1833
192.168.3.10	192.168.8.7	TCP	80	1833
192.168.3.10	192.168.8.10	TCP	236	1423

図 8.14.1-1　フロー識別モード

本装置は，このフローを識別するフィールドの組み合わせ（フロー識別モード）を変更することが可能です。各フィールドが異なるパケットを異なるフローとして転送したり，同じフローとして転送したりすることができます。

フロー識別モードに設定できるパラメータを以下に示します。

表 8.14.1-1　フロー識別モードのパラメータ

パラメータ	設定範囲	省略可能／不可
入力 Network ポート	1/1／1/2／1/3／1/4	不可
フィールド名	default　：フローの識別フィールドをデフォルトにします。VLAN ID，インナーVLAN ID，送信元 IP アドレス，宛先 IP アドレス，プロトコル番号，送信元ポート番号，宛先ポート番号をフロー識別します（注）。 vid　：VLAN ID（IEEE802.1q）または 2 重 VLAN タグ（IEEE802.1ad）の外側 VLAN ID をフロー識別します。 cos　：CoS（IEEE802.1q）または 2 重 VLAN タグ（IEEE802.1ad）の外側 CoS をフロー識別します。 inner-vid：2 重 VLAN タグの内側 VLAN ID をフロー識別します。 inner-cos：2 重 VLAN タグの内側 CoS をフロー識別します。 sip　：送信元 IP アドレスをフロー識別します。 dip　：宛先 IP アドレスをフロー識別します。 tos　：ToS または Traffic Class をフロー識別します。 proto　：プロトコル番号をフロー識別します。 sport　：送信元ポート番号をフロー識別します。 dport　：宛先ポート番号をフロー識別します。	不可

注：

トラフィックアクセラレーションを行う場合，“default”を指定してください。

本パラメータは，カンマ（，）で区切って複数指定することができます。

フロー識別モードに関する CLI は以下のコマンドがあります。

set filter mode in <slot/port> <field>	フローの識別フィールドを選択します。 <field>のデフォルト値は“default”です。

コマンドの実行例を示します。

```
PureFlow(A)> set filter mode in 1/1 cos
PureFlow(A)> set filter mode in 1/2 sip,dip
PureFlow(A)>
```

たとえば，送信元 IP アドレスと宛先 IP アドレスのみでフローを識別し，その他のフィールドが異なる IPv4 パケットは同じ IPv4 フローとしてトラフィックコントロールしたい場合，sip と dip を有効にします。このフロー識別モードの場合，“add filter”で登録した IPv4 フィルタの条件は，送信元 IP アドレス，宛先 IP アドレスがフィルタ対象となります。フロー識別モードで指定したフィールド以外のフィールドが設定されている IPv4 フィルタは，無効と見なします。

図 8.14.1-2　フロー識別モードで sip と dip を有効にした場合

指定フィールド名とフローで識別するフィールドの関係を，以下に示します。

表 8.14.1-2　指定フィールド名とフローで識別するフィールドの関係

指定フィールド名	フロー識別フィールド									
	VLAN ID	CoS	インナー VLAN ID	インナー CoS	SIP	DIP	ToS	プロトコル番号	Sport 番号	Dport 番号
default	○	×	○	×	○	○	×	○	○	○
vid	○	×	×	×	×	×	×	×	×	×
cos	×	○	×	×	×	×	×	×	×	×
inner-vid	×	×	○	×	×	×	×	×	×	×
inner-cos	×	×	×	○	×	×	×	×	×	×
sip	×	×	×	×	○	×	×	×	×	×
dip	×	×	×	×	×	○	×	×	×	×
tos	×	×	×	×	×	×	○	×	×	×
proto	×	×	×	×	×	×	×	○	×	×
sport	×	×	×	×	×	×	×	×	○	×
dport	×	×	×	×	×	×	×	×	×	○

○：フロー識別する
×：フロー識別しない

8.14.2 キュー

本装置は，各フローに対してキューを割り当て，受信したパケットを割り当てたキューに格納します。キューに格納したパケットはスケジューリングされ，トラフィックコントロール転送されます。

(1) デフォルトキュー

任意のレベル n シナリオ内で，それに属する下位のレベル n シナリオに該当しないフローを転送するためのキューです。デフォルトキューは，ベストエフォートクラス(クラス 8)となります。

任意のレベルフィルタに一致し，それに属する下位のレベルフィルタに一致しないすべてのフローを同じデフォルトキューに割り当て，トラフィックコントロールを行います。

たとえば，レベル 2 シナリオで保証帯域 100 Mbit/s に設定した場合は，以下のようになります。

本装置に，以下のフィルタを登録したと仮定します。

- レベル 2 フィルタ
  送信元 IP アドレス ： 192.168.0.0 - 192.168.255.255
  宛先 IP アドレス ： 192.168.0.0 - 192.168.255.255
- レベル 3 フィルタ
  送信元 IP アドレス ： 192.168.10.0 - 192.168.10.255
  宛先 IP アドレス ： 192.168.10.0 - 192.168.10.255

また，以下の 3 つのトラフィックが入力されたと仮定します。

- 192.168.1.1 から 192.168.1.100 へのトラフィック(フロー 1)
- 192.168.1.1 から 192.168.1.150 へのトラフィック(フロー 2)
- 192.168.1.1 から 192.168.1.200 へのトラフィック(フロー 3)

これらのフローは，レベル 2 フィルタに一致し，レベル 3 フィルタに一致しないため，デフォルトキューにパケットを格納します。

- フロー 1～3 の合計が 100 Mbit/s

レベル 2 シナリオとして，合計 100 Mbit/s の帯域を保証します。

図 8.14.2-1　デフォルトキュー

ただし，優先度が高いクラスのレベル 3 キューに割り当てられたフローが流れている場合，デフォルトキューに割り当てられたフローの合計は 100 Mbit/s の帯域を保証できません。

(2) 集約キュー(レベル n キュー)

Aggregate(集約キューモード)のレベル n シナリオとは，レベル n フィルタに一致した複数のフローを 1 つのレベル n キューに集約して割り当てる方式です。

レベル n フィルタに一致し，その下位に属するレベル n フィルタにも一致したすべてのフローを同じレベル n キューに割り当て，トラフィックコントロールを行います。

たとえば，送信元 IP アドレスが 192.168.10.1 で，宛先 IP アドレスが 192.168.10.100，192.168.10.150，192.168.10.200 の場合に，レベル n シナリオの集約キューで最大帯域 10 Mbit/s に設定した場合は，以下のようになります。

本装置に，以下のフィルタを登録したと仮定します。

- レベル 2 フィルタ
  送信元 IP アドレス ： 192.168.0.0 - 192.168.255.255
  宛先 IP アドレス ： 192.168.0.0 - 192.168.255.255
- レベル 3 フィルタ
  送信元 IP アドレス ： 192.168.10.0 - 192.168.10.255
  宛先 IP アドレス ： 192.168.10.0 - 192.168.10.255

また，以下の 3 つのトラフィックが入力されたと仮定します。

- 192.168.10.1 から 192.168.10.100 へのトラフィック(フロー 4)
- 192.168.10.1 から 192.168.10.150 へのトラフィック(フロー 5)
- 192.168.10.1 から 192.168.10.200 へのトラフィック(フロー 6)

これらのフローは，レベル 2 フィルタに一致し，レベル 3 フィルタにも一致するため，レベル 3 キュー(集約キュー)にパケットを格納します。

- フロー 4～6 の合計が 10 Mbit/s

レベル 3 シナリオとして，合計 10 Mbit/s の帯域を使用します。

図 8.14.2-2　集約キュー

(3) 個別キュー(レベル n キュー)

Individual(個別キューモード)のレベル n シナリオとは，レベル n フィルタに一致した複数のフローに対して，個別のレベル n キューを割り当てる方式です。

レベル n フィルタに一致したすべてのフローごとに個別のレベル n キューを割り当て，トラフィックコントロールを行います。下位レベルにシナリオを登録することはできますが，Individual シナリオの下位レベルシナリオにフローが割り当てられることはありません。

たとえば，送信元 IP アドレスが 192.168.20.1 で，宛先 IP アドレスが 192.168.20.100，192.168.20.150，192.168.20.200 の場合に，レベル n シナリオの個別キューで最大帯域 10 Mbit/s に設定した場合は，以下のようになります。

本装置に，以下のフィルタを登録したと仮定します。

- レベル 2 フィルタ
  送信元 IP アドレス ： 192.168.0.0 - 192.168.255.255
  宛先 IP アドレス ： 192.168.0.0 - 192.168.255.255
- レベル 3 フィルタ
  送信元 IP アドレス ： 192.168.20.0 - 192.168.20.255
  宛先 IP アドレス ： 192.168.20.0 - 192.168.20.255

また，以下の 3 つのトラフィックが入力されたと仮定します。

- 192.168.20.1 から 192.168.20.100 へのトラフィック(フロー 7)
- 192.168.20.1 から 192.168.20.150 へのトラフィック(フロー 8)
- 192.168.20.1 から 192.168.20.200 へのトラフィック(フロー 9)

これらのフローは，レベル 2 フィルタに一致し，レベル 3 フィルタにも一致するため，レベル 3 キュー(個別キュー)にパケットを格納します。

- フロー 7 は 10 Mbit/s
- フロー 8 は 10 Mbit/s
- フロー 9 は 10 Mbit/s

レベル 3 シナリオとして，合計 30 Mbit/s の帯域を使用します。

図 8.14.2-3　個別キュー

注:

モニタリングマネージャにおいて，個別キューモードのシナリオは集約キューモードと同様に 1 つのキューとして表示されます。個別キューは表示されません。

個別キューモードの場合，シナリオで割り当てる個別キューの最大数を設定することが可能です。個別キューの最大数を超えてフローを作成する場合は，キュー最大数超過アクション（ベストエフォート転送，トラフィックアトリビュート転送，または廃棄）に従います。

たとえば，上記の例で，レベル 3 シナリオの個別キュー最大数 3，キュー最大数超過アクション forwardbesteffort とします。

本装置に，Flow 7～Flow 9 に加えて，以下のトラフィックが入力されたと仮定します。

- 192.168.20.1 から 192.168.20.250 へのトラフィック（Flow 10）

このフローは，レベル 2 フィルタに一致し，レベル 3 フィルタにも一致しますが，すでに個別キューを 3 個割り当てているため，キュー最大数超過アクション（ベストエフォート転送）に従います。

図 8.14.2-4　キュー最大数超過アクション（forwardbesteffort）

なお，デフォルトキューのキューバッファサイズは 1Mbyte（固定）です。

また，キュー最大数超過アクション discard の場合は，Flow 10 のトラフィックを廃棄します。

(4) バッファサイズ

レベル n キューは，バッファサイズを設定できます。

バッファサイズは，キューで許容できる入力バースト長です。バーストでパケット受信したときに，キューに格納できるバイト数です。

図 8.14.2-5　バッファサイズ

入力バースト長が，バッファサイズを超えてしまうと，パケットを廃棄します。バッファサイズ不足により，パケットが廃棄されてしまう場合，レベル n シナリオ（トラフィックアトリビュート）でバッファサイズを設定してください。

パケットが廃棄されているかどうかは，キュー統計情報で確認することができます。（詳細は「第 12 章 統計情報」を参照してください。）

デフォルトキューおよびレベル n シナリオで割り当てるレベル n キューのバッファサイズは，バイト指定で設定します。

以下にレベル n シナリオで割り当てるレベル n キューのバッファサイズを変更するコマンドを示します。

Sample 1）　すでに存在するレベル 2 シナリオに対して，バッファサイズ 5M バイトに変更する場合

```
PureFlow(A)> update scenario "/port1/Tokyo" action aggregate bufsize 5M
```

Sample 2）　すでに存在するレベル 3 シナリオに対して，バッファサイズ 2M バイトに変更する場合

```
PureFlow(A)> update scenario "/port1/Tokyo/Shinjuku" action aggregate bufsize 2M
```

注：

システムパケットバッファとキューバッファの関係性

各キューは，キューで使用可能な帯域を超過してしまったパケットをバッファに蓄積する際，設定したバッファサイズまでシステムパケットバッファを動的に使用します。キューバッファは，システムパケットバッファの中から固定に確保されるわけではありません。シナリオのキューバッファサイズの設定値合計は，システムパケットバッファサイズを超過し設定することができます。ただし，同時に使用できるシ

ナリオのキューバッファサイズの合計は，システムパケットバッファサイズまでです。各シナリオのキューバッファは，パケットの入力順に，設定したバッファサイズを上限とし，システムパケットバッファを使用します。

(5) クラス

レベル 2 以降のキューには，クラス(キューの優先順位)を設定することが可能です。

本装置のトラフィックコントロール方式は，8 クラス(クラス 1～8)の優先度に基づくキュー間を優先度の高いものから出力していく方式(Strict Priority)です。

以下に Strict Priority 動作を示します。

本装置に以下のレベル 2，3 キューを割り当てたものと仮定します。

- レベル 2 キュー(クラス 8，保証帯域 100 Mbit/s)
- レベル 3 キュー1(クラス 1，最低帯域 60 Mbit/s／最大帯域 80 Mbit/s)
- レベル 3 キュー2(クラス 1，最低帯域 20 Mbit/s／最大帯域制限なし)
- レベル 3 キュー3(クラス 1，最低帯域保証なし／最大帯域 20 Mbit/s)
- レベル 3 キュー4(クラス 2，最低帯域 20 Mbit/s／最大帯域 30 Mbit/s)

図 8.14.2-6　クラス

a)　　レベル 2 シナリオは，帯域を保証します。

たとえば，レベル 2 シナリオ外で 990 Mbit/s のフローが流れている場合でも，レベル 2 シナリオ内のフローは 100 Mbit/s を保証します。

ただし，各レベル 2 シナリオに割り当てた保証帯域の合計がレベル 1 シナリオの帯域を超えている場合，レベル 2 シナリオの帯域を保証できません。

b)　最低帯域保証ありのレベル 3 キューに割り当てられたフローは，最低帯域を保証します。

たとえば，フロー3（100 Mbit/s）のフローが流れている場合でも，フロー1（60 Mbit/s）のフローは 60 Mbit/s，フロー2（20 Mbit/s）のフローは 20 Mbit/s でトラフィックコントロールします。

ただし，各レベル 3 シナリオに割り当てた最低帯域の合計が，レベル 2 シナリオの保証帯域を超えている場合，レベル 3 シナリオの最低帯域を保証できません。

c)　同じレベル 2 シナリオ内に，複数のクラスのレベル 3 キューを割り当てた場合，優先度が低いクラスのレベル 3 キューのフローは，最低帯域を保証できません。優先度が低いクラスのレベル 3 キューは，優先度が高いクラスの余剰帯域でトラフィックコントロールします。

たとえば，フロー1（60 Mbit/s），フロー2（20 Mbit/s），フロー3（15 Mbit/s）のフロー（クラス 1）と，フロー4（20 Mbit/s）のフロー（クラス 2）が流れている場合，フロー4 は 5 Mbit/s でトラフィックコントロールします。

d)　最大帯域制限ありのレベル 3 キューに割り当てられたフローは，その最大帯域で制限します。

たとえば，フロー3（30 Mbit/s）が流れている場合は，フロー3 は 20 Mbit/s でトラフィックコントロールします。

また，レベル 3 キューの最大帯域がレベル 2 シナリオの保証帯域を超えている場合，レベル 2 シナリオの保証帯域でトラフィックコントロールします。

e)　最大帯域制限なしのレベル 3 キューに割り当てられたフローは，レベル 2 シナリオの保証帯域でトラフィックコントロールします。

たとえば，フロー2（120 Mbit/s）が流れている場合，フロー2 は 100 Mbit/s でトラフィックコントロールします。

レベル 3 キューに優先度をつけると，優先度の高いクラスのキューに格納されたパケットを優先して転送しますので，優先度が低いクラスに比べて揺らぎが小さくなります。レベル 3 キューに優先度をつけたい場合，レベル 3 シナリオ（トラフィックアトリビュート）でクラスを設定してください。

以下にレベル 3 シナリオのクラスを変更するコマンドを示します。

Sample）　すでに存在するレベル 3 シナリオに対して，クラス 1 に変更する場合

```
PureFlow(A) > update scenario "/port1/Tokyo/Shinjuku" action aggregate class 1
```

注:

CLI コマンドなどによるシナリオのクラス変更は，対象シナリオが 1 パケット送信したあとに反映されます。優先度の高い他シナリオが帯域を占有している状態では，対象シナリオがパケットを送出できないため，クラス変更が反映されません。クラスの変更は帯域に余裕がある（最大帯域に達していない）状態で行ってください。

8.14.3 通信ギャップモード

Ethernetは，フレームを連続して送信する場合，フレームとフレームの間にギャップとプリアンブルが挿入されます。トラフィックアトリビュート（シナリオ，Network ポート）の帯域を設定するときに，これらを含めてトラフィックコントロール（ネットワーク帯域全体）を行うか，または含めないでトラフィックコントロール（フレームのみを対象）を行うかを選択することができます。本設定は装置全体に適用します。

図 8.14.3-1　イーサネットフレームのギャップとプリアンブルについて

通信ギャップモード設定に関する CLI を示します。

表 8.14.3-1　通信ギャップモード設定に関する CLI コマンド

set bandwidth mode {gap [<size>] | no_gap}	通信帯域設定で，フレーム間ギャップとプリアンブルの有効／無効を選択します。 デフォルト値は“gap”（有効）です。 gap の場合は，フレーム間ギャップおよびプリアンブルを帯域に含み，サイズを指定することができます。サイズの設定範囲は–100[Byte]～100[Byte]です。サイズを 0 に設定するとno_gap と同意になります。

コマンドの実行例を示します。

```
PureFlow(A)> set bandwidth mode gap
PureFlow(A)>
```

通信ギャップモードを有効としたときは，トラフィックアトリビュート（シナリオ，Network ポート）の帯域設定値による制御がフレーム間ギャップとプリアンブルを含めた制御となります。この設定は，帯域設定値が物理回線と同じ数値の意味を示しますので，出力 WAN 回線の帯域に対する輻輳回避や，トラフィックの優先制御を実施する場合に有効です。

通信ギャップモードを無効としたときは，トラフィックアトリビュート（シナリオ，Network ポート）の帯域設定値による制御がフレーム間ギャップとプリアンブルを含めないイーサネットフレームのみのデータレートとして制御します。この設定は，一般的にフレーム間ギャップやプリアンブルを含まないデータレートで示されているコンテンツ，映像，音声などのバースト回避のための平滑化，サーバに対して受信レートを制御するなどのコンテンツレート制御に有効です。

通信ギャップモードを無効で使用する場合は，トラフィックアトリビュート（シナリオ，Network ポート）の帯域設定値が回線帯域と異なる出力レートとなるので，通信ギャップを考慮した帯域設定値にする必要があります。たとえば回線帯域が 100 Mbit/s の場合，すべてのフレーム長（64 バイト～1522 バイト）においてフレーム落ちなく転送できる設定値は約 76 Mbit/s（（100 Mbit/s）×（64 byte/84 byte））になります。この場合，いかなるフレーム長においても 76 Mbit/s に制限するので，フレーム長が長いほど転送量に無駄が生じることになります。回線帯域を無駄なく使用する場合は，通信ギャップモードを有効に設定し，フレーム間ギャップを含めた帯域を設定してください。

注:

通信ギャップモードの設定値はパケットごとにパケット受信時に適用されます。通信ギャップモード変更時に各シナリオバッファに滞留しているパケットには適用されません。このため，通信ギャップモードの変更は，変更時に滞留していたパケットを排出したあとに反映されます。

8.14.4 ピークバーストサイズ

本装置のトラフィックコントロール（最大帯域制限）は，トークンバケット方式です。トークンバケット方式は，連続（バースト）で受信したパケットを格納するためのバッファと，一定時間ごとに最大帯域の制限レートに即した値（トークン）が加算されるバケットを用意しています。バケット中のトークンの積算値が送出するパケットサイズに対して小さい場合は，送出レートが最大帯域の制限レートを超えているためパケットをバッファに滞留し続けます。トークンの積算値が送出するパケットサイズより大きくなるとパケットを送出します。パケットを送出するとき，バケットのトークン積算値から，パケットサイズを減算することにより，次のパケットは，トークンが溜まるまで送出が抑制されます。このように，バケットに溜まったトークンの積算値によりパケットの送出タイミングを調整することで，送出レートを調整します。

バッファにパケットが滞留している場合は，一定時間ごとのトークン加算のタイミングで送出しますが，一定期間パケット受信がなく，バッファにパケットが滞留していない場合には，すでにバケットにトークンがパケット送出に十分な値に積算されているので，パケットはバッファに滞留することなく，受信と同時にバースト送信します。

本装置のトークンバケットは，トークンを一定間隔に固定値を加算するのではなく，パケットの受信および送出タイミングで加減算します（トークン加減算に時間とパケットサイズを可変値として用います）。これにより，高精度なトラフィックシェーピングを可能としています。

図 8.14.4-1　バッファに滞留していない場合の送出タイミング

例えば，1 ミリ秒ごとに 100 バイトのトークンを加算するトークンバケットがあった場合には，1500 バイトのパケットは 15 ミリ秒間隔で出力します。しかし，バッファが空になり，150 ミリ秒パケットの受信がない場合には，トークンは加算され続け，15,000 バイトにまで積算され，パケットは 10 パケットまでバッファに滞留することなく，出力してしまいます。そのため，パケットの受信がない状態においても，それ以上（一定値）トークンを積算させないトークン値の上限値がピークバーストサイズです。

先ほどの例で，1 ミリ秒ごとに 100 バイトのトークンを加算しますが，パケット受信がなければ 3,000 バイト以上には加算しない設定とします。その場合，バッファが空の状態で受信したバーストパケットは 2 パケットまでは連続（バースト）して出力されますが，3 パケット以降は，15 ミリ秒間隔で出力されます。

ピークバーストサイズは小さければ小さいほどバースト送信を制限します。

本装置のトラフィックコントロール（最大帯域制限）では，ピークバーストサイズの設定変更が可能です。送出バーストサイズを「ピークバーストサイズ＋最大フレーム長」以下になるように制御します。本設定は，すべてのシナリオに適用します。

ピークバーストサイズ設定に関する CLI を示します。

表 8.14.4-1　ピークバーストサイズ設定に関する CLI コマンド

set shaper peak burst size <size>	トラフィックコントロールのピークバーストサイズを設定します。 デフォルト値は“1536”[Byte]です。設定範囲は，Network ポートの最大フレーム長により異なります。最大フレーム長が 2048[Byte]の場合，設定範囲は 0[Byte]～9216[Byte]です。最大フレーム長が 10240[Byte]の場合，設定範囲は 0[Byte]～46080[Byte]です。 ピークバーストサイズを 0 に設定するとバースト出力しません。

コマンドの実行例を示します。

```
PureFlow(A)> set shaper peak burst size 3000
PureFlow(A)>
```

8.14.5 トラフィックアクセラレーションバイパス

トラフィックアクセラレーション実行の際に，対向装置と接続できない場合や対向装置間の RTT（Round Trip Time）が一定値未満の場合に，トラフィックアクセラレーションを行わずにバイパス転送します。

トラフィックアクセラレーションバイパスに設定できるパラメータを以下に示します。

表 8.14.5-1　トラフィックアクセラレーションバイパスのパラメータ

パラメータ	設定範囲	省略可能／不可	説明
機能の有効／無効	enable disable	不可	トラフィックアクセラレーションの自動バイパス機能の有効／無効を設定します。 有効の場合，以下の条件になるとバイパス転送状態になります。 ・TCP 接続エラー発生時 ・TCP 接続時の RTT がしきい値未満 ・Keep Alive 監視で ICMP 疎通異常 デフォルトは，有効です。 すべてのアクセラレーションモードシナリオに適用します。
バイパス回復時間	1～600 秒	不可	トラフィックアクセラレーションの自動バイパス機能のバイパス回復時間を設定します。 バイパス転送状態となった場合，本設定時間経過後の新規 TCP セッションについて，再度トラフィックアクセラレーションを試みます。 デフォルトは，60 秒です。 すべてのアクセラレーションモードシナリオに適用します。
トラフィックアクセラレーションの自動バイパスの RTT しきい値（bypass-thresh）	0～10000 ミリ秒	可能	省略時：0 トラフィックアクセラレーションの自動バイパス機能の RTT（Round Trip Time：往復遅延時間）しきい値をミリ秒単位で指定します。 デフォルトは，0 秒です。0 秒を指定すると，RTT によるバイパス転送は行わず，TCP 接続エラーによるバイパス転送を行います。 本パラメータは，アクセラレーションモードシナリオごとに指定可能です。
トラフィックアクセラレーションの自動バイパスの Keep Alive 監視の有効／無効（bypass-keepalive）	enable disable	可能	省略時：disable トラフィックアクセラレーションの自動バイパス機能の Keep Alive 監視の有効／無効を指定します。最大 100 個までのアクセラレーションモードシナリオに指定できます。 有効の場合，当該シナリオに指定した対向装置を ICMP により疎通監視します。疎通異常の場合，バイパス転送状態にします。バイパス転送状態においても疎通監視は継続し，疎通異常が回復するまではバイパス転送状態を維持します。 本パラメータは，アクセラレーションモードシナリオごとに指定可能です。

以下に，トラフィックアクセラレーションのバイパス設定に関する CLI コマンドを示します。

表 8.14.5-2　トラフィックアクセラレーションバイパス設定に関する CLI コマンド

set wan-accel bypass status {enable | disable}	トラフィックアクセラレーションの自動バイパス機能の有効／無効を設定します。
set wan-accel bypass recoverytime <duration>	トラフィックアクセラレーションの自動バイパスのバイパス回復時間を設定します。
add scenario <scenario_name> action wan-accel peer <IP_address> second-peer <IP_address> [dport <dport>] [vid <vid>]　[inner-vid <VID>] [compression {enable | disable}] [tcp-mem {auto | <size>] [cc-mode {normal | semi-fast | fast}] [bypass-thresh <rtt>] [bypass-keepalive {enable |disable}] [fec {enable |disable}] [block-size <size>] [data-block-size <size>] [fec-session <session>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [bufsize <bufsize>] [scenario <scenario_id>]	アクセラレーションモードのシナリオを登録します。 トラフィックアクセラレーションの自動バイパスの RTT しきい値（bypass-thresh），Keep Alive 監視（bypass-keepalive）の有効／無効を指定します。
update scenario <scenario_name> action wan-accel [vid <vid>]　[inner-vid <VID>] [compression {enable | disable}] [tcp-mem {auto | <size>] [cc-mode {normal | semi-fast | fast}] [bypass-thresh <rtt>] [bypass-keepalive {enable |disable}] [fec {enable |disable}] [block-size <size>] [data-block-size <size>] [fec-session <session>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [bufsize <bufsize>]	アクセラレーションモードのシナリオを変更します。 トラフィックアクセラレーションの自動バイパスの RTT しきい値（bypass-thresh），Keep Alive 監視（bypass-keepalive）の有効／無効を指定します。
switch wan-accel bypass force {enable | disable} all	トラフィックアクセラレーションの強制バイパス機能の有効／無効を設定します。 有効の場合，強制的にバイパス転送状態になります。
switch wan-accel bypass force {enable | disable} scenario <scenario_name>	指定したシナリオ名のトラフィックアクセラレーションの強制バイパス機能の有効／無効を設定します。 有効の場合，強制的にバイパス転送状態になります。
show wan-accel bypass	トラフィックアクセラレーションのバイパス情報を表示します。

RTT しきい値は，“add scenario”コマンドのアクションモード“wan-accel”でトラフィックアクセラレーションの自動バイパスの RTT しきい値を指定します。TCP 接続時に測定した RTT が設定した RTT しきい値未満の場合にバイパス転送します。通常は，6 ミリ秒に設定してください。RTT が 6 ミリ秒以内の場合，トラフィックアクセラレーションを適用しない方が高速転送できます。RTT が 6 ミリ秒を超える場合に，本装置のトラフィックアクセラレーションが効果的に機能します。

コマンドの実行例を示します。

```
PureFlow(A)> set wan-accel bypass status enable
PureFlow(A)> set wan-accel bypass recoverytime 30
```

トラフィックアクセラレーションバイパスに関する情報は，“show scenario info name”コマンドで表示します。

表 8.14.5-3　トラフィックアクセラレーションバイパスのパラメータ

トラフィックアクセラレーション バイパスのパラメータ	表示内容
Status	トラフィックアクセラレーションの自動バイパス機能の状態（有効／無効）を表示します。
Recovery time	バイパス転送状態となったシナリオが，トラフィックアクセラレーションを再試行するまでの時間を表示します。
State	トラフィックアクセラレーションの自動バイパス機能の，現在のシナリオ状態を表示します。 Standby　:トラフィック入力の待機中です。 Measuring　:RTT およびコネクション接続の測定中です。 Acceleration :トラフィックアクセラレーション適用中です。 Bypass　:バイパス転送中です。 Force Bypass:強制バイパス転送中です。
Threshold RTT	RTT しきい値を表示します。
Minimum RTT	RTT 測定値の最小値を表示します。本測定値が RTT しきい値を下回ると，トラフィックアクセラレーションを停止して，バイパス転送状態に移行します。
Low RTT	RTT しきい値未満の検出状態を表示します。 not detected :RTT しきい値未満の RTT を検出していません。 detected　:RTT しきい値未満の RTT を検出しました。
Connection Error	TCP 接続エラーの検出状態を表示します。 not detected :TCP 接続エラーを検出していません。 detected　:TCP 接続エラーを検出しました。
Keep Alive	トラフィックアクセラレーションの自動バイパス機能の Keep Alive 監視機能の状態（有効／無効）を表示します。
Keep Alive State	Keep Alive 監視の状態を表示します。 Alive　:Peer との疎通が正常であることを示します。 Timeout　:Peer との疎通がタイムアウトしたことを示します。 -----　:KeepAlive 監視を行っていないことを示します。
Acceleration Trans	トラフィックアクセラレーションのシナリオが，“Acceleration”状態へ遷移した累積回数を表示します。
Bypass Trans	トラフィックアクセラレーションのシナリオが，“Bypass”状態へ遷移した累積回数を表示します。

注:

トラフィックアクセラレーションの自動バイパス機能を使用する場合は，「TCP 高速化機能ライセンス（NF7500-L201A）」を購入していただく必要があります。

8.14.6 トラフィックアクセラレーションの冗長構成

本装置のトラフィックアクセラレーションは，ホットスタンバイ型の装置冗長構成で使用できます。冗長構成を行うためには，以下のようなネットワーク構成にします。

図 8.14.6-1　トラフィックアクセラレーションの冗長構成

WS1-1 と WS1-2-1 は通常動作でトラフィックアクセラレーションします（ここでは WS1-2-1 を Primary Peer と呼びます）。WS1-2-2 はホットスタンバイ状態で待機します（ここでは WS1-2-2 を Secondary Peer と呼びます）。通常，WS1-1 は Primary Peer とアクセラレーショントンネルを構成しますが，Primary Peer の装置が故障した場合や，WAN 側の経路の異常などで Primary Peer の装置へ通信できない場合に，Secondary Peer に切り替えます。
Client 側の WS1 にて Primary Peer／Secondary Peer のどちらを使用するか制御するため，Client 側の WS1 に対して Secondary Peer の設定を行います。上記の構成の場合，WS1-1 のシナリオに「second-peer（Secondary の対向装置の IP アドレス）」を設定します。また，WS1 を 4 台で 2 対 2 の構成とする場合は，双方向で「second-peer」を設定する必要があります。

注1：

トラフィックアクセラレーションの冗長構成とする場合は，「TCP 高速化機能ライセンス(NF7500-201A)」を購入していただく必要があります。

注2：

冗長構成で使用する場合は以下に注意してください。

① 対向装置（peer）への経路がレイヤ 3 で制御されている必要があります。

② ルータにて通常動作で使用される経路上の WS1 を Primary Peer に指定する必要があります。（たとえば，ルーティングプロトコルが OSPF の場合，OSPF の経路コストの調整により Primary Peer 側を優先する経路に設定してください）。

③ リンクダウン転送機能を有効に設定する必要があります（設定方法は「第 9 章 リンクダウン転送機能」を参照してください）。

切り替え時の動作を以下に示します。

Primary で動作（通常時）

図 8.14.6-2　通常時の動作

通常は WS1-1 と Primary Peer 装置間でアクセラレーショントンネルを構成します。
このとき，WS1-1 は Primary Peer 装置と ICMP により，3 秒に 1 回の間隔で疎通確認を行っています。

Secondary で動作（Primary 異常時）

図 8.14.6-3　装置異常時の切り替え動作

疎通確認が 3 回連続で失敗すると，WS1-1 は Primary Peer 装置と接続不能と判断して，Secondary Peer 装置と接続を行います。以降は，WS1-1 と Secondary Peer 装置間でアクセラレーショントンネルを構成します。

注 1：

WS1-1 と Primary Peer 装置でトラフィックアクセラレーション中の TCP セッションは，Secondary Peer へ切り替わりません。

Primary に切り戻し（Primary 復旧時）

図 8.14.6-4　装置復旧時の切り戻し動作

Secondary Peer 装置と接続している間も WS1-1 は Primary Peer 装置との疎通確認を継続します。Primary Peer 装置の異常（故障など）が回復して疎通確認が 3 回連続で成功すると，WS1-1 は Primary Peer 装置と接続可能と判断して，Primary Peer 装置に接続を切り戻します。以降は，WS1-1 と Primary Peer 装置と TCP アクセラレーショントンネルを復旧させます。

注 2：

WS1-1 と Secondary Peer 装置でトラフィックアクセラレーション中の TCP セッションは， Primary Peer へ切り替わりません。アクセラレーショントンネル経由の TCP セッションが終了するまで Secondary Peer 装置経由で接続します。

注 3：

冗長構成とトラフィックアクセラレーションのバイパス機能を併用した場合は，以下の手順で動作します。

① Primary Peer で障害発生　　：冗長構成機能により Secondary Peer へ切り替え

② Secondary Peer でも障害発生　：自動バイパス機能により TCP バイパス転送状態へ移行

冗長構成の場合，Primary Peer に対して ICMP による疎通監視を行いますが，Secondary Peer に対しては疎通監視を行いません。自動バイパスの Keep Alive 監視を有効にすることで，Secondary Peer と通信中に Secondary Peer に対して ICMP による疎通監視を行うことができます。

以下に，冗長構成時に使用する CLI コマンドおよびパラメータを示します。詳細な設定方法は「8.9 設定方法」の「STEP4：シナリオの設定」を参照してください。

表 8.14.6-1　冗長構成時に使用する CLI コマンドおよびパラメータ

コマンド	パラメータ	説明
add scenario	second-peer <IP_address>	Secondary Peer 装置の IP アドレスを指定します。
show scenario info name	<scenario name>	second-peer を指定したシナリオに対して実行することで，Primary/Secondary のどちらの Peer と接続しているか確認できます。
show syslog	なし	Primary と Secondary の接続状態をシステムログで確認できます。 Primary Peer から Secondary Peer への切り替え時は，以下のシステムログが記録されます。 「Wan-accel scenario switched to secondary-peer. [S:#M]」 Secondary Peer から Primary Peer への切り戻し時は，以下のシステムログが記録されます。 「Wan-accel scenario switched back to primary-peer. [S:#M]」 M には対象のシナリオ名が入ります。

参考として，以下に冗長構成時の設定例を示します。

図 8.14.6-5　冗長構成時の設定例

WS1-1-1 の設定

以下のコマンドを実行します。

＜デフォルトチャネル設定＞

PureFlow(A)> add channel “ch10000” lan 1/1 wan 1/2 default

＜チャネル設定＞

PureFlow(A)> add channel “ch1” lan 1/1 wan 1/2 vid none

＜リンクダウン転送機能の設定＞

PureFlow(A)> add lpt pair port 1/1 1/2

PureFlow(A)> set lpt enable

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow(A)> set ip channel “ch1” 20.1.2.10 netmask 255.255.255.0

＜LAN 側のルート設定＞

PureFlow(A)> add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.2.1 channel “ch1” lan

PureFlow(A)> add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.2.1 channel “ch1” lan

＜WAN 側のルート設定＞

PureFlow(A)> add route target 20.1.8.0 netmask 255.255.255.0 gateway 20.1.2.2 channel “ch1” wan

PureFlow(A)> add route target 20.1.9.0 netmask 255.255.255.0 gateway 20.1.2.2 channel “ch1” wan

PureFlow(A)> add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.2.2 channel “ch1” wan

PureFlow(A)> add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.2.2 channel “ch1” wan

＜シナリオ設定＞

PureFlow（A）> add scenario /port1/woc1-1 action wan_accel peer 20.1.8.10 second-peer 20.1.9.20

＜フィルタ設定＞

PureFlow（A）> add filter scenario /port1/woc1-1 filter “F1” ipv4 sip 40.8.1.10

WS1-1-2 の設定

以下のコマンドを実行します。

＜デフォルトチャネル設定＞

PureFlow（A）> add channel “ch10000” lan 1/1 wan 1/2 default

＜チャネル設定＞

PureFlow（A）> add channel “ch1” lan 1/1 wan 1/2 vid none

＜リンクダウン転送機能の設定＞

PureFlow(A)> add lpt pair port 1/1 1/2
PureFlow(A)> set lpt enable

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow（A）> set ip channel “ch1” 20.1.3.20 netmask 255.255.255.0

＜LAN 側のルート設定＞

PureFlow（A）> add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.3.1 channel “ch1” lan
PureFlow（A）> add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.3.1 channel “ch1” lan

＜WAN 側のルート設定＞

PureFlow（A）> add route target 20.1.8.0 netmask 255.255.255.0 gateway 20.1.3.2 channel “ch1” wan
PureFlow（A）> add route target 20.1.9.0 netmask 255.255.255.0 gateway 20.1.3.2 channel “ch1” wan
PureFlow（A）> add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.3.2 channel “ch1” wan
PureFlow（A）> add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.3.2 channel “ch1” wan

＜シナリオ設定＞

PureFlow（A）> add scenario /port1/woc1-2 action wan_accel peer 20.1.8.10 second-peer 20.1.9.20

＜フィルタ設定＞

PureFlow（A）> add filter scenario /port1/woc1-2 filter “F1” ipv4 sip 50.8.2.10

WS1-2-1 の設定

以下のコマンドを実行します。

＜デフォルトチャネル設定＞

PureFlow（A）> add channel “ch10000” lan 1/1 wan 1/2 default

＜チャネル設定＞

PureFlow（A）> add channel “ch1” lan 1/1 wan 1/2 vid none

＜リンクダウン転送機能の設定＞

PureFlow(A)> add lpt pair port 1/1 1/2
PureFlow(A)> set lpt enable

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow（A）> set ip channel “ch1” 20.1.8.10 netmask 255.255.255.0

＜LAN 側のルート設定＞

PureFlow（A）> add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.8.1 channel “ch1” lan
PureFlow（A）> add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.8.1 channel “ch1” lan

＜WAN 側のルート設定＞

PureFlow（A）> add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.8.2 channel “ch1” wan
PureFlow（A）> add route target 20.1.3.0 netmask 255.255.255.0 gateway 20.1.8.2 channel “ch1” wan
PureFlow（A）> add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.8.2 channel “ch1” wan
PureFlow（A）> add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.8.2 channel “ch1” wan

＜シナリオ設定＞

PureFlow（A）> add scenario /port1/woc2-1 action wan_accel peer 20.1.2.10 second-peer 20.1.3.20

＜フィルタ設定＞

PureFlow（A）> add filter scenario /port1/woc2-1 filter “F1” ipv4 sip 60.8.3.10

WS1-2-2 の設定

以下のコマンドを実行します。

＜デフォルトチャネル設定＞

PureFlow（A）> add channel “ch10000” lan 1/1 wan 1/2 default

＜チャネル設定＞

PureFlow（A）> add channel “ch1” lan 1/1 wan 1/2 vid none

＜リンクダウン転送機能の設定＞

PureFlow(A)> add lpt pair port 1/1 1/2
PureFlow(A)> set lpt enable

＜ネットワークインタフェースの IP アドレス設定＞

PureFlow（A）> set ip channel “ch1” 20.1.9.20 netmask 255.255.255.0

＜LAN 側のルート設定＞

PureFlow（A）> add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.9.1 channel “ch1” lan
PureFlow（A）> add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.9.1 channel “ch1” lan

＜WAN 側のルート設定＞

PureFlow（A）> add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.9.2 channel “ch1” wan
PureFlow（A）> add route target 20.1.3.0 netmask 255.255.255.0 gateway 20.1.9.2 channel “ch1” wan
PureFlow（A）> add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.9.2 channel “ch1” wan
PureFlow（A）> add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.9.2 channel “ch1” wan

＜シナリオ設定＞

PureFlow（A）> add scenario /port1/woc2-2 action wan_accel peer 20.1.2.10 second-peer 20.1.3.20

＜フィルタ設定＞

PureFlow（A）> add filter scenario /port1/woc2-2 filter “F1” ipv4 sip 70.8.4.10

8.14.7 TCP-FEC 機能

本装置は TCP-FEC 機能を備えています。本機能は，トラフィックアクセラレーションの TCP パケットに冗長データを付加し，FEC（Forward Error Correction）を行う機能です。TCP はパケット廃棄が発生すると再送を行いますが，TCP-FEC 機能を使用するとパケット廃棄時に廃棄したデータをリカバリするため，再送を必要としません。その結果，パケット廃棄が多い環境でも十分な性能を発揮することができます。
パケット廃棄発生時の動作を以下に示します。

図 8.14.7-1　TCP-FEC 機能

本機能を使用するために，データブロックサイズと FEC（冗長）ブロックサイズを設定する必要があります。
データブロックサイズと FEC ブロックサイズの関係を以下に示します。

図 8.14.7-2　データブロックサイズと FEC ブロックサイズの関係

注 1:

TCP-FEC 機能を使用する場合は，「TCP 高速化機能ライセンス(NF7500-L201A)」と「FEC 機能ライセンス(NF7500-L211A)」を購入していただく必要があります。

注2:

冗長比率(=FEC ブロックサイズ／データブロックサイズ)を大きくすることによりリカバリできる可能性は高くなりますが，回線効率は下がります。

本機能はシナリオごとに設定します。本機能を使用する場合は，シナリオ登録時またはシナリオ更新時にTCP-FEC 機能を有効にした上でパラメータを設定してください。
以下に，TCP-FEC 機能に関する CLI コマンドおよびパラメータを示します。詳細な設定方法やパラメータの確認方法は「8.9 設定方法」の「STEP4:シナリオの設定」を参照してください。

表 8.14.7-1　TCP-FEC 機能に関する CLI コマンドおよびパラメータ

コマンド	パラメータ	説明
add scenario update scenario	fec {enable | disable}	TCP-FEC 機能の有効／無効を指定します。
	block-size <size>	FEC ブロックサイズを指定します。データブロックサイズ以上の値は設定できません。また，データブロックサイズに対して割り切れる値を設定してください。
	data-block-size <size>	データブロックサイズを指定します。FEC ブロックサイズ以下の値は設定できません。また，FEC ブロックサイズに対して倍数になる値を設定してください。
	fec-session <session>	TCP-FEC 機能を使用した TCP セッションを FEC セッションと呼びます。本パラメータを設定するシナリオに使用できる FEC セッション数を制限します。 FEC セッションは装置全体で最大 1000 セッションです。なお，本パラメータで FEC セッション数が保証されるわけではありません。
show scenario	name <scenario_name>	指定したシナリオ名のシナリオ情報(TCP-FEC 機能に関するパラメータ)を表示します。

以下にコマンドの実行例を示します。

実行例①:TCP-FEC 機能付きのシナリオを追加する場合
パラメータ:FEC ブロックサイズ 4 kbyte，データブロックサイズ 8 kbyte，FEC セッション制限数 10

```
PureFlow(A)> add scenario /port1/woc1-fec action wan-accel peer 192.168.100.11 compression
             disable fec enable block-size 4k data-block-size 8k fec-session 10
```

実行例②:すでに登録されているシナリオを TCP-FEC 機能付きのシナリオにアップデートする場合
パラメータ:FEC ブロックサイズ 8 kbyte，データブロックサイズ 24 kbyte，FEC セッション制限数 100

```
PureFlow(A)> update scenario /port1/woc1 action wan-accel compression disable fec enable block-
             size 8k data-block-size 24k fec-session 100
```

実行例③:すでに登録されている TCP-FEC 機能付きのシナリオを通常のシナリオにアップデートする場合
パラメータ:TCP-FEC 機能無効

```
PureFlow(A)> update scenario /port1/woc1-fec action wan-accel compression enable fec disable
```

以下に TCP-FEC 機能の設定目安を示します。ただし，使用する回線環境によって詳細な調整を行ってください。

表 8.17.7-2　TCP-FEC 機能の設定目安

回線環境例	設定目安
TCP 通信のデータ量が小さい。	データブロックサイズを小さくしてください。
応答性を重視したい。	FEC ブロックサイズを小さくしてください。
WAN 回線のパケット廃棄率が高い。	データブロックサイズを小さく，FEC ブロックサイズを大きくしてください。
WAN 回線の回線効率を低下したくない。	FEC ブロックサイズに対してデータブロックサイズを大きくしてください。
WAN 回線のバースト廃棄が発生しやすい。	FEC ブロックサイズを大きくしてください。

8.14.8 TCP 輻輳制御機能

本装置は TCP 輻輳制御機能を備えています。本装置は，トラフィックアクセラレーションの TCP 輻輳制御を選択することが可能です。輻輳制御とは，ネットワークの輻輳を回避しつつ，通信容量を効率的に使用するため，TCP 通信の送信速度を調整する機能です。輻輳制御では，ネットワークの輻輳によってパケット損失が発生したとき送信レートを減少させ，パケット損失が発生しないときは送信速度を徐々に増加させます。これにより，ネットワークの通信容量をリアルタイムに追従するように，送信速度を調節します。しかし，従来の輻輳制御アルゴリズムには課題があり，パケット損失が多いネットワークでは，通信速度が回復するまえに，次のパケット損失によって，さらに通信速度を減少させ，その平均は通信容量の半分未満になる場合もあります。

本装置は，上記の課題を解決するため，独自の輻輳制御アルゴリズムを選択することが可能です。独自の輻輳制御アルゴリズムを使用するときは，シナリオの輻輳制御モード（cc-mode）を指定します。normal は，標準の輻輳制御アルゴリズムです。semi-fast および fast は，独自の輻輳制御アルゴリズムであり，パケット損失が発生したときの通信速度の低下を緩やかにします。その結果，パケット廃棄が多い環境でも十分な性能を発揮することができます。

semi-fast または fast を指定した場合，アクセラレーショントラフィックと非アクセラレーショントラフィックが同一回線上に混在すると，非アクセラレーショントラフィックのレートが低く抑えられる傾向になります。semi-fast または fast を使用する場合，シナリオの最大帯域（peak）を回線帯域以下に設定し，非アクセラレーショントラフィックが適切に転送されるように設定してください。

図 8.14.8-1　TCP 輻輳制御機能時の帯域設定

TCP 輻輳制御機能はシナリオごとに設定します。

表 8.14.8-1　TCP 輻輳制御機能に関する CLI コマンドおよびパラメータ

コマンド	パラメータ	説明
add scenario update scenario	cc-mode {normal ｜ semi-fast ｜ fast}	省略時：normal トラフィックアクセラレーションの輻輳制御モードを指定します。 wan-accel モードのみ有効
show scenario	name <scenario_name>	指定したシナリオ名のシナリオ情報（TCP 輻輳制御機能に関するパラメータ）を表示します。

以下にコマンドの実行例を示します。

実行例：輻輳制御モードを高速のシナリオを追加する場合
　パラメータ：輻輳制御モード fast

```
PureFlow(A)> add scenario /port1/woc1-fast action wan-accel peer 192.168.100.11
                    cc-mode fast  peak 7G
```

以下に TCP 輻輳制御機能の設定目安を示します。ただし，使用する回線環境によって詳細な調整を行ってください。

表 8.14.8-2　TCP 輻輳制御機能の設定目安

回線環境例	設定目安
WAN 回線のパケット廃棄がない。	輻輳制御モードを“normal”にしてください。
WAN 回線で少量のパケット廃棄が発生する。	輻輳制御モードを“semi-fast”にしてください。
WAN 回線のパケット廃棄が多い。	輻輳制御モードを“fast”にしてください。

注：

TCP 輻輳制御機能を使用する場合は，「TCP 高速化機能ライセンス(NF7500-L201A)」を購入していただく必要があります。

8.14.9 リマーキング機能

本装置は，IEEE802.1Q および QinQ の VLAN Tag フィールド内の“User Priority”(ユーザ優先度：CoS)と，IP ヘッダの“Type Of Service”(ToS)フィールド内の“DiffServ Code Point”(DSCP)をシナリオで指定した値に書き換える(リマーキング)機能を備えています。本装置で，CoS や DSCP を書き換えることで，WAN 回線内の優先制御サービスを適用可能となります。

図 8.14.9-1　リマーキング機能

本装置で CoS および DSCP 書き換え可能なシナリオモードを以下に示します。

表 8.14.9-1　CoS および DSCP 書き換え可能なシナリオモード

シナリオモード	CoS 書き換え	DSCP 書き換え
アクセラレーションモード(Wan-accel モード)	○	○
集約キューモード(Aggregate モード)	○	○
個別キューモード(Individual モード)	○	○
廃棄モード(Discard モード)	×	×
転送モード(Forward モード)	×	×

注：

アクセラレーションモードの場合，自装置と対向装置とアクセラレーショントンネルを構成し，トラフィックアクセラレーションを行います。アクセラレーションモードで，CoS や DSCP を書き換える場合，自装置と対向装置のシナリオ設定を同じ値にしてください。クライアントとサーバ間の TCP 通信において，WAN 側と LAN 側に送信する TCP パケットの CoS と DSCP が同じ値になります。

［Case 1］アクセラレーションモードシナリオで CoS と DSCP を設定しない場合

クライアントとサーバ間のTCPパケットのCoSとDSCPを書き換えません。

図 8.14.9-2　Case1 の設定および動作例

［Case 2］自装置と対向装置で同じ CoS と DSCP を設定した場合

クライアントとサーバ間のTCPパケットのCoSとDSCPを書き換えます。

図 8.14.9-3　Case2 の設定および動作例

［Case 3］自装置と対向装置で異なる CoS と DSCP を設定する場合

クライアントとサーバ間のTCPパケットのCoSとDSCPを書き換えます。

図 8.14.9-4　Case3 の設定および動作例

本機能では，本装置が転送する Ethernet フレームの VLAN Tag 内の上位 3 ビットであるユーザ優先度（CoS）を書き換えることができます。また，IP ヘッダ内の ToS フィールドの上位 6 ビットである DSCP を書き換えることができます。

以下に，フレームフォーマットを示します。

VLAN TagのEthernetフレームフォーマット

2重VLAN TagのEthernetフレームフォーマット

VLAN Tagのヘッダフォーマット（ユーザ優先度：CoS）

図 8.14.9-5　VLAN Tag フレームフォーマット

IPv4のヘッダフォーマット（DSCP）

IPv6のヘッダフォーマット（DSCP）

図 8.14.9-6　IPv4／IPv6 ヘッダフォーマット

本機能は，シナリオごとに設定します。本機能を使用する場合は，シナリオ登録時またはシナリオ更新時にパラメータを設定してください。シナリオ更新は，aggregate モードと individual モードのみ可能です。
以下に，リマーキング機能に関するパラメータを示します。詳細な設定方法やパラメータの確認方法は「8.9 設定方法」の「STEP4：シナリオの設定」を参照してください。

表 8.14.9-2　リマーキング機能に関する CLI コマンドおよびパラメータ

コマンド	パラメータ	説明
add scenario update scenario	cos {through ｜ <user_priority>}	VLAN Tag ありフレームの CoS 書き換え値を指定します。
	inner-cos {through ｜ <user_priority>}	2 重 VLAN Tag ありフレームの CoS 書き換え値を指定します。
	dscp {through ｜ <dscp>}	DSCP 書き換え値を指定します。
show scenario	name <scenario_name>	指定したシナリオ名のシナリオ情報（リマーキング機能に関するパラメータ）を表示します。

