SNMP による OSPF 構成管理

目次

概要

Open Shortest Path First (OSPF) ルーティング プロトコルは、RFC 2328 OSPF Version 2 により定義されています。 この文書の目的は、組織が OSPF の設計計画に照らし合わせて OSPF の展開を検証するために構成管理の手順を実行し、また、意図した設計との長期的な一貫性が保証されるようにするために OSPF の展開を定期的に監査できるようにする、手順の枠組みについて説明することです。

この文書では、ITU-T によって定義された FCAPS（fault, configuration, accounting/inventory, performance, security）モデルの構成管理機能に重点を置いています。 構成管理は、ITU-T M.3400 で定義され、NE（ネットワーク要素）の管理、識別、データ収集、およびデータ提供などの機能を提供します。

この文書の情報は、次に示すいくつかの主要なセクションに分けて説明されています。

「OSPF バックグラウンド」の項では、OSPF の展開における重要な側面の背景となる情報など、OSPF の技術的な概要について説明しています。

「プロセス定義」の項では、OSPF 構成管理を行うために使用するプロセス定義の概要について説明します。 プロセスの詳細は、目標、パフォーマンス インジケータ、入力、出力、および個々のタスクの観点から説明されています。

「タスク定義」の項では、プロセス タスク定義について詳しく説明しています。 個々のタスクについては、目的、タスクの入力、タスクの出力、タスクを実行するために必要なリソース、およびタスクを実装するために必要な職務上の技能の観点から説明しています。

「データの識別」のセクションでは、OSPF のためのデータの識別について説明しています。 データの識別では、情報の発信元または情報の所在地を判断します。 たとえば、情報は、Simple Network Management Protocol（SNMP; 簡易ネットワーク管理プロトコル）の Management Information Base（MIB; 管理情報ベース）、Syslog によって生成された ログ ファイル、または Command Line Interface（CLI; コマンドライン インターフェイス）からのみアクセス可能な内部データ構造などの中にシステムによって構築されます。

「データ収集」のセクションでは、OSPF データの収集について説明しています。 データ収集は、データの場所と密接な関係があります。 たとえば、SNMP MIB データは、トラップ、Remote Monitoring（RMON; リモート モニタリング）のアラームやイベント、またはポーリングなど、いくつかのメカニズムで収集されます。 内部データ構造によって保持されているデータは、自動スクリプトか、あるいはユーザがそのシステムに手動でログインして CLI コマンドを発行し、その出力を記録するという方法によって収集されます。

「データの表示」セクションでは、データがレポート形式で表現される方法の例について説明します。 それらのデータは識別され、収集された後、解析されます。 この文書では、OSPF 構成データの記録や比較に使用される可能性のあるレポートの例を紹介します。

「民間および公共のインターネット監視ツール」、「SNMP ポーリング データ」、および「データ収集アルゴリズムの例」の各項では、OSPF 構成管理の手順を実装するためのツールの開発に関する情報について説明しています。

OSPF の背景

OSPF は、単一の自律システムで使用されるように設計された内部ゲートウェイ プロトコルです。 OSPF では、リンクステートまたは shortest-path first（SPF; 最短パス優先）ベースのテクノロジーを使用しています。これに対し、Routing Information Protocol（RIP; ルーティング情報プロトコル）などのルーティング プロトコルでは、ディスタンスベクターまたはベルマンフォードのテクノロジーが使用されています。 個々の link-state advertisement（LSA; リンクステート アドバタイズメント）では、自律システムの全体など、OSPF ルーティング ドメインの一部について説明しています。 このような LSA はルーティング ドメイン全体にフラッドされ、リンクステート データベースを形成します。 ドメイン内の各ルータでは、全く同一のリンクステート データベースを保持します。 リンクステート データベースの同期は、信頼性の高いフラッディング アルゴリズムによって維持されています。 各ルータでは、このリンクステート データベースから

最短経路のツリーを計算し、計算を行っているルータ自身をツリーのルートとしてルーティング テーブルを構築します。 この計算は、通常はダイクストラのアルゴリズムと呼ばれます。

LSA は小さなもので、各 LSA は OSPF ルーティング ドメインの小さな一部分についての説明をします。具体的には、 1 台のルータの近隣情報、1 つのトランジット ネットワークの近隣情報、1 つの工リア間経路、または 1 つの外部経路などです。

次の表では、OSPF の主要な機能が説明されています。

機能	説明
アジャセンシー関係	OSPF ルータのペアが隣接関係になったとき、この 2 つのルータではデータベースのサマリーを OSPF データベース交換パケットの形式で交換し、それぞれのリンクステート データベースを同期させます。 その後、隣接関係ルータは、信頼性の高いフラッディング アルゴリズムを使用して、それぞれのリンクステート データベースの同期を維持します。 シリアル回線で接続されているルータは、常に隣接関係にあります。 マルチアクセス ネットワーク（イーサネット）では、そのネットワークに接続されているルータはすべて、 designated router（DR; 代表ルータ）および backup designated router（BDR; バックアップ代表ルータ）の両方と隣接関係を持ちます。
指定ルータ	すべてのマルチアクセス ネットワークにおいて 1 台の代表ルータが選ばれている場合、ネットワークのローカル環境を表すネットワーク LSA がその代表ルータから発信されます。 また、代表ルータはフラッディング アルゴリズムにおいても特別な役割を持っています。フラッディング処理では、ネットワーク上のすべてのルータが代表ルータに対して LSA を送受信することにより、リンクステート データベースを同期させています。
バックアップ指定ルータ	現在の DR が動作しなくなった場合には、マルチアクセス ネットワーク上で BDR が選ばれ、DR の移行を速やかに行います。 BDR が処理を引き継いだ場合、その local-area network（LAN; ローカルエリア ネットワーク）で隣接関係処理を行う必要はありません。 また、DR の消失が通知される以前での DR 不在の状態でも、BDR により信頼性の高いフラッディング アルゴリズムが続行できます。
非ブロードキャスト マルチアクセス ネットワークのサ	OSPF では、フレームリレーの public data network（PDN; 公衆データ網）などのネットワークを LAN のように扱います。 ただし、これらのネットワークに接続しているルータが初めて相互に認識するためには、構成情報の追加が必要です。