以下にコマンドの実行例を示します。

実行例①:シナリオで DSCP を指定する場合(CoS は未指定)

パラメータ:DSCP 5

```
PureFlow(A)> add scenario /port1/woc1 action wan-accel peer 192.168.100.11 dscp 5
PureFlow(A)> add scenario /port1/agg1 action aggregate dscp 5
```

実行例②:すでに登録されているアクセラレーションモード以外のシナリオに CoS と Inner-CoS をアップデートする場合

パラメータ:CoS 3, Inner-CoS 4

```
PureFlow(A)> update scenario /port1/agg1 action aggregate cos 3 inner-cos 4
```

実行例③:すでに登録されているアクセラレーションモード以外のシナリオを CoS と DSCP を書き換えずにアップデートする場合

パラメータ:CoS, DSCP 書き換え無効

```
PureFlow(A)> update scenario /port1/agg1 action aggregate cos through dscp through
```

8.15 トラフィックアクセラレーション実行時のアドレス

トラフィックアクセラレーションは，本装置のチャネルインタフェースに設定された IP アドレスおよび MAC アドレスを使用し動作します。以下に，トラフィックアクセラレーション実行時の IP アドレスおよび MAC アドレスの関係を示します。

[Case 1]同一サブネットでトラフィックアクセラレーションを実行する場合

図 8.15-1　Case1 の IP アドレスおよび MAC アドレス

ARP と NDP は，Client-Server 間で送受信し，本装置はそれらを透過します。

トラフィックアクセラレーションを適用した TCP セッションの送信元 MAC アドレスは，本装置の MAC アドレスに置き換わります。ここで使用される MAC アドレスは，“show module”コマンドで表示される“Channel MAC Address”となります。

[Case 2]ルータ経由でトラフィックアクセラレーションを実行する場合

図 8.15-2　Case2 の IP アドレスおよび MAC アドレス

ARP と NDP は，ホスト-Router 間で送受信し，本装置はそれらを透過します。

トラフィックアクセラレーションを適用した TCP セッションの送信元 MAC アドレスは，本装置の MAC アドレスに置き換わります。ここで使用される MAC アドレスは，“show module”コマンドで表示される“Channel MAC Address”となります。

第9章　リンクダウン転送機能

ここでは，リンクダウン転送機能について説明します。

9.1　リンクダウン転送機能 9-2

9.1　リンクダウン転送機能

本装置のリンクダウン転送機能を使用すると，「IEEE802.3ad Link Aggregation」などの回線冗長機能を使用している装置の間に本装置を挿入しても外部装置間の回線冗長機能を妨げることなく協調動作を行います。

本装置では，リンクダウンを検出すると対向のリンクをダウンさせることにより対向装置に対して警報の転送を行います。対向の装置は，そのリンクダウンを検出することにより回線を切り替えることが可能となります。

図 9.1-1　リンクダウン転送機能

リンクダウン転送機能の設定を以下に示します。

表 9.1-1　リンクダウン転送機能の設定

add lpt pair port <slot/port> <slot/port>	リンクダウン転送機能の Network ポートの組み合わせを登録します。
delete lpt pair port <slot/port> <slot/port>	リンクダウン転送機能の Network ポートの組み合わせを削除します。
set lpt {enable | disable}	リンクダウン転送機能の有効／無効を設定します。
show lpt	リンクダウン転送機能に関する情報を表示します。

コマンドの実行例を示します。

```
PureFlow(A)> add lpt pair port 1/1 1/2
PureFlow(A)> add lpt pair port 1/3 1/4
PureFlow(A)> set lpt enable
PureFlow(A)>
```

（注 1）
Network ポートの組み合わせを登録または削除するときは，リンクダウン転送機能が無効のときに行ってください。

（注 2）
登録した Network ポートは，重複して別の組み合わせで登録することはできません。

（注 3）
以下の場合は，リンクダウン転送機能有効時でもケーブルを接続しているポートが一時的にリンクアップし通信できる状態となり，約 10 秒後にリンクダウンします。

・対となるポートの両方にケーブルを接続せずに本装置を起動し，片方のポートのみにケーブルを接続した場合

・対となる両方のポートがリンクアップしている状態から片方のケーブルを抜き、次にもう片方のケーブルを抜いてから先に抜いたケーブルを接続した場合

・初期設定（リンクダウン転送機能無効）で片側ポートのみケーブルを接続して本装置を起動し，リンクダウン転送機能を有効とした場合

（空白ページ）

第10章　SSH 機能

ここでは，SSH（Secure SHell）機能について説明します。

10.1　概要 10-2
10.2　仕様一覧 10-3
10.3　SSH の利用方法 10-4
　10.3.1 本体の設定 10-4
　10.3.2 SSH クライアントの準備 10-4
　10.3.3 注意事項 10-5

10.1 概要

本装置は，SSH バージョン 2 に準拠した SSH サーバ機能を提供します。SSH サーバ機能により，本装置と SSH クライアント間の通信が暗号化され，安全性が保証されていないネットワークを経由する場合でも，セキュアな遠隔操作が可能になります。また，強力なサーバ認証機能を有し，第 3 者による「盗聴」や「なりすまし」を防止することができます。

SSH サーバによる接続を利用する場合も，不特定多数の端末から本装置への通信を制限するためのシステムインタフェースフィルタを設定することができます。詳細は，「第 7 章 システムインタフェースの設定」を参照してください。また，Telnet と同様に，ローカルに設定された root ユーザのパスワード認証だけでなく，RADIUS サーバ経由でのパスワード認証が利用できます。RADIUS 機能の詳細は，「第 13 章 RADIUS 機能」を参照してください。

図 10.1-1　SSH 機能

10.2 仕様一覧

本装置の SSH サーバ機能の仕様一覧を記載します。

表 10.2-1　仕様一覧

項目	内容
SSH バージョン	SSH Ver.2 準拠
ユーザ認証方式	パスワード認証
鍵交換アルゴリズム	ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group-exchange-sha1, diffie-hellman-group14-sha1, diffie-hellman-group1-sha1
公開鍵アルゴリズム	RSA 2048bit, DSA 1024bit, ECDSA 256bit
暗号化アルゴリズム	aes128-ctr, aes192-ctr, aes256-ctr, arcfour256, arcfour128, aes128-gcm@openssh.com, aes256-gcm@openssh.com, aes128-cbc, blowfish-cbc, cast128-cbc, aes192-cbc, aes256-cbc, arcfour, rijndael-cbc@lysator.liu.se
MAC アルゴリズム	hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, umac-64-etm@openssh.com, umac-128-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-ripemd160-etm@openssh.com, hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, umac-64@openssh.com, umac-128@openssh.com, hmac-sha2-256, hmac-sha2-512, hmac-ripemd160, hmac-ripemd160@openssh.com, hmac-sha1-96, hmac-md5-96
接続ポート番号	22
クライアント最大接続数	8（telnet 接続数と合わせて）

10.3 SSH の利用方法

10.3.1 本体の設定

本装置の SSH サーバ機能を使用するには，以下の設定が必要です。

(1) システムインタフェースの設定

本装置の IP アドレスや Gateway を設定します。接続する端末を制限する場合は，システムインタフェースフィルタを設定します。詳細は，「第 7 章 システムインタフェースの設定」を参照してください。

(2) 公開鍵（ホスト鍵）の生成

SSH サーバは，SSH クライアントとの接続を確立するために，ホスト鍵（RSA 認証鍵または DSA 認証鍵）を必要とします。このホスト鍵は，工場出荷時に無作為に生成された鍵が設定されており，装置外部からは参照できない状態で装置内部に保存しています。特に，新しく生成する必要はありませんが，必要に応じてシリアルコンソールから変更することができます。

10.3.2 SSH クライアントの準備

SSH バージョン 2 に準拠した SSH クライアントを用意してください。

10.3.3 注意事項

(1) 初めて SSH 接続を行うときの注意事項

SSH クライアントからリモートホストに初めて接続するとき，そのホストを信用していいかどうかを確認するサーバ認証を行います。このとき，SSH クライアントは，リモートホストが通知してきた認証鍵の fingerprint を表示し，このホストに接続していいかの確認を求めます。この場合は，SSH クライアントが表示したリモートホストの fingerprint と本装置の fingerprint が一致しているかどうかを確認することを推奨します。

本装置のホスト鍵の fingerprint は，“show ssh”コマンドで表示可能です。

(2) ホスト鍵の再生成

本装置の SSH サーバが使用するホスト鍵は，工場出荷時に生成され本装置内部に保存されています。このホスト鍵は，“set ssh server key”コマンドで変更することが可能ですが，このコマンドは，シリアルコンソールからログインしたときだけ実行可能です。

(3) ホスト鍵を再生成したあとの SSH 接続

SSH クライアントは，過去に接続したリモートホストの fingerprint を記憶しており，過去に通知してきた fingerprint が異なる場合，SSH クライアントは，ワーニングを表示し，リモートホストへの SSH 接続を切断します。これは，リモートホストの「なりすまし」を防止するための動作であり，多くの SSH クライアントが同様な動作をします。

本装置のホスト鍵を再生成した場合は，本装置に SSH で接続したことがある SSH クライアントから，本装置の fingerprint を削除または更新する必要があります。詳細は，SSH クライアントのマニュアルを参照してください。

(4) RADIUS 機能を有効にした場合の SSH 接続

本装置の RADIUS 機能を有効にした場合，本装置は，ログイン認証時に RADIUS サーバに問い合わせます。SSH クライアントから本装置に新しい SSH セッションの接続を試みた場合，SSH クライアントと本装置の通信は SSH 機能により暗号化されますが，RADIUS サーバと本装置の通信は暗号化されません。RADIUS サーバとの通信を傍受された場合，パスワードは RADIUS プロトコルにより秘匿されますが，ログイン名が第三者によって解読される可能性があります。

（空白ページ）

第11章　SNMP の設定

ここでは，SNMP の機能と設定について説明します。

11.1　SNMP の概要 11-2
11.2　SNMPv1/SNMPv2c の設定 11-3
11.3　SNMPv3 の設定 11-5
11.4　TRAP の設定 11-7

11.1 SNMP の概要

SNMP は，ルータやサーバなどのネットワーク機器に対してネットワークを通してリモートで管理するためのプロトコルです。SNMP では，ルータやサーバなどの管理される側をエージェントノード（またはエージェント），管理用のアプリケーションソフトウェアをインストールした PC や EWS をマネジメントノード（またはマネージャ）と呼んでいます。ネットワーク管理者はマネジメントノードのコンソールを使って，ネットワーク機器（エージェントノード）の障害を発見したり，設定を変更することで，日々のネットワーク管理業務を遂行します。

図 11.1-1　SNMP 機能

SNMP には SNMPv1，SNMPv2c，SNMPv3 の 3 つのバージョンが存在します。
本装置は SNMPv1，SNMPv2c，SNMPv3 の 3 つのバージョンをすべてサポートしています。それぞれのバージョンによる違いは以下のとおりです。

・ SNMPv1：最もシンプルで簡単なプロトコルで，管理情報の取得，設定，トラップ（警報）の 3 つのオペレーションから成り立っています。セキュリティはコミュニティ名と呼ばれる文字列（パスワードのようなもの）で実現されています。コミュニティ名は SNMPv1 データ要求とともにパケットに含まれてしまうため，ネットワークを測定器などでモニタされると盗み見されてしまいます。コミュニティ名は暗号化されないため，安全とみなすことはできません。外部の人間がネットワークに接続しないイントラネットなどでしか用いることができません。

・ SNMPv2c：管理情報の取得に，バルク転送と呼ばれるデータの一括取得処理をサポートすることで，プロトコルのオーバヘッドを軽減しました。アクセス・セキュリティは SNMPv1 と同様にコミュニティ文字列で行うため，セキュリティ強度は SNMPv1 と同等です。

・ SNMPv3：最新のプロトコルで，ユーザ名とそれに対応した暗号化パスワードでアクセスを認証します。エージェントへのアクセスはユーザ名が必要です。ユーザ名はグループという単位でまとめられ，グループごとに管理情報の取得，設定の権限の範囲を変えておくことで，コーポレートごとの管理者グループ，部門管理者グループ，一般ユーザグループという具合いに，権限を階層構造にもたせることができます。大規模イントラネットからインターネットまで，一般的な用途で利用可能です。SNMPv3 のセキュリティは暗号化機能も持ちますが，本装置では暗号化機能をサポートしていません。

一般的なマネジメントソフトウェアは，エージェントがサポートできるバージョンを自動検知し，最も高いバージョンを優先使用します。
※ OpenFlow 機能を使用する場合は，SNMP マネージャのタイムアウトを 2 秒以上に設定してください。

11.2 SNMPv1/SNMPv2c の設定

SNMPv1 および SNMPv2c はどちらもコミュニティ名と呼ばれる文字列（パスワードのようなもの）を設定することでマネジメントノードからのアクセスが可能となります。

表 11.2-1　SNMPv1/SNMPv2c の設定

add snmp community <community_string> [version {v1 | v2c}] [view <view_name>] [permission {ro | rw}]	SNMPv1/v2c のコミュニティを追加します。
delete snmp community <community_string>	コミュニティを削除します。
add snmp view <view_name> <oid> {included | excluded}	SNMP の View（管理範囲の制限）を設定します。 注）snmpv2 グループは，本コマンドで指定可能ですが SNMP によるアクセスはできません。
delete snmp view <view_name> [<oid>]	SNMP の View（管理範囲の制限）を削除します。
show snmp community [<community_string>]	設定されているコミュニティを表示します。
show snmp view [<view_name>]	設定されている View を表示します。

最初に SNMPv1 コミュニティに“netman1”，SNMPv2c コミュニティに“netman2”というコミュニティ名を設定します。

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community netman1 version v1 permission rw
PureFlow(A)> add snmp community netman2 version v2c permission rw
```

View はそのコミュニティ名でアクセスするマネジメントノードが本装置のどの MIB Tree に対してアクセス可能かを許可／制限する機構です。add snmp community で view を省略時は“All”の View 名に対してアクセスが可能となります。また，v2c のトラップ送信を使用する場合，<oid>パラメータに，“private”を指定する際は“system”と“snmpmodules”の“included”設定も追加してください。

SNMPv1 コミュニティ netman1 を interfaces グループだけにアクセス制限をかけるには，以下のコマンドを実行します。

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp view myview1 interface included
PureFlow(A)> add snmp community netman1 version v1 view myview1 permission rw
```

設定コマンドで設定した community 名や view の内容を確認するには，show snmp community コマンドと show snmp view コマンドを使用します。

```
PureFlow> show snmp community
-----------------------------------------------------------------
Community Name                    :netman1
Version                           :v1
Read View                         :myview1
Write View                        :myview1
-----------------------------------------------------------------
Community Name                    :netman2
Version                           :v2c
Read View                         :All
Write View                        :All
-----------------------------------------------------------------
PureFlow>
PureFlow> show snmp view
-------------------------------------------------------------------------------
View name                         :All
Subtree                           :iso
Access State                      :included
-------------------------------------------------------------------------------
View name                         :myview1
Subtree                           :interface
Access State                      :Included
PureFlow>
```

11.3 SNMPv3 の設定

SNMPv3 の管理フレームワークは，ユーザごとにセキュリティを設定するユーザベースセキュリティです。各ユーザはグループに属し，グループの属性として View を設定します。

図 11.3-1　SNMPv3 機能

SNMPv3 を使用するためには，グループ，ユーザ，View の設定が必要です。以下のコマンドを使用します。

表 11.3-1　SNMPv3 の設定

add snmp group <group_name> [auth_type {auth | noauth}] [read <readview>] [write <writeview>] [notify <notifyview>]	SNMPv3 のグループを追加します。
delete snmp group <group_name>	グループを削除します。
add snmp user <user_name> <group_name> [auth_type {auth | noauth}] [password <auth_password>]	SNMPv3 のユーザを追加します。パスワードを指定する場合，8 文字以上 24 文字以下で指定してください。
delete snmp user <user_name>	ユーザを削除します。
add snmp view <view_name> <oid> {included | excluded}	SNMP の View（管理範囲の制限）を設定します。 注）snmpv2 グループは，本コマンドで指定可能ですが SNMP によるアクセスはできません。
delete snmp view <view_name> [<oid>]	SNMP の View（管理範囲の制限）を削除します。
show snmp group [<group_name>]	設定されているグループを表示します。
show snmp user [<user_name>]	設定されているユーザを表示します。
show snmp view [<view_name>]	設定されている View を表示します。

View はそのグループ，ユーザ名でアクセスするマネジメントノードが本装置のどの MIB Tree に対してアクセス可能かを許可／制限する機構です。add snmp group で view を省略時は“All”の View 名に対してアクセスが可能となります。また，v3 のトラップ送信を使用する場合，<oid>パラメータに，“private”を指定する際は “system”と“snmpmodules”の“included”設定も追加してください。

以下のコマンド例は SNMPv3 ユーザ Mike と Nancy をグループ netman3 の一員として設定します。

```
PureFlow(A)> add snmp view myview3 iso included
PureFlow(A)> add snmp group netman3 auth_type auth read  myview3 write  myview3
notify myview3
PureFlow(A)> add snmp user Mike netman3 auth_type auth password T5ega8GH
PureFlow(A)> add snmp user Nancy netman3 auth_type auth password R64dWa99
```

11.4 TRAP の設定

SNMP ではエージェントノードの状態変化を検出して，マネジメントノードへ通知する機能があります。通知用の View とマネジメントノード（ホスト）のアドレスを設定することでマネジメントノードへの TRAP（ノーティフィケーション）の送信が可能となります。

表 11.4-1　TRAP の設定

add snmp view <view_name> <oid> {included | excluded}	SNMP の View（管理範囲の制限）を設定します。
add snmp host <host_address> version {v1 | v2c | v3 [auth_type { auth | noauth}] } {user | community} <community_string / username> } {trap | inform} [udp_port <port_number>] [<notification_type>]	SNMP TRAP（ノーティフィケーション）の送信先を示すホストを追加します。
delete snmp host <host_address>	TRAP の送信先を示すホストを削除します。
set snmp traps {authentication | linkup | linkdown | coldstart | modulefailurealarm | modulefailurerecovery | systemheatalarm | systemheatrecovery | powerinsert | powerextract | powerfailure | powerrecovery | faninsert | fanextract | fanfailure | fanrecovery | queuebuffalarm | queuebuffrecovery | systembuffalarm | systembuffrecovery | queueallocalarm | queueallocrecovery | maxqnumalarm | maxqnumrecovery tcpbypassalarm | tcpbypassrecovery | peeralarm | peerrecovery | bypasson | bypassoff} {enable | disable}	SNMP の TRAP 送信を有効／無効に設定します。トラップ種別ごとに設定することができます。
show snmp host [<host_address>]	TRAP の送信先を示すホストの一覧を表示します。

最初に SNMP TRAP 送信用に View を設定します。SNMP 基本 TRAP は snmpv2 オブジェクト，Enterprise TRAP は private オブジェクトに含まれています。snmpv2 オブジェクト，private オブジェクトへのアクセスを有効にすることで TRAP をマネジメントノードの送信することが可能となります。

PureFlow(A)> add snmp view All iso included

TRAP 種別とアクセス許可設定が必要な MIB Tree の OID は以下のとおりです。

coldStart　　：snmpmodules，system
linkUp／linkDown　：snmpmodules，system，interface
Enterprise　　：snmpmodules，system，private

この例の OID 名 iso は，必要な MIB Tree をすべて含んでいるので，全種別の TRAP を送信できます。

TRAP 送信先を設定します。

PureFlow(A)> add snmp host 192.168.1.10 version v1 community public trap udp_port 162

authenticationFailure TRAP の送信を無効にするには下記を設定します。

```
PureFlow(A)> set snmp traps authentication disable
```

設定コマンドで設定したホストの内容を確認するには，show snmp host コマンドを使用します。

```
PureFlow(A)> show snmp host
-----------------------------------------------------
Host Address                :192.168.1.10
Version                     :v1
Security                    :No Authentication
Security Name               :public
UDP port                    :162
Notification Type           :all
-----------------------------------------------------
Host Address                :192.168.1.11
Version                     :v2c
Security                    :No Authentication
Security Name               :public
UDP port                    :162
Notification Type           :all
PureFlow(A)>
```

設定コマンドで設定した TRAP の有効／無効の内容を確認するには，show snmp system コマンドを使用します。

```
PureFlow(A)> show snmp system
-------------------------------------------------------------------------------
System Location                 :Not Yet Set
System Contact                  :Not Yet Set
System Name                     :Not Yet Set
Engine ID                       :00:00:04:7f:00:00:00:a1:c0:a8:01:01

Traps
  authentication                :disable
  linkup                        :enable
  linkdown                      :enable
  coldstart                     :enable
  modulefailurealarm            :enable
  modulefailurerecovery         :enable
  systemheatalarm               :enable
  systemheatrecovery            :enable
  powerinsert                   :enable
  powerextract                  :enable
  powerfailure                  :enable
  powerrecovery                 :enable
  faninsert                     :enable
  fanextract                    :enable
  fanfailure                    :enable
  fanrecovery                   :enable
  queuebuffalarm                :enable
  queuebuffrecovery             :enable
  systembuffalarm               :enable
  systembuffrecovery            :enable
  queueallocalarm               :enable
  queueallocrecovery            :enable
  maxqnumalarm                  :enable
  maxqnumrecovery               :enable
  tcpbypassalarm                : enable
  tcpbypassrecovery             : enable
  peeralarm                     : enable
  peerrecovery                  : enable
  bypasson                      : enable
  bypassoff                     : enable
-------------------------------------------------------------------------------
PureFlow(A)>
```

（空白ページ）

第12章　統計情報

ここでは，統計情報について説明します。
本装置には，ポート統計情報，シナリオ統計情報があります。

12.1　ポート統計情報 12-2
　12.1.1 ポートカウンタ 12-2
12.2　シナリオ統計情報 12-3
　12.2.1 シナリオカウンタ 12-3
　12.2.2 シナリオ動作情報 12-5
　12.2.3 レート測定 12-6
　12.2.4 シナリオパラメータ決定方法 12-7

12.1 ポート統計情報

ポート統計情報には，Network ポートカウンタおよびシステムインタフェースカウンタがあります。
この情報は，Network ポートごと，およびシステムインタフェースの統計情報です。

12.1.1 ポートカウンタ

Network ポートごと，およびシステムインタフェースのカウンタです。
ポートカウンタでは，以下の内容を表示します。

- 受信バイト数
- 受信パケット数
- 受信ブロードキャストパケット数
- 受信マルチキャストパケット数
- 送信バイト数
- 送信パケット数
- 送信ブロードキャストパケット数
- 送信マルチキャストパケット数
- 受信エラーパケット数
- Collision（パケットの衝突）発生回数
- 廃棄パケット数
- 受信したパケットの平均レート（単位 kbit/s）
- 送信したパケットの平均レート（単位 kbit/s）

システムインタフェースカウンタでは，以下の内容を表示します。

- 受信バイト数
- 受信パケット数
- 送信バイト数
- 送信パケット数

ポートカウンタに関する CLI は以下のコマンドがあります。

表 12.1.1-1　ポートカウンタに関する CLI

show counter [brief]	すべての Network ポートおよびシステムインタフェースのカウンタを表示します。brief を指定した場合は，概要を表示します。
show counter {<slot/port> | system}	指定 Network ポートまたはシステムインタフェースのカウンタを表示します。
clear counter [<slot/port> | system]	指定 Network ポートまたはシステムインタフェースのカウンタをクリアします。

注：

Network ポート 1/1 と 1/2 のメディアタイプ（RJ-45 または SFP）を変更した場合，当該ポートの統計情報はクリアしません。“clear counter”コマンドで統計情報をクリアすることができます。

12.2 シナリオ統計情報

シナリオ統計情報には，シナリオカウンタ，シナリオ動作情報，レート測定があります。
この情報は，シナリオごとの統計情報です。

12.2.1 シナリオカウンタ

シナリオごとのカウンタです。
シナリオカウンタでは，以下の内容を表示します。

- 受信バイト数，受信パケット数
- 送信バイト数，送信パケット数
- 廃棄バイト数，廃棄パケット数

シナリオカウンタは，関連する下位レベルのシナリオカウンタを含めた合計値となります。

図 12.2.1-1　各レベルでのシナリオカウンタ合計値

シナリオカウンタに関する CLI は以下のコマンドがあります。

表 12.2.1-1　シナリオカウンタに関する CLI

show scenario counter name <scenario_name>	シナリオのカウンタを表示します。
show scenario counter summary	シナリオのカウンタを一覧で表示します。
clear scenario counter name <scenario_name>	シナリオのカウンタをクリアします。
clear scenario counter all	すべてのシナリオのカウンタをクリアします。

<scenario_name>は，“add scenario”コマンドで指定したシナリオ名を指定します。

12.2.2 シナリオ動作情報

シナリオごとの動作情報です。
シナリオ動作情報では，以下の内容を表示します。

＜シナリオのデフォルトキューに関する情報＞
- バッファ使用量とバッファ使用率
- バッファピークホールド（バッファ使用最大値）
- フロー数（forward シナリオの場合は総フロー数と同じ値を表示します）

＜シナリオの個別キューに関する情報（個別キューモードシナリオのみ）＞
- 個別キュー数
- バッファ使用量とバッファ使用率（現在のバッファ使用量の最大／最小／平均値も表示）
- バッファピークホールド（バッファ使用最大値）
- 今までに割り当てた個別キューの中で，バッファ使用最大値が最大の個別キュー

＜シナリオの送信レートに関する情報＞
- 送信ピークレート（直近 1 分間の最大送信レート）
- 送信平均レート（直近 1 分間の平均送信レート）
- シナリオに関連する総フロー数

図 12.2.2-1　シナリオキューに関する情報

＜トラフィックアクセラレーションに関する情報＞
- トラフィックアクセラレーションを適用している TCP セッション数
- アクセラレーショントンネルを構成している対向装置（PRIMARY／SECONDARY）
- トラフィックアクセラレーションのバイパス機能に関する設定内容および動作状態
  （バイパス機能の詳細は，「8.14.5 トラフィックアクセラレーションバイパス」を参照してください。）

シナリオ動作情報に関する CLI は以下のコマンドがあります。

表 12.2.2-1　シナリオ動作情報に関する CLI

show scenario info name <scenario_name>	シナリオに関する動作情報を表示します。
show scenario info summary	シナリオに関する動作情報を一覧で表示します。
clear scenario peakhold buffer name <scenario_name>	シナリオに関するバッファ使用最大値をクリアします。
clear scenario peakhold buffer all	すべてのシナリオのバッファ使用最大値をクリアします。

<scenario_name>は，“add scenario”コマンドで指定したシナリオ名を指定します。

12.2.3 レート測定

シナリオの受信／送信レートを測定します。受信／送信レートは，約 1 秒ごとに測定を行い，指定回数分表示します。

表示単位は kbit/s で，小数点以下 3 桁まで表示します。また，受信／送信レートの測定は，パケットのみを対象とし，フレーム間ギャップとプリアンブルを含みません。

図 12.2.3-1　レート測定

レート測定に関する CLI は以下のコマンドがあります。

表 12.2.3-1　レート測定に関する CLI

monitor rate <scenario_name> [{queue <QID> | default_queue}] [<num>]	シナリオの受信／送信レートを測定します。 “queue”は個別キューモードのシナリオに対してのみ指定できます。QID を指定すると指定個別キューを測定し，QID を省略するとすべての個別キューおよび failaction キューの総計を測定します。 “default_queue”を指定した場合，指定シナリオのデフォルトキューの受信／送信レートを測定します。

コマンドの実行例を示します。

```
PureFlow(A)> monitor rate /port1/Tokyo 3
Scenario Name : “/port1/Tokyo”
QID : -----

  Times[s]          Rcv Rate[kbps]          Trs Rate[kbps]
  ---------------------------------------------------------
        1              3587.562                1254.531
        2              3482.826                1198.426
        3              3624.692                1217.879
  ---------------------------------------------------------
  Average              3565.026                1223.612
PureFlow(A)>
```

注）CLI 中の“bps”は bit/s を表します。

12.2.4 シナリオパラメータ決定方法

シナリオ統計情報を用いることで，シナリオの平均レート，バーストサイズを測定し，パラメータ決定の参考にすることができます。以下に，決定方法を説明します。

STEP1　レート測定機能を用いた平均レートの測定方法

レート測定するためには，シナリオを割り当てる必要があります。測定対象フローに対し，シナリオとフィルタを設定します。

図 12.2.4-1　平均レートの測定方法

まず，測定用のシナリオをレベル 2 にバッファサイズ 100 Mbyte（設定可能最大値）で設定します。測定用シナリオに，測定対象フローのみがヒットするフィルタを設定します。

設定例：

```
PureFlow(A)> add scenario /port1/measscenario action aggregate bufsize 100M
PureFlow(A)> add filter scenario /port1/measscenario filter measflow ipv4 sip 192.168.10.9
```

実際にフローを流し，測定シナリオに対してレート測定を実行します。

```
PureFlow(A)> monitor rate /port1/measscenario 3
Scenario Name : "/port1/measscenario"
QID : -----

  Times[s]          Rcv Rate[kbps]          Trs Rate[kbps]
-------------------------------------------------------------
     1                3587.562                3587.562
     2                3482.826                3482.826
     3                3624.692                3624.692
-------------------------------------------------------------
  Average             3565.026                3565.026
PureFlow(A)>
```

注）CLI 中の"bps"は bit/s を表します。

レート測定の結果，平均受信レートが約 3.6 Mbit/s であることが分かります。

STEP2 バッファピークホールドを用いたバッファ使用最大値の測定方法

次にバッファサイズを決定するためにバーストサイズの測定を行います。STEP1 の測定により得られた平均受信レートに 10%程度のマージンを加えたレートをトラフィックアトリビュートに再設定します。
下記の例では，4 Mbit/s のレートをトラフィックアトリビュートに再設定しています。

```
PureFlow(A)> update scenario /port1/measscenario action aggregate peak_bw 4M
```

次にフローを流している状態で，バッファ使用最大値をクリアします。

```
PureFlow(A)> clear scenario peakhold buffer name /port1/measscenario
```

この状態で，バッファ使用最大値の再記録が行われます。通常の映像トラフィックであれば 1 分程度で映像のバーストサイズがバッファ使用最大値として記録されます。
記録されたバッファ使用最大値を以下のように表示させます。

```
PureFlow(A)> show scenario info name /port1/measscenario
Scenario 1:  "/port1/measscenario"
  Rate Control Unit:
            Create Mode          :Aggregate
            Class                :2
            Min Bandwidth        :-----
            Peak Bandwidth       :4M[bps]
  Default Queue:
            Class                :8
            Buf Size             :100M[Bytes]

  Attached Filters:
            "measflow"

Scenario Rate Information
  Recent interval Tx peak        :0[bps]
  Recent interval Tx average     :0[bps]

Default Queue Information
  Buffer Utilization
            Current              :105384( 10%)[Bytes(%)]
            Peak Hold            :149504( 14%)[Bytes(%)]
  Related Flow
            Flow Num             :1[flows]
PureFlow(A)>
```

バッファ使用最大値が 149504 バイトであることが分かります。測定により得られたバッファ使用最大値に安全率 2 を与え，bufsize を 300000 バイトとします。

```
PureFlow(A)> update scenario /port1/measscenario action aggregate bufsize 300000
```

以上で，対象フローへのトラフィックアトリビュートは下記の値となります。

PeakBandwidth：　4 Mbit/s
BufSize　：　300000 bytes

注：
安全率は，ネットワーク環境やトラフィックにより適性値を与えてください。

第13章　RADIUS 機能

ここでは，RADIUS（Remote Authentication Dial In User Service）機能について説明します。

13.1　概要 13-2
13.2　ログイン認証の制御 13-3
13.3　ログインモードの制御..... 13-3
13.4　RADIUS 機能の設定..... 13-4
13.5　RADIUS サーバの設定 13-6

13.1 概要

RADIUS 機能は，TELNET，SSH，およびシリアルコンソールのログイン時に，RADIUS（RFC2865）を使用してユーザ認証する機能です。本装置は，RADIUS クライアントとして動作し，外部に設置した RADIUS サーバのユーザ情報に基づいたユーザ認証が可能です。

図 13.1-1　RADIUS 機能

① ユーザが管理者端末からユーザ名とパスワードを入力する。

② 本装置の RADIUS クライアントから RADIUS サーバに認証要求を送信する。

③ RADIUS サーバから RADIUS クライアントに認証応答を送信する。

④ 本装置は受信した認証応答に基づいて管理者端末からの接続を許可する。

13.2 ログイン認証の制御

RADIUS 機能を有効にした場合のログイン認証の制御について説明します。RADIUS 機能が有効な場合と無効な場合におけるログイン認証の制御は以下のとおりです。

表 13.2-1　ログイン認証の制御

RADIUS 認証有効時の ログイン認証手順	RADIUIS 認証無効時の ログイン認証手順
1）　本装置に設定されたユーザ名とログインパスワードでログイン認証を実施します。 2）　ログイン認証が拒否された場合，RADIUS サーバに登録されたユーザ名とログインパスワードでログイン認証を実施します。	1）　本装置に設定されたユーザ名とログインパスワードでログイン認証を実施します。

13.3 ログインモードの制御

本装置は，RADIUS サーバに設定されるユーザごとのサービスタイプに従って，ユーザがログインしたときのログインモードを切り替えます。本装置がサポートするサービスタイプは以下のとおりです。

表 13.3-1　サポートするサービスタイプ

サービスタイプ	ログインモード
Login-User(1)	normal モード
Administrative-User(6)	administrator モード

なお，RADIUS サーバから上記以外のサービスタイプが指定された場合，Normal モードでログインします。

13.4 RADIUS 機能の設定

RADIUS 認証サーバの情報および認証用パラメータを設定することで RADIUS クライアントとしてユーザ認証することが可能となります。

表 13.4-1　RADIUS 機能の設定

set radius auth { enable ｜ disable }	RADIUS 認証の有効／無効を設定します。
set radius auth timeout <timeout>	RADIUS 認証応答パケットの受信タイムアウト値を設定します。設定範囲は 1～30[秒]です。デフォルトは 5[秒]です。
set radius auth retransmit <retry>	RADIUS 認証要求パケットの再送信回数を設定します。設定範囲は 0～10[回]です。デフォルトは 3[回]です。
set radius auth method {PAP｜CHAP｜ default}	RADIUS 認証方法を設定します。
add radius auth server <IP_address> [port <port>] key <string > [Primary]	RADIUS 認証サーバを追加します。
update radius auth server <IP_address> [port <port>] [key <string>] [Primary]	すでに存在しているRADIUS認証サーバの設定内容を変更します。
delete radius auth server <IP_address>	RADIUS 認証サーバの設定を削除します。
show radius	RADIUS 設定情報を表示します。

以下に RADIUS 機能の設定例を記述します。

① RADIUS 認証方法を設定します。例では，PAP 認証方式を設定しています。

```
PureFlow(A)> set radius auth method PAP
```

② RADIUS 認証サーバを追加します。例では，2 つのサーバを登録しています。ひとつは，サーバ IP アドレス 192.168.1.10，RADIUS 共有鍵“testing123”で設定しています。もうひとつは，サーバ IP アドレス 192.168.1.11，RADIUS 共有鍵“testing789”で設定しています。 Primary 指定は，最初にログイン認証を問い合わせする RADIUS サーバに設定します。Primary 指定がない場合は，RADIUS サーバが登録された順番にログイン認証を問い合わせします。

```
PureFlow(A)> add radius auth server 192.168.1.10 key testing123 Primary
PureFlow(A)> add radius auth server 192.168.1.11 key testing789
```

③ RADIUS 機能を有効にします。

```
PureFlow(A)> set radius auth enable
```

④ 設定内容を確認します。

```
PureFlow(A)> show radius
RADIUS Authentication  : Enable
RADIUS method          : PAP
RADIUS server entries  : 2
Retry retransmit       : 5
Retry timeout          : 3

Type Pri Server          Port   key
---- --- --------------- ----- ------------------------------------------
auth  *  192.168.1.10     1812   "testing123"
auth     192.168.1.11     1812   "testing789"
PureFlow(A)>
```

13.5 RADIUS サーバの設定

RADIUS サーバの設定方法を説明します。RADIUS サーバには，以下のユーザ情報を設定します。

RADIUS 共有鍵

本装置に設定した RADIUS 共有鍵と同一の文字列を指定します。

ユーザ ID

ユーザ ID を設定します。

認証方法

本装置に設定した認証方法と同じ認証方法（CHAP または PAP）を指定します。

パスワード

パスワードを設定します。

サービスタイプ

このパラメータは必要に応じて設定します。RADIUS サーバからサービスタイプが通知されない場合，本装置は normal モードでのログインをユーザに許可します。RADIUS サーバからサービスタイプが通知され，そのサービスタイプが Administrative-User の場合，administrator モードでのログインをユーザに許可します。

本書では，RADIUS サーバとして FreeRADIUS バージョン 1 を使用した場合を説明しますが，実際の設定についてはお使いの RADIUS サーバの種類によって異なる設定が必要となります。また，FreeRADIUS をご利用の場合でも，FreeRADIUS のバージョンによって設定方法が異なります。FreeRADIUS は，LDAP（Lightweight Directory Access Protocol），SQL Server，UNIX システムのユーザ情報などのさまざまなユーザ情報と統合可能であり，企業内の多数のユーザの管理，認証，認可に使用することができます。

（注）

Linux に FreeRADIUS がインストールされていることを前提としています。FreeRADIUS の設定方法および，使用方法の詳細は，インストールされているソフトウェアのマニュアルを参照してください。

FreeRADIUS バージョン 1 の設定方法

① RADIUS 共有鍵の設定

RADIUS サーバに RADIUS クライアントとして登録する装置の IP アドレスおよび，RADIUS 共有鍵を以下の形式で設定します。

RADIUS サーバの/usr/local/etc/raddb/clients.conf ファイルを開き，適切なセクションに以下の設定を追加してください。

```
client 192.168.37.10 {
        secret =  testing123
        shortname =  wsx
}
```

② ユーザの設定

RADIUS サーバに本装置へのログインを許可するユーザ情報を設定します。ユーザごとに，ユーザ ID，認証方法，パスワード，サービスタイプを設定します。

RADIUS サーバの/usr/local/etc/raddb/users ファイルを開き，適切なセクションに以下の設定を追加してください。

1） 認証方法に CHAP を使用する場合

normal モードでのログインを許可するユーザの設定

```
user1 Cleartext-Password:=" user1passwd "
        Auth-Type:=CHAP,
        Service-Type=Login-User
```

Administrator モードでのログインを許可するユーザの設定

```
user2 Cleartext-Password:=" user2passwd "
        Auth-Type:=CHAP,
        Service-Type= Administrative-User
```

2） 認証方法に PAP を使用する場合

normal モードでのログインを許可するユーザの設定

```
user3 Cleartext-Password:=" user3passwd "
        Auth-Type:=PAP,
        Service-Type=Login-User
```

Administrator モードでのログインを許可するユーザの設定

```
user4 Cleartext-Password:=" user4passwd "
        Auth-Type:=PAP,
        Service-Type=Administrative-User
```

（空白ページ）

第14章　ダウンロードとアップロード

ここでは，ソフトウェアやコンフィギュレーションのダウンロード／アップロードについて説明します。

14.1　ソフトウェアのダウンロード／アップロード 14-2
14.1.1 ソフトウェアを SD カードよりダウンロードする. 14-2
14.1.2 ソフトウェアを SD カードにアップロードする.... 14-2
14.1.3 ソフトウェアを USB メモリよりダウンロードする 14-3
14.1.4 ソフトウェアを USB メモリにアップロードする.. 14-3
14.1.5 ソフトウェアを TFTP によりダウンロードする .. 14-3
14.1.6 ソフトウェアを FTP によりダウンロードする..... 14-4
14.1.7 ソフトウェアを WebGUI により ダウンロードする..... 14-5
14.2　ソフトウェアアップデートパッチの適用..... 14-6
14.2.1 ソフトウェアアップデートパッチを SD カードより適用する..... 14-6
14.2.2 ソフトウェアアップデートパッチを USB メモリより適用する..... 14-6
14.3　コンフィギュレーションのダウンロード／アップロード.... 14-7
14.3.1 コンフィギュレーションを SD カードよりダウンロードする..... 14-7
14.3.2 コンフィギュレーションを SD カードにアップロードする..... 14-7
14.3.3 コンフィギュレーションを USB メモリよりダウンロードする..... 14-8
14.3.4 コンフィギュレーションを USB メモリにアップロードする..... 14-8
14.3.5 コンフィギュレーションを TFTP によりダウンロードする 14-9
14.3.6 コンフィギュレーションを TFTP によりアップロードする..... 14-9
14.3.7 コンフィギュレーションを FTP によりダウンロードする 14-10
14.3.8 コンフィギュレーションを FTP によりアップロードする..... 14-10
14.4　ソフトウェアを再起動する 14-11

ソフトウェアやコンフィギュレーションをダウンロード／アップロードする場合は，SD カードまたは USB メモリを使用します。ファイルシステムは FAT16／FAT32 を対象とします。また，ソフトウェアのダウンロード，コンフィギュレーションのダウンロード／アップロードについては，システムインタフェースから TFTP，FTP，または WebGUI により実行することもできます。システムインタフェースを使用する場合には TFTP サーバ，FTP サーバ機能，または Web ブラウザを備えた PC などを用意してください。

また，SD カードまたは USB メモリをご使用になる場合，当社オプション品をご使用ください。当社オプション品以外の動作は保証対象外です。

14.1 ソフトウェアのダウンロード／アップロード

ソフトウェアをダウンロードするときの注意事項

当社指定の正規オブジェクトファイル（ファイル名：nf7500.bin）以外をダウンロードしますと，装置が起動しません。ダウンロード（download）コマンドで正規のオブジェクトファイル以外の誤ったファイルをダウンロードしないようにご注意ください。誤ったオブジェクトファイルをダウンロードした場合は，正規のオブジェクトファイルが入った SD カードまたは USB メモリを挿入して，装置を起動してください。その後，正規のオブジェクトファイルを再度ダウンロードしてください。

正規オブジェクトファイルの入手方法は，ご購入先にお問い合わせください。

14.1.1 ソフトウェアを SD カードよりダウンロードする

SD カードスロットに，新しいソフトウェアオブジェクトが入った SD カードを挿入して，新しいソフトウェアを装置にダウンロードします。ダウンロードしたソフトウェアは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのソフトウェアは別領域に待避し，新しいソフトウェアの書き込みを行います。バージョンアップ作業中は，SD カードを抜いたり，装置の電源が切断されないようにご注意ください。万が一作業中に，SD カードを抜いたり，装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。

```
PureFlow(A)> download sd obj nf7500.bin
Download "nf7500.bin" from Flash Memory Card (y/n)? y
Loading ....................
creating Backup from Master file....................completed.
Done.
PureFlow(A)>
```

ダウンロードが完了しても，新しいソフトウェアはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.1.2 ソフトウェアを SD カードにアップロードする

SD カードスロットに SD カードを挿入してソフトウェアを SD カードにアップロードします。アップロードしたソフトウェアは挿入した SD カードに保存されます。

```
PureFlow(A)> upload sd obj nf7500.bin
Upload as "nf7500.bin" to Flash Memory Card (y/n)? y
Loading ....................
Done.
PureFlow(A)>
```

14.1.3 ソフトウェアを USB メモリよりダウンロードする

USB ポートに，新しいソフトウェアオブジェクトが入った USB メモリを挿入して，新しいソフトウェアを装置にダウンロードします。ダウンロードしたソフトウェアは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのソフトウェアは別領域に待避し，新しいソフトウェアの書き込みを行います。バージョンアップ作業中は，USB メモリを抜いたり，装置の電源が切断されないようにご注意ください。万が一作業中に，USB メモリを抜いたり，装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。

```
PureFlow(A)> download usb obj nf7500.bin
Download "nf7500.bin" from USB Memory (y/n)? y
Loading ....................
creating Backup from Master file....................completed.
Done.
PureFlow(A)>
```

ダウンロードが完了しても，新しいソフトウェアはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.1.4 ソフトウェアを USB メモリにアップロードする

USB ポートに USB メモリを挿入してソフトウェアを USB メモリにアップロードします。アップロードしたソフトウェアは挿入した USB メモリに保存されます。

```
PureFlow(A)> upload usb obj nf7500.bin
Upload as "nf7500.bin" to USB Memory (y/n)? y
Loading ....................
Done.
PureFlow(A)>
```

14.1.5 ソフトウェアを TFTP によりダウンロードする

TFTP によりソフトウェアを装置にダウンロードします。ダウンロードしたソフトウェアは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのソフトウェアは別領域に待避し，新しいソフトウェアの書き込みを行います。バージョンアップ作業中は装置の電源が切断されないようにご注意ください。万が一作業中に装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。また，ダウンロード中に通信が切断された場合は，再度ダウンロード作業をやり直してください。

ソフトウェアを装置にダウンロードするには以下のコマンドを使用します。あらかじめ TFTP サーバと通信できるようにシステムインタフェースに正しい IP アドレスを設定してください。システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。

ソフトウェアのファイルサイズが 32MByte を超えるため，RFC2349 に規定される tsize オプションに対応した TFTP サーバをお使いください。

```
PureFlow(A)> download tftp obj 192.168.100.40 nf7500.bin
Download "nf7500.bin" from 192.168.100.40 (y/n)? y
Loading ...
creating Backup from Master file....................completed.
Done.
PureFlow(A)>
```

ダウンロードが完了しても，新しいソフトウェアはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.1.6 ソフトウェアを FTP によりダウンロードする

FTP によりソフトウェアを装置にダウンロードします。ダウンロードしたソフトウェアは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのソフトウェアは別領域に待避し，新しいソフトウェアの書き込みを行います。バージョンアップ作業中は装置の電源が切断されないようにご注意ください。万が一作業中に装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。また，ダウンロード中に通信が切断された場合は，再度ダウンロード作業をやり直してください。

ソフトウェアを装置にダウンロードするには以下のコマンドを使用します。あらかじめ FTP サーバと通信できるようにシステムインタフェースに正しい IP アドレスを設定してください。システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。また，ダウンロードで使用する FTP サーバのユーザ名とパスワードを用意してください。

```
PureFlow(A)> download ftp obj 192.168.100.40 nf7500.bin
Name:ftpuser  (ユーザ名を入力)
Password:     (パスワードを入力)
Download "nf7500.bin" from 192.168.100.40 (y/n)? y
Loading ...
creating Backup from Master file....................completed.
Done.
PureFlow(A)>
```

ダウンロードが完了しても，新しいソフトウェアはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.1.7 ソフトウェアを WebGUI によりダウンロードする

WebGUI によりソフトウェアを装置にダウンロードします。ダウンロードしたソフトウェアは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのソフトウェアは別領域に待避し，新しいソフトウェアの書き込みを行います。バージョンアップ作業中は装置の電源が切断されないようにご注意ください。万が一作業中に装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。また，ダウンロード中に通信が切断された場合は，再度ダウンロード作業をやり直してください。

WebGUI についてのさらに詳細な説明は，「WebGUI 操作説明書（NF7500-W014J）」を参照してください。あらかじめ WebGUI と通信できるようにシステムインタフェースに正しい IP アドレスを設定してください。システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。

ダウンロードが完了しても，新しいソフトウェアはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.2 ソフトウェアアップデートパッチの適用

本装置のソフトウェアは，ソフトウェアアップデートパッチの適用により新しいソフトウェアに更新することもできます。パッチ適用は，ソフトウェアオブジェクトのダウンロードと同様の手順で行います。ただし，パッチ適用を TFTP または FTP 経由で行うことはできません。

ソフトウェアアップデートパッチの入手方法は，当社営業窓口へお問い合わせください。

14.2.1 ソフトウェアアップデートパッチを SD カードより適用する

SD カードスロットに，ソフトウェアアップデートパッチが入った SD カードを挿入して，装置内部のソフトウェアに適用します。パッチ適用作業中は，SD カードを抜いたり，装置の電源が切断されないようにご注意ください。万が一作業中に，SD カードを抜いたり，装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してパッチ適用作業をやり直してください。

```
PureFlow(A)> download sd patch
Apply patch from Flash Memory Card (y/n)? y
Appling file system patch ............... done
Appling apps patch ............... done
Appling fcpu patch ............... done
creating Backup from Master file....................completed.
Done.
PureFlow(A)>
```

パッチ適用が完了しても，新しいソフトウェアはすぐに反映されません。パッチ適用が完了したあとで，装置を再起動してください。

14.2.2 ソフトウェアアップデートパッチを USB メモリより適用する

USB ポートに，ソフトウェアアップデートパッチが入った USB メモリを挿入して，装置内部のソフトウェアに適用します。パッチ適用作業中は，USB メモリを抜いたり，装置の電源が切断されないようにご注意ください。万が一作業中に，USB メモリを抜いたり，装置の電源を切断してしまった場合は，別領域に待避してある古いバージョンのソフトウェアを再ロードしますので，再度装置を起動してパッチ適用作業をやり直してください。

```
PureFlow(A)> download usb patch
Apply patch from USB Memory (y/n)? y
Appling file system patch ............... done
Appling apps patch ............... done
Appling fcpu patch ............... done
creating Backup from Master file....................completed.
Done.
PureFlow(A)>
```

パッチ適用が完了しても，新しいソフトウェアはすぐに反映されません。パッチ適用が完了したあとで，装置を再起動してください。

14.3 コンフィギュレーションのダウンロード／アップロード

コンフィギュレーションをダウンロードするときの注意事項

ダウンロードするコンフィギュレーションファイルは，アップロード（upload）コマンドにより SD カード，USB メモリ，TFTP サーバ，FTP サーバにアップロードした正規コンフィギュレーションファイルを使用してください。正規コンフィギュレーションファイル以外をダウンロードしますと，装置が起動しない場合があります。誤ったコンフィギュレーションファイルをダウンロードした場合は，正規のコンフィギュレーションファイル（ファイル名：extcnf.txt）が入った SD カードまたは USB メモリを挿入して，装置を起動してください。その後，save コマンドにて設定内容を保存してください。

14.3.1 コンフィギュレーションを SD カードよりダウンロードする

SD カードスロットに SD カードを挿入して新しいコンフィギュレーションファイルを装置にダウンロードします。ダウンロードしたコンフィギュレーションファイルは自動的に内蔵フラッシュメモリに保存されます。このとき古いコンフィギュレーションファイルは別領域に待避し，新しいコンフィギュレーションファイルの書き込みを行います。ダウンロードが完了しても，新しいコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。ダウンロード作業中は，SD カードを抜いたり，装置の電源が切断されないようにご注意ください。万が一作業中に，SD カードを抜いたり，装置の電源を切断してしまった場合は，別領域に待避してある古いコンフィギュレーションファイルを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。

```
PureFlow(A)> download sd conf config.txt
Download "config.txt" from Flash Memory Card (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

ダウンロードが完了しても，ダウンロードしたコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.3.2 コンフィギュレーションを SD カードにアップロードする

SD カードスロットに SD カードを挿入してコンフィギュレーションファイルを SD カードにアップロードします。アップロードしたコンフィギュレーションファイルは挿入した SD カードに保存されます。

```
PureFlow(A)> upload sd conf config.txt
Upload as "config.txt" to Flash Memory Card (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

動作中のコンフィギュレーション情報ではなく，内部フラッシュメモリにセーブされたコンフィギュレーション情報がアップロードされます。コンフィギュレーション情報は，save config コマンドを実行したとき，内部フラッシュメモリに保存されます。

14.3.3 コンフィギュレーションを USB メモリよりダウンロードする

USB スロットに USB メモリを挿入して新しいコンフィギュレーションファイルを装置にダウンロードします。ダウンロードしたコンフィギュレーションファイルは自動的に内蔵フラッシュメモリに保存されます。このとき古いコンフィギュレーションファイルは別領域に待避し，新しいコンフィギュレーションファイルの書き込みを行います。ダウンロードが完了しても，新しいコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。ダウンロード作業中は，USB メモリを抜いたり，装置の電源が切断されないようにご注意ください。万が一作業中に，USB メモリを抜いたり，装置の電源を切断してしまった場合は，別領域に待避してある古いコンフィギュレーションファイルを再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。

```
PureFlow(A)> download usb conf config.txt
Download "config.txt" from USB Memory (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

ダウンロードが完了しても，ダウンロードしたコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.3.4 コンフィギュレーションを USB メモリにアップロードする

USB ポートに USB メモリを挿入してコンフィギュレーションファイルを USB メモリにアップロードします。アップロードしたコンフィギュレーションファイルは挿入した USB メモリに保存されます。

```
PureFlow(A)> upload usb conf config.txt
Upload as "config.txt" to USB Memory (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

動作中のコンフィギュレーション情報ではなく，内部フラッシュメモリにセーブされたコンフィギュレーション情報がアップロードされます。コンフィギュレーション情報は，save config コマンドを実行したとき，内部フラッシュメモリに保存されます。

14.3.5 コンフィギュレーションを TFTP によりダウンロードする

TFTP によりコンフィギュレーションファイルを装置にダウンロードします。ダウンロードしたコンフィギュレーションファイルは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのコンフィギュレーションファイルは別領域に待避し，新しいコンフィギュレーションファイルの書き込みを行います。ダウンロードが完了しても，新しいコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。ダウンロード作業中は装置の電源が切断されないようにご注意ください。万が一作業中に装置の電源を切断してしまった場合は，別領域に待避してある古いコンフィギュレーションファイルで再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。また，ダウンロード中に通信が切断された場合は，再度ダウンロード作業をやり直してください。

コンフィギュレーションファイルを装置にダウンロードするには以下のコマンドを使用します。あらかじめ TFTP サーバと通信できるようにシステムインタフェースに正しい IP アドレスを設定してください。システムインタフェースの設定の説明は「第 7 章　システムインタフェースの設定」を参照してください。

```
PureFlow(A)> download tftp conf 192.168.100.40 config.txt
Download ”config.txt” from 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

ダウンロードが完了しても，ダウンロードしたコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.3.6 コンフィギュレーションを TFTP によりアップロードする

TFTP によりコンフィギュレーションファイルを TFTP サーバにアップロードします。アップロードしたコンフィギュレーションファイルは TFTP サーバに保存されます。

```
PureFlow(A)> upload tftp conf 192.168.100.40 config.txt
Upload as ”config.txt” to 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

動作中のコンフィギュレーション情報ではなく，内部フラッシュメモリにセーブされたコンフィギュレーション情報がアップロードされます。

14.3.7 コンフィギュレーションを FTP によりダウンロードする

FTP によりコンフィギュレーションファイルを装置にダウンロードします。ダウンロードしたコンフィギュレーションファイルは自動的に内蔵フラッシュメモリに保存されます。このとき古いバージョンのコンフィギュレーションファイルは別領域に待避し，新しいコンフィギュレーションファイルの書き込みを行います。ダウンロードが完了しても，新しいコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。ダウンロード作業中は装置の電源が切断されないようにご注意ください。万が一作業中に装置の電源を切断してしまった場合は，別領域に待避してある古いコンフィギュレーションファイルで再ロードしますので，再度装置を起動してダウンロード作業をやり直してください。また，ダウンロード中に通信が切断された場合は，再度ダウンロード作業をやり直してください。

コンフィギュレーションファイルを装置にダウンロードするには以下のコマンドを使用します。あらかじめ FTP サーバと通信できるようにシステムインタフェースに正しい IP アドレスを設定してください。システムインタフェースの設定の説明は「第 7 章 システムインタフェースの設定」を参照してください。また，ダウンロードで使用する FTP サーバのユーザ名とパスワードを用意してください。

```
PureFlow(A)> download ftp conf 192.168.100.40 config.txt
Name:ftpuser  (ユーザ名を入力)
Password:     (パスワードを入力)
Download "config.txt" from 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

ダウンロードが完了しても，ダウンロードしたコンフィギュレーションはすぐに反映されません。ダウンロードが完了したあとで，装置を再起動してください。

14.3.8 コンフィギュレーションを FTP によりアップロードする

FTP によりコンフィギュレーションファイルを FTP サーバにアップロードします。アップロードしたコンフィギュレーションファイルは FTP サーバに保存されます。

```
PureFlow(A)> upload ftp conf 192.168.100.40 config.txt
Name:ftpuser  (ユーザ名を入力)
Password:     (パスワードを入力)
Upload as "config.txt" to 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

動作中のコンフィギュレーション情報ではなく，内部フラッシュメモリにセーブされたコンフィギュレーション情報がアップロードされます。

14.4 ソフトウェアを再起動する

ダウンロードが完了したあとは新しいソフトウェアで再起動させます。

(1) 装置を再起動する

装置の再起動方法です。電源を再投入するか以下のコマンドを使用してください。

```
PureFlow(A)> reboot system
Rebooting the system, ok(y/n)? y
```

(2) 起動ファイルを確認する

装置の起動時に起動ファイル種別の結果が表示されます。

```
Loading Object from Master.
```

ダウンロード中の電源断等でダウンロードが異常終了すると，Master ファイルが CRC エラーとなり，Backup ファイルで起動します。Backup ファイルでの起動後に再度ダウンロードしてください。CRC エラーの表示は，シリアルコンソールのボーレートを 9600 bps に設定している場合のみ表示されます。

```
checkCRC:NG

Loading Object from Backup.
```

起動ファイル種別は以下のとおりです。

表 14.4-1　起動ファイル種別

表示	説明	優先度
Loading Object from USB memory.	USB メモリ上のファイル	高
Loading Object from SD Card.	SD カード上のファイル	↑
Loading Object from Master.	Master ファイル	↓
Loading Object from Backup.	Backup ファイル	低

(3) 再起動の完了確認

再起動は Telnet/SSH の接続がいったん切断されます。装置起動後，再度 Telnet/SSH によりログインし直してください。

（空白ページ）

第15章　WebAPI 機能

ここでは，WebAPI（Web Application Program Interface）機能について説明します。

15.1　概要 15-2
15.2　通信プロトコル 15-3
15.3　HTTP メソッド 15-3
15.4　JSON 形式 15-4
15.5　API 一覧 15-5
15.6　共通エラーメッセージ 15-6
15.7　エラーメッセージ一覧 15-7

15.1 概要

WebAPI 機能は，本装置のトラフィックコントロール機能の設定を行う際に，HTTP（Hypertext Transfer Protocol：RFC2616）を使用して設定を行う機能です。本装置は，HTTP サーバとして動作し，外部に設置した管理端末の HTTP クライアントから JSON（JavaScript Object Notation：RFC4627）形式で設定を行うことができます。

クラウド環境において，クラウドサーバの構成変更に連動して手動でネットワーク装置のトラフィックコントロール設定を更新することは困難になってきています。クラウド管理端末上で JSON 形式をサポートしたプログラミング言語を利用し，クラウドサーバの構成変更に連動して本装置のトラフィックコントロール設定を更新するユーザプログラムを作成することにより，本装置の設定更新を自動化することができます。

図 15.1-1　WebAPI 機能

また，SSL 暗号化通信による HTTP 接続（HTTPS: Hypertext Transfer Secure）を使用することができます。HTTPS では WebAPI の通信が暗号化され，盗聴やなりすましを防ぐことができます。

WebAPI は同時に 4 セッションまで実行可能です。
同時に 5 セッション以上の WebAPI を実行した場合，5 セッション以上の接続は可能ですが，要求発行時にいずれかのセッションでエラーが発生します。例えば，セッション 1～4 で WebAPI を実行中に 5 セッション目の要求を発行すると，セッション 1～5 のいずれかでセッション数超過エラーやコネクションの切断が発生します。WebAPI は WebGUI と合わせて 4 セッション以内でご利用ください。なお，セッション数には Telnet セッションと SSH セッションのセッション数は含みません。

HTTP リクエストと次の HTTP リクエストまでのタイムアウト時間は 15 秒です。

15.2 通信プロトコル

WebAPI 機能では通信プロトコルとして HTTP または HTTPS を使用します。通信プロトコルの設定には以下のコマンドを使用します。

表 15.2-1　通信プロトコルの設定

set http protocol {normalhttp | httpsecure}	Web アプリケーションで使用する通信プロトコルを設定します。 デフォルトは normalhttp です。 normalhttp： HTTP を使用します。 httpsecure： HTTPS を使用します。 HTTP と HTTPS の同時利用はできません。
show http	Web アプリケーションの設定を表示します。

15.3 HTTP メソッド

WebAPI 機能がサポートする HTTP メソッドは以下のとおりです。

表 15.3-1　サポートする HTTP メソッド

HTTP メソッド	用途
HEAD	アクセス可否の判断等に使用されます。
GET	情報の取得に使用されます。 本装置では情報取得系の要求で使用します。
POST	情報の設定に使用されます。 本装置では追加，更新，削除系の要求で使用します。

なお，HTTP クライアントから上記以外のメソッドが指定された場合，HTTP ステータスコード 405（Method Not Allowed）を返します。

15.4 JSON 形式

WebAPI 機能は GET や POST メソッドで JSON 形式のデータを利用します。JSON とはデータを表現するためのデータ記述言語です。JSON の記述方法では，パラメータのキーと値の組をコロン":"でペアにします。パラメータが複数ある場合はコンマ", "で区切ります。これらの全体を中括弧"{"および"}"で括ります。

WebAPI 機能ではキーや値はすべて文字列で記述してください。API の種別を示すキー"command"と，API に相当する CLI コマンドのパラメータを指定します。WebAPI においてはキーの記述順序は順不同です。CLI コマンドのパラメータ順序と合わせる必要はありません。

下記にシナリオ追加 API の JSON 記述例を示します。

```
{
        " command": "add scenario" "
        "scenario_name": "/port1/North",
        "action": "aggregate",
        "min_bandwidth": "5M",
        "peak_bandwidth": "8M",
        "class": "2",
        "bufsize": "512k"
}
```

JSON の記述方法の詳細については「付録 E　JSON の記述方法」を参照してください。

15.5 API 一覧

WebAPI は，シナリオ，フィルタ，ルールリストに関する設定，および情報取得の API を提供します。それぞれの機能は，相当する CLI コマンドと同等です。API で指定するパラメータ，値の範囲や省略可能／不可についても同等です。各 API の詳細については「付録 F　WebAPI 詳細」を参照してください。

表 15.5-1　API 一覧

対象	操作	相当する CLI コマンド
シナリオ	追加	add scenario
	更新	update scenario
	削除	delete scenario
	情報取得	show scenario
アプリケーション高速化	追加	add apl-accel
	更新	update apl-accel
	削除	delete apl-accel
フィルタ	モード設定	set filter mode
	追加	add filter
	削除	delete filter
	情報取得	show filter
ルールリスト	グループ追加	add rulelist group
	グループ削除	delete rulelist group
	エントリ追加	add rulelist entry
	エントリ削除	delete rulelist entry
	情報取得	show rulelist
チャネル	追加	add channel
	削除	delete channel
	情報取得	show channel
チャネルインタフェース	設定	set ip channel
	解除	unset ip channel
	情報取得	show ip channel
OpenFlow コントローラ	追加	add openflow controller
	削除	delete openflow controller
	情報取得	show openflow controller
チャネルインタフェースのスタティック経路	追加	add route
	削除	delete route
	情報取得	show route target
コンフィギュレーション	保存	save config
	情報取得	show save status※

※ コンフィギュレーションの情報取得 API は，コンフィギュレーションの保存が実行中であるかどうかのステータスを取得する API です。コンフィギュレーションの保存が実行中である間はコンフィギュレーションの保存を重複して実行できません。保存の所要時間については「第 3 章　設定の基本」を参照してください。

15.6 共通エラーメッセージ

HTTP メソッド，JSON フォーマットおよび指定内容が正しい場合，HTTP ステータスコード 200（OK）に加えて，"status": "OK."を返します。HTTP メソッドおよび JSON フォーマットが正しいが，指定内容が不正な場合，HTTP ステータスコード 200（OK）に加えて"status":"error"とエラーメッセージを返します。共通エラーメッセージは以下のとおりです。

表 15.6-1　共通エラーメッセージ

エラーメッセージ	説明
Specified command is invalid.	API コマンドが不正です。 指定した JSON 形式のキーと値が正しいか確認してください。
Required parameter is not specified.	必須のパラメータが指定されていません。 指定した JSON 形式のキーと値が正しいか確認してください。
Specified command is invalid when GET request.	GET メソッドでは指定できないコマンド（追加・更新・削除）です。 指定した JSON 形式のキーと値が正しいか確認してください。
Specified command is invalid when POST request.	POST メソッドでは指定できないコマンド（情報取得）です。 指定した JSON 形式のキーと値が正しいか確認してください。
WebAPI session is full.	WebAPI の最大セッション数が超過しました。 時間をおいて再度実行してください。
Failed to create pipe.	内部通信用の PIPE 作成でエラーが発生しました。 時間をおいて再度実行してください。
No response message from LR.	内部ソフトウェアからの応答がありません。 時間をおいて再度実行してください。

15.7 エラーメッセージ一覧

各 API 個別のエラーメッセージは以下のとおりです。

表 15.7-1　エラーメッセージ一覧

API	エラーメッセージ
シナリオ追加	Specified scenario class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・class の指定が不正です。
	Specified scenario fail action class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・Fail Action class の指定が不正です。
	Specified minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Minimum Bandwidth の指定が不正です。
	Specified peak bandwidth is invalid. (Valid from 10k to 1G) ・Peak Bandwidth の指定が不正です。
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Fail Action Minimum Bandwidth の指定が不正です。
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 1G) ・Fail Action Peak Bandwidth の指定が不正です。
	Peak Bandwidth should be greater than minimum bandwidth. ・peak_bandwidth は min_bandwidth 以上に設定する必要があります。
	Specified buff size is invalid. (Valid from 2k to 100M) ・bufsize の指定が不正です。
	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is already used. ・指定のシナリオ名はすでに別のシナリオで使われています。
	Specified scenario of upper level hierarchy is not found. ・上位階層のシナリオが存在しません。
	maximum number of scenario was exceeded. ・シナリオの最大登録件数を超えました。
	Specified scenario ID is invalid. (Valid from 1 to 4096) ・シナリオインデックスが範囲外です。
	Specified scenario ID is already used. ・指定のシナリオインデックスはすでに別のシナリオで使われています。
	Specified max Q num is not licensed. (Valid from 1 to 2048) ・maxquenum が範囲外です。
	Specified max Q num is invalid. (Valid from 1 to 4096) ・maxquenum が範囲外です。

API	エラーメッセージ
シナリオ追加（続き）	Extended number of scenario is not licensed. ・シナリオ拡張ライセンスの制限数を超えてシナリオを登録することはできません。 ・シナリオ拡張ライセンスの制限数を超えた maxquenum を設定することはできません。
	Specified Q division field is invalid. Valid fields: default, vid, cos, inner-vid, inner-cos, ethertype, sip, dip, tos, proto, sport, dport (multiple fields can be specified with separated comma without space) ・quedivision のフィールド指定が不正です。
	failaction is not specified. ・failaction を指定せずに fail_min_bw, fail_peak_bw, fail_class を設定することはできません。
	Specified failaction is invalid. ・fail_min_bw, fail_peak_bw, fail_class は failaction として forwardattribute を指定した場合のみ設定可能です。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Peer IP version and second-peer IP version are different. ・peer と second-peer の IP バージョンは一致させる必要があります。
	Peer and second-peer are same IP address. ・peer と second-peer には異なる IP アドレスを設定する必要があります。
	Specified dport is invalid. (Valid from 10001 to 20000) ・dport の指定が不正です。
	Specified Dport is already used. ・指定の dport はすでに別のシナリオで使われています。
	Specified vid is invalid. (Valid from 1 to 4094) ・VLAN ID の指定が不正です。
	Specified inner-vid is invalid. (Valid from 1 to 4094) ・Inner-VLAN ID の指定が不正です。
	VID must be specified when inner-VID is specified. ・Inner VLAN ID は VLAN ID を指定した場合のみ指定できます。
	Specified cos is invalid. (Valid from 0 to 7) ・CoS 値の指定が不正です。
	Specified inner-cos is invalid. (Valid from 0 to 7) ・Inner-CoS 値の指定が不正です。
	VID must be specified when CoS is specified. ・CoS 値は VLAN ID を指定した場合のみ指定できます。
	Inner-VID must be specified when inner-cos is specified. ・Inner CoS 値は Inner VLAN ID を指定した場合のみ指定できます。

API	エラーメッセージ
シナリオ追加(続き)	Specified dscp is invalid. (Valid from 0 to 63) ・DSCP 値の指定が不正です。
	Specified tcp-mem is invalid. (Valid from 64k to 200M) ・TCP のバッファサイズの指定が不正です。
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) ・自動バイパスの RTT しきい値の指定が不正です。
	Specified peak bandwidth is not licensed. ・指定した帯域幅のライセンスがありません。
	Data block size should be divided by fec block size. ・データブロックサイズは FEC ブロックサイズで割り切れる値に設定する必要があります。
	Data block size should be greater than fec block size. ・データブロックサイズは FEC ブロックサイズより大きな値に設定する必要があります。
	Specified fec block size is invalid. (Valid from 2K to 50K) ・FEC ブロックサイズの指定が不正です。
	Specified data block size is invalid. (Valid from 2K to 200K) ・データブロックサイズの指定が不正です。
	Specified fec session is invalid. (Valid from 0 to 1000) ・FEC セッション数の指定が不正です。
	FEC function is not licensed. ・TCP-FEC 機能のライセンスがありません。
	Maximum number of secondary peer was exceeded. ・second-peer を指定したシナリオの最大登録件数を超えました。
	Maximum number of keep alive scenario was exceeded. ・bypass-keep を有効に指定したシナリオの最大登録件数を超えました。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
シナリオ更新	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・class の指定が不正です。
	Specified scenario fail action class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・Fail Action class の指定が不正です。

API	エラーメッセージ
シナリオ更新（続き）	Specified minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Minimum Bandwidth の指定が不正です。
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Fail Action Minimum Bandwidth の指定が不正です。
	Specified peak bandwidth is invalid. (Valid from 10k to 1G) ・Peak Bandwidth の指定が不正です。
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 1G) ・Fail Action Peak Bandwidth の指定が不正です。
	Peak bandwidth should be greater than minimum bandwidth. ・peak_bandwidth は min_bandwidth 以上に設定する必要があります。
	Specified buff size is invalid. (Valid from 2k to 100M) ・bufsize の指定が不正です。
	It is necessary to set one or more parameters. ・1 つ以上のパラメータを設定する必要があります。
	Specified scenario mode is invalid. ・シナリオモードの指定が不正です。
	Specified max Q num is not licensed. (Valid from 1 to 2048) ・maxquenum が範囲外です。
	Specified max Q num is invalid. (Valid from 1 to 4096) ・maxquenum が範囲外です。
	Specified Q division Field field is invalid. Valid fields: default, vid, cos, inner-vid, inner-cos, ethertype, sip, dip, tos, proto, sport, dport (multiple fields can be specified with separated comma without space) ・quedivision のフィールド指定が不正です。
	Fail action forward is incorrect. Specified Failaction is invalid. ・fail_min_bw, fail_peak_bw, fail_class は failaction として forwardattribute を指定した場合のみ設定可能です。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Specified cos is invalid. (Valid from 0 to 7) ・CoS 値の指定が不正です。
	Specified inner-cos is invalid. (Valid from 0 to 7) ・Inner-CoS 値の指定が不正です。
	Specified dscp is invalid. (Valid from 0 to 63) ・DSCP 値の指定が不正です。
	Specified tcp-mem is invalid. (Valid from 64k to 200M) ・TCP のバッファサイズの指定が不正です。
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) ・トラフィックアクセラレーションの自動バイパスの RTT しきい値の指定が不正です。

API	エラーメッセージ
シナリオ更新（続き）	Specified peak bandwidth is not licensed. ・指定した帯域幅のライセンスがありません。
	Data block size should be divided by fec block size. ・データブロックサイズは FEC ブロックサイズで割り切れる値に設定する必要があります。
	Data block size should be greater than fec block size. ・データブロックサイズは FEC ブロックサイズより大きな値に設定する必要があります。
	Specified fec block size is invalid. (Valid from 2K to 50K) ・FEC ブロックサイズの指定が不正です。
	Specified data block size is invalid. (Valid from 2K to 200K) ・データブロックサイズの指定が不正です。
	Specified fec session is invalid. (Valid from 0 to 1000) ・FEC セッション数の指定が不正です。
	FEC function is not licensed. ・TCP-FEC 機能のライセンスがありません。
	Maximum number of keep alive scenario was exceeded. ・bypass-keep を有効に指定したシナリオの最大登録件数を超えました。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
シナリオ削除	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Down level hierarchy scenario exists. ・下位階層のシナリオが存在します。

API	エラーメッセージ
シナリオ情報取得	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Next scenario is not exist. ・next シナリオが存在しません。

API	エラーメッセージ
アプリケーション高速化追加	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario name is not wan-accel mode. ・指定シナリオがアクセラレーションモードではありません。
	Specified protocol is already used. ・指定シナリオはすでに使用されています。
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) ・SMB TCP Port の指定が不正です。
	Specified smb session is invalid. (Valid from 0 to 1000) ・SMB Session の指定が不正です。
	Specified read cache size is invalid. (Valid from 64k to 60M) ・Read Cache Size の指定が不正です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
アプリケーション高速化更新	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario name is not wan-accel mode. ・指定シナリオがアクセラレーションモードではありません。
	Specified protocol is not used. ・指定プロトコルは使用されていません。
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) ・SMB TCP Port の指定が不正です。
	Specified smb session is invalid. (Valid from 0 to 1000) ・SMB Session の指定が不正です。
	Specified read cache size is invalid. (Valid from 64k to 60M) ・Read Cache Size の指定が不正です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
アプリケーション高速化削除	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario name is not wan-accel mode. ・指定シナリオがアクセラレーションモードではありません。
	Specified protocol is already disabled. ・指定されたプロトコルはすでに無効です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
フィルタモード設定	Slot #N is invalid. ・スロット指定が不正です。
	Port <slot/port> is invalid. ・ポート指定が不正です。
	Specified field is invalid. Valid fields: default, vid, cos, inner-vid, inner-cos, sip, dip, tos, proto, sport, dport (multiple fields can be specified with separated comma without space) ・フローを識別するフィールド名の指定が不正です。

API	エラーメッセージ
フィルタ追加	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid filter name length is from 1 to 48.) ・フィルタ名の指定が不正です。
	Specified filter Name is already used. ・指定のフィルタ名はすでに別のフィルタで使われています。
	Specified ether type is invalid. (Valid from 0x0000 to 0xFFFF) ・Ether type の指定が不正です。
	Specified vid is invalid. (Valid from 0 to 4094, Or Start - End) ・VLAN ID の指定が不正です。

API	エラーメッセージ
フィルタ追加(続き)	Specified cos is invalid. (Valid from 0 to 7, Or Start - End) ・CoS 値の指定が不正です。
	Specified inner-vid is invalid. (Valid from 0 to 4094, Or Start - End) ・Inner-VLAN ID の指定が不正です。
	VID must be specified when inner-VID is specified. ・Inner VLAN ID は VLAN ID を指定した場合のみ指定できます。
	Specified inner-cos is invalid. (Valid from 0 to 7, Or Start - End) ・Inner-CoS 値の指定が不正です。
	The format or value of the specified source IP address is invalid. ・Source IP address の指定が不正です。
	The format or value of the specified destination IP address is invalid. ・Destination IP address の指定が不正です。
	The format or value of the specified source IPv6 address is invalid. ・Source IPv6 address の指定が不正です。
	The format or value of the specified destination IPv6 address is invalid. ・Destination IPv6 address の指定が不正です。
	Specified rulelist name of source IP address is invalid. Specified rulelist name of destination IP address is invalid. Specified rulelist name of source port is invalid. Specified rulelist name of destination port is invalid. ・ルールリスト名が不正です。
	Specified rulelist name of source IP address is not used. Specified rulelist name of destination IP address is not used. Specified rulelist name of source port is not used. Specified rulelist name of destination port is not used. ・指定ルールリストが存在しません。
	IP Filter and rulelist of source IP address is not same type. IP Filter and rulelist of destination IP address is not same type. IP Filter and rulelist of source port is not same type. IP Filter and rulelist of destination port is not same type. ・対象ルールリストと種別が異なります。
	Specified tos is invalid. (Valid from 0 to 255, Or Start - End) ・ToS 値の指定が不正です。
	Specified protocol number is invalid. (Valid from 0 to 255, Start - End, Or tcp/udp/icmp/icmpv6) ・プロトコル番号の指定が不正です。
	Specified source TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・sport 番号の指定が不正です。
	Specified destination TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・dport 番号の指定が不正です。

API	エラーメッセージ
フィルタ追加（続き）	Specified filter priority is invalid. (Valid from 1 to 40000) ・フィルタ優先度の指定が不正です。
	maximum number of filter was exceeded. ・フィルタの最大登録件数を超えました。
	It is necessary to set one or more parameters other than Priority. ・Ethernet フィルタは Priority 以外で少なくとも 1 つのパラメータを指定する必要があります。
	Filter type is different. Please specify same type of wan-accel scenario. ・wan-accel シナリオの peer と同じ IP バージョンを指定してください。

API	エラーメッセージ
フィルタ削除	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid filter name length is from 1 to 48.) ・フィルタ名の指定が不正です。
	Specified filter name is not used. ・指定フィルタが存在しません。

API	エラーメッセージ
フィルタ情報取得	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid filter name length is from 1 to 48.) ・フィルタ名の指定が不正です。
	Specified filter name is not used. ・指定フィルタが存在しません。
	Next filter is not exist. ・次のフィルタが存在しません。

API	エラーメッセージ
ルールリストグループ追加	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is already in use. ・同一名のルールリストがすでに存在します。
	Maximum number of rulelist was exceeded. ・ルールリストの最大登録件数を超えました。

API	エラーメッセージ
ルールリストグループ削除	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。
	Rulelist is used by filter. ・ルールリストがフィルタに設定されています。

API	エラーメッセージ
ルールリストエントリ追加	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。
	The format or value of the specified IP address is invalid. ・IP address の指定が不正です。
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・TCP/UDP ポート番号の指定が不正です。
	Maximum number of rulelist entry was exceeded. ・指定ルールリストのルールリストエントリ最大登録件数(512 件)を超えました。
	Maximum number of total rulelist entry was exceeded. ・全ルールリスト合計のルールリストエントリ最大登録件数(64000 件)を超えました。
	Specified rulelist entry is already in use. ・指定ルールリストエントリはすでに登録されています。
	Rulelist entry and rulelist is not same type. ・対象ルールリストと種類が異なります。

API	エラーメッセージ
ルールリストエントリ削除	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。
	The format or value of the specified IP address is invalid. ・IP address の指定が不正です。
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・TCP/UDP ポート番号の指定が不正です。
	Rulelist entry and rulelist is not same type. ・対象ルールリストと種別が異なります。
	Specified rulelist entry is not used. ・指定ルールリストエントリが存在しません。

API	エラーメッセージ
ルールリスト情報取得	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。

API	エラーメッセージ
チャネル追加	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Channel name already exists. ・指定のチャネル名はすでに別のチャネルで使われています。
	Slot #N is invalid. ・スロット指定が不正です。
	Port <slot/port> is invalid. ・ポート指定が不正です。
	Specified vid is invalid. (Valid from 1 to 4094) ・VLAN ID の指定が不正です。
	Specified TPID is invalid. (Valid 0x8100,0x88a8,0x9100,0x9200 or 0x9300.) ・TPID の指定が不正です。
	Specified inner-vid is invalid. (Valid from 1 to 4094) ・Inner-VLAN ID の指定が不正です。

API	エラーメッセージ
チャネル追加(続き)	VID must be specified when inner-VID is specified. ・Inner VLAN ID は VLAN ID を指定した場合のみ指定できます。
	Specified mtu is invalid. (Valid from 300 to 10200) ・mtu の指定が不正です。
	Specified vid and inner-vid is already used on channel “channel name”. ・指定された vid と inner-vid はすでに“channel name”チャネルで使われています。
	Specified port is already used on other default-channel. ・指定のポートはすでに別のデフォルトチャネルで使われています。
	Maximum number of channel was exceeded. ・チャネルの最大登録件数を超えました。

API	エラーメッセージ
チャネル削除	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。

API	エラーメッセージ
チャネル情報取得	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Next channel is not exist. ・次チャネルが存在しません。

API	エラーメッセージ
インタフェース設定	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。
	Default-channel cannot be set for this command. ・デフォルトチャネルは指定できません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
インタフェース解除	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Cannot specified "ipv4" or "ipv6". ・“all”に対して“IPv4”と“IPv6”は指定できません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
インタフェース情報表示	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	IP interface is not configured. ・指定チャネルまたは次チャネルに IP アドレスが設定されていません。
	Next IP channel is not exist. ・次チャネルインタフェースが存在しません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
スタティック経路追加	Route entry already exists. ・すでに存在するルートエントリです。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。 ・指定したプレフィックス長の値が不正です。
	Invalid gateway ・ゲートウェイ IP アドレスのフォーマットまたは値が不正です。
	Default-channel cannot be set for this command. ・デフォルトチャネルは指定できません。
	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。

API	エラーメッセージ
スタティック経路追加 （続き）	Target IP address and gateway is not same IP version. ・宛先 IP アドレスとゲートウェイ IP アドレスのバージョンが一致しません。
	Maximum number of route was exceeded. ・スタティック経路の最大登録件数を超えました。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
スタティック経路削除	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。
	Invalid gateway ・ゲートウェイ IP アドレスのフォーマットまたは値が不正です。
	Route info is not found. ・指定スタティック経路が存在しません。
	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Target IP address and gateway is not same IP version. ・宛先 IP アドレスとゲートウェイ IP アドレスのバージョンが一致しません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
スタティック経路 情報表示	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。 ・指定したプレフィックス長の値が不正です。
	Invalid gateway ・ゲートウェイ IP アドレスのフォーマットまたは値が不正です。
	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Target IP address and gateway is not same IP version. ・宛先 IP アドレスとゲートウェイ IP アドレスのバージョンが一致しません。

API	エラーメッセージ
スタティック経路 情報表示 （続き）	Route is not configured. ・指定チャネルに経路が設定されていません。
	Next route is not exist. ・次スタティック経路情報が存在しません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
OpenFlow コントローラ 追加	Specified IP address already used. ・指定の IP アドレスはすでに使われています。
	The format of value of the specified IP address is invalid. ・IP address の指定が不正です。
	Specified TCP port number is invalid.(Valid from 1 to 65535) ・TCP ポート番号の指定が不正です。
	Maximum number of openflow controller was exceeded. ・OpenFlow コントローラの最大登録件数を超えました。
	System busy: Another conflicting command is in progress. ・OpenFlow コマンドの実行中です。
	System busy: Please try again later. ・OpenFlow コマンドがタイムアウトしました。
	OpenFlow function is not licensed. ・OpenFlow 機能のライセンスがありません。

API	エラーメッセージ
OpenFlow コントローラ 削除	Specified IP address is not used. ・指定 IP アドレスが存在しません。
	The format of value of the specified IP address is invalid. ・IP address の指定が不正です。
	System busy: Another conflicting command is in progress. ・OpenFlow コマンドの実行中です。
	System busy: Please try again later. ・OpenFlow コマンドがタイムアウトしました。
	OpenFlow function is not licensed. ・OpenFlow 機能のライセンスがありません。

API	エラーメッセージ
OpenFlow コントローラ情報表示	No OpenFlow controller is set. ・OpenFlow コントローラが登録されていません。
	System busy: Another conflicting command is in progress. ・OpenFlow コマンドの実行中です。
	OpenFlow function is not licensed. ・OpenFlow 機能のライセンスがありません。

API	エラーメッセージ
コンフィギュレーション保存	configuration save is in progress. ・コンフィギュレーション保存中です。
コンフィギュレーション情報取得	なし

（空白ページ）

第16章　OpenFlow 機能

ここでは，OpenFlow 機能について説明します。

16.1　概要 16-2
16.2　OpenFlow バージョン 16-3
16.3　OpenFlow 対応メッセージ..... 16-4
16.4　CLI コマンド対応 OpenFlow メッセージ 16-6
16.5　JSON 形式..... 16-7
16.6　対応コマンド一覧..... 16-8
16.7　共通エラーメッセージ 16-9
16.8　エラーメッセージ一覧 16-10

16.1 概要

OpenFlow 機能は，本装置のトラフィックコントロール機能の設定を行う際に，OpenFlow プロトコルを使用して設定を行う機能です。本装置は，外部に設置した OpenFlow コントローラから JSON（JavaScript Object Notation：RFC4627）形式で設定を行うことができます。

クラウド環境において，クラウドサーバの構成変更に連動して手動でトラフィックコントロール設定を更新することは困難になってきています。クラウド上の OpenFlow コントローラ上で OpenFlow プロトコルを利用し，クラウドサーバの構成変更に連動して本装置のトラフィックコントロール設定を更新するユーザプログラムを作成することにより，本装置の設定更新を自動化することができます。

図 16.1-1　OpenFlow 機能による装置管理

OpenFlow にはプロアクティブ型とリアクティブ型のプロトコル方式があります。本装置では，プロアクティブ型の方式を採用しています。

- プロアクティブ型
  OpenFlow コントローラから本装置に対して，シナリオ・フィルタ等の設定を事前に設定する方式
- リアクティブ型
  未知のパケットを受信したとき，Packet-In メッセージを使用して OpenFlow コントローラに問い合わせる方式

また，本装置では，役割（Role）の変更は非サポートです。MASTER/SLAVE 方式はとらず常に Equal で動作します。OpenFlow は同時に 2 つのコントローラまで接続可能です。

注：

OpenFlow 機能を使用する場合は，「OpenFlow 機能ライセンス(NF7500-L131A)」を購入していただく必要があります。

16.2 OpenFlow バージョン

本装置がサポートする OpenFlow バージョンは v1.3.4 に準拠します。

v1.3 以外のバージョンを使用した場合は，接続が切断されます。また，v1.3 であれば接続は可能ですが，本装置が準拠している v1.3.4 のみをサポートします。

表 16.2-1　OpenFlow 対応バージョン

OpenFlow バージョン	備考
1.0	非対応
1.1	非対応
1.2	非対応
1.3.0	非対応
1.3.1	非対応
1.3.2	非対応
1.4.0	非対応
1.3.3	非対応
1.3.4	対応
1.3.5	非対応
1.4.1	非対応
1.5.1	非対応

16.3 OpenFlow 対応メッセージ

本装置が対応する OpenFlow メッセージは以下のとおりです。

表 16.3-1　OpenFlow 対応メッセージ

メッセージ(type)	用途	サポート
OFPT_HELLO (0)	バージョン情報等の交換に使用します。	○
OFPT_ERROR (1)	エラーを伝達します。	○
OFPT_ECHO_REQUEST (2)	エコーリクエストです。	○
OFPT_ECHO_REPLY (3)	エコーリプライです。	○
OFPT_EXPERIMENTER (4)	Experimenter メッセージのデータ部に JSON 形式の入力をサポートします。本装置の設定コマンドが実行できます。	○
OFPT_FEATURES_REQUEST (5)	データパス情報等の交換に使用します。	○
OFPT_FEATURES_REPLY (6)	データパス情報等の交換に使用します。	○
OFPT_GET_CONFIG_REQUEST (7)	OpenFlow のコンフィグ情報をリクエストします。	○
OFPT_GET_CONFIG_REPLY (8)	OpenFlow のコンフィグ情報のリクエストに対する応答をします。	○
OFPT_SET_CONFIG (9)	OpenFlow のコンフィグを設定します。	○
OFPT_PACKET_IN (10)	－	×
OFPT_FLOW_REMOVED (11)	－	×
OFPT_PORT_STATUS (12)	－	×
OFPT_PACKET_OUT (13)	－	×
OFPT_FLOW_MOD (14)	Flow Mod メッセージのコマンド部でフローの追加・削除を，Match 部でフローマッチ条件を指定し，アクションフィールドの Experimenter データ部で JSON 形式の入力をサポートします。	○
OFPT_GROUP_MOD (15)	－	×
OFPT_PORT_MOD (16)	－	×
OFPT_TABLE_MOD (17)	－	×
OFPT_MULTIPART_REQUEST (18)	各種情報の取得のリクエストです。 また，Multipart タイプが OFPMP_EXPERIMENTER(0xffff)の場合 Experimenter データ部に JSON 形式の入力をサポートします。	○
OFPT_MULTIPART_REPLY (19)	各種情報の取得のリクエストです。	○
OFPT_BARRIER_REQUEST (20)	メッセージ実行完了通知リクエストです。	○
OFPT_BARRIER_REPLY (21)	メッセージ実行完了通知リプライです。	○
OFPT_QUEUE_GET_CONFIG_REQUEST (22)	－	×
OFPT_QUEUE_GET_CONFIG_REPLY (23)	－	×

メッセージ（type）	用途	サポート
OFPT_ROLE_REQUEST (24)	OpenFlow コントローラの役割通知リクエストです。	○
OFPT_ROLE_REPLY (25)	OpenFlow コントローラの役割通知リプライです。	○
OFPT_GET_ASYNC_REQUEST (26)	－	×
OFPT_GET_ASYNC_REPLY (27)	－	×
OFPT_SET_ASYNC (28)	－	×
OFPT_METER_MOD (29)	－	×

※ 非サポートメッセージを受信した場合，エラーメッセージ（OFPT_ERROR）を送信します。

なお，各メッセージの詳細は「付録 I　OpenFlow メッセージ詳細」を参照してください。また，本装置にコマンド投入が可能な OpenFlow 対応メッセージについては「付録 H　CLI コマンド対応 OpenFlow メッセージ詳細」を参照してください。

16.4 CLI コマンド対応 OpenFlow メッセージ

OpenFlow プロトコルを使用して本装置にコマンド入力が可能な OpenFlow メッセージは，以下のとおりです。

表 16.4-1　CLI コマンド対応 OpenFlow メッセージ

メッセージ	用途
OFPT_EXPERIMENTER	OFPT_EXPERIMENTER メッセージのデータ部に JSON 形式のデータを入力してください。
OFPT_MULTIPART_REQUEST	OFPT_MULTIPART_REQUEST メッセージの multipart type が OFPMP_EXPERIMENTER の場合のデータ部に JSON 形式のデータを入力してください。
OFPT_FLOW_MOD	OFPT_FLOW_MOD メッセージのコマンド部でフローの追加・削除を，Match 部でフローマッチ条件を指定し，アクションフィールドの Experimenter データ部に JSON 形式のデータを入力してください。

なお，詳細は「付録 H　CLI コマンド対応 OpenFlow メッセージ詳細」を参照してください。

16.5 JSON 形式

本装置のコマンドに対応した OpenFlow 機能は JSON 形式のデータを利用します。JSON とはデータを表現するためのデータ記述言語です。JSON の記述方法では，パラメータのキーと値の組をコロン":"でペアにします。パラメータが複数ある場合はコンマ"，"で区切ります。これらの全体を中括弧"{"および"}"で括ります。

キーや値はすべて文字列で記述してください。コマンドの種別を示すキー"command"と，CLI コマンドのパラメータを指定します。OpenFlow においてはキーの記述順序は順不同です。CLI コマンドのパラメータ順序と合わせる必要はありません。

下記にシナリオ追加の JSON 記述例を示します。

```
{
        " command": "add scenario" "
        "scenario_name": "/port1/North",
        "action": "aggregate",
        "min_bandwidth": "5M",
        "peak_bandwidth": "8M",
        "class": "2",
        "bufsize": "512k"
}
```

JSON の記述方法の詳細については「付録 E JSON の記述方法」を参照してください。

16.6 対応コマンド一覧

OpenFlow は，シナリオ，フィルタ，ルールリストに関する設定，および情報取得のコマンドを提供します。それぞれの機能は，相当する CLI コマンドと同等です。指定するパラメータ，値の範囲や省略可能／不可についても同等です。詳細については「付録 H　CLI コマンド対応 OpenFlow メッセージ詳細」を参照してください。

表 16.6-1　対応コマンド一覧

対象	操作	相当する CLI コマンド
シナリオ	追加	add scenario
	更新	update scenario
	削除	delete scenario
	情報取得	show scenario
	カウンタ取得	show scenario counter
アプリケーション高速化	追加	add apl-accel
	更新	update apl-accel
	削除	delete apl-accel
フィルタ	モード設定	set filter mode
	追加	add filter
	削除	delete filter
	情報取得	show filter
ルールリスト	グループ追加	add rulelist group
	グループ削除	delete rulelist group
	エントリ追加	add rulelist entry
	エントリ削除	delete rulelist entry
	情報取得	show rulelist
チャネル	追加	add channel
	削除	delete channel
	情報取得	show channel
チャネルインタフェース	設定	set ip channel
	解除	unset ip channel
	情報取得	show ip channel
チャネルインタフェースのスタティック経路	追加	add route
	削除	delete route
	情報取得	show route target
トラフィックアクセラレーションバイパス	有効設定	set wan-accel bypass status
	リカバリタイム設定	set wan-accel bypass recoverytime
	強制設定	switch wan-accel bypass force

16.7 共通エラーメッセージ

JSON フォーマットは正しいが，指定内容が不正な場合，エラーメッセージを返します。共通エラーメッセージは以下のとおりです。

表 16.7-1　共通エラーメッセージ

エラーメッセージ	説明
Specified command is invalid.	コマンドが不正です。 指定した JSON 形式のキーと値が正しいか確認してください。
Required parameter is not specified.	必須のパラメータが指定されていません。 指定した JSON 形式のキーと値が正しいか確認してください。
Failed to create pipe.	内部通信用の PIPE 作成でエラーが発生しました。 時間をおいて再度実行してください。
No response message from LR.	内部ソフトウェアからの応答がありません。 時間をおいて再度実行してください。
System busy.	システムビジー状態です。 時間をおいて再度実行してください。

16.8 エラーメッセージ一覧

各 API 個別のエラーメッセージは以下のとおりです。

表 16.8-1　API 個別エラーメッセージ一覧

API	エラーメッセージ
シナリオ追加	Specified scenario class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・class の指定が不正です。
	Specified scenario fail action class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・Fail Action class の指定が不正です。
	Specified minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Minimum Bandwidth の指定が不正です。
	Specified peak bandwidth is invalid. (Valid from 10k to 1G) ・Peak Bandwidth の指定が不正です。
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Fail Action Minimum Bandwidth の指定が不正です。
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 1G) ・Fail Action Peak Bandwidth の指定が不正です。
	Peak Bandwidth should be greater than minimum bandwidth. ・peak_bandwidth は min_bandwidth 以上に設定する必要があります。
	Specified buff size is invalid. (Valid from 2k to 100M) ・bufsize の指定が不正です。
	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is already used. ・指定のシナリオ名はすでに別のシナリオで使われています。
	Specified scenario of upper level hierarchy is not found. ・上位階層のシナリオが存在しません。
	maximum number of scenario was exceeded. ・シナリオの最大登録件数を超えました。
	Specified scenario ID is invalid. (Valid from 1 to 4096) ・シナリオインデックスが範囲外です。
	Specified scenario ID is already used. ・指定のシナリオインデックスはすでに別のシナリオで使われています。
	Specified max Q num is not licensed. (Valid from 1 to 2048) ・maxquenum が範囲外です。
	Specified max Q num is invalid. (Valid from 1 to 4096) ・maxquenum が範囲外です。

API	エラーメッセージ
シナリオ追加(続き)	Extended number of scenario is not licensed. ・シナリオ拡張ライセンスの制限数を超えてシナリオを登録することはできません。 ・シナリオ拡張ライセンスの制限数を超えた maxquenum を設定することはできません。
	Specified Q division field is invalid. Valid fields: default, vid, cos, inner-vid, inner-cos, ethertype, sip, dip, tos, proto, sport, dport ・quedivision のフィールド指定が不正です。
	failaction is not specified. ・failaction を指定せずに fail_min_bw, fail_peak_bw, fail_class を設定することはできません。
	Specified failaction is invalid. ・fail_min_bw,fail_peak_bw, fail_class は failaction として forwardattribute を指定した場合のみ設定可能です。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Peer IP version and second-peer IP version are different. ・peer と second-peer の IP バージョンは一致させる必要があります。
	Peer and second-peer are same IP address. ・peer と second-peer には異なる IP アドレスを設定する必要があります。
	Specified dport is invalid. (Valid from 10001 to 20000) ・dport の指定が不正です。
	Specified Dport is already used. ・指定の dport はすでに別のシナリオで使われています。
	Specified vid is invalid. (Valid from 1 to 4094) ・VLAN ID の指定が不正です。
	Specified inner-vid is invalid. (Valid from 1 to 4094) ・Inner-VLAN ID の指定が不正です。
	VID must be specified when inner-VID is specified. ・Inner VLAN ID は VLAN ID を指定した場合のみ指定できます。
	Specified cos is invalid. (Valid from 0 to 7) ・CoS 値の指定が不正です。
	Specified inner-cos is invalid. (Valid from 0 to 7) ・Inner-CoS 値の指定が不正です。
	VID must be specified when CoS is specified. ・CoS 値は VLAN ID を指定した場合のみ指定できます。
	Inner-VID must be specified when inner-cos is specified. ・Inner CoS 値は Inner VLAN ID を指定した場合のみ指定できます。
	Specified dscp is invalid. (Valid from 0 to 63) ・DSCP 値の指定が不正です。

API	エラーメッセージ
シナリオ追加（続き）	Specified tcp-mem is invalid. (Valid from 64k to 200M) ・TCP のバッファサイズの指定が不正です。
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) ・自動バイパスの RTT しきい値の指定が不正です。
	Specified peak bandwidth is not licensed. ・指定した帯域幅のライセンスがありません。
	Data block size should be divided by fec block size. ・データブロックサイズは FEC ブロックサイズで割り切れる値に設定する必要があります。
	Data block size should be greater than fec block size. ・データブロックサイズは FEC ブロックサイズより大きな値に設定する必要があります。
	Specified fec block size is invalid. (Valid from 2K to 50K) ・FEC ブロックサイズの指定が不正です。
	Specified data block size is invalid. (Valid from 2K to 200K) ・データブロックサイズの指定が不正です。
	Specified fec session is invalid. (Valid from 0 to 1000) ・FEC セッション数の指定が不正です。
	FEC function is not licensed. ・TCP-FEC 機能のライセンスがありません。
	Maximum number of secondary peer was exceeded. ・second-peer を指定したシナリオの最大登録件数を超えました。
	Maximum number of keep alive scenario was exceeded. ・bypass-keep を有効に指定したシナリオの最大登録件数を超えました。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
シナリオ更新	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・class の指定が不正です。
	Specified scenario fail action class is invalid. It must be either of 1,2,3,4,5,6,7,8. ・Fail Action class の指定が不正です。
	Specified minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Minimum Bandwidth の指定が不正です。

API	エラーメッセージ
シナリオ更新（続き）	Specified fail action minimum bandwidth is invalid. (Valid from 0, 1k to 1G) ・Fail Action Minimum Bandwidth の指定が不正です。
	Specified peak bandwidth is invalid. (Valid from 10k to 1G) ・Peak Bandwidth の指定が不正です。
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 1G) ・Fail Action Peak Bandwidth の指定が不正です。
	Peak bandwidth should be greater than minimum bandwidth. ・peak_bandwidth は min_bandwidth 以上に設定する必要があります。
	Specified buff size is invalid. (Valid from 2k to 100M) ・bufsize の指定が不正です。
	It is necessary to set one or more parameters. ・1 つ以上のパラメータを設定する必要があります。
	Specified scenario mode is invalid. ・シナリオモードの指定が不正です。
	Specified max Q num is not licensed. (Valid from 1 to 2048) ・maxquenum が範囲外です。
	Specified max Q num is invalid. (Valid from 1 to 4096) ・maxquenum が範囲外です。
	Specified Q Division division Field field is invalid. Valid fields: default, vlanvid, cos, inner-vlanid, inner-cos, ethertype, sip, dip, tos, proto, sport, dport (multiple fields can be specified with separated comma without space) ・quedivision のフィールド指定が不正です。
	Fail action forward is incorrect.Specified Failaction is invalid. ・fail_min_bw , fail_peak_bw , fail_class は failaction として forwardattribute を指定した場合のみ設定可能です。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Specified cos is invalid. (Valid from 0 to 7) ・CoS 値の指定が不正です。
	Specified inner-cos is invalid. (Valid from 0 to 7) ・Inner-CoS 値の指定が不正です。
	Specified dscp is invalid. (Valid from 0 to 63) ・DSCP 値の指定が不正です。
	Specified tcp-mem is invalid. (Valid from 64k to 200M) ・TCP のバッファサイズの指定が不正です。
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) ・トラフィックアクセラレーションの自動バイパスの RTT しきい値の指定が不正です。
	Specified peak bandwidth is not licensed. ・指定した帯域幅のライセンスがありません。

API	エラーメッセージ
シナリオ更新（続き）	Data block size should be divided by fec block size. ・データブロックサイズは FEC ブロックサイズで割り切れる値に設定する必要があります。
	Data block size should be greater than fec block size. ・データブロックサイズは FEC ブロックサイズより大きな値に設定する必要があります。
	Specified fec block size is invalid. (Valid from 2K to 50K) ・FEC ブロックサイズの指定が不正です。
	Specified data block size is invalid. (Valid from 2K to 200K) ・データブロックサイズの指定が不正です。
	Specified fec session is invalid. (Valid from 0 to 1000) ・FEC セッション数の指定が不正です。
	FEC function is not licensed. ・TCP-FEC 機能のライセンスがありません。
	Maximum number of keep alive scenario was exceeded. ・bypass-keep を有効に指定したシナリオの最大登録件数を超えました。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
シナリオ削除	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Down level hierarchy scenario exists. ・下位階層のシナリオが存在します。

API	エラーメッセージ
シナリオ情報取得	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Next scenario is not exist. ・next シナリオが存在しません。

API	エラーメッセージ
シナリオカウンタ情報取得	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Next scenario is not exist. ・next シナリオが存在しません。

API	エラーメッセージ
アプリケーション高速化追加	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario name is not wan-accel mode. ・指定シナリオがアクセラレーションモードではありません。
	Specified protocol is already used. ・指定シナリオはすでに使用されています。
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) ・SMB TCP Port の指定が不正です。
	Specified smb session is invalid. (Valid from 0 to 1000) ・SMB Session の指定が不正です。
	Specified read cache size is invalid. (Valid from 64k to 60M) ・Read Cache Size の指定が不正です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
アプリケーション高速化更新	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario name is not wan-accel mode. ・指定シナリオがアクセラレーションモードではありません。
	Specified protocol is not used. ・指定プロトコルは使用されていません。
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) ・SMB TCP Port の指定が不正です。
	Specified smb session is invalid. (Valid from 0 to 1000) ・SMB Session の指定が不正です。

API	エラーメッセージ
アプリケーション高速化更新（続き）	Specified read cache size is invalid. (Valid from 64k to 60M) ・Read Cache Size の指定が不正です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
アプリケーション高速化削除	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified scenario name is not wan-accel mode. ・指定シナリオがアクセラレーションモードではありません。
	Specified protocol is already disabled. 指定されたプロトコルはすでに無効です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
フィルタモード設定	Slot #N is invalid. ・スロット指定が不正です。
	Port <slot/port> is invalid. ・ポート指定が不正です。
	Specified field is invalid. Valid fields: default, vid, cos, inner-vid, inner-cos, sip, dip, tos, proto, sport, dport (multiple fields can be specified with separated comma without space) ・フローを識別するフィールド名の指定が不正です。

API	エラーメッセージ
フィルタ追加	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid filter name length is from 1 to 48.) ・フィルタ名の指定が不正です。