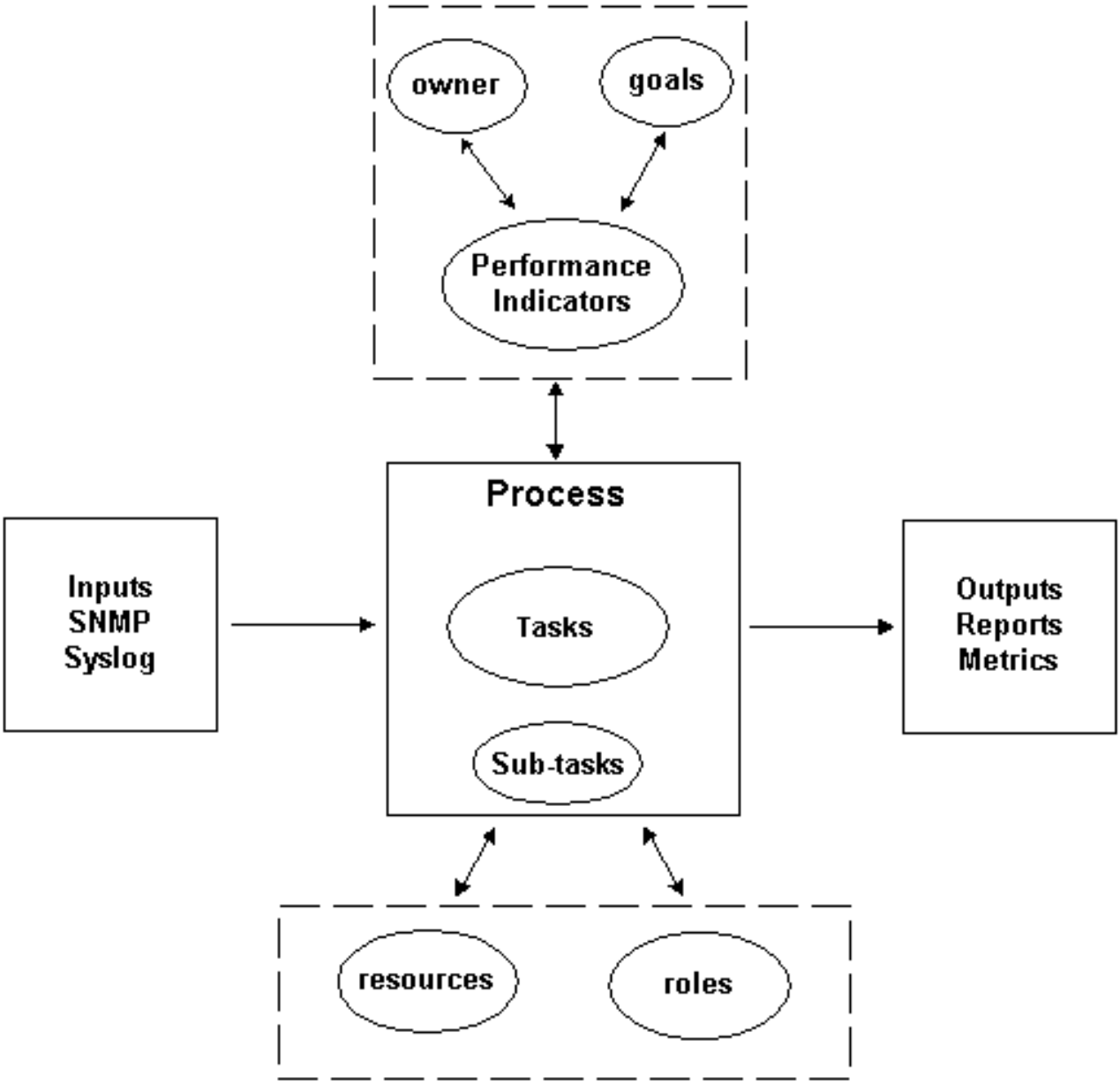
ポート	
OSPF 構成管理エリア	OSPF では、自律システムをエリアに区分することができます。 これにより、さらに高いレベルのルーティング保護が可能となり、エリア内のルーティングはエリア外のすべての情報から保護されるようになります。 また、自律システムをエリアに分割することで、CPU サイクルの面でダイクストラ処理の負担が軽減されます。
仮想リンク	OSPF では、仮想リンクの設定を可能にすることで、自律システムでのエリアのレイアウトに関するトポロジ制限を解消しています。
ルーティング プロトコル交換の認証	OSPF ルータがルーティング プロトコルのパケットを受信するたびに、オプションでそのパケットの処理前に認証を行うことができます。
柔軟性に富んだルーティング メトリック	OSPF では、メトリックは発信ルータのインターフェイスに割り当てられています。 このパスのコストは、そのパスを構成しているインターフェイスの合計になります。 デフォルトでは、ルーティング メトリックはその回線の帯域幅から導かれます。 ルーティング メトリックはネットワーク管理者によって割り当てられ、ネットワークの特性を表す遅延、帯域幅、およびコストなどと組み合わせて表現されます。
等コスト マルチパス	1 つの宛先に対して最適なコストの経路が複数ある場合、OSPF ではこれらを検出し、その宛先へのトラフィックの負荷分散に使用します。
可変長サブネットのサポート	ネットワーク マスクとそれぞれのアドバタイズされた宛先を搬送することにより、可変長のサブネット マスクをサポートしています。
スタブ エリアのサポート	メモリが不足しているルータをサポートするために、エリアをスタブとして設定できます。 外部 LSA は、このスタブ エリアへはフラッドされません。 スタブ エリアにある外部宛先へのルーティングは、デフォルトのみをベースとしています。