API	エラーメッセージ
フィルタ追加（続き）	Specified filter Name is already used. ・指定のフィルタ名はすでに別のフィルタで使われています。
	Specified ether type is invalid. (Valid from 0x0000 to 0xFFFF) ・Ether type の指定が不正です。
	Specified vid is invalid. (Valid from 0 to 4094, Or Start - End) ・VLAN ID の指定が不正です。
	Specified cos is invalid. (Valid from 0 to 7, Or Start - End) ・CoS 値の指定が不正です。
	Specified inner-vid is invalid. (Valid from 0 to 4094, Or Start - End) ・Inner-VLAN ID の指定が不正です。
	VID must be specified when inner-VID is specified. ・Inner VLAN ID は VLAN ID を指定した場合のみ指定できます。
	Specified inner-cos is invalid. (Valid from 0 to 7, Or Start - End) ・Inner-CoS 値の指定が不正です。
	The format or value of the specified source IP address is invalid. ・Source IP address の指定が不正です。
	The format or value of the specified destination IP address is invalid. ・Destination IP address の指定が不正です。
	The format or value of the specified source IPv6 address is invalid. ・Source IPv6 address の指定が不正です。
	The format or value of the specified destination IPv6 address is invalid. ・Destination IPv6 address の指定が不正です。
	Specified rulelist name of source IP address is invalid. Specified rulelist name of destination IP address is invalid. Specified rulelist name of source port is invalid. Specified rulelist name of destination port is invalid. ・ルールリスト名が不正です。
	Specified rulelist name of source IP address is not used. Specified rulelist name of destination IP address is not used. Specified rulelist name of source port is not used. Specified rulelist name of destination port is not used. ・指定ルールリストが存在しません。
	IP Filter and rulelist of source IP address is not same type. IP Filter and rulelist of destination IP address is not same type. IP Filter and rulelist of source port is not same type. IP Filter and rulelist of destination port is not same type. ・対象ルールリストと種別が異なります。
	Specified tos is invalid. (Valid from 0 to 255, Or Start - End) ・ToS 値の指定が不正です。
	Specified protocol number is invalid. (Valid from 0 to 255, Start - End, Or tcp/udp/icmp/icmpv6) ・プロトコル番号の指定が不正です。

API	エラーメッセージ
フィルタ追加(続き)	Specified source TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・sport 番号の指定が不正です。
	Specified destination TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・dport 番号の指定が不正です。
	Specified filter priority is invalid. (Valid from 1 to 40000) ・フィルタ優先度の指定が不正です。
	maximum number of filter was exceeded. ・フィルタの最大登録件数を超えました。
	It is necessary to set one or more parameters other than Priority. ・Ethernet フィルタは Priority 以外で少なくとも 1 つのパラメータを指定する必要があります。
	Filter type is different. Please specify same type of wan-accel scenario. ・wan-accel シナリオの peer と同じ IP バージョンを指定してください。

API	エラーメッセージ
フィルタ削除	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid filter name length is from 1 to 48.) ・フィルタ名の指定が不正です。
	Specified filter name is not used. ・指定フィルタが存在しません。

API	エラーメッセージ
フィルタ情報取得	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid filter name length is from 1 to 48.) ・フィルタ名の指定が不正です。

API	エラーメッセージ
フィルタ情報取得（続き）	Specified filter name is not used. ・指定フィルタが存在しません。
	Next filter is not exist. ・次のフィルタが存在しません。

API	エラーメッセージ
ルールリストグループ追加	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is already in use. ・同一名のルールリストがすでに存在します。
	Maximum number of rulelist was exceeded. ・ルールリストの最大登録件数を超えました。

API	エラーメッセージ
ルールリストグループ削除	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。
	Rulelist is used by filter. ・ルールリストがフィルタに設定されています。

API	エラーメッセージ
ルールリストエントリ追加	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。
	The format or value of the specified IP address is invalid. ・IP address の指定が不正です。
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・TCP/UDP ポート番号の指定が不正です。

API	エラーメッセージ
ルールリストエントリ追加（続き）	Maximum number of rulelist entry was exceeded. ・指定ルールリストのルールリストエントリ最大登録件数（512 件）を超えました。
	Maximum number of total rulelist entry was exceeded. ・全ルールリスト合計のルールリストエントリ最大登録件数（64000 件）を超えました。
	Specified rulelist entry is already in use. ・指定ルールリストエントリはすでに登録されています。
	Rulelist entry and rulelist is not same type. ・対象ルールリストと種類が異なります。

API	エラーメッセージ
ルールリストエントリ削除	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。
	The format or value of the specified IP address is invalid. ・IP address の指定が不正です。
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) ・TCP/UDP ポート番号の指定が不正です。
	Rulelist entry and rulelist is not same type. ・対象ルールリストと種別が異なります。
	Specified rulelist entry is not used. ・指定ルールリストエントリが存在しません。

API	エラーメッセージ
ルールリスト情報取得	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specfied.) (Valid rulename length is from 1 to 32.) ・ルールリスト名が不正です。
	Specified rulelist name is not used. ・指定ルールリストが存在しません。

API	エラーメッセージ
チャネル追加	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Channel name already exists. ・指定のチャネル名はすでに別のチャネルで使われています。
	Slot #N is invalid. ・スロット指定が不正です。
	Port <slot/port> is invalid. ・ポート指定が不正です。
	Specified vid is invalid. (Valid from 1 to 4094) ・VLAN ID の指定が不正です。
	Specified TPID is invalid. (Valid 0x8100,0x88a8,0x9100,0x9200 or 0x9300.) ・TPID の指定が不正です。
	Specified inner-vid is invalid. (Valid from 1 to 4094) ・Inner-VLAN ID の指定が不正です。
	VID must be specified when inner-VID is specified. ・Inner VLAN ID は VLAN ID を指定した場合のみ指定できます。
	Specified mtu is invalid. (Valid from 300 to 10200) ・mtu の指定が不正です。
	Specified vid and inner-vid is already used on channel “channel name”. ・指定された vid と inner-vid はすでに“channel name”チャネルで使われています。
	Specified port is already used on other default-channel. ・指定のポートはすでに別のデフォルトチャネルで使われています。
	Maximum number of channel was exceeded. ・チャネルの最大登録件数を超えました。

API	エラーメッセージ
チャネル削除	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。

API	エラーメッセージ
チャネル情報取得	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Next channel is not exist. ・次チャネルが存在しません。

API	エラーメッセージ
インタフェース設定	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。
	Default-channel cannot be set for this command. ・デフォルトチャネルは設定できません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
インタフェース解除	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Cannot specified "ipv4" or "ipv6". ・“all”に対して“IPv4”と“IPv6”は指定できません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
インタフェース情報表示	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Next IP channel is not exist. ・次チャネルインタフェースが存在しません。
	IP interface is not configured. ・指定チャネルまたは次チャネルに IP アドレスが設定されていません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
スタティック経路追加	Route entry already exists. ・すでに存在するルートエントリです。
	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。 ・指定したプレフィックス長の値が不正です。
	Invalid gateway ・ゲートウェイ IP アドレスのフォーマットまたは値が不正です。
	Default-channel cannot be set for this command. ・デフォルトチャネルは指定できません。
	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Target IP address and gateway is not same IP version. ・宛先 IP アドレスとゲートウェイ IP アドレスのバージョンが一致しません。
	Maximum number of route was exceeded. ・スタティック経路の最大登録件数を超えました。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
スタティック経路削除	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。
	Invalid gateway ・ゲートウェイ IP アドレスのフォーマットまたは値が不正です。
	Route info is not found. ・指定スタティック経路が存在しません。
	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Specified channel name is not used. ・指定チャネルが存在しません。
	Target IP address and gateway is not same IP version. ・宛先 IP アドレスとゲートウェイ IP アドレスのバージョンが一致しません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
スタティック経路情報表示	Invalid IP address ・指定した IP アドレスのフォーマットまたは値が不正です。
	Invalid netmask ・指定したサブネットマスクのフォーマットまたは値が不正です。 ・指定したプレフィックス長の値が不正です。
	Invalid gateway ・ゲートウェイ IP アドレスのフォーマットまたは値が不正です。
	Specified channel name is invalid. ・チャネル名の指定が不正です。
	Target IP address and gateway is not same IP version. ・宛先 IP アドレスとゲートウェイ IP アドレスのバージョンが一致しません。
	Route is not configured. ・指定チャネルに経路が設定されていません。
	Next route is not exist. ・次スタティック経路情報が存在しません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
トラフィックアクセラレーションバイパス有効設定	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
トラフィックアクセラレーションバイパスリカバリタイム設定	Duration is valid from 1 to 600 ・バイパス回復時間が範囲外です。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

API	エラーメッセージ
トラフィックアクセラレーション強制バイパス有効設定	Specified scenario name is invalid. ・シナリオ名の指定が不正です。
	Specified scenario name is not used. ・指定シナリオが存在しません。
	Scenario type is different. Please specify a wan-accel scenario. ・指定したシナリオがアクセラレーションモードシナリオではありません。
	TCP Acceleration Function is not licensed. ・TCP 高速化機能ライセンスがありません。

第17章　ネットワークバイパス機能

ここでは，ネットワークバイパス機能と設定について説明します。

17.1　概要 17-2
17.2　設定と確認方法 17-3
17.3　注意事項 17-6

17.1 概要

本装置は Network ポートのバイパス機能（ネットワークバイパス機能）を実装しています。装置異常発生時に Network ポートをバイパスして，通信経路を確保することが可能です。

図 17.1-1　バイパス動作時の通信経路

Network ポートがバイパスされると，本装置はネットワークから切り離された状態となり，トラフィックコントロールは機能しません。バイパス状態では，クロスケーブルとして動作するため，対向装置を直結した場合と同じになります。
バイパス状態へ切り替わった際，対向装置の接続ポートが一時的にリンクダウン状態となりますが，対向装置間で再度リンクが確立され通信が再開されます。

注：

本機能は，Network ポート 1/1 と 1/2 のメディアタイプを両方とも RJ-45 に選択しているときのみ動作します。Network ポートのメディアタイプは，“set port media-type”コマンドで選択することができます。

17.2 設定と確認方法

本装置の動作中に自動または任意のバイパス操作を行うことができます。

バイパス操作は以下のコマンドで行います。

表 17.2-1　ネットワークバイパス機能 CLI コマンド一覧

set bypass {auto | on | off}	ネットワークバイパス機能の制御モードを設定します。 auto を指定すると，装置異常検出時の自動バイパス制御が有効になります。 on を指定すると，強制的にバイパス状態になります。 off を指定すると，強制的に非バイパス状態になります。 デフォルトは auto です。
bypass time <time> {on | off}	一時的にネットワークバイパスの切り替えを行います。 on を指定すると，強制的にバイパス状態に切り替え，time 秒経過後に自動的に以前の状態に戻します。 off を指定すると，強制的に非バイパス状態に切り替え，time 秒経過後に自動的に以前の状態に戻します。 本コマンドを実行すると，現在時刻およびタイマの満了時刻が表示されます。 注) 本コマンドは，save config コマンドによる保存はできません。
show bypass	ネットワークバイパス機能の設定および状態を表示します。

Network ポートを強制的にバイパス状態にする場合，以下に示すコマンドを実行します。

システムインタフェースの通信ポートを Ethernet ポートに設定している場合

```
PureFlow(A)> set bypass on
PureFlow(A)>
```

システムインタフェースの通信ポートを Network ポートに設定している場合

```
PureFlow(A)> set bypass on
System interface might be disconnected from the network, ok (y/n)? y
Done
PureFlow(A)>
```

Network ポートを一時的に 300 秒間バイパス状態にする場合，以下に示すコマンドを実行します。

システムインタフェースの通信ポートを Ethernet ポートに設定している場合

```
PureFlow(A)> bypass time 300 on
Current time : Feb 29    17:38:47
Expiring time: Feb 29    17:43:47
PureFlow(A)>
```

システムインタフェースの通信ポートを Network ポートに設定している場合

```
PureFlow(A)> bypass time 300 on
System interface might be disconnected from the network, ok (y/n)? y
Current time : Feb 29   17:38:47
Expiring time: Feb 29   17:43:47
Done
PureFlow(A)>
```

300 秒を待たずに実行前の状態に戻したい場合は，短い時間（1 秒等）で再度設定し直してください。

設定コマンドで設定した内容や，現在の Network ポートのバイパス状態を確認するには，“show bypass”コマンドを使用します。

```
PureFlow(A)> show bypass
Control mode        : auto
Bypass state        : off
Timer remaining     : 12[s]
PureFlow(A)>
```

自動バイパス制御を有効にすることで，装置異常が発生した際，ネットワークの停止を回避することができます。“set bypass”コマンドで auto を指定している場合，以下のタイミングでバイパス状態となります。

- 装置起動完了時
  フォワーディング系処理部の起動異常時にバイパス状態となります。正常に起動した場合は非バイパス状態となります。

- 装置異常検出時
  フォワーディング系処理部の異常検出時および制御系処理部の停止異常のような重度のエラー発生時にバイパス状態となります。

- “reboot system”コマンド実行時
  再起動前にバイパス状態となります。

- 電源切断時
  電源が遮断されたときバイパス状態となります。

注:

“set bypass auto”コマンドによる自動バイパス制御は，上記条件発生時のみ作動します。
本コマンドを実行しても，上記条件が発生しなければ，バイパス状態は変化しません。
コマンドでバイパス操作を行った後，auto 設定で運用する場合は，“set bypass off”コマンドで非バイパス状態にしてから，“set bypass auto”コマンドを実行し，運用を開始してください。バイパス状態で“set bypass auto”コマンドを実行しても，自動的に非バイパス状態にはなりません。

制御系処理部での重度のエラー発生時のバイパス操作については syslog への記録は行われませんが，ほかのバイパス操作はいずれも syslog へ記録されます。記録される syslog メッセージは次のとおりです。
なお，自動バイパス制御が作動した場合，その原因については下記 syslog メッセージの直前に記録されたメッセージを参照してください。

- バイパス状態に変化
  Bypass state was changed to on

- 非バイパス状態に変化
  Bypass state was changed to off

17.3 注意事項

ネットワークバイパス機能を使用する際は，下記事項に注意してください。

バイパス状態では，本装置はクロスケーブルとして動作します。本装置と対向装置を接続するケーブルの種類（クロス／ストレート），長さは，バイパス状態／非ハイパス状態のいずれでも通信できるように「PureFlow WS1 トラフィックシェーパー NF7500 シリーズ 取扱説明書」を参照し正しく選定してください。

ネットワークバイパス操作時は対向装置のポートは一度リンクダウンし，数秒後リンクが再確立します。
再確立するまでの時間は接続装置の特性により異なります。実運用前に確認することを推奨します。

バイパス状態での本装置の Network ポートはリンクダウン状態となり，Link LED は消灯します。このため，ネットワークバイパス操作時に SNMP のリンク変化トラップやリンク変化 syslog が送出されます。ただし，装置異常検出時のバイパス操作ではリンク変化が検出されない場合があります。

Network ポート経由で装置管理を行っている場合，バイパス接続状態では本装置のリモート管理ができなくなります。バイパス接続状態においてもリモート管理を行いたい場合，Ethernet ポート経由で管理してください。

第18章　トップカウンタ機能

ここでは，トップカウンタ機能について説明します。

18.1　概要 18-2
18.2　トップカウンタの表示単位について 18-2
18.3　トップカウンタの測定範囲について 18-3
18.4　トラフィックカウンタについて 18-4
18.5　アプリケーションポート番号の測定について 18-5
18.6　操作コマンド一覧 18-5
18.7　操作手順 18-6
18.8　操作例 18-7
18.9　注意事項 18-9

18.1 概要

トップカウンタ機能は，トラフィックの利用状況を把握するための機能です。この機能は，IP アドレスごとまたはアプリケーションポート番号ごとにトラフィック量を自動認識，流量測定し，トラフィック量が多い順に上位 25 位までのトラフィック量を表示します。

また，モニタリングマネージャを使用することにより，利用状況をリアルタイムにグラフ表示し，過去のデータを含めたレポートを作成することができます。詳細は「PureFlow モニタリングマネージャ2 NF7201A 取扱説明書」，「PureFlow Profiler モニタリングマネージャ3 NF7202A 取扱説明書」を参照してください。

図 18.1-1

18.2 トップカウンタの表示単位について

トップカウンタ機能は，以下の 4 種の表示単位でトラフィックを計測し，それぞれの表示単位ごとに，上位 25 位までのトラフィック量を表示します。

- 送信元 IP アドレス（SIP）
- 宛先 IP アドレス（DIP）
- 送信元 IP アドレスと宛先 IP アドレスの組（SIP_DIP）
- アプリケーションポート番号（APPLI）

18.3 トップカウンタの測定範囲について

トップカウンタ機能は，本装置を通過する全トラフィックの中から，トップカウンタを測定する範囲を指定することができます。測定範囲として，任意のシナリオを指定でき，最大で 200 個まで登録できます。

図 18.3-1　トップカウンタの測定範囲

たとえば，あるレベル n シナリオを通過するトラフィックにおいて，通信帯域をより多く消費しているトラフィックを観測する場合は，測定範囲に該当するレベル n シナリオを指定します。これにより，シナリオに入力されたトラフィックの中から，送出量が最も多いトラフィックを把握することができます。

アクセラレーションモードを使用する場合は，以下の点にご注意ください。

（1）アクセラレーションモードシナリオを通過し，アクセル対象となるトラフィック

① LAN 側ネットワークからアクセラレーションモードシナリオに入力されたトラフィックの中で，端末からの送出量が最も多いトラフィックを把握することになります。

② LAN 側ネットワークに出力するアクセル対象トラフィックについては，アクセラレーションモードシナリオではなく，ポートシナリオ，またはレベル n シナリオを通過するトラフィックの中から，送出量が最も多いトラフィックを把握することになります。

図 18.3-2　アクセラレーションモードシナリオを通過し，アクセル対象となるトラフィック

(2) アクセラレーションモードシナリオを通過するが，アクセル対象とはならないトラフィック

③ LAN 側ネットワークからアクセラレーションモードシナリオに入力されたトラフィックの中で，シナリオからの送出量が最も多いトラフィックを把握することになります。

④ LAN 側ネットワークに出力するアクセル対象トラフィックについては，アクセラレーションモードシナリオではなく，ポートシナリオ，またはレベル n シナリオを通過するトラフィックの中から，シナリオからの送出量が最も多いトラフィックを把握することになります。

図 18.3-3　アクセラレーションモードシナリオを通過するが，アクセル対象とはならないトラフィック

18.4 トラフィックカウンタについて

トラフィックカウンタは，トラフィックの IP アドレスやアプリケーションポート番号ごとなど，自動認識したトラフィックごとに自動配置され，それぞれの送信トラフィック量を測定するカウンタです。

トップカウンタ機能を使用する場合，あらかじめ，利用可能なトラフィックカウンタの最大数をそれぞれの測定範囲ごとに指定する必要があります。トラフィックカウンタの総数は，全測定対象合計で 400,000 個までです。

図 18.4-1　トラフィックカウンタ

18.5 アプリケーションポート番号の測定について

トップカウンタ機能は，特定のアプリケーションポート番号だけにトラフィックカウンタを割り当て，トラフィック量を測定します。公知のアプリケーションについては測定を実施するようにデフォルトで登録済です。デフォルト状態で測定を実施するアプリケーションポート番号は，show topcounter config all コマンドで確認してください。

また，任意のアプリケーションポート番号も測定することができます。測定したいアプリケーションポート番号を add topcounter config appli port コマンドで追加してください。

アプリケーションポート番号の測定では，任意のアプリケーションを常時監視することもできます。常時監視に指定すると，当該アプリケーションポート番号のトラフィックカウンタを固定的に確保します。また，その実際の順位が上位 25 位以内でなくても，show topcounter target コマンドの測定結果に常に表示します。常時監視したいアプリケーションポート番号は，測定範囲（シナリオ）ごとに，add topcounter config appli port static コマンドで登録してください。

18.6 操作コマンド一覧

トップカウンタ機能の操作は，以下のコマンドで行います。

表 18.6-1　操作コマンド一覧

set topcounter	トップカウンタの有効／無効を設定します。
set topcounter config interval time	トップカウンタの収集周期を設定します。
add topcounter target	トップカウンタの測定範囲を追加します。
update topcounter target	トップカウンタの測定範囲に指定されているパラメータを変更します。
delete topcounter target	トップカウンタの測定範囲を削除します。
show topcounter config	トップカウンタの設定を表示します。
show topcounter target	トップカウンタを表示します。
add topcounter config appli port	トップカウンタを測定するアプリケーションポート番号を追加します。
delete topcounter config appli port	トップカウンタを測定するアプリケーションポート番号を削除します。
add topcounter config appli port static	常時監視するアプリケーションポート番号を登録します。
delete topcounter config appli port static	常時監視するアプリケーションポート番号を削除します。

18.7 操作手順

トップカウンタ機能を使用するための操作手順は以下のとおりです。

(1) トップカウンタの測定範囲を設定する。
“add topcounter target”コマンドを使用し，トップカウンタを測定するトラフィックを指定してください。測定範囲として，任意のシナリオのトラフィックを指定することができます。

(2) 必要に応じて，トップカウンタの収集周期を設定する。
set topcounter config interval time コマンドを使用し，トップカウンタの収集周期を変更することができます。ただし，モニタリングマネージャを接続している場合，収集周期が変更される場合があります（「18.9 注意事項(2)」参照）。動作中の収集周期は，show topcounter config コマンドで確認することができます。

(3) 必要に応じて，トップカウンタで測定するアプリケーションポート番号を追加する。
デフォルト設定以外のアプリケーションポート番号を測定する場合は，add topcounter config appli port コマンドを使用し，任意のポート番号を追加することができます。デフォルト設定のポート番号は，show topcounter config all コマンドで確認することができます。

(4) 必要に応じて，常時監視するアプリケーションポート番号を登録する。
任意のアプリケーションポート番号を add topcounter config appli port static コマンドを使用し，常時監視するように登録することができます。常時監視するアプリケーションポート番号の登録は測定範囲（シナリオ）ごとに行ってください。

(5) トップカウンタの収集を有効にする。
set topcounter enable コマンドを使用し，トップカウンタ機能を有効にしてください。トップカウンタ機能が有効になってから，収集周期が経過した後，次の(6)でトップカウンタを表示します。

(6) トップカウンタを表示する。
show topcounter target コマンドを使用し，トップカウンタを表示します。送信元 IP アドレスごと，宛先 IP アドレスごと，送信元 IP アドレスと宛先 IP アドレスの組み合わせごと，アプリケーションポート番号ごとなど，それぞれのトップカウンタを表示することができます。

18.8 操作例

以下の表に示す設定で，トップカウンタ機能を使用するときのコマンド設定例を記載します。

表 18.8-1　コマンド例

ユーザ設定項目	設定値	備考
測定範囲	Network ポート 1/1 /port1	トラフィックカウンタ数を任意に設定
	レベル 2 シナリオ /port1/North	トラフィックカウンタ数をデフォルト設定
	レベル 3 シナリオ /port1/North/SiteA	トラフィックカウンタ数をデフォルト設定
収集周期	5 分	モニタリングマネージャを接続した場合，収集周期が変更される場合があります（「18.9　注意事項(2)」参照）。
アプリケーションポート番号	測定するアプリケーションポート番号を追加 10000 20000～20003	デフォルト設定のアプリケーションポート番号に加えて，10000，20000，20001，20002，20003のアプリケーションポート番号を測定する。
	常時監視するアプリケーションポート番号を登録 シナリオ　/port1 ポート番号　80	HTTP（ポート番号　80）トラフィックを常時監視する。

設定コマンドは，以下のとおりです。

```
PureFlow(A)> add topcounter target scenario /port1 sip 10000 dip 10000 sip_dip 10000
             appli 250
PureFlow(A)> add topcounter target scenario /port1/North
PureFlow(A)> add topcounter target scenario /port1/North/SiteA
PureFlow(A)> set topcounter config interval time 5
PureFlow(A)> add topcounter config appli port 10000
PureFlow(A)> add topcounter config appli port 20000-20003
PureFlow(A)> add topcounter config appli port static /port1 80
PureFlow(A)> set topcounter enable
PureFlow(A)>
```

トップカウンタは，以下のように表示されます。

```
PureFlow(A)> show topcounter target scenario /port1 group sip
From          : 2016 Dec 02 19:47:55   To              : 2016 Dec 02 19:57:55
Total Octet: 1475806000                Total Packet: 1475806

Order   IP Address                                  Tx Octet        Tx Packet
-----   ----------------------------------------    ----------------  ----------------
  1     192.168.101.121                                   8214              111
  2     192.168.101.122                                   5846               79
  3     fe80:0000:0000:0000:0290:ccff:fe22:8b4c           5772               78
  4     fe80:0000:0000:0000:0290:ccff:fe22:8b4d           5698               77
  5     fe80:0000:0000:0000:0290:ccff:fe22:8b4e           3848               52
PureFlow(A)>

PureFlow(A)> show topcounter target scenario /port1 group appli
From          : 2016 Dec 02 19:47:55   To              : 2016 Dec 02 19:57:55
Total Octet: 1475806000                Total Packet: 1475806

Order   TCP/UDP Port                 Type            Tx Octet        Tx Packet
-----   ----------------------------------  ----------------  ----------------  ----------------
  1                        10000                                  22625              276
  2                        20000                                   1288               46
  3                        20001                                    446               12
  4                        20002                                    446               12
  5                        20003                                    240               20
  6                           80   static                            0                0
PureFlow(A)>
```

18.9 注意事項

(1) トラフィックカウンタが不足した場合，正確なトップカウンタを表示しない場合があります。
割り当てたトラフィックカウンタの数よりも，実際に通信している通信ノードが多い場合，トラフィックカウンタが不足する場合があります。トラフィックカウンタが割り当てられていない通信ノードは，個別の流量を測定することができないため，トップカウンタとして表示されません。

(2) モニタリングマネージャを使用している場合，CLI で設定した収集周期とは異なる周期でトップカウンタを集計する場合があります。
本装置にモニタリングマネージャが接続された場合，トップカウンタの収集周期がモニタリングマネージャによって変更される場合があります。CLI で設定された収集周期と，モニタリングマネージャの GUI で設定された収集周期を比較し，より長いほうの周期でトップカウンタを収集します。動作中の収集周期は，“show topcounter config”コマンドで確認してください。

(3) 受信した TCP/IP パケットにおいて，送信元ポート番号と宛先ポート番号の両方が，トップカウンタを測定するアプリケーションポート番号として登録されている場合，そのパケットは，宛先ポート番号のトラフィックカウンタに計上されます。送信元ポート番号のトラフィックカウンタには計上されません。

(4) トップカウンタを測定するアプリケーションポート番号を必要に応じて追加できますが，デフォルトで設定されているアプリケーションポート番号は削除できません。

(5) CLI またはモニタリングマネージャからトップカウンタの収集周期を変更した場合，一度だけ，設定されている収集周期よりも短い期間で集計されたトップカウンタを表示する場合があります。これは，前回の収集周期に達した時刻から，収集周期を変更した時刻までのトップカウンタの集計結果です。

(6) トップカウンタは，トップカウンタの収集周期に到達してから約 1 分経過したときに更新されます。

(7) トップカウンタの収集周期が 1 分の場合は，全測定対象合計は 100,000 個までに制限されます。

(8) トップカウンタ有効時に測定範囲を追加した場合，追加した対象シナリオは，即座に測定対象にはなりません。即座に測定を行いたい場合は，トップカウンタを無効にしてから測定範囲を追加し，トップカウンタを有効にしてください。

（空白ページ）

第19章　ドメインフィルタ機能

ここでは，ドメインフィルタ機能について説明します。

19.1　概要 19-2
19.2　ドメインフィルタ機能仕様について 19-3
19.3　設定手順 19-5
19.4　確認手順 19-7
19.5　DNS レスポンスパケット装置不通過の場合の対応..... 19-9
19.6　注意事項 19-10

19.1 概要

ドメインフィルタ機能は，帯域制御のパケット分類識別子としてドメイン名を使用することができる機能です。本機能を使用することで，制御対象通信識別を「example.com」のようなドメイン名で指定することができます。その結果，IP アドレスが動的に変わるクラウドサービスのような通信でも IP アドレスを意識することなく帯域制御が可能となります。

ドメインフィルタ機能は，装置を通過する DNS レスポンスパケットにより，指定したドメイン名から IP アドレスを自動検索して学習します。その IP アドレス情報をフィルタに設定することで対象通信の帯域制御が可能となります。IP アドレスの自動検索および学習は DNS レスポンスパケットが発生するごとに行うため，対象となるサイトの IP アドレスが変更されても追従します。また一度学習した IP アドレスは一定時間保持します。

図 19.1-1　ドメインフィルタ機能概要

ドメインフィルタ機能を使用する場合は，「NF7500-L141A ドメインフィルタ機能ライセンス」が必要です。

19.2 ドメインフィルタ機能仕様について

本章ではドメインフィルタ機能に関する仕様を説明します。

(1) ドメイン名

以下に，設定可能なドメイン名の仕様を示します。

表 19.2-1 ドメイン名仕様

項目	仕様
ドメイン名の長さ	253 文字以下
ラベルの長さ	63 文字以下
使用可能文字	1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ -.* 「*」は、ワイルドカード設定時に使用 日本語ドメインは対応しません。
大文字小文字の区別	大文字と小文字の混在設定は可能ですが区別しません。 （例） 「example.com」と「EXAMPLE.com」は同じ文字列として扱います。
文字列一括設定	後方一致のみ，「*」によるワイルドカード設定が可能です。 （例） 「*.com」「*.example.com」「*example1.example2.com」
その他	トップドメインは設定必須です。 「.」「*」「*.」は設定できません。

(2)　設定可能なドメイン名の数

ドメインフィルタ機能は「ルールリスト」により設定します。まず，制御対象ドメイン名群を一括管理するためのルールリストグループを作成します。その中にエントリとして制御対象ドメイン名を設定します。そのため，各項目の設定数はルールリストの仕様に準じていますが，ドメイン名で検索，学習したIPアドレス(以下，エントリ IP アドレス)はグループあたり 512 件，全体で 10,000 件保持することが可能です。

表 19.2-2　ドメイン名／エントリ IP アドレス設定数

項目	仕様
最大グループ数	1024 件(全種別(ipv4, ipv6, l4port, domain)合計)
最大エントリ数	512 件／グループ 装置全体 10,000 件(全種別(ipv4, ipv6, l4port, domain)合計)
最大エントリ IP アドレス数	512 件／グループ 装置全体 10,000 件 学習した IP アドレスの保持時間は学習時の TTL 値＋86400(1 日)秒です。

図 19.2-1　ドメインフィルタ機能対応ルールリスト体系

なお，ルールリストについては「8.10　ルールリストの設定方法」を参照してください

19.3 設定手順

ドメインフィルタ機能を使用する際の設定はルールリストで行います。その手順を以下に記します。

（例 1）ドメイン名「example1.com」と「example2.com」向けの通信を一括して制御するため，ドメインフィルタ機能を使用する。

手順 1）ルールリストにて，対象ドメイン名を制御するためのルールリストグループ「1-site-EX」を作成します。

```
PureFlow (A)> add rulelist group 1-site-EX domain
```

手順 2）手順 1 で作成したグループ「1-site-EX」にエントリとして対象ドメイン名を登録します。

```
PureFlow(A)> add rulelist entry 1-site-EX domain example1.com
PureFlow(A)> add rulelist entry 1-site-EX domain example2.com
```

これで機能が有効となり，DNS レスポンスパケットにより対象ドメイン名から IP アドレスを自動的に検索して学習します。
帯域制御シナリオ設定後，ここで作成したルールリストグループをフィルタ条件として設定することで帯域制御が行われます。
以下は，「1-site-EX」で学習した IP アドレス向けに上限 50Mbps で制御するための設定を示します。

```
PureFlow(A)> add scenario /port1/1-site-ex action aggregate peak_bw 50M
PureFlow(A)> add filter scenario /port1/1-site-ex filter site ipv4 dip list 1-site-EX
```

（例 2） トップレベルドメインとセカンドレベルドメインのラベルが「example3.com」である場合の通信を制御するため，ドメインフィルタ機能を使用する。

手順 1） ルールリストにて，対象ドメイン名を制御するためのルールリストグループを作成します。

```
PureFlow (A)> add rulelist group 2-site-EX3 domain
```

手順 2） 手順 1 で作成したグループにエントリとして対象ドメイン名をワイルドカード「*」付きで登録します。

```
PureFlow(A)> add rulelist entry 2-site-EX3 domain *.example3.com
```

ここで DNS レスポンスパケットから「example3.com」が存在するすべての IP アドレスを自動的に検索して学習します。
（例 1）と同様に，帯域制御シナリオ設定後，ここで作成したルールリストグループをフィルタ条件として設定することで帯域制御が行われます。
以下は，「2-site-EX3」で学習した IP アドレス向けに上限 70Mbps で制御するための設定を示します。

```
PureFlow(A)> add scenario /port1/2-site-ex3 action aggregate peak_bw 70M
PureFlow(A)> add filter scenario /port1/2-site-ex3 filter site ipv4 dip list 2-site-EX3
```

19.4 確認手順

ドメインフィルタ機能で学習したドメイン名と IP アドレス情報は，ルールリストの”show rulelist”コマンドで確認することが可能です。

確認方法に関しても設定方法と同様に，CLI のみの対応です。ただし，WebGUI に関してはドメインフィルタ機能で作成したグループ名およびエントリ名（ドメイン名）を表示します。学習したエントリ IP アドレスは表示しません。

（例 1） 設定したルールリストすべての情報を確認する。

```
PureFlow(A)> show rulelist all
Total rulelist groups: 2

ListName: 1-site-EX
        Type                    : domain
        Rulelist Index          : 1
Number of Rules  :
        Total                   : 512
        Used                    : 1
        Available     : 511
Number of IP Address Learning:
        Total                   : 512
        Used                    : 1
        Available     : 511
Rules:
        [   1]          : example1.com
        <Domain IP>
        NAME            : example1.com          ←自動登録したドメイン名を表示します。
        CNAME           : abc.example1.com      ←自動登録した CNAME を表示します。
        Address         : 192.0.2.10            ←自動登録した IP アドレスを表示します。
        TTL             : 87000[s]              ←保存時間を表示します。
ListName: 1-site-EX3
        Type                    : domain
        Rulelist Index        : 2
Number of Rules:
        Total                   : 512
        Used                    : 1
        Available     : 511
Number of IP Address Learning:
        Total                   : 512
        Used                    : 1
        Available     : 511
Rules:
        [   1]          : *.example3.com
        <Domain IP>
        NAME              : zzz.example3.com
        CNAME           : yy.zzz. example.com
        Address         : 198.51.100.10
        TTL             : 86400[s]
```

（例 2） 任意のルールリスト情報を確認する。

```
PureFlow(A)> show rulelist name 1-site-EX
Total rulelist groups: 2

ListName: 1-site-EX
        Type                  : domain
        Rulelist Index        : 1
        Number of Rules       :
            Total             : 512
            Used              : 1
            Available         : 511
        Number of IP Address Learning:
            Total             : 512
            Used              : 1
            Available         : 511
        Rules:
                [   1]        : example1.com
                <Domain IP>
                NAME      : example1.com          ←自動登録したドメイン名を表示します。
                CNAME     : abc.example1.com      ←自動登録した CNAME を表示します。
                Address   : 192.0.2.10            ←自動登録した IP アドレスを表示します。
                TTL       : 87000[s]              ←保存時間を表示します。

                NAME      : example1.com          ←自動登録したドメイン名を表示します。
                CNAME     : def.example1.com      ←自動登録した CNAME を表示します。
                Address   : 192.0.2.20            ←自動登録した IP アドレスを表示します。
                TTL       : 88000[s]              ←保存時間を表示します。
```

また，エントリ IP アドレスのリソース状況については”show resource”コマンドで確認することができます。

```
PureFlow(A)> show resource
Resource information
                                          Total        Used      Available
-------------------------------------------------------------------
Scenario                        :         4100           4       4096 [entry]
   Individual Que               :         4096           0       4096 [entry]
   Second Peer                  :          100           0        100 [entry]
   Keep Alive                   :          100           0        100 [entry]
Filter                          :        10000           0      10000 [entry]
Rulelist                        :         1024           2       1022 [group]
Total Rulelist Entry            :        10000           2       9998 [entry]
Total Domain IP Entry           :        10000         356       9644 [entry]
                                                ↑エントリ IP アドレスのリソース状況が確認できます。
          :                              :            :            :
          :                              :            :            :
```

エントリ IP アドレスが装置上限に達した場合は syslog に記録します。syslog に関しては「付録B SYSLOG 一覧」を参照してください。

19.5 DNS レスポンスパケット装置不通過の場合の対応

本装置が存在する経路に DNS レスポンスパケットが通過しないがドメインフィルタ機能を使用する必要がある場合の対応方法について説明します。

図 19.5-1 に示す構成で Site-EX 向けの通信を制御する場合，Site-EX の IP アドレス変化に追従する必要があります。しかしながら DNS サーバはユーザ側にあるため，DNS レスポンスパケットが装置を通過せず，ドメインフィルタ機能を使用することができません。その場合図 19.5-2 に示す構成で対応することができます

図 19.5-1　DNS レスポンスパケット装置不通過ケース

図 19.5-2　通信ポート以外の Network ポートによるドメインフィルタ機能実現構成

DNS レスポンスパケットが通過する経路上のスイッチの任意ポートをモニタモードとして設定し，DNS レスポンスパケットを流せるようにします。モニタモードとして設定したポートに装置の空きポート（図の例では /port3）を接続し,DNS レスポンスパケットを装置に受信させることでドメインフィルタ機能による Site-EX の IP アドレス学習が可能となります。

（設定例）

```
PureFlow (A)> add rulelist group 1-site-EX domain
PureFlow (A)> add rulelist entry 1-site-EX domain *.example.com
PureFlow(A)> add scenario /port1/1-site-ex action aggregate peak_bw 50M
PureFlow(A)> add filter scenario /port1/1-site-ex filter site ipv4 dip list 1-site-EX
```

19.6 注意事項

(1) クラウドサービスやアップデートサービスなどでお使いになる場合は，対象サービスのドメイン名がプロバイダーや Web サイト等で公開されている必要があります。本装置は，あらかじめ制御対象のドメイン名を登録しておくことで，名前解決発生時に IP アドレスを自動学習し，その IP アドレスをフィルタ条件として通信を制御します。

(2) 本機能で動作する DNS レスポンスパケットは UDP でかつデータサイズ（UDP ヘッダ＋データ）が 512 バイト以下（RFC1035 準拠）のものです。DNSSEC および，ゾーン転送などで用いられる TCP パケットでは動作しません。また，DNS の拡張方式である EDNS0（RFC6891）にも対応しておりません。

(3) 本機能の設定および確認は CLI のみで，WebGUI，WebAPI，OpenFlow には対応していません。
ただし，設定したグループおよびエントリは削除できます。
WebGUI に関しては本機能で作成したグループ名およびエントリ名（ドメイン名）を確認できますが，学習したエントリ IP アドレスは確認できません。

(4) 本機能は，IPv6 パケットには対応していません。

(5) エントリにワイルドカードを使用した場合，本機能において発見したドメイン名が設定した複数のドメイン名に一致する場合があります。その場合，一致する文字数の多いドメイン名を設定したルールリストグループに登録します。

（例）
・エントリとして設定したドメイン名
設定ドメイン名①　*.example.com　――ルールリストグループ A に設定
設定ドメイン名②　aaa.example.com　――ルールリストグループ B に設定
・ドメインフィルタ機能により発見したドメイン名および IP アドレス
発見ドメイン名／IP アドレス　aaa.example.com／203.0.113.10
上記の場合，発見ドメイン名／IP アドレス「aaa.example.com／203.0.113.10」はラベル数が多いほうのルールリストグループ B に登録されます。

(6) 「NF7500-L141A ドメインフィルタ機能ライセンス」は，「NF7500-L201A TCP 高速化機能ライセンス」との同時適用はできません。

(7) インターネットアクセスにプロキシサーバを適用した構成で，装置をプロキシサーバよりユーザ側に設置した場合，プロキシサーバ宛ての通信はドメインフィルタ機能で分類することができません。

(8) 異なるドメイン名を設定して検索した IP アドレスが両方とも同じであった場合，それぞれのドメイン名に対して同じ IP アドレスを学習します。この場合，ドメイン名ごとに帯域制御は行えません。

第 20 章　トラフィック分析機能

ここでは，トラフィック分析機能について説明します。

20.1　概要 20-2
20.2　トラフィック分析の測定項目 20-3
20.3　トラフィック分析の集計方法 20-6
20.4　トラフィック分析の設定方法 20-9
20.5　トラフィック生成機能 20-16
20.6　注意事項 20-21

20.1 概要

トラフィック分析機能は，ネットワークの転送性能，および，アプリケーションの転送性能を測定するための機能です。本装置を経由するアプリケーショントラフィックのパケットヘッダを参照し，パケット損失や転送遅延を測定します。これらを継続して測定することで,ネットワークの状態変化を監視するための指標として使用します。

図 20.1-1　トラフィック分析の概要

また，PureFlow Profiler を使用することにより，トラフィック分析結果をグラフ表示し，過去のデータを含めたレポートを作成することができます。詳細は，「PureFlow Profiler モニタリングマネージャ3 NF7202A 取扱説明書」を参照してください。

図 20.1-2　PureFlow Profiler の表示例

20.2 トラフィック分析の測定項目

トラフィック分析機能において,アプリケーションの種別ごとに測定項目が異なります。ここでは,アプリケーション種別ごとの測定項目を説明します。

20.2.1 アプリケーション種別

トラフィック分析機能は，いくつかのアプリケーションを測定対象とします。測定対象の一覧を以下に記載します。

表 20.2.1-1　測定対象のアプリケーション種別

アプリケーション種別	概要
TCP	IPv4 および IPv6 の TCP 通信を測定します。
ICMP	IPv4 および IPv6 の ICMP 通信を測定します。測定対象は ICMP echo 要求と応答のみです。

各アプリケーションに合わせ，様々な項目を測定します。測定項目は以下の説明を参照してください。

20.2.2 TCP の測定項目

TCP 通信を検出したとき，トラフィック分析機能は，以下の項目を測定します。

表 20.2.2-1　TCP の測定項目

測定項目	測定方法	用途
ネットワーク遅延 Network RTT	端末が送信した SYN パケットを転送してから，対向側の端末が送信した応答パケット（ACK パケットまたは RST パケット）を転送するまでの時間を測定する。	ネットワークの往復遅延を予測する指標として使用する。
サーバ遅延 ServerRTT	端末が送信した最初のデータパケットを転送してから，対向側の端末が送信したデータパケットを転送するまでの時間を測定する。	アプリケーションの起動時間を予測する指標として使用する。
データ往復遅延 Data RTT	端末がデータを送信してから，対向側の端末がデータを送信し，本装置に到達するまでの時間を測定する。	アプリケーションの応答時間の変化を予測する指標しとして使用する。
データ ACK 遅延 Data ACK RTT	端末がデータを送信してから，対向側の端末がデータ ACKを送信し，本装置に到達するまでの時間を測定する。パケット再送があった場合は，再送時間も含めた時間を測定する。	データ転送に要した時間を予測する指標として使用する。
データ損失 Segments lost	TCP プロトコルのシーケンス番号をパケットごとに参照し，データ損失を測定する。TCP ヘッダのシーケンス番号が，次に来るべきシーケンス番号よりも大きいとき，データ損失と判定する。	ネットワークのデータ損失を予測する指標として使用する。
データ再送 Segments Retransmitted	TCP プロトコルのシーケンス番号をパケットごとに参照し，データ再送を測定する。TCP ヘッダのシーケンス番号が，次に来るべきシーケンス番号よりも小さいとき，データ再送と判定する。	TCP によるデータ再送を予測する指標として使用する。
SYN 受信回数 SYN received	TCP ヘッダの SYN フラグがセットされたパケットを計上する。	通信の有無を確認するために使用する。
ACK 受信回数 ACK received	TCP ヘッダの ACK フラグがセットされたパケットを計上する。	通信の有無を確認するために使用する。

DATA 受信回数 DATA received	TCP パケットにおいてデータ長が1以上のパケットを計上する。再送パケットも含む。	通信の有無を確認するために使用する。
FIN 受信回数 FIN received	TCP ヘッダの FIN フラグがセットされたパケットを計上する。	通信の有無を確認するために使用する。
RST 受信回数 RST received	TCP ヘッダの Reset フラグがセットされたパケットを計上する。	通信の有無を確認するために使用する。

図 20.2.2-1　TCP の測定項目

20.2.3 ICMP の測定項目

ICMP echo 要求を検出したとき，トラフィック分析機能は，以下の項目を測定します。

表 20.2.3-1　ICMP の測定項目

測定項目	測定方法	用途
ネットワーク遅延 Network RTT	端末が送信した ICMP echo 要求を転送してから，対向側の端末が送信した ICMP echo 応答を転送するまでの時間を計測します。	ネットワークの往復遅延を測定する指標として使用する。

図 20.2.3-1　ICMP の測定項目

20.3 トラフィック分析の集計方法

トラフィック分析の集計方法として，2種類あります。ここでは，トラフィック分析のシナリオ集計とトップ集計について説明します。

20.3.1 シナリオ集計

シナリオ集計は，シナリオを通過するトラフィックを対象に，トラフィック分析で得られた NetworkRTT などの測定値を集計し，最大値，平均値，最小値，ヒストグラムなどの統計情報を生成します。シナリオを階層的に指定することにより，統計情報も階層的に集計することができます。

図 20.3.1-1　シナリオ集計

シナリオ集計は，トラフィック種別ごとに様々な統計情報を生成します。シナリオ集計で生成する統計情報を以下の表に記します。

表 20.3.1-1　シナリオ集計の統計情報

トラフィック種別	測定項目	統計情報					
		最大値	最小値	平均値	合計値	比率	ヒストグラム
TCP	Network RTT ネットワーク遅延	○	○	○	－	－	○
	Server RTT サーバ遅延	○	○	○	－	－	○
	Data RTT データ RTT	○	○	○	－	－	○
	Data ACK RTT データ ACK RTT	○	○	○	－	－	○
	Segments sent out データ送信数	－	－	－	○	－	－
	Segments lost データ損失数	－	－	－	○	○※1	－
	Segments retransmitted データ再送数	－	－	－	○	○※2	－
	Number of Flow フロー数	－	－	－	○	－	－
	Number of Flow with loss データ損失が発生したフロー数	－	－	－	○	－	－
	Number of Flow with retransmit データ再送が発生したフロー数	－	－	－	○	－	－
	SYN received TCP SYN フラグパケットの受信数	－	－	－	○	－	－
	ACK received TCP ACK フラグパケットの受信数	－	－	－	○	－	－
	DATA received TCP データパケットの受信数	－	－	－	○	－	－
	FIN received TCP FIN フラグパケットの受信数	－	－	－	○	－	－
	RST received TCP RST フラグパケットの受信数	－	－	－	○	－	－

※1）データ損失率を表示します。「Segments lost÷Segments send out」で算出した結果です。
※2）データ再送率を表示します。「Segments retransmitted÷Segments send out」で算出した結果です。

20.3.2 トップ集計

トップ集計は，トラフィック分析で得られた測定値を送受信 IP アドレスなどのセッション単位で細分化して集計します。トップ集計は，シナリオごとに最大で 100 セッションまでを集計し，Network RTT が大きい順に各セッションをソートして表示します。

トップ集計が生成する統計情報を以下に記します。

表 20.3.2-1　トップ集計の集計項目

トラフィック種別	集計項目	説明
TCP	Time セッション生成時刻	TCP セッションが生成された時刻を記録します。
	Direction フロー方向	TCP セッションの方向を記録します。TCP クライアントのフローを TCP SYN，TCP サーバのフローを TCP SYNACK と表示します。
	Type IP プロトコルのバージョン	IPv4，IPv6 などの IP バージョンを記録します。
	Src Addr Source IP アドレス	IP ヘッダの Source IP アドレスを記録します。
	Dst Addr Destination IP アドレス	IP ヘッダの Destination IP アドレスを記録します。
	Protocol プロトコル番号	IP ヘッダのプロトコル名（TCP）を記録します。
	Src Port Source Port 番号	TCP ヘッダの Source Port 番号を記録します。
	Dst Port Destination Port 番号	TCP ヘッダの Destination Port 番号を記録します。
	Network RTT ネットワーク遅延	Network RTT の測定値を記録します。
	Server RTT サーバ遅延	Server RTT の測定値を記録します。
ICMP	Time セッション生成時刻	ICMP echo 要求を転送した時刻を記録します。複数の ICMP echo 要求があった場合，集計周期内の最後の ICMP echo 要求を転送した時刻を記録します。
	Direction フロー方向	ICMP Request，および，ICMP Reply など，フローの方向を記録します。
	Type IP プロトコルのバージョン	IPv4，IPv6 などの IP バージョンを記録します。
	Src Addr Source IP アドレス	IP ヘッダの Source IP アドレスを記録します。
	Dst Addr Destination IP アドレス	IP ヘッダの Destination IP アドレスを記録します。
	Protocol プロトコル番号	IP ヘッダのプロトコル名（ICMP）を記録します。
	Network RTT ネットワーク遅延	測定した Network RTT を記録します。

20.4 トラフィック分析の設定方法

トラフィック分析の設定方法と表示方法について説明します。

20.4.1 シナリオ集計の設定方法

シナリオ集計を使用するには，あらかじめ，トラフィック分析有効設定（“set analysis”コマンド）と測定対象シナリオの追加（“add analysis target”コマンド）を実行してください。

まず，トラフィック分析を有効に設定します。

```
PureFlow(A)> set analysis enable
```

次に，測定対象シナリオを追加します。測定対象シナリオを追加するとき，シナリオ集計を行うトラフィック種別を指定します。例えば，シナリオ“/port1”を通過する TCP トラフィックの統計情報を集計するとき．測定対象シナリオを以下のように追加します。

```
PureFlow(A)> add analysis target scenario /port1 tcp
```

反対方向のトラフィックについても統計情報を集計する場合は，測定対象シナリオとして反対方向のシナリオも追加します。例えば，TCP トラフィックが“/port1”と “/port2”を通過する場合は，測定対象シナリオとして“/port2”も追加します。シナリオ集計において，測定対象シナリオは最大 200 個まで追加できます。

```
PureFlow(A)> add analysis target scenario /port2 tcp
```

集計周期（デフォルト 5 分）が経過するまで待ち，結果を表示します。例えば，シナリオ“/port1”を通過するトラフィックの統計情報を表示した場合，トラフィック分析表示（“show analysis target”コマンド）で，以下のように表示します。

```
PureFlow(A)> show analysis target scenario /port1 histogram
From        : 2020 Oct 27 11:15:24  To          : 2020 Oct 27 11:20:24

TCP
  Network RTT   (Min/Avg/Max)      :       0.144/    0.290/    0.485[msec]     6[times]
  Server RTT    (Min/Avg/Max)      :       0.912/    7.191/   19.802[msec]     6[times]
  Data RTT      (Min/Avg/Max)      :       0.561/   57.653/  446.927[msec]    15[times]
  Data Ack RTT  (Min/Avg/Max)      :       0.033/    2.836/   33.982[msec]    15[times]
  Segments sent out                :          6936[bytes]
  Segments lost                    :             0[bytes]          0[%]        0[times]
  Segments retransmitted           :             0[bytes]          0[%]        0[times]
  Number of Flow                   :             9[flows]
  Number of Flow with loss         :             0[flows]
```

```
  Number of Flow with retransmit  :            0[flows]
  SYN  received                   :            6[times]
  ACK  received                   :          714[times]
  DATA received                   :           15[times]
  FIN  received                   :            9[times]
  RST  received                   :            0[times]

Histogram (Network RTT)                 Histogram (Server RTT)
  Time Interval  Count                    Time Interval  Count
  -------------  --------------------   -------------  --------------------
           1ms                     6            1ms                     1
           2ms                     0            2ms                     1
           4ms                     0            4ms                     0
           6ms                     0            6ms                     1
         10ms                      0           10ms                     2
         20ms                      0           20ms                     1
         40ms                      0           40ms                     0
         60ms                      0           60ms                     0
        100ms                      0          100ms                     0
        200ms                      0          200ms                     0
        400ms                      0          400ms                     0
        600ms                      0          600ms                     0
       1000ms                      0         1000ms                     0
       2000ms                      0         2000ms                     0
       4000ms                      0         4000ms                     0
  above 4000ms                     0   above 4000ms                     0
  Histogram (Data RTT)                  Histogram (Data Ack RTT)
  Time Interval  Count                    Time Interval  Count
  -------------  --------------------   -------------  --------------------
           1ms                     2            1ms                    10
           2ms                     4            2ms                     3
           4ms                     1            4ms                     1
           6ms                     1            6ms                     0
         10ms                      2           10ms                     0
         20ms                      1           20ms                     0
         40ms                      0           40ms                     1
         60ms                      1           60ms                     0
        100ms                      1          100ms                     0
        200ms                      0          200ms                     0
        400ms                      1          400ms                     0
        600ms                      1          600ms                     0
       1000ms                      0         1000ms                     0
       2000ms                      0         2000ms                     0
       4000ms                      0         4000ms                     0
  above 4000ms                     0   above 4000ms                     0
```

統計情報の表示フォーマットについては，「PureFlow WS1 トラフィックシェーパーNF7500 シリーズコマンドリファレンス」の”show analysis target”コマンドの表示を参照してください。

なお，シナリオ集計は測定対象のトラフィックが流れてから，次の集計周期が経過した時に測定結果を集計します。トラフィック分析表示(“show analysis target”コマンド)を連続で実行した場合，次の集計周期が経過するまで，同じ集計結果を表示します。

20.4.2 トップ集計の設定方法

トップ集計を使用するには，あらかじめ，トラフィック分析有効設定（“set analysis”コマンド）と測定対象シナリオの追加（“add topanalysis target”コマンド）を実行してください。

まず，トラフィック分析を有効に設定します。

```
PureFlow(A)＞ set analysis enable
```

次に,測定対象シナリオを追加します。測定対象シナリオを追加するとき，トップ集計で集計する集計単位を指定します。例えば，シナリオ“/port1”を通過するトラフィックについて，フロー単位で細分化して表示する場合は，以下のように設定します。

```
PureFlow(A)＞ add topanalysis target scenario /port1 flow
```

反対方向のトラフィックについても統計情報を集計する場合は，測定対象シナリオとして反対方向のシナリオも追加します。例えば，TCP トラフィックが“/port1”と “/port2”を通過する場合は，測定対象シナリオとして“/port2”も追加します。トップ集計において，測定対象シナリオを最大 25 個まで設定できます。

```
PureFlow(A)＞ add topanalysis target scenario /port2 flow
```

集計周期（デフォルト 5 分）が経過するまで待ち，結果を表示します。例えば，シナリオ“/port1”を通過するトラフィックの統計情報を表示した場合，トラフィック分析表示(“show topanalysis target”コマンド)は，以下のように表示します。測定結果をフロー単位で集計し，NetworkRTT が大きい順にソートして表示します。

```
PureFlow(A)> show topanalysis target scenario /port1
From        : 2020 Oct 27 11:15:00  To          : 2020 Oct 27 11:20:00

Sort Type    : Network RTT
Flow
    Flow 1:
      Time                        : 2020 Oct 27 11:15:38
      Direction                   : TCP SYN
      Type                        : IPv4
      Src Addr                    : 192.168.37.15
      Dst Addr                    : 192.168.37.1
      Protocol                    : TCP
      Src Port                    : 56647
      Dst Port                    : 80
      Network RTT                 : 5.485     [msec]
      Server RTT                  : 19.802    [msec]
    Flow 2:
      Time                        : 2020 Oct 27 11:16:42
```

```
      Direction                    : TCP SYN
      Type                         : IPv4
      Src Addr                     : 192.168.37.16
      Dst Addr                     : 192.168.37.1
      Protocol                     : TCP
      Src Port                     : 56649
      Dst Port                     : 80
      Network RTT                  : 2.312      [msec]
      Server RTT                   : 18.640     [msec]
PureFlow(A)>
```

統計情報の表示フォーマットは，「PureFlow WS1 トラフィックシェーパーNF7500 シリーズコマンドリファレンス」の”show topanalysis target”コマンドの表示を参照してください。

なお，トップ集計もシナリオ集計と同様に，測定対象のトラフィックが流れてから，次の集計周期が経過した時に測定結果を集計します。トラフィック分析表示(“show topanalysis target”コマンド)を連続で実行した場合，次の集計周期が経過するまで，同じ集計結果を表示します。

20.4.3 トラフィック分析の測定対象シナリオ

各ポート方向のシナリオ(/port1 や/port2 など)を測定対象にした場合，各シナリオで確認できるネットワーク遅延(Network RTT)とサーバ遅延(Server RTT)を，以下に示します。シナリオのツリーモード設定(inbound または outbound)により，逆方向のシナリオが対象となります。

図 20.4.3-1　TCP のネットワーク遅延とサーバ遅延の測定項目

表 20.4.3-1　シナリオツリーモード(入力側:inbound 設定時)の測定対象シナリオ

パケット種別	PureFlow 1		PureFlow 2	
	/port1 シナリオ (LAN⇒WAN)	/port2 シナリオ (WAN⇒LAN)	/port2 シナリオ (WAN⇒LAN)	/port1 シナリオ (LAN⇒WAN)
SYN	①Network RTT	－	②Network RTT	
SYN/ACK		④Network RTT		③Network RTT
ACK	－			
Data Req	⑤Server RTT	－	⑥Server RTT	
Data Resp		－		

表 20.4.3-2　シナリオツリーモード(出力側:outbound 設定時)の測定対象シナリオ

パケット種別	PureFlow 1		PureFlow 2	
	/port1 シナリオ (LAN⇒WAN)	/port2 シナリオ (WAN⇒LAN)	/port2 シナリオ (WAN⇒LAN)	/port1 シナリオ (LAN⇒WAN)
SYN	－	①Network RTT	－	②Network RTT
SYN/ACK	④Network RTT		③Network RTT	
ACK		－		－
Data Req	－	⑤Server RTT	－	⑥Server RTT
Data Resp	－		－	

なお，ICMP のネットワーク遅延(Network RTT)も同様です。

次にデータ損失（Segments lost）とデータ再送（Segments Retransmitted）を，以下に示します。

表 20.4.3-3　シナリオツリーモード（入力側：inbound 設定時）の測定対象シナリオ

パケット種別	PureFlow 1		PureFlow 2	
	/port1 シナリオ (LAN⇒WAN)	/port2 シナリオ (WAN⇒LAN)	/port2 シナリオ (WAN⇒LAN)	/port1 シナリオ (LAN⇒WAN)
HTTD Data1	－	－	－	－
HTTP Data2	－	－	－	－
HTTP Data3	－	－	－	－
HTTP Data4	①Segments lost	－	①Segments lost	－
HTTP Data5	②Segments Retransmitted	－	②Segments Retransmitted	－

表 20.4.3-4　シナリオツリーモード（出力側：outbound 設定時）の測定対象シナリオ

パケット種別	PureFlow 1		PureFlow 2	
	/port1 シナリオ (LAN⇒WAN)	/port2 シナリオ (WAN⇒LAN)	/port2 シナリオ (WAN⇒LAN)	/port1 シナリオ (LAN⇒WAN)
HTTD Data1	－	－	－	－
HTTP Data2	－	－	－	－
HTTP Data3	－	－	－	－
HTTP Data4	－	①Segments lost	－	①Segments lost
HTTP Data5	－	②Segments Retransmitted	－	②Segments Retransmitted

20.5 トラフィック生成機能

トラフィック生成機能は，ネットワークに接続されたサーバや端末などに対し，本装置から定期的にトラフィックを生成する機能です。この機能を使うことにより，端末からの通信が全くない状態が継続する場合でも，トラフィック分析を実施し，転送遅延などの測定を継続することができます。

20.5.1 生成するトラフィックの種類

トラフィック生成機能は， HTTP/HTTPS トラフィックと ICMP トラフィックを生成します。これらのトラフィックについて説明します。

(1)HTTP/HTTPS トラフィック

TCP を使った HTTP/HTTPS トラフィックを定期的に生成します。生成する周期はデフォルトで1分です。生成周期が経過するたびに TCP セッション接続，HTTP/HTTPS コマンドの送信， TCP セッション切断を行います。

図 20.5.1-1　HTTP トラフィック

(2)ICMP echo トラフィック

ICMP echo トラフィックを定期的に生成します。生成する周期はデフォルトで1分です。生成周期が経過するたびに ICMP echo 要求を送信します。

図 20.5.1-2　ICMP トラフィック

20.5.2 トラフィックを送信するインタフェース

トラフィック生成機能は，本装置のシステムインタフェースからトラフィックを送信します。

トラフィック生成機能で送信するトラフィックをトラフィック分析の測定対象とする場合は，システムインタフェースのポート設定(“set ip sytsem port”コマンド)において，“network”を指定してご利用ください。”ethernet“を指定されている場合は，システムインタフェースから送信されるトラフィックが，本装置の Network ポートを経由するように，ネットワークケーブルを接続してください。

図 20.5.2-1　システムインタフェースのポート設定”が”network“の場合

図 20.5.2-2　システムインタフェースのポート設定”が”ethernet“の場合

生成したトラフィックを送受信する両 Network ポート，または，両ポートに属するシナリオをトラフィック分析の測定対象として追加してください。

20.5.3 トラフィック生成の設定方法

トラフィック生成機能の設定方法を説明します。

まず，システムインタフェースを設定します。システムインタフェースのパケットを Network ポートで送受信する場合の例です。すでに設定されている場合は，不要です。

生成するトラフィックをトラフィック分析の測定対象とする場合，シナリオによるトラフィックコントロールを有効に設定します。

```
PureFlow(A)＞ set ip system port network scenario enable
PureFlow(A)＞ set ip system port network in all
PureFlow(A)＞ set ip system 192.168.37.10 netmask 255.255.255.0 up
PureFlow(A)＞ set ip system gateway 192.168.37.1
```

トラフィック分析を有効に設定します。

```
PureFlow(A)＞ set analysis enable
```

次に，トラフィックを生成する設定(“add analysis traffic_generator”コマンド)を追加します。

HTTP サーバ 192.168.37.20 に対して HTTP コマンドを定期的に生成する場合，以下のように設定します。

```
PureFlow(A)＞ add analysis traffic_generator ipv4 dip 192.168.37.20 normalhttp url
index.html
```

送信するトラフィックの種類を追加するときは，設定を追加します。以下は，ICMP トラフィックを追加するときの例です。

```
PureFlow(A)＞ add analysis traffic_generator ipv4 dip 192.168.37.20 icmp
```

送信するトラフィックは最大 25 個まで追加できます。

生成するトラフィックをトラフィック分析の測定対象とする場合，両 Network ポートに属するシナリオとフィルタを追加し，測定対象シナリオを追加します。

例えば，生成する HTTP トラフィックを測定対象とする場合は，以下のように両 Network ポートにシナリオとフィルタを設定し，これらのシナリオを測定対象シナリオとして追加します。フィルタ設定においては，宛先IPアドレス，送信元IPアドレス，プロトコル，HTTP のポート番号を指定します。

```
PureFlow(A)＞ add scenario ″/port1/ana1″ action forward
PureFlow(A)＞ add filter scenario ″/port1/ana1″ filter ″http1″ ipv4 sip 192.168.37.10
dip 192.168.37.20 proto tcp dport 80
```

```
PureFlow(A)＞ add filter scenario "/port1/ana1" filter "http2" ipv4 sip 192.168.37.20
dip 192.168.37.10 proto tcp sport 80
PureFlow(A)＞ add scenario "/port2/ana1" action forward
PureFlow(A)＞ add filter scenario "/port2/ana1" filter "http1" ipv4 sip 192.168.37.10
dip 192.168.37.20 proto tcp dport 80
PureFlow(A)＞ add filter scenario "/port2/ana1" filter "http2" ipv4 sip 192.168.37.20
dip 192.168.37.10 proto tcp sport 80
PureFlow(A)＞
PureFlow(A)＞ add topanalysis target scenario "/port1/ana1"  flow
PureFlow(A)＞ add topanalysis target scenario "/port2/ana1"  flow
```

また，生成する HTTP トラフィックに加え，ICMP トラフィックも測定対象とする場合は、上記の設定に加え，以下のように ICMP のフィルタ設定を追加します。

```
PureFlow(A)＞ add filter scenario "/port1/ana1" filter "icmp1" ipv4 sip 192.168.37.10
dip 192.168.37.20 proto icmp
PureFlow(A)＞ add filter scenario "/port1/ana1" filter "icmp2" ipv4 sip 192.168.37.20
dip 192.168.37.10 proto icmp
PureFlow(A)＞ add filter scenario "/port2/ana1" filter "icmp1" ipv4 sip 192.168.37.10
dip 192.168.37.20 proto icmp
PureFlow(A)＞ add filter scenario "/port2/ana1" filter "icmp2" ipv4 sip 192.168.37.20
dip 192.168.37.10 proto icmp
```

20.6 注意事項

トラフィック分析を使用する際の注意事項を説明します。

(1)　フィルタモードの設定について

フィルタモード設定(“set filter mode”コマンド)において，sip,dip,proto,sport,dport のすべてが含まれない場合，トラフィック分析を実行しません。また，フィルタモード設定(“set filter mode”コマンド)において，tos,cos が含まれる場合，トラフィック分析を実行しません。

(2)　測定対象シナリオのシナリオモードについて

トラフィック分析は，シナリオの種別として”aggregate”，”individual”，”forward”のいずれかが設定されたシナリオを測定対象としたとき，トラフィック分析を実行します。シナリオ種別として”wan-accel”または”discard”が指定されたシナリオの場合，トラフィック分析を実行しません。

(3)　トップ集計のセッション数について

トップ集計において，シナリオに 100 セッション以上のトラフィックがある場合，集計できないセッションが発生します。トップ集計を使用する場合は，シナリオに設定するフィルタ設定(“add filter scenario”コマンド)でパケット分類を細分化し，シナリオを通過するセッション数を調整してください。

(4)　トラフィック生成機能について

トラフィック分析が無効の場合は，トラフィック生成設定(“add analysis traffic_generator”コマンド)で設定されたトラフィックは送信されません。また，システムインタフェースの通信ポート設定が Network ポート経由で，トラフィック生成機能を使用する場合，測定対象シナリオで制御する必要があるため，“set ip system network port scenario”コマンドを“enable（有効）”に設定してください。パケット送受信に対する測定対象シナリオ方向（/port1 や/port2 など）については「第 7 章 システムインタフェースの設定」を参照してください。

(5)　システムインタフェース通信のポート設定との関係

Network ポート経由のシステムインタフェース通信は，当該出力ポートが無効設定の場合，システムインタフェース通信不可のため，トラフィック生成機能によるトラフィック分析もできません。
例えば，出力ポートが Network ポート 1/2 の場合，以下のようになります。

①Network ポート経由ですべてのポート（1/1, 1/2, 1/3, 1/4）を有効
　set ip system port network in all vid none inner-vid none　⇒トラフィック分析可能

②Network ポート経由でポート（1/1）のみ有効
　set ip system port network in 1/1 vid none inner-vid none　⇒トラフィック分析不可

③Network ポート経由でポート（1/2）のみ有効
　set ip system port network in 1/2 vid none inner-vid none　⇒トラフィック分析可能

（空白ページ）

付録A　デフォルト値

本装置には，機能に応じていくつもの設定項目があります。項目の中にはその機能を使用しない限り設定する必要のないものもありますが，設定が必須なものもあります。設定値を必要とする項目については，あらかじめ値が設定されています。表 A-1 に設定項目と設定値を示します。コマンドの詳細については「PureFlow WS1 トラフィックシェーパー NF7500 シリーズ コマンドリファレンス」を参照してください。

表 A-1　デフォルト値一覧

設定項目	コマンド	既設定値	設定範囲
ユーザ名	ユーザ名	root	設定なし
プロンプト	set prompt	PureFlow	最大 32 文字
ボーレート	set console baudrate	9600 bps	9600/19200/38400/115200 bps
ページャ	set pager	enable	enable/disable
オートログアウト	set autologout time	10 分	1～30 分
パスワード	set password	(なし)	最大 16 文字
	set adminpassword	(なし)	最大 16 文字
Network ポート設定（1000BASE-T RJ-45/SFP のみ適用）	set port media-type	rj45	rj45/sfp
	set port autonegotiation	enable	enable/disable
	set port speed	1G	1G/100M/10M
	set port duplex	full	full/half
フローコントロール	set port flow_control	auto	auto Pause 受信 on/off Pause 送信 on/off
最大フレーム長	set port mtu	2048	2048/10240
Ethernet ポート設定	set port autonegotiation system	enable	enable/disable
	set port speed system	1G	1G/100M/10M
	set port duplex system	full	full/half
SYSLOG	set syslog host	disable	enable/disable
	add syslog host (IP Address)	(なし)	IP Address
	add syslog host (UDP port)	514	1～65534
	set syslog severity	notice(5)	0～6
	set syslog facility ccpu	16(local0)	0～23
	set syslog facility fcpu	17(local1)	0～23
SNMP	set snmp syscontact	Not Yet Set	最大 200 文字
	set snmp syslocation	Not Yet Set	最大 200 文字
	set snmp sysname	Not Yet Set	最大 200 文字
	set snmp traps	すべて enable	トラップごとに enable/disable

設定項目	コマンド	既設定値	設定範囲
SNMP (続き)	add snmp view	(なし)	view レコード名 OID included/excluded
	add snmp community	(なし)	コミュニティ名 バージョン View 名 ReadOnly/ReadWrite
	add snmp group	(なし)	グループ名 認証方式 ReadView WriteView NotifyView
	add snmp user	(なし)	ユーザ名 グループ名 認証方式 パスワード
	add snmp host	(なし)	IPv4 Address バージョン 認証方式 ユーザ名/コミュニティ名 Trap/Inform UDP ポート番号 送信ノーティフィケーション
タイムゾーン	set timezone	UTC +09:00	UTC からのオフセット
夏時間	set summertime	(なし)	開始日時 終了日時 オフセット
SNTP	set sntp	disable	enable/disable
	set sntp server	(なし)	IP Address
	set sntp interval	3600 秒	60～86400 秒
RADIUS	set radius auth	disable	enable/disable
	set radius auth timeout	5	1～30 秒
	set radius auth retransmit	3	0～10 回
	set radius auth method	CHAP	CHAP/PAP
RADIUS サーバ	add radius auth server	(なし)	IP アドレス ポート番号 共通鍵 Primary
システム インタフェース	set ip system(IPv4 Address)	192.168.1.1	IPv4 Address
	set ip system(IPv4 netmask)	255.255.255.0	IPv4 Address
	set ip system(IPv4 up/down)	up	up/down
	set ip system(IPv6 Address)	::192.168.1.1	IPv6 Address
	set ip system(IPv6 prefixlen)	64	0～128
	set ip system(IPv6 up/down)	up	up/down

設定項目	コマンド	既設定値	設定範囲
システム インタフェース (続き)	set ip system port (ethernet/network)	ethernet	ethernet/network
	set ip system port network scenario	disable	enable/disable
	set ip system gateway(IPv4)	(なし)	IPv4 Address
	set ip system gateway(IPv6)	(なし)	IPv6 Address
システムインタフェースフィルタ	add ip system filter	(なし)	フィルタ Index sip, dip, tos, proto, sport, dport permit/deny
自動リブート	set autoreboot	enable	enable/disable
フロー識別モード	set filter mode	default	default, vid, cos, inner-vid, inner-cos, sip, dip, tos, proto, sport, dport
フローエージングタイム	set agingtime	300 秒	1～1800 秒
通信ギャップモード設定	set bandwidth mode	gap	gap/no_gap
ピークバーストサイズ	set shaper peak burst size	1536 Byte	0～9216 Byte
シナリオツリーモード	set scenario tree mode	inbound	inbound/outbound
トラフィックアクセラレーションバイパス	set wan-accel bypass status	enable	enable/disable
	set wan-accel bypass recoverytime	60 秒	1～600 秒
リンクダウン転送機能	set lpt	disable	enable/disable
	add lpt pair port	(なし)	ポート番号
Telnet 接続設定	set telnet	enable	enable/disable
SSH 接続設定	set ssh	enable	enable/disable
HTTP プロトコル	set http protocol	normalhttp	normalhttp/httpsecure
ネットワークバイパス設定	set bypass	auto	auto/on/off
トップカウンタ	set topcounter	disable	enable/disable
	set topcounter config interval time	5 分	1 / 5 / 60 / 180 / 1440 分
トラフィック分析	set analysis	disable	enable/disable

（空白ページ）

付録B　SYSLOG 一覧

syslog の一覧を表 B-1 に示します。表 B-1 は severity（カッコ内は重大度）ごとにまとめています。

（参考）
syslog メッセージには括弧（[ ]や< >）で囲まれた 16 進数が付加されるものがあります。括弧内の 16 進数はソースコード上の位置や変数値を表しており，当社内でのトラブルシューティングで参照します。

表 B-1　syslog 一覧

Severity	syslog メッセージ	発生条件	対応方法
Emergency (0)	Temperature #N of the system is critical：xx.xx	システムの温度が危険域 （#N は 1） （xx.xx は温度（℃））	このまま使用を続けるとハードウェアが損傷を受ける可能性があります。ただちに電源を落としてください。
Alert (1)	Temperature #N of the system is OK：xx.xx	システムの温度範囲が正常値に復帰 （#N は 1） （xx.xx は温度（℃））	回復措置は不要です。
	Temperature #N of the system is abnormal：xx.xx	システムの温度が異常 （#N は 1） （xx.xx は温度（℃））	設置環境の温度が範囲内（0～40℃）であることを確認してください。範囲内である場合は装置を交換してください。範囲外である場合は設置場所を変えてください。
	Power #N inserted	電源ユニットの装着 （#N は 0）	回復措置は不要です。 NF7501A では記録されません。
	Power #N removed	電源ユニットの抜去 （#N は 0）	回復措置は不要です。 NF7501A では記録されません。
	Power #N failed	電源ユニットの異常検出 （#N は 0）	下記を確認してください。 ・電源ケーブルは接続されているか。 ・供給電圧は規定内（AC 100 V～AC 127 V/ AC 200 V～AC 240 V）か。 NF7501A では記録されません。
	Power #N OK	電源ユニットの異常回復 （#N は 0）	回復措置は不要です。 NF7501A では記録されません。
	Fan #N inserted	ファンユニットの装着 （#N は 0）	回復措置は不要です。 NF7501A では記録されません。
	Fan #N removed	ファンユニットの抜去 （#N は 0）	回復措置は不要です。 NF7501A では記録されません。
	Fan #N failed	ファンユニットの異常検出 （#N は 0）	下記を確認してください。 ・ファンは回転しているか。
	Fan #N OK	ファンユニットの異常回復 （#N は 0）	回復措置は不要です。
	No response from Slot #N	モジュールからの応答なし （#N は 1）	弊社サポートまでご連絡ください。
	Slot #N response is OK	モジュールからの応答が回復 （#N は 1）	回復措置は不要です。

Severity	syslog メッセージ	発生条件	対応方法
Alert (1) （続き）	System Buffer %s almost full	システムバッファ%s のバッファ使用量が 90%を超過した。	トラフィック状況および各種設定をチェックしてください。
	System Buffer %s recovered	システムバッファ%s のバッファ使用量が 90%を超えたあと，50%を下回った。	回復措置は不要です。
	TCP WARP Engine Buffer #N almost full	TCP WARP エンジンバッファのバッファ使用量が 90%を超過した。 （#N は 1～100）	トラフィック状況および各種設定をチェックしてください。
	TCP WARP Engine Buffer #N recovered	TCP WARP エンジンバッファのバッファ使用量が 90%を超えたあと，50%を下回った。 （#N は 1～100）	回復措置は不要です。
	Critical error on FCPU Core[#N], Code[#M] Data1[0xxxxxxxxx] Data2[0xxxxxxxxx]	フォワーディング系処理部でコア停止異常が発生した。	弊社サポートまでご連絡ください。
	Queue blocktime exceeded. [S:#M Q:#Q]	シナリオ M で生成されたキューQ のパケット送出停止を検出した。	弊社サポートまでご連絡ください。
	Detected FCPU IIC error on port[#N/#M]	フォワーディング系処理部で IIC インタフェースの異常が発生した。 （#N は 1） （#M は 1～4）	弊社サポートまでご連絡ください。
Error (3)	CLI Command %s, failed during restoration %msg	起動時のコンフィギュレーションリストアでコマンド%s のエラーが発生した。 エラーメッセージは%msg。	弊社サポートまでご連絡ください。
Notice (5)	The buffer of queue exceeded the limit. [S:#M,Q:#Q]	シナリオ M で生成されたキューQ のパケットバッファ使用量が制限値を超過した。	キューバッファフルのためパケット廃棄が発生しています。入力バースト長の設定をチェックしてください。
	The buffer of queue is less than 50% of the limit.[S:#M,Q:#Q]	シナリオ M で生成されたキューQ のパケットバッファ使用量が制限値を超えたあと，制限値の 50%を下回った。	回復措置は不要です。
	Flow registration failure for the system.	装置内のフローが最大数を超えた。	トラフィック状況および各種設定をチェックしてください。
	Flow registration available for the system.	装置内のフローが最大数に達したあと，最大数の 50%を下回った。	回復措置は不要です。
	Queue allocation failure for the system.	装置内の個別キューが最大数を超えた。	個別キューが装置の最大数に達しているため最大数超過時のアクションを適用しています。トラフィック状況をチェックしてください。

Severity	syslog メッセージ	発生条件	対応方法
Notice (5) （続き）	Queue allocation available for the system.	装置内の個別キューが最大数に達したあと，最大数の 90%を下回った。	回復措置は不要です。
	Queue allocation failure for the scenario.[S:#M]	シナリオ M の個別キュー数が制限値を超過した。	個別キューがシナリオの制限数に達しているため最大数超過時のアクションを適用しています。トラフィック状況をチェックしてください。
	Queue allocation available for the scenario. [S:#M]	シナリオ M の個別キューが制限値に達したあと，制限値の 50%を下回った。	回復措置は不要です。
	Flow learn queue overflow	TCP セッション学習性能を超えるトラフィックが入力された。	学習できなかったセッションは TCP アクセラレーションを行いません。トラフィック状況を確認してください。
	Domain IP Entry exceeded the limit for the system.	装置内のドメイン IP エントリが最大数を超えた。	ドメイン IP エントリが装置の最大数に達しています。トラフィック状況を確認してください。
	Domain IP Entry is more than 80% of the limit for the system.	装置内のドメイン IP エントリが 80%を超えた。	ドメイン IP エントリが装置の最大数の 80%に達しています。トラフィック状況を確認してください。
	Domain IP Entry is less than 70% of the limit for the system.	装置内のドメイン IP エントリが 80%に達したあと，最大数の 70%を下回った。	回復措置は不要です。
	Domain IP Entry exceeded the limit for the rule list.[#M]	ルールリスト M のドメイン IP エントリが最大数を超えた。 #M は，ルールリスト名	ルールリスト M のドメイン IP エントリが装置の最大数に達しています。IP エントリの学習状況を確認し，複数のルールリストエントリやワイルドカードで指定している場合はルールリストを分割するなどの措置を行ってください。
	Detected MCU failure[xx]	MCU でエラーを検出した。	弊社サポートまでご連絡ください。
	Detected MCU recovery	MCU で検出したエラーが回復した。	回復措置は不要です。
	Wan-accel scenario switched to secondary-peer. [S:#M]	WAN-accel シナリオ M は対向装置を Primary-peer から Secondary-peer に切り替えた。	Primary-peer の回線状況や対向装置の状態を確認してください。
	Wan-accel scenario switched back to primary-peer. [S:#M]	WAN-accel シナリオ M は対向装置を Secondary-peer から Primary-peer に切り戻した。	回復措置は不要です。
	Wan-accel scenario switched to Bypass status.(TCP connection error) [S:#M]	WAN-accel シナリオ M は通信状態を Bypass に切り替えた。(TCP connection error)	回復措置は不要です。

Severity	syslog メッセージ	発生条件	対応方法
Notice (5) （続き）	Wan-accel scenario switched to Bypass status.(RTT threshold) [S:#M]	WAN-accel シナリオ M は通信状態を Bypass に切り替えた。(RTT threshold を下回った)	回復措置は不要です。
	Wan-accel scenario switched to Bypass status.(ping timeout) [S:#M]	WAN-accel シナリオ M は通信状態を Bypass に切り替えた。(ping が通らなかった)	回復措置は不要です。
	Wan-accel scenario switched to Bypass status.(Peer scenario error) [S:#M]	WAN-accel シナリオ M は通信状態を Bypass に切り替えた。(シナリオが見つからなかった)	回復措置は不要です。
	Wan-accel scenario switched to Acceleration status. [S:#M]	WAN-accel シナリオ M は通信状態を Acceleration に切り替えた。	回復措置は不要です。
	Wan-accel scenario switched to Force Bypass status. [S:#M]	WAN-accel シナリオ M は通信状態を Force Bypass に切り替えた。	回復措置は不要です。
	Appli-Accel Sessions exceeded the limit.[P:#P]	装置全体で使用可能な Appli-Accel セッションの上限を超えたため，超えた分のセッションは Appli-Accel が無効になりました。プロトコルは#P です。	回復措置は不要です。
	Appli-Accel Sessions exceeded the limit.[P:#P, S:#M]	シナリオごとに設定した Appli-Accel セッションの上限を超えたため，超えた分のセッションは Appli-Accel が無効になりました。プロトコルは#P です。シナリオは#M です。	回復措置は不要です。
	Appli-Accel Sessions is less than 50% of the limit.[P:#P]	Appli-Accel セッションの使用率が装置全体で使用可能な上限値の 50%以下に回復しました。プロトコルは#P です。	回復措置は不要です。
	Appli-Accel Sessions is less than 50% of the limit.[P:#P, S:#M]	Appli-Accel セッションの使用率がシナリオごとに設定した上限値の 50%以下に回復しました。プロトコルは#P です。シナリオは#M です。	回復措置は不要です。
	Appli-Accel Buffer almost full.[P:#P, ID:#I]	Appli-Accel で使用中のバッファの使用率が 90%を上回りました。プロトコルは#P です。バッファの固有 ID は#I です。	プロトコル#P の Appli-Accel を使用している WAN-accel シナリオの設定を確認してください。

Severity	syslog メッセージ	発生条件	対応方法
Notice (5) （続き）	Appli-Accel Buffer recoverd.[P:#P, ID:#I]	Appli-Accel で使用中のバッファの使用率が 50%以下に回復しました。プロトコルは#P です。バッファの固有 ID は#I です。	回復措置は不要です。
	Session limits between monitoring manager occurred.	モニタリングマネージャの接続数制限を超過した。	下記の制限値を超えるとモニタリングマネージャでの情報収集ができない場合があります。制限値を超過しないように使用してください。 収集周期　シナリオ数　モニタリングマネージャ接続数 10 秒　2000　2 10 秒　4000　1 30 秒　制限なし　4 60 秒　制限なし　4
	Session limits between monitoring manager is released.	モニタリングマネージャの接続数制限を超過したあと，制限を下回った。	回復措置は不要です。
	Monitoring manager session connected. （xxx.xxx.xxx.xxx）	モニタリングマネージャ（xxx.xxx.xxx.xxx）と接続した。	回復措置は不要です。
	Monitoring manager session disconnected [State:#N]. （xxx.xxx.xxx.xxx）	モニタリングマネージャ（xxx.xxx.xxx.xxx）との接続を切断した。 （State:#N は通信状態）	モニタリングマネージャとの通信経路に異常が発生していないかチェックしてください。
	Bypass state was changed to on.	Network ポートをバイパス ON 状態にした。	本 syslog の直前にバイパス接続の原因となった syslog が記録されています。 バイパスが接続状態となった理由を特定し，必要な措置を行ってください。
	Bypass state was changed to off.	Network ポートをバイパス OFF 状態にした。	回復措置は不要です。
	Exceeds max no. of sessions.	Telnet または SSH セッションの接続数制限を超過した。	Telnet と SSH を合わせ，最大 8 セッションまで同時利用可能です。制限値を超過しないように使用してください。
	Exceeds max no. of sessions from serial console.	シリアルコンソールからの端末セッションへの接続数制限を超過した。	シリアルコンソールは，RJ-45 または miniUSB を合わせ，最大 1 セッションまで同時利用可能です。制限値を超過しないように使用してください。

Severity	syslog メッセージ	発生条件	対応方法
Informa tional (6)	Port #N/#M changed Up from Down.	ポートがリンクアップ (#N は 1) (#M は 1～4)	回復措置は不要です。
	Port #N/#M changed Down from Up.	ポートがリンクダウン (#N は 1) (#M は 1～4)	下記を確認してください。 ケーブル断は起きていないか。 正しいケーブル(マルチモード／シングルモード，ストレート／クロス)を使用しているか。 Network ポートの Speed／Duplex および Pause の設定が接続装置と合っているか。
	Port #N/#M changed PowerDown with Link Pass Through.	リンクダウン転送機能が動作 (#N は 1) (#M は 1～4)	下記を確認してください。 ケーブル断は起きていないか。 正しいケーブル(マルチモード／シングルモード，ストレート／クロス)を使用しているか。 Network ポートの Speed／Duplex および Pause の設定が接続装置と合っているか。
	Warning. Port #N/#M Oper duplex is Half.	ポートが半二重でリンクアップ (#N は 1) (#M は 1～4)	下記を確認してください。 Network ポートの Speed／Duplex の設定が接続装置と合っているか。
	Management Ethernet Port changed Up from Down.	Management Ethernet ポートがリンクアップ	回復措置は不要です。
	Management Ethernet Port changed Down from Up.	Management Ethernet ポートがリンクダウン	下記を確認してください。 ケーブル断は起きていないか。 正しいケーブルを使用しているか。
	Warning. Management Ethernet Port Oper duplex is Half.	Management Ethernet ポートが半二重でリンクアップ	下記を確認してください。 Management Ethernet ポートの Speed／Duplex の設定が接続装置と合っているか。
	AnritsuPureFlow Software Version x.x.x	装置起動	回復措置は不要です。
	Loading Object from Master.	内蔵フラッシュメモリの Master ファイルからソフトウェアオブジェクトを読み込みました。	回復措置は不要です。
	Loading Object from Backup.	内蔵フラッシュメモリの Backup ファイルからソフトウェアオブジェクトを読み込みました。	回復措置は不要です。
	Loading Object from USB memory.	外部メディア(USB メモリ)からソフトウェアオブジェクトを読み込みました。	回復措置は不要です。

Severity	syslog メッセージ	発生条件	対応方法
Informa tional (6) （続き）	Loading Object from SD Card.	外部メディア（SD カード）からソフトウェアオブジェクトを読み込みました。	回復措置は不要です。
	Loading Configuration from Master.	内蔵フラッシュメモリの Master ファイルからコンフィギュレーションファイルを読み込みました。	回復措置は不要です。
	Loading Configuration from Backup.	内蔵フラッシュメモリの Backup ファイルからコンフィギュレーションファイルを読み込みました。	回復措置は不要です。
	Loading Configuration from USB memory.	外部メディア（USB メモリ）からコンフィギュレーションファイルを読み込みました。	回復措置は不要です。
	Loading Configuration from SD Card.	外部メディア（SD カード）からコンフィギュレーションファイルを読み込みました。	回復措置は不要です。
	User %s authentication from RADIUS server was Accept	ユーザ名%s の RADIUS 認証が accept された。	回復措置は不要です。
	User %s authentication from RADIUS server was Reject	ユーザ名%s の RADIUS 認証が reject された。	回復措置は不要です。
	User %s authentication from RADIUS server was Timeout	ユーザ名%s の RADIUS 認証がタイムアウトした。	回復措置は不要です。
	User root logged in by SSH(xxx.xxx.xxx.xxx)	SSH host のユーザが本装置にログイン	回復措置は不要です。
	User root logged in by TELNET	TELNET host のユーザが本装置にログイン	回復措置は不要です。
	OpenFlow session connected. （xxx.xxx.xxx.xxx）	OpenFlow コントローラ（xxx.xxx.xxx.xxx）と接続した。	回復措置は不要です。
	OpenFlow session disconnected. （xxx.xxx.xxx.xxx）	OpenFlow コントローラ（xxx.xxx.xxx.xxx）と切断した。	回復措置は不要です。
	SNTP Corrected TIME. （xxx.xxx.xxx.xxx）	NTP サーバ（xxx.xxx.xxx.xxx）と同期し，時刻を修正しました。	回復措置は不要です。
	SNTP Lost synchronization. （xxx.xxx.xxx.xxx）	NTP サーバ（xxx.xxx.xxx.xxx）と同期していません。	NTP サーバとの通信経路に異常が発生していないかチェックしてください。

Severity	syslog メッセージ	発生条件	対応方法
Informa tional (6) （続き）	Traffic acceleration switched to bypass status. (Peer system IP)	対向装置のシステムインタフェース通信がネットワークポート経由の場合，システムインタフェース宛ての通信をトラフィックアクセラレーションバイパスに切り替えた。	回復措置は不要です。
	Channel does not exist.	コンフィギュレーションにチャネルが存在しない状態で装置の起動が完了しました。	“ンフィギュレーションにチコマンドを実行してチャネルを登録してください。

付録C　SNMP Trap 一覧

SNMP Trap の一覧を表 C-1 に示します。

Trap は有効に設定されているもののみ送出されます。Trap の有効／無効の設定は，“set snmp traps”コマンドを使用して設定します。コマンドの詳細については，「PureFlow WS1 トラフィックシェーパー NF7500 シリーズ コマンドリファレンス」を参照してください。

表 C-1　SNMP Trap 一覧

MIB オブジェクト名	コマンドでの設定名	発生条件	対応方法
coldStart(1.3.6.1.6.3.1.1.5.1)	coldstart	装置起動完了	下記を確認してください。 ・電源断は発生していないか。 ・再起動コマンドを実行していないか。 ・自動リブート機能が働いていないか。
warmStart(1.3.6.1.6.3.1.1.5.2)	warmstart	出力されません。	
linkDown(1.3.6.1.6.3.1.1.5.3)	linkdown	ポートのリンクダウン	下記を確認してください。 ・ケーブルが切断されていないか。 ・正しいケーブル（シングルモード／マルチモード，ストレート／クロス）を使用しているか。 ・Network ポートの Speed/Duplex および Pause の設定が接続装置と整合が取れているか。
linkUp(1.3.6.1.6.3.1.1.5.4)	linkup	リンクアップ	回復措置は不要です。
authenticationFailure(1.3.6.1.6.3.1.1.5.5)	authentication	SNMP の不正アクセス検出	本装置に設定したアクセス許可 community 名，IP address，レベル（get/set）が，SNMP manager 側と整合が取れているか確認してください。
pfGsPowerInsertEvent(1.3.6.1.4.1.1151.2.1.7.20.0.3)	powerinsert	電源ユニットの装着	回復措置は不要です。 NF7501A では送出されません。
pfGsPowerExtractEvent(1.3.6.1.4.1.1151.2.1.7.20.0.4)	powerextract	電源ユニットの抜去	回復措置は不要です。 NF7501A では送出されません。

MIB オブジェクト名	コマンドでの設定名	発生条件	対応方法
pfGsPowerFailureEvent(1.3.6.1.4.1.1151.2.1.7.20.0.5)	powerfailure	電源ユニットの異常検出	下記を確認してください。 ・電源ケーブルは接続されているか。 ・供給電圧は規定内（AC 100 V～AC 127 V/AC 200 V～AC 240 V）か。 NF7501A では送出されません。
pfGsPowerRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.6)	powerrecovery	電源ユニットの異常回復	回復措置は不要です。 NF7501A では送出されません。
pfGsModuleFailureAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.7)	modulefailurealarm	モジュール異常の検出	弊社サポートまでご連絡ください。
pfGsModuleFailureRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.8)	modulefailurerecovery	モジュール異常の回復	回復措置は不要です。
pfGsQueueBuffAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.15)	queuebuffalarm	当該シナリオのパケットバッファ使用量が制限値を超過した。	キューバッファフルのためパケット廃棄が発生しています。入力バースト長の設定をチェックしてください。
pfGsQueueBuffRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.16)	queuebuffrecovery	当該シナリオのパケットバッファ使用量が制限値を超えたあと，制限値の 50%を下回った。	回復措置は不要です。
pfGsSystemBuffAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.17)	systembuffalarm	当該システムバッファのバッファ使用量が 90%を超過した。	トラフィック状況，および各種設定をチェックしてください。
pfGsSystemBuffRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.18)	systembuffrecovery	当該システムバッファのバッファ使用量が 90%を超えたあと，50%を下回った。	回復措置は不要です。
pfGsxSystemHeatAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.19)	systemheatalarm	システム温度が 50℃を超えた，または–5℃を下回った。	環境温度が 40℃以下，および 0℃以上になるように空調または機器配置を見直してください。
pfGsxSystemHeatRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.20)	systemheatrecovery	システム温度が 50℃を超えたあと，45℃を下回った。または–5℃を下回ったあと，0℃を超えた。	回復措置は不要です。
pfGsIndividualQueueAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.21)	queueallocalarm	装置内の個別キューが最大数を超えた。	個別キューが装置の最大数に達しているため最大数超過時のアクションを適用しています。トラフィック状況をチェックしてください。
pfGsIndividualQueueRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.22)	queueallocrecovery	装置内の個別キューが最大数に達したあと，最大数の 90%を下回った。	回復処置は不要です。

MIB オブジェクト名	コマンドでの設定名	発生条件	対応方法
pfGsMaxQnumAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.23)	maxqnumalarm	当該シナリオの個別キュー数が制限値を超過した。	個別キューがシナリオの制限数に達しているため最大数超過時のアクションを適用しています。トラフィック状況をチェックしてください。
pfGsMaxQnumRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.24)	maxqnumrecovery	当該シナリオの個別キューが制限値に達したあと，制限値の 50%を下回った。	回復処置は不要です。
pfGsQueueBuffByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.25)	queuebuffalarm	当該シナリオのパケットバッファ使用量が制限値を超過した。	キューバッファフルのためパケット廃棄が発生しています。入力バースト長の設定をチェックしてください。 NF7501 では送出されません。
pfGsQueueBuffByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.26)	queuebuffrecovery	当該シナリオのパケットバッファ使用量が制限値を超えたあと，制限値の 50%を下回った。	回復措置は不要です。 NF7501 では送出されません。
pfGsMaxQnumByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.27)	maxqnumalarm	当該シナリオの個別キュー数が制限値を超過した。	個別キューがシナリオの制限数に達しているため最大数超過時のアクションを適用しています。トラフィック状況をチェックしてください。
pfGsMaxQnumByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.28)	maxqnumrecovery	当該シナリオの個別キューが制限値に達したあと，制限値の 50%を下回った。	回復処置は不要です。
pfGsTcpAccelBypassByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.29)	tcpbypassalarm	当該 WAN-accel シナリオが通信状態を Bypass に切り替えた。	回復処置は不要です。
pfGsTcpAccelBypassByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.30)	tcpbypassrecovery	当該 WAN-accel シナリオが通信状態を Bypass から切り替えた。	回復処置は不要です。
pfGsTcpAccelPeerByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.31)	peeralarm	当該 WAN-accel シナリオが対向装置を Primary-peer から Secondary-peer に切り替えた。	Primary-peer の回線状況や対向装置の状態を確認してください。
pfGsTcpAccelPeerByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.32)	peerrecovery	当該 WAN-accel シナリオが対向装置を Secondary-peer から Primary-peer に切り戻した。	回復処置は不要です。
pfGsBypassOnEvent (1.3.6.1.4.1.1151.2.1.7.20.0.33)	bypasson	ネットワークバイパス機能が通信経路を Normal 側から切断し，Bypass 側へ接続した。	バイパス状態となった理由を特定し，必要な処置を行ってください。

MIB オブジェクト名	コマンドでの設定名	発生条件	対応方法
pfGsBypassOffEvent (1.3.6.1.4.1.1151.2.1.7.20.0.34)	bypassoff	ネットワークバイパス機能が通信経路を Bypass 側から切断し，Normal 側へ接続した。	回復処置は不要です。
pfGsxFanUnitInsertEvent(1.3.6.1.4.1.1151.2.1.7.20.0.35)	faninsert	ファンユニットの装着	回復処置は不要です。 NF7501A では送出されません。
pfGsxFanUnitExtractEvent(1.3.6.1.4.1.1151.2.1.7.20.0.36)	fanextract	ファンユニットの抜去	回復処置は不要です。 NF7501A では送出されません。
pfGsxFanUnitFailureEvent(1.3.6.1.4.1.1151.2.1.7.20.0.37)	fanfailure	ファンユニットの異常検出	下記を確認してください。 ・ファンは回転しているか。
pfGsxFanUnitRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.38)	fanrecovery	ファンユニットの異常回復	回復処置は不要です。

付録D　Enterprise MIB 一覧

本装置の Enterprise MIB オブジェクト一覧を表 D-1 に示します。

表 D-1　PureFlow WS1 シリーズ Enterprise MIB 一覧

MIB グループ	MIB オブジェクト名	説明
pureFlowGsMib		PureFlow GS Enterprise MIB ツリーです。オブジェクト ID は 1.3.6.1.4.1.1151.2.1.7 です。 以下にツリー内のオブジェクトと，そのオブジェクト ID（カッコ内の値）を示します。 PureFlow GS Enterprise MIB ツリーは PureFlow GS シリーズ共通の MIB ツリーです。本書では PureFlow WS1 シリーズの MIB オブジェクトを示します。
pfGsSystem(1.3.6.1.4.1.1151.2.1.7.1)	pfGsSystemType(1.3.6.1.4.1.1151.2.1.7.1.1)	システムソフトウェアの形名を表します。 nf7500s001a(7)　:NF7500-S001A
	pfGsSystemSlotNumber(1.3.6.1.4.1.1151.2.1.7.1.2)	モジュールを実装するスロットの数を表します。
	pfGsSystemSoftwareRev(1.3.6.1.4.1.1151.2.1.7.1.3)	システムソフトウェアのバージョンを表します。
	pfGsSystemOperationTime(1.3.6.1.4.1.1151.2.1.7.1.5)	装置が起動してからの経過時間を表します。単位は 10 ms です。この MIB オブジェクトは 1 時間ごとに更新されます。したがって，時間以下の単位は常に 0 となります。
	pfGsSystemCcpu5sec(1.3.6.1.4.1.1151.2.1.7.1.6)	制御系処理部の CPU 使用率を，最近 5 秒の平均値で表します。
	pfGsSystemCcpu1min(1.3.6.1.4.1.1151.2.1.7.1.7)	制御系処理部の CPU 使用率を，最近 1 分の平均値で表します。
	pfGsSystemCcpu5min(1.3.6.1.4.1.1151.2.1.7.1.8)	制御系処理部の CPU 使用率を，最近 5 分の平均値で表します。
	pfGsSystemCcpuMemory5sec(1.3.6.1.4.1.1151.2.1.7.1.9)	制御系処理部のメモリ使用率を，最近 5 秒の平均値で表します。
	pfGsSystemCcpuMemory1min(1.3.6.1.4.1.1151.2.1.7.1.10)	制御系処理部のメモリ使用率を，最近 1 分の平均値で表します。
	pfGsSystemCcpuMemory5min(1.3.6.1.4.1.1151.2.1.7.1.11)	制御系処理部のメモリ使用率を，最近 5 分の平均値で表します。
	pfGsSystemFcpuTable(1.3.6.1.4.1.1151.2.1.7.1.12)	フォワーディング系処理部の CPU およびメモリ使用率のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsSystemFcpuEntry(1.3.6.1.4.1.1151.2.1.7.1.12.1)	フォワーディング系処理部の CPU およびメモリ使用率のエントリテーブルです。テーブルインデックスは pfSystemFcpuIndex です。 このテーブルには以下のオブジェクトが含まれています。

MIB グループ	MIB オブジェクト名	説明
pfGsSystem(1.3.6.1.4.1.1151.2.1.7.1)（続き）	pfGsSystemFcpuIndex(1.3.6.1.4.1.1151.2.1.7.1.12.1.1)	フォワーディング系処理部の番号を表します。 正面図 1
	pfGsSystemFcpu5sec(1.3.6.1.4.1.1151.2.1.7.1.12.1.2)	フォワーディング系処理部の CPU 使用率を，最近 5 秒の平均値で表します。
	pfGsSystemFcpu1min(1.3.6.1.4.1.1151.2.1.7.1.12.1.3)	フォワーディング系処理部の CPU 使用率を，最近 1 分の平均値で表します。
	pfGsSystemFcpu5min(1.3.6.1.4.1.1151.2.1.7.1.12.1.4)	フォワーディング系処理部の CPU 使用率を，最近 5 分の平均値で表します。
	pfGsSystemFcpuMemory5sec(1.3.6.1.4.1.1151.2.1.7.1.12.1.5)	フォワーディング系処理部のメモリ使用率を，最近 5 秒の平均値で表します。
	pfGsSystemFcpuMemory1min(1.3.6.1.4.1.1151.2.1.7.1.12.1.6)	フォワーディング系処理部のメモリ使用率を，最近 1 分の平均値で表します。
	pfGsSystemFcpuMemory5min(1.3.6.1.4.1.1151.2.1.7.1.12.1.7)	フォワーディング系処理部のメモリ使用率を，最近 5 分の平均値で表します。
	pfGsSystemBuffTable(1.3.6.1.4.1.1151.2.1.7.1.13)	システムバッファのテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsSystemBuffEntry(1.3.6.1.4.1.1151.2.1.7.1.13.1)	システムバッファのエントリテーブルです。テーブルインデックスは pfGsSystemBuffIndex です。 このテーブルには以下のオブジェクトが含まれています。
	pfGsSystemBuffIndex(1.3.6.1.4.1.1151.2.1.7.1.13.1.1)	システムバッファの番号を表します。 1 :パケットバッファ 2 :TCP WARP エンジンのメッセージブロック 3 :パケット出力コマンド領域 4 :インバンドで送信するパケットのパケットバッファ 5 :未使用 6 :未使用 7 :未使用 8 :未使用 9 :処理中のパケットの一時領域
	pfGsSystemBuffMax(1.3.6.1.4.1.1151.2.1.7.1.13.1.2)	システムバッファの最大容量を表します。
	pfGsSystemBuffRemaining(1.3.6.1.4.1.1151.2.1.7.1.13.1.3)	システムバッファの残容量を表します。
	pfGsSystemTempTable(1.3.6.1.4.1.1151.2.1.7.1.14)	システム温度のテーブルです。 このテーブルには以下のオブジェクトが含まれています。

MIB グループ	MIB オブジェクト名	説明
pfGsSystem(1.3.6.1.4.1.1151.2.1.7.1)(続き)	pfGsSystemTempEntry(1.3.6.1.4.1.1151.2.1.7.1.14.1)	システム温度のエントリテーブルです。テーブルインデックスは pfGsSystemTempIndex です。 このテーブルには以下のオブジェクトが含まれています。
	pfGsSystemTempIndex(1.3.6.1.4.1.1151.2.1.7.1.14.1.1)	システム温度の番号を表します。 1 :吸気 2 :未使用 3 :未使用 4 :未使用 5 :未使用 6 :未使用 7 :未使用 8 :未使用 9 :未使用
	pfGsSystemTempValue(1.3.6.1.4.1.1151.2.1.7.1.14.1.2)	システム温度の値を表します。 単位は摂氏です。
	pfGsSystemBypassMode(1.3.6.1.4.1.1151.2.1.7.1.15)	ネットワークバイパス機能の制御モードを表します。 notAvailable(0) :このシステムではネットワークバイパス機能を利用できません auto (1) :自動制御 on (2) :強制バイパス off (3) :強制非バイパス
	pfGsSystemBypassState(1.3.6.1.4.1.1151.2.1.7.1.16)	ネットワークバイパスの状態を表します。 notAvailable(0) :このシステムではネットワークバイパス機能を利用できません on (1) :バイパス状態 off (2) :非バイパス状態
	pfGsSystemBypassTimeRemaining(1.3.6.1.4.1.1151.2.1.7.1.17)	一時的なバイパス切り替えの残り時間を秒単位で表します。一時的なバイパス切り替えが実行中でない場合は 0 秒を表示します。
pfGsModule(1.3.6.1.4.1.1151.2.1.7.2)	pfGsModuleTable(1.3.6.1.4.1.1151.2.1.7.2.1)	モジュール情報のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsModuleEntry(1.3.6.1.4.1.1151.2.1.7.2.1.1)	モジュール情報のエントリテーブルです。テーブルインデックスは pfGsModuleIndex です。 このテーブルには以下のオブジェクトが含まれています。
	pfGsModuleIndex(1.3.6.1.4.1.1151.2.1.7.2.1.1.1)	モジュールの番号を表します。 正面図 1
	pfGsModuleLocation(1.3.6.1.4.1.1151.2.1.7.2.1.1.2)	モジュールの実装スロット番号を表します。 (モジュール番号と同じ値になります) 正面図 1