プロセス定義

プロセス定義とは、特定の目的を達成するためにエージェントによって実行される一連の動作、活動、および変更を意味します。

プロセス制御とは、プロセスを効果的かつ効率的に実行することを目的とした計画および調整の過程を意味します。

図で表すと、次のようになります。



プロセスの出力は、組織によって定義された運用基準と、ビジネス上の目的に準拠している必要があります。 一連の基準に準拠しているプロセスは、反復、測定、および管理が可能であり、ビジネス上の目的の達成に貢献するため、効果的であると考えられます。 また、最小限の労力で活動を実行できるプロセスは、効率的であると考えられます。

プロセス オーナー

プロセスは、さまざまな組織的境界をまたがります。 したがって、プロセス定義に責任を負うプロセス所有者は 1 人だけにする必要があります。 所有者は、そのプロセスが効果的および効率的であるかどうかを判断し、レポートする際の中心となります。 そのプロセスが効果的または効率的ではない場合、プロセス所有者はそのプロセスの修正を余儀なくされます。 プロセスを修正する際には、変更管理とチェックの手順が原則となります。

プロセスの目的

プロセスの目標は、プロセス定義の方向付けと範囲を設定するために定められます。 また、目標はプロセスの有効性を測定するためのメトリックを定義するためにも使用されます。

このプロセスの目標は、OSPF 実装の展開した構成を意図した設計に照らし合わせて検証するためのフレームワークを提供し、さらに OSPF の展開を定期的に監査して、意図した設計と長期にわたって整合性が保たれるようにするためのメカニズムを提供することです。

プロセス パフォーマンス インジケータ

プロセス性能インジケータは、プロセス定義の有効性を測定するために使用されます。 パフォーマンス インジケータは、測定および定量化が可能な基準である必要があります。 次に一覧されている性能インジケータは、数値化するものか、時間で測定するものかのいずれかです。 OSPF 構成管理の性能インジケータは、次のように定義されています。

- プロセス全体を一巡するために必要な時間の長さ。
- OSPF に関係する問題を、ユーザに影響を与える前に予防的に検出するために必要な実行頻度。
- プロセスの実行に関連するネットワークの負荷。
- プロセスによって推奨される修正処理の回数。
- プロセスの結果として実施された修正操作の数。
- 修正処理を実行するために必要な時間の長さ。
- 修正処理を実行するために必要な時間の長さ。
- 修正操作の未処理件数
- OSPF に関する問題に起因するダウンタイム。
- シード ファイル内で追加、削除、または修正された項目の数。 これは正確性と安定性を示します。

プロセスの入力

プロセスの入力は、プロセスの基準と前提条件を定義するために使用されます。 プロセス入力の識別により、外部の依存関係に関する情報が何度も提供されます。 OSPF 構成管理に関連する入力の一覧は次のとおりです。

- OSPF 設計文書
- SNMP ポーリングによって収集された OSPF MIB データ
- syslog 情報

プロセスの出力

プロセスの出力は、次のように定義されます。

- このドキュメントの「データの表示」の項で定義された OSPF 構成レポート
- 修正処理を実行するための OSPF 構成の推奨

タスク定義

以降のセクションでは、OSPF 構成管理に関連する初期化タスクと反復タスクを定義します。

初期化タスク

初期化タスクは、プロセスを実装するときに 1 回実行されます。プロセスが反復される際には実行しないようにします。

必須タスクの確認

必須タスクの確認中に、いずれかのタスクが実施されていないか、あるいはこの手順の要求に応えるだけの十分な情報が提供されていない場合は、この事実をプロセス所有者が文書化し、管理者に提出する必要があります 次の表では、必須の初期化タスクを概説しています。

必須タスク	説明
タスクの目的と入力	1. OSPF 設計文書が存在し、そのネットワーク設計文書の中に次の情報が明確に記されていることを確認します。 エリア定義：名前、アドレス範囲、およびエリアのタイプエリア境界ルータおよび自律システム境界ルータ（ABR/ASBR）の識別DR/BDR の識別エリアに割り当てられるインターネット レジストリ（IR）ノードとインターフェイス 2. SNMP 標準構成のテンプレートを使用して、ネットワークに SNMP が構成されていることを確認します。 **注:** これは、後ほどシードファイルを使用するための入力として使用されます。 3. Syslog 標準構成のテンプレートを使用して、ネットワークに Syslog が展開されていることを確認します。
タスクの出力	タスク出力は、必須タスクの条件に関するステータス レポートです。 サポートされているタスクのいずれかが効率的でないと思われる場合は、プロセス所有者が要求を発行し、サポートされているプロセスが更新されるようにします。 サポートされているプロセスを更新できない場合は、このプロセスの影響について評価を行います。
タスクの役割	ネットワーク エンジニアのスキル セット

シード ファイルの作成

OSPF 構成管理プロセスでは、シード ファイルを作成して、ネットワーク調査機能の必要性をなくすことが必要です。 シード ファイルには、OSPF プロセスによって管理されるルータのセットが記録され、組織の変更管理プロセスとの調整の中心として使用されます。 たとえば、ネットワークに新しいノードが組み込まれる場合、これらは OSPF シード ファイルに追加される必要があります。 セキュリティ上の理由で SNMP コミュニティ名に変更を加えた場合は、それらの変更をシード ファイルに反映する必要があります。 次の表では、シード ファイルの作成手順を概説

しています。

Process	説明
タスクの目的	OSPF 構成管理ソフトウェアを初期化するために使用されるシード ファイルを作成します。 シード ファイルの形式は、OSPF 構成管理プロセスの実装に使用されるリソースによって異なります。 カスタム スクリプトが作成される場合、シード ファイルの形式は、ソフトウェアの設計によって定義されます。 ネットワーク管理システム (NMS) が使用されている場合、シード ファイルの形式は、NMS の文書によって定義されています。
タスクの入力	1. シード ファイルのフォーマットします。 2. OSPF 設計文書を参照して、次のデータを割り出します。 全ノードの IP アドレス SNMP コミュニティ ストリングTelnet および CLI ログインのアカウントとパスワード 3. ネットワーク変更管理プロセスをスケジュールし、名前を問い合わせます。
タスクの出力	OSPF 構成管理プロセスのためのシード ファイル。
タスクのリソース	• 商用 NMS システム • カスタム開発されたソフトウェア システム • 手作業による処理：各ネットワーク要素にログインし、コマンドラインを発行して、その出力を記録します。
タスクの役割	• NMS： ネットワーク エンジニア、NMS 管理者、NMS スクリプト スキル セット。 • カスタム スクリプト：ネットワーク エンジニアおよび NMS スクリプト スキル セット。 • 手作業による処理：ネットワーク エンジニア。