MIB グループ	MIB オブジェクト名	説明
pfGsModule(1.3.6.1.4.1.1151.2.1.7.2) （続き）	pfGsModuleType(1.3.6.1.4.1.1151.2.1.7.2.1.1.3)	モジュールの種別を表します。 unknown(1) :下記以外 empty(2) :未実装 ge2gt(3) :GbE/2T fe2ft(4) :FE/2T xge2sfp(5) :10GbE/2SFP+ xge4sfp(6) :10GbE/4SFP+ ge4sfp(7) :GbE/4SFP ge2gt4sfp(8) :GbE/2T, GbE/4SFP
	pfGsModuleDescr(1.3.6.1.4.1.1151.2.1.7.2.1.1.4)	モジュールの名前を表します。
	pfGsModulePortNumber(1.3.6.1.4.1.1151.2.1.7.2.1.1.5)	モジュールの実装ポート数を表します。
	pfGsModuleOperStatus(1.3.6.1.4.1.1151.2.1.7.2.1.1.6)	モジュールの状態を表します。 other(1) :下記以外 operational(2) :正常 malfunctioning(3) :6 以外の異常 notPresent(4) :未実装 standby(5) :（未使用です） notResponding(6) :応答なし
	pfGsModuleRevision(1.3.6.1.4.1.1151.2.1.7.2.1.1.7)	モジュールのハードウェアレビジョンを表します。
	pfGsModuleSerialNumber(1.3.6.1.4.1.1151.2.1.7.2.1.1.8)	モジュールのシリアル番号を表します。
pfGsPower(1.3.6.1.4.1.1151.2.1.7.3)	pfGsPowerTable(1.3.6.1.4.1.1151.2.1.7.3.1)	電源ユニット情報のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsPowerEntry(1.3.6.1.4.1.1151.2.1.7.3.1.1)	電源ユニット情報のエントリテーブルです。テーブルインデックスは pfGsPowerIndex です。 このテーブルには以下のオブジェクトが含まれています。
	pfGsPowerIndex(1.3.6.1.4.1.1151.2.1.7.3.1.1.1)	電源ユニットの番号を表します。 背面図 Power1 / FanUnit 1 / FanDev 1 / FanDev 2
	pfGsPowerOperStatus(1.3.6.1.4.1.1151.2.1.7.3.1.1.2)	電源ユニットの状態を表します。 other(1) :下記以外 operational(2) :正常 malfunctioning(3) :異常（入力異常またはファン停止） notPresent(4) :未実装 outputerror(5) :（未使用です） inputerror(6) :（未使用です） fanfailure(7) :（未使用です）

MIB グループ	MIB オブジェクト名	説明
pfGsPower(1.3.6.1.4.1.1151.2.1.7.3) (続き)	pfGsPowerUpTime(1.3.6.1.4.1.1151.2.1.7.3.1.1.3)	電源ユニットが装着されてからの経過時間を表します。単位は 10 ms です。
pfGsFlowInformation(1.3.6.1.4.1.1151.2.1.7.8)	pfGsFlowInformationResourceTotal(1.3.6.1.4.1.1151.2.1.7.8.1)	装置で使用可能なフロー数の総数を表示します。
	pfGsFlowInformationResourceUsed(1.3.6.1.4.1.1151.2.1.7.8.2)	装置で使用中のフロー数を表示します。
	pfGsFlowInformationResourceAvailable(1.3.6.1.4.1.1151.2.1.7.8.3)	装置で使用前のフロー数を表示します。
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9)	pfGsxScenarioStatisticsTable(1.3.6.1.4.1.1151.2.1.7.9.1)	シナリオカウンタのテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxScenarioStatisticsEntry(1.3.6.1.4.1.1151.2.1.7.9.1.1)	シナリオカウンタのエントリテーブルです。テーブルインデックスは pfGsxScenarioStatisticsScenarioSortIndex です。 このテーブルには以下のオブジェクトが含まれています。 参考)　このテーブル内オブジェクトの OID を求める方法をこの表の次に示します。
	pfGsxScenarioStatisticsScenarioSortIndex(1.3.6.1.4.1.1151.2.1.7.9.1.1.1)	シナリオのソート番号を表します。 ソート番号はシナリオ登録／削除時に自動付加されます。 シナリオツリーの並び順に対応した番号になります。
	pfGsxScenarioStatisticsScenarioName(1.3.6.1.4.1.1151.2.1.7.9.1.1.2)	シナリオのシナリオ名を表します。
	pfGsxScenarioStatisticsScenarioType(1.3.6.1.4.1.1151.2.1.7.9.1.1.3)	シナリオのタイプを表します。 discard(0)　:廃棄シナリオ individual(1)　:個別キューシナリオ aggregate(2)　:集約キューシナリオ application(3)　:(未使用です) wanaccel(4)　:トラフィックアクセラレーションシナリオ forward(5)　:転送シナリオ
	pfGsxScenarioStatisticsRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.4)	シナリオの受信オクテット数を表します。
	pfGsxScenarioStatisticsRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.5)	シナリオの受信パケット数を表します。
	pfGsxScenarioStatisticsTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.6)	シナリオの送信オクテット数を表します。
	pfGsxScenarioStatisticsTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.7)	シナリオの送信パケット数を表します。
	pfGsxScenarioStatisticsDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.8)	シナリオの廃棄オクテット数を表します。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9) （続き）	pfGsxScenarioStatisticsDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.9)	シナリオの廃棄パケット数を表します。
	pfGsxScenarioStatisticsHCRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.10)	シナリオの受信オクテット数を 64 ビットで表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsHCRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.11)	シナリオの受信パケット数を 64 ビットで表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsHCTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.12)	シナリオの送信オクテット数を 64 ビットで表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsHCTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.13)	シナリオの送信パケット数を 64 ビットで表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.14)	シナリオの廃棄オクテット数を 64 ビットで表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.15)	シナリオの廃棄パケット数を 64 ビットで表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsDefaultQueRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.16)	シナリオのデフォルトキューの受信オクテット数を表します。
	pfGsxScenarioStatisticsDefaultQueRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.17)	シナリオのデフォルトキューの受信パケット数を表します。
	pfGsxScenarioStatisticsDefaultQueTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.18)	シナリオのデフォルトキューの送信オクテット数を表します。
	pfGsxScenarioStatisticsDefaultQueTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.19)	シナリオのデフォルトキューの送信パケット数を表します。
	pfGsxScenarioStatisticsDefaultQueDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.20)	シナリオのデフォルトキューの廃棄オクテット数を表します。
	pfGsxScenarioStatisticsDefaultQueDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.21)	シナリオのデフォルトキューの廃棄パケット数を表します。
	pfGsxScenarioStatisticsDefaultQueHCRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.22)	シナリオのデフォルトキューの受信オクテット数を表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsDefaultQueHCRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.23)	シナリオのデフォルトキューの受信パケット数を表します。 注)　このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9) (続き)	pfGsxScenarioStatisticsDefaultQueHCTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.24)	シナリオのデフォルトキューの送信オクテット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsDefaultQueHCTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.25)	シナリオのデフォルトキューの送信パケット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsDefaultQueHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.26)	シナリオのデフォルトキューの廃棄オクテット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatisticsDefaultQueHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.27)	シナリオのデフォルトキューの廃棄パケット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10)	pfGsxScenarioInformationTable(1.3.6.1.4.1.1151.2.1.7.10.1)	シナリオ情報のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxScenarioInformationEntry(1.3.6.1.4.1.1151.2.1.7.10.1.1)	シナリオ情報のエントリテーブルです。テーブルインデックスは pfGsxScenarioInformationScenarioSortIndex です。 このテーブルには以下のオブジェクトが含まれています。 参考)このテーブル内オブジェクトの OID を求める方法をこの表の次に示します。
	pfGsxScenarioInformationScenarioSortIndex(1.3.6.1.4.1.1151.2.1.7.10.1.1.1)	シナリオのソート番号を表します。 ソート番号はシナリオ登録／削除時に自動付加されます。 シナリオツリーの並び順に対応した番号になります。
	pfGsxScenarioInformationScenarioName(1.3.6.1.4.1.1151.2.1.7.10.1.1.2)	シナリオのシナリオ名を表します。
	pfGsxScenarioInformationScenarioType(1.3.6.1.4.1.1151.2.1.7.10.1.1.3)	シナリオのタイプを表します。 discard(0) :廃棄シナリオ individual(1) :個別キューシナリオ aggregate(2) :集約キューシナリオ application(3) :(未使用です) wanaccel(4) :トラフィックアクセラレーションシナリオ forward(5) :転送シナリオ
	pfGsxScenarioInformationDefFlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.4)	シナリオに関連して生成されたデフォルトフローの数を表します。forward シナリオの場合はフローの総数と同じ値になります。
	pfGsxScenarioInformationClass1FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.5)	シナリオに関連して生成されたクラス 1 フローの数を表します。 注)未サポートです。値は 0 固定です。
	pfGsxScenarioInformationClass2FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.6)	シナリオに関連して生成されたクラス 2 フローの数を表します。 注)未サポートです。値は 0 固定です。
	pfGsxScenarioInformationClass3FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.7)	シナリオに関連して生成されたクラス 3 フローの数を表します。 注)未サポートです。値は 0 固定です。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10)（続き）	pfGsxScenarioInformationClass4FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.8)	シナリオに関連して生成されたクラス 4 フローの数を表します。 注）未サポートです。値は 0 固定です。
	pfGsxScenarioInformationClass5FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.9)	シナリオに関連して生成されたクラス 5 フローの数を表します。 注）未サポートです。値は 0 固定です。
	pfGsxScenarioInformationClass6FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.10)	シナリオに関連して生成されたクラス 6 フローの数を表します。 注）未サポートです。値は 0 固定です。
	pfGsxScenarioInformationClass7FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.11)	シナリオに関連して生成されたクラス 7 フローの数を表します。 注）未サポートです。値は 0 固定です。
	pfGsxScenarioInformationClass8FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.12)	シナリオに関連して生成されたクラス 8 フローの数を表します。 注）未サポートです。値は 0 固定です。
	pfGsxScenarioInformationTotalFlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.13)	シナリオに関連して生成されたフローの総数を表します。
	pfGsxScenarioInformationMaxBuffScenarioId(1.3.6.1.4.1.1151.2.1.7.10.1.1.14)	現在のバッファ使用量が最大のキューについて，該当するキューの QID を示します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationMaxBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.15)	現在のバッファ使用量が最大のキューについて，該当するキューの最大バッファサイズに対するバッファ使用率を表します。単位は％です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationMaxBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.16)	現在のバッファ使用量が最大のキューについて，該当するキューのバッファ使用量を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationMinBuffScenarioId(1.3.6.1.4.1.1151.2.1.7.10.1.1.17)	現在のバッファ使用量が最小のキューについて，該当するキューの QID を示します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationMinBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.18)	現在のバッファ使用量が最小のキューについて，該当するキューの最大バッファサイズに対するバッファ使用率を表します。単位は％です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationMinBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.19)	現在のバッファ使用量が最小のキューについて，該当するキューのバッファ使用量を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationAveBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.20)	現在のバッファ使用率の平均値を表します。単位は％です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationAveBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.21)	現在のバッファ使用量の平均値を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10) (続き)	pfGsxScenarioInformationPeakBuffScenarioId(1.3.6.1.4.1.1151.2.1.7.10.1.1.22)	今までに割り当てたキューの中で，バッファ使用量ピークが最大のキューについて，該当するキューの QID を示します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationPeakBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.23)	今までに割り当てたキューの中で，バッファ使用量ピークが最大のキューについて，該当するキューの最大バッファサイズに対するバッファ使用率を表します。単位は%です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationPeakBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.24)	今までに割り当てたキューの中で，バッファ使用量ピークが最大のキューについて，該当するキューのバッファ使用量を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationDefBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.25)	シナリオのデフォルトキューの，現在のバッファ使用率を表します。単位は%です。
	pfGsxScenarioInformationDefBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.26)	シナリオのデフォルトキューの，現在のバッファ使用量を表します。単位はバイトです。
	pfGsxScenarioInformationDefPeakBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.27)	シナリオのデフォルトキューの，現在のバッファ使用率ピークを表します。単位は%です。
	pfGsxScenarioInformationDefPeakBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.28)	シナリオのデフォルトキューの，現在のバッファ使用量ピークを表します。単位はバイトです。
	pfGsxScenarioInformationTxPeakRateBps(1.3.6.1.4.1.1151.2.1.7.10.1.1.29)	シナリオの直近 1 分間の送信レートピークを表します。単位は bit/s です。 注) このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioInformationTxAveRateBps(1.3.6.1.4.1.1151.2.1.7.10.1.1.31)	シナリオの直近 1 分間の送信レート平均を表します。単位は bit/s です。 注) このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioInformationIndQueNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.33)	個別キューモードシナリオの現在の個別キュー数を表します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationAccelSessNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.34)	アクセラレーションモードシナリオの現在の WAN 高速化を適用している TCP セッションの数を表します。 アクセラレーションモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInformationAccelBypassStatus(1.3.6.1.4.1.1151.2.1.7.10.1.1.35)	アクセラレーションモードシナリオの現在のバイパス状態を表します。
	pfGsxScenarioInformationAccelActivePeer(1.3.6.1.4.1.1151.2.1.7.10.1.1.36)	アクセラレーションモードシナリオの現在のアクティブ Peer 情報を表します。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11)	pfGsxScenarioStatByScIdTable(1.3.6.1.4.1.1151.2.1.7.11.1)	シナリオカウンタのテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxScenarioStatByScIdEntry(1.3.6.1.4.1.1151.2.1.7.11.1.1)	シナリオカウンタのエントリテーブルです。テーブルインデックスは pfGsxScenarioStatByScIdScenarioId です。 このテーブルには以下のオブジェクトが含まれています。 参考）このテーブル内オブジェクトの OID を求める方法をこの表の次に示します。
	pfGsxScenarioStatByScIdScenarioId(1.3.6.1.4.1.1151.2.1.7.11.1.1.1)	シナリオのシナリオ ID を表します。 シナリオ ID はシナリオ登録時に指定可能です。 シナリオ登録時にシナリオ ID の指定を省略した場合，シナリオ ID は自動割り当てされます。
	pfGsxScenarioStatByScIdScenarioName(1.3.6.1.4.1.1151.2.1.7.11.1.1.2)	シナリオのシナリオ名を表します。
	pfGsxScenarioStatByScIdScenarioType(1.3.6.1.4.1.1151.2.1.7.11.1.1.3)	シナリオのタイプを表します。 discard(0) :廃棄シナリオ individual(1) :個別キューシナリオ aggregate(2) :集約キューシナリオ application(3) :（未使用です） wanaccel(4) :トラフィックアクセラレーションシナリオ forward(5) :転送シナリオ
	pfGsxScenarioStatByScIdRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.4)	シナリオの受信オクテット数を表します。
	pfGsxScenarioStatByScIdRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.5)	シナリオの受信パケット数を表します。
	pfGsxScenarioStatByScIdTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.6)	シナリオの送信オクテット数を表します。
	pfGsxScenarioStatByScIdTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.7)	シナリオの送信パケット数を表します。
	pfGsxScenarioStatByScIdDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.8)	シナリオの廃棄オクテット数を表します。
	pfGsxScenarioStatByScIdDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.9)	シナリオの廃棄パケット数を表します。
	pfGsxScenarioStatByScIdHCRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.10)	シナリオの受信オクテット数を 64 ビットで表します。 注）このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdHCRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.11)	シナリオの受信パケット数を 64 ビットで表します。 注）このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdHCTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.12)	シナリオの送信オクテット数を 64 ビットで表します。 注）このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11)（続き）	pfGsxScenarioStatByScIdHCTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.13)	シナリオの送信パケット数を 64 ビットで表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.14)	シナリオの廃棄オクテット数を 64 ビットで表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.15)	シナリオの廃棄パケット数を 64 ビットで表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdDefaultQueRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.16)	シナリオのデフォルトキューの受信オクテット数を表します。
	pfGsxScenarioStatByScIdDefaultQueRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.17)	シナリオのデフォルトキューの受信パケット数を表します。
	pfGsxScenarioStatByScIdDefaultQueTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.18)	シナリオのデフォルトキューの送信オクテット数を表します。
	pfGsxScenarioStatByScIdDefaultQueTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.19)	シナリオのデフォルトキューの送信パケット数を表します。
	pfGsxScenarioStatByScIdDefaultQueDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.20)	シナリオのデフォルトキューの廃棄オクテット数を表します。
	pfGsxScenarioStatByScIdDefaultQueDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.21)	シナリオのデフォルトキューの廃棄パケット数を表します。
	pfGsxScenarioStatByScIdDefaultQueHCRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.22)	シナリオのデフォルトキューの受信オクテット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdDefaultQueHCRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.23)	シナリオのデフォルトキューの受信パケット数を表します。 注) このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdDefaultQueHCTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.24)	シナリオのデフォルトキューの送信オクテット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdDefaultQueHCTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.25)	シナリオのデフォルトキューの送信パケット数を表します。 注)このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11) (続き)	pfGsxScenarioStatByScIdDefaultQueHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.26)	シナリオのデフォルトキューの廃棄オクテット数を表します。 注) このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioStatByScIdDefaultQueHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.27)	シナリオのデフォルトキューの廃棄パケット数を表します。 注) このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12)	pfGsxScenarioInfoByScIdTable(1.3.6.1.4.1.1151.2.1.7.12.1)	シナリオ情報のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxScenarioInfoByScIdEntry(1.3.6.1.4.1.1151.2.1.7.12.1.1)	シナリオ情報のエントリテーブルです。テーブルインデックスは pfGsxScenarioInfoByScIdScenarioId です。 このテーブルには以下のオブジェクトが含まれています。 参考) このテーブル内オブジェクトの OID を求める方法をこの表の次に示します。
	pfGsxScenarioInfoByScIdScenarioId(1.3.6.1.4.1.1151.2.1.7.12.1.1.1)	シナリオのシナリオ ID を表します。 シナリオ ID はシナリオ登録時に指定可能です。 シナリオ登録時にシナリオ ID の指定を省略した場合，シナリオ ID は自動割り当てされます。
	pfGsxScenarioInfoByScIdScenarioName(1.3.6.1.4.1.1151.2.1.7.12.1.1.2)	シナリオのシナリオ名を表します。
	pfGsxScenarioInfoByScIdScenarioType(1.3.6.1.4.1.1151.2.1.7.12.1.1.3)	シナリオのタイプを表します。 discard(0) ：廃棄シナリオ individual(1) ：個別キューシナリオ aggregate(2) ：集約キューシナリオ application(3) ：(未使用です) wanaccel(4) ：トラフィックアクセラレーションシナリオ forward(5) ：転送シナリオ
	pfGsxScenarioInfoByScIdDefFlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.4)	シナリオに関連して生成されたデフォルトフローの数を表します。forward シナリオの場合はフローの総数と同じ値になります。
	pfGsxScenarioInfoByScIdClass1FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.5)	シナリオに関連して生成されたクラス 1 フローの数を表します。 注) 未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdClass2FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.6)	シナリオに関連して生成されたクラス 2 フローの数を表します。 注) 未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdClass3FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.7)	シナリオに関連して生成されたクラス 3 フローの数を表します。 注) 未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdClass4FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.8)	シナリオに関連して生成されたクラス 4 フローの数を表します。 注) 未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdClass5FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.9)	シナリオに関連して生成されたクラス 5 フローの数を表します。 注) 未サポートです。値は 0 固定です。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12) (続き)	pfGsxScenarioInfoByScIdClass6FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.10)	シナリオに関連して生成されたクラス 6 フローの数を表します。 注)未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdClass7FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.11)	シナリオに関連して生成されたクラス 7 フローの数を表します。 注)未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdClass8FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.12)	シナリオに関連して生成されたクラス 8 フローの数を表します。 注)未サポートです。値は 0 固定です。
	pfGsxScenarioInfoByScIdTotalFlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.13)	シナリオに関連して生成されたフローの総数を表します。
	pfGsxScenarioInfoByScIdMaxBuffScenarioId(1.3.6.1.4.1.1151.2.1.7.12.1.1.14)	現在のバッファ使用量が最大のキューについて，該当するキューの QID を示します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdMaxBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.15)	現在のバッファ使用量が最大のキューについて，該当するキューの最大バッファサイズに対するバッファ使用率を表します。単位は%です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdMaxBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.16)	現在のバッファ使用量が最大のキューについて，該当するキューのバッファ使用量を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdMinBuffScenarioId(1.3.6.1.4.1.1151.2.1.7.12.1.1.17)	現在のバッファ使用量が最小のキューについて，該当するキューの QID を示します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdMinBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.18)	現在のバッファ使用量が最小のキューについて，該当するキューの最大バッファサイズに対するバッファ使用率を表します。単位は%です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdMinBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.19)	現在のバッファ使用量が最小のキューについて，該当するキューのバッファ使用量を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdAveBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.20)	現在のバッファ使用率の平均値を表します。単位は%です。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdAveBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.21)	現在のバッファ使用量の平均値を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdPeakBuffScenarioId(1.3.6.1.4.1.1151.2.1.7.12.1.1.22)	今までに割り当てたキューの中で，バッファ使用量ピークが最大のキューについて，該当するキューの QID を示します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdPeakBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.23)	今までに割り当てたキューの中で，バッファ使用量ピークが最大のキューについて，該当するキューの最大バッファサイズに対するバッファ使用率を表します。単位は%です。 個別キューモード以外のシナリオでは 0 固定です。

MIB グループ	MIB オブジェクト名	説明
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12) （続き）	pfGsxScenarioInfoByScIdPeakBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.24)	今までに割り当てたキューの中で，バッファ使用量ピークが最大のキューについて，該当するキューのバッファ使用量を表します。単位はバイトです。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdDefBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.25)	シナリオのデフォルトキューの，現在のバッファ使用率を表します。単位は%です。
	pfGsxScenarioInfoByScIdDefBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.26)	シナリオのデフォルトキューの，現在のバッファ使用量を表します。単位はバイトです。
	pfGsxScenarioInfoByScIdDefPeakBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.27)	シナリオのデフォルトキューの，現在のバッファ使用率ピークを表します。単位は%です。
	pfGsxScenarioInfoByScIdDefPeakBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.28)	シナリオのデフォルトキューの，現在のバッファ使用量ピークを表します。単位はバイトです。
	pfGsxScenarioInfoByScIdTxPeakRateBps(1.3.6.1.4.1.1151.2.1.7.12.1.1.29)	シナリオの直近 1 分間の送信レートピークを表します。単位は bit/s です。 注）このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioInfoByScIdTxAveRateBps(1.3.6.1.4.1.1151.2.1.7.12.1.1.31)	シナリオの直近 1 分間の送信レート平均を表します。単位は bit/s です。 注）このオブジェクトに SNMPv1 でアクセスすることはできません。v2c 以上でアクセスしてください。
	pfGsxScenarioInfoByScIdIndQueNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.33)	個別キューモードシナリオの現在の個別キュー数を表します。 個別キューモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdAccelSessNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.34)	アクセラレーションモードシナリオの現在の WAN 高速化を適用している TCP セッションの数を表します。 アクセラレーションモード以外のシナリオでは 0 固定です。
	pfGsxScenarioInfoByScIdAccelBypassStatus(1.3.6.1.4.1.1151.2.1.7.12.1.1.35)	アクセラレーションモードシナリオの現在のバイパス状態を表します。
	pfGsxScenarioInfoByScIdAccelActivePeer(1.3.6.1.4.1.1151.2.1.7.12.1.1.36)	アクセラレーションモードシナリオの現在のアクティブ Peer 情報を表します。
pfGsxFanUnit(1.3.6.1.4.1.1151.2.1.7.13)	pfGsxFanUnitTable(1.3.6.1.4.1.1151.2.1.7.13.1)	複数のファンデバイスが組み込まれたファンユニット情報のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxFanUnitEntry(1.3.6.1.4.1.1151.2.1.7.13.1.1)	複数のファンデバイスが組み込まれたファンユニット情報のエントリテーブルです。テーブルインデックスは pfGsxFanUnitIndex です。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxFanUnitIndex(1.3.6.1.4.1.1151.2.1.7.13.1.1.1)	ファンユニットの番号を表します。 背面図 Power1 / FanUnit 1 / FanDev 2 / FanDev 1

MIB グループ	MIB オブジェクト名	説明
pfGsxFanUnit(1.3.6.1.4.1.1151.2.1.7.13) （続き）	pfGsxFanUnitOperStatus(1.3.6.1.4.1.1151.2.1.7.13.1.1.2)	ファンユニットの状態を表します。 other(1)　　:下記以外 operational(2)　　:正常 malfunctioning(3)　:異常（ファン停止） notPresent(4)　　:未実装
	pfGsxFanUnitUpTime(1.3.6.1.4.1.1151.2.1.7.13.1.1.3)	ファンユニットが装着されてからの経過時間を表します。 単位は 10 ms です。
pfGsxFanDevice(1.3.6.1.4.1.1151.2.1.7.14)	pfGsxFanDeviceTable(1.3.6.1.4.1.1151.2.1.7.14.1)	ファンデバイス情報のテーブルです。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxFanDeviceEntry(1.3.6.1.4.1.1151.2.1.7.14.1.1)	ファンデバイス情報のエントリテーブルです。テーブルインデックスは pfGsxFanDeviceUnitIndex と pfGsxFanDeviceIndex です。 このテーブルには以下のオブジェクトが含まれています。
	pfGsxFanDeviceUnitIndex(1.3.6.1.4.1.1151.2.1.7.14.1.1.1)	ファンデバイスが組み込まれているファンユニットの番号を表します。 背面図 Power1　FanUnit 1　FanDev 2　FanDev 1
	pfGsxFanDeviceIndex(1.3.6.1.4.1.1151.2.1.7.14.1.1.2)	ファンデバイスの番号を表します。 背面図 Power1　FanUnit 1　FanDev 2　FanDev 1
	pfGsxFanDeviceSpeed(1.3.6.1.4.1.1151.2.1.7.14.1.1.3)	ファンデバイスのファンの回転数を表します。単位はRPM です。

参考）

シナリオカウンタ，シナリオインフォメーションテーブルの OID を求める方法

テーブル内オブジェクトの OID を求めるには以下を参考にしてください。

pfGsxScenarioStatisticsTable の場合
pfGsxScenarioStatisticsEntry の OID は次のようになります。
1.3.6.1.4.1.1151.2.1.7.9.1.1.EntryOID.ScenarioSortIndex
固定値

EntryOID ：テーブル内エントリの番号です。表 4 の順序通りに 1 から並んでいます。長さは 1 です。

pfGsxScenarioStatisticsScenarioSortIndex	1
pfGsxScenarioStatisticsScenarioName	2
pfGsxScenarioStatisticsScenarioType	3
pfGsxScenarioStatisticsRxOctets	4
pfGsxScenarioStatisticsRxPackets	5
pfGsxScenarioStatisticsTxOctets	6
pfGsxScenarioStatisticsTxPackets	7
pfGsxScenarioStatisticsDiscardOctets	8
pfGsxScenarioStatisticsDiscardPackets	9
pfGsxScenarioStatisticsHCRxOctets	10
pfGsxScenarioStatisticsHCRxPackets	11
pfGsxScenarioStatisticsHCTxOctets	12
pfGsxScenarioStatisticsHCTxPackets	13
pfGsxScenarioStatisticsHCDiscardOctets	14
pfGsxScenarioStatisticsHCDiscardPackets	15
pfGsxScenarioStatisticsDefaultQueRxOctets	16
pfGsxScenarioStatisticsDefaultQueRxPackets	17
pfGsxScenarioStatisticsDefaultQueTxOctets	18
pfGsxScenarioStatisticsDefaultQueTxPackets	19
pfGsxScenarioStatisticsDefaultQueDiscardOctets	20
pfGsxScenarioStatisticsDefaultQueDiscardPackets	21
pfGsxScenarioStatisticsDefaultQueHCRxOctets	22
pfGsxScenarioStatisticsDefaultQueHCRxPackets	23
pfGsxScenarioStatisticsDefaultQueHCTxOctets	24
pfGsxScenarioStatisticsDefaultQueHCTxPackets	25
pfGsxScenarioStatisticsDefaultQueHCDiscardOctets	26
pfGsxScenarioStatisticsDefaultQueHCDiscardPackets	27

ScenarioSortIndex：　シナリオのソート番号です。長さは 8 です。ソート番号はシナリオツリーの並び順に対応した番号を表し，シナリオ登録／削除時に自動割り当てされます。シナリオ登録／削除の度に再割り当てされますので，シナリオ構成を変えるとソート番号も変化します。特定シナリオのソート番号を求めるには，シナリオ構成が決定された状態で pfGsxScenarioStatisticsTable を getnext で全取得し，シナリオ名をキーにして求めるエントリを探してください。

pfGsxScenarioInformationTable の場合
pfGsxScenarioInformationEntry の OID は次のようになります。
1.3.6.1.4.1.1151.2.1.7.10.1.1.EntryOID.ScenarioSortIndex
固定値
EntryOID:テーブル内エントリの番号です。番号は連続していませんので注意してください。長さは 1 です。

pfGsxScenarioInformationScenarioSortIndex	1
pfGsxScenarioInformationScenarioName	2
pfGsxScenarioInformationScenarioType	3
pfGsxScenarioInformationDefFlowNum	4
pfGsxScenarioInformationTotalFlowNum	13
pfGsxScenarioInformationMaxBuffScenarioId	14
pfGsxScenarioInformationMaxBuffRatio	15
pfGsxScenarioInformationMaxBuff	16
pfGsxScenarioInformationMinBuffScenarioId	17
pfGsxScenarioInformationMinBuffRatio	18
pfGsxScenarioInformationMinBuff	19
pfGsxScenarioInformationAveBuffRatio	20
pfGsxScenarioInformationAveBuff	21
pfGsxScenarioInformationPeakBuffScenarioId	22
pfGsxScenarioInformationPeakBuffRatio	23
pfGsxScenarioInformationPeakBuff	24
pfGsxScenarioInformationDefBuffRatio	25
pfGsxScenarioInformationDefBuff	26
pfGsxScenarioInformationDefPeakBuffRatio	27
pfGsxScenarioInformationDefPeakBuff	28
pfGsxScenarioInformationTxPeakRateBps	29
pfGsxScenarioInformationTxAveRateBps	31
pfGsxScenarioInformationIndQueNum	33
pfGsxScenarioInformationAccelSessNum	34
pfGsxScenarioInformationAccelBypassStatus	35
pfGsxScenarioInformationAccelActivePeer	36

ScenarioSortIndex: シナリオのソート番号です。長さは 8 です。求め方は pfGsxScenarioStatisticsTable のソート番号と同様です。

pfGsxScenarioStatByScIdTable の場合
pfGsxScenarioStatByScIdEntry の OID は次のようになります。
1.3.6.1.4.1.1151.2.1.7.11.1.1.EntryOID.ScenarioId
固定値
EntryOID:テーブル内エントリの番号です。表 4 の順序通りに 1 から並んでいます。長さは 1 です。

pfGsxScenarioStatByScIdScenarioId	1
pfGsxScenarioStatByScIdScenarioName	2
pfGsxScenarioStatByScIdScenarioType	3
pfGsxScenarioStatByScIdRxOctets	4
pfGsxScenarioStatByScIdRxPackets	5
pfGsxScenarioStatByScIdTxOctets	6
pfGsxScenarioStatByScIdTxPackets	7
pfGsxScenarioStatByScIdDiscardOctets	8
pfGsxScenarioStatByScIdDiscardPackets	9
pfGsxScenarioStatByScIdHCRxOctets	10
pfGsxScenarioStatByScIdHCRxPackets	11
pfGsxScenarioStatByScIdHCTxOctets	12
pfGsxScenarioStatByScIdHCTxPackets	13
pfGsxScenarioStatByScIdHCDiscardOctets	14
pfGsxScenarioStatByScIdHCDiscardPackets	15
pfGsxScenarioStatByScIdDefaultQueRxOctets	16
pfGsxScenarioStatByScIdDefaultQueRxPackets	17
pfGsxScenarioStatByScIdDefaultQueTxOctets	18
pfGsxScenarioStatByScIdDefaultQueTxPackets	19
pfGsxScenarioStatByScIdDefaultQueDiscardOctets	20
pfGsxScenarioStatByScIdDefaultQueDiscardPackets	21
pfGsxScenarioStatByScIdDefaultQueHCRxOctets	22
pfGsxScenarioStatByScIdDefaultQueHCRxPackets	23
pfGsxScenarioStatByScIdDefaultQueHCTxOctets	24
pfGsxScenarioStatByScIdDefaultQueHCTxPackets	25
pfGsxScenarioStatByScIdDefaultQueHCDiscardOctets	26
pfGsxScenarioStatByScIdDefaultQueHCDiscardPackets	27

ScenarioId: シナリオのシナリオ ID です。長さは 1 です。シナリオ登録時に指定したシナリオ ID です。
シナリオ登録時にシナリオ ID の指定を省略した場合，シナリオ ID は自動割り当てされます。この場合，showscenarioname コマンドで割り当てられたシナリオ ID を確認してください。
ポートシナリオのシナリオ ID はポート 1 では 4097，ポート 2 では 4098，ポート 3 では 4099，ポート 4 では 4100 が割り当てられます。

pfGsxScenarioInfoByScIdTable の場合
pfGsxScenarioInfoByScIdEntry の OID は次のようになります。
1.3.6.1.4.1.1151.2.1.7.12.1.1.EntryOID.ScenarioId

固定値

EntryOID:テーブル内エントリの番号です。番号は連続していませんので注意してください。長さは 1 です。

pfGsxScenarioInfoByScIdScenarioId	1
pfGsxScenarioInfoByScIdScenarioName	2
pfGsxScenarioInfoByScIdScenarioType	3
pfGsxScenarioInfoByScIdDefFlowNum	4
pfGsxScenarioInfoByScIdTotalFlowNum	13
pfGsxScenarioInfoByScIdMaxBuffScenarioId	14
pfGsxScenarioInfoByScIdMaxBuffRatio	15
pfGsxScenarioInfoByScIdMaxBuff	16
pfGsxScenarioInfoByScIdMinBuffScenarioId	17
pfGsxScenarioInfoByScIdMinBuffRatio	18
pfGsxScenarioInfoByScIdMinBuff	19
pfGsxScenarioInfoByScIdAveBuffRatio	20
pfGsxScenarioInfoByScIdAveBuff	21
pfGsxScenarioInfoByScIdPeakBuffScenarioId	22
pfGsxScenarioInfoByScIdPeakBuffRatio	23
pfGsxScenarioInfoByScIdPeakBuff	24
pfGsxScenarioInfoByScIdDefBuffRatio	25
pfGsxScenarioInfoByScIdDefBuff	26
pfGsxScenarioInfoByScIdDefPeakBuffRatio	27
pfGsxScenarioInfoByScIdDefPeakBuff	28
pfGsxScenarioInfoByScIdTxPeakRateBps	29
pfGsxScenarioInfoByScIdTxAveRateBps	31
pfGsxScenarioInfoByScIdIndQueNum	33
pfGsxScenarioInfoByScIdAccelSessNum	34
pfGsxScenarioInfoByScIdAccelBypassStatus	35
pfGsxScenarioInfoByScIdAccelActivePeer	36

ScenarioId: シナリオのシナリオ ID です。長さは 1 です。求め方は pfGsxScenarioStatByScIdTable のシナリオ ID と同様です。

（空白ページ）

付録E　　JSON の記述方法

JSON（JavaScript Object Notation:RFC4627）による記述方法を示します。

JSON は RFC4627 に規定されるテキストベースの簡易なデータ記述言語です。
JSON には 4 つの型と 2 つの構造体があります。本装置の WebAPI では string 型と object 構造体のみを使用します。

表 E-1　JSON の型と構造体

	種　別	記述例	説　明
型	string	"PureFlow"	文字列
	number	123	数値
	boolean	true	真（true）または偽（false）を示します。
	null	null	値なしを示します。
構造体	object	{name:value}	0 個以上の名前と値のペアを並べたものです。
	array	[value, value]	0 個以上の値を並べたものです。

以下，「付録 F　WebAPI 詳細」の下記シナリオ追加 API を例にして記述方法を示します。

表 E-2　JSON の記述方法

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加（Discard）	"command"（必須）	"add scenario"	add scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"action"（必須）	"discard"	action discard
	"scenario_id"（省略可）	シナリオ ID	[scenario <scenario_id>]

キーと値を":"コロンでペアにします。

```
"command":"add scenario"
"scenario_name":"/port1/North"
"action":"discard"
"scenario_id":"1"
```

シナリオ ID の指定が不要な場合は省略できます。

```
"command":"add scenario"
"scenario_name":"/port1/North"
"action":"discard"
```

これら 3 つのパラメータを", "カンマでつなげます。最後のパラメータにはカンマを加えません。

```
"command":"add scenario", "scenario_name":"/port1/North", "action":"discard"
```

最後に波括弧"{"と"}"で囲って object 構造体にします。

```
{"command":"add scenario", "scenario_name":"/port1/North", "action":"discard"}
```

記述を見やすくするために，波括弧，コロン，カンマの前後には半角スペース，Tab，改行を加えることができます。

```
{
    "command" : "add scenario",
    "scenario_name" : "/port1/North",
    "action" : "discard"
}
```

なお，本装置の WebAPI ではパラメータの順序は順不同です。「付録 F　WebAPI 詳細」の順序に合わせる必要はありません。

```
{
    "action" : "discard",
    "scenario_name" : "/port1/North",
    "command" : "add scenario"
}
```

付録F　　WebAPI 詳細

本装置の WebAPI 詳細を示します。

WebAPI では以下の URL に対して JSON データを与えます。
http://システムインタフェースの IP アドレス/shapermng/json

HTTPS（Hypertext Transfer Secure）を利用する場合は，URL の先頭を"https"にしてください。
https://システムインタフェースの IP アドレス/shapermng/json

キーと値はすべて文字列で指定します。省略可能なパラメータは指定が不要な場合は記述不要です。キーにスペルミスがある場合，そのパラメータは無視されます。指定必須パラメータのスペルミスはエラーとなりますが，省略可能なパラメータのスペルミスや，未定義のパラメータはエラーとならないことに注意してください。

指定する値の詳細は「PureFlow WS1 トラフィックシェーパー NF7500 シリーズ コマンドリファレンス」を参照してください。

表 F-1　JSON キー一覧

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加（Discard）	"command"（必須）	"add scenario"	add scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"action"（必須）	"discard"	action discard
	"scenario_id"（省略可）	シナリオ ID	[scenario <scenario_id>]
シナリオ追加（Aggregate）	"command"（必須）	"add scenario"	add scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"action"（必須）	"aggregate"	action aggregate
	"cos"（省略可）	Cos 値	[cos <user_priority>]
	"inner-cos"（省略可）	Inner-Cos 値	[inner-cos <user_priority>]
	"dscp"（省略可）	dscp	[dscp <dscp>]
	"min_bandwidth"（省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth"（省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"class "（省略可）	クラス	[class <class>]

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加（Aggregate）（続き）	"bufsize"（省略可）	バッファサイズ	[bufsize <bufsize>]
	"scenario_id"（省略可）	シナリオ ID	[scenario <scenario_id>]
シナリオ追加（Individual）	"command"（必須）	"add scenario"	add scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"action"（必須）	"individual"	action individual
	"cos"（省略可）	Cos 値	[cos <user_priority>]
	"inner-cos"（省略可）	Inner-Cos 値	[inner-cos <user_priority>]
	"dscp"（省略可）	dscp	[dscp <dscp>]
	"min_bandwidth"（省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth"（省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"class "（省略可）	クラス	[class <class>]
	"bufsize"（省略可）	バッファサイズ	[bufsize <bufsize>]
	"scenario_id"（省略可）	シナリオ ID	[scenario <scenario_id>]
	"maxqnum"（省略可）	個別キュー最大数	[maxquenum <quenum>]
	"quedivision"（省略可）	個別キュー分割対象	[quedivision <field>]
	"failaction"（省略可）	個別キュー数超過時の動作	[failaction <discard | forwardbesteffort | forwardattribute>]
	"fail_min_bw"（省略可）	個別キュー数超過時の最低帯域	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw"（省略可）	個別キュー数超過時の最大帯域	[fail_peak_bw <peak_bandwidth>]
	"fail_class"（省略可）	個別キュー数超過時のクラス	[fail_class <class>]

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加 (Wan-accel)	"command" (必須)	"add scenario"	add scenario
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"action" (必須)	"wan-accel"	action wan-accel
	"peer" (必須)	IP アドレス	peer <IP_address>
	"second_peer" (省略可)	IP アドレス	[second-peer < IP_address >]
	"dport" (省略可)	宛先ポート番号	[dport <port>]
	"vid" (省略可)	VLAN ID	[vid <VID>]
	"inner vid" (省略可)	Inner-VLAN ID	[inner-vid <VID>]
	"cos" (省略可)	Cos 値	[cos <user_priority>]
	"inner-cos" (省略可)	Inner-Cos 値	[inner-cos <user_priority>]
	"dscp" (省略可)	dscp	[dscp <dscp>]
	"compression" (省略可)	圧縮 "enable": 圧縮を有効にします。 "disable": 圧縮を無効にします。 省略時は"disable"を適用します。	[compression {enable | disable}]
	"tcp_mem" (省略可)	TCP バッファサイズ	[tcp-mem {auto | <size>}]
	"cc_mode" (省略可)	輻輳制御モード "normal": 輻輳制御モードを通常モードにします。 "semi-fast": 輻輳制御モードを高速モードにします。 "fast": 輻輳制御モードを高速モードにします。 省略時は"normal"を適用します。	[cc-mode {normal | semi-fast | fast}]

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加（Wan-accel）(続き)	"bypass_thresh" （省略可）	RTT	[bypass-thresh <rtt>]
	"bypass_keepalive" （省略可）	自動バイパスの keepalive "enable": keepalive を有効にします。 "disable": keepalive を無効にします。 省略時は"disable"を適用します。	[bypass-keepalive {enable | disable}]
	"fec" （省略可）	FEC "enable": FEC 機能を有効にします。 "disable": FEC 機能を無効にします。 省略時は"disable"を適用します。	[fec {enable | disable}]
	"block_size" （省略可）	FEC ブロックサイズ	[block-size <size>]
	"data_block_size" （省略可）	FEC データブロックサイズ	[data-block-size <size>]
	"fec_session" （省略可）	FEC セッション数	[fec-session <session>]
	"min_bandwidth" （省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth" （省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"bufsize" （省略可）	バッファサイズ	[bufsize <bufsize>]
シナリオ更新（Aggregate）	"command" （必須）	"update scenario"	update scenario
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"action" （必須）	"aggregate"	action aggregate
	"min_bandwidth" （省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth" （省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"class " （省略可）	クラス	[class <class>]
	"bufsize" （省略可）	バッファサイズ	[bufsize <bufsize>]

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ更新(Individual)	"command" (必須)	"update scenario"	update scenario
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"action" (必須)	"individual"	action individual
	"min_bandwidth" (省略可)	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth" (省略可)	最大帯域	[peak_bw <peak_bandwidth>]
	"class " (省略可)	クラス	[class <class>]
	"bufsize" (省略可)	バッファサイズ	[bufsize <bufsize>]
	"maxqnum" (省略可)	個別キュー最大数	[maxquenum <quenum>]
	"quedivision" (省略可)	個別キュー分割対象	[quedivision <field>]
	"failaction" (省略可)	個別キュー数超過時の動作	[failaction <discard | forwardbesteffort | forwardattribute>]
	"fail_min_bw" (省略可)	個別キュー数超過時の最低帯域	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw" (省略可)	個別キュー数超過時の最大帯域	[fail_peak_bw <peak_bandwidth>]
	"fail_class" (省略可)	個別キュー数超過時のクラス	[fail_class <class>]
シナリオ更新(Wan-accel)	"command" (必須)	"update scenario"	update scenario
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"action" (必須)	"wan-accel"	action wan-accel
	"compression" (省略可)	圧縮 "enable": 圧縮を有効にします。 "disable": 圧縮を無効にします。 省略時は"disable"を適用します。	[compression {enable | disable}]
	"tcp_mem" (省略可)	TCP バッファサイズ	[tcp-mem {auto | <size>}]

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ更新 (Wan-accel) (続き)	"cc_mode" (省略可)	輻輳制御モード "normal": 輻輳制御モードを通常モードにします。 "semi-fast": 輻輳制御モードを高速モードにします。 "fast": 輻輳制御モードを高速モードにします。 省略時は"normal"を適用します。	[cc-mode {normal | semi-fast | fast}]
	"bypass_thresh" (省略可)	RTT	[bypass-thresh <rtt>]
	"bypass_keepalive" (省略可)	自動バイパスの keepalive "enable": keepalive を有効にします。 "disable": keepalive を無効にします。 省略時は"disable"を適用します。	[bypass-keepalive {enable | disable}]
	"fec" (省略可)	FEC "enable": FEC 機能を有効にします。 "disable": FEC 機能を無効にします。 省略時は"disable"を適用します。	[fec {enable | disable}]
	"block_size" (省略可)	FEC ブロックサイズ	[block-size <size>]
	"data_block_size" (省略可)	FEC データブロックサイズ	[data-block-size <size>]
	"fec_session" (省略可)	FEC セッション数	[fec-session <session>]
	"min_bandwidth" (省略可)	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth" (省略可)	最大帯域	[peak_bw <peak_bandwidth>]
	"bufsize" (省略可)	バッファサイズ	[bufsize <bufsize>]
シナリオ削除 (全指定)	"command" (必須)	"delete scenario"	delete scenario
	"scenario_name" (必須)	"all"	all

API	キー	値	相当する CLI コマンドとパラメータ
シナリオ削除（シナリオ指定）	"command" （必須）	"delete scenario"	delete scenario
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"recursive" （省略可）	"recursive"	[recursive]
シナリオ情報取得	"command" （必須）	"show scenario"	show scenario
	"scenario_name" （必須）	シナリオ名	name <scenario_name>
	"search_type" （省略可）	取得方法 "exact"：指定したシナリオの情報を取得します。 "next"：指定したシナリオの次のシナリオ情報を取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

シナリオ情報取得 API について

シナリオ情報取得 API では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"scenario_name"で指定したシナリオの情報を取得します。
"next"　"scenario_name"で指定したシナリオの次のシナリオの情報を取得します。
取得する順序は"show scenario"CLI コマンドと同様にシナリオツリー順です。

"search_type"を省略した場合は，"exact"を適用します。
特定のシナリオ情報を取得したい場合は，そのシナリオ名を指定して"exact"で取得してください。
CLI コマンドの"show scenario all"のようにすべてのシナリオ情報を取得したい場合は，"next"を使用して下記の手順で取得してください。

最初のシナリオ情報取得は"scenario_name"に空文字を指定します。
　"scenario_name"："（空文字）
　"search_type"："next"

シナリオツリーの先頭のシナリオ"/port1"の情報を取得できます。
続いて，"scenario_name"に取得したシナリオ名を指定します。
　"scenario_name"："/port1"
　"search_type"："next"

シナリオツリーで"/port1"の次に位置するシナリオの情報を取得できます。
このように，取得したシナリオ名を指定して"next"による取得を繰り返します。シナリオツリーの最後尾を指定して"next"による取得を行うと"Next scenario is not exist."のエラーになります。

管理PC
本装置
最初のシナリオ情報はシナリオ名に空文字を指定して取得します
"command" : "show scenario"
"scenario_name" : ""　（空文字）
"search_type" : "next"
"scenario_name" : "/port1"
"/port1"のシナリオ情報
取得したシナリオ名を指定して次のシナリオ情報を取得します
"command" : "show scenario"
"scenario_name" : "/port1"
"search_type" : "next"
"scenario_name" : "/port1/North"
"/port1/North"のシナリオ情報
取得したシナリオ名の指定を繰り返します
"command" : "show scenario"
"scenario_name" : "/port1/North"
"search_type" : "next"
"scenario_name" : "/port1/North/Area1"
"/port1/North/Area1"のシナリオ情報
"Next scenario is not exist."のエラーになるまで繰り返します
"error" : "Next scenario is not exist."

API	キー	値	相当する CLI コマンドとパラメータ
フィルタモード設定	"command" (必須)	"set filter mode"	set filter mode
	"slot/port" (必須)	スロット番号／ポート番号	<slot/port>
	"field" (必須)	フィールド	<field>
フィルタ追加(Bridge-ctrl)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"bridge-ctrl"	bridge-ctrl
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]
フィルタ追加(Ethernet)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"ethernet"	Ethernet
	"vid" (省略可)	VLAN ID	[vid {<VID> | none}]
	"cos" (省略可)	CoS	[cos <user_priority>]
	"inner-vid" (省略可)	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"inner-cos" (省略可)	Inner-CoS	[inner-cos <user_priority>]
	"ethertype" (省略可)	Ethernet Type/Length	[ethertype <type>]
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]
フィルタ追加(IPv4)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"ipv4"	ipv4

API	キー	値	相当する CLI コマンドとパラメータ
フィルタ追加 (IPv4) (続き)	"vid" (省略可)	VLAN ID	[vid {<VID> | none}]
	"cos" (省略可)	CoS	[cos <user_priority>]
	"inner-vid" (省略可)	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"inner-cos" (省略可)	Inner-CoS	[inner-cos <user_priority>]
	"sip" または "sip list" (省略可)	送信元 IPv4 アドレス または ルールリスト名 "sip"と"sip list"を同時に使用した場合"sip list"が優先されます。	[sip [list] {<src_IP_address> | <list_name>}]
	"dip" または "dip list" (省略可)	宛先 IPv4 アドレス または ルールリスト名 "dip"と"dip list"を同時に使用した場合"dip list"が優先されます。	[dip [list] {<dst_IP_address> | <list_name>}]
	"tos" (省略可)	ToS	[tos <type_of_service>]
	"proto" (省略可)	プロトコル番号	[proto <protocol>]
	"sport" または "sport list" (省略可)	送信元ポート番号 または ルールリスト名 "sport"と"sport list"を同時に使用した場合"sport list"が優先されます。	[sport [list] {<sport> | <list_name>}]
	"dport" または "dport list" (省略可)	宛先ポート番号 または ルールリスト名 "dport"と"dport list"を同時に使用した場合"dport list"が優先されます。	[dport [list] {<dport> | <list_name>}]
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]
フィルタ追加 (IPv6)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"ipv6"	ipv6

API	キー	値	相当する CLI コマンドとパラメータ
フィルタ追加(IPv6)(続き)	"vid" (省略可)	VLAN ID	[vid {<VID> | none}]
	"cos" (省略可)	CoS	[cos <user_priority>]
	"inner-vid" (省略可)	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"inner-cos" (省略可)	Inner-CoS	[inner-cos <user_priority>]
	"sip" または "sip list" (省略可)	送信元IPv6アドレス または ルールリスト名 "sip"と"sip list"を同時に使用した場合"sip list"が優先されます。	[sip [list] {<src_IP_address> | <list_name>}]
	"dip" または "dip list" (省略可)	宛先 IPv6 アドレス または ルールリスト名 "dip"と"dip list"を同時に使用した場合"dip list"が優先されます。	[dip [list] {<dst_IP_address> | <list_name>}]
	"tos" (省略可)	ToS	[tos <type_of_service>]
	"proto" (省略可)	プロトコル番号	[proto <protocol>]
	"sport" または "sport list" (省略可)	送信元ポート番号 または ルールリスト名 "sport"と"sport list"を同時に使用した場合"sport list"が優先されます。	[sport [list] {<sport> | <list_name>}]
	"dport" または "dport list" (省略可)	宛先ポート番号 または ルールリスト名 "dport"と" dport list"を同時に使用した場合" dport list"が優先されます。	[dport [list] {<dport> | <list_name>}]
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]
フィルタ削除(全指定)	"command" (必須)	"delete filter"	delete filter
	"scenario_name" (必須)	"all"	All
フィルタ削除(シナリオ指定)	"command" (必須)	"delete filter"	delete filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>

API	キー	値	相当する CLI コマンドとパラメータ
フィルタ削除（フィルタ指定）	"command"（必須）	"delete filter"	delete filter
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"filter_name"（必須）	フィルタ名	filter <scenario_name>
フィルタ情報取得	"command"（必須）	"show filter"	show filter
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"filter_name"（必須）	フィルタ名	filter <scenario_name>
	"search_type"（省略可）	取得方法 "exact"：指定したフィルタの情報を取得します。 "next"：指定したフィルタの次のフィルタ情報を取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

フィルタ情報取得 API について

フィルタ情報取得 API では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"scenario_name"および"filter_name"で指定したフィルタの情報を取得します。

"next"　"scenario_name"および"filter_name"で指定したフィルタの次のフィルタの情報を取得します。取得する順序は"show filter" CLI コマンドと同様にフィルタ名のアルファベット順です。シナリオの最後尾のフィルタを指定した場合は，次のシナリオの先頭のフィルタ情報を取得します。

特定のフィルタ情報を取得したい場合は，そのシナリオ名およびフィルタ名を指定して"exact"で取得してください。

CLI コマンドの"show filter all"のようにすべてのシナリオのすべてのフィルタ情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得 API と同様です。

API	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化追加	"command" （必須）	"add apl-accel"	add apl-accel
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"protocol " （必須）	プロトコル名	protocol smb
	" tcp_port" （省略可）	TCP ポート番号	[tcp <port>]
	"smb-session" （省略可）	セッション数	[smb-session <session>]
	"read-attr" （省略可）	read 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-attr {enable | disable}]
	"read-operation" （省略可）	read 操作の SMB2 READ コマンド代理応答 "enable": SMB2 READ コマンドの代理応答を有効にします。 "disable": SMB2 READ コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-operation {enable | disable}]
	"read-cache-size" （省略可）	read 操作の代理応答キャッシュサイズ	[read-cache-size <size>]

API	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化追加 (続き)	"write-attr" (省略可)	write 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの属性代理応答を無効にします。 省略時は"enable"を適用します。	[write-attr {enable | disable}]
	"write-attr-1st" (省略可)	write 操作前の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-1st {enable | disable}]
	"write-attr-2nd" (省略可)	write 操作後の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-2nd {enable | disable}]

API	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化追加 (続き)	"write-operation" (省略可)	write 操作の SMB2 WRITE コマンドの代理応答 "enable": SMB2 WRITE コマンドの代理応答を有効にします。 "disable": SMB2 WRITE コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[write-operation {enable | disable}]
アプリケーション高速化更新	"command" (必須)	"update apl-accel"	update apl-accel
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"protocol " (必須)	プロトコル名	protocol smb
	"tcp_port" (省略可)	TCP ポート番号	[tcp <port>]
	"smb-session" (省略可)	セッション数	[smb-session <session>]
	"read-attr" (省略可)	read 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-attr {enable | disable}]

API	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化更新 (続き)	"read-operation" (省略可)	read 操作の SMB2 READ コマンド代理応答 "enable": SMB2 READ コマンドの代理応答を有効にします。 "disable": SMB2 READ コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-operation {enable | disable}]
	"read-cache-size" (省略可)	read 操作の代理応答キャッシュサイズ	[read-cache-size <size>]
	"write-attr" (省略可)	write 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの属性代理応答を無効にします。 省略時は"enable"を適用します。	[write-attr {enable | disable}]
	"write-attr-1st" (省略可)	write 操作前の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-1st {enable | disable}]

API	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化更新 (続き)	"write-attr-2nd" (省略可)	write 操作後の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-2nd {enable | disable}]
	"write-operation" (省略可)	write 操作の SMB2 WRITE コマンドの代理応答 "enable": SMB2 WRITE コマンドの代理応答を有効にします。 "disable": SMB2 WRITE コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[write-operation {enable | disable}]
アプリケーション高速化削除	"command" (必須)	"delete apl-accel"	delete apl-accel
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"protocol " (必須)	プロトコル名	protocol smb

API	キー	値	相当する CLI コマンドとパラメータ
ルールリストグループ追加	"command" (必須)	"add rulelist group"	add rulelist group
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	ルールリスト種別	{ipv4 | ipv6 | l4port}
ルールリストグループ削除(全指定)	"command" (必須)	"delete rulelist group"	delete rulelist group
	"list_name" (必須)	"all"	all
ルールリストグループ削除(グループ指定)	"command" (必須)	"delete rulelist group"	delete rulelist group
	"list_name" (必須)	ルールリスト名	<list_name>
ルールリストエントリ追加(IPv4)	"command" (必須)	"add rulelist entry"	add rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv4"	ipv4
	"IP_address" (必須)	IPv4 アドレス	<IP_address>
ルールリストエントリ追加(IPv6)	"command" (必須)	"add rulelist entry"	add rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv6"	ipv6
	"IP_address" (必須)	IPv6 アドレス	<IP_address>
ルールリストエントリ追加(L4Port)	"command" (必須)	"add rulelist entry"	add rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"l4port"	l4port
	"port" (必須)	L4 ポート番号	<port>
ルールリストエントリ削除(全指定)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"all"	all

API	キー	値	相当する CLI コマンドとパラメータ
ルールリストエントリ削除(IPv4)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv4"	ipv4
	"IP_address" (必須)	IPv4 アドレス	<IP_address>
ルールリストエントリ削除(IPv6)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv6"	ipv6
	"IP_address" (必須)	IPv6 アドレス	<IP_address>
ルールリストエントリ削除(IPv6)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv6"	ipv6
	"port" (必須)	L4 ポート番号	<port>
ルールリスト情報取得	"command" (必須)	"show rulelist"	show rulelist
	"list_name" (必須)	ルールリスト名	[<list_name>]
	"rules" (必須)	ルールリストエントリ	なし
	"search_type" (省略可)	取得方法 "exact":指定したルールリストエントリを取得します。 "next": 指定したルールリストエントリの次のルールリストエントリを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

ルールリスト情報取得 API について

ルールリスト情報取得 API では"show rulelist" CLI コマンドにはない"rules"パラメータがあります。

"rules"には値としてルールリストエントリ（IP アドレスまたは L4 ポート番号）を指定します。ルールリストエントリは単一の値であっても常にハイフンを使用した範囲指定で指定してください。

IPv4 アドレス	192.168.1.1-192.168.1.1
IPv6 アドレス	FE80::0001-FE80::0001
L4 ポート番号	1000-1000

なお，ルールリストエントリが設定されていないルールリストでは"none"が取得されます。

取得方法を指定する"search_type"には値として"exact"か"next"を指定します。

"exact"	"list_name"および"rules"で指定したルールリストエントリを取得します。
"next"	"list_name"および"rules"で指定したルールリストエントリの次のルールリストエントリを取得します。取得する順序は"show rulelist" CLI コマンドと同様です。ルールリストの最後尾のルールリストエントリを指定した場合は，次のルールリストの先頭のルールリストエントリを取得します。

特定のルールリストエントリを取得したい場合は，そのルールリスト名およびルールリストエントリを指定して"exact"で取得してください。
CLI コマンドの"show rulelist all"のようにすべてのルールリストのすべてのルールリストエントリを取得したい場合は"next"を使用します。"next"での取得手順はシナリオ取得 API と同様です。

API	キー	値	相当する CLI コマンドとパラメータ
チャネル追加（通常チャネル）	"command" （必須）	"add channel"	add channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"lan" （必須）	Lan 側ポート番号 または ポートグループ	lan {<slot/port>| <group_name>}
	"wan" （必須）	Wan 側ポート番号 または ポートグループ	wan {<slot/port>| <group_name>}
	"channel_type" （必須）	チャネル種別 "normal": 通常チャネルと追加します。 "default": デフォルトチャネルを追加します。	なし
	"vid" （必須）	VLAN ID	vid {<VID> | none}
	"inner_vid" （省略可）	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"tpid" （省略可）	tpid	[tpid <tpid>]
	"inner_tpid" （省略可）	inner-tpid	[inner-tpid <tpid>]
	"mtu" （省略可）	mtu	[mtu <mtu>]
チャネル追加（デフォルトチャネル）	"command" （必須）	"add channel"	add channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"lan" （必須）	Lan 側ポート番号 または ポートグループ	lan {<slot/port>| <group_name>}
	"wan" （必須）	Wan 側ポート番号 または ポートグループ	wan {<slot/port>| <group_name>}
	"channel_type" （必須）	チャネル種別 "normal": 通常チャネルと追加します。 "default": デフォルトチャネルを追加します。	なし

API	キー	値	相当する CLI コマンドとパラメータ
チャネル削除 (全指定)	"command" (必須)	"delete channel"	delete channel
	"channel_name" (必須)	"all"	all
チャネル削除 (チャネル名指定)	"command" (必須)	"delete channel"	delete channel
	"channel_name" (必須)	チャネル名	<channel_name>
チャネル 情報表示	"command" (必須)	"show channel"	show channel
	"channel_name" (必須)	チャネル名	name <channel_name>
	"search_type" (省略可)	取得方法 "exact":指定したチャネルを取得します。 "next": 指定したチャネルの次のチャネルを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

チャネル情報取得 API について

チャネル情報取得 API では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"channel_name"で指定したチャネルの情報を取得します。
"next"　" channel_name"で指定したチャネルの次のチャネルの情報を取得します。取得する順序は"show channel" CLI コマンドと同様にチャネル名のアルファベット順です。

特定のチャネル情報を取得したい場合は，そのチャネル名を指定して"exact"で取得してください。
CLI コマンドの"show channel all"のようにすべてのチャネル情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得 API と同様です。

API	キー	値	相当する CLI コマンドとパラメータ
インタフェース設定	"command" （必須）	"set ip channel"	set ip channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"IP_address" （必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask" （必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
インタフェース解除 （全指定）	"command" （必須）	"unset ip channel"	unset ip channel
	"channel_name" （必須）	"all"	all
インタフェース解除 （チャネル名指定）	"command" （必須）	"unset ip channel"	unset ip channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"type" （省略可）	設定解除対象 "ipv4": IPv4 チャネルインタフェースの設定を解除します。 "ipv6": IPv6 チャネルインタフェースの設定を解除します。 省略時は IPv4 と IPv6 のチャネルインタフェース両方の設定を解除します。	[{ipv4 | ipv6}]
インタフェース情報表示	"command" （必須）	"show ip channel"	show ip channel
	"channel_name" （必須）	チャネル名	name <channel_name>
	"search_type" （省略可）	取得方法 "exact":指定したチャネルを取得します。 "next": 指定したチャネルの次のチャネルを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

インタフェース情報取得 API について

インタフェース情報取得 API では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"channel_name"で指定したインタフェースの情報を取得します。
"next"　" channel_name"で指定したインタフェースの次のインタフェースの情報を取得します。取得する順序は"show ip interface" CLI コマンドと同様にチャネル名のアルファベット順です。

特定のインタフェース情報を取得したい場合は，そのインタフェース名を指定して"exact"で取得してください。
CLI コマンドの"show ip interface all"のようにすべてのインタフェース情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得 API と同様です。

API	キー	値	相当する CLI コマンドとパラメータ
スタティック経路追加（宛先指定）	"command"（必須）	"add route"	add route
	"route_type"（必須）	target	target
	"IP_address"（必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask"（必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
スタティック経路追加（デフォルトゲートウェイ）	"command"（必須）	"add route"	add route
	"route_type"（必須）	default	default
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
スタティック経路削除（全指定）	"command"（必須）	"delete route"	delete route
	"route_type"（必須）	all	all
スタティック経路削除（宛先指定）	"command"（必須）	"delete route"	delete route

API	キー	値	相当する CLI コマンドとパラメータ
スタティック経路削除（宛先指定）(続き)	"route_type"（必須）	target	target
	"IP_address"（必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask"（必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
スタティック経路削除（デフォルトゲートウェイ）	"command"（必須）	"delete route"	delete route
	"route_type"（必須）	default	default
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
	"type"（省略可）	設定解除対象 "ipv4": IPv4 スタティック経路情報の設定を削除します。 "ipv6": IPv6 スタティック経路情報の設定を削除します。 省略時は IPv4 と IPv6 のスタティック経路情報両方の設定を削除します。	[{ipv4 | ipv6}]

API	キー	値	相当する CLI コマンドとパラメータ
スタティック経路情報表示（宛先指定）	"command"（必須）	"show route target"	show route target
	"IP_address"（必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask"（必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
	"search_type"（省略可）	取得方法 "exact": 指定したチャネルを取得します。 "next": 指定したチャネルの次のチャネルを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

経路情報取得 API について

経路情報取得 API では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　入力パラメータすべてに一致した経路の情報を取得します。
"next"　入力パラメータすべてに一致した経路の次の経路の情報を取得します。取得する順序は"show route" CLI コマンドと同様にチャネル名のエントリ順です。

特定の経路情報を取得したい場合は，その経路情報の情報全てを指定して"exact"で取得してください。CLI コマンドの"show route all"のようにすべての経路情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得 API と同様です。

API	キー	値	相当する CLI コマンドとパラメータ
OpenFlow コントローラー追加	"command"（必須）	"add openflow controller"	add openflow controller
	"IP_address"（必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"tcp"（任意）	TCP ポート番号	[tcp <port>]
OpenFlow コントローラー削除	"command"（必須）	"delete openflow controller"	delete openflow controller
	"IP_address"（必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
OpenFlow コントローラー情報取得	"command"（必須）	"show openflow controller"	show openflow controller

API	キー	値	相当する CLI コマンドとパラメータ
コンフィギュレーション保存	"command"（必須）	"save config"	save config
コンフィギュレーション保存実行状態取得	"command"（必須）	"show save status"	show save status

コンフィギュレーション保存 API について

コンフィギュレーション保存 API は保存の完了を待たずに終了します。コンフィギュレーション保存はバックグラウンドで実行されます。保存が実行されている最中にさらに本 API でコンフィギュレーションの保存を指示した場合，"configuration save is in progress"のエラーメッセージを返します。コンフィギュレーション保存の所要時間については「第 3 章　設定の基本」を参照してください。

コンフィギュレーション保存実行状態取得 API について

本 API はコンフィギュレーション保存の実行状態を取得します。

"configuration save is in progress"　　:コンフィギュレーション保存が実行中です。
"configuration save is not in progress"　:コンフィギュレーション保存は完了しています。

付録G　WebAPI サンプルプログラム

WebAPI で広く使用されているプログラミング言語に Python があります。Python には標準で HTTP および JSON のライブラリが含まれており，本装置の WebAPI 利用にも適しています。

付録 F の各 WebAPI について Python バージョン 2.7.2 によるサンプルプログラムを示します。

設定系

設定の追加で add 系を，設定の更新で update 系を，設定の削除で delete 系の API を使用します。
add，update，set，および delete 系の API では，コマンドとパラメータを送信してレスポンスを確認する手順になります。いずれの API においても同様ですので，"add scenario"の例を示します。

① 単一の設定を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
params = {
          'command': 'add scenario',
          'scenario_name' : '/port1/North',
          'action' : 'aggregate',
          'min_bandwidth' : '100M',
          'peak_bandwidth' : '1G',
          'bufsize' : '1M'
          }
json_data = json.dumps(params)

# POST リクエスト
response = urllib2.urlopen(url, json_data)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL     :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '----------'
print data
print
```

Python の urlopen は，HTTP リクエスト完了時に終了し，バックグラウンドで本装置とのセッション終了処理を行います。このため，複数の urlopen を使用するとき，次の urlopen 使用時に，本装置側では前回のセッションが終了していない場合があります。この操作を繰り返すと，本装置側のセッションリソースが枯渇し，WebAPI が一時的に利用できなくなります。

複数の API を続けて実行する場合，HTTP の持続的接続を利用し，HTTP 接続を維持したまま複数の API を発行するようにプログラミングを行ってください。以下に持続的接続を利用したサンプルプログラムを示します。

② 接続を維持したまま複数の設定を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
params = {
          'command': 'add scenario',
          'scenario_name' : '/port1/North',
          'action' : 'aggregate',
          'min_bandwidth' : '100M',
          'peak_bandwidth' : '1G',
          'bufsize' : '1M'
          }
json_data = json.dumps(params)

# POST リクエスト
conn.request("POST", '/'+file, json_data)
response = conn.getresponse()

# レスポンスの表示
print 'RESPONSE:', response

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print

# コネクションを開放
conn.close()
```

コンフィギュレーション保存

一連の設定変更が完了したら，コンフィギュレーション保存 API によってコンフィギュレーションの変更を保存してください。

コンフィギュレーション保存 API ではコマンドを送信してレスポンスを確認する手順になります。

コンフィギュレーション保存 API は保存の完了を待たずにレスポンスを返し，バックグラウンドでコンフィギュレーション保存を実行します。保存が実行されている最中にさらに本 API でコンフィギュレーションの保存を指示した場合，"configuration save is in progress"のエラーメッセージを返しますので，レスポンス内容にこのエラーメッセージが表示された場合は時間を空けてからもう一度実行してください。コンフィギュレーション保存の所要時間については「第 3 章　設定の基本」を参照してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
params = {
          'command': 'save config'
          }
json_data = json.dumps(params)

# POST リクエスト
response = urllib2.urlopen(url, json_data)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL     :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

コンフィギュレーション保存実行状態取得

コンフィギュレーション保存が実行中かどうかを本 API によって取得できます。
本 API はレスポンスに下記のメッセージを返します。

"configuration save is in progress"　　:コンフィギュレーション保存が実行中です。
"configuration save is not in progress"　:コンフィギュレーション保存は完了しています。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
params = {
          'command': 'show save status'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL     :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

表示系

設定内容を確認したい場合は show 系の API を使用します。
show 系 API では，コマンドとパラメータを送信してレスポンスの確認とデータを表示する手順になります。1 エントリのみの取得と，全エントリの取得ではプログラミング方法が異なります。各 API についてそれぞれのサンプルコードを示します。

(1) チャネル情報取得(チャネル指定)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# search_type は exact を指定する
params = {
          'command': 'show channel',
          'channel_name': 'dc_tokyo',
          'search_type': 'exact'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

(2) チャネル情報取得(全取得)

① 接続を維持したまま全チャネル情報の取得を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
# チャネル全表示のときは，最初のチャネル名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show channel',
          'channel_name': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url    = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print
        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにチャネル名が存在しない場合は終了
        if json_data.has_key("channel_name")==False:
                break

        # チャネル名を取得する
        channel_name = json_data['channel_name']
```

（続き）

```
        # チャネル名を取得したものに更新して続行
        params['channel_name'] = channel_name

# コネクションを開放
conn.close()
```

② 1チャネルの取得ごとに接続と切断を行うサンプルプログラム。

下記サンプルプログラムを使用した場合，実行端末の処理性能によっては，本装置側のセッションリソースが枯渇し，urlopen がエラー終了する場合があります。エラーが発生する場合は，①の接続を維持したまま全シナリオの取得を行うサンプルプログラムを使用してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# チャネル全表示のときは，最初のチャネル名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show channel',
          'channel_name': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url     = urllib.urlencode(params)

        # GET リクエスト
        response = urllib2.urlopen(url+'?'+params_url)

        # レスポンスの表示
        print 'RESPONSE:', response
        print 'URL      :', response.geturl()

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにチャネル名が存在しない場合は終了
        if json_data.has_key("channel_name")==False:
                break

        # チャネル名を取得する
        channel_name = json_data['channel_name']
```

（続き）

```
        # チャネル名を取得したものに更新して続行
        params['channel_name'] = channel_name
```

(3) IP インタフェース情報取得(チャネル指定)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# search_type は exact を指定する
params = {
          'command': 'show ip interface',
          'channel_name': 'dc_tokyo',
          'search_type': 'exact'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

(4) IP インタフェース情報取得(全取得)

① 接続を維持したまま全チャネルの IP インタフェース情報の取得を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
# IP インタフェース全表示のときは，最初のチャネル名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show ip interface',
          'channel_name': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url    = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print
        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにチャネル名が存在しない場合は終了
        if json_data.has_key("channel_name")==False:
                break
```

（続き）

```
        # チャネル名を取得する
        channel_name = json_data['channel_name']

        # チャネル名を取得したものに更新して続行
        params['channel_name'] = channel_name

# コネクションを開放
conn.close()
```

② 1チャネルのIPインタフェース情報の取得ごとに接続と切断を行うサンプルプログラム。

下記サンプルプログラムを使用した場合，実行端末の処理性能によっては，本装置側のセッションリソースが枯渇し，urlopen がエラー終了する場合があります。エラーが発生する場合は，①の接続を維持したまま全シナリオの取得を行うサンプルプログラムを使用してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# チャネル全表示のときは，最初のチャネル名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show ip interface',
          'channel_name': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url     = urllib.urlencode(params)

        # GET リクエスト
        response = urllib2.urlopen(url+'?'+params_url)

        # レスポンスの表示
        print 'RESPONSE:', response
        print 'URL      :', response.geturl()

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにチャネル名が存在しない場合は終了
        if json_data.has_key("channel_name")==False:
                break

        # チャネル名を取得する
        channel_name = json_data['channel_name']
```

（続き）

```
        # チャネル名を取得したものに更新して続行
        params['channel_name'] = channel_name
```

(5) スタティック経路情報取得(宛先指定)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# search_type は exact を指定する
params = {
          'command': 'show route target',
          'IP_address': '192.168.100.0',
          'netmask': '255.255.255.0',
          'gateway': '192.168.100.1',
          'channel_name': 'ch1',
          'output_if': 'lan',
          'search_type': 'exact'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

(6) スタティック経路情報取得(全取得)

① 接続を維持したままスタティック全経路情報の取得を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
# スタティック経路情報全表示のときは, 全パラメータを０文字指定する
# search_type は next を指定する
params = {
          'command': 'show route target',
          'IP_address': '',
          'netmask': '',
          'gateway': '',
          'channel_name': '',
          'output_if': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url    = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print
```

（続き）

```
# レスポンスのデータ部（JSON 形式の文字列）から
# Python dictionary データを取得する
json_data1 = json.loads(data)
json_data2 = json.loads(data)
json_data3 = json.loads(data)
json_data4 = json.loads(data)
json_data5 = json.loads(data)

# JSON キーに target IP が存在しない場合は終了
if json_data1.has_key("target")==False:
    break
# JSON キーに netmask が存在しない場合は終了
if json_data2.has_key("netmask")==False:
    break
# JSON キーに gateway が存在しない場合は終了
if json_data3.has_key("gateway")==False:
    break
# JSON キーに channel_name が存在しない場合は終了
if json_data4.has_key("channel_name")==False:
    break
# JSON キーに output_if が存在しない場合は終了
if json_data5.has_key("output_if")==False:
    break
# target IP を取得する
IP_address = json_data1['target']
# netmask を取得する
netmask = json_data2['netmask']
# gateway を取得する
gateway = json_data3['gateway']
# channel_name を取得する
channel_name = json_data4['channel_name']
# output_if を取得する
output_if = json_data5['output_if']

# target IP を取得したものに更新して続行
params['IP_address'] = IP_address
# netmask を取得したものに更新して続行
params['netmask'] = netmask
# gateway を取得したものに更新して続行
params['gateway'] = gateway
# channel_name を取得したものに更新して続行
params['channel_name'] = channel_name
# output_if を取得したものに更新して続行
params['output_if'] = output_if

# コネクションを開放
conn.close()
```

（続き）

② 1スタティック経路情報の取得ごとに接続と切断を行うサンプルプログラム。

下記サンプルプログラムを使用した場合，実行端末の処理性能によっては，本装置側のセッションリソースが枯渇し，urlopen がエラー終了する場合があります。エラーが発生する場合は，①の接続を維持したまま全シナリオの取得を行うサンプルプログラムを使用してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# スタティック経路情報全表示のときは，全パラメータを０文字指定する
# search_type は next を指定する
params = {
          'command': 'show route target',
          'IP_address': '',
          'netmask': '',
          'gateway': '',
          'channel_name': '',
          'output_if': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url     = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print
        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data1 = json.loads(data)
        json_data2 = json.loads(data)
        json_data3 = json.loads(data)
        json_data4 = json.loads(data)
        json_data5 = json.loads(data)
```

（続き）

```
# JSON キーに target IP が存在しない場合は終了
if json_data1.has_key("target")==False:
    break
# JSON キーに netmask が存在しない場合は終了
if json_data2.has_key("netmask")==False:
    break
# JSON キーに gateway が存在しない場合は終了
if json_data3.has_key("gateway")==False:
    break
# JSON キーに channel_name が存在しない場合は終了
if json_data4.has_key("channel_name")==False:
    break
# JSON キーに output_if が存在しない場合は終了
if json_data5.has_key("output_if")==False:
    break
# target IP を取得する
IP_address = json_data1['target']
# netmask を取得する
netmask = json_data2['netmask']
# gateway を取得する
gateway = json_data3['gateway']
# channel_name を取得する
channel_name = json_data4['channel_name']
# output_if を取得する
output_if = json_data5['output_if']

# target IP を取得したものに更新して続行
params['IP_address'] = IP_address
# netmask を取得したものに更新して続行
params['netmask'] = netmask
# gateway を取得したものに更新して続行
params['gateway'] = gateway
# channel_name を取得したものに更新して続行
params['channel_name'] = channel_name

# output_if を取得したものに更新して続行
params['output_if'] = output_if
```

(7) シナリオ情報取得(シナリオ指定)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# search_type は exact を指定する
params = {
          'command': 'show scenario',
          'scenario_name': '/port1/North',
          'search_type': 'exact'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL     :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

(8) シナリオ情報取得(全取得)

① 接続を維持したまま全シナリオの取得を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
# シナリオ全表示のときは，最初のシナリオ名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show scenario',
          'scenario_name': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url    = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print
        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにシナリオ名が存在しない場合は終了
        if json_data.has_key("scenario_name")==False:
                break

        # シナリオ名を取得したものに更新して続行
        params['scenario_name'] = scenario_name
```

（続き）

```
        # シナリオ名を取得する
        scenario_name = json_data['scenario_name']

# コネクションを開放
conn.close()
```

② 1シナリオの取得ごとに接続と切断を行うサンプルプログラム。

下記サンプルプログラムを使用した場合，実行端末の処理性能によっては，本装置側のセッションリソースが枯渇し，urlopen がエラー終了する場合があります。エラーが発生する場合は，①の接続を維持したまま全シナリオの取得を行うサンプルプログラムを使用してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# シナリオ全表示のときは，最初のシナリオ名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show scenario',
          'scenario_name': '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url     = urllib.urlencode(params)

        # GET リクエスト
        response = urllib2.urlopen(url+'?'+params_url)

        # レスポンスの表示
        print 'RESPONSE:', response
        print 'URL      :', response.geturl()

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにシナリオ名が存在しない場合は終了
        if json_data.has_key("scenario_name")==False:
                break

        # シナリオ名を取得する
        scenario_name = json_data['scenario_name']
```

（続き）

```
        # シナリオ名を取得したものに更新して続行
        params['scenario_name'] = scenario_name
```

(9) フィルタ情報取得(フィルタ指定)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# search_type は exact を指定する
params = {
          'command': 'show filter',
          'scenario_name' : '/port1/North',
          'filter_name' : 'filter1',
          'search_type' : 'exact'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

(10)　フィルタ情報取得(全取得)

① 接続を維持したまま全フィルタの取得を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
# フィルタ全表示のときは，最初のシナリオ名とフィルタ名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show filter',
          'scenario_name' : '',
          'filter_name' : '',
          'search_type' : 'next'
          }

while 1:
        # URL エンコードする
        params_url    = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)
```

（続き）

```
        # JSON キーにシナリオ名が存在しない場合は終了
        if json_data.has_key("scenario_name")==False:
                break

        # JSON キーにフィルタ名が存在しない場合は終了
        if json_data.has_key("filter_name")==False:
                break

        # シナリオ名を取得する
        scenario_name = json_data['scenario_name']

        # フィルタ名を取得する
        filter_name = json_data['filter_name']

        # シナリオ名とフィルタ名を取得したものに更新して続行
        params['scenario_name'] = scenario_name
        params['filter_name'] = filter_name

# コネクションを開放
conn.close()
```

② 1フィルタの取得ごとに接続と切断を行うサンプルプログラム。

下記サンプルプログラムを使用した場合，実行端末の処理性能によっては，本装置側のセッションリソースが枯渇し，urlopen がエラー終了する場合があります。エラーが発生する場合は，①の接続を維持したまま全シナリオの取得を行うサンプルプログラムを使用してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# フィルタ全表示のときは，最初のシナリオ名とフィルタ名を０文字指定する
# search_type は next を指定する
params = {
          'command': 'show filter',
          'scenario_name' : '',
          'filter_name' : '',
          'search_type' : 'next'
          }

while 1:
        # URL エンコードする
        params_url     = urllib.urlencode(params)

        # GET リクエスト
        response = urllib2.urlopen(url+'?'+params_url)

        # レスポンスの表示
        print 'RESPONSE:', response
        print 'URL      :', response.geturl()

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにシナリオ名が存在しない場合は終了
        if json_data.has_key("scenario_name")==False:
                break
```

（続き）

```
# JSON キーにフィルタ名が存在しない場合は終了
if json_data.has_key("filter_name")==False:
                break

# シナリオ名を取得する
scenario_name = json_data['scenario_name']


# フィルタ名を取得する
filter_name = json_data['filter_name']

# シナリオ名とフィルタ名を取得したものに更新して続行
params['scenario_name'] = scenario_name
params['filter_name'] = filter_name
```

(11)　ルールリスト情報取得(ルールリストエントリ指定)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# search_type は exact を指定する
params = {
          'command': 'show rulelist',
          'list_name' : 'v4servers',
          'rules' : '192.168.10.1-192.168.10.1',
          'search_type': 'exact'
          }

# URL エンコードする
params_url     = urllib.urlencode(params)

# GET リクエスト
response = urllib2.urlopen(url+'?'+params_url)

# レスポンスの表示
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '---------'
print data
print
```

(12)　ルールリスト情報取得(全取得)

① 接続を維持したまま全ルールリストの取得を行うサンプルプログラム。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# URL 定義 WebAPI の IP アドレスとファイル名
ip    = '192.168.1.1'
file  = 'shapermng/json'

# コネクションを生成 HTTP
conn = httplib.HTTPConnection(ip)
# コネクションを生成 HTTPS
#conn = httplib.HTTPSConnection(ip)

# パラメータ定義
# ルールリスト全表示のときは，ルールリスト名とルールリストエントリを
# 0文字指定する
# search_type は next を指定する
params = {
          'command': 'show rulelist',
          'list_name' : '',
          'rules' : '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url    = urllib.urlencode(params)

        # GET リクエスト
        conn.request("GET", '/'+file+'?'+params_url)
        response = conn.getresponse()

        # レスポンスの表示
        print 'RESPONSE:', response

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)
```

（続き）

```
        # JSON キーにルールリスト名が存在しない場合は終了
        if json_data.has_key("list_name")==False:
                break

        # JSON キーにルールリストエントリが存在しない場合は終了
        if json_data.has_key("rules")==False:
                break

        # ルールリスト名を取得する
        list_name = json_data['list_name']

        # ルールリストエントリを取得する
        rules = json_data['rules']

        # ルールリスト名とルールリストエントリを取得したものに更新して続行
        # none の場合は次のルールリストエントリがないことを示し、
        # 次のルールリストを取得するために rules を 0 文字指定する
        params['list_name'] = list_name
        if rules == 'none':
                params['rules'] = ''
        else:
                params['rules'] = rules

# コネクションを開放
conn.close()
```

② 1ルールリストの取得ごとに接続と切断を行うサンプルプログラム。

下記サンプルプログラムを使用した場合，実行端末の処理性能によっては，本装置側のセッションリソースが枯渇し，urlopen がエラー終了する場合があります。エラーが発生する場合は，①の接続を維持したまま全シナリオの取得を行うサンプルプログラムを使用してください。

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# URL 定義 WebAPI の URL HTTP
url     = 'http://192.168.1.1/shapermng/json'
# URL 定義 WebAPI の URL HTTPS
#url     = 'https://192.168.1.1/shapermng/json'

# パラメータ定義
# ルールリスト全表示のときは，ルールリスト名とルールリストエントリを
# ０文字指定する
# search_type は next を指定する
params = {
          'command': 'show rulelist',
          'list_name' : '',
          'rules' : '',
          'search_type': 'next'
          }

while 1:
        # URL エンコードする
        params_url     = urllib.urlencode(params)

        # GET リクエスト
        response = urllib2.urlopen(url+'?'+params_url)

        # レスポンスの表示
        print 'RESPONSE:', response
        print 'URL      :', response.geturl()

        data = response.read()
        print 'LENGTH  :', len(data)
        print 'DATA    :'
        print '---------'
        print data
        print

        # レスポンスのデータ部（JSON 形式の文字列）から
        # Python dictionary データを取得する
        json_data = json.loads(data)

        # JSON キーにルールリスト名が存在しない場合は終了
        if json_data.has_key("list_name")==False:
                break
```

（続き）

```
        # JSON キーにルールリストエントリが存在しない場合は終了
        if json_data.has_key("rules")==False:
                break