反復タスク

反復タスクは、プロセスの反復ごとに実行され、性能インジケータを向上させるためにその頻度が判別され、修正されます。

シード ファイルを管理する

シード ファイルは、OSPF 構成管理プロセスを効果的に実装するために重要なものです。 したがって、シード ファイルの現在の状態を積極的に管理する必要があります。 シード ファイルの内容に影響を及ぼすようなネットワークの変更は、OSPF 構成管理プロセスの所有者が追跡する必要があります。

Process	説明
タスクの目的	1. ネットワークの移動、追加、変更、およびネットワーク構成の変更を管理する、組織機能による追跡や相互対話を通じて、現在のシード ファイルを維持します。 2. シード ファイルに対するバージョン制御とバックアップ制御を行います。
タスクの入力	1. 移動、追加、変更など、シード ファイルの内容に影響を及ぼすものに関する変更管理からの情報。 2. シード ファイルの内容に影響を及ぼすエンジニアリングおよび設計からの情報。
タスクの出力	1. 現在のシード ファイルに関する週単位のレポート。 2. シード ファイルのバックアップの場所と復元方法に関する定義と文書。
タスクのリソース	• 商用 NMS システム • カスタム開発されたソフトウェア システム • 手作業による処理：各ネットワーク要素にログインし、コマンドラインを発行して、その出力を記録します。
タスクの役割	• NMS： ネットワーク エンジニア、NMS 管理者、NMS スクリプト スキル セット。 • カスタム スクリプト：ネットワーク エンジニアおよび NMS スクリプト スキル セット。 • 手作業による処理：ネットワーク エンジニア。

OSPF スキャンの実行

OSPF スキャンの実行には、次の 2 つの手順が使用されます。

1. データ収集。
2. データの分析

プロセスの使用方法によっては、これらの 2 つの手順の頻度が変わります。 たとえば、このプロセスをインストールの変更の確認に使用することができます。 この場合、変更の前後にデータ収集が実行され、変更後にデータの解析が実行されて、変更が正しく行われたかどうかが判断されます。

このプロセスが OSPF 構成管理の設計の記録を確認するために使用される場合は、データ収集と解析の頻度は、ネットワークで変更が行われる頻度によって変化します。 たとえば、ネットワーク上で行われる変更の数が著しく多い場合、設計の検証は 1 週間に 1 度行います。 ネットワーク上での変更の数が非常に少ない場合は、設計の検証は 1 か月に 1 度行う程度です。

OSPF レポートの確認

OSPF 構成管理レポートの形式は、OSPF 構成管理プロセスの実装に使用されるリソースによっ

て異なります。 次の表では、カスタム開発されたレポートの推奨される形式について説明しています。

[レポート（Report）]	書式
タスクの入力	OSPF 構成管理レポートについては、このドキュメントの「データの表示」の項を参照してください。
タスクの出力	スキャン レコードと論理設計レコードの間で問題が見つかった場合は、どちらの項目が正しく、どちらの項目が誤っているかについて、判断を下す必要があります。 誤った項目は修正する必要があります。 これには、設計レコードの変更やネットワークの変更順序が関係している場合があります。
タスクのリソース	• 商用 NMS システム • カスタム開発されたソフトウェア システム • 手作業：各ネットワーク要素にログインし、コマンドラインを発行して、その出力を記録します。
タスクの役割	• NMS： ネットワーク エンジニア、NMS 管理者、NMS スクリプト スキル セット。 • カスタム スクリプト：ネットワーク エンジニアおよび NMS スクリプト スキル セット。 • 手作業による処理：ネットワーク エンジニア。

データの識別

一般的なデータの特徴

次の表では、OSPF 構成管理に適用されるデータについて説明します。

データ	説明
OSPF エリア	ルータが接続されたエリアについて説明する情報。次のものが含まれます。 • エリア ID • エリア認証 • SPF の実行 • エリア内の ABR の数 • エリア内の ASBR の数 • エリア LSA の数：エリア内のルータ全体の整合性 • エリア LSA のチェックサム：エリア内のルータ全体の整合性

	• エリアごとのアドレッシング エラーによるパケット廃棄の頻度 • エリアごとのルーティング処理によるプロトコル パケットの廃棄の頻度 • エリアごとにルートが見つからない状態によるルーテッド パケットの廃棄の頻度
OSPF インターフェイス	OSPF の観点から見たインターフェイスの説明。次の項目が含まれます。 • IP アドレス • エリア ID • 管理ステータス • このインターフェイスに割り当てられた OSPF メトリック • このインターフェイスに割り当てられた OSPF タイマー • OSPF の状態
OSPF 近隣ルータの状態	OSPF 近隣ルータの説明。 • 近接ルータのルータ ID • 近隣ルータの状態 • 近隣ルータでのイベント：ネイバー関係により状態が変化したり、エラーが発生した回数。 • 近隣ルータの再送信キュー：再送信キューの現在の長さ。

SNMP データの識別

シスコは現在 RFC 1253 OSPF Version 2 MIB をサポートしています。 RFCRFC 1253 には、OSPF の SNMP トラップ定義は含まれていません。 OSPF MIB の最新バージョンは RFC 1850 OSPF Version 2 です。 SNMP トラップは、RFC 1850 で OSPF 向けに定義されています。 RFC 1850 は、シスコの OSPF MIB の実装ではサポートされていません。

詳細は、このドキュメントの「SNMP ポーリング データ」の項を参照してください。

プラットフォームおよびコードバージョンでサポートされている MIB に関する詳細なリストについては、『Cisco ネットワーク管理用ソフトウェア』のページを参照してください。

RMON データの識別

この手順で必要な RMON 特有のデータはありません。

Syslog データの識別

一般的には、Syslog によって、異なるテクノロジーに対するサービス固有のメッセージが生成されます。 syslog 情報は障害や性能の管理に適していますが、ここで提供される情報は参照用です。 シスコのデバイスによって生成される OSPF Syslog 情報の例は、『OSPF エラー メッセージ』を参照してください。

ファシリティによるシステム メッセージの全リストは、『メッセージと回復の手順』を参照してください。

Cisco IOS CLI データの識別

このバージョンの OSPF 構成管理手順では、CLI データは必要ありません。

データ収集

SNMP データ収集

次の表では、SNMP データの収集に関する各種のコンポーネントについて説明します。

SNMP データのコンポーネント	定義
一般的な SNMP の設定	SNMP 設定のベスト プラクティスに関する一般的な情報については、『SNMP の設定』を参照してください。
サービスに固有の SNMP 構成	この手順に必要なサービス固有の SNMP 構成はありません。
SNMP MIB の要件	前述の「データの識別」セクションを参照してください。
SNMP MIB ポーリングの収集	SNMP ポーリング データは民間システム（hp OpenView など）またはカスタム スクリプトにより収集されます。 収集アルゴリズムの詳細は、このドキュメントの「データ収集アルゴリズムの例」の項を参照してください。
SNMP MIB トラップの収集	シスコのデバイスでサポートされている OSPF MIB の現在のバージョンでは、SNMP トラップをサポートしていません。この手順に必要な SNMP トラップはありません。

RMON データの収集

この手順の現在のバージョンでは、RMON 構成とデータは必要ありません。

Syslog データの収集

一般的な syslog 構成のガイドラインは、この文書の範囲外です。 詳細については、『単一の内部ネットワークでの Cisco Secure PIX Firewall の設定とトラブルシューティング』を参照してください。

OSPF 固有の要件は、次のコマンドを使用して、近隣ルータの変更を syslog メッセージによって記録するよう OSPF ルータを設定することです。

```
OSPF_ROUTER(config)# ospf log-adj-changes
```

Cisco IOS CLI データの収集

通常、NE によって蓄積された未加工の情報に最も直接的にアクセスできるのは Cisco IOS CLI です。 しかし、CLI によるアクセスは、この手順で定義されているグローバルな構成管理よりも、トラブルシューティングや変更管理のアクティビティにより適しています。 CLI によるアクセスは、大規模なネットワークの管理には適していません。 その場合は自動情報アクセスが必要になります。

このバージョンの OSPF 構成管理手順では、CLI 構成とデータは必要ありません。

データ表示

OSPF エリア レポート

OSPF エリア レポートの形式の例を次に示します。 このレポートの形式は、商用 NMS のいずれかが使用されていればその機能によって、あるいは設計されているカスタム スクリプトの出力によって決まります。

エリア	データ フィールド	前回の実行	今回の実行
エリア ID #1	認証		
	SPF の実行		
	ABR の回数		
	ASBR の回数		
	LSA の回数		
	LSA のチェックサム		
	アドレス エラー		
	ルーティングの廃棄		
	ルートが見つからない		
エリア ID #n	認証		
	SPF の実行		
	ABR の回数		
	ASBR の回数		
	LSA の回数		
	LSA のチェックサム		
	アドレス エラー		
	ルーティングの廃棄		
	ルートが見つからない		

OSPF インターフェイス レポート

OSPF インターフェイス レポートの形式の例を次に示します。 実際には、このレポートの形式は、商用 NMS のいずれかが使用されていればその機能によって、あるいは設計されているカスタム スクリプトの出力によって決まります。

エリア	デバイス	Interface	データ フィールド	前回の実行	今回の実行
エリア ID #1	ノード ID #1	インターフェイス ID #1	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		
		インターフェイス ID #n	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		
	ノード ID #n	インターフェイス ID #1	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		
		インターフェイス ID #n	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		
エリア ID #n	ノード ID #1	インターフェイス ID #1	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		
		インターフェイス ID #n	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		
	ノード ID #n	インターフェイス ID #1	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コス		

			ト/タイマー		
		インターフェイス ID #n	IP アドレス		
			エリア ID		
			管理状態		
			OSPF の状態		
			メトリック/コスト/タイマー		

OSPF 隣接ルータ レポート

OSPF 近隣ルータ レポートの形式の例を次に示します。 実際には、このレポートの形式は、商用 NMS のいずれかが使用されていればその機能によって、あるいは設計されているカスタム スクリプトの出力によって決まります。

エリア	デバイス	ネイバー	データ フィールド	前回の実行	今回の実行
エリア ID #1	ノード ID #1	近隣ルータ ID #1	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		
		近隣ルータ ID #n	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		
	ノード ID #n	近隣ルータ ID #1	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		
		近隣ルータ ID #n	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		

			再送信キュー		
エリア ID #n	ノード ID #1	近隣ルータ ID #1	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		
		近隣ルータ ID #n	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		
	ノード ID #n	近隣ルータ ID #1	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		
		近隣ルータ ID #n	ルータ ID		
			ルータの IP アドレス		
			State		
			イベント		
			再送信キュー		

商用および一般的なインターネット監視ツール

商用ツールの目的は、syslog 情報の収集と処理を支援することと、一般的な SNMP MIB 変数のポーリングを収集することです。

この手順で定義した OSPF 構成管理をサポートする商用または公的なインターネット監視ツールは知られていません。 したがって、ローカルなカスタム スクリプトや手順が必要になります。