        # ルールリスト名を取得する
        list_name = json_data['list_name']

        # ルールリストエントリを取得する
        rules = json_data['rules']

        # ルールリスト名とルールリストエントリを取得したものに更新して続行
        # none の場合は次のルールリストエントリがないことを示し、
        # 次のルールリストを取得するために rules を 0 文字指定する
        params['list_name'] = list_name
        if rules == 'none':
                params['rules'] = ''
        else:
                params['rules'] = rules
```

付録H　CLI コマンド対応 OpenFlow メッセージ詳細

本装置の OpenFlow 詳細を示します。

OpenFlow ではメッセージタイプが OFPT_EXPERIMENTER(4)の場合のデータ部と，メッセージタイプが OFPT_MULTIPART_REQUEST(18)で multipart type が OFPMP_EXPERIMENTER(0xffff)のパケットのデータ部に対して JSON データを与えます。
また，メッセージタイプが OFPT_FLOW_MOD(14)の場合では match フィールド，instruction フィールド，action フィールドを使用してフィルタコマンドの設定が行えます。

キーと値はすべて文字列で指定します。省略可能なキーは指定が不要な場合は記述不要です。キーにスペルミスがある場合，そのキーと値は無視されます。指定必須キーのスペルミスはエラーとなりますが，省略可能なキーのスペルミスや，未定義のキーはエラーとならないことに注意してください。

指定する値の詳細は「PureFlow WS1 トラフィックシェーパーNF7500 シリーズ コマンドリファレンス」を参照してください。

次にメッセージタイプが OFPT_EXPERIMENTER(4)の場合のデータ部と，メッセージタイプが OFPT_MULTIPART_REQUEST(18)で multipart type が OFPMP_EXPERIMENTER(0xffff)のパケットのデータ部に対して JSON データを示します。

このとき，EXPERIMENTER ID には 0x00000091 としてください。

また，exp_type フィールドには CLI コマンドの種類に対応した値を設定してください

表 H-1　exp_type フィールド一覧

相当する CLI コマンド	exp_type
add scenario	1
update scenario	2
delete scenario	3
show scenario	4
show scenario counter	5
add apl-accel	6
update apl-accel	7
delete apl-accel	8
add filter	9
delete filter	10
show filter	11
add rulelist group	12
delete rulelist group	13
add rulelist entry	14
delete rulelist entry	15
show rulelist	16
add channel	17

相当する CLI コマンド	exp_type
delete channel	18
show channel	19
set ip channel	20
unset ip channel	21
show ip channel	22
add route	23
delete route	24
show route target	25
set wan-accel bypass status	26
set wan-accel bypass recoverytime	27
switch wan-accel bypass force	28
set filter mode	29

次にデータ部に対する JSON データを示します。

表 H-2　JSON キー一覧

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加(Discard)	"command" (必須)	"add scenario"	add scenario
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"action" (必須)	"discard"	action discard
	"scenario_id" (省略可)	シナリオ ID	[scenario <scenario_id>]
シナリオ追加(Aggregate)	"command" (必須)	"add scenario"	add scenario
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"action" (必須)	"aggregate"	action aggregate
	"cos" (省略可)	Cos 値	[cos <user_priority>]
	"inner-cos" (省略可)	Inner-Cos 値	[inner-cos <user_priority>]
	"dscp" (省略可)	dscp	[dscp <dscp>]
	"min_bandwidth" (省略可)	最低帯域	[min_bw <min_bandwidth>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加(Aggregate)(続き)	"peak_bandwidth"(省略可)	最大帯域	[peak_bw <peak_bandwidth>]
	"class "(省略可)	クラス	[class <class>]
	"bufsize"(省略可)	バッファサイズ	[bufsize <bufsize>]
	"scenario_id"(省略可)	シナリオ ID	[scenario <scenario_id>]
シナリオ追加(Individual)	"command"(必須)	"add scenario"	add scenario
	"scenario_name"(必須)	シナリオ名	scenario <scenario_name>
	"action"(必須)	"individual"	action individual
	"cos"(省略可)	Cos 値	[cos <user_priority>]
	"inner-cos"(省略可)	Inner-Cos 値	[inner-cos <user_priority>]
	"dscp"(省略可)	dscp	[dscp <dscp>]
	"min_bandwidth"(省略可)	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth"(省略可)	最大帯域	[peak_bw <peak_bandwidth>]
	"class "(省略可)	クラス	[class <class>]
	"bufsize"(省略可)	バッファサイズ	[bufsize <bufsize>]
	"scenario_id"(省略可)	シナリオ ID	[scenario <scenario_id>]
	"maxqnum"(省略可)	個別キュー最大数	[maxquenum <quenum>]
	"quedivision"(省略可)	個別キュー分割対象	[quedivision <field>]
	"failaction"(省略可)	個別キュー数超過時の動作	[failaction <discard | forwardbesteffort | forwardattribute>]
	"fail_min_bw"(省略可)	個別キュー数超過時の最低帯域	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw"(省略可)	個別キュー数超過時の最大帯域	[fail_peak_bw <peak_bandwidth>]
	"fail_class"(省略可)	個別キュー数超過時のクラス	[fail_class <class>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加（Wan-accel）	"command" （必須）	"add scenario"	add scenario
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"action" （必須）	"wan-accel"	action wan-accel
	"peer" （必須）	IP アドレス	peer <IP_address>
	"second_peer" （省略可）	IP アドレス	[second-peer < IP_address >]
	"dport" （省略可）	宛先ポート番号	[dport <port>]
	"vid" （省略可）	VLAN ID	[vid <VID>]
	"inner-vid" （省略可）	Inner-VLAN ID	[inner-vid <VID>]
	"cos" （省略可）	Cos 値	[cos <user_priority>]
	"inner-cos" （省略可）	Inner-Cos 値	[inner-cos <user_priority>]
	"dscp" （省略可）	dscp	[dscp <dscp>]
	"compression" （省略可）	圧縮 "enable": 圧縮を有効にします。 "disable": 圧縮を無効にします。 省略時は"disable"を適用します。	[compression {enable | disable}]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ追加（Wan-accel）(続き)	"cc_mode" （省略可）	輻輳制御モード "normal": 輻輳制御モードを通常モードにします。 "semi-fast": 輻輳制御モードを高速モードにします。 "fast": 輻輳制御モードを高速モードにします。 省略時は"normal"を適用します。	[cc-mode {normal | semi-fast | fast}]
	"bypass_thresh" （省略可）	RTT	[bypass-thresh <rtt>]
	"bypass_keepalive" （省略可）	自動バイパスの keepalive "enable": keepalive を有効にします。 "disable": keepalive を無効にします。 省略時は"disable"を適用します。	[bypass-keepalive {enable | disable}]
	"fec" （省略可）	FEC "enable": FEC 機能を有効にします。 "disable": FEC 機能を無効にします。 省略時は"disable"を適用します。	[fec {enable | disable}]
	"block_size" （省略可）	FEC ブロックサイズ	[block-size <size>]
	"data_block_size" （省略可）	FEC データブロックサイズ	[data-block-size <size>]
	"fec_session" （省略可）	FEC セッション数	[fec-session <session>]
	"min_bandwidth" （省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth" （省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"bufsize" （省略可）	バッファサイズ	[bufsize <bufsize>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ更新（Aggregate）	"command"（必須）	"update scenario"	update scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"action"（必須）	"aggregate"	action aggregate
	"min_bandwidth"（省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth"（省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"class "（省略可）	クラス	[class <class>]
	"bufsize"（省略可）	バッファサイズ	[bufsize <bufsize>]
シナリオ更新（Individual）	"command"（必須）	"update scenario"	update scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"action"（必須）	"individual"	action individual
	"min_bandwidth"（省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth"（省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"class "（省略可）	クラス	[class <class>]
	"bufsize"（省略可）	バッファサイズ	[bufsize <bufsize>]
	"maxqnum"（省略可）	個別キュー最大数	[maxquenum <quenum>]
	"quedivision"（省略可）	個別キュー分割対象	[quedivision <field>]
	"failaction"（省略可）	個別キュー数超過時の動作	[failaction <discard | forwardbesteffort | forwardattribute>]
	"fail_min_bw"（省略可）	個別キュー数超過時の最低帯域	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw"（省略可）	個別キュー数超過時の最大帯域	[fail_peak_bw <peak_bandwidth>]
	"fail_class"（省略可）	個別キュー数超過時のクラス	[fail_class <class>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ更新（Wan-accel）	"command" （必須）	"update scenario"	update scenario
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"action" （必須）	"wan-accel"	action wan-accel
	"compression" （省略可）	圧縮 "enable": 圧縮を有効にします。 "disable": 圧縮を無効にします。 省略時は"disable"を適用します。	[compression {enable | disable}]
	"tcp_mem" （省略可）	TCP バッファサイズ	[tcp-mem {auto | <size>}]
	"cc_mode" （省略可）	輻輳制御モード "normal": 輻輳制御モードを通常モードにします。 "semi-fast": 輻輳制御モードを高速モードにします。 "fast": 輻輳制御モードを高速モードにします。 省略時は"normal"を適用します。	[cc-mode {normal | semi-fast | fast}]
	"bypass_thresh" （省略可）	RTT	[bypass-thresh <rtt>]
	"bypass_keepalive" （省略可）	自動バイパスの keepalive "enable": keepalive を有効にします。 "disable": keepalive を無効にします。 省略時は"disable"を適用します。	[bypass-keepalive {enable |disable}]
	"fec" （省略可）	FEC "enable": FEC 機能を有効にします。 "disable": FEC 機能を無効にします。 省略時は"disable"を適用します。	[fec {enable |disable}]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオ更新（Wan-accel）(続き)	"block_size"（省略可）	FEC ブロックサイズ	[block-size <size>]
	"data_block_size"（省略可）	FEC データブロックサイズ	[data-block-size <size>]
	"fec_session"（省略可）	FEC セッション数	[fec-session <session>]
	"min_bandwidth"（省略可）	最低帯域	[min_bw <min_bandwidth>]
	"peak_bandwidth"（省略可）	最大帯域	[peak_bw <peak_bandwidth>]
	"bufsize"（省略可）	バッファサイズ	[bufsize <bufsize>]
シナリオ削除（全指定）	"command"（必須）	"delete scenario"	delete scenario
	"scenario_name"（必須）	"all"	all
シナリオ削除（シナリオ指定）	"command"（必須）	"delete scenario"	delete scenario
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"recursive"（省略可）	"recursive"	[recursive]
シナリオ情報取得	"command"（必須）	"show scenario"	show scenario
	"scenario_name"（必須）	シナリオ名	name <scenario_name>
	"search_type"（省略可）	取得方法 "exact"：指定したシナリオの情報を取得します。 "next"：指定したシナリオの次のシナリオ情報を取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
シナリオカウンタ情報取得	"command" （必須）	"show scenario counter"	show scenario counter
	"scenario_name" （必須）	シナリオ名	name <scenario_name>
	"search_type" （省略可）	取得方法 "exact"：指定したシナリオの情報を取得します。 "next"：指定したシナリオの次のシナリオ情報を取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし
	"default_queue" （省略可）	"default_queue"	[default_queue]

シナリオ情報取得について

シナリオ情報取得では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"scenario_name"で指定したシナリオの情報を取得します。
"next"　"scenario_name"で指定したシナリオの次のシナリオの情報を取得します。
取得する順序は"show scenario"CLI コマンドと同様にシナリオツリー順です。

"search_type"を省略した場合は，"exact"を適用します。
特定のシナリオ情報を取得したい場合は，そのシナリオ名を指定して"exact"で取得してください。
CLI コマンドの"show scenario all"のようにすべてのシナリオ情報を取得したい場合は，"next"を使用して下記の手順で取得してください。

最初のシナリオ情報取得は"scenario_name"に空文字を指定します。
　"scenario_name"："（空文字）
　"search_type"："next"

シナリオツリーの先頭のシナリオ"/port1"の情報を取得できます。
続いて，"scenario_name"に取得したシナリオ名を指定します。
　"scenario_name"："/port1"
　"search_type"："next"

シナリオツリーで"/port1"の次に位置するシナリオの情報を取得できます。
このように，取得したシナリオ名を指定して"next"による取得を繰り返します。シナリオツリーの最後尾を指定して"next"による取得を行うと"Next scenario is not exist."のエラーになります。

管理PC
本装置
最初のシナリオ情報はシナリオ名に空文字を指定して取得します
"command" : "show scenario"
"scenario_name" : ""　（空文字）
"search_type" : "next"
"scenario_name" : "/port1"
"/port1"のシナリオ情報
取得したシナリオ名を指定して次のシナリオ情報を取得します
"command" : "show scenario"
"scenario_name" : "/port1"
"search_type" : "next"
"scenario_name" : "/port1/North"
"/port1/North"のシナリオ情報
取得したシナリオ名の指定を繰り返します
"command" : "show scenario"
"scenario_name" : "/port1/North"
"search_type" : "next"
"scenario_name" : "/port1/North/Area1"
"/port1/North/Area1"のシナリオ情報
"Next scenario is not exist."のエラーになるまで繰り返します
"error" : "Next scenario is not exist."

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
フィルタモード設定	"command" (必須)	"set filter mode"	set filter mode
	"slot/port" (必須)	スロット番号／ポート番号	<slot/port>
	"field" (必須)	フィールド	<field>
フィルタ追加(Bridge-ctrl)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"bridge-ctrl"	bridge-ctrl
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]
フィルタ追加(Ethernet)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"ethernet"	ethernet
	"vid" (省略可)	VLAN ID	[vid {<VID> | none}]
	"cos" (省略可)	CoS	[cos <user_priority>]
	"inner-vid" (省略可)	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"inner-cos" (省略可)	Inner-CoS	[inner-cos <user_priority>]
	"ethertype" (省略可)	Ethernet Type/Length	[ethertype <type>]
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
フィルタ追加(IPv4)	"command" (必須)	"add filter"	add filter
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"filter_name" (必須)	フィルタ名	filter <scenario_name>
	"type" (必須)	"ipv4"	ipv4
	"vid" (省略可)	VLAN ID	[vid {<VID> | none}]
	"cos" (省略可)	CoS	[cos <user_priority>]
	"inner-vid" (省略可)	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"inner-cos" (省略可)	Inner-CoS	[inner-cos <user_priority>]
	"sip" または "sip list" (省略可)	送信元 IPv4 アドレス または ルールリスト名 "sip"と"sip list"を同時に使用した場合"sip list"が優先されます。	[sip [list] {<src_IP_address> | <list_name>}]
	"dip" または "dip list" (省略可)	宛先 IPv4 アドレス または ルールリスト名 "dip"と"dip list"を同時に使用した場合"dip list"が優先されます。	[dip [list] {<dst_IP_address> | <list_name>}]
	"tos" (省略可)	ToS	[tos <type_of_service>]
	"proto" (省略可)	プロトコル番号	[proto <protocol>]
	"sport" または "sport list" (省略可)	送信元ポート番号 または ルールリスト名 "sport"と"sport list"を同時に使用した場合"sport list"が優先されます。	[sport [list] {<sport> | <list_name>}]
	"dport" または "dport list" (省略可)	宛先ポート番号 または ルールリスト名 "dport"と"dport list"を同時に使用した場合"dport list"が優先されます。	[dport [list] {<dport> | <list_name>}]
	"priority" (省略可)	フィルタ優先度	[priority <filter_pri>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
フィルタ追加（IPv6）	"command" （必須）	"add filter"	add filter
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"filter_name" （必須）	フィルタ名	filter <scenario_name>
	"type" （必須）	"ipv6"	ipv6
	"vid" （省略可）	VLAN ID	[vid {<VID> | none}]
	"cos" （省略可）	CoS	[cos <user_priority>]
	"inner-vid" （省略可）	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"inner-cos" （省略可）	Inner-CoS	[inner-cos <user_priority>]
	"sip" または "sip list" （省略可）	送信元 IPv6 アドレス または ルールリスト名 "sip"と"sip list"を同時に使用した場合"sip list"が優先されます。	[sip [list] {<src_IP_address> | <list_name>}]
	"dip" または "dip list" （省略可）	宛先 IPv6 アドレス または ルールリスト名 "dip"と"dip list"を同時に使用した場合"dip list"が優先されます。	[dip [list] {<dst_IP_address> | <list_name>}]
	"tos" （省略可）	ToS	[tos <type_of_service>]
	"proto" （省略可）	プロトコル番号	[proto <protocol>]
	"sport" または "sport list" （省略可）	送信元ポート番号 または ルールリスト名 "sport"と"sport list"を同時に使用した場合"sport list"が優先されます。	[sport [list] {<sport> | <list_name>}]
	"dport" または "dport list" （省略可）	宛先ポート番号 または ルールリスト名 "dport"と" dport list"を同時に使用した場合" dport list"が優先されます。	[dport [list] {<dport> | <list_name>}]
	"priority" （省略可）	フィルタ優先度	[priority <filter_pri>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
フィルタ削除（全指定）	"command"（必須）	"delete filter"	delete filter
	"scenario_name"（必須）	"all"	all
フィルタ削除（シナリオ指定）	"command"（必須）	"delete filter"	delete filter
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
フィルタ削除（フィルタ指定）	"command"（必須）	"delete filter"	delete filter
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"filter_name"（必須）	フィルタ名	filter <scenario_name>
フィルタ情報取得	"command"（必須）	"show filter"	show filter
	"scenario_name"（必須）	シナリオ名	scenario <scenario_name>
	"filter_name"（必須）	フィルタ名	filter <scenario_name>
	"search_type"（省略可）	取得方法 "exact"：指定したフィルタの情報を取得します。 "next"：指定したフィルタの次のフィルタ情報を取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

フィルタ情報取得について

フィルタ情報取得では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"scenario_name"および"filter_name"で指定したフィルタの情報を取得します。

"next"　"scenario_name"および"filter_name"で指定したフィルタの次のフィルタの情報を取得します。取得する順序は"show filter" CLI コマンドと同様にフィルタ名のアルファベット順です。シナリオの最後尾のフィルタを指定した場合は，次のシナリオの先頭のフィルタ情報を取得します。

特定のフィルタ情報を取得したい場合は，そのシナリオ名およびフィルタ名を指定して"exact"で取得してください。

CLI コマンドの"show filter all"のようにすべてのシナリオのすべてのフィルタ情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得と同様です。

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化追加	"command" （必須）	"add apl-accel"	add apl-accel
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"protocol " （必須）	プロトコル名	protocol smb
	"tcp_port" （省略可）	TCP ポート番号	[tcp <port>]
	"smb-session" （省略可）	セッション数	[smb-session <session>]
	"read-attr" （省略可）	read 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-attr {enable | disable}]
	"read-operation" （省略可）	read 操作の SMB2 READ コマンド代理応答 "enable": SMB2 READ コマンドの代理応答を有効にします。 "disable": SMB2 READ コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-operation {enable | disable}]
	"read-cache-size" （省略可）	read 操作の代理応答キャッシュサイズ	[read-cache-size <size>]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化追加 (続き)	"write-attr" (省略可)	write 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの属性代理応答を無効にします。 省略時は"enable"を適用します。	[write-attr {enable | disable}]
	"write-attr-1st" (省略可)	write 操作前の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-1st {enable | disable}]
	"write-attr-2nd" (省略可)	write 操作後の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-2nd {enable | disable}]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化追加 (続き)	"write-operation" (省略可)	write 操作の SMB2 WRITE コマンドの代理応答 "enable": SMB2 WRITE コマンドの代理応答を有効にします。 "disable": SMB2 WRITE コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[write-operation {enable | disable}]
アプリケーション高速化更新	"command" (必須)	"update apl-accel"	update apl-accel
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"protocol " (必須)	プロトコル名	protocol smb
	"tcp_port" (省略可)	TCP ポート番号	[tcp <port>]
	"smb-session" (省略可)	セッション数	[smb-session <session>]
	"read-attr" (省略可)	read 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-attr {enable | disable}]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化更新(続き)	"read-operation" (省略可)	read 操作の SMB2 READ コマンド代理応答 "enable": SMB2 READ コマンドの代理応答を有効にします。 "disable": SMB2 READ コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[read-operation {enable | disable}]
	"read-cache-size" (省略可)	read 操作の代理応答キャッシュサイズ	[read-cache-size <size>]
	"write-attr" (省略可)	write 操作の SMB2 QUERY_INFO コマンドの代理応答 "enable": SMB2 QUERY_INFO コマンドの代理応答を有効にします。 "disable": SMB2 QUERY_INFO コマンドの属性代理応答を無効にします。 省略時は"enable"を適用します。	[write-attr {enable | disable}]
	"write-attr-1st" (省略可)	write 操作前の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-1st {enable | disable}]

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
アプリケーション高速化更新(続き)	"write-attr-2nd" (省略可)	write 操作後の SMB2 SET_INFO コマンドの代理応答 "enable": SMB2 SET_INFO コマンドの代理応答を有効にします。 "disable": SMB2 SET_INFO コマンドの代理応答を無効にします。 省略時は"disable"を適用します。	[write-attr-2nd {enable | disable}]
	"write-operation" (省略可)	write 操作の SMB2 WRITE コマンドの代理応答 "enable": SMB2 WRITE コマンドの代理応答を有効にします。 "disable": SMB2 WRITE コマンドの代理応答を無効にします。 省略時は"enable"を適用します。	[write-operation {enable | disable}]
アプリケーション高速化削除	"command" (必須)	"delete apl-accel"	delete apl-accel
	"scenario_name" (必須)	シナリオ名	scenario <scenario_name>
	"protocol " (必須)	プロトコル名	protocol smb

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
ルールリストグループ追加	"command" (必須)	"add rulelist group"	add rulelist group
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	ルールリスト種別	{ipv4 | ipv6 | l4port}
ルールリストグループ削除(全指定)	"command" (必須)	"delete rulelist group"	delete rulelist group
	"list_name" (必須)	"all"	all
ルールリストグループ削除(グループ指定)	"command" (必須)	"delete rulelist group"	delete rulelist group
	"list_name" (必須)	ルールリスト名	<list_name>
ルールリストエントリ追加(IPv4)	"command" (必須)	"add rulelist entry"	add rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv4"	ipv4
	"IP_address" (必須)	IPv4 アドレス	<IP_address>
ルールリストエントリ追加(IPv6)	"command" (必須)	"add rulelist entry"	add rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv6"	ipv6
	"IP_address" (必須)	IPv6 アドレス	<IP_address>
ルールリストエントリ追加(L4Port)	"command" (必須)	"add rulelist entry"	add rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"l4port"	l4port
	"port" (必須)	L4 ポート番号	<port>
ルールリストエントリ削除(全指定)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"all"	all

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
ルールリストエントリ削除(IPv4)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv4"	ipv4
	"IP_address" (必須)	IPv4 アドレス	<IP_address>
ルールリストエントリ削除(IPv6)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv6"	ipv6
	"IP_address" (必須)	IPv6 アドレス	<IP_address>
ルールリストエントリ削除(IPv6)	"command" (必須)	"delete rulelist entry"	delete rulelist entry
	"list_name" (必須)	ルールリスト名	<list_name>
	"type" (必須)	"ipv6"	ipv6
	"port" (必須)	L4 ポート番号	<port>
ルールリスト情報取得	"command" (必須)	"show rulelist"	show rulelist
	"list_name" (必須)	ルールリスト名	[<list_name>]
	"rules" (必須)	ルールリストエントリ	なし
	"search_type" (省略可)	取得方法 "exact":指定したルールリストエントリを取得します。 "next": 指定したルールリストエントリの次のルールリストエントリを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

ルールリスト情報取得について

ルールリスト情報取得では"show rulelist" CLI コマンドにはない"rules"パラメータがあります。

"rules"には値としてルールリストエントリ（IP アドレスまたは L4 ポート番号）を指定します。ルールリストエントリは単一の値であっても常にハイフンを使用した範囲指定で指定してください。

IPv4 アドレス	192.168.1.1-192.168.1.1
IPv6 アドレス	FE80::0001-FE80::0001
L4 ポート番号	1000-1000

なお，ルールリストエントリが設定されていないルールリストでは"none"が取得されます。

取得方法を指定する"search_type"には値として"exact"か"next"を指定します。

"exact"	"list_name"および"rules"で指定したルールリストエントリを取得します。
"next"	"list_name"および"rules"で指定したルールリストエントリの次のルールリストエントリを取得します。取得する順序は"show rulelist" CLI コマンドと同様です。ルールリストの最後尾のルールリストエントリを指定した場合は，次のルールリストの先頭のルールリストエントリを取得します。

特定のルールリストエントリを取得したい場合は，そのルールリスト名およびルールリストエントリを指定して"exact"で取得してください。
CLI コマンドの"show rulelist all"のようにすべてのルールリストのすべてのルールリストエントリを取得したい場合は"next"を使用します。"next"での取得手順はシナリオ取得と同様です。

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
チャネル追加（通常チャネル）	"command" （必須）	"add channel"	add channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"lan" （必須）	Lan 側ポート番号 または ポートグループ	lan {<slot/port>| <group_name>}
	"wan" （必須）	Wan 側ポート番号 または ポートグループ	wan {<slot/port>| <group_name>}
	"channel_type" （必須）	チャネル種別 "normal": 通常チャネルと追加します。 "default": デフォルトチャネルを追加します。	なし
	"vid" （必須）	VLAN ID	vid {<VID> | none}
	"inner_vid" （省略可）	Inner-VLAN ID	[inner-vid {<VID> | none}]
	"tpid" （省略可）	tpid	[tpid <tpid>]
	"inner_tpid" （省略可）	inner-tpid	[inner-tpid <tpid>]
	"mtu" （省略可）	mtu	[mtu <mtu>]
チャネル追加（デフォルトチャネル）	"command" （必須）	"add channel"	add channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"lan" （必須）	Lan 側ポート番号 または ポートグループ	lan {<slot/port>| <group_name>}
	"wan" （必須）	Wan 側ポート番号 または ポートグループ	wan {<slot/port>| <group_name>}
	"channel_type" （必須）	チャネル種別 "normal": 通常チャネルと追加します。 "default": デフォルトチャネルを追加します。	なし

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
チャネル削除 (全指定)	"command" (必須)	"delete channel"	delete channel
	"channel_name" (必須)	"all"	all
チャネル削除 (チャネル名指定)	"command" (必須)	"delete channel"	delete channel
	"channel_name" (必須)	チャネル名	<channel_name>
チャネル 情報表示	"command" (必須)	"show channel"	show channel
	"channel_name" (必須)	チャネル名	name <channel_name>
	"search_type" (省略可)	取得方法 "exact":指定したチャネルを取得します。 "next": 指定したチャネルの次のチャネルを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

チャネル情報取得について

チャネル情報取得では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"channel_name"で指定したチャネルの情報を取得します。

"next"　"channel_name"で指定したチャネルの次のチャネルの情報を取得します。取得する順序は"show channel" CLI コマンドと同様にチャネル名のアルファベット順です。

特定のチャネル情報を取得したい場合は，そのチャネル名を指定して"exact"で取得してください。
CLI コマンドの"show channel all"のようにすべてのチャネル情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得と同様です。

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
インタフェース設定	"command" （必須）	"set ip channel"	set ip channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"IP_address" （必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask" （必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
インタフェース解除 （全指定）	"command" （必須）	"unset ip channel"	unset ip channel
	"channel_name" （必須）	"all"	all
インタフェース解除 （チャネル名指定）	"command" （必須）	"unset ip channel"	unset ip channel
	"channel_name" （必須）	チャネル名	<channel_name>
	"type" （省略可）	設定解除対象 "ipv4": IPv4 チャネルインタフェースの設定を解除します。 "ipv6": IPv6 チャネルインタフェースの設定を解除します。 省略時は IPv4 と IPv6 のチャネルインタフェース両方の設定を解除します。	[{ipv4 | ipv6}]
インタフェース情報表示	"command" （必須）	"show ip channel"	show ip channel
	"channel_name" （必須）	チャネル名	name <channel_name>
	"search_type" （省略可）	取得方法 "exact":指定したチャネルを取得します。 "next": 指定したチャネルの次のチャネルを取得します。 省略時は"exact"を適用します。	なし

インタフェース情報取得について

インタフェース情報取得では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　"channel_name"で指定したインタフェースの情報を取得します。
"next"　"channel_name"で指定したインタフェースの次のインタフェースの情報を取得します。取得する順序は"show ip interface" CLI コマンドと同様にチャネル名のアルファベット順です。

特定のインタフェース情報を取得したい場合は，そのインタフェース名を指定して"exact"で取得してください。
CLI コマンドの"show ip interface all"のようにすべてのインタフェース情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得と同様です。

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
スタティック経路追加（宛先指定）	"command"（必須）	"add route"	add route
	"route_type"（必須）	target	target
	"IP_address"（必須）	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask"（必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
スタティック経路追加（デフォルトゲートウェイ）	"command"（必須）	"add route"	add route
	"route_type"（必須）	default	default
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
スタティック経路削除（全指定）	"command"（必須）	"delete route"	delete route
	"route_type"（必須）	all	all
スタティック経路削除（宛先指定）	"command"（必須）	"delete route"	delete route
	"route_type"（必須）	target	target

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
スタティック経路削除 (宛先指定) (続き)	"IP_address" (必須)	IPv4 アドレス または IPv6 アドレス	<IP_address>
	"netmask" (必須)	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
	"gateway" (必須)	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name" (必須)	チャネル名	channel <channel_name>
	"output_if" (必須)	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
スタティック経路削除 (デフォルトゲートウェイ)	"command" (必須)	"delete route"	delete route
	"route_type" (必須)	default	default
	"channel_name" (必須)	チャネル名	channel <channel_name>
	"output_if" (必須)	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
	"type" (省略可)	設定解除対象 "ipv4": IPv4 スタティック経路情報の設定を削除します。 "ipv6": IPv6 スタティック経路情報の設定を削除します。 省略時は IPv4 と IPv6 のスタティック経路情報両方の設定を削除します。	[{ipv4 | ipv6}]
スタティック経路情報表示 (宛先指定)	"command" (必須)	"show route target"	show route target
	"IP_address" (必須)	IPv4 アドレス または IPv6 アドレス	<IP_address>

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
スタティック経路情報表示（宛先指定）（続き）	"netmask"（必須）	IPv4 ネットマスク または IPv6 プレフィックス	netmask <netmask>
	"gateway"（必須）	IPv4 アドレス または IPv6 アドレス	gateway <gateway>
	"channel_name"（必須）	チャネル名	channel <channel_name>
	"output_if"（必須）	送信 Network ポート "lan": 送信 Network ポートが LAN 側 "wan": 送信 Network ポートが WAN 側	{lan | wan}
	"search_type"（省略可）	取得方法 "exact": 指定したチャネルを取得します。 "next": 指定したチャネルの次のチャネルを取得します。 省略時・値のスペルミス時は"exact"を適用します。	なし

経路情報取得について

経路情報取得では取得方法を指定する"search_type"パラメータがあります。"search_type"には値として"exact"か"next"を指定します。

"exact"　入力パラメータすべてに一致した経路の情報を取得します。

"next"　入力パラメータすべてに一致した経路の次の経路の情報を取得します。取得する順序は"show route" CLI コマンドと同様にチャネル名のエントリ順です。

特定の経路情報を取得したい場合は，その経路情報の情報すべてを指定して"exact"で取得してください。CLI コマンドの"show route all"のようにすべての経路情報を取得したい場合は，"next"を使用します。"next"での取得手順はシナリオ取得と同様です。

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
トラフィックアクセラレーションバイパス設定	"command" （必須）	"set wan-accel bypass status"	set wan-accel bypass status
	"status" （必須）	バイパス "enable": バイパスを有効にします。 "disable": バイパスを無効にします。	{enable | disable}
トラフィックアクセラレーションバイパスリカバリタイム設定	"command" （必須）	"set wan-accel bypass recoverytime"	set wan-accel bypass recoverytime
	"recoverytime" （必須）	バイパスリカバリタイム	<duration>
トラフィックアクセラレーション強制バイパス設定 （全指定）	"command" （必須）	"switch wan-accel bypass force"	switch wan-accel bypass force
	"status" （必須）	強制バイパス "enable": 強制バイパスを有効にします。 "disable": 強制バイパスを無効にします。	{enable | disable}
	"scenario_name" （必須）	"all"	all
トラフィックアクセラレーション強制バイパス設定 （シナリオ指定）	"command" （必須）	"switch wan-accel bypass force"	switch wan-accel bypass force
	"status" （必須）	強制バイパス "enable": 強制バイパスを有効にします。 "disable": 強制バイパスを無効にします。	{enable | disable}
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>

次に Modify Flow Entry（以降 FlowMod）詳細を示します。

OpenFlow の FlowMod(OFPT_FLOW_MOD)メッセージを送ることで，フィルタの登録および削除を行います。
フィルタの登録および削除を行うには，command フィールドに以下を指定してください。

表 H-3　Command フィールド一覧

CLI コマンドの種類	Command フィールド	相当する CLI コマンドとパラメータ
フィルタ追加	OFPFC_ADD	add filter
フィルタ削除	OFPFC_DELETE_STRICT	delete filter

FlowMod メッセージのパラメータは，Match フィールドおよび JSON で指定します。Match フィールドで指定できるパラメータを以下に示します。Match フィールドはすべて省略可能です。省略したパラメータはデフォルト値として扱われます。

表 H-4　Match フィールド一覧

CLI コマンドの種類	フィールド	相当する CLI コマンドとパラメータ	前提条件
フィルタ追加	OFPXMT_OFB_ETH_TYPE	[ethertype <type>]	なし
	OFPXMT_OFB_IP_PROTO	[proto <protocol>]	ethertype に“ 0x0800 ”(ipv4) または“ 0x86dd ”(ipv6)が指定されていること。
	OFPXMT_OFB_IPV4_SRC	[sip <src_IP_address>]	ethertype に“ 0x0800 ”(ipv4)が指定されていること。
	OFPXMT_OFB_IPV4_DST	[dip <dst_IP_address>]	ethertype に“ 0x0800 ”(ipv4)が指定されていること。
	OFPXMT_OFB_IPV6_SRC	[sip <src_IP_address>]	ethertype に“ 0x86dd ”(ipv6)が指定されていること。
	OFPXMT_OFB_IPV6_DST	[dip <dst_IP_address>]	ethertype に“ 0x86dd ”(ipv6)が指定されていること。
	OFPXMT_OFB_TCP_SRC	[sport <sport>]	proto に“6”(tcp)が指定されていること。
	OFPXMT_OFB_TCP_DST	[dport <dport>]	proto に“6”(tcp)が指定されていること。
	OFPXMT_OFB_UDP_SRC	[sport <sport>]	proto に“17”(udp)が指定されていること。
	OFPXMT_OFB_UDP_DST	[sport <sport>]	proto に“17”(udp)が指定されていること。
フィルタ削除	なし	なし	なし

フィルタ削除の場合，上記の Match フィールドは無視されます。同じ CLI コマンドパラメータに複数の値を指定することはできません。前提条件を満たさないリクエストを送信した場合，フィルタが正しく登録されない場合があります。
指定する値の詳細は「PureFlow WS1 トラフィックシェーパー NF7500 シリーズ コマンドリファレンス」を参照してください。

次に，JSON の指定方法について以下に示します。Instructions フィールドに OFPIT_APPLY_ACTIONS を，actions フィールドに OFPAT_EXPERIMENTER を指定し，EXPERIMENTER のデータ部に JSON 形式の文字列を指定してください。また，EXPERIMENTER ID には 0x00000091 を指定してください。

表 H-5　JSON の指定方法

CLI コマンドの種類	instructions	actions
フィルタ追加	OFPIT_APPLY_ACTIONS	OFPAT_EXPERIMENTER
フィルタ削除		

JSON で指定できるパラメータを以下に示します。JSON のパラメータには，必ず指定しなければいけないパラメータと，省略可能なパラメータがあります。省略したパラメータはデフォルト値として扱われます。JSON の形式については「付録 E WebAPI 詳細」を参照してください。

表 H-6　JSON キー一覧

CLI コマンドの種類	キー	値	相当する CLI コマンドとパラメータ
フィルタ追加	"type" （必須）	フィルタの種類	{bridge-ctrl | ethernet | ipv4 | ipv6}
	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"filter_name" （必須）	フィルタ名	filter <filter_name>
	"priority" （省略可）	フィルタ優先度	[priority <filter_pri>]
	"vid" （省略可）	VLAN ID	[vid {<VID>|none}]
	"inner-vid" （省略可）	INNER VLAN ID	[inner-vid {<VID>|none}]
	"sip list" （省略可）	ルールリスト名	[sip list <list_name>]
	"dip list" （省略可）	ルールリスト名	[dip list <list_name>]
	"sport list" （省略可）	ルールリスト名	[sport list <list_name>]
	"dport list" （省略可）	ルールリスト名	[dport list <list_name>]
フィルタ削除	"scenario_name" （必須）	シナリオ名	scenario <scenario_name>
	"filter_name" （省略可）	フィルタ名	[filter <filter_name>]

キーと値はすべて文字列で指定します。省略可能なパラメータは指定が不要な場合は記述不要です。キーにスペルミスがある場合，そのパラメータは無視されます。指定必須パラメータのスペルミスはエラーとなりますが，省略可能なパラメータのスペルミスや，未定義のパラメータはエラーとならないことに注意してください。

付録 I　　OpenFlowメッセージ詳細

本装置における OpenFlow メッセージの対応を示します。

- OFPT_HELLO

本装置でサポートします。

OpenFlow コントローラと本装置間でサポートする OpenFlow プロトコルのバージョンを交換します。

メッセージタイプ

OFPT_HELLO (0)

OpenFlow プロトコルヘッダの version フィールと Hello メッセージの bitmap フィールドのビット位置は以下のように定義されています。

表 I-1　Hello メッセージの bitmap フィールドのビット位置

OpenFlow バージョン	条件
1.0	0x01
1.1	0x02
1.2	0x03
1.3	0x04
1.4	0x05

本装置では，v1.3 のみをサポートする Hello メッセージを送信します（version フィールドは，0x04，Hello メッセージの type フィールドには OFPHET_VERSIONBITMAP(0x0001)，bitmaps フィールドには 0x00000010 を設定する）。

エラーメッセージの条件を以下に示します。

表 I-2　エラーメッセージの条件

エラータイプ	エラーコード	サポート	条件
OFPET_BAD_REQUEST	OFPBRC_BAD_TYPE	○	OpenFlow コントローラと Hello メッセージを交換した後に再度 Hello メッセージを受信したとき（OpenvSwitch の既存処理）
OFPET_HELLO_FAILED(0)	OFPHFC_INCOMPATIBLE(0)	○	OpenFlow コントローラが v1.3 をサポートしない
	OFPHFC_EPERM(1)	×	権限エラー

OFPHFC_INCOMPATIBLE のエラーメッセージについて

① OpenFlow プロトコルヘッダの version フィールドは，不一致になったバージョンの値を設定します（OpenFlow コントローラが識別できるようにするため）。

② データ部には，エラーの詳細を示す文字列を格納します。

例.

・ OpenFlow コントローラが v1.0 のみサポートしている場合
We support version 0x04, you support version 0x01, no common versions.

・ OpenFlow コントローラが v1.4 のみサポートしている場合
We support version 0x04, you support version 0x05, no common versions.

OpenFlow コントローラから OpenFlow プロトコルヘッダのみの Hello パケットを受信した場合

本装置は version フィールドに設定してあるバージョンのみをサポートしているコントローラと判断します。
エラーメッセージ送信後，RST（リセット）パケットにより TCP セッションを切断し，再接続します。
再接続回数：制限なし
再送間隔：約 10 秒以内

- OFPT_ECHO_REQUEST
- OFPT_ECHO_REPLY

本装置でサポートします。

OpenFlow コントローラと TCP セッションを確立後，本装置から 5 秒間隔で Echo 要求メッセージを送信します。

メッセージタイプ

OFPT_ECHO_REQUEST (2)

OFPT_ECHO_REPLY (3)

OpenFlow コントローラからの Echo 応答メッセージを（4 回連続で）受信できない場合は，RST（リセット）パケットにより TCP セッションを切断し，再接続します。

再接続回数：制限なし
再送間隔：約 10 秒以内

- OFPT_EXPERIMENTER

本装置でサポートします。
設定・表示系の CLI コマンドに対応する際に使用するメッセージです。
なお，Experimenter Multipart メッセージでも同様に，設定・表示系の CLI コマンドに対応しています。

メッセージタイプ

OFPT_EXPERIMENTER (4)

データ構造（struct ofp_experimenter_header）を以下に示します。

表 I-3　データ構造(struct ofp_experimenter_header)

項目	型	名前	説明
ヘッダ	struct ofp_header	header	OpenFlow プロトコルヘッダ
拡張 ID	uint32_t	experimenter	IEEE OUI（Anritsu：0x00000091）
拡張 タイプ	uint32_t	exp_type	任意の ID（無視する）
データ部	uint8_t	data[0]	WebAPI と同様に JSON 形式でパラメータを設定する {“command”：“add scenario”, …}

データ部の設定内容の詳細については「付録 H CLI コマンド対応 OpenFlow メッセージ詳細」を参照してください。

応答メッセージについて以下に示します。

表 I-4　応答メッセージ

種別		説明
設定系 CLI コマンド	正常リクエスト	何も返さない。
	異常リクエスト	エラーメッセージを送信します。
表示系 CLI コマンド	正常リクエスト	WebAPI と同様の JSON 形式の表示データを Experimenter メッセージのデータ部に設定し送信します。
	異常リクエスト	エラーメッセージを送信します。

エラーメッセージの条件を以下に示します。

表 I-5　エラーメッセージの条件

エラータイプ	エラーコード	条件
OFPET_BAD_REQUEST	OFPBRC_BAD_EXPERIMENTER	Experimenter フィールドが 0x00000091（Anritsu）以外
OFPET_BAD_REQUEST	OFPBRC_BAD_EXP_TYPE	本装置では，exp_type フィールドは任意とするため非サポート
OFPET_EXPERIMENTER	なし	CLI コマンドがエラーとなるリクエストを受信した場合※

※ 以下に Experimenter エラーメッセージのフォーマットを示します。

表 I-6　Experimenter エラーメッセージのフォーマット

項目	型	名前	説明
ヘッダ	struct ofp_header	header	OpenFlow プロトコルヘッダ
タイプ	uint16_t	type	OFPET_EXPERIMENTER
拡張タイプ	uint16_t	exp_type	本装置では，任意
拡張 ID	uint32_t	experimenter	IEEE OUI（Anritsu：0x00000091）
データ部	uint8_t	data[0]	本装置では，コマンドのエラーメッセージを返します。 以下の JSON 形式を設定する {“error”：“CLI エラーメッセージ”}

- OFPT_FEATURES_REQUEST
- OFPT_FEATURES_REPLY

本装置でサポートします。
OpenFlow 機能を OpenFlow コントローラと接続時に交換するためのメッセージです。
セッション確立時に OpenFlow コントローラが本装置に対して Features 要求メッセージを送信します。
本装置は OpenFlow コントローラに Features 応答メッセージを返します。

メッセージタイプ

OFPT_FEATURES_REQUEST(5)

OFPT_FEATURES_REPLY(6)

応答メッセージの項目を以下に示します。

表 I-7　応答メッセージの項目

項目	サポート	説明
データパス	○	スイッチを識別するためのユニーク ID 下位 48 bit は，本装置のシステムインタフェースの MAC アドレス 上位 16 bit は，0x0000
バッファ	○	バッファ領域に一度に格納できるパケット数（256 固定）
テーブル数	○	スイッチがサポートする flow テーブル数（1 固定）
接続種別	○	スイッチからコントローラへ接続の種類（メイン接続は 0，補助接続は 0 以外）（0固定）
機能	○	機能フラグを設定する（ポート統計情報のみ対応）（0x00000004 を固定）
OFPC_FLOW_STATS	×	フロー統計情報
OFPC_TABLE_STATS	×	テーブル統計情報
OFPC_PORT_STATS	○	ポート統計情報
OFPC_GROUP_STATS	×	グループ統計情報
OFPC_IP_REASM	×	IP フラグメントを再構成（リアセンブル）
OFPC_QUEUE_STATS	×	キュー統計情報
OFPC_PORT_BLOCKED	×	パケットのループを防止するためのトポロジループとブロックポートを検出

エラー処理はありません。上記で定義した応答を必ず送信します。

- OFPT_GET_CONFIG_REQUEST
- OFPT_GET_CONFIG_REPLY
- OFPT_SET_CONFIG

本装置でサポートします。
設定の場合，応答メッセージを返す必要はないため，本装置は応答メッセージを返しません。
取得の場合，本装置は OpenFlow コントローラに GET_CONFIG 応答メッセージを返します。
OpenFlow コントローラは正しく設定されているかどうかを確認するためには，GET_CONFIG 要求メッセージを送信します。

メッセージタイプ

OFPT_SET_CONFIG (9)

OFPT_GET_CONFIG_REQUEST (7)

OFPT_GET_CONFIG_REPLY(8)

SET_CONFIG メッセージと GET_CONFIG 応答メッセージの項目を以下に示します。

表 I-8　応答メッセージの項目

項目	サポート	説明
Flags	△	IP フラグメントの処理を設定する 以下ビットマップ OFPC_FRAG_NORMAL（フラグメントの処理を行わない） OFPC_FRAG_DROP（廃棄） OFPC_FRAG_REASM（リアセンブル） OFPC_FRAG_MASK 本装置では，OFPC_FRAG_NORMAL のみサポートする。
Miss send length	△	OpenFlow コントローラに送信される各パケットのバイト数を定義する 設定範囲：0～65535（0xFFFF）

上記設定項目は，本装置から OpenFlow コントローラへ送信するメッセージの IP フラグメント処理とバイト長に関する設定です。本装置では，Packet-in メッセージをサポートしないため，SET_CONFIG メッセージの設定内容を使用することはありません。本装置の内部処理としては，無視します。

エラーメッセージの条件を以下に示します。

表 I-9　エラーメッセージの条件

エラータイプ	エラーコード	サポート	条件
OFPET_SWITCH_CONFIG_FAILED(10)	OFPSCFC_BAD_FLAGS (0)	○	OFPC_FRAG_NORMAL 以外の Flags を受信
	OFPSCFC_BAD_LEN (1)	×	不正な Miss send length を受信
	OFPSCFC_EPERM (2)	×	権限エラー 本装置では，Role なしのため未サポート

➢ OFPT_FLOW_MOD

本装置でサポートします。
OpenFlow コントローラからフローエントリを追加・変更するメッセージです。
本装置ではフィルタの追加・削除に対応します。
設定内容の詳細については「付録 H CLI コマンド対応 OpenFlow メッセージ詳細」を参照してください。

メッセージタイプ

OFPT_FLOW_MOD (14)

各フィールドを以下に示します。

表 I-10　OFPT_FLOW_MOD メッセージのフィールド

フィールド	サポート	説明
cookie	×	
cookie_mask	×	
table_id	○	0 以外エラーとする。
idle_timeout	○	0 以外エラーとする。
hard_timeout	○	0 以外エラーとする。
priority	×	
buffer_id	×	OFP_NO_BUFFER(0xFFFFFFFF)
out_port	×	OFPP_ANY(0xFFFFFFFF)
out_group	×	OFPG_ANY(0xFFFFFFFF)

command フィールドを以下に示します。

表 I-11　command フィールド

項目	サポート	説明
OFPFC_ADD	○	追加（本装置の add filter に対応）
OFPFC_MODIFY	×	修正（マッチした全エントリが対象）
OFPFC_MODIFY_STRICT	×	修正（ワイルドカードと優先度を含めてマッチしたエントリが対象）
OFPFC_DELETE	×	削除（マッチした全エントリが対象）
OFPFC_DELETE_STRICT	○	削除（ワイルドカードと優先度を含めてマッチしたエントリが対象）（本装置の delete filter に対応）

flag フィールドを以下に示します。

表 I-12　flag フィールド

項目	サポート	説明
OFPFF_SEND_FLOW_REM	×	タイムアウトでフローエントリが消えるときコントローラに Flow Remove メッセージ送信　有効無効
OFPFF_CHECK_OVERLAP	×	フローテーブルの優先度競合時, エラーにするか
OFPFF_RESET_COUNTS	×	カウントリセット
OFPFF_NO_PKT_COUNTS	×	パケットカウント無効
OFPFF_NO_BYT_COUNTS	×	バイトカウント無効

Flow Match フィールドを以下に示します。

表 I-13　Flow Match フィールド

項目	サポート	説明
OFPXMT_OFB_IN_PORT	○	OpenFlow スイッチの入力ポート
OFPXMT_OFB_IN_PHY_PORT	×	OpenFlow スイッチの入力物理ポート IN_PORT 必須
OFPXMT_OFB_METADATA	×	フローテーブルのメタデータ
OFPXMT_OFB_ETH_DST	×	宛先イーサネットアドレス
OFPXMT_OFB_ETH_SRC	×	送信元イーサネットアドレス
OFPXMT_OFB_ETH_TYPE	○	イーサネットフレームタイプ
OFPXMT_OFB_VLAN_VID	×	VLAN ID
OFPXMT_OFB_VLAN_PCP	×	VLAN の優先度 VLAN_VID != NONE 必須
OFPXMT_OFB_IP_DSCP	×	DSCP ETH_TYPE=0x0800 or ETH_TYPE=0x86dd 必須
OFPXMT_OFB_IP_ECN	×	ECN ETH_TYPE=0x0800 or ETH_TYPE=0x86dd 必須
OFPXMT_OFB_IP_PROTO	○	IP プロトコル番号 ETH_TYPE=0x0800 or ETH_TYPE=0x86dd 必須
OFPXMT_OFB_IPV4_SRC	○	送信元 IPv4 アドレス ETH_TYPE=0x0800 必須
OFPXMT_OFB_IPV4_DST	○	宛先 IPv4 アドレス ETH_TYPE=0x0800 必須
OFPXMT_OFB_TCP_SRC	○	送信元 TCP ポート番号 IP_PROTO=6 必須
OFPXMT_OFB_TCP_DST	○	宛先 TCP ポート番号 IP_PROTO=6 必須
OFPXMT_OFB_UDP_SRC	○	送信元 UDP ポート番号 IP_PROTO=17 必須

項目	サポート	説明
OFPXMT_OFB_UDP_DST	○	宛先 UDP ポート番号 IP_PROTO=17 必須
OFPXMT_OFB_SCTP_SRC	×	送信元 SCTP ポート番号 IP_PROTO=132 必須
OFPXMT_OFB_SCTP_DST	×	宛先 SCTP ポート番号 IP_PROTO=132 必須
OFPXMT_OFB_ICMPV4_TYPE	×	IPv4ICMP のタイプ IP_PROTO=1 必須
OFPXMT_OFB_ICMPV4_CODE	×	IPv4ICMP のコード IP_PROTO=1 必須
OFPXMT_OFB_ARP_OP	×	ARP の OP コード ETH_TYPE=0x0806 必須
OFPXMT_OFB_ARP_SPA	×	ARP の送信元 IPv4 アドレス ETH_TYPE=0x0806 必須
OFPXMT_OFB_ARP_TPA	×	ARP の宛先 IPv4 アドレス ETH_TYPE=0x0806 必須
OFPXMT_OFB_ARP_SHA	×	ARP の送信元ハードウエアアドレス ETH_TYPE=0x0806 必須
OFPXMT_OFB_ARP_THA	×	ARP の宛先ハードウエアアドレ ETH_TYPE=0x0806 必須
OFPXMT_OFB_IPV6_SRC	○	送信元 IPv6 アドレス ETH_TYPE=0x86dd 必須
OFPXMT_OFB_IPV6_DST	○	宛先 IPv6 アドレス ETH_TYPE=0x86dd 必須
OFPXMT_OFB_IPV6_FLABEL	×	IPv6 フローラベル ETH_TYPE=0x86dd 必須
OFPXMT_OFB_ICMPV6_TYPE	×	ICMPv6 タイプ IP_PROTO=58 必須
OFPXMT_OFB_ICMPV6_CODE	×	ICMPv6 コード IP_PROTO=58 必須
OFPXMT_OFB_IPV6_ND_TARGET	×	Neighbor Discovery のターゲットアドレス ICMPV6_TYPE=135 or ICMPV6_TYPE=136 必須
OFPXMT_OFB_IPV6_ND_SLL	×	Neighbor Discovery の source Link-Layer ICMPV6_TYPE=135 必須
OFPXMT_OFB_IPV6_ND_TLL	×	Neighbor Discovery の target Link-Layer ICMPV6_TYPE=136 必須
OFPXMT_OFB_MPLS_LABEL	×	MPLS ラベル ETH_TYPE=0x8847or ETH_TYPE=0x8848 必須
OFPXMT_OFB_MPLS_TC	×	MPLS トラフィッククラス ETH_TYPE=0x8847or ETH_TYPE=0x8848 必須

項目	サポート	説明
OFPXMT_OFB_MPLS_BOS	×	1 つめの MPLS shim ヘッダに含まれる Bos(bottom of stack)ビット ETH_TYPE=0x8847or ETH_TYPE=0x8848 必須
OFPXMT_OFB_PBB_ISID	×	1 つめの PBB サービスインスタンスタグに含まれる I-SID ETH_TYPE=0x88e7 必須
OFPXMT_OFB_TUNNEL_ID	×	ロジカルポートに関連付けられたメタデータ
OFPXMT_OFB_IPV6_EXTHDR	×	IPv6 拡張ヘッダ用の pseudo フィールド ETH_TYPE=0x86dd 必須

Instruction タイプを以下に示します。

表 I-14　Instruction タイプ

項目	サポート	説明
OFPIT_GOTO_TABLE	×	指定したフローテーブルへ処理を引き継ぐ
OFPIT_WRITE_METADATA	×	以降のテーブルで参照できるメタデータをセットする
OFPIT_WRITE_ACTIONS	×	現在のアクションセットに指定されたアクションを追加する
OFPIT_APPLY_ACTIONS	○	アクションセットは変更せず，指定されたアクションを直ちに適用する
OFPIT_CLEAR_ACTIONS	×	現在のアクションセットのすべてのアクションを削除する
OFPIT_METER	×	指定したメーターにパケットを適用する
OFPIT_EXPERIMENTER	×	実験者用領域

エラーメッセージの条件を以下に示します。

表 I-15　エラーメッセージの条件

エラータイプ	エラーコード	サポート	条件
OFPET_FLOW_MOD_FAILED(5)	OFPFMFC_UNKNOWN(0)	×	予期しないエラー
	OFPFMFC_TABLE_FULL(1)	×	フィルタが最大件数登録されている場合
	OFPFMFC_BAD_TABLE_ID(2)	○	テーブル ID が 0 以外の場合
	OFPFMFC_OVERLAP(3)	×	CHECK_OVERLAP flag が設定されている場合の重複エラー
	OFPFMFC_EPERM(4)	×	権限エラー
	OFPFMFC_BAD_TIMEOUT(5)	○	非サポートの idle/hard timeout が指定されている場合（0 以外）
	OFPFMFC_BAD_COMMAND(6)	○	非サポートの command が指定されている場合
	OFPFMFC_BAD_FLAGS(7)	○	非サポートの flags が指定されている場合

エラータイプ	エラーコード	サポート	条件
OFPET_BAD_MATCH(4)	OFPBMC_BAD_TYPE(0)	○	非サポートの match type が指定されている場合（OFPMT_OXM 以外）
	OFPBMC_BAD_LEN(1)	○	マッチフィールドの length エラー
	OFPBMC_BAD_TAG(2)	×	サポートしていない tag/encap
	OFPBMC_BAD_DL_ADDR_MASK(3)	×	データリンクアドレスマスクをサポートしていない場合
	OFPBMC_BAD_NW_ADDR_MASK(4)	×	ネットワークアドレスマスクをサポートしていない場合
	OFPBMC_BAD_WILDCARDS(5)	×	非サポートのマスクまたは省略の組み合わせ
	OFPBMC_BAD_FIELD(6)	○	非サポートの Flow Match フィールドが指定された
	OFPBMC_BAD_VALUE(7)	×	非サポートの value が指定された
	OFPBMC_BAD_MASK(8)	×	非サポートの mask が指定された
	OFPBMC_BAD_PREREQ(9)	○	必須のフィールドが指定されていない場合
	OFPBMC_DUP_FIELD(10)	△	Flow Match フィールドが重複している
	OFPBMC_EPERM(11)	×	権限エラー

エラータイプ	エラーコード	サポート	条件
OFPET_BAD_INSTRUCTION(3)	OFPBIC_UNKNOWN_INST(0)	×	不明な instruction
	OFPBIC_UNSUP_INST(1)	○	非サポートの instruction を受信
	OFPBIC_BAD_TABLE_ID(2)	×	テーブル ID が 0 以外の場合
	OFPBIC_UNSUP_METADATA(3)	×	データパスによってサポートされていないメタデータ値
	OFPBIC_UNSUP_METADATA_MASK(4)	×	データパスによってサポートされていないメタデータのマスク値
	OFPBIC_BAD_EXPERIMENTER (5)	×	Experimenter フィールドが 0x00000091(Anritsu)以外
	OFPBIC_BAD_EXP_TYPE(6)	×	本装置では，exp_type フィールドは任意とするため非サポート
	OFPBIC_BAD_LEN(7)	×	instruction の length エラー
	OFPBIC_EPERM(8)	×	権限エラー

エラータイプ	エラーコード	サポート	条件
OFPET_BAD_ACTION (2)	OFPBAC_BAD_TYPE (0)	○	非サポートの action type が指定されている場合(Experimenter 以外)
	OFPBAC_BAD_LEN (1)	○	アクションフィールドの length エラー
	OFPBAC_BAD_EXPERIMENTER (2)	○	不明な ExperimenterID
	OFPBAC_BAD_EXP_TYPE (3)	×	ExperimenterID に対するアクションが不明
	OFPBAC_BAD_OUT_PORT (4)	×	出力ポート不正
	OFPBAC_BAD_ARGUMENT (5)	×	アクションの引数不正
	OFPBAC_EPERM (6)	×	パーミッションエラー
	OFPBAC_TOO_MANY (7)	×	アクションが多すぎて扱えない
	OFPBAC_BAD_QUEUE (8)	×	出力キュー不正
	OFPBAC_BAD_OUT_GROUP (9)	×	フォワードアクションのグループ ID が不正
	OFPBAC_MATCH_INCONSISTENT (10)	×	このマッチにアクションを適用できない
	OFPBAC_UNSUPPORTED_ORDER (11)	×	Apply-Action のアクションリストの順番が不正
	OFPBAC_BAD_TAG (12)	×	アクションが非サポートの Tag/encap を使用している
	OFPBAC_BAD_SET_TYPE (13)	×	SET_FIELD アクションの type が不正
	OFPBAC_BAD_SET_LEN (14)	×	SET_FIELD アクションの長さが不正
	OFPBAC_BAD_SET_ARGUMENT (15)	×	SET_FIELD アクションの引数が不正

- OFPT_MULTIPART_REQUEST
- OFPT_MULTIPART_REPLY

本装置でサポートします。

メッセージタイプ	マルチパートタイプ
OFPT_MULTIPART_REQUEST(18)	OFPMP_PORT_STATS (4)
OFPT_MULTIPART_REPLY(19)	

ポートカウンタに関する統計情報の取得するメッセージです。
要求メッセージのフレームフォーマットを以下に示します。

port_no フィールドには，ポート番号を指定します。

表 I-16　port_no フィールド

port_no	サポート	説明
1	○	管理ポート(mgmt0)
2	○	Network ポート(1/1)
3	○	Network ポート(1/2)
4	○	Network ポート(1/3)
5	○	Network ポート(1/4)
0, 6～0xffffeff	×	範囲外
OFPP_MAX (0xffffff00)	△	予約ポート エラーにしない。応答メッセージのデータ部は空とする。
OFPP_IN_PORT (0xfffffff8)	△	同上
OFPP_TABLE (0xfffffff9)	△	同上
OFPP_NORMAL (0xfffffffa)	△	同上
OFPP_FLOOD (0xfffffffb)	△	同上
OFPP_ALL (0xfffffffc)	△	同上
OFPP_CONTROLLER (0xfffffffd)	△	同上
OFPP_LOCAL (0xfffffffe)	○	予約ポート，ローカルポートの統計情報を応答する。
OFPP_ANY (0xffffffff)	○	予約ポート，全ポートの統計情報を応答する。

統計情報の項目（上図の body 部）を以下に示します。（show counter コマンドと同様）

表 I-17　統計情報の項目

項目	サポート	説明
port_no	○	ポート番号，全ポート指定時は OFPP_ANY をセット
rx_packets	○	受信したパケットの総数
tx_packets	○	送信したパケットの総数
rx_bytes	○	受信したパケットの総バイト数
tx_bytes	○	送信したパケットの総バイト数
rx_dropped	○	受信時にドロップしたパケットの総数
tx_dropped	×	送信時にドロップしたパケットの総数
rx_errors	○	受信エラーとなったパケットの総数
tx_errors	×	送信エラーとなったパケットの総数
rx_frame_err	×	受信時にフレーム割り当てのエラーが発生した回数
rx_over_err	×	受信時にパケットのオーバーランが発生してロストした回数
rx_crc_err	×	受信時に CRC エラーが発生した回数
collisions	○	イーサネットレイヤでコリジョンが発生した回数
duration_sec	△	ポートが有効になってからの時間（sec） 管理ポート（mgmt0），Network ポート（1/1～1/4）では未サポートとする。
duration_nsec	△	ポートが有効になってからの時間の sec 未満の桁（nsec） 管理ポート（mgmt0），Network ポート（1/1～1/4）では未サポートとする。

未サポートの項目については，ALL 0xFF 設定します。（port_no と duration は 32bit，その他 64bit）

エラーメッセージの条件を以下に示します。

表 I-18　エラーメッセージの条件

エラータイプ	エラーコード	条件
OFPET_BAD_REQUEST(1)	OFPBRC_BAD_PORT(11)	ポート番号が範囲外
OFPET_BAD_REQUEST(1)	FPBRC_MULTIPART_ BUFFER_OVERFLOW(13)	flags フィールドに OFPMPF_REQ_MORE(1)が設定されている場合※

※ 要求メッセージの flags フィールドに OFPMPF_REQ_MORE(1)が設定されている場合，上記エラーとなります。上記フラグを設定しないようにしてください。

メッセージタイプ	マルチパートタイプ
OFPT_MULTIPART_REQUEST(18)	OFPMP_PORT_DESC (13)
OFPT_MULTIPART_REPLY(19)	

全ポートの Description（状態，速度等）を取得するメッセージです。

本装置では，OFPP_LOCAL ポート，管理ポート，Network ポート（1/1～1/4）の Description（状態，速度等）を応答します。

応答メッセージの項目（上図の body 部）を以下に示します。

表 I-19　応答メッセージの項目

項目	サポート	説明
Port no	○	ポート番号 OFPP_LOCAL ポート：0xfffffffe 管理ポート：0x1 Network ポート：1/1：0x2，1/2：0x3，1/3：0x4，1/4：0x5
Hw addr	×	MAC アドレス OFPP_LOCAL ポート：システムインタフェースの MAC アドレス 管理ポート：システムインタフェースの MAC アドレス Network ポート：チャネルインタフェースの MAC アドレス
Name	○	ポート名 OFPP_LOCAL ポート：br0 管理ポート：mgmt0 Network ポート：1/1～1/4
Config	×	設定（以下ビットマップ） OFPPC_PORT_DOWN，OFPPC_NO_RECV OFPPC_NO_FWD，OFPPC_NO_PACKET_IN OFPP_LOCAL ポート：0x1(OFPPC_PORT_DOWN) 管理ポート：0x0 Network ポート：0x0
State	△	状態（以下ビットマップ） OFPPS_LINK_DOWN，OFPPS_BLOCKED OFPPS_LIVE OFPP_LOCAL ポート：0x1(OFPPS_LINK_DOWN)のみサポートする。 管理ポート：リンクダウン取得不可 Network ポート：0x0（リンクアップ），0x1（リンクダウン）

項目	サポート	説明
Current	×	現在の特長・機能（以下ビットマップ） OFPPF_10MB_HD, OFPPF_10MB_FD OFPPF_100MB_HD, OFPPF_100MB_FD OFPPF_1GB_HD, OFPPF_1GB_FD OFPPF_10GB_FD, OFPPF_40GB_FD OFPPF_100GB_FD, OFPPF_1TB_FD OFPPF_OTHER, OFPPF_COPPER OFPPF_FIBER, OFPPF_AUTONEG OFPPF_PAUSE, OFPPF_PAUSE_ASYM 非サポートのため全ビット 0 を設定する。
Advertised	×	広告されている特長・機能（ビットマップは Current と同様） 非サポートのため全ビット 0 を設定する。
Supported	×	サポートされている特長・機能（ビットマップは Current と同様） 非サポートのため全ビット 0 を設定する。
Peer	×	ピア（OpenFlow コントローラ）が広告した特長・機能（ビットマップは Current と同様） 非サポートのため全ビット 0 を設定する。
Curr speed	○	現在速度（単位 kb/s） show port コマンドの Oper speed（通信速度）から設定する。
Max speed	○	最大速度 OFPP_LOCAL ポート：0 管理ポート：1G = 1,000,000[kb/s] show port コマンドの Port type（ポート種別）から設定する。

エラーメッセージの条件を以下に示す。

表 I-20　エラーメッセージの条件

エラータイプ	エラーコード	条件
OFPET_BAD_REQUEST(1)	FPBRC_MULTIPART_ BUFFER_OVERFLOW (13)	flags フィールドに OFPMPF_REQ_MORE(1)が設定されている場合※

※ 要求メッセージの flags フィールドに OFPMPF_REQ_MORE(1)が設定されている場合，上記エラーとなります。上記フラグを設定しないようにしてください。

メッセージタイプ	マルチパートタイプ
OFPT_MULTIPART_REQUEST(18)	OFPMP_EXPERIMENTER (0xffff)
OFPT_MULTIPART_REPLY(19)	

設定・表示系の CLI コマンドに対応する際に使用します。
なお，Experimenter メッセージでも同様に，設定・表示系の CLI コマンドに対応しています。

拡張ヘッダ以降のデータ構造（struct ofp_experimenter_multipart_header）を以下に示します。

表 I-21　データ構造(struct ofp_experimenter_multipart_header)

項目	型	名前	説明
拡張 ID	uint32_t	experimenter	IEEE OUI（Anritsu：0x00000091）
拡張タイプ	uint32_t	exp_type	任意の ID（無視する）
データ部	uint8_t	data[0]	WebAPI と同様に JSON 形式でパラメータを設定する {“command”：“show scenario counter”, …}

データ部の設定内容の詳細については「付録 H CLI コマンド対応 OpenFlow メッセージ詳細」を参照してください。

シナリオカウンタに関する統計情報の取得については，以下に項目を応答メッセージのデータ部に JSON 形式で設定し送信します。（show scenario counter と同様）

表 I-22　応答メッセージの項目

項目	表示	説明
Scenario	○	シナリオインデックスとシナリオ名 ポートシナリオのシナリオインデックスはポート 1 では 4097，ポート 2 では 4098，ポート 3 では 4099，ポート 4 では 4100 が表示される
Rate Control Unit	×	帯域制御に対する設定内容
Default Queue	×	デフォルトキューに対する設定内容
Attached Filters	×	シナリオに追加されているフィルタのフィルタ名
Rx Octets	○	受信したパケットのバイト数
Rx Packets	○	受信したパケット数
Tx Octets	○	送信したパケットのバイト数
Tx Packets	○	送信したパケット数
Discard Octets	○	廃棄したパケットのバイト数
Discard Packets	○	廃棄したパケット数

その他の表示系コマンドについては，WebAPI 同様の JSON 形式の表示データを応答メッセージのデータ部に設定し送信します。

応答メッセージについて以下に示します。

表 I-23　応答メッセージ

種別		説明
設定系 CLI コマンド	正常リクエスト	Experimenter Multipart の応答メッセージを送信します。 データ部は空とします。
	異常リクエスト	エラーメッセージを送信します。
表示系 CLI コマンド	正常リクエスト	WebAPI と同様の JSON 形式の表示データを Experimenter Multipart の応答メッセージのデータ部に設定し送信します。
	異常リクエスト	エラーメッセージを送信します。

エラーメッセージの条件を以下に示します。

表 I-24　エラーメッセージの条件

エラータイプ	エラーコード	条件
OFPET_BAD_REQUEST	OFPBRC_BAD_EXPERIMENTER	Experimenter フィールドが 0x00000091（Anritsu）以外
OFPET_BAD_REQUEST	OFPBRC_BAD_EXP_TYPE	本装置では，exp_type フィールドは任意とするため非サポート
OFPET_BAD_REQUEST	FPBRC_MULTIPART_BUFFER_OVERFLOW	flags フィールドに OFPMPF_REQ_MORE(1)が設定されている場合※1
OFPET_EXPERIMENTER	なし	CLI コマンドがエラーとなるリクエストを受信した場合※2

※1 要求メッセージの flags フィールドに OFPMPF_REQ_MORE(1)が設定されている場合，上記エラーとなります。上記フラグを設定しないようにしてください。

※2 以下に Experimenter エラーメッセージのフォーマットを示します。

表 I-25　Experimenter エラーメッセージのフォーマット

項目	型	名前	説明
ヘッダ	struct ofp_header	header	OpenFlow プロトコルヘッダ
タイプ	uint16_t	type	OFPET_EXPERIMENTER
拡張タイプ	uint16_t	exp_type	本装置では，任意とする（無視する）。
拡張 ID	uint32_t	experimenter	IEEE OUI（Anritsu：0x00000091）
データ部	uint8_t	data[0]	本装置では，コマンドのエラーメッセージを返します。 以下の JSON 形式を設定する {“error”：“CLI エラーメッセージ”}

- OFPT_BARRIER_REQUEST
- OFPT_BARRIER_REPLY

本装置でサポートします。

メッセージタイプ

OFPT_BARRIER_REQUEST (20)

OFPT_BARRIER_REPLY(21)

コントローラの要求に対する処理が完了したことを確認する際に用いられるメッセージです。

OpenFlow コントローラは，本装置でこれまで受信したメッセージ処理が完了したか確認するために，Barrier 要求メッセージを送信します。本装置は，これまで受信したメッセージ処理が完了した後に，Barrier 応答メッセージを送信します。

- OFPT_ROLE_REQUEST
- OFPT_ROLE_REPLY

本装置でサポートします。

メッセージタイプ

OFPT_ROLE_REQUEST (24)

OFPT_ROLE_REPLY(25)

OpenFlow コントローラの役割を通知する際に用いられるメッセージです。

OpenFlow コントローラは，本装置に役割を通知するために，Role 要求メッセージを送信します。本装置は，Role 応答メッセージを送信します。

Anritsu

アンリツ株式会社

管理番号：

NF7500-W013J

Printed in Japan