SNMP ポーリング データ

ルート テーブル RFC 1213

Object Name	オブジェクトの説明
ipRouteDest	このルートの宛先 IP アドレス。 0.0.0.0 の値を持つエントリは、デフォルト ルートと見なされます。 Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use. :: = { ipRouteEntry 1 } オブジェクト識別子 = 1.3.6.1.2.1.4.21.1.1
ipRouteMask	ipRouteDest フィールドの値と比較される前に、宛先アドレスに対して論理的にマスクがけを行うことを示します。 任意のサブネット マスクをサポートしていないシステムの場合、エージェントにより、ipRouteDest フィールドがクラス A、B、C のいずれのネットワークに属するかによって、次のマスク ネットワークの中から ipRouteMask の値が構築されます。 • クラス A = 255.0.0.0 • クラス B = 255.255.0.0 • クラス C = 255.255.255.0 ipRouteDest の値が 0.0.0.0、すなわちデフォルト ルートの場合、マスクの値も 0.0.0.0 になります。 **注:** IP ルーティングのサブシステムは、すべて暗黙的にこのメカニズムを使用しています。 :: = { ipRouteEntry 11 } オブジェクト識別子 = 1.3.6.1.2.1.4.21.1.11
ipRouteNextHop	このルートのネクスト ホップの IP アドレス。 ブロードキャスト メディアによって実現されているインターフェイスへのルート境界の場合は、このフィールドの値はそのインターフェイスのエージェントの IP アドレスになります。 :: = { ipRouteEntry 7 } オブジェクト識別子 = 1.3.6.1.2.1.4.21.1.7
ipRouteIfIndex	このルートのネクスト ホップが通過するローカル インターフェイスを一意に識別するインデックス値。 このインターフェイスは、IfIndex の値で識別されるインターフェイスと同一です。 :: = { ipRouteEntry 2 } オブジェクト識別子 = 1.3.6.1.2.1.4.21.1.2

RFC 1213 の各種オブジェクト

Object Name	オブジェクトの説明
ipAdEntIfIndex	このエントリを受け入れるインターフェイスを一意に識別するインデックス値。 このインターフェイスは、IfIndex の値で識別されるインターフェイスと同一です。 :: = { ipAddrEntry 2 } オブジェクト識別子 = 1.3.6.1.2.1.4.20.1.2

ipInAddrErrors	IP ヘッダー内の IP アドレスがエントリに対して誤った宛先を指していたために廃棄された入力データグラムの数。 この数には、無効なアドレス (0.0.0.0) およびサポートされていないクラス アドレス (クラス E) も含まれます。 IP ゲートウェイでなく、データグラムを転送しないエンティティの場合、このカウンタには宛先アドレスがローカル アドレスでないために廃棄されたデータグラムが含まれます。 { ip 5 } オブジェクト識別子 = 1.3.6.1.2.1.4.5
ipRoutingDiscards	廃棄された有効なルーティング エントリの数。 このようなエントリを廃棄する理由のひとつとしては、バッファ スペースを他のルーティング エントリに対して開放したことが考えられます。 { ip 23 } オブジェクト識別子 = 1.3.6.1.2.1.4.23
ipOutNoRoutes	宛先に転送するためのルートが見つからなかったために廃棄された IP データグラムの数。 { ip 12 } オブジェクト識別子 = 1.3.6.1.2.1.4.12

RFC 1253 OSPF エリア テーブル

Object Name	オブジェクトの説明
ospfAreaID	エリアを一意に識別する 32 ビットの整数。 エリア ID 0.0.0.0 は、OSPF バックボーンのために使用されます。 :: = { ospfAreaEntry 1 } オブジェクト識別子 = 1.3.6.1.2.1.14.2.1.1
ospfAuthType	このエリアに対して指定された認証タイプ。 その他の認証タイプは、エリアごとに口ーカルに割り当てることができます。 デフォルト値は 0 です。:: = { ospfAreaEntry 2 } オブジェクト識別子 = 1.3.6.1.2.1.14.2.1.2
OspfSpfRuns	エリア内のルート テーブルが、このエリアのリンクステート データベースを使用して計算された回数。 オブジェクト識別子 = 1.3.6.1.2.1.14.2.1.4
ospfAreaBdrRtrCount	このエリア内で到達可能な ABR の回数。 この値は最初はデフォルト値の 0 であり、各 SPF パスごとに計算されます。 :: = { ospfAreaEntry 5 } オブジェクト識別子 = 1.3.6.1.2.1.14.2.1.5
ospfASBdrRtrCount	このエリア内で到達可能な ABSR の回数。 この値は最初はデフォルト値の 0 であり、各 SPF パスごとに計算されます。 :: = { ospfAreaEntry 6 } オブジェクト識別子 = 1.3.6.1.2.1.14.2.1.6
ospfAreaLSACount	エリアのリンクステート データベース内の LSA の合計数。外部 LSA は含まれません。 デフォルト値は 0 です。:: = { ospfAreaEntry 7 } オブジェクト識別子 =

	1.3.6.1.2.1.14.2.1.7
ospfAreaLSACksumSum	エリアのリンクステート データベースに含まれる LSA の LS チェックサムの合計を表す 32 ビット符号なしの値。 この合計値には外部 (LS タイプ 5) の LSA は含まれません。 この合計値は、ルータのリンクステート データベースに変更があったかどうかを判断するためと、2 台のルータのリンクステート データベースを比較するために使用されます。 デフォルト値は 0 です。:: = { ospfAreaEntry 8 } オブジェクト識別子 = 1.3.6.1.2.1.14.2.1.8

RFC 1253 OSPF インターフェイス テーブル

Object Name	オブジェクトの説明
OspfIfIpAddress	OSPF インターフェイスの IP アドレス。 オブジェクト識別子 = 1.3.6.1.2.1.14.7.1.1
OspfIfEvents	OSPF インターフェイスが自身の状態を変更した回数、あるいはエラーが発生した回数。 オブジェクト識別子 = 1.3.6.1.2.1.14.7.1.15
OspfIfState	OSPF インターフェイスの状態。 オブジェクト識別子 = 1.3.6.1.2.1.14.7.1.12

RFC 1253 OSPF ネイバー テーブル

Object Name	オブジェクトの説明
OspfNbrIpAddr	この隣接ルータの IP アドレス。 :: = { ospfNbrEntry 1 } オブジェクト識別子 = 1.3.6.1.2.1.14.10.1.1
ospfNbrAddressLessIndex	IP アドレスを持たないインデックスで、インターネット標準の MIB の IfIndex に対応する値。 列を作成するときに、そのインスタンスから得ることができます。 :: = { ospfNbrEntry 2 } オブジェクト識別子 = 1.3.6.1.2.1.14.10.1.2
ospfNbrRtrId	IpAddress として表現される 32 ビットの整数で、自律システムでの近隣ルータを一意に識別します。 デフォルト値は 0.0.0.0 です。 :: = { ospfNbrEntry 3 } オブジェクト識別子 = 1.3.6.1.2.1.14.10.1.3
ospfNbrState	近隣ルータとの関係の状態。 状態には次のものがあります。 • down (1) • attempt (2) • init (3) • twoWay (4) • exchangeStart (5) • exchange (6) • loading (7) • full (8) :: = { ospfNbrEntry 6 } オブジェクト識別子

	= 1.3.6.1.2.1.14.10.1.6
ospfNbrEvents	近隣ルータの関係により状態が変化したり、エラーが発生した回数。 デフォルト値は 0 です。:: = { ospfNbrEntry 7 } オブジェクト識別子 = 1.3.6.1.2.1.14.10.1.7
ospfNbrLSRetransQLen	再送信キューの現在の長さ。 デフォルト値は 0 です。:: = { ospfNbrEntry 8 } オブジェクト識別子 = 1.3.6.1.2.1.14.10.1.8

データ収集アルゴリズムの例

このドキュメントの調査で、プロトタイプとなる C プログラムを開発しました。 このプログラムは oscan といい、Microsoft Developer Studio 97 と Visual C++ バージョン 5.0 を使用して作成されました。 このプログラムでは、SNMP 関数の application programming interface (API; アプリケーション プログラミング インターフェイス) を提供する固有のライブラリが 2 種類使用されています。 そのライブラリは、snmpapi.lib と mgmtapi.lib です。

この Microsoft API によって提供される関数は 3 つの主要なカテゴリに分類され、次の表に一覧されています。

エージェント関数	マネージャ 関数	ユーティリティ関数
SnmpExtensionInit SnmpExtensionInitEx SnmpExtensionQuery SnmpExtensionTrap	SnmpMgrClose SnmpMgrGetTrap SnmpMgrOidToStr SnmpMgrOpen SnmpMgrRequest SnmpMgrStrToOid SnmpMgrTrapListen	SnmpUtilMemAlloc SnmpUtilMemFree SnmpUtilMemReAlloc SnmpUtilOidAppend SnmpUtilOidCmp SnmpUtilOidCpy SnmpUtilOidFree SnmpUtilOidNCmp SnmpUtilPrintAsnAny SnmpUtilVarBindCpy SnmpUtilVarBindListCpy SnmpUtilVarBindFree SnmpUtilVarBindListFree

oscan プロトタイプ コードでは、次に一覧されている一連の関数に、Microsoft API をカプセル化しています。

- snmpWalkStrOid
- snmpWalkAsnOid
- snmpWalkVarBind
- snmpWalkVarBindList

これらの関数は、OSPF 構成データを維持するために使用される各種の SNMP MIB テーブルにアクセスできるようにする一般的な API を提供しています。 アクセスされるテーブルの object identifier (OID; オブジェクト識別子) は、テーブル固有のコール バック関数と共に oscan の API に渡されます。 コール バック関数には、テーブルから返されたデータを処理するための情報が含まれます。

メイン ルーチン

最初の作業は、oscan プログラムのターゲットとなるノードの一覧を構築することです。「デバイス検出」の問題を回避するには、スキャンされるノードを識別するためのシード ファイルが必要です。 シード ファイルは、IP アドレスや SNMP の 読み取り専用コミュニティ ストリングのような情報を提供します。

oscan プログラムでは、ルータによって収集された SNMP 情報を保存するための、いくつかの内部データ構造を維持する必要があります。 一般的には、収集されたそれぞれの SNMP MIB テーブル用の内部データ構造があります。

```
Main
load node array based on information in the seed file.
while more entries in the node array
start SNMP session for this node
collect IP route table for this node
collect OSPF area table for this node
collect OSPF Neighbor table for this node
collect  sysName for this node
collect OSPF Interface table for this node
end SNMP session for this node
end while
```

IP ルート テーブル

この操作はルータの CPU に負荷をかけやすいため、SNMP によって IP ルート テーブルにアクセスする際は十分に注意してください。 また、この理由から、oscan プログラムではユーザによる設定が可能な遅延パラメータを使用しています。 このパラメータを使用することによって、各 SNMP 要求間での遅延が設定できます。 大規模な環境の場合は、このために情報の収集にかかる総時間が非常に長くなる場合があります。

ルート テーブルには、oscan が関係する 4 つの情報があります。

- ipRouteDest
- ipRouteMask
- ipRouteNextHop
- ipRouteIfIndex

ルート テーブルは、ipRouteDest によって示されます。 したがって、SNMP **get-request** によって返される各オブジェクトの OID には ipRouteDest 付いています。

ipRouteIfIndex オブジェクトは、IP アドレス テーブル (ipAddrTable) を示す整数です。 ipAddrTable は、ipAdEntAddr オブジェクト (そのインターフェイスの IP アドレス) を使用して示されます。 インターフェイスの IP アドレスを取得するには、4 段階の処理が必要です。

1. ルーティング テーブルから ipRouteIfIndex を収集する。
2. ipRouteIfIndex を使用して ipAddrTable にアクセスし、パターン マッチングを行う。
3. パターンが見つかったら、OID をストリングに変換し、そのインターフェイスの IP アドレスとなる直前のドットで分割された 10 進数のフィールドを収集する。
4. インターフェイスの IP アドレスを IP ルート テーブルに戻す。

IP ルート テーブルにアクセスするための一般的なアルゴリズムを次に示します。 このとき、ipRouteIfIndex の整数値だけが保存されます。 この処理の後の方で、インターフェイス情報を収集するとき、ipAddrTable にアクセスが行われ、残りの情報が収集されて、内部 IP ルート テーブルに配置されます。

```
OID List =
ipRouteDestOID,
ipRouteMaskOID,
ipRouteNextHopOID,
ipRouteIfIndexOID;

For each object returned by  SNMP route table walk
Sleep // user configurable polling delay.
check varbind oid against OID list
if OID is ipRouteDestOID
add new entry in the internal route table array
if OID is one of the others
search internal route array for matching index value
store information in array
```

収集された情報は、次のように今までに馴染みのあるルータの CLI からの出力に似た表で表現されます。

```
ROUTE TABLE
*********************************************************
Destination     Mask                GW              Interface
10.10.10.4      255.255.255.252     10.10.10.5      10.10.10.5
10.10.10.16     255.255.255.252     10.10.10.6      10.10.10.5
10.10.10.24     255.255.255.252     10.10.10.25     10.10.10.25
10.10.10.28     255.255.255.252     10.10.11.2      10.10.11.1
10.10.10.36     255.255.255.252     10.10.10.6      10.10.10.5
10.10.11.0      255.255.255.0       10.10.11.1      10.10.11.1
10.10.13.0      255.255.255.0       10.10.11.2      10.10.11.1
```

OSPF エリア テーブル

OSPF エリア テーブルからの情報の収集は、OSPF エリア テーブル（ospfAreaTable）のスキャンと、返されたデータの処理によって行われます。 ospfAreaTable を示すものは osfpAreaId です。 ospfAreaId は、IP アドレスと同様の、ドットで区分された 10 進数の形式で保存されます。 したがって、ipRouteTable と ipRouteIfIndex を処理および検索するためのサブルーチンと同じサブルーチンがここで再利用できます。

このセクションの OSPF エリア テーブルには現在含まれていないデータ アイテムがいくつかあります。 たとえば、ipInAddrErrors、IpRoutingDiscards、および ipOutNoRoute オブジェクトは、MIB-2 の定義に入っていますが、OSPF エリアとは関連がありません。 これらのオブジェクトは、ルータと関連するものです。 したがって、これらのカウンターは、エリアの各ノードの値をエリア カウンターに追加することで、エリアのメトリックとして使用されます。 たとえば、OSPF エリア レポートでは、ルートが見つからないために廃棄されたパケットの数は、実際には、その エリア内の全ルータで廃棄されたパケットの合計数になります。 これはエリア内のルーティングの状態の概観を提供する高度なレベルのメトリックです。

```
OID List =
ipInAddrErrorsOID,
ipRoutingDiscardsOID,
ipOutNoRouteOID,
areaIdOID,
authTypeOID,
spfRunsOID,
abrCountOID,
asbrCountOID,
lsaCountOID,
```

```
lsaCksumSumOID;

For object returned from the SNMP walk of  the Area Table
Sleep // user configurable polling delay.
check varbind oid against OID list.
if OID is ospfAreaId
add new entry in the internal route table array
if OID one of the others
search internal array for matching index value
store information in array
end of for loop
get ipInAddrErrors, ipRoutingDiscards, ipOutNoRoute
add values to overall Area counters
```

収集された情報は、次の ASCII の表で表現されます。

```
AREAS
*********************************************************
AREA = 0.0.0.0AREA = 0.0.0.2
authType = 0authType = 0
spfRuns = 38spfRuns = 18
abrCount = 2abrCount = 1
asbrCount = 0asbrCount = 0
lsaCount = 11lsaCount = 7
lsaCksumSum = 340985lsaCksumSum = 319204
ipInAddrErrors = 0   ipInAddrErrors = 0
ipRoutingDiscards = 0ipRoutingDiscards = 0
ipOutNoRoutes = 0ipOutNoRoutes = 0
```

OSPF ネイバー テーブル

近隣ルータ テーブルのインデックスには、次の 2 つの値があります。

- ospfNbrIpAddr：ospfNbrIpAddr は近隣ルータの IP アドレスです。
- ospfNbrAddressLessIndex：ospfNbrAddressLessIndex は、次の 2 つの値のいずれかをとります。IP アドレスが割り当てられているインターフェイスの場合、0。IP アドレスが割り当てられていないインターフェイスの場合、インターネット標準の MIB から IfIndex として変換された値。

このインデックスには 2 つの値があるため、これより前に使用した、返された OID に外部情報を付加したアルゴリズムを調整する必要があります。 調整を行った後、ipRouteTable と ipRouteIfIndex を処理および検索するためのサブルーチンと同じサブルーチンがここで再利用できます。

```
OID List =
ospfNbrIpAddrOID,
ospfNbrAddressLessIndexOID,
ospfNbrRtrIdOID,
ospfNbrStateOID,
ospfNbrEventsOID,
ospfNbrLSRetransQLenOID,

For object returned from the SNMP walk of the Neighbor Table
Sleep // user configurable polling delay.
check varbind OID against OID list.
if OID matches ospfNbrIpAddr
add new entry in the internal neighbor table array
```

```
if OID matches one of the others
search array for matching index value
store information in array
```

収集された情報は、次の ASCII の表で表現されます。

```
NEIGHBORS
************************************************************
NEIGHBOR #0NEIGHBOR #1
Nbr Ip Addr = 10.10.10.6Nbr Ip Addr = 10.10.11.2
Nbr Rtr Id = 10.10.10.17Nbr Rtr Id = 10.10.10.29
Nbr State = 8Nbr State = 8
Nbr Events = 6Nbr Events = 30
Nbr Retrans = 0Nbr Retrans = 0
```

関連情報

- OSPF 設定ガイド
- RFC 1246 Experience with the OSPF Protocol
- RFC 1245 OSPF Protocol Analysis
- RFC 1224 Techniques for Managing Asynchronously Generated Alerts
- OSPF に関するサポート ページ
- IP ルーティングに関するサポート ページ
- テクニカルサポート - Cisco Systems